

Advanced/Special Permissions : SUID & GUID

Snehal Deshmukh

We create a file script.sh & try to excute in another user it gives you error perm.denied :

```
File Edit View Search Terminal Help
snehal@ubuntu:/home$ ls -la
total 48
drwsr-xr-x  11 root          root          4096 Feb 13 00:06 .
drwxr-xr-x  24 root          root          4096 Feb  9 21:23 ..
drwxr-xr-x   3 demo          devops       4096 Feb 13 00:04 demo
drwxr-xr-x   3 root          root         4096 Jan 24 18:16 deshmutk
drwxrwxr-x   4 devops        devops       4096 Feb  7 08:53 devops
drwxrwxr-x+  2 root          root         4096 Feb 10 23:42 facl
drwxr-xr-x   2 root          root         4096 Feb 12 10:01 script
-rw-r--r--   1 root          root           0 Feb 13 00:06 script.sh
drwxr-xr-x  23 snehal        snehal       4096 Feb 12 23:23 snehal
-rwxr-xr-x   1 snehal        root          33 Feb 12 09:57 test.sh
drwxr-xr-x   3 user1_facl    facl_grp     4096 Feb 12 21:25 user1_facl
drwxr-xr-x   3 user2_facl    facl_grp     4096 Feb 12 21:31 user2_facl
drwxr-xr-x   2 user2_facl_grp user2_facl_grp 4096 Feb 11 06:33 user2_facl_grp
snehal@ubuntu:/home$
```

So we set the SUID bit on it. To do this using the numeric method :

```
snehal@ubuntu:/home$ sudo chmod 4755 script.sh
snehal@ubuntu:/home$ ls -l
total 40
drwxr-xr-x  3 demo          devops          4096 Feb 13 00:04 demo
drwxr-xr-x  3 root         root            4096 Jan 24 18:16 desh mukh
drwxrwxr-x  4 devops       devops          4096 Feb  7 08:53 devops
drwxrwxr-x+ 2 root         root            4096 Feb 10 23:42 fac l
drwxr-xr-x  2 root         root            4096 Feb 12 10:01 script
-rwsr-xr-x  1 root         root            0 Feb 13 00:06 script.sh
drwxr-xr-x 23 snehal       snehal          4096 Feb 12 23:23 snehal
-rwxr-xr-x  1 snehal       root            33 Feb 12 09:57 test.sh
drwxr-xr-x  3 user1_fac l  fac l_grp       4096 Feb 12 21:25 user1_fac l
drwxr-xr-x  3 user2_fac l  fac l_grp       4096 Feb 12 21:31 user2_fac l
drwxr-xr-x  2 user2_fac l_grp user2_fac l_grp 4096 Feb 11 06:33 user2_fac l_grp
snehal@ubuntu:/home$
```

When you set the file permissions to 4755 (which is equivalent to `rwxr-sr-x`), it means that the owner of the file has read, write, and execute permissions, while others (that is, users who are not the owner and do not belong to the same group as the owner) have only execute permissions. This means that other users will not be able to write to the file.

```
snehal@ubuntu:/home$ sudo chmod 4755 script.sh
snehal@ubuntu:/home$ ls -la
total 52
drwsr-xr-x 11 root      root      4096 Feb 13 00:21 .
drwxr-xr-x 24 root      root      4096 Feb  9 21:23 ..
drwxr-xr-x  3 demo      devops   4096 Feb 13 00:22 demo
drwxr-xr-x  3 root      root     4096 Jan 24 18:16 deshmkh
drwxrwxr-x  4 devops    devops   4096 Feb  7 08:53 devops
drwxrwxr-x+ 2 root      root     4096 Feb 10 23:42 fac1
drwxr-xr-x  2 root      root     4096 Feb 12 10:01 script
-rwsr-xr-x  1 root      root      27 Feb 13 00:21 script.sh
drwxr-xr-x 23 snehal     root     4096 Feb 13 00:20 snehal
-rwxr-xr-x  1 snehal     root      33 Feb 12 09:57 test.sh
drwxr-xr-x  3 user1_fac1 fac1_grp  4096 Feb 12 21:25 user1_fac1
drwxr-xr-x  3 user2_fac1 fac1_grp  4096 Feb 12 21:31 user2_fac1
drwxr-xr-x  2 user2_fac1_grp user2_fac1_grp 4096 Feb 11 06:33 user2_fac1_grp
snehal@ubuntu:/home$ su demo
Password:
demo@ubuntu:/home$ ./script.sh
hello
```

chmod 755 script.sh

that is, owner has read, write, and execute permissions, while group and others have read and execute permissions only). This effectively removes the SUID bit from the file.

```
snehal@ubuntu:/home$ sudo chmod 755 script.sh
snehal@ubuntu:/home$ ls -l
total 44
drwxr-xr-x  3 demo          devops          4096 Feb 13 00:22 demo
drwxr-xr-x  3 root          root            4096 Jan 24 18:16 deshmukh
drwxrwxr-x  4 devops        devops          4096 Feb  7 08:53 devops
drwxrwxr-x+ 2 root          root            4096 Feb 10 23:42 fac1
drwxr-xr-x  2 root          root            4096 Feb 12 10:01 script
-rwxr-xr-x  1 root          root            27 Feb 13 00:21 script.sh
drwxr-xr-x 23 snehal        root            4096 Feb 13 00:20 snehal
-rwxr-xr-x  1 snehal        root            33 Feb 12 09:57 test.sh
drwxr-xr-x  3 user1_fac1    fac1_grp        4096 Feb 12 21:25 user1_fac1
drwxr-xr-x  3 user2_fac1    fac1_grp        4096 Feb 12 21:31 user2_fac1
drwxr-xr-x  2 user2_fac1_grp user2_fac1_grp  4096 Feb 11 06:33 user2_fac1_grp
snehal@ubuntu:/home$
```

The file will run with the permissions of its owner

```
snehal@ubuntu:/home$ ./script.sh
hello
snehal@ubuntu:/home$ su demo
Password:
demo@ubuntu:/home$ ./script.sh
hello
demo@ubuntu:/home$ vim script.sh
demo@ubuntu:/home$
```

Before Set SUID	After Set SUID
The file will run with the permissions of the user who executes it.	The file will run with the permissions of its owner, regardless of who executes it.
The file will run with the permissions of its owner	The file will run with the permissions of its owner, regardless of who executes it.
generally considered a more secure configuration, as it reduces the attack surface of the file and makes it less likely to be used for malicious purposes.	setting the SUID bit on a file can be a security risk, as it allows anyone who can run the file to execute it with elevated privileges.

SGID- check directory info

```
snehal@ubuntu:/home/deshmukh$ cd ..  
snehal@ubuntu:/home$ getfacl deshmukh  
# file: deshmukh  
# owner: root  
# group: root  
user::rwx  
group::r-x  
other::r-x  
  
snehal@ubuntu:/home$
```


`sudo chmod 2777 deshmukh` : sets the SGID bit in the group permissions, which allows new subdirectories created within the directory to inherit the group ownership of the parent directory

```
snehal@ubuntu:/home$ sudo chmod 2777 deshmukh
snehal@ubuntu:/home$ getfacl deshmukh
# file: deshmukh
# owner: root
# group: root
# flags: -s-
user::rwx
group::rwx
other::rwx
```

```
snehal@ubuntu:/home$ ls -l
total 44
drwxr-xr-x  3 demo      devops      4096 Feb 13 01:02 demo
drwxrwsrwx  3 root      root        4096 Jan 24 18:16 deshmkh
drwxrwxr-x  4 devops    devops      4096 Feb  7 08:53 devops
drwxrwxr-x+ 2 root      root        4096 Feb 10 23:42 fac1
drwxr-xr-x  2 root      root        4096 Feb 12 10:01 script
-rwsr-xr-x  1 root      root        27 Feb 13 00:21 script.sh
drwxr-xr-x 23 snehal     root        4096 Feb 13 00:20 snehal
-rwxr-xr-x  1 snehal     root        33 Feb 12 09:57 test.sh
drwxr-xr-x  3 user1_fac1 fac1_grp    4096 Feb 12 21:25 user1_fac1
drwxr-xr-x  3 user2_fac1 fac1_grp    4096 Feb 12 21:31 user2_fac1
drwxr-xr-x  2 user2_fac1_grp user2_fac1_grp 4096 Feb 11 06:33 user2_fac1_grp
```

Additionally, the 7 in the group and others permissions fields grants full permissions (read, write, and execute) to the group and others.

```
react test1
snehal@ubuntu:/home/deshmukh$ sudo mkdir angular
snehal@ubuntu:/home/deshmukh$ ls -l
total 12
drwxr-sr-x 2 root root 4096 Feb 13 02:04 angular
drwxr-sr-x 2 root root 4096 Feb 13 02:03 react
drwxr-xr-x 3 root root 4096 Jan 24 18:17 test1
snehal@ubuntu:/home/deshmukh$ sudo touch angular.txt
snehal@ubuntu:/home/deshmukh$ ls -l
```