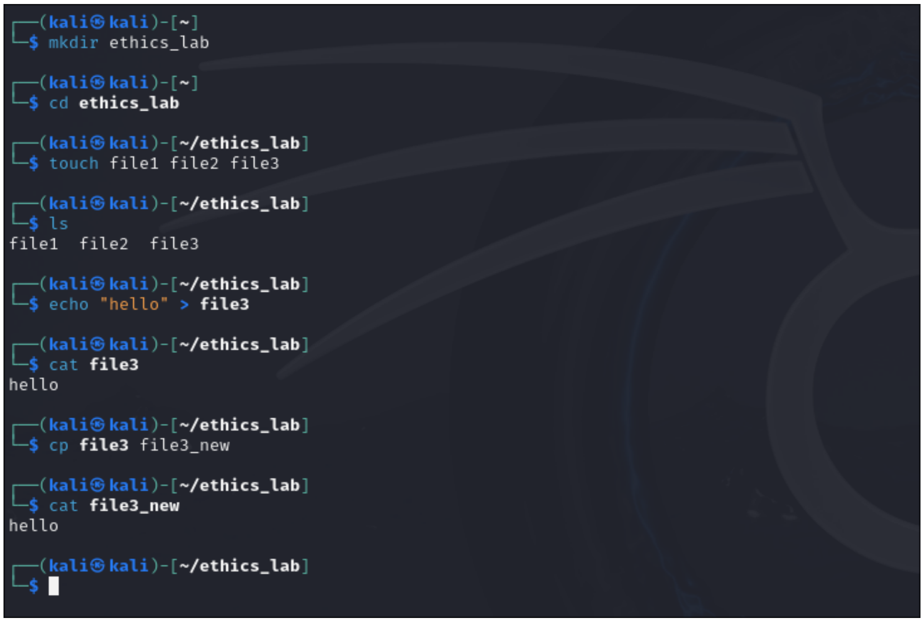
Ethical Hacking in ECE Assignment 2 - Aidan Sharpe

Assignment Tasks

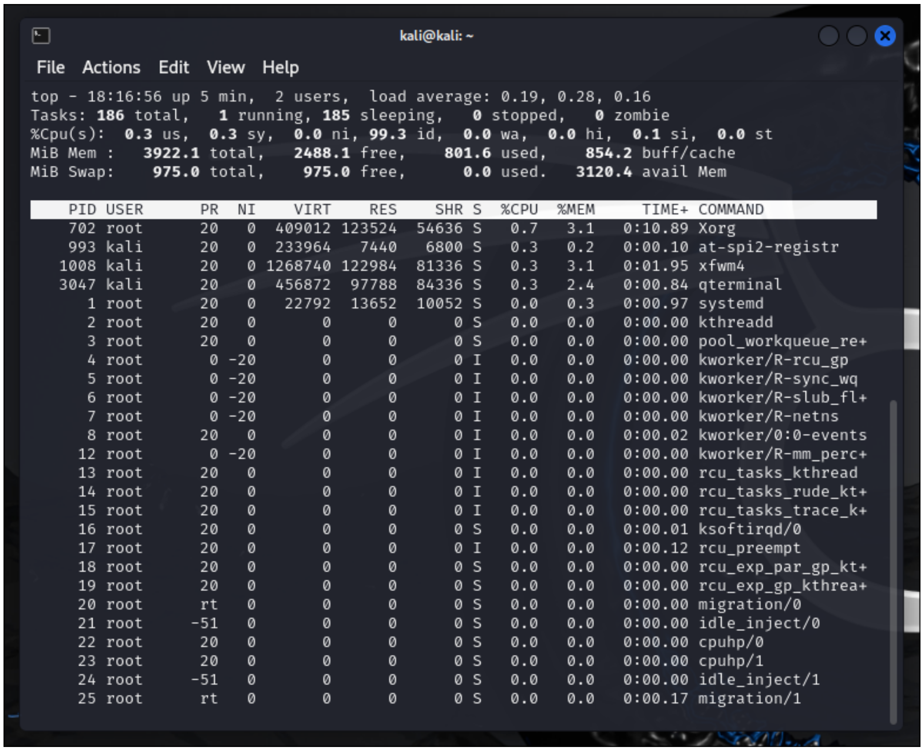
Task 1

Create a directory named ethics_lab and manage files within it using commands like mkdir, cd, mv, and touch.



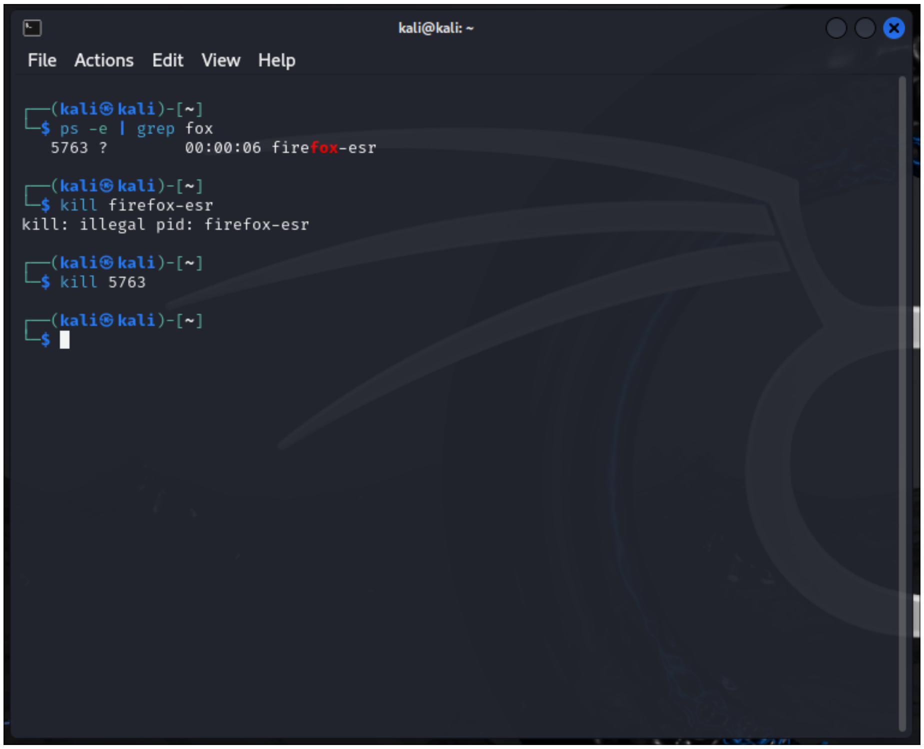
```
┌──(kali㉿kali)-[~]
└─$ mkdir ethics_lab

┌──(kali㉿kali)-[~]
└─$ cd ethics_lab

┌──(kali㉿kali)-[~/ethics_lab]
└─$ touch file1 file2 file3

┌──(kali㉿kali)-[~/ethics_lab]
└─$ ls
file1  file2  file3

┌──(kali㉿kali)-[~/ethics_lab]
└─$ echo "hello" > file3

┌──(kali㉿kali)-[~/ethics_lab]
└─$ cat file3
hello

┌──(kali㉿kali)-[~/ethics_lab]
└─$ cp file3 file3_new

┌──(kali㉿kali)-[~/ethics_lab]
└─$ cat file3_new
hello

┌──(kali㉿kali)-[~/ethics_lab]
└─$ █
```

Create a directory called ethics_lab using the mkdir command. Then navigate to ethics_lab using cd. Create three files (file1, file2, and file3) using the touch command. Show the contents of the ethics_lab directory using ls. Write the text "hello" into file3 using echo and the write to file symbol (>). Print the contents of file3 to the terminal using the cat command. ### Task 2 Monitor and manage processes on your system using tools like ps, top, and kill. Identify and terminate a process.



```
top - 18:16:56 up 5 min,  2 users,  load average: 0.19, 0.28, 0.16
Tasks: 186 total,   1 running, 185 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st
MiB Mem :   3922.1 total,   2488.1 free,    801.6 used,    854.2 buff/cache
MiB Swap:    975.0 total,    975.0 free,      0.0 used.   3120.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    702 root      20   0  409012 123524  54636 S   0.7   3.1   0:10.89 Xorg
    993 kali      20   0  233964   7440   6800 S   0.3   0.2   0:00.10 at-spi2-registr
   1008 kali      20   0 1268740 122984  81336 S   0.3   3.1   0:01.95 xfwm4
   3047 kali      20   0  456872  97788  84336 S   0.3   2.4   0:00.84 qterminal
      1 root      20   0   22792  13652  10052 S   0.0   0.3   0:00.97 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_re+
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_fl+
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
      8 root      20   0       0      0      0 I   0.0   0.0   0:00.02 kworker/0:0-events
     12 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_perc+
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kt+
     15 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_k+
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
     17 root      20   0       0      0      0 I   0.0   0.0   0:00.12 rcu_preempt
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_exp_par_gp_kt+
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_exp_gp_kthrea+
     20 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     21 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     22 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     23 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     24 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     25 root      rt   0       0      0      0 S   0.0   0.0   0:00.17 migration/1
```

View an active, sorted list of all tasks running on the system using `top`.

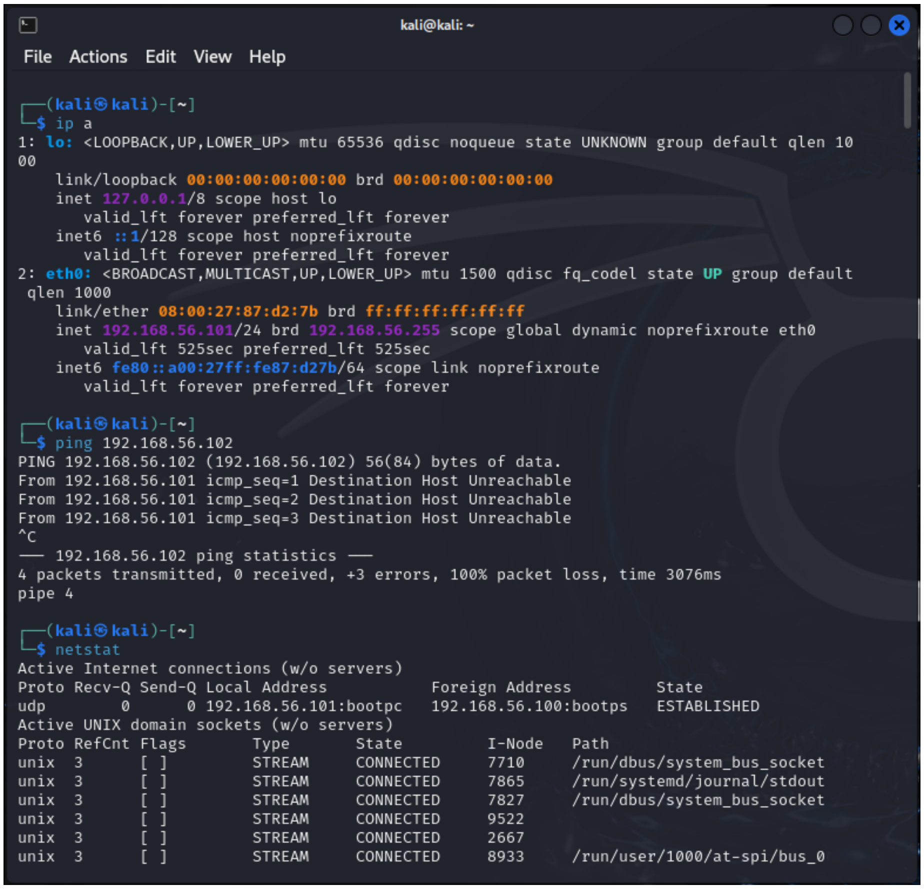


```
┌──(kali㉿kali)-[~]
└─$ ps -e | grep fox
   5763 ?        00:00:06 firefox-esr

┌──(kali㉿kali)-[~]
└─$ kill firefox-esr
kill: illegal pid: firefox-esr

┌──(kali㉿kali)-[~]
└─$ kill 5763

┌──(kali㉿kali)-[~]
└─$ 
```

List all tasks running on the system using `ps -e` then pass the output into `grep` and filter for the text "fox". This shows the information for the firefox-esr task. The number at the beginning of the line is the process identifier, and the process can be ended by executing `kill` followed by that process ID.

Task 3

Use `ifconfig`/`ip a`, `ping`, and `netstat`.

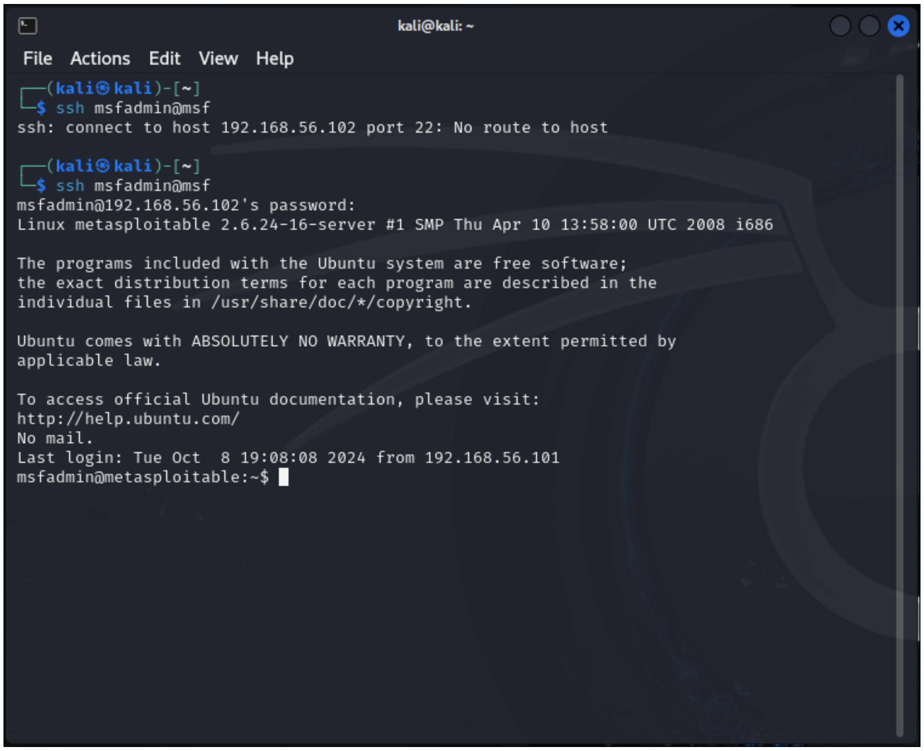


```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 10
00
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
 qlen 1000
    link/ether 08:00:27:87:d2:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
       valid_lft 525sec preferred_lft 525sec
    inet6 fe80::a00:27ff:fe87:d27b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
From 192.168.56.101 icmp_seq=1 Destination Host Unreachable
From 192.168.56.101 icmp_seq=2 Destination Host Unreachable
From 192.168.56.101 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.56.102 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3076ms
pipe 4

┌──(kali㉿kali)-[~]
└─$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 192.168.56.101:bootpc   192.168.56.100:bootps   ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  3      [ ]         STREAM     CONNECTED     7710     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     7865     /run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     7827     /run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     9522
unix  3      [ ]         STREAM     CONNECTED     2667
unix  3      [ ]         STREAM     CONNECTED     8933     /run/user/1000/at-spi/bus_0
```

Running `ip a` shows IP address information. For example, we can see that the current IP address is `192.168.56.101`. We also ran this command on another virtual machine and found that its IP address was `192.168.56.102`, so we can ping it using `ping` to see if a connection can be established. Running`netstat` shows a list of network connections.

Task 4

Set up a second VM and connect to it using `ssh`.

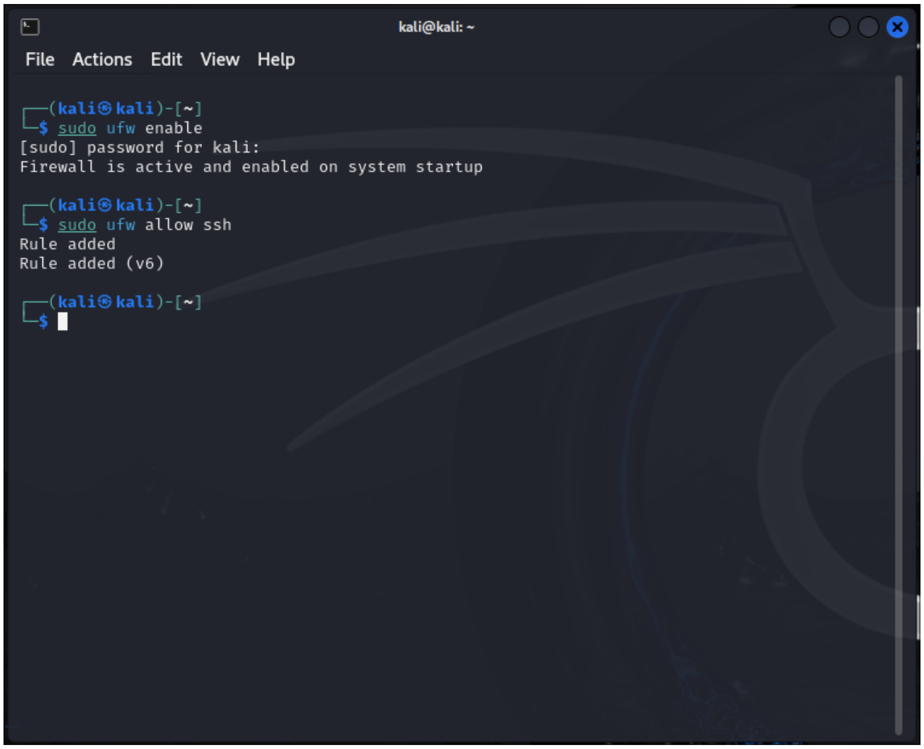


```
┌──(kali㉿kali)-[~]
└─$ ssh msfadmin@msf
ssh: connect to host 192.168.56.102 port 22: No route to host

┌──(kali㉿kali)-[~]
└─$ ssh msfadmin@msf
msfadmin@192.168.56.102's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Tue Oct  8 19:08:08 2024 from 192.168.56.101
msfadmin@metasploitable:~$ 
```

Prior to running the command, a host configuration was created called `msf`. This contained the known ip address and the preferred key algorithm. The command `ssh msfadmin@msf` creates a login tunnel for the user account `msfadmin` between the host and remote system.

Task 5

Configure the firewall using `ufw`. Enable the firewall and allow SSH traffic.



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
[sudo] password for kali:
Firewall is active and enabled on system startup

┌──(kali㉿kali)-[~]
└─$ sudo ufw allow ssh
Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$ 
```

Reflection

Learning basic Linux command is an important skill for ethical hacking for multiple reasons.

Reason 1 - Most Ethical Hacking Tools Run on Linux

Most ethical hacking tools run on or are designed specifically for Linux machines. Knowing how to better use the machines that your tools are running on is always advantageous.

Reason 2 - Most Servers Run Linux

Since over 90% of the servers on the internet run Linux, that means that most databases (the places where pretty much all valuable information is stored) are hosted on Linux-based servers. By being familiar with basic Linux commands, navigating remote server file systems becomes a much easier task.

Skills Gained from This Assignment

One skill I learned from this assignment is forcing a specific key algorithm for different hosts. While I have used SSH many times in the past, I was not aware that different machines restricted the type of keys used. Frankly, I thought they all used the same type of key. After completing this exercises, I am now practiced in the configuration of host key algorithms.