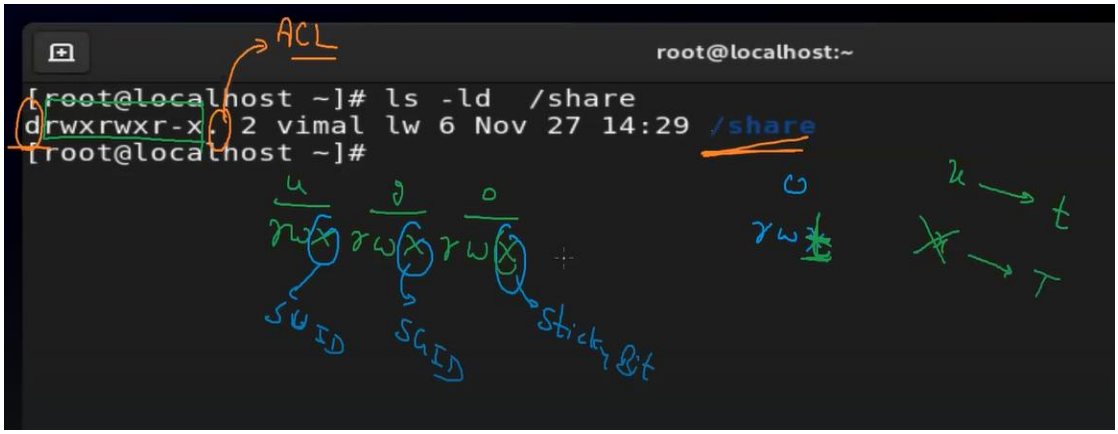
RHEL9

Session 10 – 26th Nov 2022 Summary

- The advanced permissions in linux is called as "**Special Permissions**". There are three types, for different use cases
    - Sticky Bit(t) - (1)
    - SGID (set group ID)- (s) - (2)
    - SUID(set user ID) – (s) – (4)
- The common folder "/share", shared by multiple users, "lw" is the group owner and the group has been given all the power "rwx". The users of this group automatically gets the power.

```
[root@localhost ~]# ls -ld  /share
drwxr-xr-x. 2 vimal lw 6 Nov 27 14:29 /share
[root@localhost ~]# chmod  g+rwx /share
[root@localhost ~]# ls -ld  /share
drwxrwxr-x. 2 vimal lw 6 Nov 27 14:29 /share
[root@localhost ~]#
```

- The challenge here is users in this group can create and delete files of other users in the group, to overcome this challenge we can use "**sticky bit**".
- The sticky bit is set in the common folder or shared folder



- Login as user “nitesh”, create a file “f.txt” in the shared folder “/share”

```
[root@localhost ~]# ls -ld  /share
drwxrwxr-x. 2 vimal lw 6 Nov 27 14:29 /share
[root@localhost ~]# su - nitesh
[nitesh@localhost ~]$ cd /share/
[nitesh@localhost share]$ touch f.txt
[nitesh@localhost share]$ ls -l
total 0
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 14:49 f.txt
[nitesh@localhost share]$
```

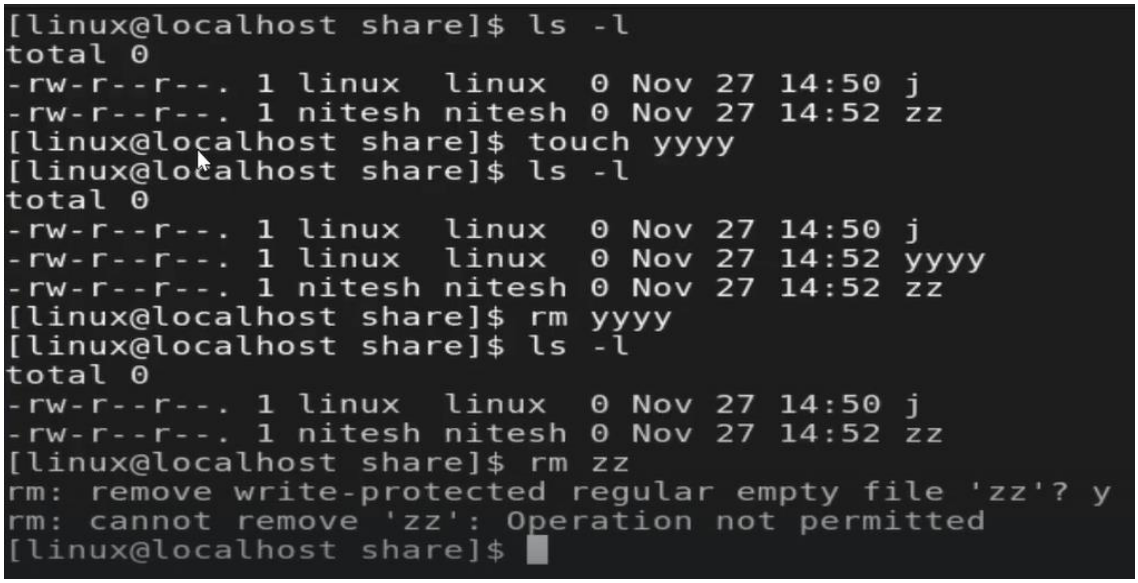
- Login as user “linux”, we see that this user is able to remove the file created by user “nitesh”

```
[root@localhost ~]# su - linux
[linux@localhost ~]$ groups
linux lw
[linux@localhost ~]$ ls -ld /share/
drwxrwxr-x. 2 vimal lw 19 Nov 27 14:49 /share/
[linux@localhost ~]$ cd /share/
[linux@localhost share]$ ls -l
total 0
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 14:49 f.txt
[linux@localhost share]$ touch j
[linux@localhost share]$ ls -l
total 0
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 14:49 f.txt
-rw-r--r--. 1 linux  linux  0 Nov 27 14:50 j
[linux@localhost share]$ rm f.txt
rm: remove write-protected regular empty file 'f.txt'? y
[linux@localhost share]$ ls -l
total 0
-rw-r--r--. 1 linux linux 0 Nov 27 14:50 j
```

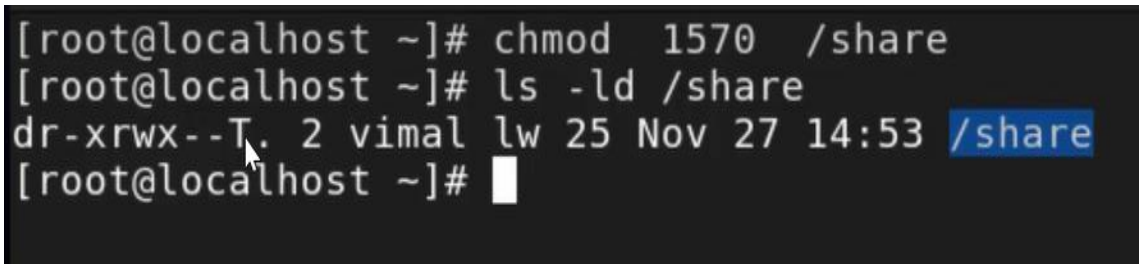
- The command used to set sticky bit on the shared or common folder “/share/” is “**chmod o+t /share/**”, “**t**” represents sticky bit **with execution(x)** power and “**T**” represents sticky bit **without execution(x)**

```
[root@localhost ~]# ls -ld /share/
drwxrwxr-x. 2 vimal lw 15 Nov 27 14:50 /share/
[root@localhost ~]# chmod   o+t  /share/
[root@localhost ~]# ls -ld /share/
drwxrwxr-t. 2 vimal lw 15 Nov 27 14:50 /share/
[root@localhost ~]#
```

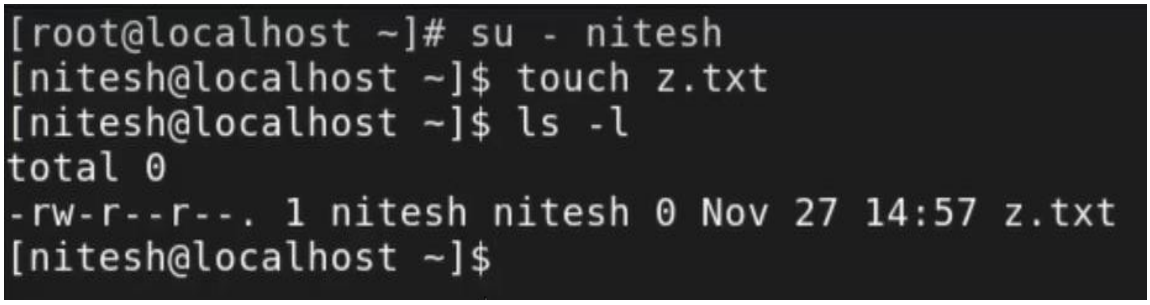
- After the sticky bit is set, now we see that other users are not permitted to delete file of the user, who created it



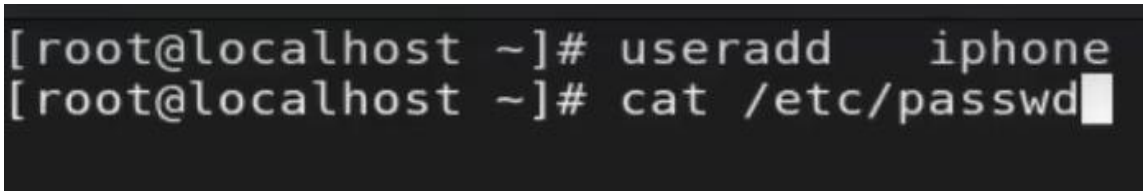
- The command to set **sticky bit**(1), **users** with **read and execute** (4+1=5) permissions, **groups** with **read, write and execute** (4+2+1=7) and **others** with **no permissions** (0) on the shared folder "/share" is "**chmod 1570 /share**"



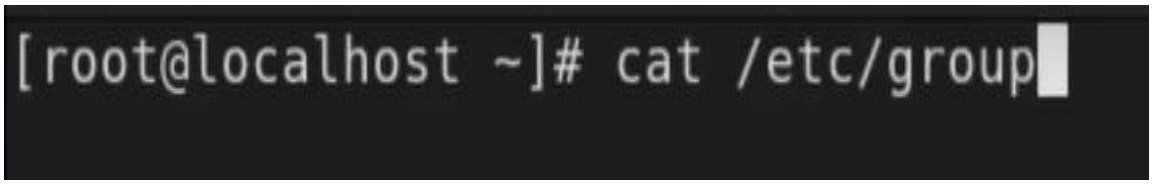
- Login as "nitesh" user, create a file "z.txt", we see that automatically the user and group owner is set to "nitesh", the name is same but one is the user name and the other is the group name.



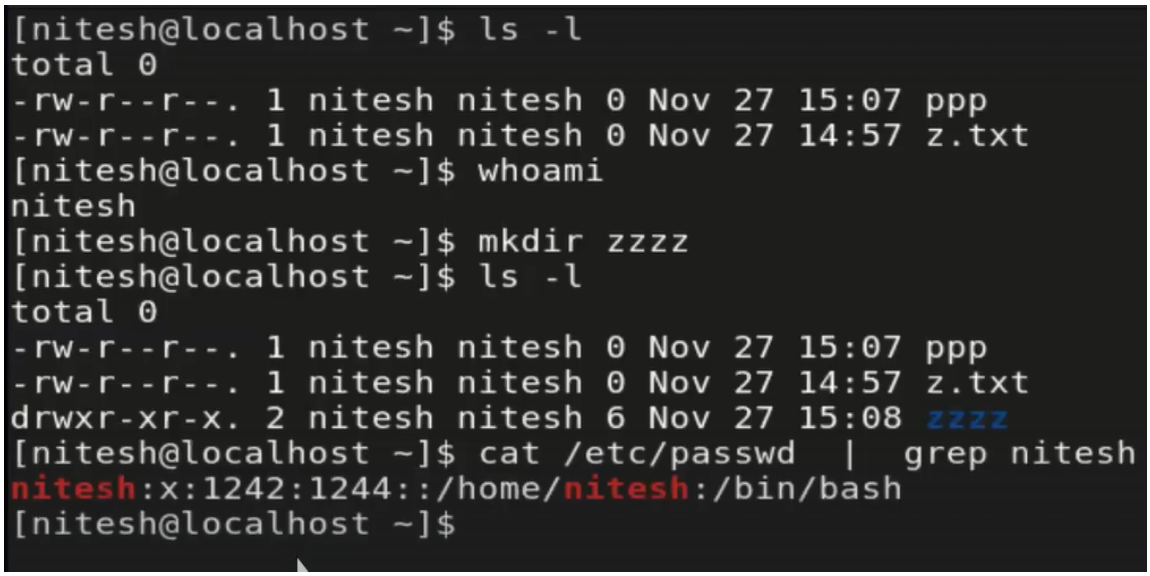
- The “useradd” command, creates users and updates the file “/etc/passwd”, automatically it creates groups and updates the file “/etc/group”. The user name and group name are similar but purpose is different.



- At the same time group is created , this is called as **primary group**



- Login as “nitesh” user, create a directory “zzzz”, we see that automatically the user and group owner is set



- To change the primary group of directory "/zzzz", it can only be done as "root user, but technically it's not possible to always contact root user to do this

```
[nitesh@localhost ~]$ chgrp  lwgroup  zzzz/
chgrp: changing group of 'zzzz/': Operation not permitted
[nitesh@localhost ~]$ ls -l
total 0
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 15:07 ppp
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 14:57 z.txt
drwxr-xr-x. 2 nitesh nitesh 6 Nov 27 15:08 zzzz
[nitesh@localhost ~]$
```

- The **primary group** of user "nitesh" is "**nitesh**",

```
[nitesh@localhost ~]$ id
uid=1242(nitesh) gid=1244(nitesh) groups=1244(nitesh),1243(lw) conte
d_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[nitesh@localhost ~]$ ls -l
total 0
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 15:07 ppp
-rw-r--r--. 1 nitesh nitesh 0 Nov 27 14:57 z.txt
drwxr-xr-x. 2 nitesh nitesh 6 Nov 27 15:08 zzzz
```

- The command to give group password is "**gpasswd lwgroup**" and this is updated in a file **"/etc/gshadow**"

```
[root@localhost ~]# gpasswd lwgroup
Changing the password for group lwgroup
New Password:
Re-enter new password:
[root@localhost ~]# cat /etc/gshadow
```

```
lwgroup:$6$TVFOaS1NsjPKRwem$cCafZ8vcP0e4FbB8kEvDT0J1dek7b4GMjo9TF9x0
t7usf9mYjjg1WmtbqvonEH0d9V4rE/z/n.::yash,sarah,raj
```

- The command to change the primary group to “lwgroup” of “nitesh” user is “**newgrp lwgroup**”

```
[nitesh@localhost ~]$ newgrp  lwgroup
Password:
[nitesh@localhost ~]$
```

```
[nitesh@localhost ~]$ id
uid=1242(nitesh) gid=1242(lwgroup) groups=1242(lwgroup),1243(lw),124
ntext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[nitesh@localhost ~]$
```

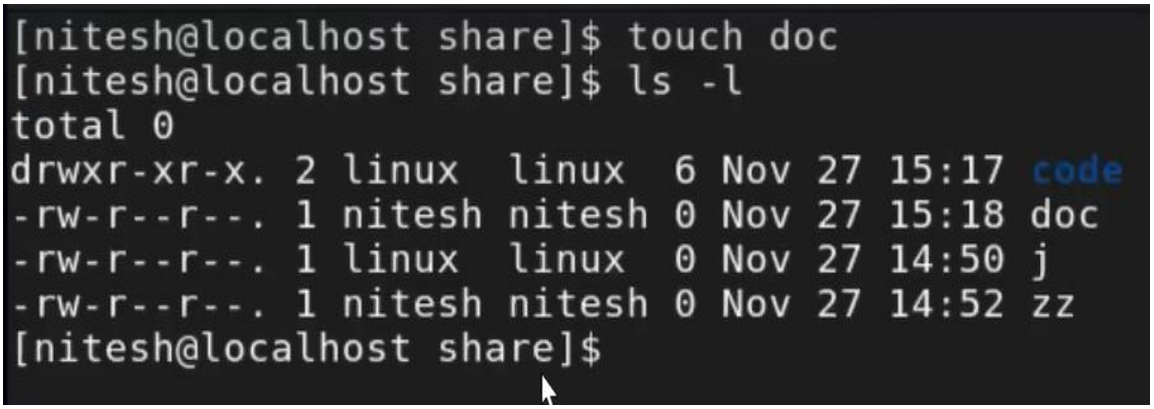
- If user nitesh creates file or directories , we see it is automatically set to the group “lwgroup”

```
[nitesh@localhost ~]$ touch aaaa
[nitesh@localhost ~]$ ls -l
total 0
-rw-r--r--. 1 nitesh lwgroup 0 Nov 27 15:14 aaaa
-rw-r--r--. 1 nitesh nitesh  0 Nov 27 15:07 ppp
drwxr-xr-x. 2 nitesh nitesh  6 Nov 27 15:13 z
-rw-r--r--. 1 nitesh nitesh  0 Nov 27 14:57 z.txt
drwxr-xr-x. 2 nitesh nitesh  6 Nov 27 15:08 zzzz
[nitesh@localhost ~]$ mkdir data
[nitesh@localhost ~]$ ls -l
total 0
-rw-r--r--. 1 nitesh lwgroup 0 Nov 27 15:14 aaaa
drwxr-xr-x. 2 nitesh lwgroup 6 Nov 27 15:15 data
-rw-r--r--. 1 nitesh nitesh  0 Nov 27 15:07 ppp
drwxr-xr-x. 2 nitesh nitesh  6 Nov 27 15:13 z
-rw-r--r--. 1 nitesh nitesh  0 Nov 27 14:57 z.txt
drwxr-xr-x. 2 nitesh nitesh  6 Nov 27 15:08 zzzz
[nitesh@localhost ~]$
```

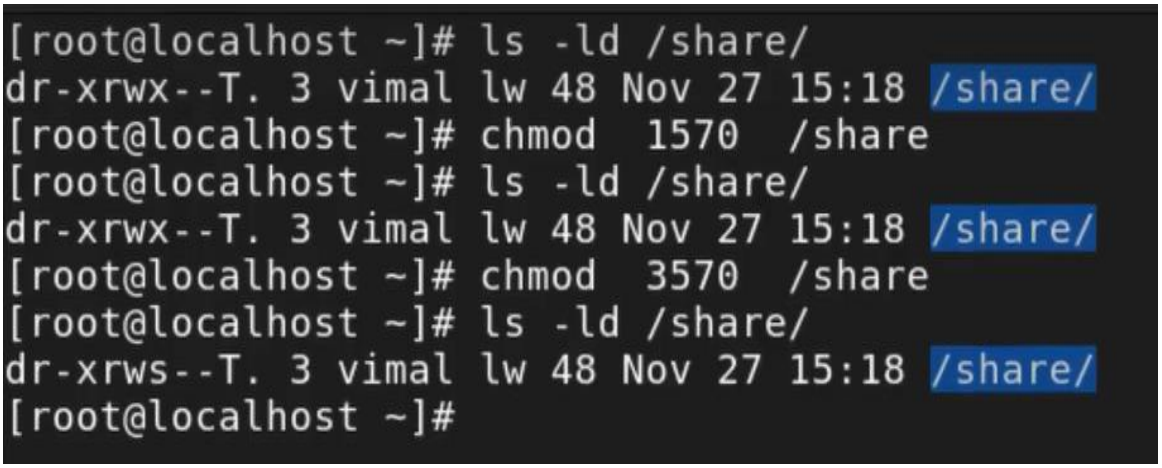
- The user “nitesh” has the power to set permissions on the folder “data/” using the command “**chmod g+rwx data/**”

```
[nitesh@localhost ~]$ chmod g+rwx data/
[nitesh@localhost ~]$ ls -l
total 0
-rw-r--r--. 1 nitesh lwgroup 0 Nov 27 15:14 aaaa
drwxrwxr-x. 2 nitesh lwgroup 6 Nov 27 15:15 data
-rw-r--r--. 1 nitesh nitesh  0 Nov 27 15:07 ppp
drwxr-xr-x. 2 nitesh nitesh  6 Nov 27 15:13 z
-rw-r--r--. 1 nitesh nitesh  0 Nov 27 14:57 z.txt
drwxr-xr-x. 2 nitesh nitesh  6 Nov 27 15:08 zzzz
```

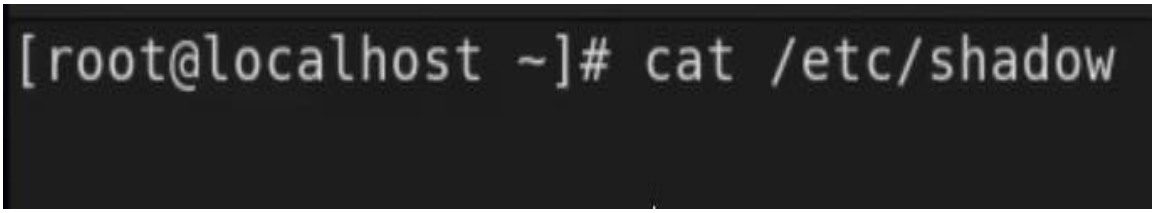
- In a sharable folder, if the user creates a file, automatically the UID and GID of the file is set to the user only, this is by default but it has to be shared by all users in group



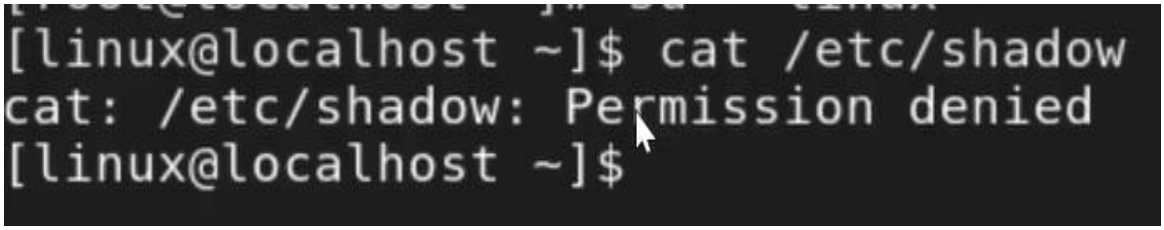
- To set SGID(2) and sticky bit(1) on the shared folder "/share"



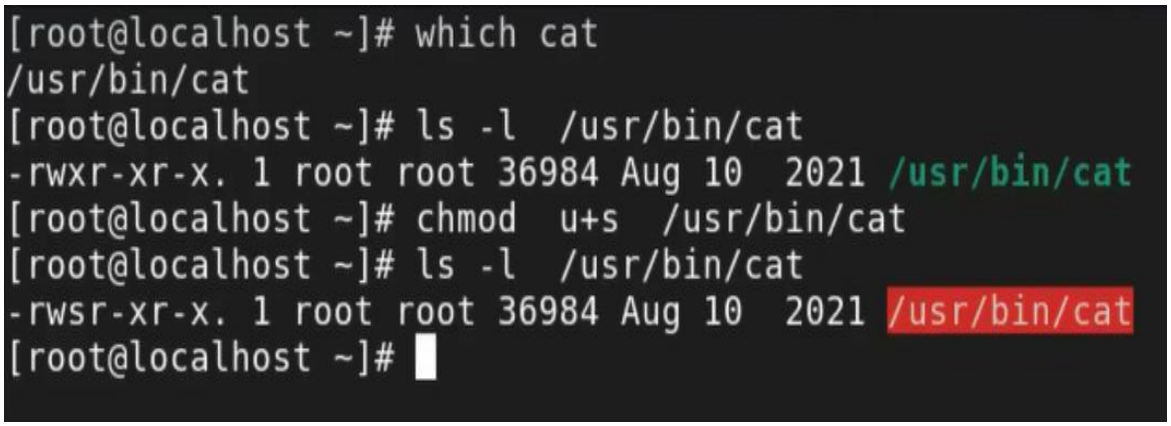
- The SUID is very dangerous, as setting SUID in wrong place may compromise your system, it gets the power of root without the password of root
- Whenever we run a command or program, it's by the power of the user, the "**cat**" command is run by power of "**root**" user, in linux the root has unlimited power so the "cat" command also gets the unlimited power



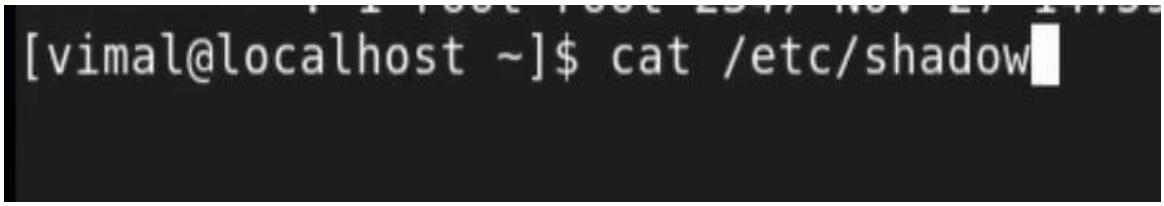
- The same “cat” command run by the user “linux”, the permission is denied to read the file “/etc/shadow”

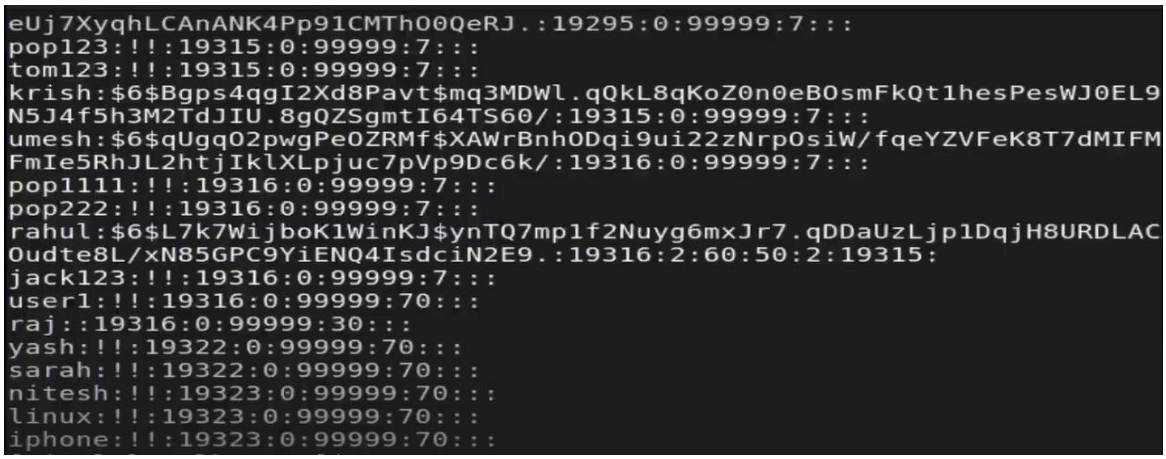


- To set SUID on the executable file **“/usr/bin/cat**” of cat command, the command is “**chmod u+s /usr/bin/cat**”



- Since SUID is set, the user is able to read the file, behind the scene its run by the power of root (**the power of the owner of the file)**





- The SUID is set to the executable file of "passwd" command, so the users are able to change password using "passwd" command
- The user has no power to do anything on the file **"/etc/shadow**" and without going into the file, the user can change the password using "**passwd**" command, this is possible because SUID is set on the executable file **"/usr/bin/passwd**" of "**passwd**" command.