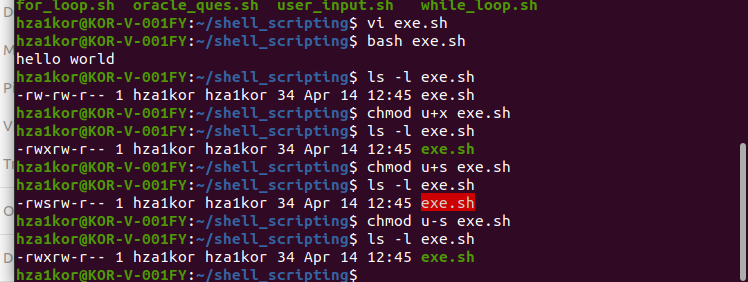
**What is SUID and SGID in Linux? | MPrashant**

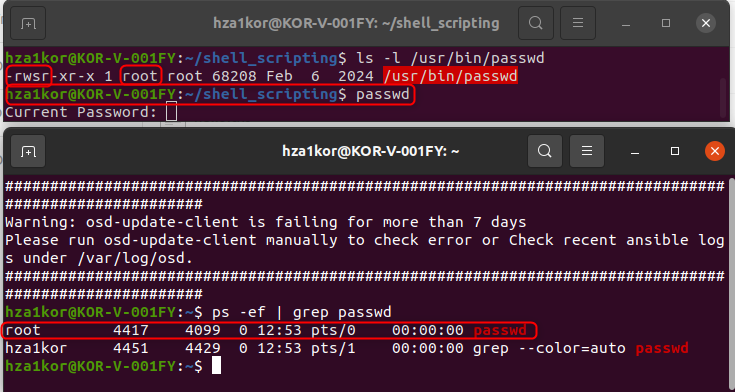
* SUID is a special permission for the user
* SUID means Set user ID
* A file with SUID set is always executed as the user who owns the file, regardless of the who is executing it, meaning if there is an executable with SUID set, no matter who is executing this executable, it will execute with details of the owner of the executable
* Say hza1kor is the owner of exe.sh then even if sapna or sam or tani is executing it, it will execute with details of hza1kor only
* SUID is shown as ‘s’ or ‘S’
* In the below image we can see that the permissions are shown as rws instead of rwx



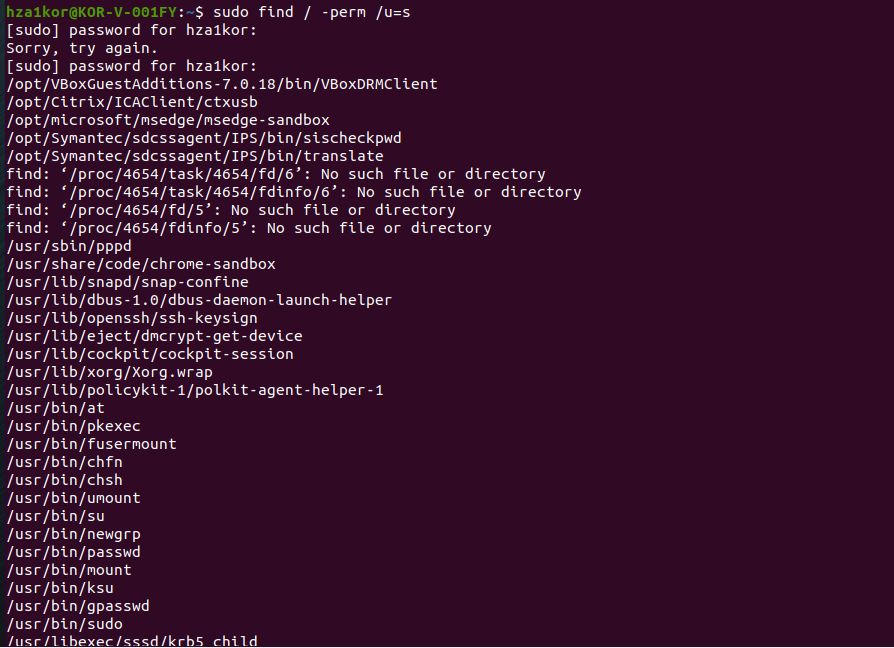
* If we see s instead of x means SUID is set here
* To set SUID – chmod u+s <filename>
* To unset SUID – chmod u-s <filename>
* If we see the below image carefully, first I created a simple script called exe.sh that will just echo hello world
* Then I executed it same response
* Then I say the permissions of exe.sh, I see no execute for user
* So I added u+x for exe.sh
* Then again I do ls -l and I see exe.sh in green
* Then I do u+s to add the SUID, now when I run ls -l, I can see the user permissions as rws and not rwx
* Also the exe.sh is somehow getting highlighted in red
* Then I tried removing SUID and again tried ls -l, now the permissions for user/owner are rwx
* So I guess it can either be rwx or rws



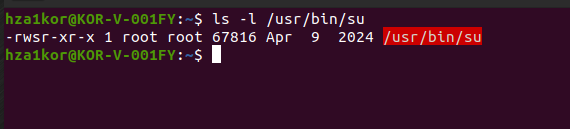
* For another example, we have another file called usr/bin/passwd where SUID is set
* Now from user hza1kor I am running the passwd command, meaning hza1kor has started the process
* But when I check ps -ef | grep passwd, I can see that root has started this process because SUID is set, meaning even if hza1kor is running this executable, process will still be initialized on owner(root) name



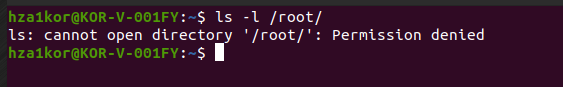
* Command to print all the files that have user permissions as s – sudo find / -perm /u=s



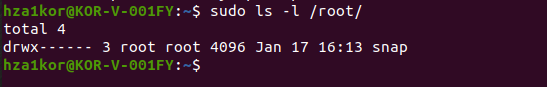
* Even su has SUID set so when I run su -, or su – hza1kor then it shows as root is executing these commands



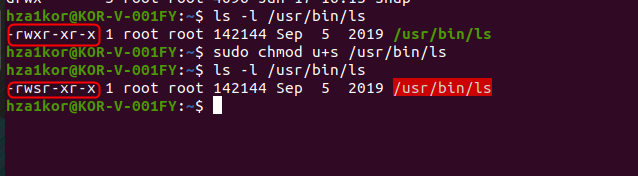
* **Lets take another example now**
* When I run ls -l /root without sudo I will get permission denied because root folder can only be accessed by root no one else



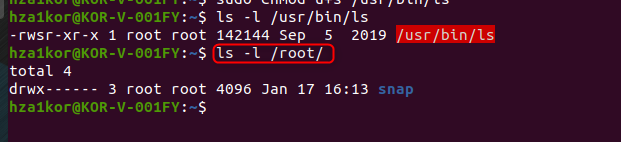
* The below image has the permissions of root folder, basically root is the owner and also the group owner, so we have rwx only for owner, no group and others have rwx permissions



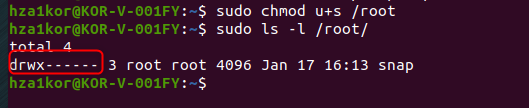
* When I run the same command with sudo, the command is basically executed as root user not the existing user. Therefore, sudo ls -l /root/ works because the process has root’s permissions, bypassing the restrictions you have as a non-root user.
* Running sudo ls -l /root executes ls with root privileges, so it succeeds because the process runs as root, and root has permission to access the /root directory.
* Now I want to execute ls -l /root without sudo or root access, I want that every user can list root folder content
* For that I change the ls permissions of user to chmod u+s /usr/bin/ls



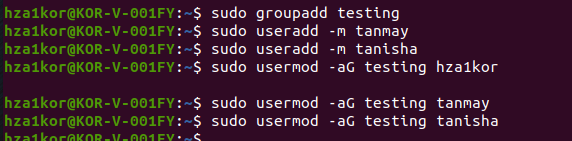
* Now even if a normal user is executing the ls command, it will show as if root or owner of the file is executing the command
* Since root folder can only be accessed by root, when is run ls -l /root after setting SUID for ls, the program is executed as root instead of your user credentials



* More detailed explanation below
* Now when I set SUID for ls, it means that whenever ls command is run, the process’s effective UID(EUID) is set to the owner of the fil, regardless of who is executing it
* Since /usr/bin/ls is owned by root, the sudo chmod u+s /usr/bin/ls command changes the file’s mode, so that whenever any user runs ls, the OS treats the process as if its run by the owner(in this case root)
* **What is SGID –**
* If set on a file, it allows the file to be executed as the group owner that owns the file
* It set on a directory, any files created in the directory will have their group ownership set to that of the directory owner
* Not that SUID could be set only for files not for directories, when I was trying to set SUID for root folder there was no o/p

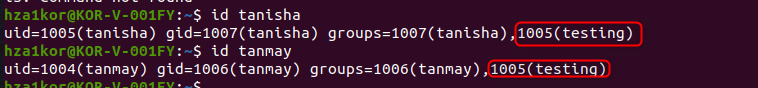


* SGID is especially useful for directories that are often used in collaborative efforts between members of a group
* Any member of the group can access any new file
* To set GUID – sudo chmod g+s <filename>
* To unset GUID – sudo chmod g-s <filename>
* Example –
* Say I want to create a group named testing and add 3 users hza1kor, tanmay, tanisha to it
* So first I will create a group using command – sudo groupadd testing
* To remove a group – sudo groupdel <group\_name>
* Now I add 2 users using – sudo useradd -m tanmay and sudo useradd -m tanisha
* To delete a user – sudo userdel <username>
* Now I have a group named testing, and 2 users are also added
* Now to add hza1kor, tanmay and tanisha to this testing group I will run commands
  + sudo usermod -aG testing hza1kor
  + sudo usermod -aG testing tanmay
  + sudo usermod -aG testing tanisha

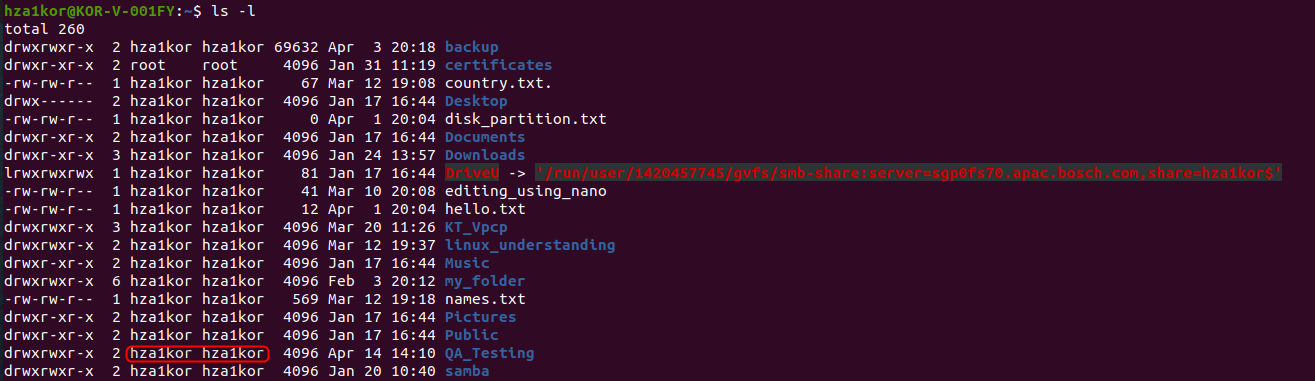


* next we can check if these users are added to these groups using id <username>

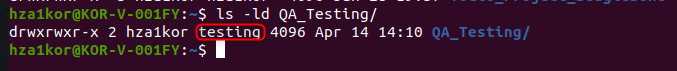




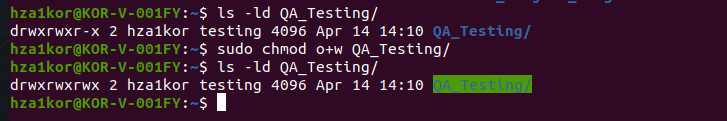
* then I create a folder called QA\_Testing, and I can see the owner and the group owner is currently hza1kor only



* first we have to change the group owner to testing using chgrp command – sudo chgrp testing QA\_Testing, To see permissions of a folder use ls -ld foldername



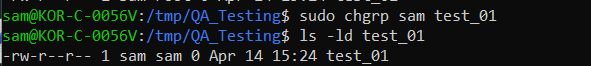
* now lets give all permissions to QA\_Testing group so that everyone can access it, since all permissions are present, except write for others



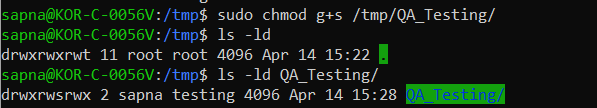
* now testing group has hza1kor, tanmay and tanisha present, so these 3 people can access the QA\_Testing folder
* now say I create another user named sampi using sudo useradd -m sampi, now this user is not in the testing team
* now sampi will create a file called test\_01 in QA\_Testing folder(make sure don’t create QA\_Testing in home directory of any users, you wont be able to access it, say I create QA\_Testing in /home/sapna, then sam or tanmay wouldn’t be able to access it)



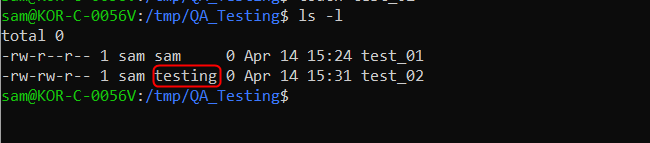
* now when sapna, tanmay tanisha see this file there are like what who the hell is this guy and they try to edit this file, but they are not able to edit it because the owner is sampi and the group owner is also sampi



* so now sapna decides to set group ID (SGID) for the folder QA\_Testing using command – sudo chmod g+s /tmp/QA\_Testing



* Now next time when sam tries to create another file called test\_02, the owner is sam, but the group owner will be set to Testing since hza1kor who created this folder QA\_Testing had set the group owner as Testing



* Woww impressive right
* SUID is used to set owner so that anyone executes it it looks like the owner is executing the file
* SGID is used for directories so that the group owner of all the files created will be same as the group owner of the directory