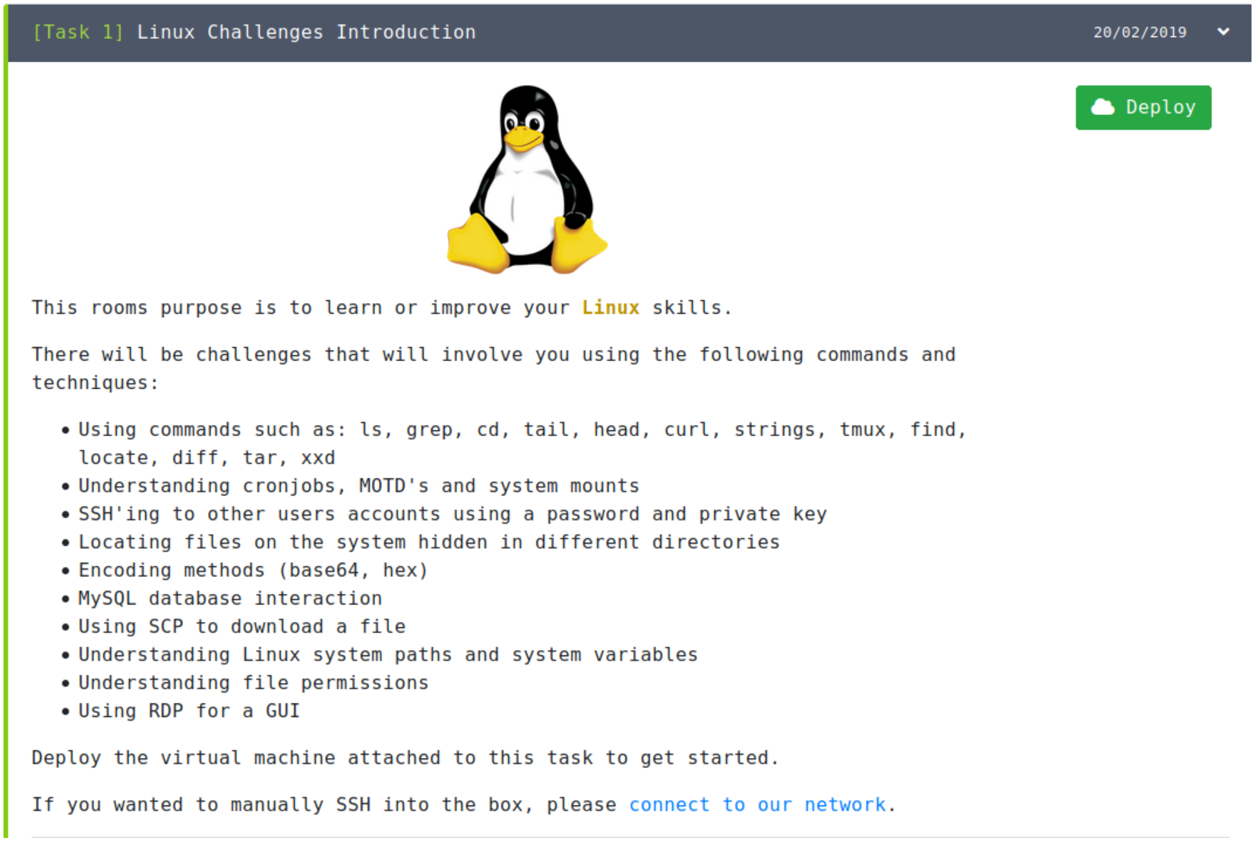
Introduction:

Hello and welcome to the tryhackme Linux CTF. This is my first CTF on tryhackme and I found it to be very enjoyable. I hope you explore other CTF on tryhackme as I will completing more and uploading them to my GitHub.

Machine Set-up:

I am running a Kali Linux machine and accessing the machine via OpenVPN so I can use the environment I am used to. Tryhackme.com can provide further details on how to connect to their network via ssh.

Task 1:



[Task 1] Linux Challenges Introduction 20/02/2019

This rooms purpose is to learn or improve your **Linux** skills.

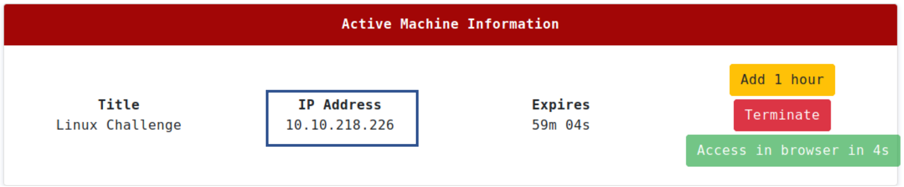
There will be challenges that will involve you using the following commands and techniques:

- Using commands such as: ls, grep, cd, tail, head, curl, strings, tmux, find, locate, diff, tar, xxd
- Understanding cronjobs, MOTD's and system mounts
- SSH'ing to other users accounts using a password and private key
- Locating files on the system hidden in different directories
- Encoding methods (base64, hex)
- MySQL database interaction
- Using SCP to download a file
- Understanding Linux system paths and system variables
- Understanding file permissions
- Using RDP for a GUI

Deploy the virtual machine attached to this task to get started.

If you wanted to manually SSH into the box, please connect to our network.

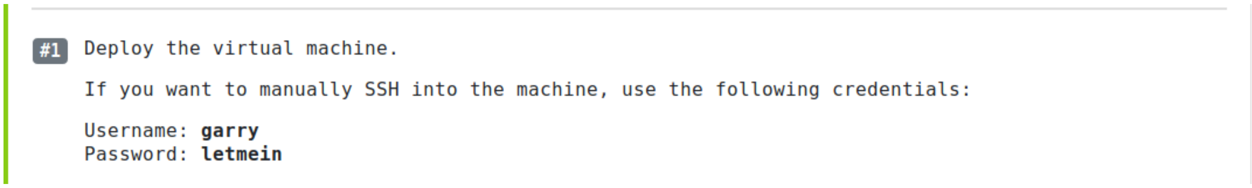
Here we find the introduction to the CTF and the challenges and techniques it will cover. In the top right corner, we see the deployed machine. Let's go ahead and deploy the machine and set up via ssh and OpenVPN.



Active Machine Information

Title	IP Address	Expires
Linux Challenge	10.10.218.226	59m 04s

After we deploy the machine we are given the IP we need to launch our ssh shown in the box.

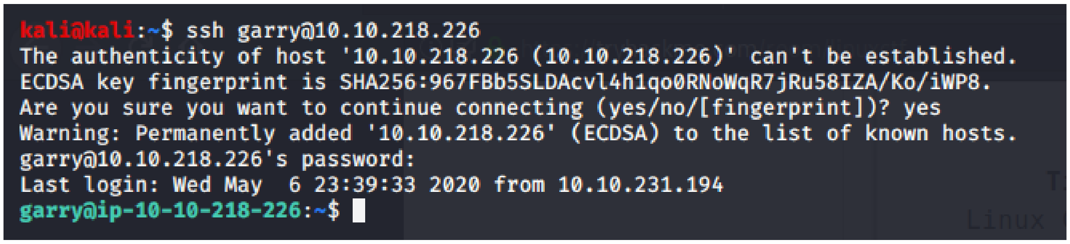


#1 Deploy the virtual machine.

If you want to manually SSH into the machine, use the following credentials:

Username: **garry**
Password: **letmein**

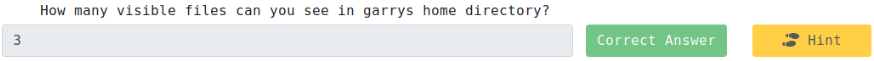
Before we connect to the network via ssh let look at the first task. It is asking us to ssh into the machine with the following credentials.



```
kali@kali:~$ ssh garry@10.10.218.226
The authenticity of host '10.10.218.226 (10.10.218.226)' can't be established.
ECDSA key fingerprint is SHA256:967FBb5SLDAcvl4h1qo0RNoWqR7jRu58IZA/Ko/iWP8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.218.226' (ECDSA) to the list of known hosts.
garry@10.10.218.226's password:
Last login: Wed May  6 23:39:33 2020 from 10.10.231.194
garry@ip-10-10-218-226:~$ 
```

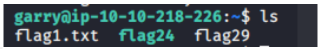
We have now logged into the machine with the login credentials that were given and can see that we are now the user garry.

In this task, we need to list all the files in garrys home directory. This can be done using the ls command.



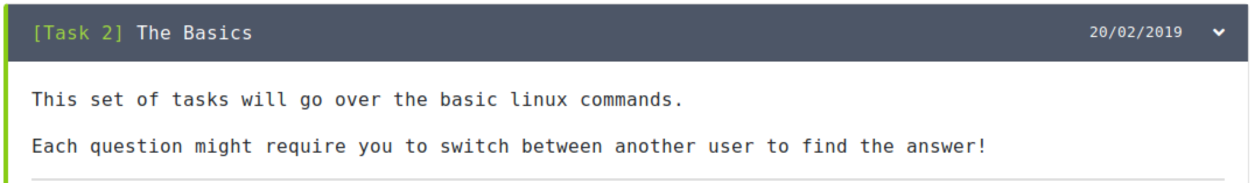
How many visible files can you see in garrys home directory?

3



```
garry@ip-10-10-218-226:~$ ls
flag1.txt  flag24  flag29
```

Task 2:

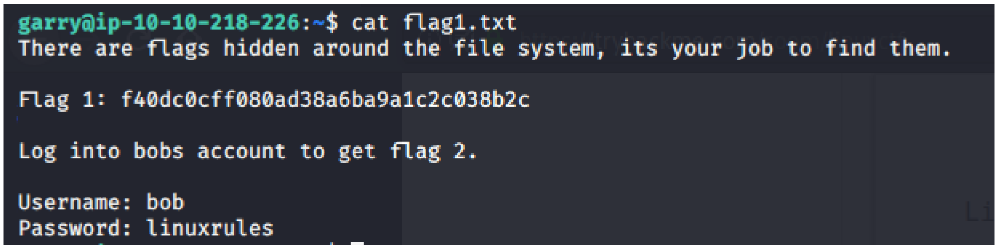


[Task 2] The Basics 20/02/2019

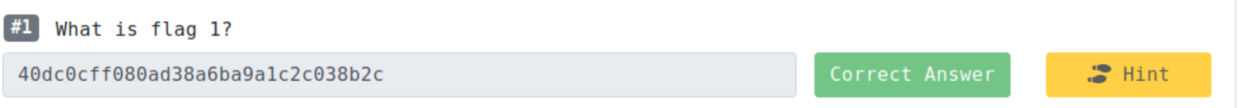
This set of tasks will go over the basic linux commands.

Each question might require you to switch between another user to find the answer!

In this task, we cover basic Linux commands. I have some experience working in Linux from my past semester so this was not too hard to complete but cool to see how we get to each flag. I also completed this section before starting the write-up.



```
garry@ip-10-10-218-226:~$ cat flag1.txt
There are flags hidden around the file system, its your job to find them.

Flag 1: f40dc0cff080ad38a6ba9a1c2c038b2c

Log into bobs account to get flag 2.

Username: bob
Password: linuxrules
```

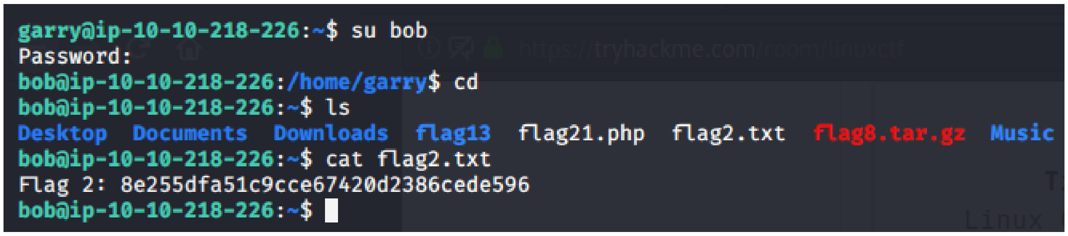


#1 What is flag 1?

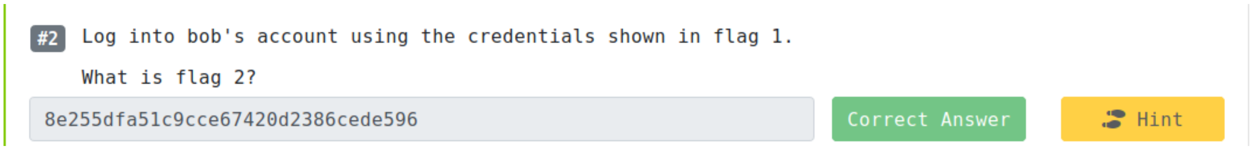
40dc0cff080ad38a6ba9a1c2c038b2c

This once was easy but that’s what we excepted considering it is the first flag we need to find. I cat the flag1.txt file and we can see that the flag came up and instructions for the next flag appeared.

For #2 we log into bobs account and go to his home directory. Then using ls we see that there is a flag2.txt file. We cat the flag2.txt file and are given the second flag.



```
garry@ip-10-10-218-226:~$ su bob
Password:
bob@ip-10-10-218-226:/home/garry$ cd
bob@ip-10-10-218-226:~$ ls
Desktop  Documents  Downloads  flag13  flag21.php  flag2.txt  flag8.tar.gz  Music
bob@ip-10-10-218-226:~$ cat flag2.txt
Flag 2: 8e255dfa51c9cce67420d2386cede596
bob@ip-10-10-218-226:~$ 
```

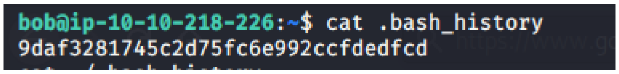


#2 Log into bob's account using the credentials shown in flag 1.

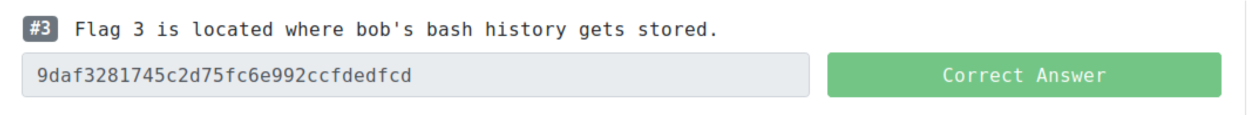
What is flag 2?

8e255dfa51c9cce67420d2386cede596

To find this flag we needed to extract info about the bash history. This can be found in the *.bash_history* file. So again I just used the cat command to extract the data. The flag was the first line in the bash history.



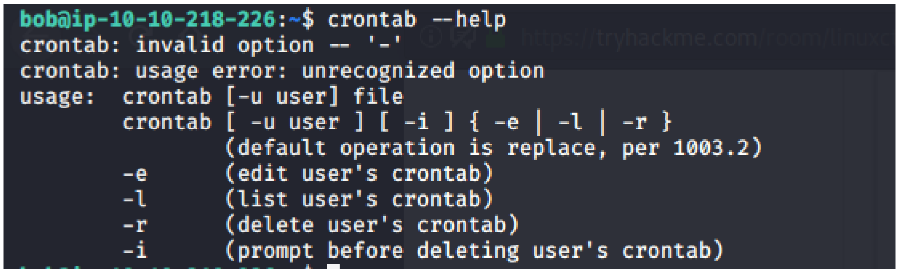
```
bob@ip-10-10-218-226:~$ cat .bash_history
9daf3281745c2d75fc6e992ccfdedfcd
```



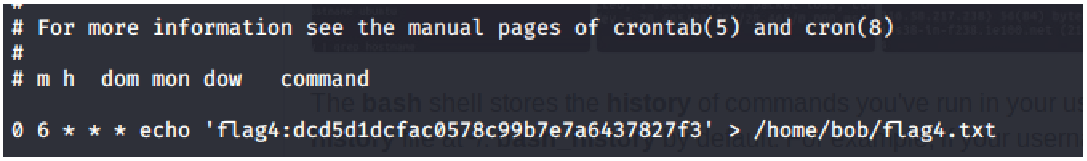
#3 Flag 3 is located where bob's bash history gets stored.

9daf3281745c2d75fc6e992ccfdedfcd

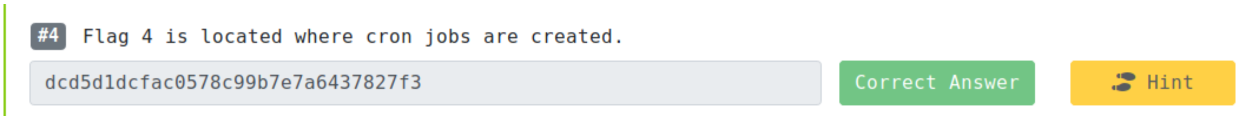
In this task, we need to find where cron jobs are created. I used the hint to give me some direction as to where to start. It prompted crontabs. Again I did not know what crontabs were so I looked at the manual on how to use them. I then used the command crontab -e to look at the user’s crontab file. Sure enough flag four was at the bottom.



```
bob@ip-10-10-218-226:~$ crontab --help
crontab: invalid option -- '-'
crontab: usage error: unrecognized option
usage:  crontab [-u user] file
        crontab [ -u user ] [ -i ] { -e | -l | -r }
                (default operation is replace, per 1003.2)
        -e      (edit user's crontab)
        -l      (list user's crontab)
        -r      (delete user's crontab)
        -i      (prompt before deleting user's crontab)
```

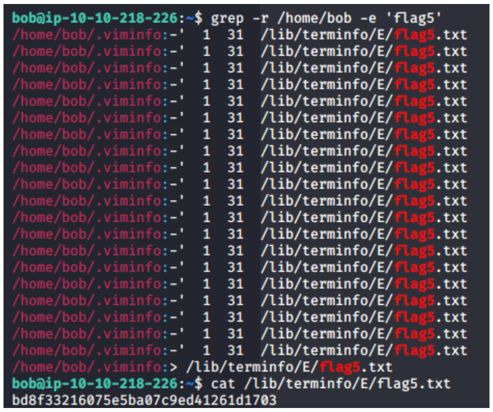


```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 6 * * * echo 'flag4:dcd5d1dcfac0578c99b7e7a6437827f3' > /home/bob/flag4.txt
```



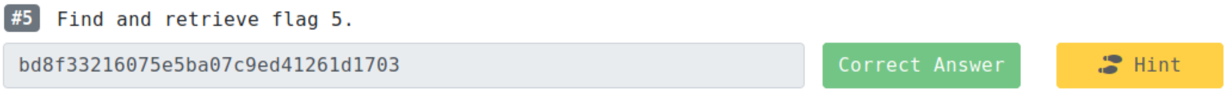
#4 Flag 4 is located where cron jobs are created.

dcd5d1dcfac0578c99b7e7a6437827f3



```
bob@ip-10-10-218-226:~$ grep -r /home/bob -e 'flag5'
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:-'  1  31  /lib/terminfo/E/flag5.txt
/home/bob/.viminfo:> /lib/terminfo/E/flag5.txt
bob@ip-10-10-218-226:~$ cat /lib/terminfo/E/flag5.txt
bd8f33216075e5ba07c9ed41261d1703
```

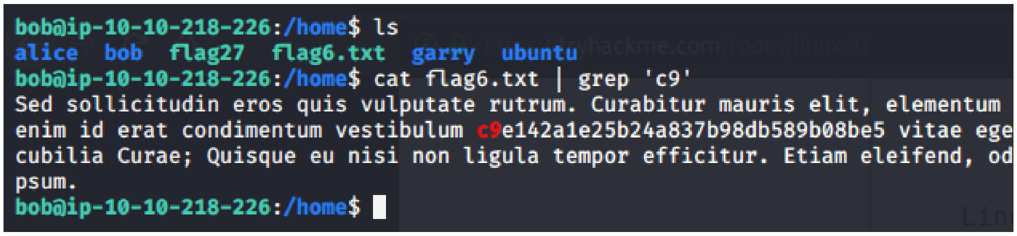
For this flag, we needed to search all the directories in /home/bob to find the directory that holds falg5.txt. I used grep -r /home/bob -e ‘flag5’ as my command specifying this needs to be recursive(-r) and we need to search for all patterns ‘flag5’ (-e). This outputted all the files that contain ‘flag5’. We can then see the file using the cat command.



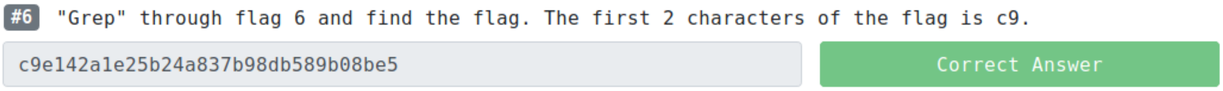
#5 Find and retrieve flag 5.

bd8f33216075e5ba07c9ed41261d1703

To find flag 6 it was much easier than the last flag we needed to find. I first went to Bob's home directory and listed all the files. Flag6.txt was already in there so I used the cat command combined with grep to get the exact location of the flag.



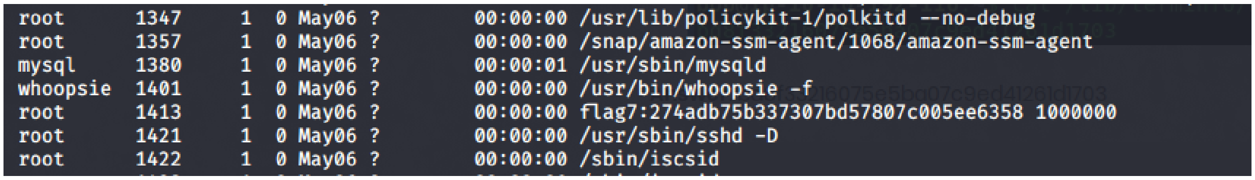
```
bob@ip-10-10-218-226:/home$ ls
alice  bob  flag27  flag6.txt  garry  ubuntu
bob@ip-10-10-218-226:/home$ cat flag6.txt | grep 'c9'
Sed sollicitudin eros quis vulputate rutrum. Curabitur mauris elit, elementum
enim id erat condimentum vestibulum c9e142a1e25b24a837b98db589b08be5 vitae ege
cubilia Curae; Quisque eu nisi non ligula tempor efficitur. Etiam eleifend, od
psum.
bob@ip-10-10-218-226:/home$ 
```



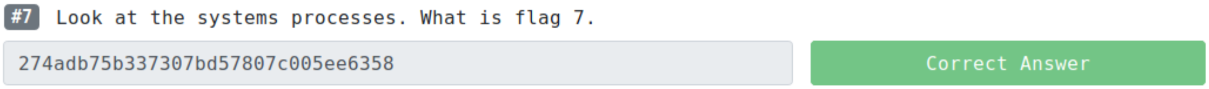
#6 "Grep" through flag 6 and find the flag. The first 2 characters of the flag is c9.

c9e142a1e25b24a837b98db589b08be5

For flag 7 we needed to look at the processes that were running on this machine. Using the ps command is the best way to do this. I first looked at the ps manual and saw that to list every running process you use the command *ps -ef.* I used this command and was able to find flag7 in the mix of all the processes.



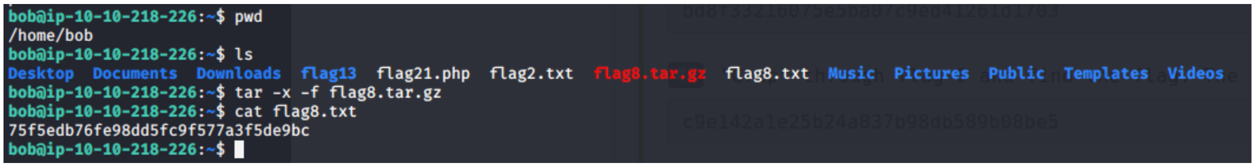
```
root       1347     1  0 May06 ?        00:00:00 /usr/lib/policykit-1/polkitd --no-debug
root       1357     1  0 May06 ?        00:00:00 /snap/amazon-ssm-agent/1068/amazon-ssm-agent
mysql      1380     1  0 May06 ?        00:00:01 /usr/sbin/mysqld
whoopsie   1401     1  0 May06 ?        00:00:00 /usr/bin/whoopsie -f
root       1413     1  0 May06 ?        00:00:00 flag7:274adb75b337307bd57807c005ee6358 1000000
root       1421     1  0 May06 ?        00:00:00 /usr/sbin/sshd -D
root       1422     1  0 May06 ?        00:00:00 /sbin/iscsid
```



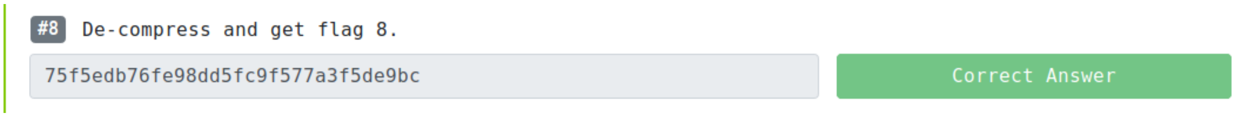
#7 Look at the systems processes. What is flag 7.

274adb75b337307bd57807c005ee6358

To get flag 8 all we needed to do was extract it from the flag8.tar.gz file. To do this I remembered this was in bobs directory so I used the command tar -x (for extraction) -f (use archive file) flag8.tar.gz. This extracted all the data and we can then use the cat command on flag8.txt.qw



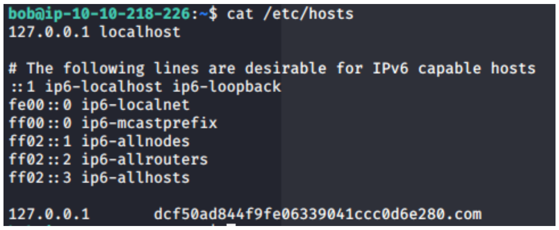
```
bob@ip-10-10-218-226:~$ pwd
/home/bob
bob@ip-10-10-218-226:~$ ls
Desktop  Documents  Downloads  flag13  flag21.php  flag2.txt  flag8.tar.gz  flag8.txt  Music  Pictures  Public  Templates  Videos
bob@ip-10-10-218-226:~$ tar -x -f flag8.tar.gz
bob@ip-10-10-218-226:~$ cat flag8.txt
75f5edb76fe98dd5fc9f577a3f5de9bc
bob@ip-10-10-218-226:~$ 
```



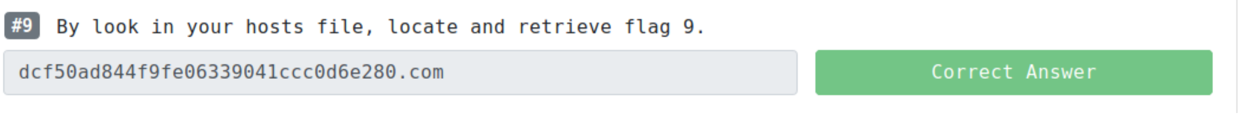
#8 De-compress and get flag 8.

75f5edb76fe98dd5fc9f577a3f5de9bc

Flag 9 was quite simple as well I just used the cat command to extract all the contents of the /etc/hosts file. Instantly I saw the flag.



```
bob@ip-10-10-218-226:~$ cat /etc/hosts
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

127.0.0.1       dcf50ad844f9fe06339041ccc0d6e280.com
```

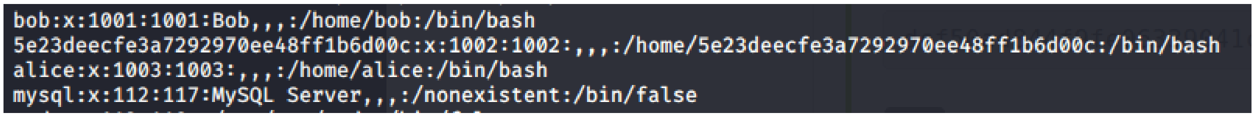


#9 By look in your hosts file, locate and retrieve flag 9.

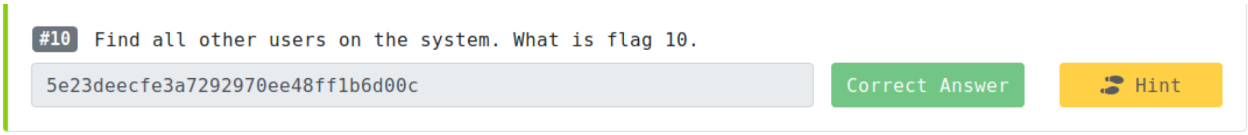
dcf50ad844f9fe06339041ccc0d6e280.com

This was fairly simple as well. I know that users are stored in the passwd file so I used the command

cat /etc/passwd. I then was given all the users and in the middle was the flag!!



```
bob:x:1001:1001:Bob,,,:/home/bob:/bin/bash
5e23deecfe3a7292970ee48ff1b6d00c:x:1002:1002:,,,:/home/5e23deecfe3a7292970ee48ff1b6d00c:/bin/bash
alice:x:1003:1003:,,,:/home/alice:/bin/bash
mysql:x:112:117:MySQL Server,,,:/nonexistent:/bin/false
```

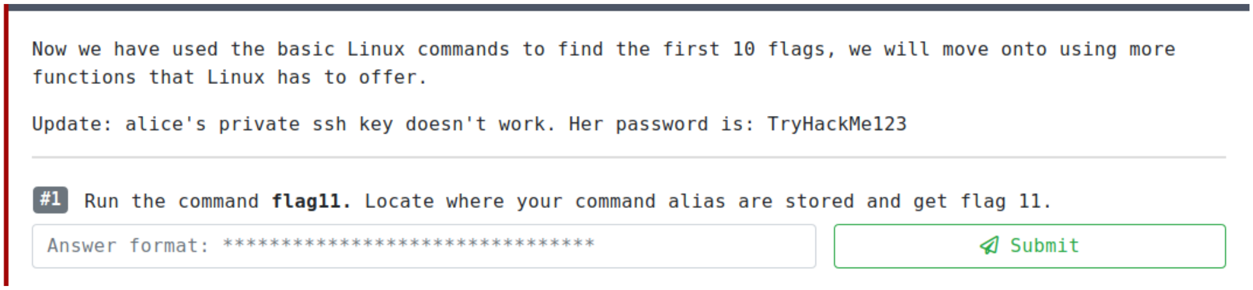


#10 Find all other users on the system. What is flag 10.

5e23deecfe3a7292970ee48ff1b6d00c

Task 3: Linux Functionality

YAY!! We completed our first real section of looking and finding flags using Linux commands! Now we are going to move into different functionalities Linux has to offer.

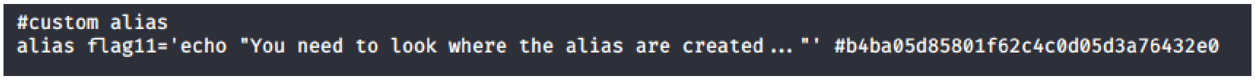


Now we have used the basic Linux commands to find the first 10 flags, we will move onto using more functions that Linux has to offer.

Update: alice's private ssh key doesn't work. Her password is: TryHackMe123

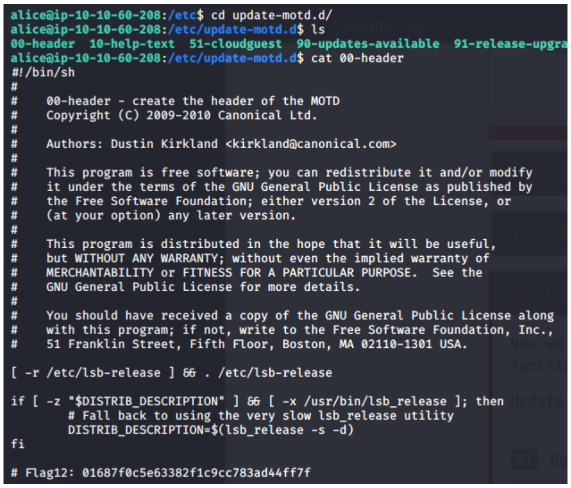
#1 Run the command **flag11**. Locate where your command alias are stored and get flag 11.

To get this flag I had to find where the command alias is stored. After some research, I found that typically they are stored in the .bashrc file for ubuntu. I then logged into Alice's account and went to the ubuntu directory. I then used the command *cat .bashrc* and we were able to find flag11.

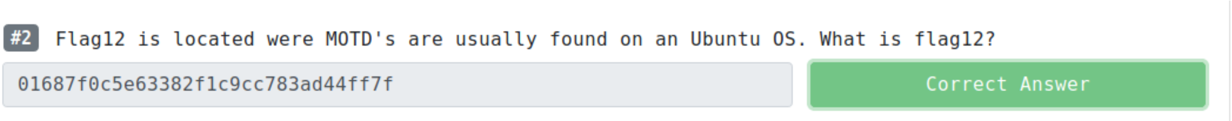


```
#custom alias
alias flag11='echo "You need to look where the alias are created..."' #b4ba05d85801f62c4c0d05d3a76432e0
```

To locate flag 12 also need some research. I found that motd is dynamic now in ubuntu OS systems so I had to find the update.motd.d folder which was inside of /etc. I then navigated to the 00-header file and used cat to see the contents. Sure enough, there was flag12.



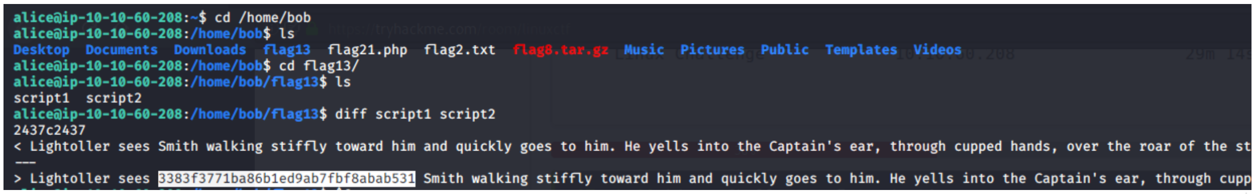
```
alice@ip-10-10-60-208:/etc$ cd update-motd.d/
alice@ip-10-10-60-208:/etc/update-motd.d$ ls
00-header  10-help-text  51-cloudguest  90-updates-available  91-release-upgra
alice@ip-10-10-60-208:/etc/update-motd.d$ cat 00-header
#!/bin/sh
#
#    00-header - create the header of the MOTD
#    Copyright (C) 2009-2010 Canonical Ltd.
#
#    Authors: Dustin Kirkland <kirkland@canonical.com>
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 2 of the License, or
#    (at your option) any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License along
#    with this program; if not, write to the Free Software Foundation, Inc.,
#    51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

[ -r /etc/lsb-release ] && . /etc/lsb-release

if [ -z "$DISTRIB_DESCRIPTION" ] && [ -x /usr/bin/lsb_release ]; then
        # Fall back to using the very slow lsb_release utility
        DISTRIB_DESCRIPTION=$(lsb_release -s -d)
fi

# Flag12: 01687f0c5e63382f1c9cc783ad44ff7f
```



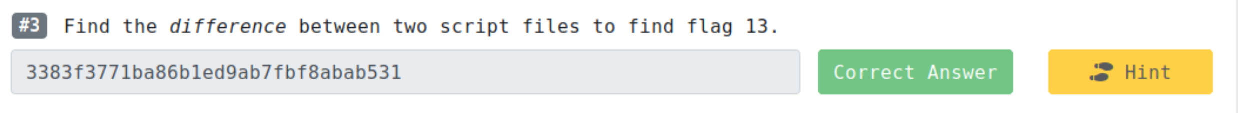
#2 Flag12 is located were MOTD's are usually found on an Ubuntu OS. What is flag12?

01687f0c5e63382f1c9cc783ad44ff7f

This flag was easy as we already know where the flag13 directory is. I then used the diff command and was able to see the difference between both scripts and get the flag.



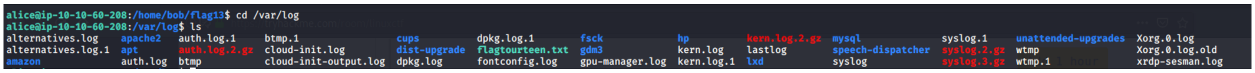
```
alice@ip-10-10-60-208:~$ cd /home/bob
alice@ip-10-10-60-208:/home/bob$ ls
Desktop  Documents  Downloads  flag13  flag21.php  flag2.txt  flag8.tar.gz  Music  Pictures  Public  Templates  Videos
alice@ip-10-10-60-208:/home/bob$ cd flag13/
alice@ip-10-10-60-208:/home/bob/flag13$ ls
script1  script2
alice@ip-10-10-60-208:/home/bob/flag13$ diff script1 script2
2437c2437
< Lightoller sees Smith walking stiffly toward him and quickly goes to him. He yells into the Captain's ear, through cupped hands, over the roar of the st
---
> Lightoller sees 3383f3771ba86b1ed9ab7fbf8abab531 Smith walking stiffly toward him and quickly goes to him. He yells into the Captain's ear, through cupp
```



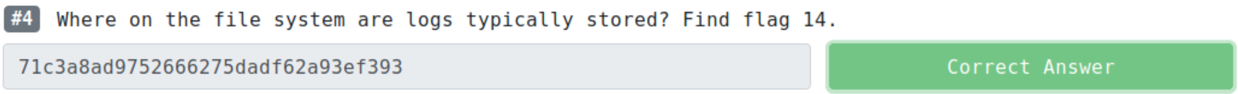
#3 Find the *difference* between two script files to find flag 13.

3383f3771ba86b1ed9ab7fbf8abab531

For flag14 all I did was go to the directory that logs are found in */var/log.* Inside there was the flag14 text file. I then used cat to look at the contents and the flag was at the very bottom.



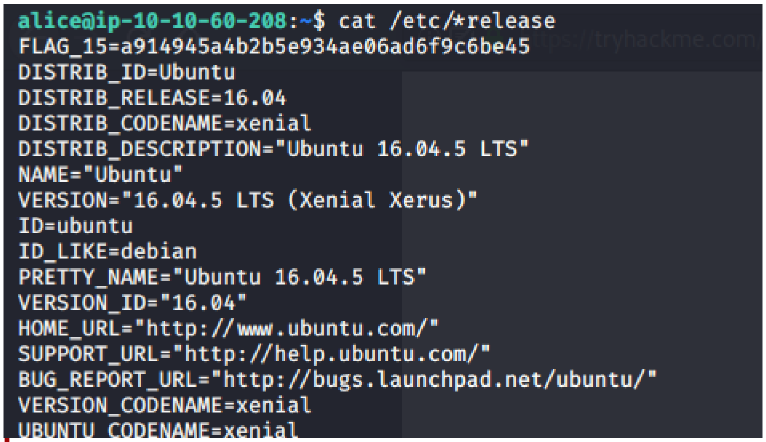
```
alice@ip-10-10-60-208:/home/bob/flag13$ cd /var/log
alice@ip-10-10-60-208:/var/log$ ls
alternatives.log    apache2  auth.log.1     btmp.1                 cups           dpkg.log.1        fsck              hp          kern.log.2.gz  mysql               syslog.1     unattended-upgrades  Xorg.0.log
alternatives.log.1  apt      auth.log.2.gz  cloud-init.log         dist-upgrade   flagtourteen.txt  gdm3              kern.log    lastlog        speech-dispatcher   syslog.2.gz  wtmp                 Xorg.0.log.old
amazon              auth.log btmp           cloud-init-output.log  dpkg.log       fontconfig.log    gpu-manager.log   kern.log.1  lxd            syslog              syslog.3.gz  wtmp.1               xrdp-sesman.log
```



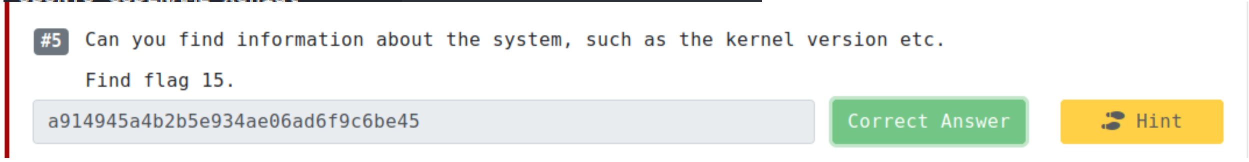
#4 Where on the file system are logs typically stored? Find flag 14.

71c3a8ad9752666275dadf62a93ef393

For flag 15 we just needed to find out the information the system. I wasn't sure where I was going wrong using the uname command but then found it.



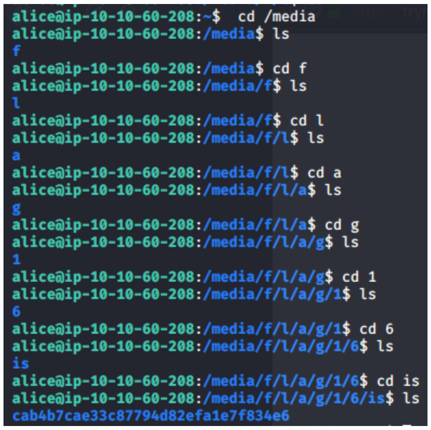
```
alice@ip-10-10-60-208:~$ cat /etc/*release
FLAG_15=a914945a4b2b5e934ae06ad6f9c6be45
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.5 LTS"
NAME="Ubuntu"
VERSION="16.04.5 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.5 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
```



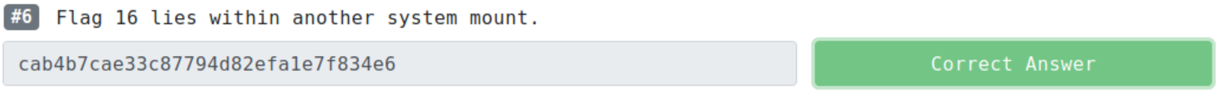
#5 Can you find information about the system, such as the kernel version etc.

Find flag 15.

a914945a4b2b5e934ae06ad6f9c6be45



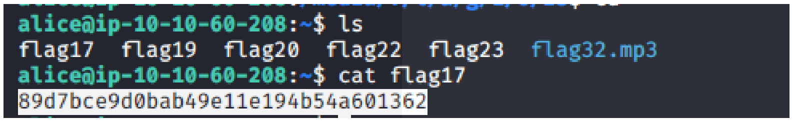
```
alice@ip-10-10-60-208:~$  cd /media
alice@ip-10-10-60-208:/media$ ls
f
alice@ip-10-10-60-208:/media$ cd f
alice@ip-10-10-60-208:/media/f$ ls
l
alice@ip-10-10-60-208:/media/f$ cd l
alice@ip-10-10-60-208:/media/f/l$ ls
a
alice@ip-10-10-60-208:/media/f/l$ cd a
alice@ip-10-10-60-208:/media/f/l/a$ ls
g
alice@ip-10-10-60-208:/media/f/l/a$ cd g
alice@ip-10-10-60-208:/media/f/l/a/g$ ls
1
alice@ip-10-10-60-208:/media/f/l/a/g$ cd 1
alice@ip-10-10-60-208:/media/f/l/a/g/1$ ls
6
alice@ip-10-10-60-208:/media/f/l/a/g/1$ cd 6
alice@ip-10-10-60-208:/media/f/l/a/g/1/6$ ls
is
alice@ip-10-10-60-208:/media/f/l/a/g/1/6$ cd is
alice@ip-10-10-60-208:/media/f/l/a/g/1/6/is$ ls
cab4b7cae33c87794d82efa1e7f834e6
```



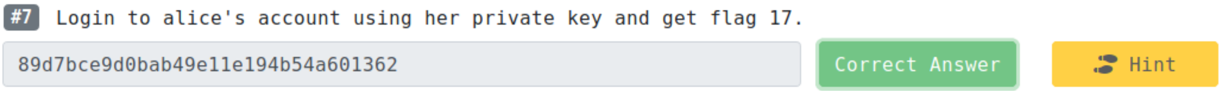
#6 Flag 16 lies within another system mount.

cab4b7cae33c87794d82efa1e7f834e6

This one was quite hard for me to find as I wasn't sure where to look for system mounts. I eventually found that mounted items are stored in the */media* folder and decided to look there. This led me through a bunch of directories that spelled out flag 16 and we were able to find the flag.



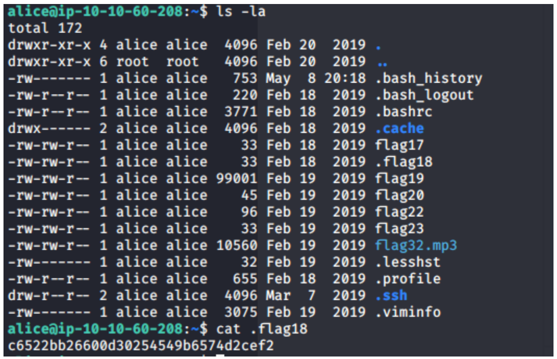
```
alice@ip-10-10-60-208:~$ ls
flag17  flag19  flag20  flag22  flag23  flag32.mp3
alice@ip-10-10-60-208:~$ cat flag17
89d7bce9d0bab49e11e194b54a601362
```



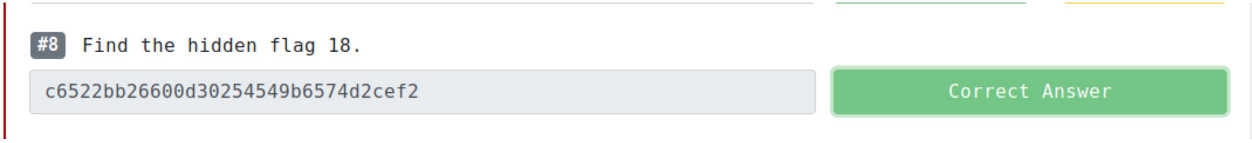
#7 Login to alice's account using her private key and get flag 17.

89d7bce9d0bab49e11e194b54a601362

Flag 17 was easy. Since we were already logged into Alice's account and knew where flag 17 was I simply looked at the file and was able to find it.



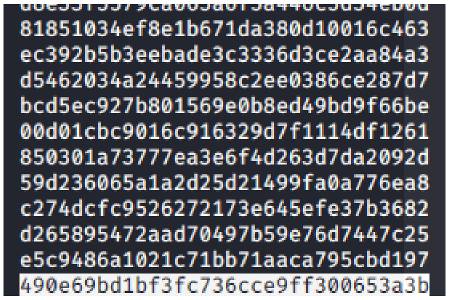
```
alice@ip-10-10-60-208:~$ ls -la
total 172
drwxr-xr-x 4 alice alice  4096 Feb 20  2019 .
drwxr-xr-x 6 root  root   4096 Feb 20  2019 ..
-rw------- 1 alice alice   753 May  8 20:18 .bash_history
-rw-r--r-- 1 alice alice   220 Feb 18  2019 .bash_logout
-rw-r--r-- 1 alice alice  3771 Feb 18  2019 .bashrc
drwx------ 2 alice alice  4096 Feb 18  2019 .cache
-rw-rw-r-- 1 alice alice    33 Feb 18  2019 flag17
-rw-rw-r-- 1 alice alice    33 Feb 18  2019 .flag18
-rw-rw-r-- 1 alice alice 99001 Feb 19  2019 flag19
-rw-rw-r-- 1 alice alice    45 Feb 19  2019 flag20
-rw-rw-r-- 1 alice alice    96 Feb 19  2019 flag22
-rw-rw-r-- 1 alice alice    33 Feb 19  2019 flag23
-rw-rw-r-- 1 alice alice 10560 Feb 19  2019 flag32.mp3
-rw------- 1 alice alice    32 Feb 19  2019 .lesshst
-rw-r--r-- 1 alice alice   655 Feb 18  2019 .profile
drw-r--r-- 2 alice alice  4096 Mar  7  2019 .ssh
-rw------- 1 alice alice  3075 Feb 19  2019 .viminfo
alice@ip-10-10-60-208:~$ cat .flag18
c6522bb26600d30254549b6574d2cef2
```



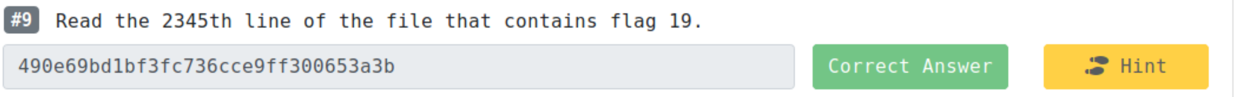
#8 Find the hidden flag 18.

c6522bb26600d30254549b6574d2cef2

This task was also simple. We used *ls -la* to show all files in the current directory. This gave us the hidden file for flag 18.



```
81851034ef8e1b671da380d10016c463
ec392b5b3eebade3c3336d3ce2aa84a3
d5462034a24459958c2ee0386ce287d7
bcd5ec927b801569e0b8ed49bd9f66be
00d01cbc9016c916329d7f1114df1261
850301a73777ea3e6f4d263d7da2092d
59d236065a1a2d25d21499fa0a776ea8
c274dcfc9526272173e645efe37b3682
d265895472aad70497b59e76d7447c25
e5c9486a1021c71bb71aaca795cbd197
490e69bd1bf3fc736cce9ff300653a3b
```



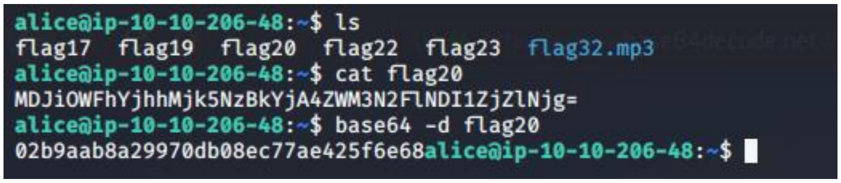
#9 Read the 2345th line of the file that contains flag 19.

490e69bd1bf3fc736cce9ff300653a3b

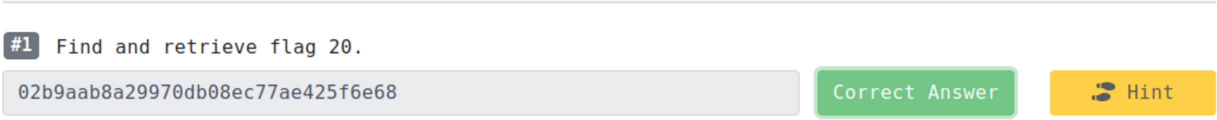
This was also simple as I used the command *head -n 2345 file19.* The -n is used to read lines instead of bytes which is -c. The last line outputted to screen was your flag!!

Task 4: Data Representation, Strings, and Permissions

Now we are moving onto the fourth task of the CTF learning about data representation, strings, and permissions. Now since we are still logged into Alice's account we can find flag20 fairly easily. Finding the contents of the flag was kinda weird. When we used the cat command you can see that this is not the flag. I then used the hint and it said "*base64*". I then figured I could use the base64 command to decrypt the message.



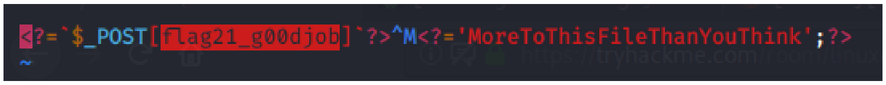
```
alice@ip-10-10-206-48:~$ ls
flag17  flag19  flag20  flag22  flag23  flag32.mp3
alice@ip-10-10-206-48:~$ cat flag20
MDJiOWFhYjhhMjk5NzBkYjA4ZWM3N2FlNDI1ZjZlNjg=
alice@ip-10-10-206-48:~$ base64 -d flag20
02b9aab8a29970db08ec77ae425f6e68alice@ip-10-10-206-48:~$ 
```



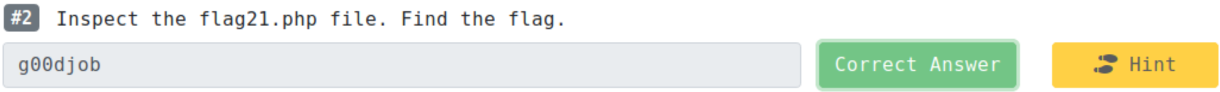
#1 Find and retrieve flag 20.

02b9aab8a29970db08ec77ae425f6e68

Flag 21 was simple as well as we just opened flag 21 in vim to see all the code of the PHP file. I was also surprised to see that this flag was very short.



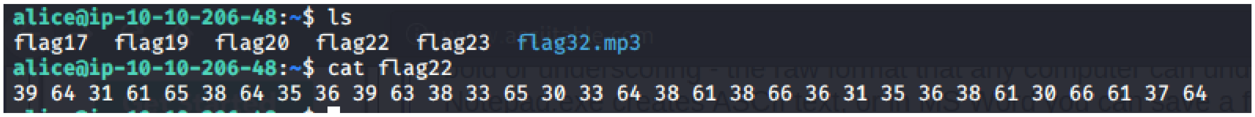
```
<?=`$_POST[flag21_g00djob]`?>^M<?='MoreToThisFileThanYouThink';?>
~
```



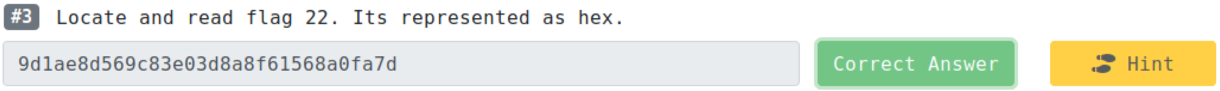
#2 Inspect the flag21.php file. Find the flag.

g00djob

For this flag, I brute-forced the answer, which was just a long and tedious process. We were given the hex values for the flag. I then looked at an ASCII table so I could translate it to the flag.



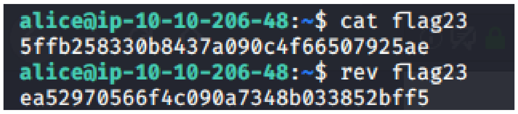
```
alice@ip-10-10-206-48:~$ ls
flag17  flag19  flag20  flag22  flag23  flag32.mp3
alice@ip-10-10-206-48:~$ cat flag22
39 64 31 61 65 38 64 35 36 39 63 38 33 65 30 33 64 38 61 38 66 36 31 35 36 38 61 30 66 61 37 64
```



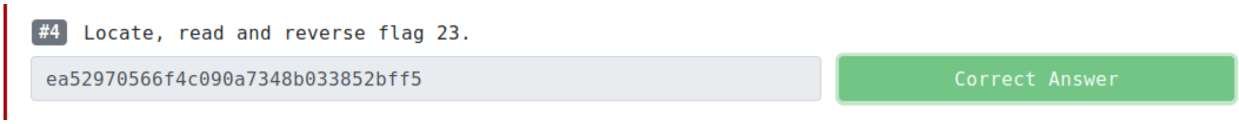
#3 Locate and read flag 22. Its represented as hex.

```
9d1ae8d569c83e03d8a8f61568a0fa7d
```

Flag 23 was also simple as I just reversed the string that was used for flag23.



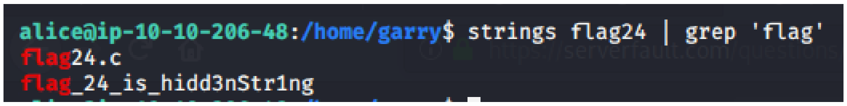
```
alice@ip-10-10-206-48:~$ cat flag23
5ffb258330b8437a090c4f66507925ae
alice@ip-10-10-206-48:~$ rev flag23
ea52970566f4c090a7348b033852bff5
```



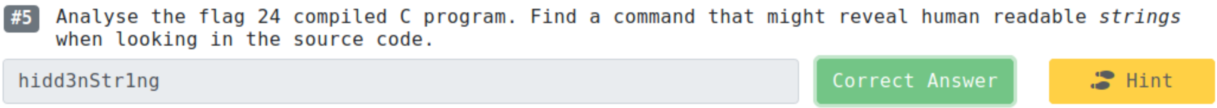
#4 Locate, read and reverse flag 23.

```
ea52970566f4c090a7348b033852bff5
```

For this flag, I used the strings command to print out all the readable strings in the flag24 executable. This printed out a bunch of information so I then used the grep command to search for ‘flag’ to get a better search.



```
alice@ip-10-10-206-48:/home/garry$ strings flag24 | grep 'flag'
flag24.c
flag_24_is_hidd3nStr1ng
```

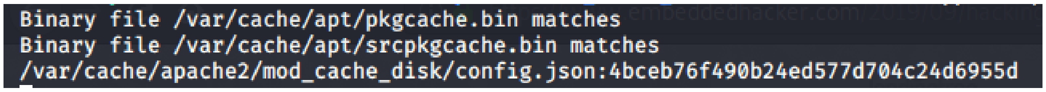


#5 Analyse the flag 24 compiled C program. Find a command that might reveal human readable *strings* when looking in the source code.

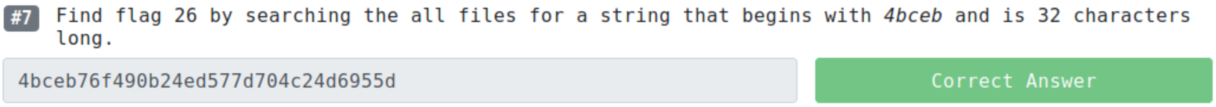
```
hidd3nStr1ng
```

There was no flag 25 ☹

I was able to find this flag after one-two hours of research on the different command-line syntax to use but we ended up finding it in the /var directory.



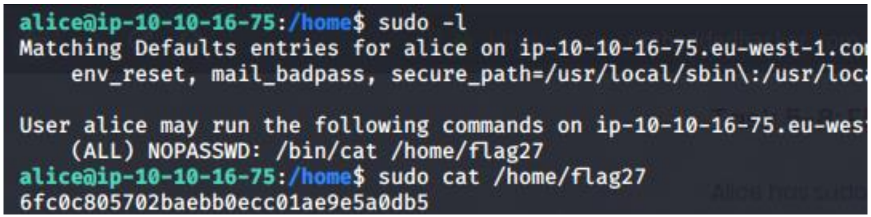
```
Binary file /var/cache/apt/pkgcache.bin matches
Binary file /var/cache/apt/srcpkgcache.bin matches
/var/cache/apache2/mod_cache_disk/config.json:4bceb76f490b24ed577d704c24d6955d
```



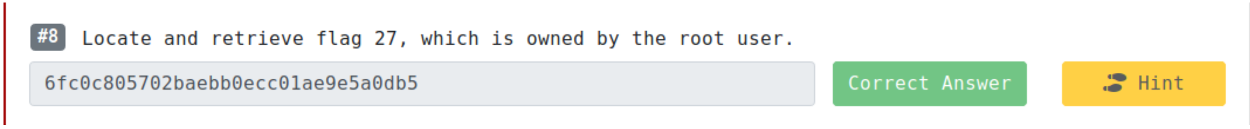
#7 Find flag 26 by searching the all files for a string that begins with *4bceb* and is 32 characters long.

```
4bceb76f490b24ed577d704c24d6955d
```

For this flag, it was super simple. We can use the command *sudo -l* to see all the files that sudo can access without a password. We can see that we don't need a password for flag27 so we run *sudo cat /home/flag27* and are given the flag.



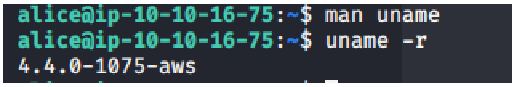
```
alice@ip-10-10-16-75:/home$ sudo -l
Matching Defaults entries for alice on ip-10-10-16-75.eu-west-1.co
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loc

User alice may run the following commands on ip-10-10-16-75.eu-wes
    (ALL) NOPASSWD: /bin/cat /home/flag27
alice@ip-10-10-16-75:/home$ sudo cat /home/flag27
6fc0c805702baebb0ecc01ae9e5a0db5
```



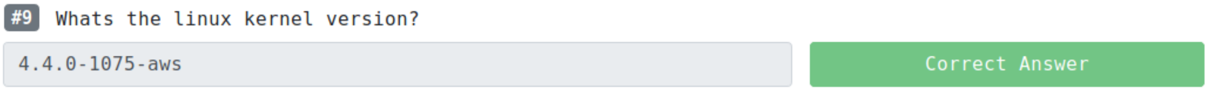
#8 Locate and retrieve flag 27, which is owned by the root user.

6fc0c805702baebb0ecc01ae9e5a0db5

Using the uname command with the -r parameter allows us to get the kernel release version. I also used the -v parameter but this did not match the answer format so I switch it the -r.



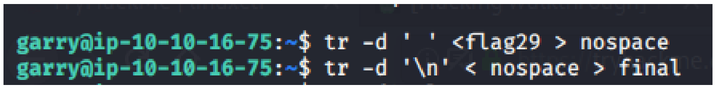
```
alice@ip-10-10-16-75:~$ man uname
alice@ip-10-10-16-75:~$ uname -r
4.4.0-1075-aws
```



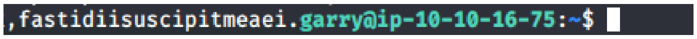
#9 Whats the linux kernel version?

4.4.0-1075-aws

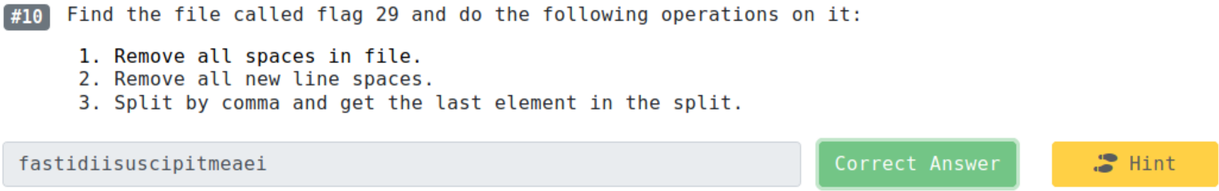
I used the tr command to remove all the white space and then the new lines as well. I then looked at the final to see the last comma and saw that it was at the end of the file so assumed that was the flag and I was correct.



```
garry@ip-10-10-16-75:~$ tr -d ' ' <flag29 > nospace
garry@ip-10-10-16-75:~$ tr -d '\n' < nospace > final
```



```
,fastidiisuscipitmeaei.garry@ip-10-10-16-75:~$
```



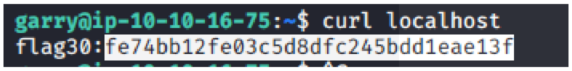
#10 Find the file called flag 29 and do the following operations on it:

1. Remove all spaces in file.
2. Remove all new line spaces.
3. Split by comma and get the last element in the split.

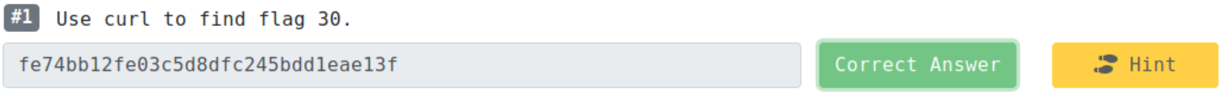
fastidiisuscipitmeaei

Task 6: MySQL, FTP, groups, and RDP

Now we are onto the last tasks in this CTF. I wasn’t sure how to go about this first task since we need to use the curl command and were not given an address. I used the hint and it suggested seeing if anything was running on localhost. I then was able to get flag30.



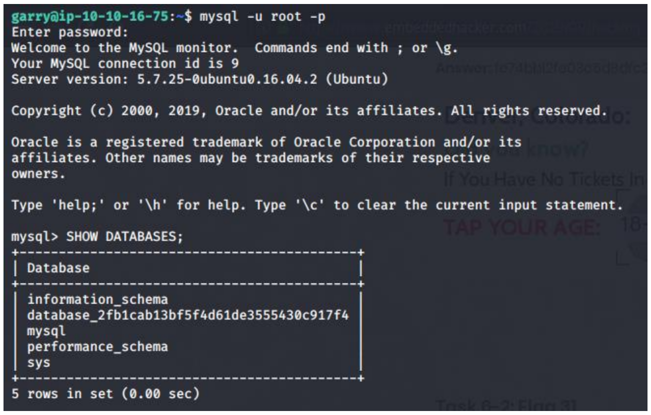
```
garry@ip-10-10-16-75:~$ curl localhost
flag30:fe74bb12fe03c5d8dfc245bdd1eae13f
```



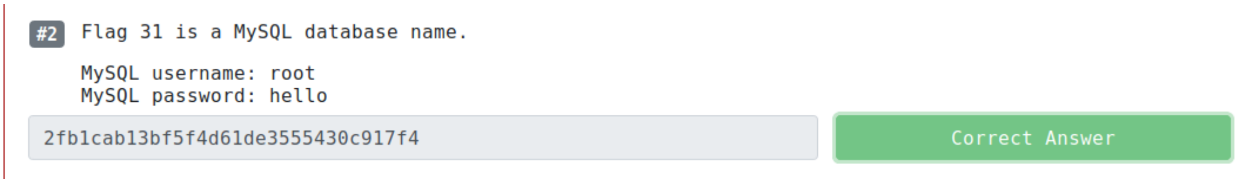
#1 Use curl to find flag 30.

fe74bb12fe03c5d8dfc245bdd1eae13f

Flag 31 had us log into a MySQL database. I have not worked too much in this environment so I had to do some additional research but I ended up using the SHOW DATABASES; command and was able to find flag31.



```
garry@ip-10-10-16-75:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.25-0ubuntu0.16.04.2 (Ubuntu)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------------------------------+
| Database                                   |
+--------------------------------------------+
| information_schema                         |
| database_2fb1cab13bf5f4d61de3555430c917f4  |
| mysql                                      |
| performance_schema                         |
| sys                                        |
+--------------------------------------------+
5 rows in set (0.00 sec)
```

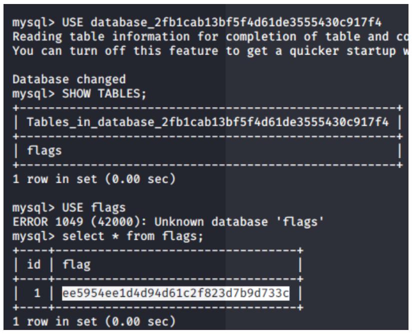


#2 Flag 31 is a MySQL database name.

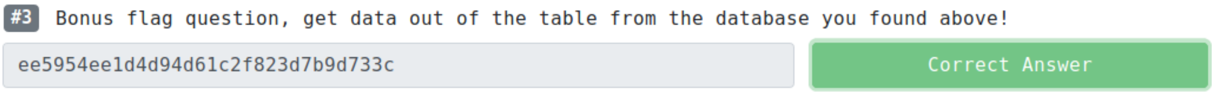
```
MySQL username: root
MySQL password: hello
```

2fb1cab13bf5f4d61de3555430c917f4

Going off the last flag I needed to look up syntax but was still able to extract the data from the flag table inside the database with flag31.



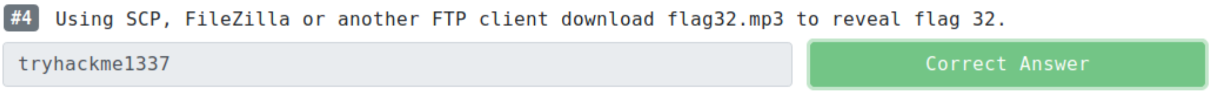
```
mysql> USE database_2fb1cab13bf5f4d61de3555430c917f4
Reading table information for completion of table and co
You can turn off this feature to get a quicker startup w

Database changed
mysql> SHOW TABLES;
+--------------------------------------------------+
| Tables_in_database_2fb1cab13bf5f4d61de3555430c917f4 |
+--------------------------------------------------+
| flags                                            |
+--------------------------------------------------+
1 row in set (0.00 sec)

mysql> USE flags
ERROR 1049 (42000): Unknown database 'flags'
mysql> select * from flags;
+----+----------------------------------+
| id | flag                             |
+----+----------------------------------+
|  1 | ee5954ee1d4d94d61c2f823d7b9d733c |
+----+----------------------------------+
1 row in set (0.00 sec)
```



#3 Bonus flag question, get data out of the table from the database you found above!

ee5954ee1d4d94d61c2f823d7b9d733c

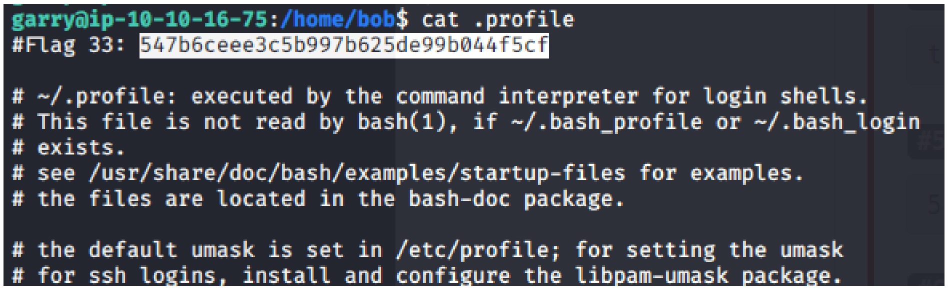
This flag was cool because I had ever worked with Filezilla and it was cool to learn another tool on Kali Linux. I had to install Filezilla but after that was able to connect to Alice and transfer her file onto my machine and was able to listen to it. It gave the password “tryhackme1337”.



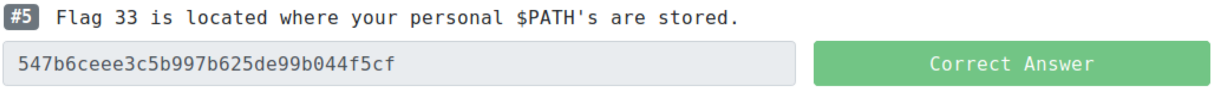
#4 Using SCP, FileZilla or another FTP client download flag32.mp3 to reveal flag 32.

tryhackme1337

This was a trial and error flag. I knew the $PATH’s could be found in the .profile file in each user but I did not know what user the flag was in. I finally found it in Bob's profile.



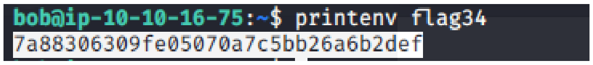
```
garry@ip-10-10-16-75:/home/bob$ cat .profile
#Flag 33: 547b6ceee3c5b997b625de99b044f5cf

# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
```



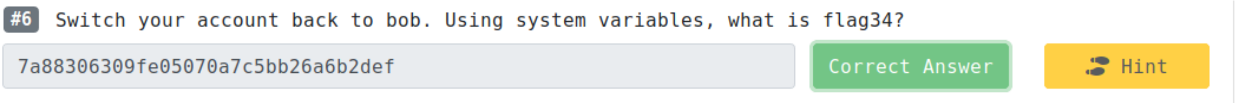
#5 Flag 33 is located where your personal $PATH's are stored.

547b6ceee3c5b997b625de99b044f5cf

This flag was easy I just printed out the system variable that was associated with flag34.



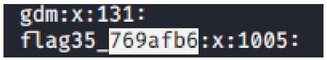
```
bob@ip-10-10-16-75:~$ printenv flag34
7a88306309fe05070a7c5bb26a6b2def
```



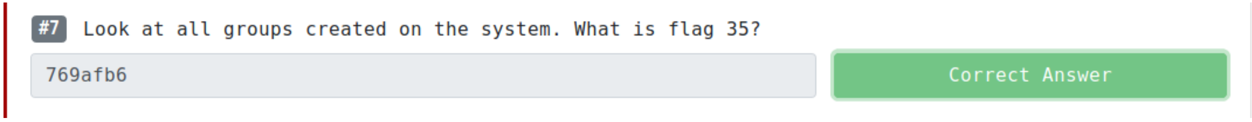
#6 Switch your account back to bob. Using system variables, what is flag34?

7a88306309fe05070a7c5bb26a6b2def

For this flag, I used the cat command to view the contents in the /etc/group folder. Flag 35 was given at the bottom.



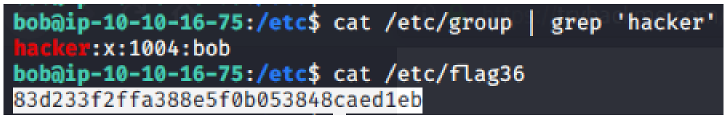
```
gdm:x:131:
flag35_769afb6:x:1005:
```



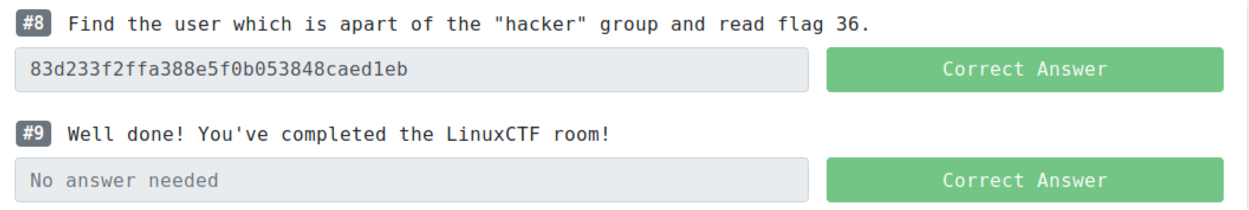
#7 Look at all groups created on the system. What is flag 35?

769afb6

This was the last flag in the CTF and was fun to find. I was able to see that bob was part of the 'hacker' group. We also knew that the flag was stored in the /etc directory from exploring all the directories throughout the CTF. I was able to get the last flag and finish the Linux CTF.



```
bob@ip-10-10-16-75:/etc$ cat /etc/group | grep 'hacker'
hacker:x:1004:bob
bob@ip-10-10-16-75:/etc$ cat /etc/flag36
83d233f2ffa388e5f0b053848caed1eb
```

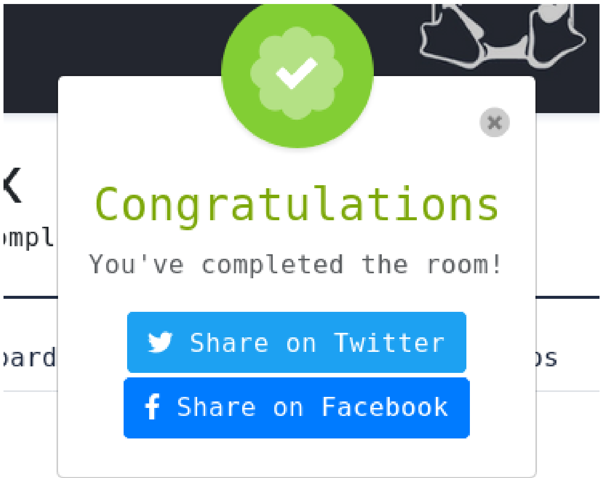


#8 Find the user which is apart of the "hacker" group and read flag 36.

83d233f2ffa388e5f0b053848caed1eb

#9 Well done! You've completed the LinuxCTF room!

No answer needed



Congratulations

You've completed the room!

*****END NOTE*****

This was my first CTF on tryhackme as I am trying to grow my skills over the summer. I enjoyed exploring this machine and all the things that we can do with Linux. I hope you enjoyed reading this and can find this informative.

*If you have any questions please feel free to contact me at matthew.barich@western.edu. GOOD LUCK AND HAPPY HUNTING!!!