Assignment 6

SUID Use Case

SUID is nothing but the **special permission** given to **a user** to a **program/file** with the permission of the **file owner**. SUID stands for set user ID.

Normal User → having limited command access

So SUID is specially used to give access to execute particular commands from normal users. IT means that with the help of SUID we can permit the normal user to execute a particular command that the normal user does not have permission.

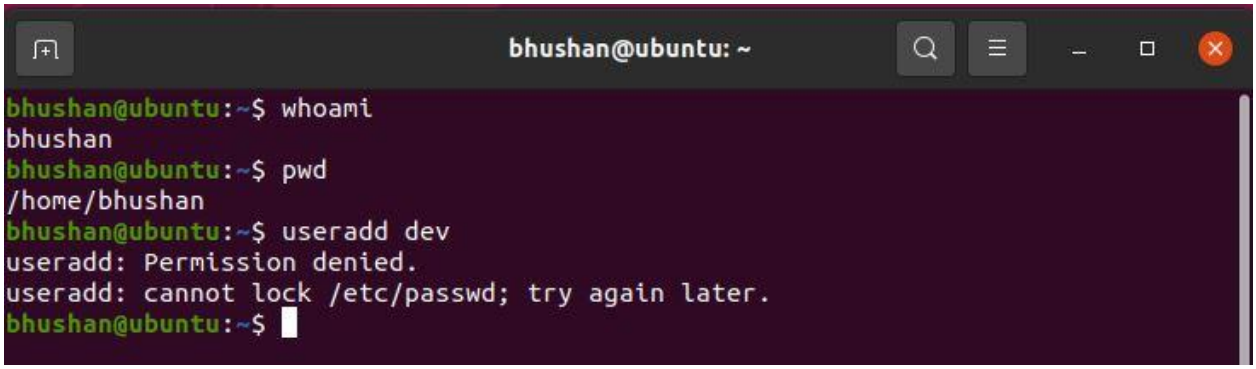
We can **apply** SUID by using 2 methods :

1) Symbolic Method → chmod u+s <command file path>
2) Numeric Method → chmod 4755 <command file path>

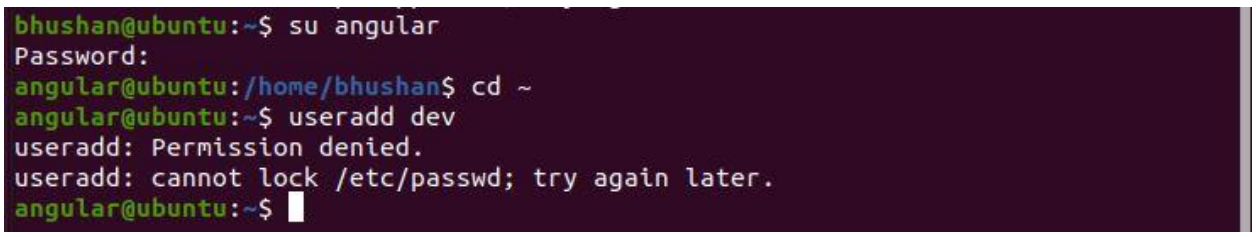
We can **remove** SUID by using 2 methods :

1) Symbolic Method → chmod u-s <command file path>
2) Numeric Method → chmod 755 <command file path>

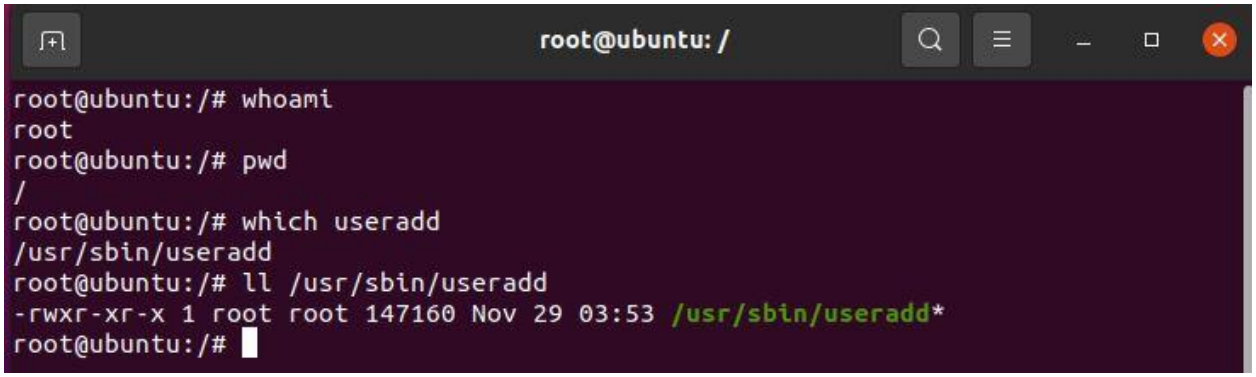
Point 1: If we login with a normal user and try to execute useradd command then we get an error message. Because normal users didn't have permission to run or execute useradd commands are as follows :-



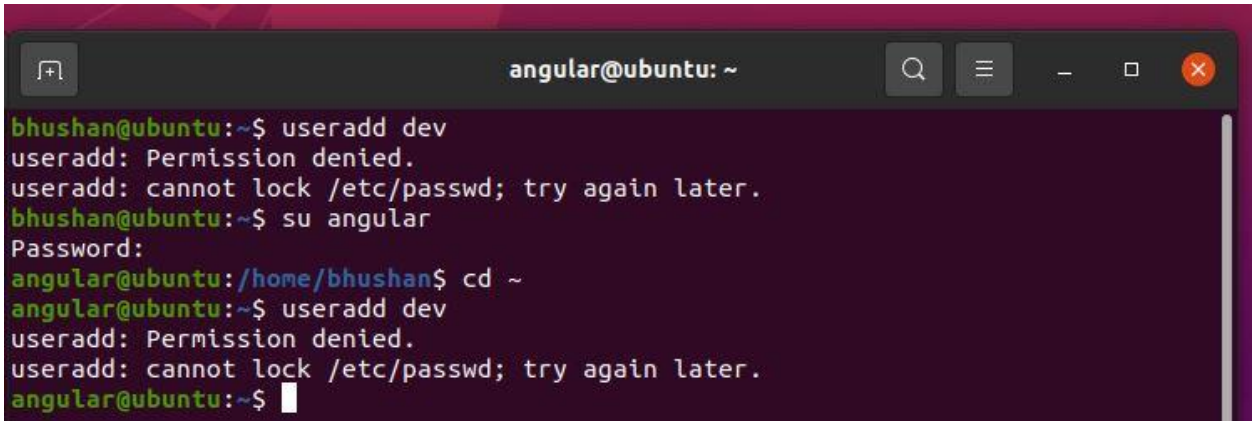
If we login with another normal user angular and try to execute useradd command then we get an error message. Because normal users didn't have permission to run or execute useradd commands are as follows :-



Point 2: Now , Login with root user and try to check from where the useradd command will get executed or run with the help of which command and check the permission of the file or directory only the user has execute permission,group and others have execute permission.



Point 3: The user ,group and others have execute permission. Login with a normal user and try to execute useradd command. First we logged in with bhushan user and then with angular. From both we are unable to run this command.



Point 4: So to solve the above problem or to give special permission we use SUID. so here login with root user and give special permission to users with numeric mode with the help of chmod command.

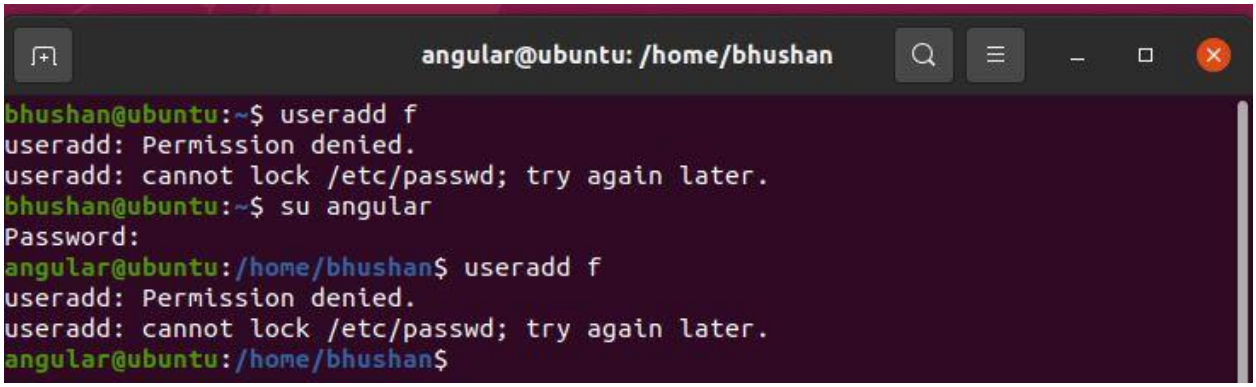
Point 5: After giving the permission, log in with a normal user and try to execute useradd command. First we logged in with bhushan user and then with angular. From both we are able to run this command.

```
bhushan@ubuntu:~$ useradd tester
bhushan@ubuntu:~$ cat /etc/passwd | grep tester
tester:x:1010:1011::/home/tester:/bin/sh
bhushan@ubuntu:~$ su angular
Password:
angular@ubuntu:/home/bhushan$ cd ~
angular@ubuntu:~$ useradd manager
angular@ubuntu:~$ cat /etc/passwd | grep manager
manager:x:1011:1012::/home/manager:/bin/sh
angular@ubuntu:~$
```

Point 6: Now , we remove the SUID from the user. We use the chmod command in numeric mode from the root.

```
root@ubuntu:/# ll /usr/sbin/useradd
-rwsr-xr-x 1 root root 147160 Nov 29 03:53 /usr/sbin/useradd*
root@ubuntu:/# chmod 755 /usr/sbin/useradd
root@ubuntu:/# ll /usr/sbin/useradd
-rwxr-xr-x 1 root root 147160 Nov 29 03:53 /usr/sbin/useradd*
root@ubuntu:/#
```

And now if we try to use the useradd command from a normal user then we are unable to execute it as we remove the SUID.



Assignment 8

Sticky Bit Use Case

Sticky bit is nothing but a permission bit which is set on a file or folder, thereby permitting only the owner or root user of the file or folder to modify, rename or delete the concerned directory or file. In other words, **Sticky bit is a permission bit which is used to control all other users. It basically restricts the delete or modify operations.**

We can **apply** Sticky bit by using 2 methods :

1) Symbolic Method → chmod o+t <file or directory path>
2) Numeric Method → chmod 1777 <file or directory path>

We can **remove** SUID by using 2 methods :

3) Symbolic Method → chmod o-t <file or directory path>
4) Numeric Method → chmod 777 <file or directory path>

Point 1: Consider there are 3 users one is root and another 2 is normal user that is bhushan and angular in may case. Log in with root user and create one directory called impdata and check the permission of it. Users have all access but groups and others have read and execute access. Give them all permission to directory impdata.

```
root@ubuntu:/# mkdir impdata
root@ubuntu:/# ls -la | grep impdata
drwxr-xr-x   2 root root      4096 Feb  9 09:31 impdata
root@ubuntu:/# chmod 777 impdata
root@ubuntu:/# ls -la | grep impdata
drwxrwxrwx   2 root root      4096 Feb  9 09:31 impdata
root@ubuntu:/#
```

Point 2: Login with bhushan user go inside the impdata and make 3 directories using mkdir. Similarly , login with angular user go inside the impdata and make 3 files. After that, bhushan users can read and modify the angular user's file or directory and angular users can read and modify the bhushan user's file or directory.

```
bhushan@ubuntu:~$ cd /
bhushan@ubuntu:/$ ls
bin    dev   impdata  lib64       media  proc  sbin  swapfile  usr
boot   etc   lib      libx32      mnt    root  snap  sys       var
cdrom  home  lib32    lost+found  opt    run   srv   tmp
bhushan@ubuntu:/$ cd impdata/
bhushan@ubuntu:/impdata$ mkdir bhushan_dir{1..3}
bhushan@ubuntu:/impdata$ ls
bhushan_dir1  bhushan_dir2  bhushan_dir3
bhushan@ubuntu:/impdata$
```

```
angular@ubuntu:~$ cd /
angular@ubuntu:/$ cd impdata/
angular@ubuntu:/impdata$ ls
bhushan_dir1  bhushan_dir2  bhushan_dir3
angular@ubuntu:/impdata$ touch angular_file{1..4}
angular@ubuntu:/impdata$ ls
angular_file1  angular_file3  bhushan_dir1  bhushan_dir3
angular_file2  angular_file4  bhushan_dir2
angular@ubuntu:/impdata$
```

If we delete the file from bhushan user which is owned by angular user and vice versa then it is possible.

```
bhushan@ubuntu:/impdata$ ls
bhushan_dir1  bhushan_dir2  bhushan_dir3
bhushan@ubuntu:/impdata$ rm -rf angular_file3
bhushan@ubuntu:/impdata$ ls
angular_file1  angular_file2  angular_file4  bhushan_dir1  bhushan_dir2
bhushan@ubuntu:/impdata$
```

```
angular@ubuntu:/impdata$ ls
angular_file1  angular_file3  bhushan_dir1  bhushan_dir3
angular_file2  angular_file4  bhushan_dir2
angular@ubuntu:/impdata$ rm -rf bhushan_dir3
angular@ubuntu:/impdata$ ls
angular_file1  angular_file3  bhushan_dir1
angular_file2  angular_file4  bhushan_dir2
angular@ubuntu:/impdata$
```

Point 3: Any user can access any user's files or directories so to restrict delete or modify operations we use sticky bit. Now, we apply sticky bit on impdata directory with numeric mode.

```
root@ubuntu:/# ls -la | grep impdata
drwxrwxrwx   2 root root      4096 Feb  9 09:31 impdata
root@ubuntu:/# chmod 1777 impdata
root@ubuntu:/# ls -la | grep impdata
drwxrwxrwt   4 root root      4096 Feb  9 09:45 impdata
root@ubuntu:/#
```

Now try to modify or delete the file or directory from bhushan and angular user, we get an operation permitted error.

```
bhushan@ubuntu:/impdata$ ls
angular_file1  angular_file2  angular_file4  bhushan_dir1  bhushan_dir2
bhushan@ubuntu:/impdata$ rm -rf angular_file1
rm: cannot remove 'angular_file1': Operation not permitted
bhushan@ubuntu:/impdata$ rm -rf angular_file2
rm: cannot remove 'angular_file2': Operation not permitted
bhushan@ubuntu:/impdata$
```

```
angular@ubuntu:/impdata$ ls
angular_file1  angular_file3  bhushan_dir1
angular_file2  angular_file4  bhushan_dir2
angular@ubuntu:/impdata$ rm -rf bhushan_dir1
rm: cannot remove 'bhushan_dir1': Operation not permitted
angular@ubuntu:/impdata$
```

Point 4: If we want to remove the sticky bit, we use a numeric method to revoke it.

```
root@ubuntu:/# ls -la | grep impdata
drwxrwxrwt   4 root root      4096 Feb  9 09:45 impdata
root@ubuntu:/# chmod 777 impdata
root@ubuntu:/# ls -la | grep impdata
drwxrwxrwx   4 root root      4096 Feb  9 09:45 impdata
root@ubuntu:/#
```

After removing the sticky bit try to delete the file or directory we are able to delete it easily.

```
bhushan@ubuntu:/impdata$ rm -rf angular_file1
bhushan@ubuntu:/impdata$ rm -rf angular_file2
bhushan@ubuntu:/impdata$ ls
angular_file4  bhushan_dir1  bhushan_dir2
bhushan@ubuntu:/impdata$ rm -rf angular_file4
bhushan@ubuntu:/impdata$
```

```
angular@ubuntu:/impdata$ rm -rf bhushan_dir1
angular@ubuntu:/impdata$ rm -rf bhushan_dir2
angular@ubuntu:/impdata$ ls
angular@ubuntu:/impdata$
```

Assignment 7

GUID Use Case

GUID is nothing but the **special permission** given to **a group** which is used to inherit the changed group to all the newly created sub directories / inclusive files within the parent directory. GUID stands for set group ID.

We can **apply** GUID by using 2 methods :

1) Symbolic Method → chmod g+s <file or Dir>
2) Numeric Method → chmod 2744 <file or Dir>

We can **remove** GUID by using 2 methods :

1) Symbolic Method → chmod g-s <file or Dir>
2) Numeric Method → chmod 2744 <file or Dir>

Point 1: Consider , if we create a directory inside the directory and create 2 files. Check the user owner and group owner of the directory as well as files. All the directories and files having root as the user owner and group owner.

```
root@ubuntu:/# mkdir demo course
root@ubuntu:/# ls
bin    course  etc   lib32   lost+found  opt   run   srv       tmp
boot   demo    home  lib64   media       proc  sbin  swapfile  usr
cdrom  dev     lib   libx32  mnt         root  snap  sys       var
root@ubuntu:/# cd course/
root@ubuntu:/course# mkdir bca bsc
```

```
root@ubuntu:/course# ls -la
total 16
drwxr-xr-x  4 root root 4096 Feb  9 10:44 .
drwxr-xr-x 22 root root 4096 Feb  9 10:43 ..
drwxr-xr-x  2 root root 4096 Feb  9 10:44 bca
drwxr-xr-x  2 root root 4096 Feb  9 10:44 bsc
root@ubuntu:/course# cd bca/
root@ubuntu:/course/bca# touch bca_file{1..2}
root@ubuntu:/course/bca# ls -la
total 8
drwxr-xr-x 2 root root 4096 Feb  9 10:46 .
drwxr-xr-x 4 root root 4096 Feb  9 10:44 ..
-rw-r--r-- 1 root root    0 Feb  9 10:46 bca_file1
-rw-r--r-- 1 root root    0 Feb  9 10:46 bca_file2
root@ubuntu:/course/bca#
```

Point 2: Now, if we change the group owner of the main directory i.e course then check whether all the directories inside it and files have the same reflection or not. But observations say that only the main directory i.e course shows the changed group not inside files or directories.

```
root@ubuntu:/# ls
bin    course  etc   lib32   lost+found  opt   run   srv       tmp
boot   demo    home  lib64   media       proc  sbin  swapfile  usr
cdrom  dev     lib   libx32  mnt         root  snap  sys       var
root@ubuntu:/# chgrp devops course
root@ubuntu:/# ls -la | grep course
drwxr-xr-x   4 root devops      4096 Feb  9 10:44 course
root@ubuntu:/# ls -la | grep /course
root@ubuntu:/# cd course/
root@ubuntu:/course# ls -la | grep bca
drwxr-xr-x  2 root root   4096 Feb  9 10:46 bca
root@ubuntu:/course# cd bca
root@ubuntu:/course/bca# ls -la
total 8
drwxr-xr-x 2 root root   4096 Feb  9 10:46 .
drwxr-xr-x 4 root devops 4096 Feb  9 10:44 ..
-rw-r--r-- 1 root root      0 Feb  9 10:46 bca_file1
-rw-r--r-- 1 root root      0 Feb  9 10:46 bca_file2
root@ubuntu:/course/bca#
```

Point 3: Now, we provide special permission to the group i.e GUID with numeric command. So after giving the GUID then if we change the group owner this time there will be reflection in all files and directories showing the changed owner.

```
root@ubuntu:/# ls -la | grep course
d--x--x--x   4 root devops      4096 Feb  9 10:44 course
root@ubuntu:/# chmod 2755 course
root@ubuntu:/# ls -la | grep course
drwxr-sr-x   4 root devops      4096 Feb  9 10:44 course
root@ubuntu:/# cd course
root@ubuntu:/course# mkdir faculty
root@ubuntu:/course# ls -la | grep faculty
drwxr-sr-x  2 root devops 4096 Feb  9 11:08 faculty
root@ubuntu:/course# cd faculty
root@ubuntu:/course/faculty# touch file1
root@ubuntu:/course/faculty# ls -la | grep file1
-rw-r--r-- 1 root devops    0 Feb  9 11:08 file1
root@ubuntu:/course/faculty#
```

Point 4: If we want to revoke the permission of the guid, we use a numeric method to revoke it.

```
root@ubuntu:/# ls -la | grep course
drwxr-sr-x   3 root root      4096 Feb  9 11:25 course
root@ubuntu:/# chmod 755 course
root@ubuntu:/# ls -la | grep course
drwxr-sr-x   3 root root      4096 Feb  9 11:25 course
root@ubuntu:/# chmod g-s course
root@ubuntu:/# ls -la | grep course
drwxr-xr-x   3 root root      4096 Feb  9 11:25 course
root@ubuntu:/#
```

After revoking permission , if we create directories or files then we are unable to inherit the group ownership of the directory.

Note:-When it's set on directories, all new files in the directory inherit the group ownership of the directory.