NAME: Diana Wanjiru

ADMISSION NUMBER:CS-EH02-24103

LINK (Completed Module):

https://academy.hackthebox.com/achievement/1242115/18

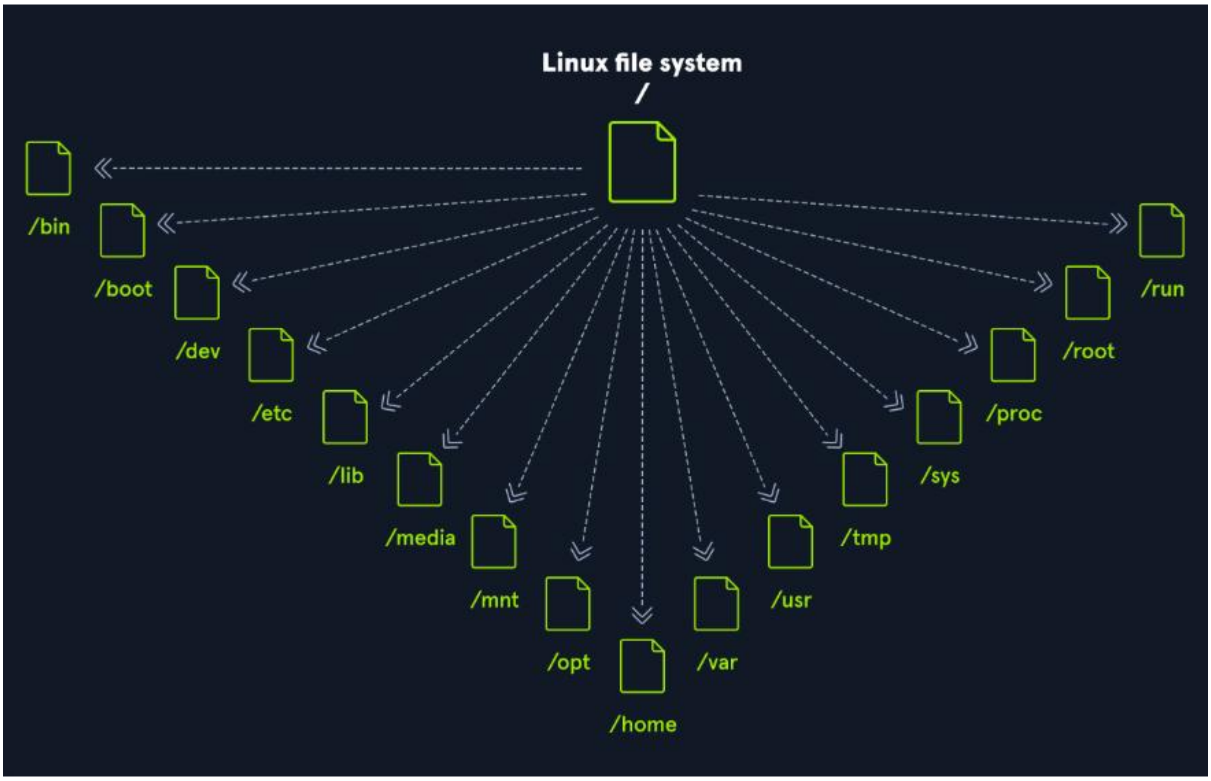
LINUX FUNDAMENTALS

Introduction

I understand that building a strong foundation in cybersecurity is the way to go. Linux being the first step, understanding the fundamentals strengthens my cyber security journey as it helps me to navigate through Linux terminal with ease.

1.1 Linux Structure:

I had a look at the definition of linux, the history of linux where I understood how linux has evolved over the years. Also, I had an in-depth look at philosophy, components, linux architecture and file system hierarchy of the linux Os where I got to understand the different directories. Here's a screenshot of the file system hierarchy.



1.2 Linux Distributions:

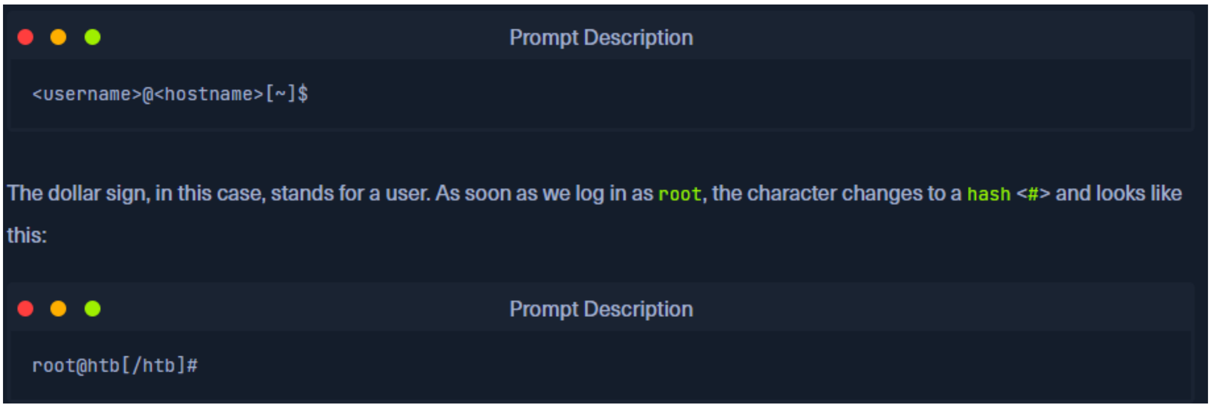
I got to understand that Linux distributions - or distros - are operating systems based on the Linux kernel and each Linux distribution is different, with its own set of features, packages, and tools. I looked at the different distros that are there and it stuck with me that Debian distro is a widely used and well-respected Linux distribution known for its stability and reliability.

1.3 Introduction to shell:

I learnt that knowing how to use the operating system to control it effectively requires understanding and mastering Linux's essential part, the Shell. The shell is also known as the terminal or the command line, and it provides a text-based input/output (I/O) interface between users and the kernel for a computer system. I understood about terminal emulation which is software that emulates the function of a terminal and finally about the different types of shell Bash being the commonly used one.

1.4 Prompt Description:

I looked at the **Unprivileged - User Shell** prompt and the **Privileged - Root Shell** prompt where one is represented with a dollar sign ($) and the other one with a hash tag (#) respectively. Here's a screenshot of both.



```
<username>@<hostname>[~]$
```

The dollar sign, in this case, stands for a user. As soon as we log in as `root`, the character changes to a `hash` <#> and looks like this:

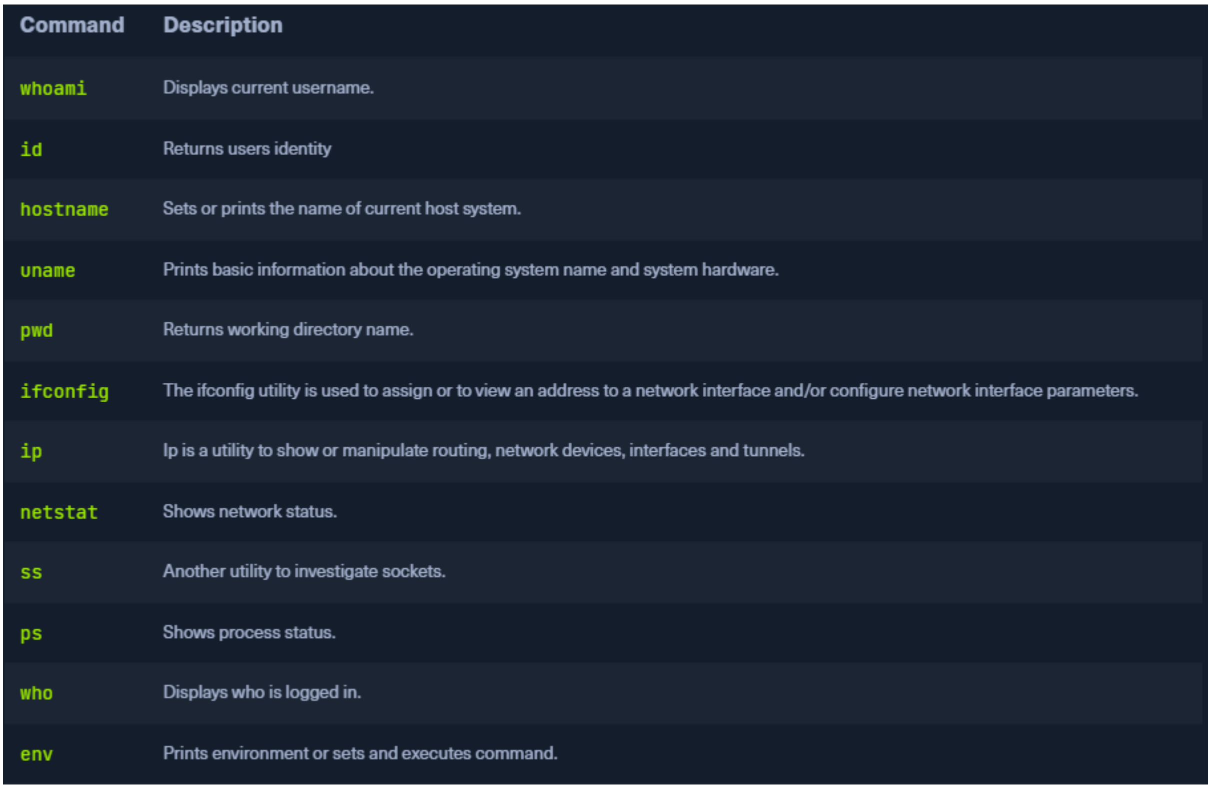
```
root@htb[/htb]#
```

1.5 Getting Help:

I understood the different ways I can get help in the terminal through various commands such as man <tool>, <tool> --help, <tool> -h and apropos <keyword>.

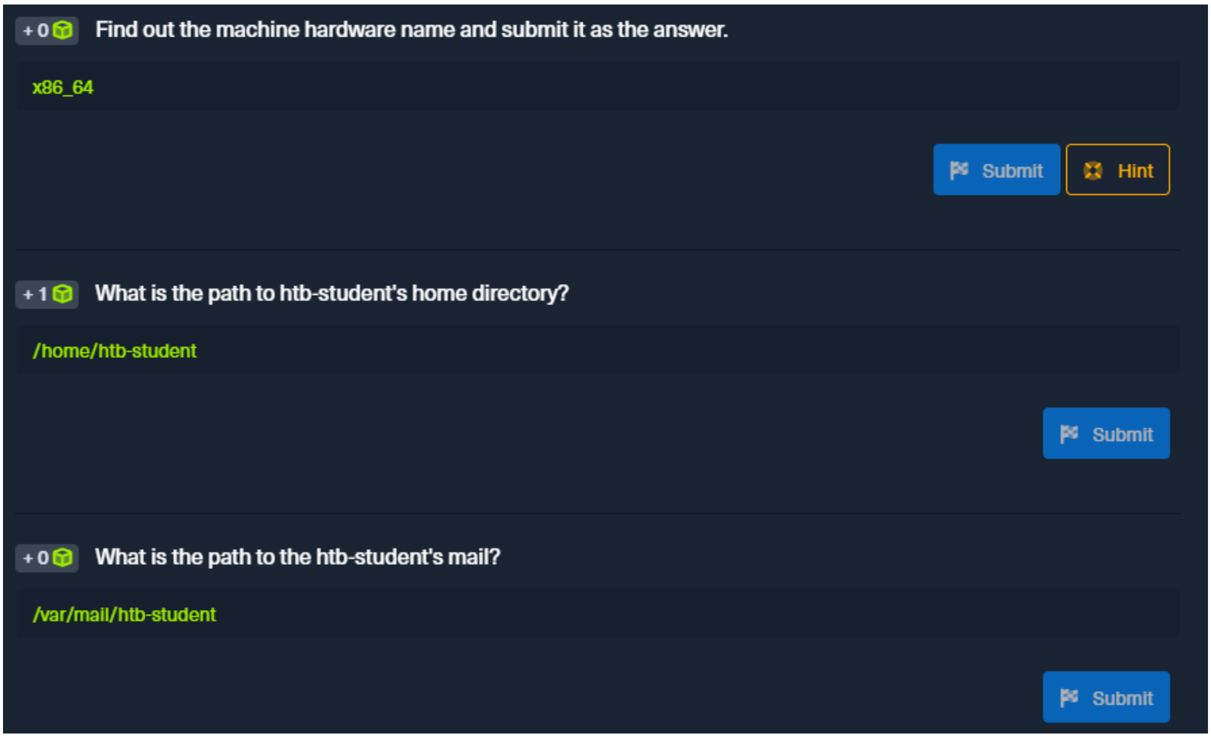
1.6 System Information:

I had a look into various commands used to get information about the system. Here's a screenshot of the commands with their descriptions.



Command	Description
`whoami`	Displays current username.
`id`	Returns users identity
`hostname`	Sets or prints the name of current host system.
`uname`	Prints basic information about the operating system name and system hardware.
`pwd`	Returns working directory name.
`ifconfig`	The ifconfig utility is used to assign or to view an address to a network interface and/or configure network interface parameters.
`ip`	Ip is a utility to show or manipulate routing, network devices, interfaces and tunnels.
`netstat`	Shows network status.
`ss`	Another utility to investigate sockets.
`ps`	Shows process status.
`who`	Displays who is logged in.
`env`	Prints environment or sets and executes command.

Also, looked at SSH command which is used to access and execute commands or actions on remote computers. It has the following syntax ssh username@ [IP address].
I also tackled the following questions. Here's a screenshot of the questions and answers I gave as well as screenshot of the process I used to get the ans



+ 0 Find out the machine hardware name and submit it as the answer.

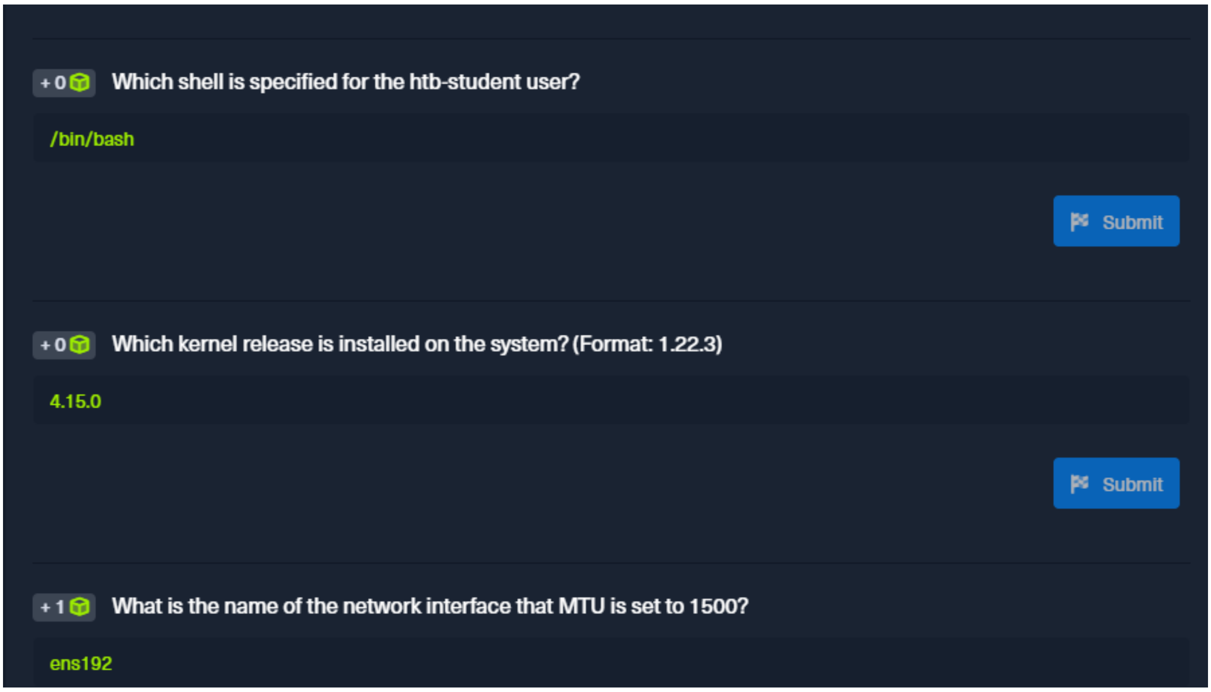
x86_64

+ 1 What is the path to htb-student's home directory?

/home/htb-student

+ 0 What is the path to the htb-student's mail?

/var/mail/htb-student



+ 0 Which shell is specified for the htb-student user?

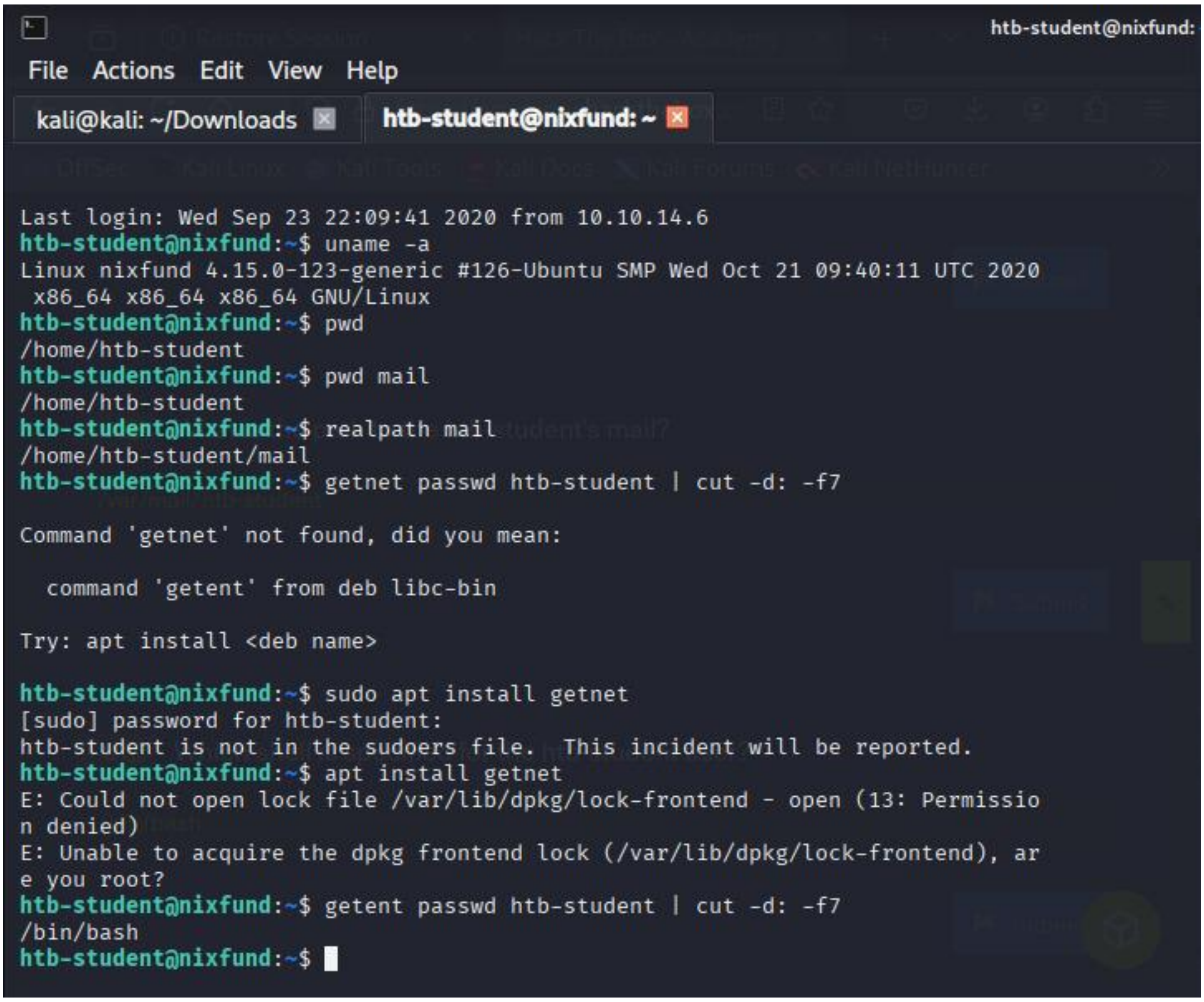
/bin/bash

+ 0 Which kernel release is installed on the system? (Format: 1.22.3)

4.15.0

+ 1 What is the name of the network interface that MTU is set to 1500?

ens192

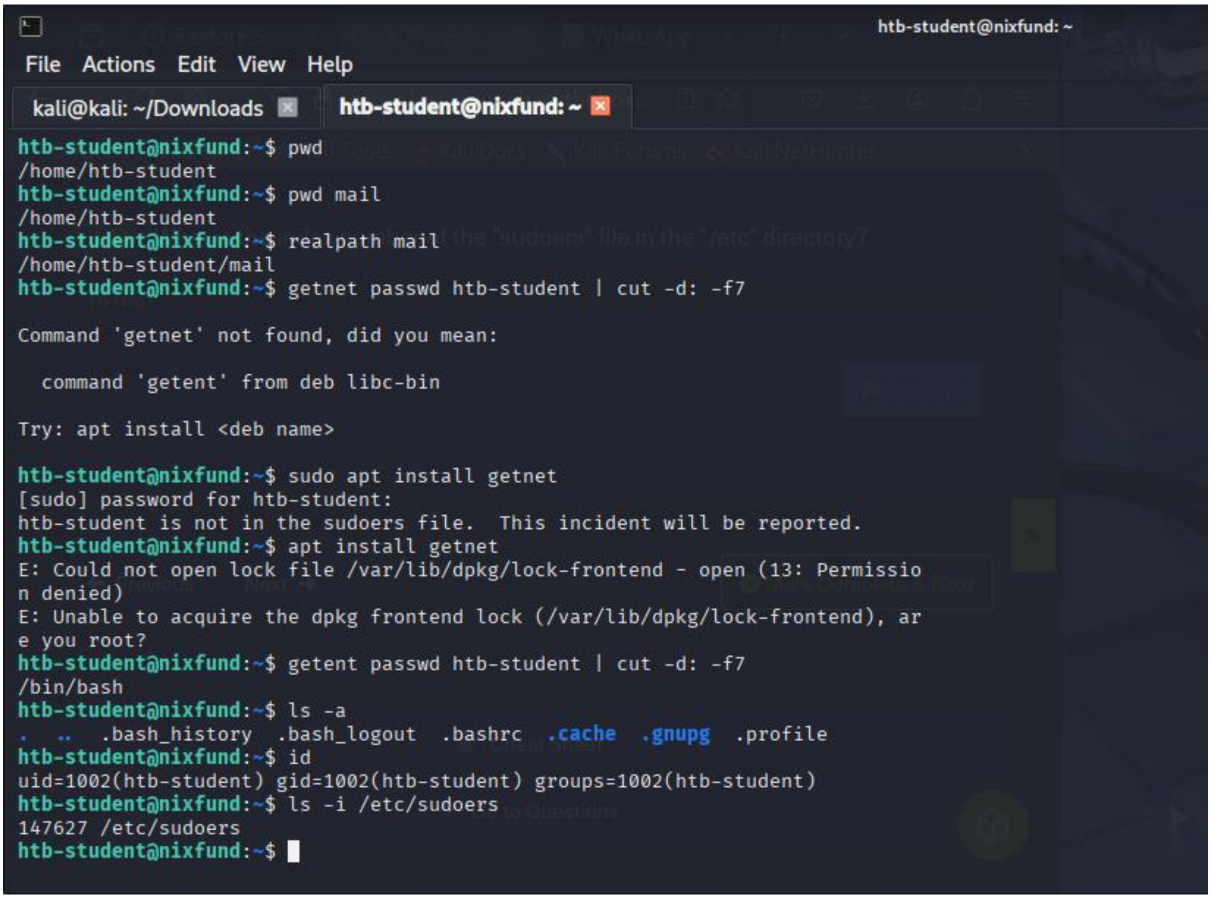


```
Last login: Wed Sep 23 22:09:41 2020 from 10.10.14.6
htb-student@nixfund:~$ uname -a
Linux nixfund 4.15.0-123-generic #126-Ubuntu SMP Wed Oct 21 09:40:11 UTC 2020
 x86_64 x86_64 x86_64 GNU/Linux
htb-student@nixfund:~$ pwd
/home/htb-student
htb-student@nixfund:~$ pwd mail
/home/htb-student
htb-student@nixfund:~$ realpath mail
/home/htb-student/mail
htb-student@nixfund:~$ getnet passwd htb-student | cut -d: -f7

Command 'getnet' not found, did you mean:

  command 'getent' from deb libc-bin

Try: apt install <deb name>

htb-student@nixfund:~$ sudo apt install getnet
[sudo] password for htb-student:
htb-student is not in the sudoers file.  This incident will be reported.
htb-student@nixfund:~$ apt install getnet
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permissio
n denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), ar
e you root?
htb-student@nixfund:~$ getent passwd htb-student | cut -d: -f7
/bin/bash
htb-student@nixfund:~$ █
```

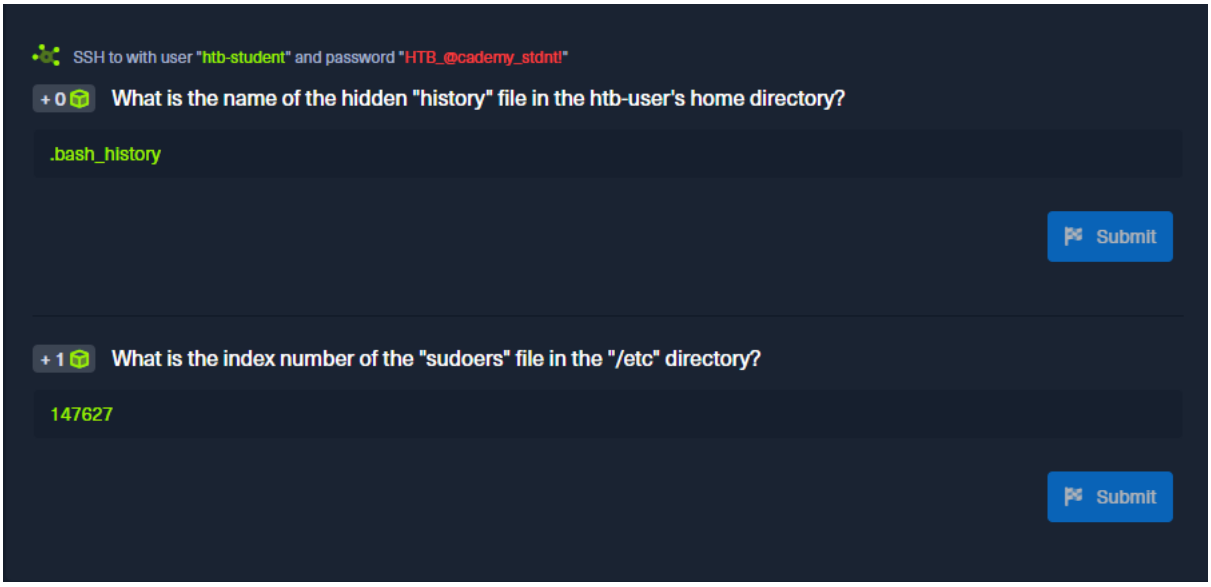
1.7 Navigation:

I have gone through various ways to navigate through linux through the terminal via commands such as pwd, cd and ls.

Got to answer the following questions and here's the screenshot.



```
htb-student@nixfund:~$ pwd
/home/htb-student
htb-student@nixfund:~$ pwd mail
/home/htb-student
htb-student@nixfund:~$ realpath mail
/home/htb-student/mail
htb-student@nixfund:~$ getnet passwd htb-student | cut -d: -f7

Command 'getnet' not found, did you mean:

  command 'getent' from deb libc-bin

Try: apt install <deb name>

htb-student@nixfund:~$ sudo apt install getnet
[sudo] password for htb-student:
htb-student is not in the sudoers file.  This incident will be reported.
htb-student@nixfund:~$ apt install getnet
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permissio
n denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), ar
e you root?
htb-student@nixfund:~$ getent passwd htb-student | cut -d: -f7
/bin/bash
htb-student@nixfund:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .gnupg  .profile
htb-student@nixfund:~$ id
uid=1002(htb-student) gid=1002(htb-student) groups=1002(htb-student)
htb-student@nixfund:~$ ls -i /etc/sudoers
147627 /etc/sudoers
htb-student@nixfund:~$ █
```



SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 0 What is the name of the hidden "history" file in the htb-user's home directory?

.bash_history

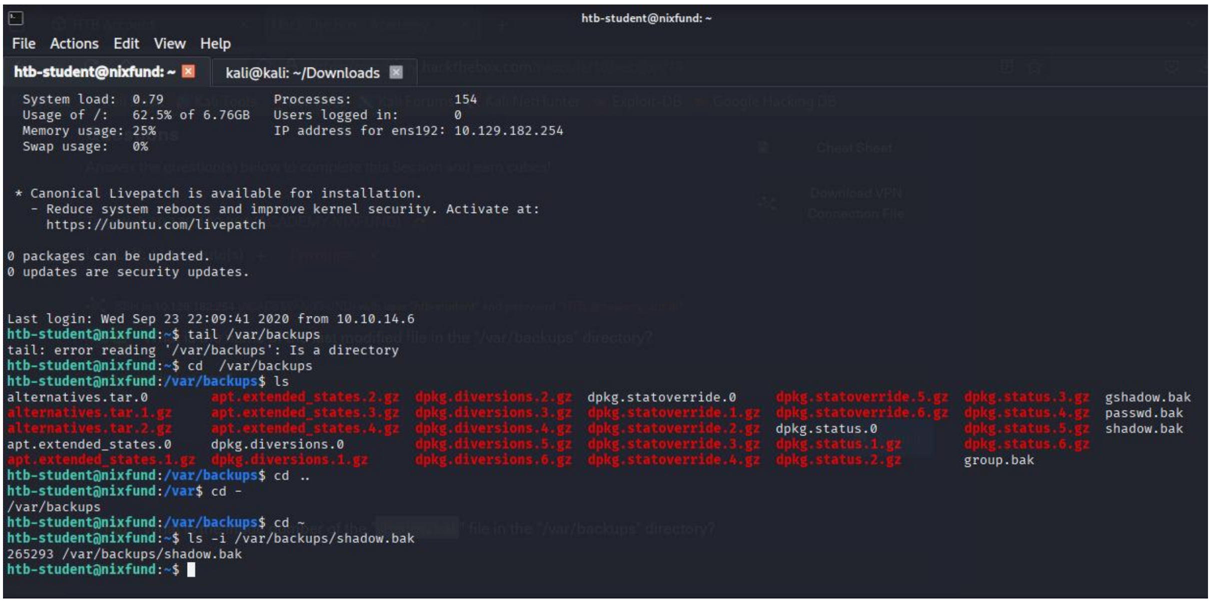
+ 1 What is the index number of the "sudoers" file in the "/etc" directory?

147627

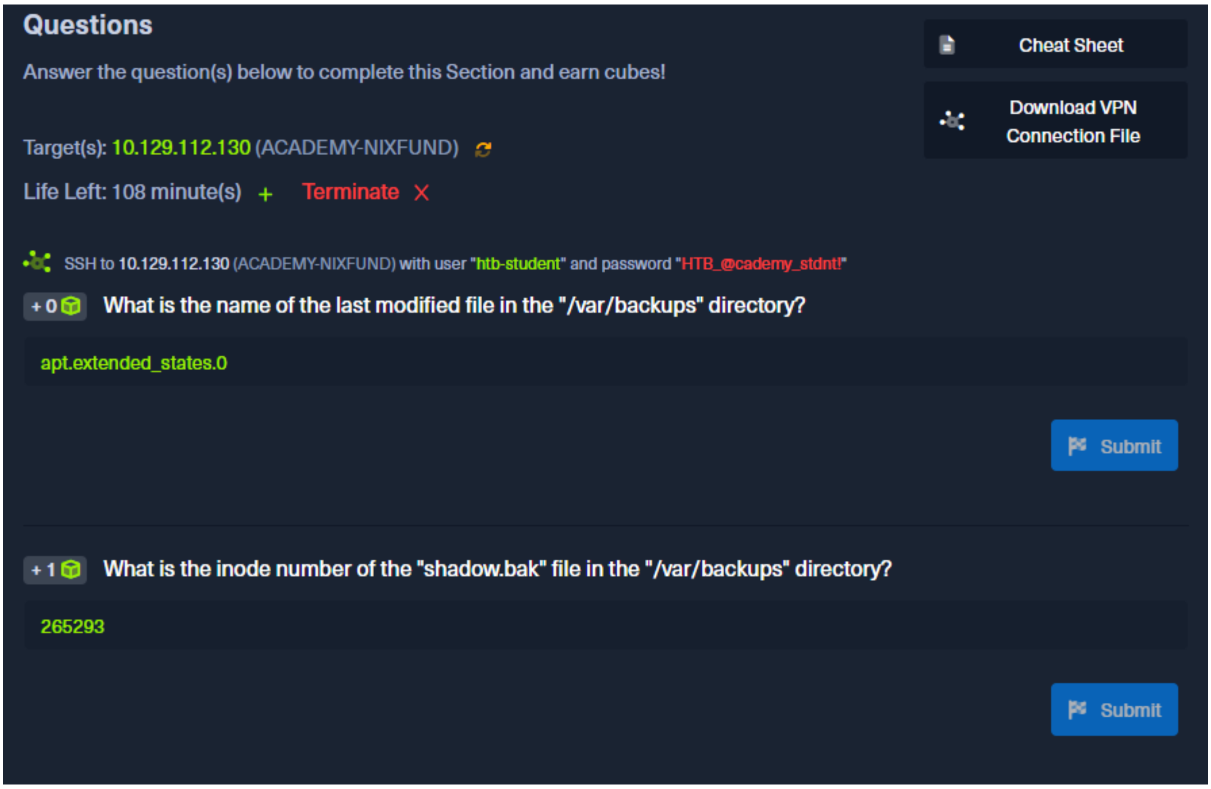
1.8 Working with files and directories:

I got to look at different ways to work with files and directories. Creating files through touch command, creating directories via mkdir command, moving files via mv command and finally coping files via cp command.

I answered the following questions. Here's a screenshot of the questions and answers with the demonstration of how I got the answers.



```
  System load:  0.79               Processes:             154
  Usage of /:   62.5% of 6.76GB    Users logged in:       0
  Memory usage: 25%                IP address for ens192: 10.129.182.254
  Swap usage:   0%

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.


Last login: Wed Sep 23 22:09:41 2020 from 10.10.14.6
htb-student@nixfund:~$ tail /var/backups
tail: error reading '/var/backups': Is a directory
htb-student@nixfund:~$ cd  /var/backups
htb-student@nixfund:/var/backups$ ls
alternatives.tar.0        apt.extended_states.2.gz  dpkg.diversions.2.gz  dpkg.statoverride.0     dpkg.statoverride.5.gz  dpkg.status.3.gz  gshadow.bak
alternatives.tar.1.gz     apt.extended_states.3.gz  dpkg.diversions.3.gz  dpkg.statoverride.1.gz  dpkg.statoverride.6.gz  dpkg.status.4.gz  passwd.bak
alternatives.tar.2.gz     apt.extended_states.4.gz  dpkg.diversions.4.gz  dpkg.statoverride.2.gz  dpkg.status.0           dpkg.status.5.gz  shadow.bak
apt.extended_states.0     dpkg.diversions.0         dpkg.diversions.5.gz  dpkg.statoverride.3.gz  dpkg.status.1.gz        dpkg.status.6.gz
apt.extended_states.1.gz  dpkg.diversions.1.gz      dpkg.diversions.6.gz  dpkg.statoverride.4.gz  dpkg.status.2.gz        group.bak
htb-student@nixfund:/var/backups$ cd ..
htb-student@nixfund:/var$ cd -
/var/backups
htb-student@nixfund:/var/backups$ cd ~
htb-student@nixfund:~$ ls -i /var/backups/shadow.bak
265293 /var/backups/shadow.bak
htb-student@nixfund:~$ 
```



Questions

Answer the question(s) below to complete this Section and earn cubes!

Target(s): 10.129.112.130 (ACADEMY-NIXFUND)

Life Left: 108 minute(s) +

SSH to 10.129.112.130 (ACADEMY-NIXFUND) with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 0 What is the name of the last modified file in the "/var/backups" directory?

apt.extended_states.0

+ 1 What is the inode number of the "shadow.bak" file in the "/var/backups" directory?

265293

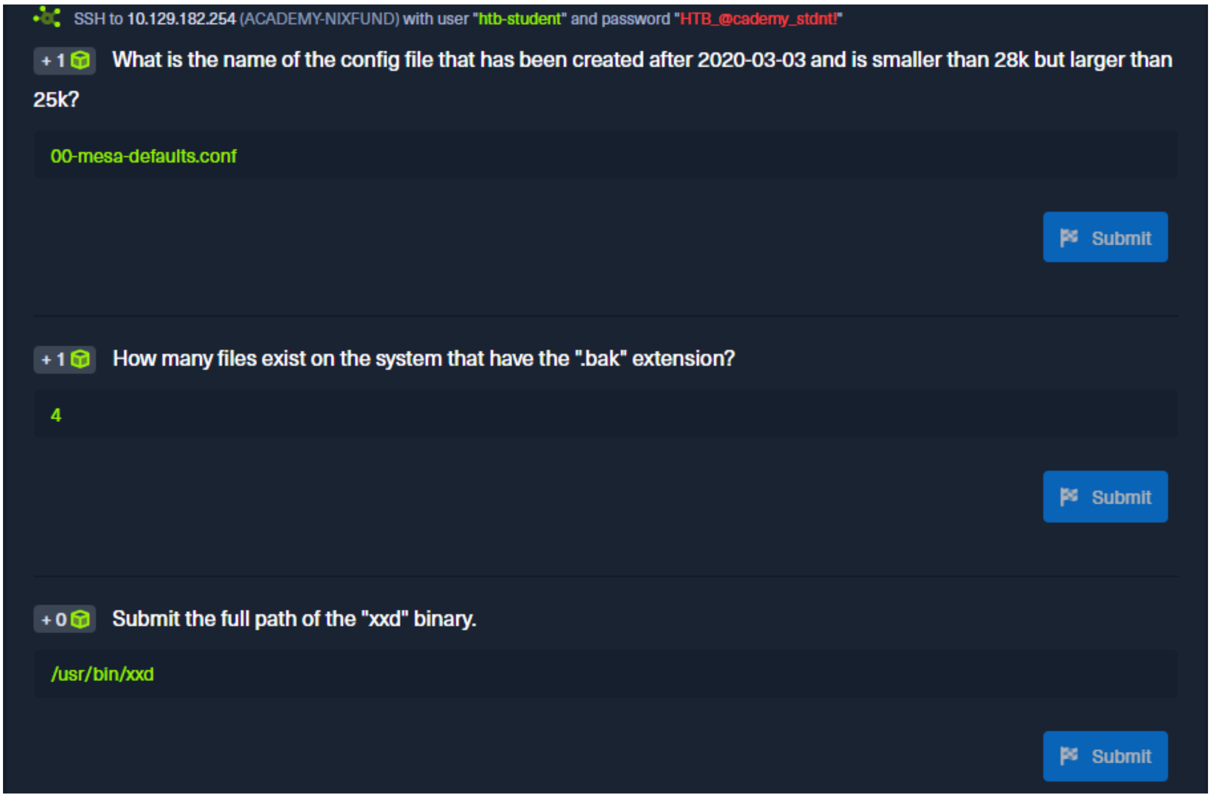
1.9 Editing Files:

Editing files has never been easy until I learned about nano and vim editors. To create and edit a file using Nano, you can specify the file name directly as the first parameter when launching the editor. For example, to create and open a new file named notes.txt.

Vim is an open-source editor for all kinds of ASCII text, just like Nano. It is an improved clone of the previous Vi. It is an extremely powerful editor that focuses on the essentials, namely editing text.

1.10 Find files and directories:

Here I looked at various tools/command used to find files and directories. Such commands are which e.g. which python, find e.g. find <location> <options>, locate e.g. locate *.conf which helped me answer the following questions.



SSH to 10.129.182.254 (ACADEMY-NIXFUND) with user "htb-student" and password "HTB_@cademy_stdnt!"

+1 What is the name of the config file that has been created after 2020-03-03 and is smaller than 28k but larger than 25k?

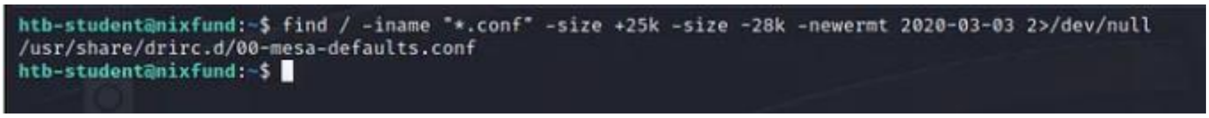
00-mesa-defaults.conf

+1 How many files exist on the system that have the ".bak" extension?

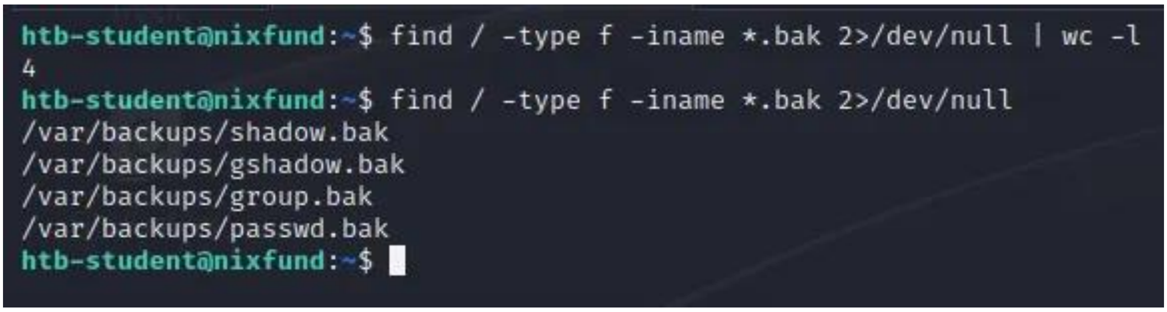
4

+0 Submit the full path of the "xxd" binary.

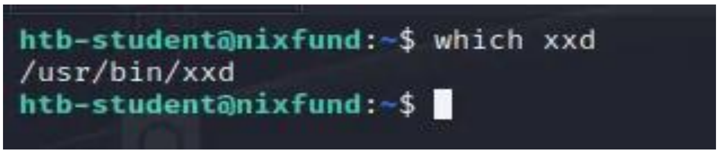
/usr/bin/xxd



```
htb-student@nixfund:~$ find / -iname "*.conf" -size +25k -size -28k -newermt 2020-03-03 2>/dev/null
/usr/share/drirc.d/00-mesa-defaults.conf
htb-student@nixfund:~$
```



```
htb-student@nixfund:~$ find / -type f -iname *.bak 2>/dev/null | wc -l
4
htb-student@nixfund:~$ find / -type f -iname *.bak 2>/dev/null
/var/backups/shadow.bak
/var/backups/gshadow.bak
/var/backups/group.bak
/var/backups/passwd.bak
htb-student@nixfund:~$
```



```
htb-student@nixfund:~$ which xxd
/usr/bin/xxd
htb-student@nixfund:~$
```

1.11 File Descriptors and Redirections:

I got to understand the 3 file descriptors in linux which are:

1. Data Stream for Input

STDIN – 0

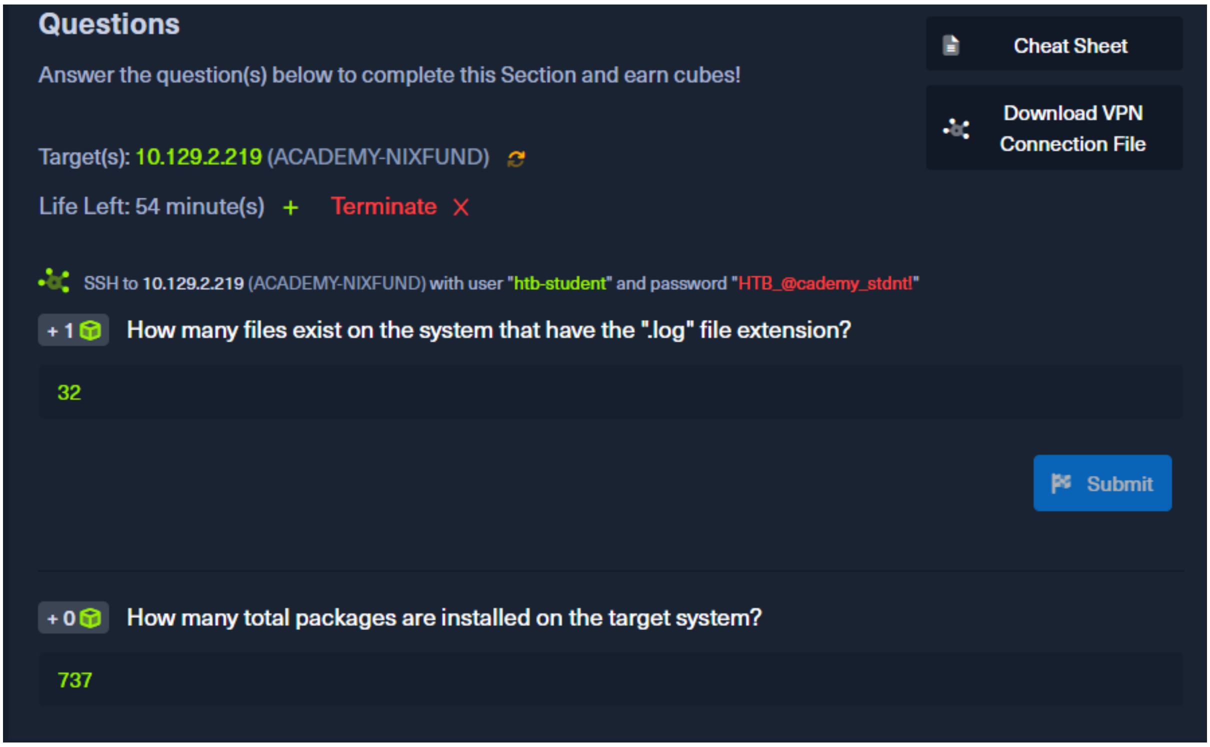
2. Data Stream for Output

STDOUT – 1

3.Data Stream for Output that relates to an error occurring.

STDERR – 2

I then got to do the following questions:



Questions

Answer the question(s) below to complete this Section and earn cubes!

Target(s): 10.129.2.219 (ACADEMY-NIXFUND)

Life Left: 54 minute(s) + Terminate ✕

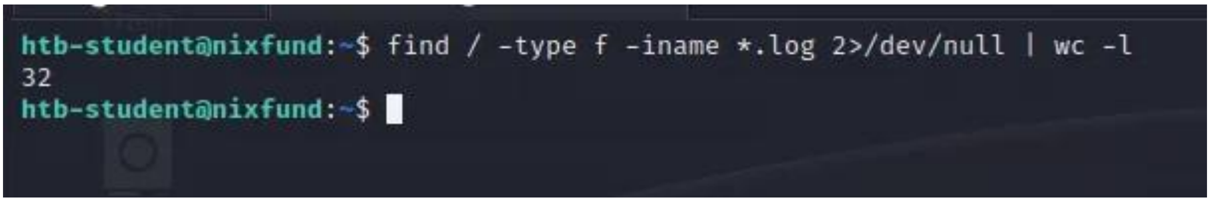
SSH to 10.129.2.219 (ACADEMY-NIXFUND) with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 1 How many files exist on the system that have the ".log" file extension?

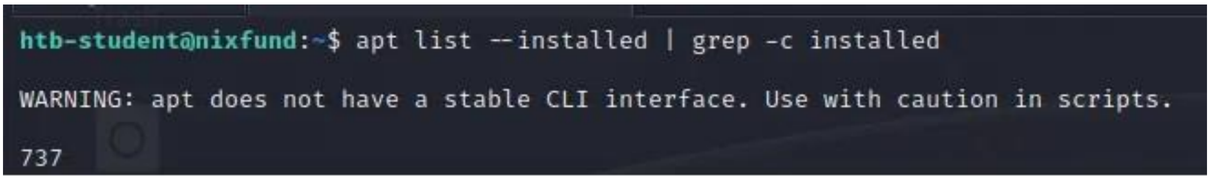
32

+ 0 How many total packages are installed on the target system?

737



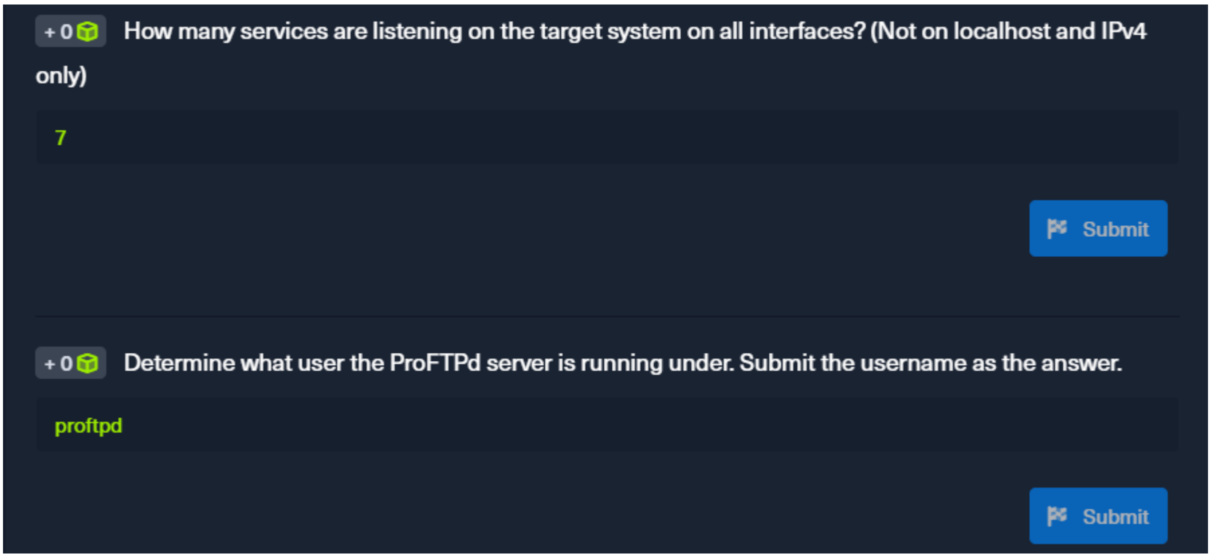
```
htb-student@nixfund:~$ find / -type f -iname *.log 2>/dev/null | wc -l
32
htb-student@nixfund:~$ █
```



```
htb-student@nixfund:~$ apt list --installed | grep -c installed

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

737
```

1.12 **Filter Contents:**

In Linux, several commands make it easier to read, filter, and manipulate text files directly from the command line. Pagers like more and less allow interactive viewing of files one screen at a time, with less offering more features and not leaving the output in the terminal when closed. To display only part of a file, head shows the first lines while tail shows the last. For organizing data, sort arranges content alphabetically or numerically, and grep is used to search for patterns or exclude results with the -v option. When working with delimited text, cut extracts specific fields, and tr replaces characters, while column formats the output into neat tables. More advanced text handling can be done with awk, which extracts specific fields, and sed, which substitutes text patterns across input. Finally, wc helps count lines, words, or characters, with -l being useful for line counts. Together, these commands provide a strong foundation for filtering and processing text efficiently.

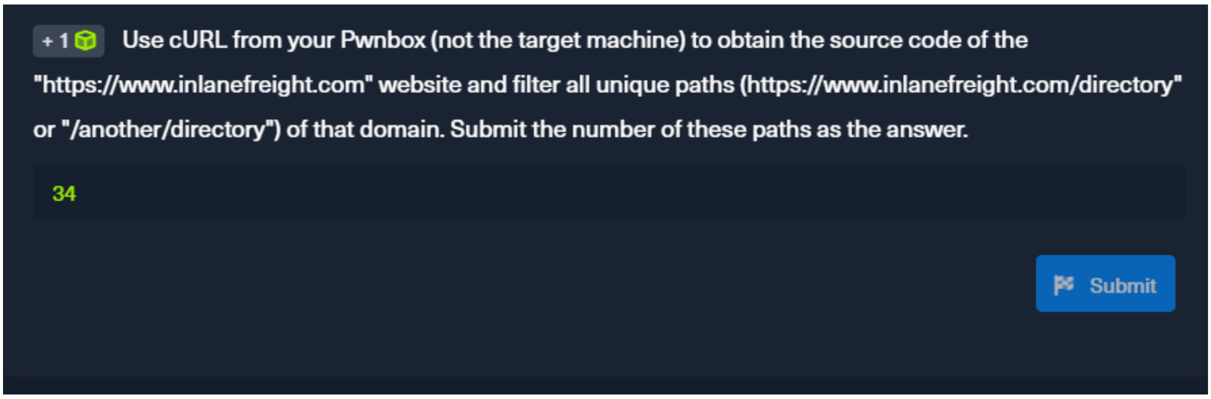


How many services are listening on the target system on all interfaces? (Not on localhost and IPv4 only)

7

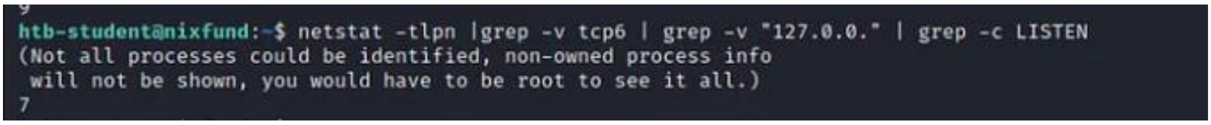
Determine what user the ProFTPd server is running under. Submit the username as the answer.

proftpd

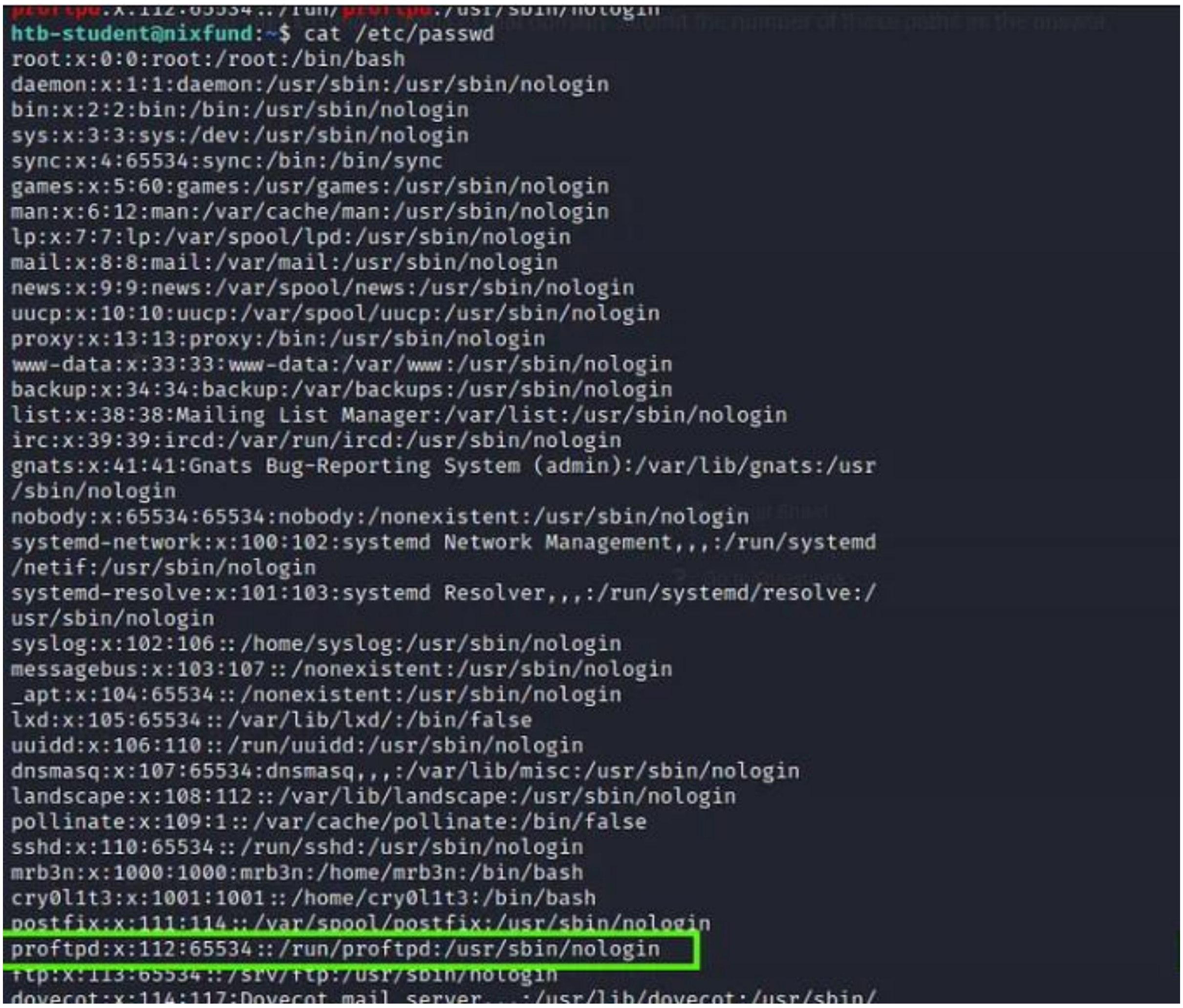


Use cURL from your Pwnbox (not the target machine) to obtain the source code of the "https://www.inlanefreight.com" website and filter all unique paths (https://www.inlanefreight.com/directory" or "/another/directory") of that domain. Submit the number of these paths as the answer.

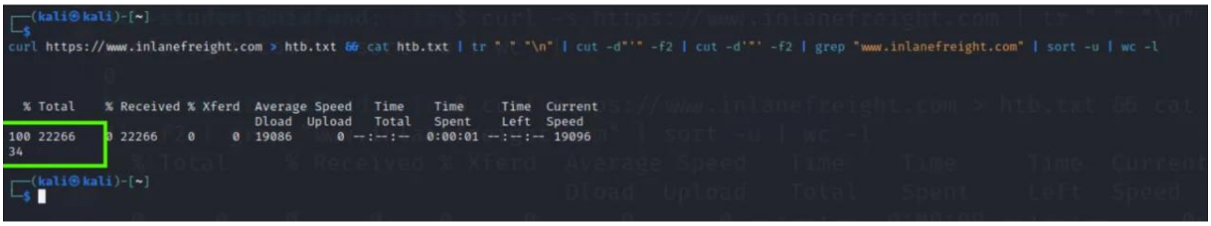
34



```
htb-student@nixfund:~$ netstat -tlpn |grep -v tcp6 | grep -v "127.0.0." | grep -c LISTEN
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
7
```



```
proftpd:x:112:65534::/run/proftpd:/usr/sbin/nologin
htb-student@nixfund:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr
/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd
/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/
usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
mrb3n:x:1000:1000:mrb3n:/home/mrb3n:/bin/bash
cry0l1t3:x:1001:1001::/home/cry0l1t3:/bin/bash
postfix:x:111:114::/var/spool/postfix:/usr/sbin/nologin
proftpd:x:112:65534::/run/proftpd:/usr/sbin/nologin
ftp:x:113:65534::/srv/ftp:/usr/sbin/nologin
dovecot:x:114:117:Dovecot mail server,,,:/usr/lib/dovecot:/usr/sbin/
```



```
┌──(kali㉿kali)-[~]
└─$
curl https://www.inlanefreight.com > htb.txt && cat htb.txt | tr " " "\n" | cut -d"'" -f2 | cut -d'"' -f2 | grep "www.inlanefreight.com" | sort -u | wc -l


  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 22266    0 22266    0     0  19086      0 --:--:--  0:00:01 --:--:-- 19096
34

┌──(kali㉿kali)-[~]
└─$ █
```

1.13 Regular Expressions:

Regular expressions (RegEx) are patterns for searching and manipulating text with tools like grep and sed. Using brackets (...), [...], {m,n} and operators like | (OR) or .* (sequence), they allow flexible matching. With grep -E or combined grep commands, you can filter files such as /etc/ssh/sshd_config to find non-comment lines, words starting with *Permit*, ending with *Authentication*, containing *Key*, or beginning/ending with *Password* or *yes*.

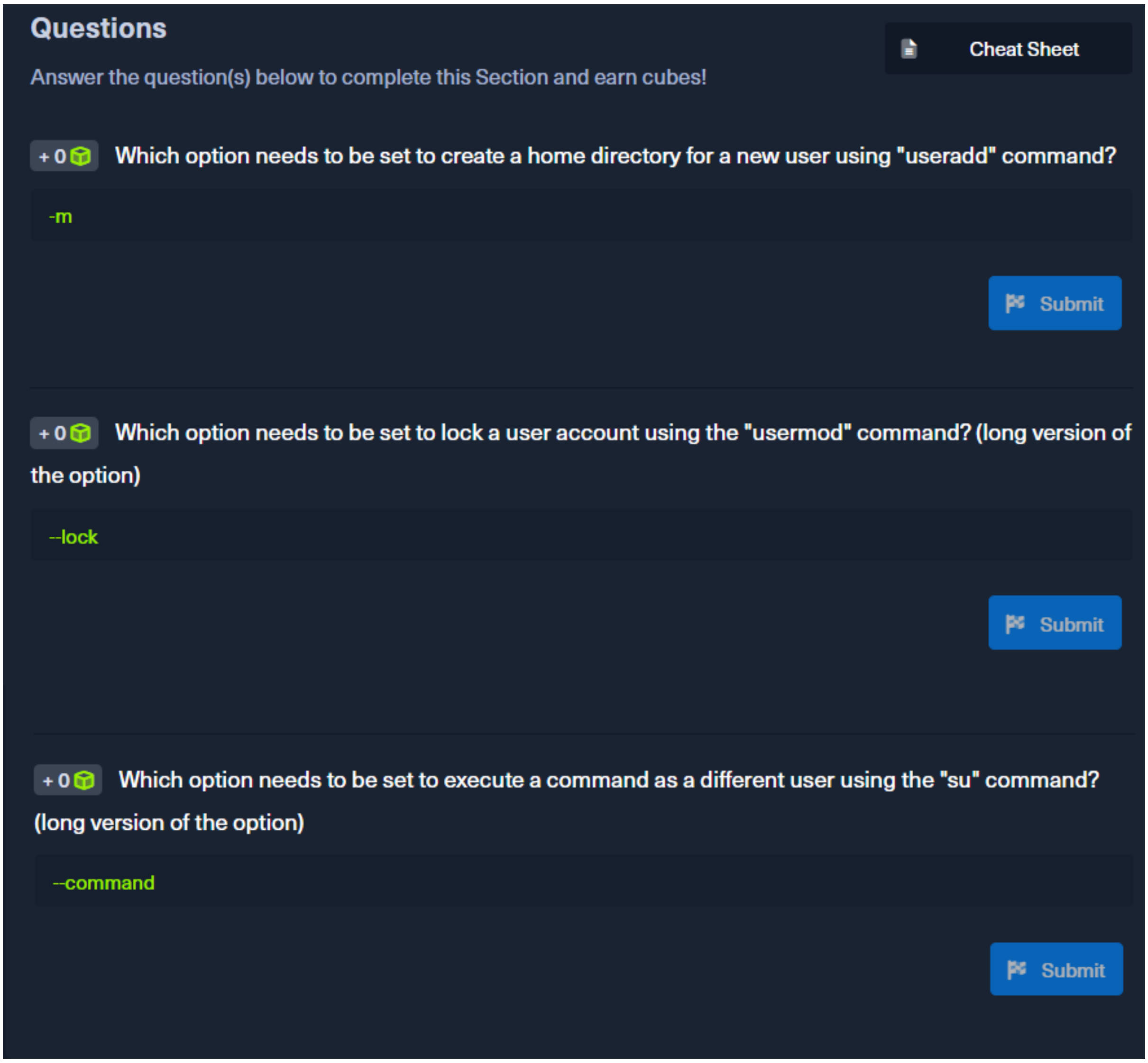
1.14 Permission Management:

Linux permissions control access to files and directories using **read (r)**, **write (w)**, and **execute (x)** rights for the owner, group, and others. They can be managed with **ls -l** to view permissions, **chmod** (symbolic or octal) to change them, and **chown** to reassign ownership. Special permissions include **SUID** and **SGID**, which allow programs to run with the file owner's or group's privileges, and the **sticky bit**, which restricts file deletion in shared directories. Together, these tools ensure security and collaboration in multi-user environments.

1.15 User Management:

User/group management controls account creation, group membership, and privilege use to keep systems secure (use **sudo**, **su**); manage accounts with **useradd**, **userdel**, **usermod**, **addgroup**, **delgroup**, and **passwd**.

I got to answer the following questions whose answers were directly from the notes given.



Questions

Answer the question(s) below to complete this Section and earn cubes!

+0 Which option needs to be set to create a home directory for a new user using "useradd" command?

-m

+0 Which option needs to be set to lock a user account using the "usermod" command? (long version of the option)

--lock

+0 Which option needs to be set to execute a command as a different user using the "su" command? (long version of the option)

--command

1.16 Package Management:

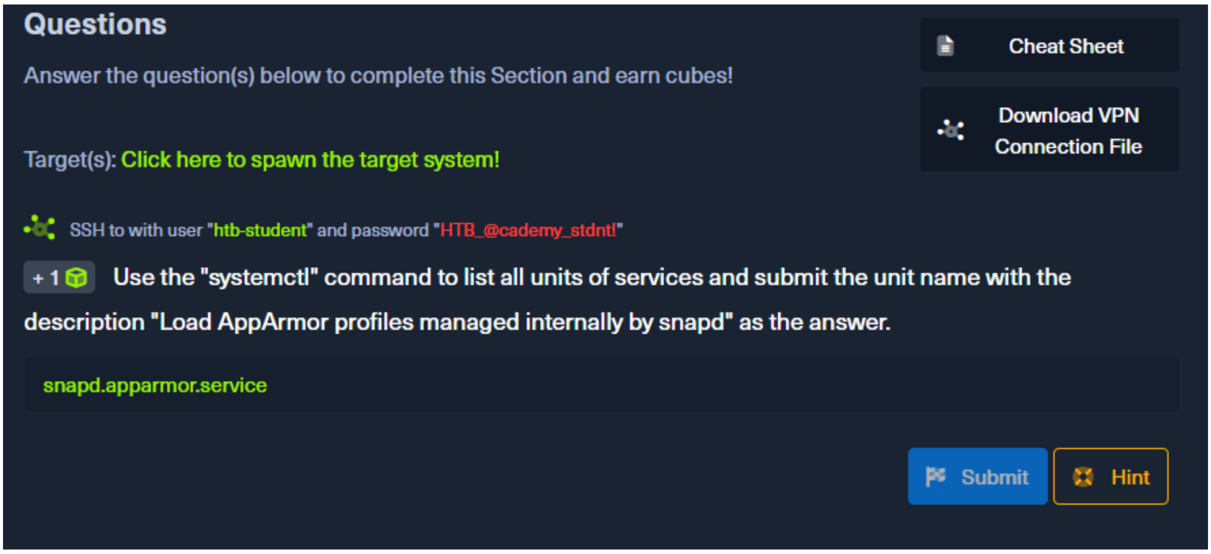
Linux uses package managers to download, resolve dependencies, and install/remove software (common formats: **.deb**, **.rpm**). Use low-level and high-level tools like **dpkg** (manage .deb files), **apt/aptitude** (higher-level APT front ends), and **snap** (snap packages).

Language/ecosystem installers include **gem** (Ruby) and **pip** (Python); **git** fetches source from repositories. Use **apt-cache** and **apt list --installed** to search/list packages, and

sudo apt install <pkg> to install from repos.
You can clone projects with **mkdir && git clone <repo>**, download .deb files with **wget**, then install those files with **sudo dpkg -i <file>.deb**.

1.17 Service and Process Management:

Services run in the background and are managed with **systemctl** (e.g., systemctl start ssh, systemctl status ssh, systemctl enable ssh, systemctl list-units --type=service) while processes can be inspected with **ps** and logs viewed with **journalctl -u <service> --no-pager**.
To control processes you send signals with **kill** (see all with kill -l), or use **pkill/pgrep/killall**; common signals include **SIGKILL (9)** and **SIGTERM (15)**.
Job control uses the keyboard shortcut **Ctrl+Z** to suspend, **jobs** to list, **bg** to resume in background, **fg <ID>** to bring back to foreground, or run commands directly in background with & (e.g., ping -c 10 host &).
Combine commands with ; (always run), && (run next only if previous succeeds), or pipes | (pass output to next), and use these tools together to start/stop services, troubleshoot with logs, and manage running processes.



Questions

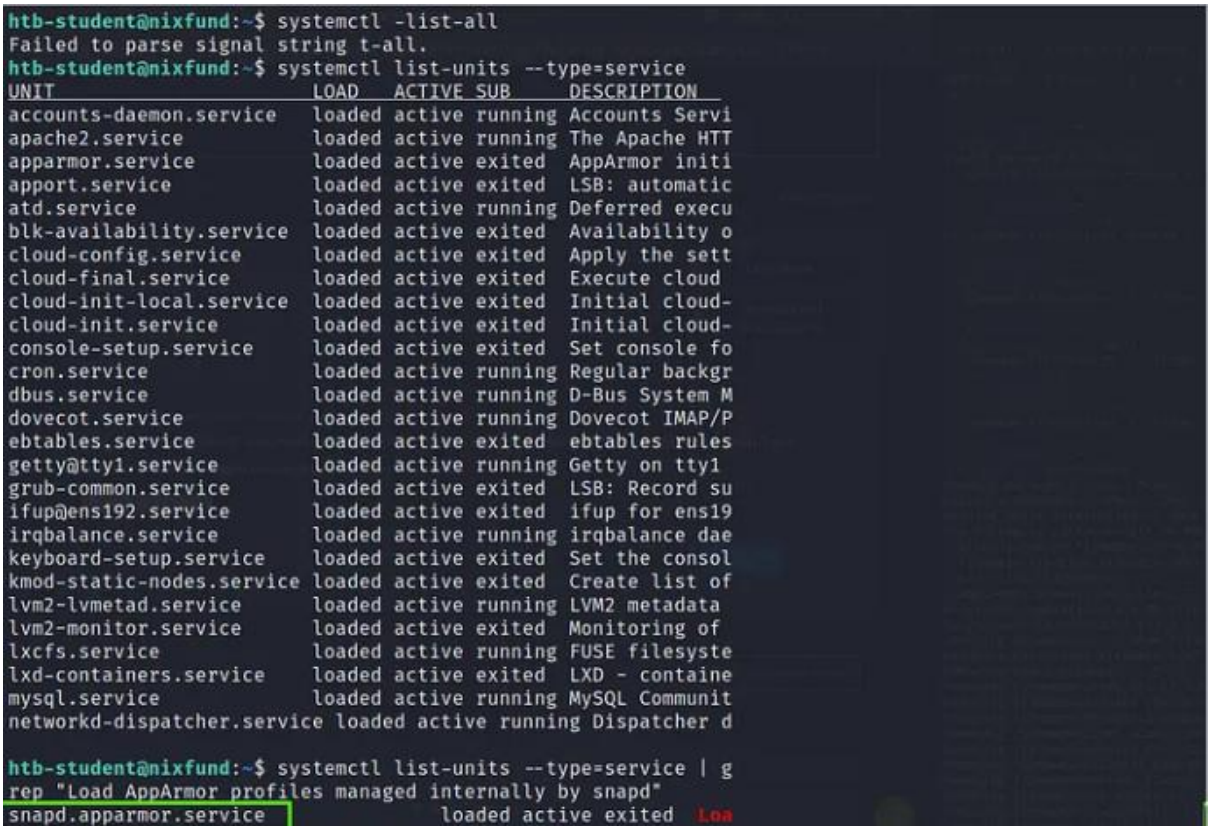
Answer the question(s) below to complete this Section and earn cubes!

Target(s): Click here to spawn the target system!

SSH to with user "htb-student" and password "HTB_@cademy_stdnt!"

+1 Use the "systemctl" command to list all units of services and submit the unit name with the description "Load AppArmor profiles managed internally by snapd" as the answer.

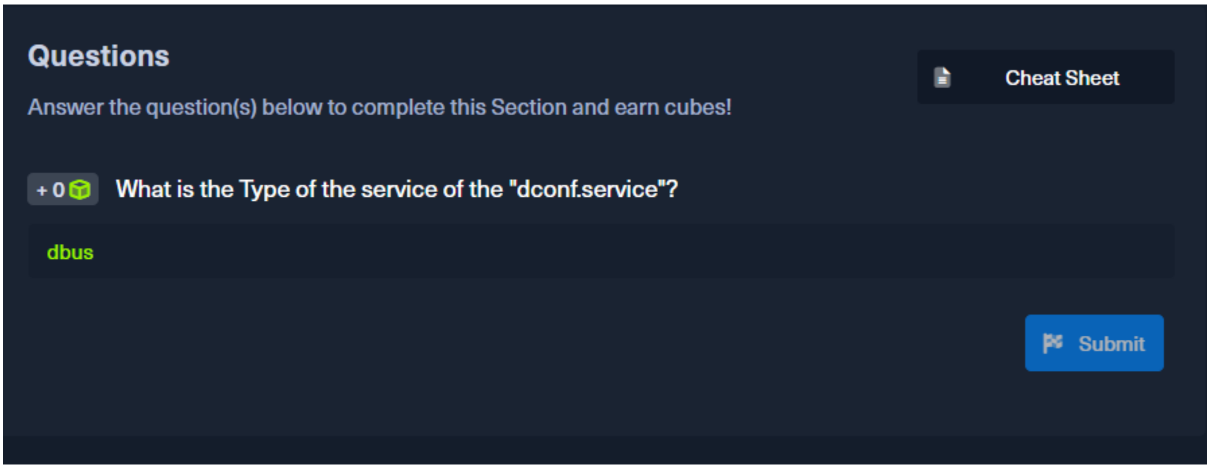
snapd.apparmor.service



```
htb-student@nixfund:~$ systemctl -list-all
Failed to parse signal string t-all.
htb-student@nixfund:~$ systemctl list-units --type=service
UNIT                         LOAD   ACTIVE SUB     DESCRIPTION
accounts-daemon.service      loaded active running Accounts Servi
apache2.service              loaded active running The Apache HTT
apparmor.service             loaded active exited  AppArmor initi
apport.service               loaded active exited  LSB: automatic
atd.service                  loaded active running Deferred execu
blk-availability.service     loaded active exited  Availability o
cloud-config.service         loaded active exited  Apply the sett
cloud-final.service          loaded active exited  Execute cloud
cloud-init-local.service     loaded active exited  Initial cloud-
cloud-init.service           loaded active exited  Initial cloud-
console-setup.service        loaded active exited  Set console fo
cron.service                 loaded active running Regular backgr
dbus.service                 loaded active running D-Bus System M
dovecot.service              loaded active running Dovecot IMAP/P
ebtables.service             loaded active exited  ebtables rules
getty@tty1.service           loaded active running Getty on tty1
grub-common.service          loaded active exited  LSB: Record su
ifup@ens192.service          loaded active exited  ifup for ens19
irqbalance.service           loaded active running irqbalance dae
keyboard-setup.service       loaded active exited  Set the consol
kmod-static-nodes.service    loaded active exited  Create list of
lvm2-lvmetad.service         loaded active running LVM2 metadata
lvm2-monitor.service         loaded active exited  Monitoring of
lxcfs.service                loaded active running FUSE filesyste
lxd-containers.service       loaded active exited  LXD - containe
mysql.service                loaded active running MySQL Communit
networkd-dispatcher.service loaded active running Dispatcher d

htb-student@nixfund:~$ systemctl list-units --type=service | g
rep "Load AppArmor profiles managed internally by snapd"
snapd.apparmor.service                loaded active exited  Loa
```

1.18 **Task Scheduling:**

Task scheduling in Linux automates recurring tasks using **systemd** or **cron**.
With **systemd**, create timer/service files, reload with `sudo systemctl daemon-reload`, then start and enable using `sudo systemctl start mytimer.timer` and `sudo systemctl enable mytimer.timer`.
With **cron**, add crontab entries to run tasks on a schedule.
Systemd uses unit files and events, while cron relies on time fields.



Questions

Answer the question(s) below to complete this Section and earn cubes!

+0 What is the Type of the service of the "dconf.service"?

dbus

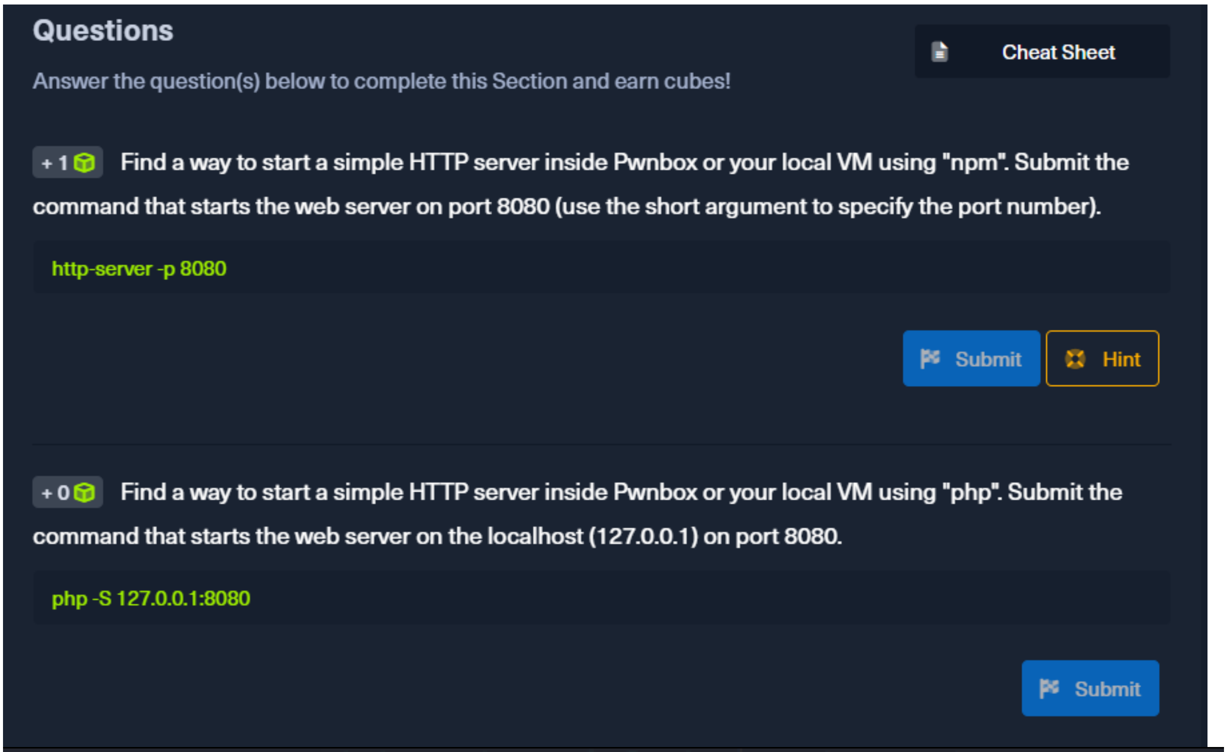
1.19 **Network Services:**

Use SSH: `sudo apt install openssh-server -y` → `systemctl status ssh` → `ssh user@host`.
NFS (share/mount): `sudo apt install nfs-kernel-server -y` → edit `/etc/exports` → `mount <host>:/remote/path ~/target_nfs`.

File hosting & VPN: `sudo apt install apache2 -y` or `python3 -m http.server`; `sudo apt install openvpn -y` → `sudo openvpn --config file.ovpn`.

1.20 Working with web services:

Apache is a widely used web server that supports static and dynamic content, extended through modules like `mod_ssl` for encryption and `mod_proxy` for traffic control. To set it up, install and start it with:
`sudo apt install apache2 -y` and `sudo systemctl start apache2`.
If port 80 is occupied, edit `/etc/apache2/ports.conf`, then restart with `sudo systemctl restart apache2`. Verify using `curl -I http://localhost:8080`.
For interaction, use `curl http://localhost` to fetch webpage source or `wget http://localhost` to download it locally. Alternatively, run a lightweight web server with `python3 -m http.server`.



Questions

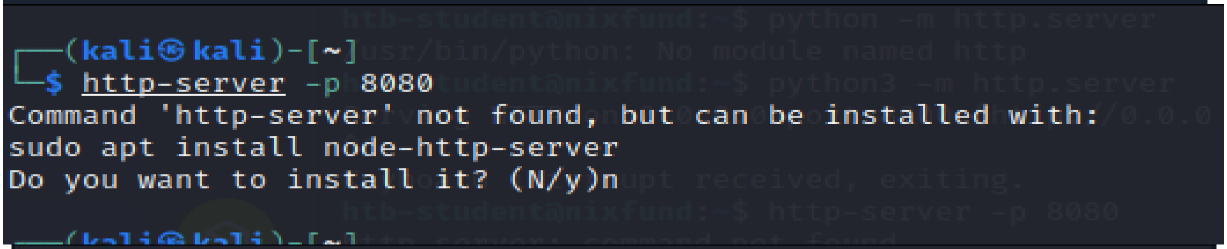
Answer the question(s) below to complete this Section and earn cubes!

+1 Find a way to start a simple HTTP server inside Pwnbox or your local VM using "npm". Submit the command that starts the web server on port 8080 (use the short argument to specify the port number).

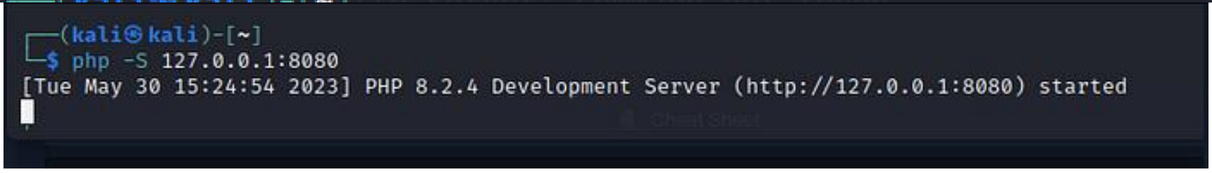
```
http-server -p 8080
```

+0 Find a way to start a simple HTTP server inside Pwnbox or your local VM using "php". Submit the command that starts the web server on the localhost (127.0.0.1) on port 8080.

```
php -S 127.0.0.1:8080
```



```
┌──(kali㉿kali)-[~]
└─$ http-server -p 8080
Command 'http-server' not found, but can be installed with:
sudo apt install node-http-server
Do you want to install it? (N/y)n
```



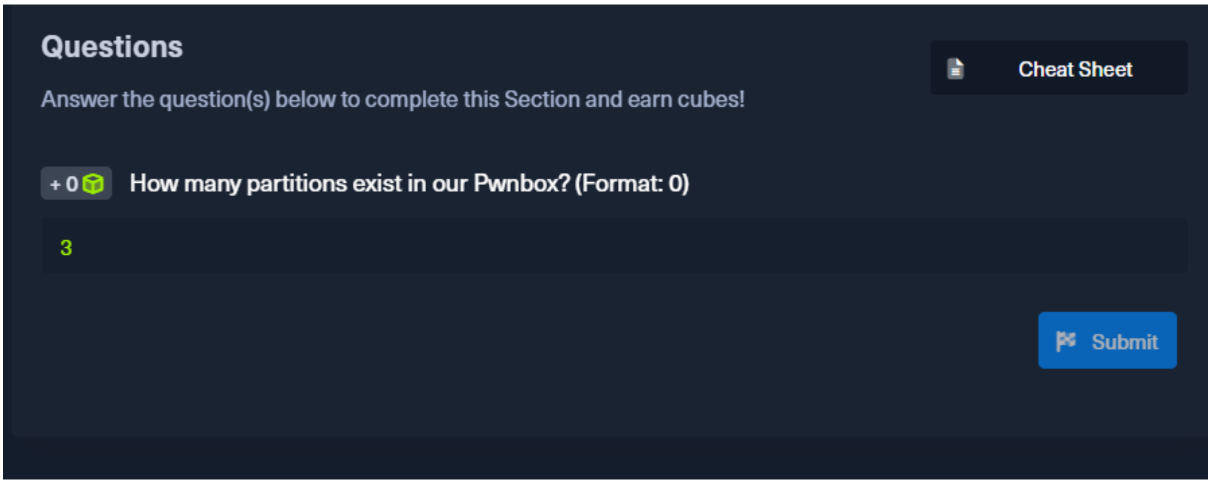
```
┌──(kali㉿kali)-[~]
└─$ php -S 127.0.0.1:8080
[Tue May 30 15:24:54 2023] PHP 8.2.4 Development Server (http://127.0.0.1:8080) started
```

1.21 Backup and Restore:

Linux provides backup tools like Rsync, Duplicity, and Deja Dup. Install Rsync with sudo apt install rsync -y.
Backup: rsync -av /path/to/mydirectory user@backup_server:/path/to/backup/directory
Incremental/Compressed: rsync -avz --backup --backup-dir=/path/to/backup/folder --delete /path/to/mydirectory user@backup_server:/path/to/backup/directory
Restore: rsync -av user@remote_host:/path/to/backup/directory /path/to/mydirectory | Secure transfer: rsync -avz -e ssh /path/to/mydirectory user@backup_server:/path/to/backup/directory | Automation: ssh-keygen -t rsa -b 2048, ssh-copy-id user@backup_server, chmod +x RSYNC_Backup.sh, crontab -e → 0 * * * * /path/to/RSYNC_Backup.sh

1.22 File System Management:

Linux supports multiple file systems like ext2, ext3, ext4, XFS, Btrfs, and NTFS, each with specific use cases. Inodes store metadata, viewable using ls -il. Disk partitions are managed with sudo fdisk -l, while mounting is done with sudo mount /dev/sdb1 /mnt/usb and checked via mount; unmounting uses sudo umount /mnt/usb, and open files can be found with lsof. Persistent mounts are defined in /etc/fstab using cat /etc/fstab. Swap space is managed by creating it with mkswap, enabling it with swapon, and tuning its options as needed.

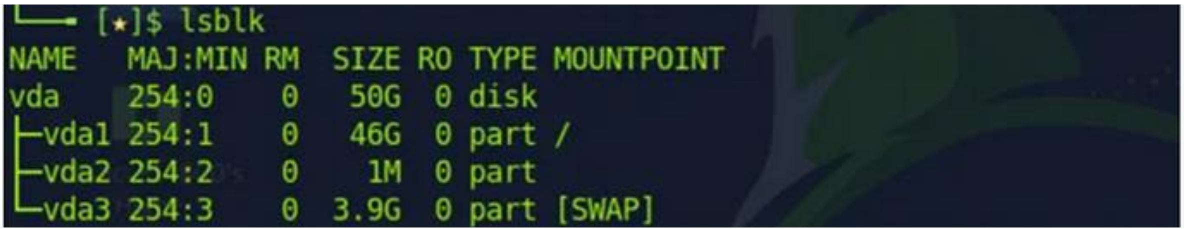


Questions

Answer the question(s) below to complete this Section and earn cubes!

+0 How many partitions exist in our Pwnbox? (Format: 0)

3



```
└──╼ [★]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    254:0    0   50G  0 disk
├─vda1 254:1    0   46G  0 part /
├─vda2 254:2    0    1M  0 part
└─vda3 254:3    0  3.9G  0 part [SWAP]
```

1.23 Containerization:

Containerization runs apps in isolated environments using tools like Docker and LXC. Docker is installed with sudo apt update, sudo apt install ..., sudo usermod -aG docker htb-student, and tested using docker run hello-world. Images are built via docker build -t FS_docker ., run with docker run -p <host>:<container> -d FS_docker, and managed using docker ps, docker stop, docker start, docker restart, docker rm, docker rmi, and docker logs. LXC is installed using sudo apt-get install lxc lxc-utils -y, containers are created with sudo lxc-create -n linuxcontainer -t ubuntu, managed with lxc-ls, lxc-start, lxc-stop, lxc-restart, lxc-config, and lxc-attach, and resource limits are set by editing sudo vim /usr/share/lxc/config/linuxcontainer.conf with cgroup options, then applying with sudo systemctl restart lxc.service.

1.24 Network Configuration:

Linux network configuration is a core skill for penetration testers: it covers managing interfaces and IPs, controlling access (DAC/MAC/RBAC), monitoring traffic, troubleshooting connectivity, and hardening hosts with controls like SELinux, AppArmor and TCP wrappers. Knowing the right commands to view/change interfaces, test reachability, trace routes, inspect sockets and logs lets you build test environments, find weaknesses, and verify fixes quickly.

1.25 Remote Desktop Protocols in Linux:

Remote desktop protocols let admins open a full graphical session on a remote machine for management and troubleshooting. On Linux you can use X11 (X server + X11 forwarding over SSH), or full desktop sharing with VNC; X11 forwards individual apps but is unencrypted unless tunnelled, while VNC serves full desktops (usually on ports 5900+). Secure practice: enable X11Forwarding in SSH and tunnel VNC over SSH to protect the session.

1.26 Linux Security:

Linux security relies on regular updates (`apt update && apt dist-upgrade`), proper firewall rules, and secure SSH settings (disabling root login and password authentication). Tools like **fail2ban** protect against brute-force attacks, while SELinux/AppArmor enforce strict access controls. Additional security comes from auditing permissions, disabling unnecessary services, enforcing strong passwords, and monitoring with tools such as **Snort, chkrootkit, rkhunter, and Lynis**.

TCP Wrappers add host-based access control using `/etc/hosts.allow` and `/etc/hosts.deny`. For example:

```
cat /etc/hosts.allow
```

1.27 Firewall Setup:

Firewalls control and monitor network traffic to prevent unauthorized access and mitigate threats. In Linux, this is achieved using the Netfilter framework with tools such as **iptables**, **nftables**, **ufw**, and **firewalld**. Among these, **iptables** remains widely used, organizing rules into **tables**, **chains**, and **rules** that define how packets are processed.

1.28 System logs and Monitoring:

System logs on Linux record kernel, system, authentication, application and security events and are essential for monitoring, troubleshooting and detecting suspicious activity during penetration tests. Key files include /var/log/syslog, /var/log/auth.log,

/var/log/kern.log and app-specific logs like /var/log/apache2/*; access and audit logs show who did what and when. Regularly configure log rotation, protect log files, and review them to spot failed logins, unexpected service activity, clear-text credentials or other anomalies that indicate compromise. Use simple command-line tools to inspect and search logs so you can quickly validate findings and tune tests.

1.29 Solaris:

Solaris, developed by Sun Microsystems (later Oracle), is a proprietary Unix OS for enterprises, unlike open-source Linux. It uses showrev -a, pkgadd -d, find / -perm -4000, share -F nfs, mount -F nfs, pfiles, and truss, while Linux uses uname -a, apt-get, find / -perm 4000, lsof, and strace. These command differences reflect Solaris' enterprise-grade design, security, and system management tools.

1.30 Shortcuts:

- **[TAB]** – Auto-complete commands, files, or directories.
- **[CTRL] + A** – Move cursor to beginning of line.
- **[CTRL] + E** – Move cursor to end of line.
- **[CTRL] + [←] / [→]** – Jump to beginning of previous/next word.
- **[ALT] + B / F** – Jump backward/forward one word.
- **[CTRL] + U** – Erase from cursor to beginning of line.
- **[CTRL] + K** – Erase from cursor to end of line.
- **[CTRL] + W** – Erase word before cursor.
- **[CTRL] + Y** – Paste erased text.
- **[CTRL] + C** – Kill/stop current process.
- **[CTRL] + D** – End-of-File (close STDIN).
- **[CTRL] + L** – Clear terminal.
- **[CTRL] + Z** – Suspend current process.
- **[CTRL] + R** – Search through command history.
- **[↑] / [↓]** – Scroll through command history.
- **[ALT] + [TAB]** – Switch between open applications.
- **[CTRL] + [+]** – Zoom in.
- **[CTRL] + [-]** – Zoom out.

CONCLUSION

Having looked at this linux module, I now have the confidence to fully work on linux and making it my primary Os. The module gave me a deep understanding of how to navigate through the terminal and also work with files as well as networking and troubleshooting my system.