

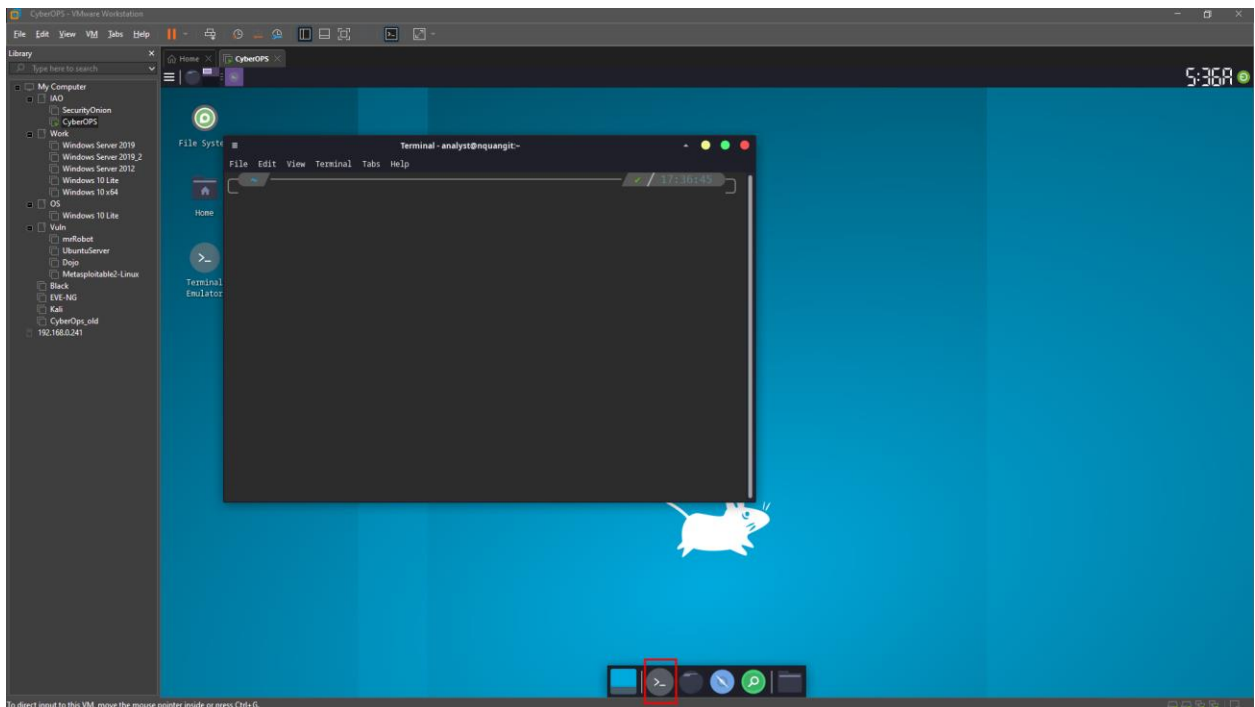
## Lab - Linux Servers

### Instructions

#### Servers

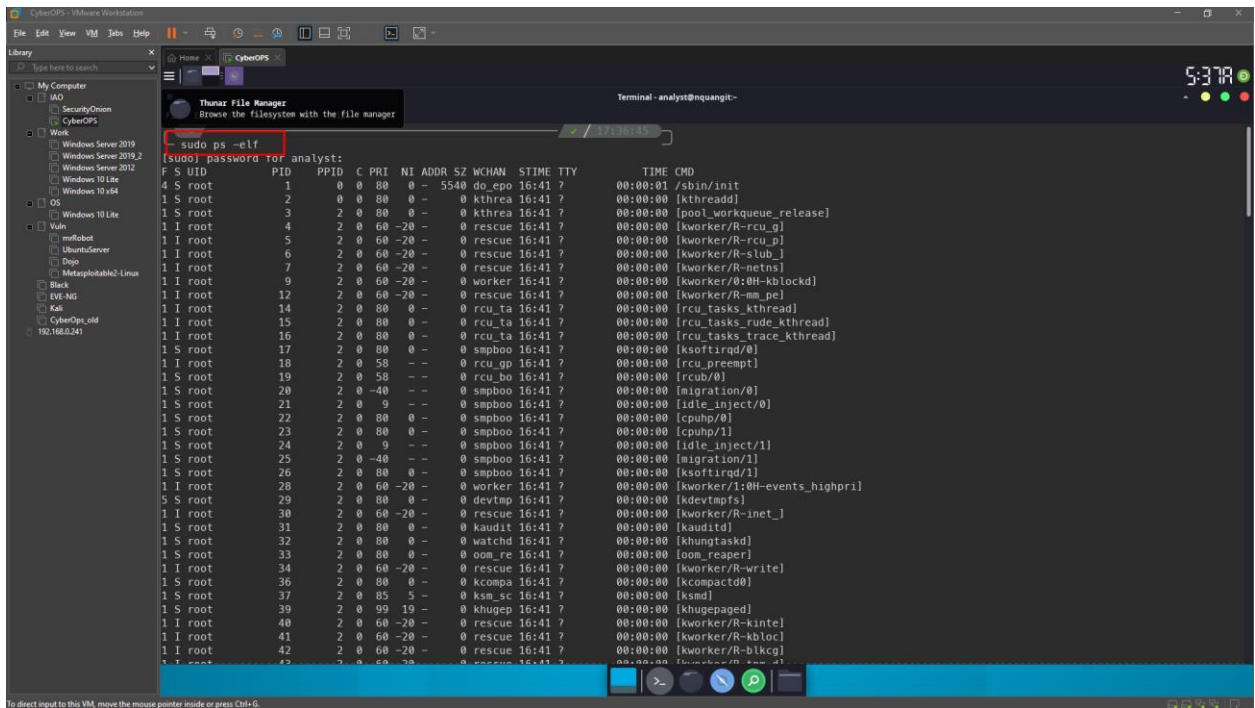
Access the command line.

- Log on to the CyberOps Workstation VM as the **analyst**, using the password **cyberops**. The account **analyst** is used as the example user account throughout this lab.
- To access the command line, click the **terminal** icon located in the Dock, at the bottom of VM screen. The terminal emulator opens.



Display the services currently running.

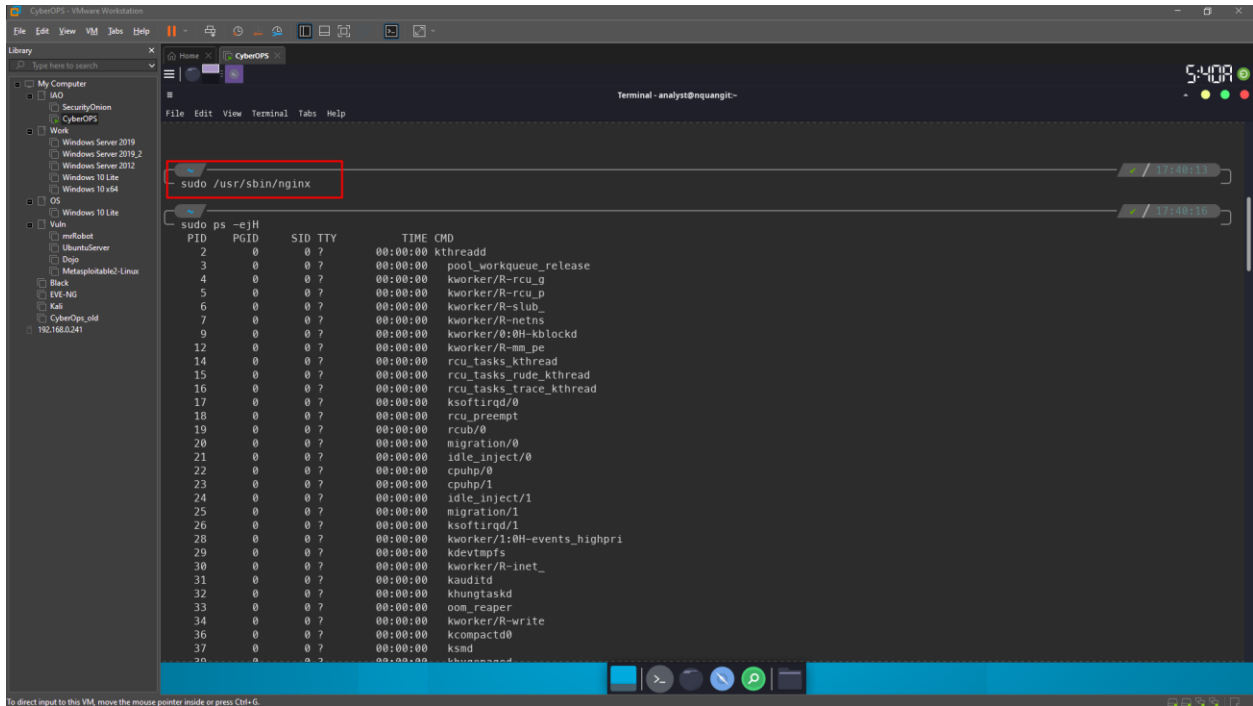
- Use the **ps** command to display all the programs running in the background:



Why was it necessary to run ps as root (prefacing the command with sudo)?

Because some processes do not belong to the current user and can not be shown if running ps as a normal user.

- b. In Linux, programs can also call other programs. The **ps** command can also be used to display such process hierarchy. Use **-ejH** options to display the currently running process tree after starting the nginx webserver with elevated privileges.



```

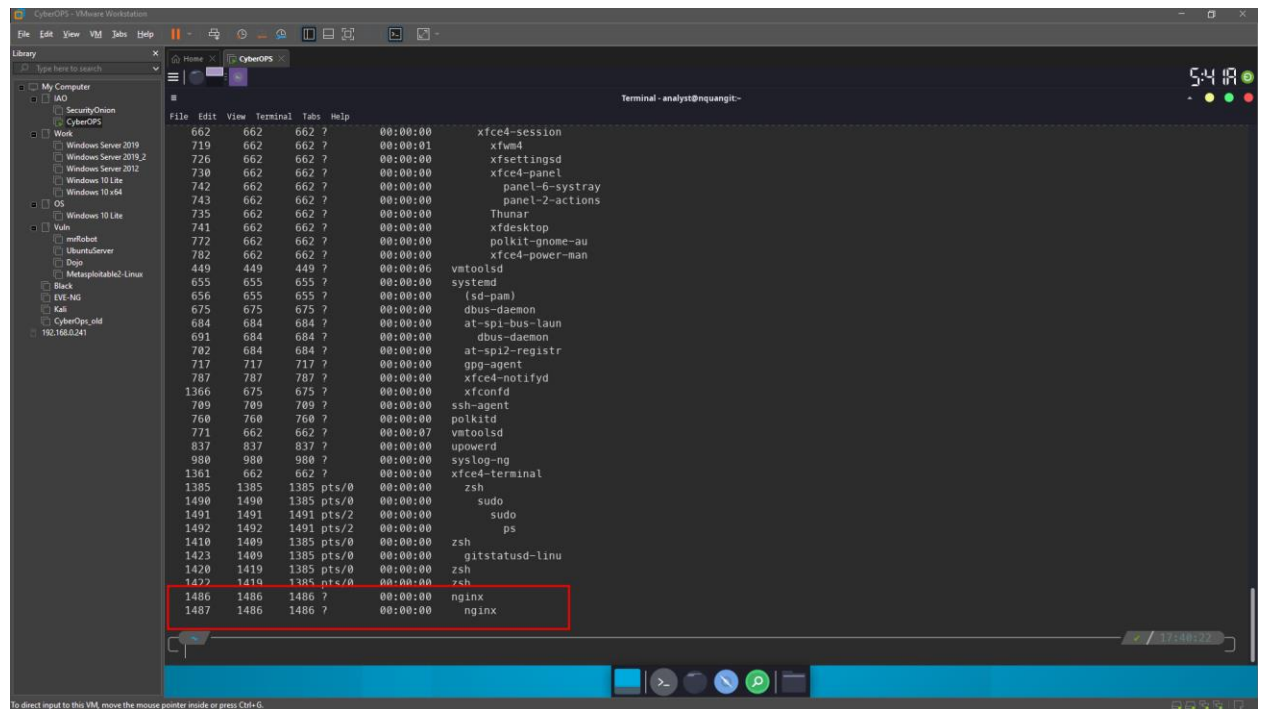
sudo /usr/sbin/nginx

sudo ps -eJH
  PID   PGID   SID TTY      TIME CMD
    2     0     0 ?        00:00:00 kthreadd
    3     0     0 ?        00:00:00 pool_workqueue_release
    4     0     0 ?        00:00:00 kworker/R-rcu_g
    5     0     0 ?        00:00:00 kworker/R-rcu_p
    6     0     0 ?        00:00:00 kworker/R-slub
    7     0     0 ?        00:00:00 kworker/R-netns
    9     0     0 ?        00:00:00 kworker/R-oh-ahblockd
   12     0     0 ?        00:00:00 kworker/R-mm_pe
   14     0     0 ?        00:00:00 rcu_tasks_kthread
   15     0     0 ?        00:00:00 rcu_tasks_rude_kthread
   16     0     0 ?        00:00:00 rcu_tasks_trace_kthread
   17     0     0 ?        00:00:00 ksoftirqd/0
   18     0     0 ?        00:00:00 rcu_preempt
   19     0     0 ?        00:00:00 rcub/0
   20     0     0 ?        00:00:00 migration/0
   21     0     0 ?        00:00:00 idle_inject/0
   22     0     0 ?        00:00:00 cpuhp/0
   23     0     0 ?        00:00:00 cpuhp/1
   24     0     0 ?        00:00:00 idle_inject/1
   25     0     0 ?        00:00:00 migration/1
   26     0     0 ?        00:00:00 ksoftirqd/1
   28     0     0 ?        00:00:00 kworker/1:0H-events_highpri
   29     0     0 ?        00:00:00 kdevtmpfs
   30     0     0 ?        00:00:00 kworker/R-lnet_
   31     0     0 ?        00:00:00 kauditd
   32     0     0 ?        00:00:00 khungtaskd
   33     0     0 ?        00:00:00 oom_reaper
   34     0     0 ?        00:00:00 kworker/R-write
   36     0     0 ?        00:00:00 kcompactd0
   37     0     0 ?        00:00:00 ksm
   38     0     0 ?        00:00:00 khugepaged

```

How is the process hierarchy represented by ps?

Uses indentation.

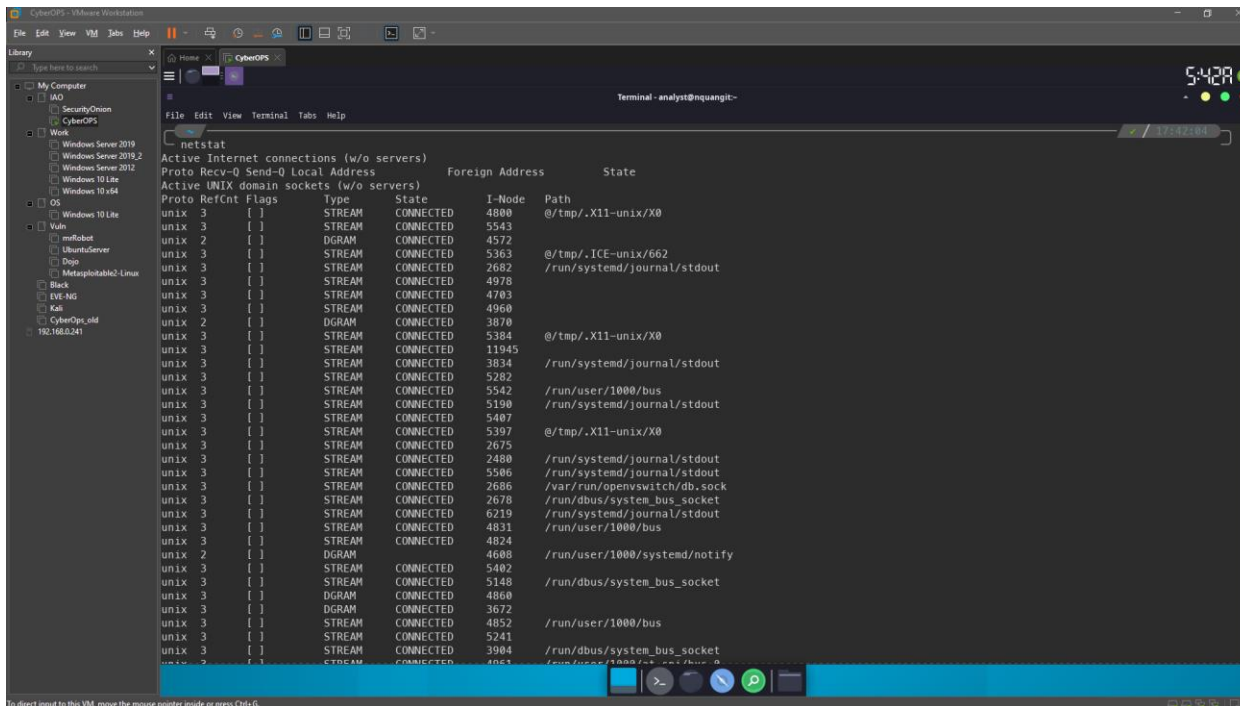


```

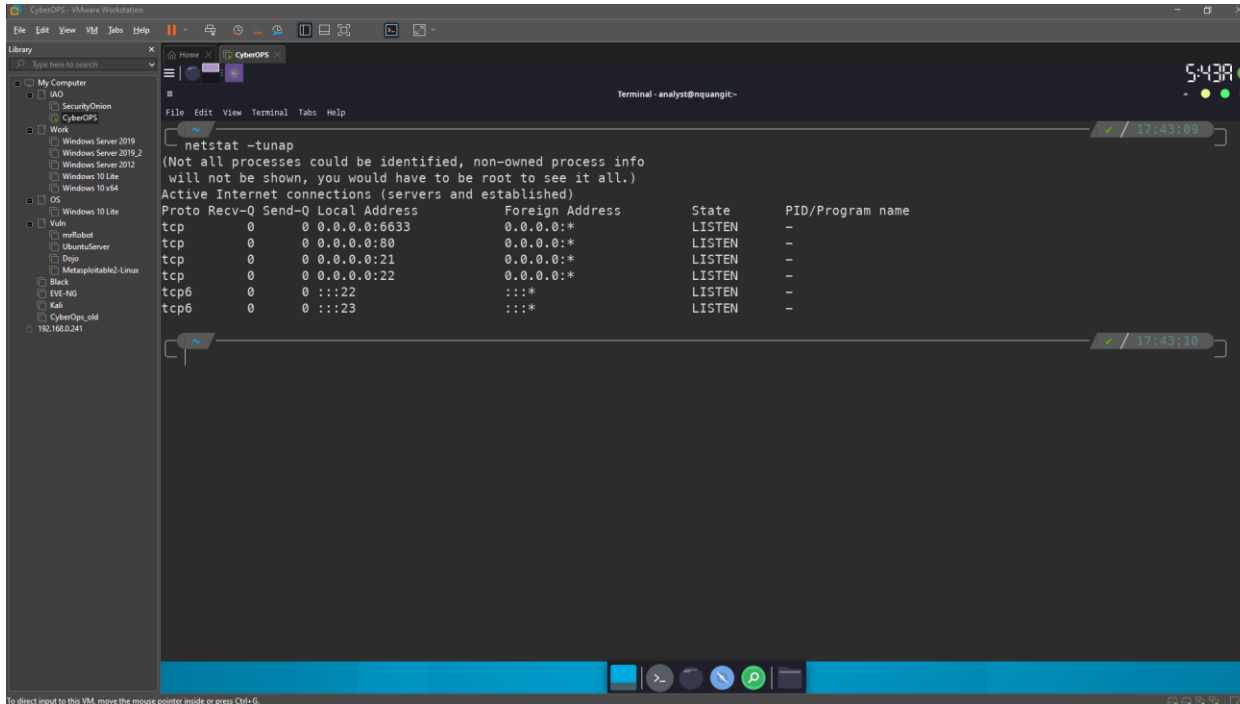
ps -eJH
  PID   PGID   SID TTY      TIME CMD
    662   662   662 ?        00:00:00 xfce4-session
    719   662   662 ?        00:00:01 xfwm4
    726   662   662 ?        00:00:00 xfsettingsd
    730   662   662 ?        00:00:00 xfce4-panel
    742   662   662 ?        00:00:00 panel-6-systray
    743   662   662 ?        00:00:00 panel-2-actions
    735   662   662 ?        00:00:00 Thunar
    741   662   662 ?        00:00:00 xfdesktop
    772   662   662 ?        00:00:00 polkit-gnome-au
    782   662   662 ?        00:00:00 xfce4-power-man
    449   449   449 ?        00:00:06 vmtoolsd
    655   655   655 ?        00:00:00 systemd
    656   655   655 ?        00:00:00 (sd-pam)
    675   675   675 ?        00:00:00 dbus-daemon
    684   684   684 ?        00:00:00 at-spi-bus-lau
    691   684   684 ?        00:00:00 dbus-daemon
    702   684   684 ?        00:00:00 at-spi2-registr
    717   717   717 ?        00:00:00 gpg-agent
    787   787   787 ?        00:00:00 xfce4-notifyd
    1366   675   675 ?        00:00:00 xfconfd
    789   789   789 ?        00:00:00 ssh-agent
    760   760   760 ?        00:00:00 polkitd
    771   662   662 ?        00:00:07 vmtoolsd
    837   837   837 ?        00:00:00 upowerd
    980   980   980 ?        00:00:00 syslog-ng
    1361   662   662 ?        00:00:00 xfce4-terminal
    1385   1385   1385 pts/0    00:00:00 zsh
    1490   1490   1385 pts/0    00:00:00 sudo
    1491   1491   1491 pts/2    00:00:00 sudo
    1492   1492   1491 pts/2    00:00:00 ps
    1410   1409   1385 pts/0    00:00:00 zsh
    1423   1409   1385 pts/0    00:00:00 gitstatusd-linu
    1420   1419   1385 pts/0    00:00:00 zsh
    1422   1419   1385 pts/0    00:00:00 zsh
    1486   1486   1486 ?        00:00:00 nginx
    1487   1486   1486 ?        00:00:00 nginx

```

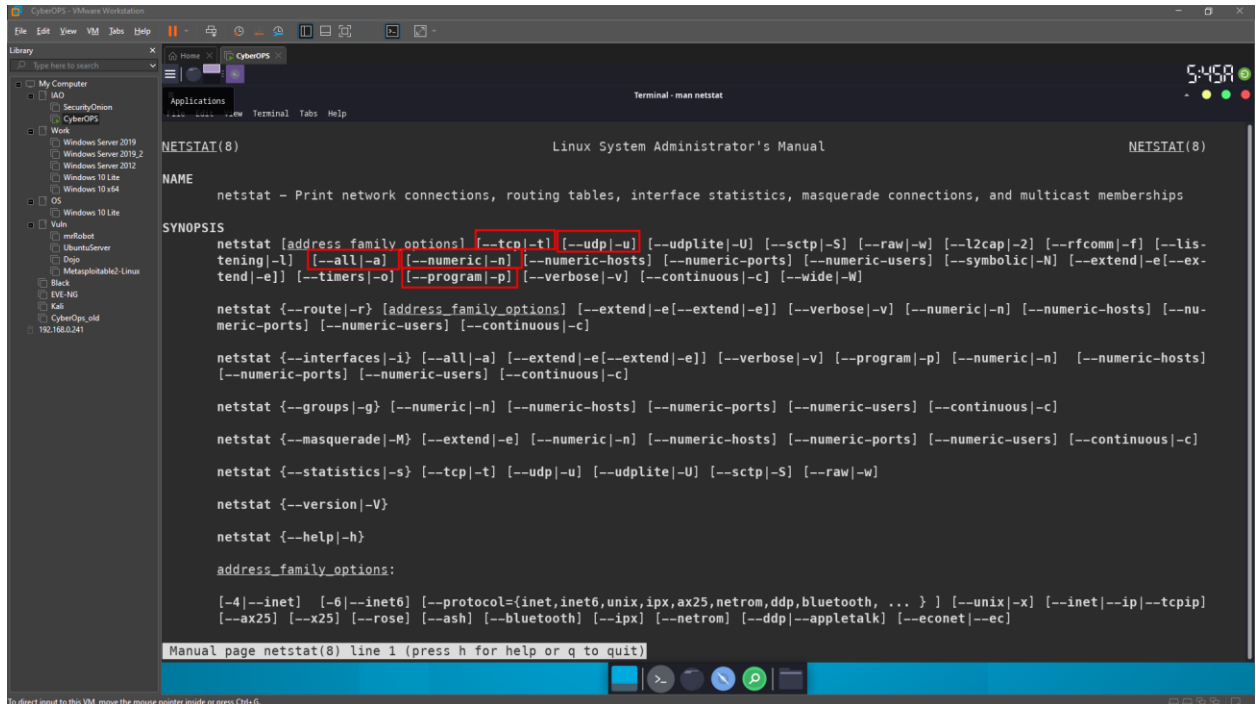
- c. As mentioned before, servers are essentially programs, often started by the system itself at boot time. The task performed by a server is called a *service*. In such fashion, a web server provides web services.



- d. Use **netstat** with the **-tunap** options to adjust the output of **netstat**. Notice that **netstat** allows multiple options to be grouped together under the same “-” sign.



What is the meaning of the -t, -u, -n, -a and -p options in netstat? (use man netstat to answer)



-t: list tcp services/processes/connections

-u: list udp services/processes/connections

-n: Show numerical addresses instead of trying to determine symbolic host, port or user names.

-a: Show both listening and non-listening sockets. With the --interfaces option, show interfaces that are not up

-p: Show the PID and name of the program to which each socket belongs. A hyphen is shown if the socket belongs to the kernel (e.g. a kernel service, or the process has exited but the socket hasn't finished closing yet).

Is the order of the options important to **netstat**?

No, the option order is irrelevant.

Based on the **netstat** output shown in item (d), what is the Layer 4 protocol, connection status, and PID of the process running on port 80?

While port numbers are just a convention, can you guess what kind of service is running on port 80 TCP?

- e. Sometimes it is useful to cross the information provided by **netstat** with **ps**. Based on the output of item (d), it is known that a process with **PID 395** is bound to TCP port 80. Port 395 is used in this example. Use **ps** and **grep** to list all lines of the **ps** output that contain **PID 395**. Replace 395 with the PID number for your particular running instance of nginx:

The process PID 395 is nginx. How could that be concluded from the output above?

What is **nginx**? What is its function? (Use google to learn about nginx)

The second line shows that process 396 is owned by a user named http and has process number 395 as its parent process. What does that mean? Is this common behavior?

Why is the last line showing grep 395?

## Using Telnet to Test TCP Services

- a. In Part 1, **nginx** was found to be running and assigned to port 80 TCP. Although a quick internet search revealed that **nginx** is a lightweight web server, how would an analyst be sure of that? What if an attacker changed the name of a malware program to **nginx**, just to make it look like the popular webserver? Use **telnet** to connect to the local host on port 80 TCP:
- b. Press a few letters on the keyboard. Any key will work. After a few keys are pressed, press ENTER. Below is the full output, including the Telnet connection establishment and the random keys pressed (fdsafsdaf, this case):  
  
Why was the error sent as a web page?
- c. Looking at the **netstat** output presented earlier, it is possible to see a process attached to port 22. Use Telnet to connect to it.

Use Telnet to connect to port 68. What happens? Explain.

## Reflection Questions

1. What are the advantages of using netstat?
2. What are the advantages of using Telnet? Is it safe?