Maxwell Caron
CCDC
9 August 2024

Week 4 Homework

SadServers

Find the secret combination

1. Find the number of lines with occurrences of the string Alice (case sensitive) in the .txt files in the /home/admin directory

Change into the correct directory

```
cd /home/admin
```

Use `ls` to find all of the .txt files

```
ls -al
```

```
admin@i-078cf88bb952ece8a:~$ ls -al
total 2000
drwxr-xr-x 5 admin admin   4096 Nov 20  2022 .
drwxr-xr-x 3 root  root    4096 Aug 31  2022 ..
-rw------- 1 admin admin      0 Nov 21  2022 .bash_history
-rw-r--r-- 1 admin admin    220 Aug  4  2021 .bash_logout
-rw-r--r-- 1 admin admin   3526 Aug  4  2021 .bashrc
-rw------- 1 admin admin     51 Nov 20  2022 .lesshst
drwxr-xr-x 3 admin admin   4096 Nov 20  2022 .local
-rw-r--r-- 1 admin admin    807 Aug  4  2021 .profile
drwx------ 2 admin admin   4096 Aug 31  2022 .ssh
-rw-r--r-- 1 admin admin 174313 Nov 20  2022 11-0.txt
-rw-r--r-- 1 admin admin 772181 Nov 20  2022 1342-0.txt
-rw-r--r-- 1 admin admin 607430 Nov 20  2022 1661-0.txt
-rw-r--r-- 1 admin admin 448819 Nov 20  2022 84-0.txt
drwxr-xr-x 2 admin admin   4096 Nov 21  2022 agent
-rw-r--r-- 1 admin admin      0 Nov 21  2022 solution
```

Find the number of lines where "Alice" is present

```
cat ./*.txt | grep "Alice" | wc -l
```

```
admin@i-078cf88bb952ece8a:~$ cat ./*.txt | grep "Alice" | wc -l
411
```

2. There's a file where Alice appears exactly once. In that file, in the line after that "Alice" occurrence there's a number

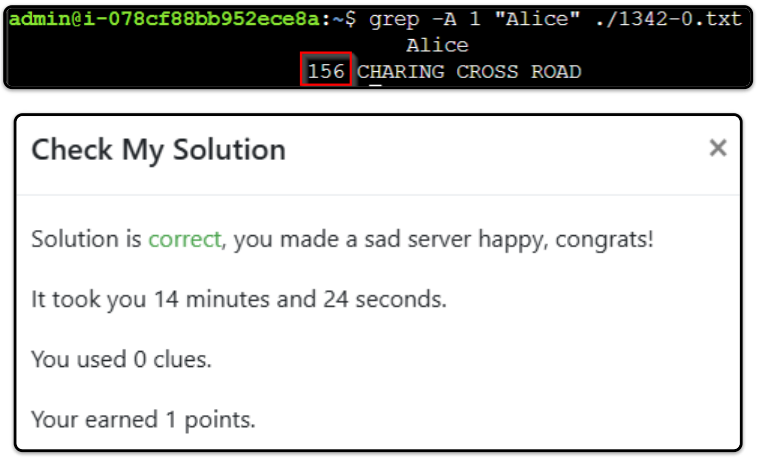
Find the file where "Alice" appears only once

```
for file in $(find . -name '*-0.txt'); do (echo $file; grep -c "Alice" $file; echo ''); done
```

```
admin@i-078cf88bb952ece8a:~$ for file in $(find . -name '*-0.txt'); do (echo $file; grep -c "Alice" $file;
echo ''); done
./84-0.txt
0

./11-0.txt
398

./1342-0.txt
1

./1661-0.txt
12
```

Find the number in the line below "Alice"

```
grep -A 1 "Alice" ./1342-0.txt
```



Break a CSV file

Create bash script

```
nano script.sh
```

```
#!/bin/bash
header=$(sed -n 1p data.csv)
count=0

split -n 10 -d data.csv data- --additional-suffix=.csv

for f in data-*; do
    sed -i "1i\\$header" $f
    count=$((count + 1))
done
```

Make script executable

```
chmod +x script.sh
```

Run script

```
./script.sh
```

List files - verifying the creation and size

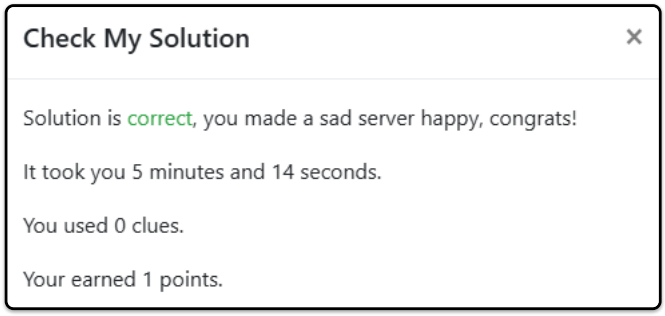
```
ls -al
```

```
admin@i-0e719dfa39cc20295:~$ ls -al
total 672
drwxr-xr-x 6 admin admin   4096 Aug  3 22:31 .
drwxr-xr-x 3 root  root    4096 Feb 17 22:46 ..
drwx------ 3 admin admin   4096 Feb 17 22:47 .ansible
-rw-r--r-- 1 admin admin    220 Mar 27  2022 .bash_logout
-rw-r--r-- 1 admin admin   3526 Mar 27  2022 .bashrc
drwxr-xr-x 3 admin admin   4096 Aug  3 22:28 .local
-rw-r--r-- 1 admin admin    807 Mar 27  2022 .profile
drwx------ 2 admin admin   4096 Feb 17 22:46 .ssh
-rw-r--r-- 1 admin admin    422 Jul 20 16:49 README.txt
drwxr-xr-x 2 admin root    4096 Jul 20 16:49 agent
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-00.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-01.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-02.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-03.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-04.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-05.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-06.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-07.csv
-rw-r--r-- 1 admin admin  31643 Aug  3 22:31 data-08.csv
-rw-r--r-- 1 admin admin  31648 Aug  3 22:31 data-09.csv
-rw-r--r-- 1 admin admin 312715 Jul 20 16:49 data.csv
-rwxr-xr-x 1 admin admin    180 Aug  3 22:31 script.sh
```

Check the first line to see if it has the same header

```
head data-09.csv
```

```
admin@i-0e719dfa39cc20295:~$ head data-09.csv
Province,Electoral District Name/Nom de circonscription,Electoral District Number/Num
éro de circonscription,Candidate/Candidat,Candidate Residence/Résidence du candidat,C
andidate Occupation/Profession du candidat,Votes Obtained/Votes obtenus,Percentage of
 Votes Obtained /Pourcentage des votes obtenus,Majority/Majorité,Majority Percentage/
Pourcentage de majorité
     5439,   9.1
"British Columbia/Colombie-Britannique","Langley--Aldergrove","59016","Leon Jensen Li
beral/Libéral","Langley, B.C./ C.-B.","Retired/Retraité",21894,36.6,,
```



Fun with Mr Jason

Check structure of `json`

```
cat station_information.json | jq | head
```

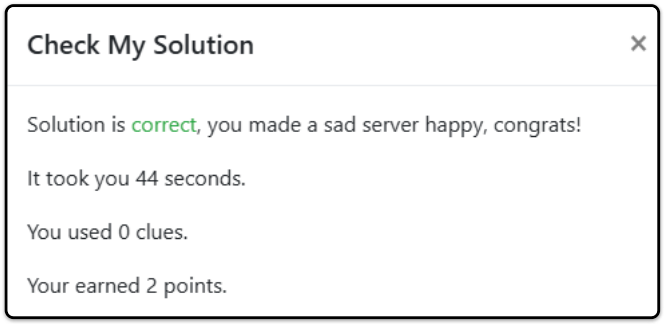
```
admin@i-0984d9aa940334536:~$ cat station_information.json | jq | head
{
  "data": {
    "stations": [
      {
        "eightd_has_key_dispenser": false,
        "rental_methods": [
          "KEY",
          "CREDITCARD"
        ],
        "external_id": "c00ef46d-fcde-48e2-afbd-0fb595fe3fa7",
```

Use `jq` to query the `json` file based on the requirements

```
cat station_information.json | jq '.data.stations[] | select(.has_kiosk == false and .capacity > 30)'
```

Add the `station_id` to the solutions file

```
echo "05e5e17c-7aa9-49b7-9da3-9db4858ec1fc" > ~/mysolution
```

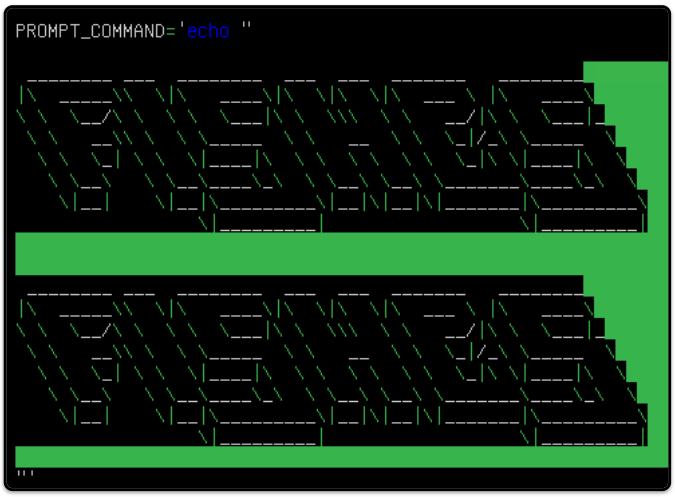


Troubleshooting Lab

1. Every time sflay types out a command, we get a funny picture of this thing or something similar.

As the `root` user, we can remediate this by deleting the following lines from `sflay`'s `/home/sflay/.bashrc` file by using a command such as `nano`.

```
nano /home/sflay/.bashrc
```



Now, we must unset the `PROMPT_COMMAND` environment variable to remove the message from our current session.

```
sflay@ubuntu22:~$ unset PROMPT_COMMAND
```

As we can see, the ASCII art is no longer visible when `sflay` executes a command.

```
sflay@ubuntu22:~$ whoami
sflay
```

2. We can't hit the Internet from this server. How are we supposed to manage our submarines without Internet? Please fix this. Oh, Jangles tried to fix this one earlier. Multiple things might now be broken.

If we try to `ping` anything on the internet, we get the following error message.

```
root@ubuntu22:/etc/netplan# ping 8.8.8.8
ping: connect: Network is unreachable
```

Looking at the status of our network interface cards, we see that `ens160` is `DOWN`

```
ip a s
```

```
root@ubuntu22:/home/fixer# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:50:56:97:47:53 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
```

Looking at the `netplan` configuration `.yaml` file, we can see that this network interface card is administratively down because `activation-mode` is set to "off"

```
root@ubuntu22:/etc/netplan# cat 00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens160:
      activation-mode: off
      dhcp4: false
      addresses:
        - 192.168.1.60/24
      nameservers:
        addresses: [8.8.8.8]
  version: 2
```

Using `nano` we can edit the `.yaml` file.

```
nano /etc/netplan/00-installer-config.yaml
```

```
network:
    ethernets:
        ens160:
            addresses:
                - 10.10.10.2/24
            nameservers:
                addresses: [8.8.8.8]
    version: 2
```

Save the configuration changes

```
sudo netplan apply
```

Using the `ip` command once again, we see that our configuration changes were successful.

```
ip a s
```

```
root@ubuntu22:/etc/netplan# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:97:47:53 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.1.60/24 brd 192.168.1.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe97:4753/64 scope link
       valid_lft forever preferred_lft forever
```

When trying to ping the internet, we still get the same error.

```
root@ubuntu22:/etc/netplan# ping 8.8.8.8
ping: connect: Network is unreachable
```

This is because we have no default gateway configured. We can once again edit our `netplan` configuration `.yaml` file.

```
nano /etc/netplan/00-installer-config.yaml
```

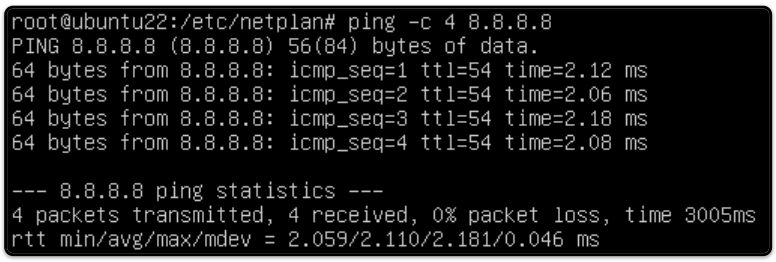
```
network:
    ethernets:
        ens160:
            addresses:
                - 10.10.10.2/24
            nameservers:
                addresses: [8.8.8.8]
            routes:
                - to: default
                  via: 192.168.1.1
    version: 2
```

Save the configuration changes

```
sudo netplan apply
```

Now we can connect to the internet.

```
ping 8.8.8.8
```

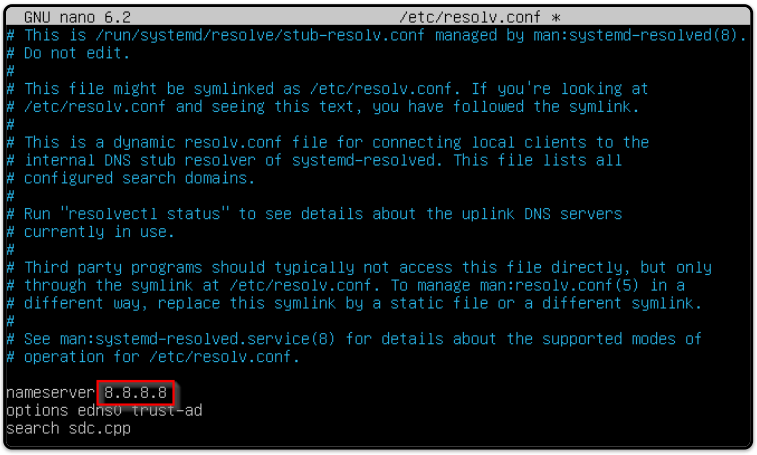


Out last test is to see if our Domain Name System (DNS) is working properly. Trying to `curl` a domain such as `google.com`, we see that we can not resolve the host.

```
curl google.com
```

Editing `/etc/resolv.conf` we can set a proper nameserver such as `8.8.8.8`

```
nano /etc/resolv.conf
```

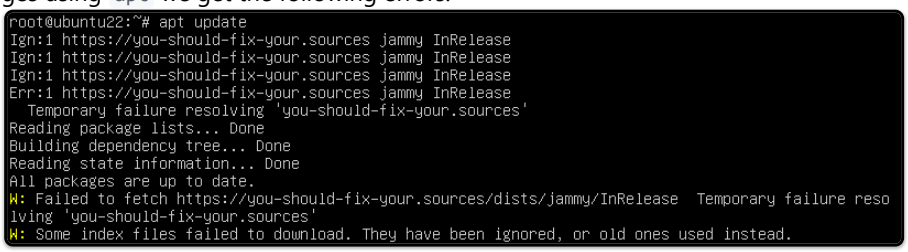


Trying to `curl` once again, we are now able to resolve the domain name

```
curl google.com
```

3. Even before this server lost Internet, we could not download or update anything. Can you fix this?

Trying to update packages using `apt` we get the following errors.



Inspecting `/etc/apt/sources.list`, which specifies the locations from which to retrieve software packages, we see that there is only one invalid entry.

Looking at the directory of the `sources.list` file, we see that there is an old version.

```
root@ubuntu22:~# ls -al /etc/apt/
total 40
drwxr-xr-x  8 root root 4096 Aug  6 00:27 .
drwxr-xr-x 98 root root 4096 Aug  1 06:26 ..
drwxr-xr-x  2 root root 4096 Jul 30 22:35 apt.conf.d
drwxr-xr-x  2 root root 4096 Apr  8  2022 auth.conf.d
drwxr-xr-x  2 root root 4096 Apr  8  2022 keyrings
drwxr-xr-x  2 root root 4096 Apr  8  2022 preferences.d
-rw-r--r--  1 root root   82 Jul 30 22:36 sources.list
-rw-r--r--  1 root root 2403 Apr 21  2022 sources.list.curtin.old
drwxr-xr-x  2 root root 4096 Jul 30 22:36 sources.list.d
drwxr-xr-x  2 root root 4096 Apr 21  2022 trusted.gpg.d
```

Reading the contents of this old file, we see that the correct entries are present.

```
  GNU nano 6.2                        /etc/apt/sources.list.curtin.old
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://archive.ubuntu.com/ubuntu/ jammy main restricted
# deb-src http://archive.ubuntu.com/ubuntu/ jammy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://archive.ubuntu.com/ubuntu/ jammy-updates main restricted
# deb-src http://archive.ubuntu.com/ubuntu/ jammy-updates main restricted
```

We can copy the old version of the `sources.list` file to the main one in order for `apt` to use it.

```
cp /etc/apt/sources.list.curtin.old /etc/apt/sources.list
```

Trying to update the packages once again, we see it succeeds

```
Fetched 35.2 MB in 7s (4,889 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
202 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

4. Those pesky interns are trying to spread misinformation about the company. They did something and now a message plays every so often. Please figure out what is causing it and delete the program.

Cronjobs are one way that a user could readily run a task, or in this case display a message. Looking at the `/etc/crontab` file, we see that a suspicious file named `thetruth` is being ran every two minutes.

```
root@ubuntu22:~# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly
)
*/2 * * * * root /usr/local/sbin/thetruth
#
```

To remediate this, we can first delete the following line from `/etc/crontab` to stop this file from running.

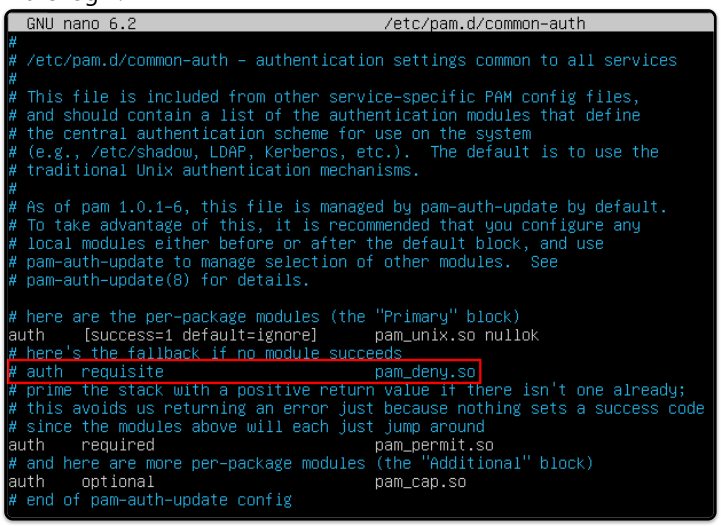
```
*/2 * * * * root /usr/local/sbin/thetruth
```

Finally, delete the file.

```
rm /usr/local/sbin/thetruth
```

5. Something is wrong with logging in. I didn't notice it at first, but Sue started questioning me on why I logged in 5 times on different servers. Turns out everyone was using my account. Nobody knows my password, so something else is wrong. I did notice that logging in was easier. Even though I always type the wrong password, I've had a 100% login success rate on all authentications.

Looking at the `/etc/pam.d/common-auth` file we can see that the fallback rule is commented out, meaning that even if an invalid password is used, it will still allow the login.



Deleting the comment (`#`) from the highlighted line will reinstate the deny rule. Now, when we try to authenticate with an invalid password, we will receive an authentication failure.

```
fixer@ubuntu22:~$ su root
Password:
su: Authentication failure
```

6. We want to set up some firewalls using iptables, but the command doesn't seem to work. Can you fix that?

When trying to run the `iptables` command, the only output we get is the string "iptables"

```
fixer@ubuntu22:~$ iptables --help
iptables
```

Looking at our `PATH` environment variable, we see that `/etc` is taking precedence over all other paths. This means that the operating system will search the directories in this order from left to right looking for an executable named "iptables" since we are not specifying the complete file path.

```
fixer@ubuntu22:~$ echo $PATH
/etc:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/
bin
```

Looking in the `/etc` directory, we see that there is an executable named "iptables." We can assume that this is the file being executed rather than the legitimate command.

```
fixer@ubuntu22:~$ ls -al /etc | grep iptables
-rwxr-xr-x  1 root root         26 Jul 30 21:50 iptables
```

Reading the contents of this file, we see that this executable echoes the string "iptables" when it is ran.

```
fixer@ubuntu22:~$ cat /etc/iptables
#!/bin/bash
echo iptables
```

There are two main ways of fixing this issue. First, the `/etc` directory is very rarely set in the `PATH` environment variable by default. If the `/etc` directory is not required for any other operations, we can remove it from the environment variable using the following `export` command.

```
fixer@ubuntu22:~$ export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/game
s:/usr/local/games:/snap/bin
```

Since changing the `PATH` environment variable only affects that specific user, for a more system-wide change, we can delete the `/etc/iptables` executable.

```
rm /etc/iptables
```

Applying either, or both, of these fixes result in the `iptables` command successfully running as intended.

```
fixer@ubuntu22:~$ iptables
iptables v1.8.7 (nf_tables): no command specified
Try `iptables -h' or 'iptables --help' for more information.
```

7. We are afraid that the interns may have left some users on the machines. Can you do a user audit? The only users should be sflay, jjangles, jchen, and wcarter. Also, make sure only wcarter and sflay (along with yourself) have admin permissions.

Looking through `/etc/passwd` we can see the users who have a default shell of `/bin/bash`. One user stands out, `toor`.

```
root@ubuntu22:~# cat /etc/passwd | grep "/bin/bash"
root:x:0:0:root:/root:/bin/bash
toor:x:0:0:root:/root:/bin/bash
sflay:x:1001:1001:,,,:/home/sflay:/bin/bash
fixer:x:1002:1002:,,,:/home/fixer:/bin/bash
jjangles:x:1003:1003:,,,:/home/jjangles:/bin/bash
wcarter:x:1004:1004:,,,:/home/wcarter:/bin/bash
jchen:x:1005:1005:,,,:/home/jchen:/bin/bash
```

We can delete this user by using the following command.

```
userdel -rf toor
```

When we try to login as the `toor` user now, we see that this user no longer exists.

```
root@ubuntu22:~# su toor
su: user toor does not exist or the user entry does not contain all the required fields
```

All of the users have `sudo` or admin permissions, to change this, we can use the `gpasswd` command on `jjangles` and `jchen`.

```
root@ubuntu22:~# gpasswd -d jjangles sudo
Removing user jjangles from group sudo
root@ubuntu22:~# gpasswd -d jchen sudo
Removing user jchen from group sudo
```

This does not completely solve our issue, these users still have access to use `sudo`. Looking at the `/etc/sudoers` file, we can see that there is an entry that allows all members of the `users` group to use `sudo` on any command with no password needed.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
%users ALL=(ALL) NOPASSWD:ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

After deleting the overly permissive entry, we can verify our changes using `sudo -l`.

```
jchen@ubuntu22:~$ sudo -l
[sudo] password for jchen:
Sorry, user jchen may not run sudo on ubuntu22.
```

```
jjangles@ubuntu22:~$ sudo -l
[sudo] password for jjangles:
Sorry, user jjangles may not run sudo on ubuntu22.
```