# Lab 7

1. **Using the useradd command, add accounts for the following users in your system: user1, user2, user3, user4, user5, user6 and user7. Remember to give each user a password**

```
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user1
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user2
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user3
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user4
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user5
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user6
gharabawy@gharabawy-virtual-machine:~$ sudo useradd user7
```

➔ **Set passwd :**

```
gharabawy@gharabawy-virtual-machine:~$ sudo passwd user1
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
gharabawy@gharabawy-virtual-machine:~$ sudo passwd user2
```

2. **Using the groupadd command, add the following groups to your system.**

| Group | GID |
|---|---|
| sales | 10000 |
| hr | 10001 |
| web | 10002 |

```
gharabawy@gharabawy-virtual-machine:~$ sudo groupadd -g 10000 sales
gharabawy@gharabawy-virtual-machine:~$ sudo groupadd -g 10001 hr
gharabawy@gharabawy-virtual-machine:~$ sudo groupadd -g 10002 web
```

**Why should you set GID in this manner instead of allowing the system to set the GID by default?**

➔ **Because it is hard for the system to do this operation while the system gets the last id and increments the id of new group to it**

3. **Using the usermod command to add user1 and user2 to the sales secondary group, user3 and user4 to the hr secondary group. User5 and user6 to web secondary group. And add user7 to all secondary groups**

```
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG sales user1
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG sales user2
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG hr user3
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG hr user4
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG web user5
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG web user6
gharabawy@gharabawy-virtual-machine:~$ sudo usermod -aG sales,hr,web user7
```

➔ **Test :**

```
gharabawy@gharabawy-virtual-machine:~$ cat /etc/group | grep sales
sales:x:10000:user7,user1,user2
gharabawy@gharabawy-virtual-machine:~$ cat /etc/group | grep hr
hr:x:10001:user7,user3,user4
gharabawy@gharabawy-virtual-machine:~$ cat /etc/group | grep web
web:x:10002:user7,user5,user6
```

4. **Login as each user and use id command to verify that they are in the appropriate groups. How else might you verify this information?**

```
gharabawy@gharabawy-virtual-machine:~$ id user1
uid=1004(user1) gid=1004(user1) groups=1004(user1),10000(sales)
gharabawy@gharabawy-virtual-machine:~$ id user2
uid=1005(user2) gid=30001(user2) groups=30001(user2),10000(sales)
gharabawy@gharabawy-virtual-machine:~$ id user3
uid=1006(user3) gid=1006(user3) groups=1006(user3),10001(hr)
gharabawy@gharabawy-virtual-machine:~$ id user4
uid=1007(user4) gid=1007(user4) groups=1007(user4),10001(hr)
gharabawy@gharabawy-virtual-machine:~$ id user5
uid=1008(user5) gid=1008(user5) groups=1008(user5),10002(web)
gharabawy@gharabawy-virtual-machine:~$ id user6
uid=1009(user6) gid=1009(user6) groups=1009(user6),10002(web)
gharabawy@gharabawy-virtual-machine:~$ id user7
uid=1010(user7) gid=1010(user7) groups=1010(user7),10000(sales),10001(hr),10002(web)
```

5. **Create a directory called /depts with a sales, hr, and web directory within the /depts directory.**

```
gharabawy@gharabawy-virtual-machine:~$ sudo mkdir -p /depts/sales
gharabawy@gharabawy-virtual-machine:~$ sudo mkdir -p /depts/hr
gharabawy@gharabawy-virtual-machine:~$ sudo mkdir -p /depts/web
```

6. **Using the chgrp command, set the group ownership of each directory to the group with the matching name**

```
gharabawy@gharabawy-virtual-machine:~$ sudo chgrp sales /depts/sales
gharabawy@gharabawy-virtual-machine:~$ sudo chgrp hr /depts/hr
gharabawy@gharabawy-virtual-machine:~$ sudo chgrp web /depts/web
```

7. **Set the permissions on the /depts directory to 755, and each subdirectory to 770**

```
gharabawy@gharabawy-virtual-machine:~$ sudo chmod 755 /depts
gharabawy@gharabawy-virtual-machine:~$ sudo chmod 770 /depts/sales /depts/hr /depts/web
gharabawy@gharabawy-virtual-machine:~$ ls -ld /depts
drwxr-xr-x 5 root root 4096 Dec 15 17:01 /depts
gharabawy@gharabawy-virtual-machine:~$ ls -ld /depts/hr
drwxrwx--- 2 root hr 4096 Dec 15 17:01 /depts/hr
```

8. **Set the set-gid bit on each departmental directory**

```
gharabawy@gharabawy-virtual-machine:~$ sudo chmod g+s /depts/sales /depts/hr /depts/web
gharabawy@gharabawy-virtual-machine:~$ ls -ld /depts/*
drwxrws--- 2 root hr    4096 Dec 15 17:01 /depts/hr
drwxrws--- 2 root sales 4096 Dec 15 17:01 /depts/sales
drwxrws--- 2 root web   4096 Dec 15 17:01 /depts/web
```

9. **Use the su command to switch to the user2 account and attempt the following commands:**
   **touch /depts/sales/user2.txt**
   **touch /depts/hr/ user2.txt**
   **touch /depts/web/ user2.txt**

```
gharabawy@gharabawy-virtual-machine:~$ su user2
Password:
user2@gharabawy-virtual-machine:/home/gharabawy$ touch /depts/sales/user2.txt
user2@gharabawy-virtual-machine:/home/gharabawy$ touch /depts/hr/user2.txt
touch: cannot touch '/depts/hr/user2.txt': Permission denied
user2@gharabawy-virtual-machine:/home/gharabawy$ touch /depts/web/user2.txt
touch: cannot touch '/depts/web/user2.txt': Permission denied
```

   **Which of these commands succeeded and which failed? What is the group ownership of the files that were created?**

➔ **User2 can touch only first file because user2 is in the group that owns the sales directory.**

**10. Configure sudoers file to allow user3 and user4 to use /bin/mount and /bin/umount commands, while allowing user5 only to use fdisk command.**

➔ Open suduoers file using sudo visudo command and add this lines :

```
root    ALL=(ALL:ALL) ALL
user3 ALL=(ALL) /bin/mount, /bin/umount
user4 ALL=(ALL) /bin/mount, /bin/umount
user5 ALL=(ALL) /sbin/fdisk
```

**11. Login by user3 and try to unmount /boot.**

```
gharabawy@gharabawy-virtual-machine:~$ su user3
Password:
user3@gharabawy-virtual-machine:/home/gharabawy$ sudo umount /boot
[sudo] password for user3:
umount: /boot: not mounted.
```

**12. Login by user4 and remount /boot. Also try to view the partition table using fdisk.**

```
gharabawy@gharabawy-virtual-machine:~$ su user4
Password:
user4@gharabawy-virtual-machine:/home/gharabawy$ sudo fdisk -l
Disk /dev/loop0: 55.66 MiB, 58363904 bytes, 113992 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/loop1: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

**13. Create a directory with permissions rwxrwx---, grant a second group (sales) r-x permissions**

```
gharabawy@gharabawy-virtual-machine:~$ sudo setfacl -m g:sales:rx /example
gharabawy@gharabawy-virtual-machine:~$ getfacl /example
getfacl: Removing leading '/' from absolute path names
# file: example
# owner: root
# group: sales
# flags: -s-
user::rwx
group::rwx
group:sales:r-x
mask::rwx
other::---
```

**14. create a file on that directory and grant read and write to a second group (sales)**

```
gharabawy@gharabawy-virtual-machine:~$ sudo setfacl -m g:sales:rw /example/file.txt
gharabawy@gharabawy-virtual-machine:~$ sudo getfacl /example/file.txt
getfacl: Removing leading '/' from absolute path names
# file: example/file.txt
# owner: root
# group: sales
user::rw-
group::rw-
group:sales:rw-
mask::rw-
other::r--
```

➔ **Test :**

```
gharabawy@gharabawy-virtual-machine:~$ sudo ls -l /example/
total 0
-rw-rw-r--+ 1 root sales 0 Dec 15 18:05 file.txt
```

**15. set the the owning group as the owning group of any newly created file in that directory.**

```
gharabawy@gharabawy-virtual-machine:~$ sudo chmod g+s /example
gharabawy@gharabawy-virtual-machine:~$ sudo ls -ld /example
drwxrws--- 2 root sales 4096 Dec 15 18:05 /example
gharabawy@gharabawy-virtual-machine:~$
```

```
gharabawy@gharabawy-virtual-machine:~$ sudo touch /example/file2.txt
gharabawy@gharabawy-virtual-machine:~$ sudo ls -l /example/file2.txt
-rw-r--r-- 1 root sales 0 Dec 15 18:21 /example/file2.txt
```

**16. Grand your colleagues a collective directory called /opt/research, where they can store generated research results. Only members of group profs and grads should be able to create new files in the directory, and new file should have the following properties:**

- **the directory should be owned by root**
- **new files should be group owned by group grads**
- **group profs should automatically have read/write access to new files**
- **group interns should automatically have read only access to new files**
- **other users should not be able to access the directory and its contents at all.**

```
gharabawy@gharabawy-virtual-machine:~$ sudo mkdir /opt/research
gharabawy@gharabawy-virtual-machine:~$ sudo groupadd grads
gharabawy@gharabawy-virtual-machine:~$ sudo chgrp grads /opt/research
gharabawy@gharabawy-virtual-machine:~$ sudo chmod g=rwxs /opt/research
gharabawy@gharabawy-virtual-machine:~$ sudo groupadd profs
gharabawy@gharabawy-virtual-machine:~$ sudo setfacl -m d:g:profs:rw /opt/research
gharabawy@gharabawy-virtual-machine:~$ sudo groupadd interns
gharabawy@gharabawy-virtual-machine:~$ sudo setfacl -m d:g:interns:r /opt/research
gharabawy@gharabawy-virtual-machine:~$ sudo chmod o= /opt/research
gharabawy@gharabawy-virtual-machine:~$ sudo getfacl /opt/research/
getfacl: Removing leading '/' from absolute path names
# file: opt/research/
# owner: root
# group: grads
# flags: -s-
user::rwx
group::rwx
other::---
default:user::rwx
default:group::rwx
default:group:profs:rw-
default:group:interns:r--
default:mask::rwx
default:other::r-x
```

**17. Change your default SELinux mode to permissive and reboot.**

**First : open sudo nano /etc/selinux/config and change SELINUX to permissive**

**Then reboot**

```
SELINUX=permissive



SELINUXTYPE=default


SETLOCALDEFS=0
```

**18. After reboot, verify the system is in permissive mode.**

```
gharabawy@gharabawy-virtual-machine:~$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                   permissive
Mode from config file:          permissive
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     requested (insecure)
Max kernel policy version:      33
```

**19. Change the default SELinux mode to enforcing.**
**First : open sudo nano /etc/selinux/config and change SELINUX enforcing**
**Then reboot**

```
SELINUX=enforcing



SELINUXTYPE=default


SETLOCALDEFS=0
```

```
liveuser@localhost-live:~$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

**20. Change the current SELinux mode to enforcing.**

```
liveuser@localhost-live:~$ sudo setenforce enforcing
liveuser@localhost-live:~$ getenforce
Enforcing
```

**21. Copy /etc/resolv.conf file to root's home directory.**

```
liveuser@localhost-live:~$ sudo cp /etc/resolv.conf /root
liveuser@localhost-live:~$ sudo ls -l /root
total 8
-rw-------. 1 root root 3020 Oct 31 21:15 anaconda-ks.cfg
-rw-r--r--. 1 root root  930 Dec 15 17:01 resolv.conf
```

**22. Observe the SELinux context of the intial /etc/resolv.conf**

```
liveuser@localhost-live:~$ sudo ls -lZ /etc/resolv.conf
lrwxrwxrwx. 1 root root system_u:object_r:net_conf_t:s0 39 Oct 31 21:09 /etc/resolv.conf -> ../run/systemd/resolve/stub-resolv.conf
```

**23. Move resolv.conf from root's home directory to /etc/resolv.conf**

```
liveuser@localhost-live:~$ sudo mv /root/resolv.conf /etc/resolv.conf
```

**24. Observe the SELinux of the newly copied /etc/resolv.conf**

```
liveuser@localhost-live:~$ sudo ls -lZ /etc/resolv.conf
-rw-r--r--. 1 root root unconfined_u:object_r:admin_home_t:s0 930 Dec 15 17:01 /etc/resolv.conf
```

**25. Restore the SELinux context of the newly positioned /etc/resolv.conf**

```
liveuser@localhost-live:~$ sudo restorecon /etc/resolv.conf
```

**26. Observe the SELinux context of the restored /etc/resolv.conf**

```
liveuser@localhost-live:~$ ls -lZ /etc/resolv.conf
-rw-r--r--. 1 root root unconfined_u:object_r:net_conf_t:s0 930 Dec 15 17:01 /etc/resolv.conf
```

**27. Configure OpenSSH to allow pulic key-based login credentials**

➔ **First open /etc/ssh/sshd_config**

➔ **Second type :**

```
# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys
```

## 28. Create an SSH key-pair

```
liveuser@localhost-live:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/liveuser/.ssh/id_rsa):
Created directory '/home/liveuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liveuser/.ssh/id_rsa
Your public key has been saved in /home/liveuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AfilMW0L8mqhE0k4kEhazpFjvTeae3brHIxaGMCU2XA liveuser@localhost-live
The key's randomart image is:
+---[RSA 3072]----+
|=+=BE...         |
|**Bo= +.+        |
|.+=o = B..       |
|  o.o * ..       |
|   o.* .S        |
|  o =o o         |
|   o..o o        |
|    .oo...       |
|    .o o+.       |
+----[SHA256]-----+
```

## 29. Configure to login without the need of a password.

```
liveuser@localhost-live:~$ ssh-copy-id liveuser@localhost-live
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.
              (if you think this is a mistake, you may want to use -f option)
```

## 30. Configure SSH to prevent root logins.

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

## 31. Configure logrotate default setting to compress log files when they are rotated.

```
# uncomment this if you want your log files compressed
compress
```