Lab 3 Linux

GIK2NV - Data Storage and Management Technologies

13th October 2022

Mouayad Mouayad h20moumo
Kevin Vlajkovic h20kevvl
Mohamed Almefrej h20mohaa

LINUX LABORATORY: Kevin, Mouayad och Mohamad.

Part 1.

Case Scenario

As the Linux Administrator for fast-growing company, you have been tasked with creating, modifying, and removing user accounts from the Linux server. The company has just hired 9 new employees to fill 3 newly designed departments. The departments that have been created are Engineering, Sales and HR. The server must be setup with the appropriate files, folders, users, groups and permissions to ensure a successful launch of the newly designed departments.

Objectives

1. ~~Create a directory at the root (/) of the file system for each department. This name should reflect the department name that will use the directory.~~
   cd /
   sudo mkdir Engineering Sales HR

```
k-user@LAPTOP-2KJQ3HMA:/home$ cd /
k-user@LAPTOP-2KJQ3HMA:/$ ls
bin   dev  home  lib    lib64   lost+found  mnt  proc  run           sbin  srv  tmp    user2  var
boot  etc  init  lib32  libx32  media       opt  root  sampletest.sh  snap  sys  user1  usr
k-user@LAPTOP-2KJQ3HMA:/$ sudo mkdir Engineering Sales HR
[sudo] password for k-user:
k-user@LAPTOP-2KJQ3HMA:/$ ls
Engineering  Sales  boot  etc   init  lib32  libx32      media  opt   root  sampletest.sh  snap  sys  user1  usr
HR           bin    dev   home  lib   lib64  lost+found  mnt    proc  run   sbin           srv   tmp  user2  var
k-user@LAPTOP-2KJQ3HMA:/$
```

   - “sudo” means admin privilege.
   - “mkdir” means make directory.
   - “Engineering Sales HR” are just names for the 3 folders that will be made.

2. ~~Create a group for each department. This name should reflect the department name that the group will be assigned.~~
   sudo groupadd engi-team
   sudo groupadd sales-team
   sudo groupadd hr-team
   getent group

```
k-user@LAPTOP-2KJQ3HMA:/$ sudo groupadd engi-team
k-user@LAPTOP-2KJQ3HMA:/$ sudo groupadd sales-team
k-user@LAPTOP-2KJQ3HMA:/$ sudo groupadd hr-team
k-user@LAPTOP-2KJQ3HMA:/$ getent group
root:x:0:
daemon:x:1:
```

```
engi-team:x:1004:
sales-team:x:1005:
hr-team:x:1006:
k-user@LAPTOP-2KJQ3HMA:/$
```

- “groupadd” means that we will create a group.
- “engi-team” is just the group name.
- “getent group” getent command means we want to search for something in the system and then we type “group” means we are searching for every group in the system that will later show up below.

3. ~~Create an administrative user for each of the departments.~~
   a. ~~The user will have a Bash login shell~~
   b. ~~The user will belong to the respective group for each department. This will need to be the user’s primary group.~~

   sudo useradd -s /bin/bash -m E-admin
   sudo useradd -s /bin/bash -m S-admin
   sudo useradd -s /bin/bash -m HR-admin
   sudo usermod -g engi-team E-admin
   sudo usermod -g sales-team S-admin
   sudo usermod -g hr-team HR-admin
   groups E-admin <-- See primary group

```
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m E-admin
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m S-admin
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m HR-admin
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g engi-team E-admin
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g sales-team S-admin
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g hr-team HR-admin
k-user@LAPTOP-2KJQ3HMA:/$ groups E-admin
E-admin : engi-team
k-user@LAPTOP-2KJQ3HMA:/$ groups S-admin
S-admin : sales-team
k-user@LAPTOP-2KJQ3HMA:/$ groups HR-admin
HR-admin : hr-team
k-user@LAPTOP-2KJQ3HMA:/$
```

Explaining command useradd!

- “useradd” means we are creating a user.
- “-m” means create a directory for the user.
- “-s /bin/bash” means that we are creating a shell for the user.

Explaining command usermod!

- “usermod” means that we are changing properties for a user.
- “-g” means group then we specify “engi-team”.
- “groups” means we are searching for a specific group.

4. ~~Create two additional users for each department.~~
    a. ~~The users will have a Bash login shell.~~
    b. ~~The users will belong to their respective group for each department. This will need to be the user's primary group.~~

sudo useradd -s /bin/bash -m E1-user
sudo useradd -s /bin/bash -m E2-user
sudo usermod -g engi-team E1-user
sudo usermod -g engi-team E2-user

sudo useradd -s /bin/bash -m S1-user
sudo useradd -s /bin/bash -m S2-user
sudo usermod -g sales-team S1-user
sudo usermod -g sales-team S2-user

sudo useradd -s /bin/bash -m HR1-user
sudo useradd -s /bin/bash -m HR2-user
sudo usermod -g hr-team HR1-user
sudo usermod -g hr-team HR2-user

groups S1-user
groups HR1-user <- see primary group

```
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m E1-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m E2-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g engi-team E1-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g engi-team E2-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m S1-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m S2-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g sales-team S1-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g sales-team S2-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m HR1-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo useradd -s /bin/bash -m HR2-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g hr-team HR1-user
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g hr-team HR1-user
usermod: no changes
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g hr-team HR1-user
usermod: no changes
k-user@LAPTOP-2KJQ3HMA:/$ sudo usermod -g hr-team HR2-user
k-user@LAPTOP-2KJQ3HMA:/$ groups S1-user
S1-user : sales-team
k-user@LAPTOP-2KJQ3HMA:/$ groups HR1-user
HR1-user : hr-team
k-user@LAPTOP-2KJQ3HMA:/$ groups E1-user
E1-user : engi-team
k-user@LAPTOP-2KJQ3HMA:/$
```

5. ~~For security reasons, the following modifications will need to be made to each of the departments' respective directories:~~
    a. ~~Ensure that the owner of each of the directories is the department administrator and the group ownership is the group for each department.~~
    b. ~~The department administrator will have full access to their respective department directories.~~
    c. ~~Ensure that only the owner of a file in the department's directory can delete the file. The user will also have ownership of their respective department folders.~~
    d. ~~Normal users in each department will have full access (Read, Write and Execute) to their respective department folders.~~
    e. ~~The department folders will ONLY be accessible by users/administrators in each of the respective departments. Ensure that no one else will have permissions to the folders.~~

    sudo chown E-admin Engineering <--- file ownership
    sudo chown S-admin Sales
    sudo chown HR-admin HR

    sudo chgrp engi-team /Engineering <-- Group ownership of directory
    sudo chgrp sales-team /Sales
    sudo chgrp hr-team /HR
    ls -l <- Shows drwx = directory, read, write and execute.

```
k-user@LAPTOP-2KJQ3HMA:/$ sudo chown E-admin Engineering
[sudo] password for k-user:
k-user@LAPTOP-2KJQ3HMA:/$ sudo chown S-admin Sales
k-user@LAPTOP-2KJQ3HMA:/$ sudo chown HR-admin HR
k-user@LAPTOP-2KJQ3HMA:/$ sudo chgrp engi-team /Engineering
k-user@LAPTOP-2KJQ3HMA:/$ sudo chgrp sales-team /Sales
k-user@LAPTOP-2KJQ3HMA:/$ sudo chgrp hr-team /HR
k-user@LAPTOP-2KJQ3HMA:/$
```

```
drwxr-xr-x   2 E-admin  engi-team    4096 Oct 12 17:50 Engineering
drwxr-xr-x   2 HR-admin hr-team      4096 Oct 12 17:50 HR
drwxr-xr-x   2 S-admin  sales-team   4096 Oct 12 17:50 Sales
```

    sudo chmod g+rwx Engineering <- Give users in group rwx
    sudo chmod g+rwx Sales
    sudo chmod g+rwx HR
    ls -l

```
k-user@LAPTOP-2KJQ3HMA:/$ sudo chmod g+rwx Engineering
k-user@LAPTOP-2KJQ3HMA:/$ sudo chmod g+rwx Sales
k-user@LAPTOP-2KJQ3HMA:/$ sudo chmod g+rwx HR
k-user@LAPTOP-2KJQ3HMA:/$ ls -l
total 708
drwxrwxr-x   2 E-admin  engi-team    4096 Oct 12 17:50 Engineering
drwxrwxr-x   2 HR-admin hr-team      4096 Oct 12 17:50 HR
drwxrwxr-x   2 S-admin  sales-team   4096 Oct 12 17:50 Sales
```

sudo chmod o-rwx Engineering <- Normal users cant do anything in the file
sudo chmod o-rwx Sales
sudo chmod o-rwx HR
ls -l

```
drwxrwx---   2 E-admin  engi-team    4096 Oct 12 17:50 Engineering
drwxrwx---   2 HR-admin hr-team      4096 Oct 12 17:50 HR
drwxrwx---   2 S-admin  sales-team   4096 Oct 12 17:50 Sales
```

- “chown” means we are changing file ownership.
- “chgrp” means we can change group ownership of files.
- “ls -l” means we are showing ourselves a log of information about different files and directories.
- “g+rwx” means groups of a specific file/directory get rwx = read, write and execute permissions. “+” means add and “-” means remove. g = group, o = others and u = user.

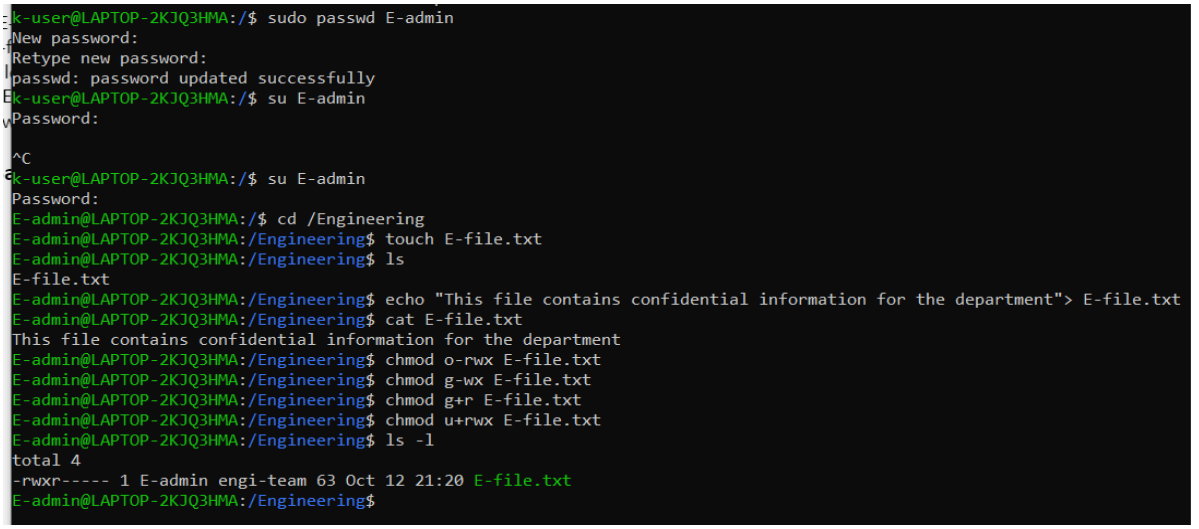
~~6. Create a document in each of the department directories.~~

~~a. The ownerships on this file will be the same as the directory it is located in.~~
~~b. The document should contain only one line of text that states, “This file contains confidential information for the department.”~~
~~c. This file can be read by any user in the department but can only be modified by the department administrator. No one else has permissions to this file.~~

sudo passwd E-admin
kaj <-- Password
su E-admin
cd /Engineering

touch E-file.txt
ls <-- checks file in the directory
echo "This file contains confidential information for the department" > E-file.txt
cat E-file.txt
ls -l <-- Show file ownerships
chmod o-rwx E-file.txt
chmod g-wx E-file.txt
chmod g+r E-file.txt
chmod u+rwx E-file.txt
ls -l <-- Show rwx

Repeat for Sales and HR directory!



- “passwd” means we are manually creating a password for a user.
- “touch” means we are creating for example a file.
- “echo” means we can input data inside the file.
- “cat” means we can view the content inside the file.

Part 2.

Case Scenario

In the User Management challenge lab, you were tasked with creating users and groups. Using the commands one at a time from the command line can be a tedious process and could lead to potential errors in syntax. It is your duty, as an administrator, to make the process as seamless and efficient as possible.

Objectives

1. ~~Create a bash script to perform user management tasks as outlined below:~~
    a. ~~Create a new group. Each group must have a unique name. The script must check to ensure that no duplicate group names exist on the system. If a duplicate is found, an error needs to be reported, and the administrator must try another group name.~~

    MUST BE ROOT USER TO CREATE SCRIPT!
    This "<--" means changes to the script!
    An explanation for the code will be at the end, since a lot here gets repeated.

    ```
    sudo su
    vi groupscript.sh

    #!/bin/bash
    read -p "Enter a name for the new group: " group

    if grep -q $group /etc/group
    then
        echo "The group $group already exists, try a new name"
        exit 1
    fi

    sudo groupadd $group

    :wq

    bash groupscript.sh <- To test it!
    ```

```
root@LAPTOP-2KJQ3HMA: /home/k-user
#!/bin/bash
read -p "Enter a name for the new group: " group

if grep -q $group /etc/group
then
        echo "The group $group already exists, try a new name"
        exit 1
fi

sudo groupadd $group
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"groupscript.sh" 12L, 187C
```

```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash groupscript.sh
Enter a name for the new group: erikgroup
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

After typing “getent group” we get this:

```
erikgroup:x:1016:
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

Now we use the same bash script again and try the same group name to see what happens:

```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash groupscript.sh
Enter a name for the new group: erikgroup
The group erikgroup already exists, try a new name
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

Everything is working as intended.

b. ~~Create a new user. Each user must have a unique name. The script must check to ensure that no duplicate usernames exist on the system. If a duplicate is found, an error needs to be reported and the administrator must try another username. The user will have a Bash login shell and belong to the group that was created in the previous step.~~

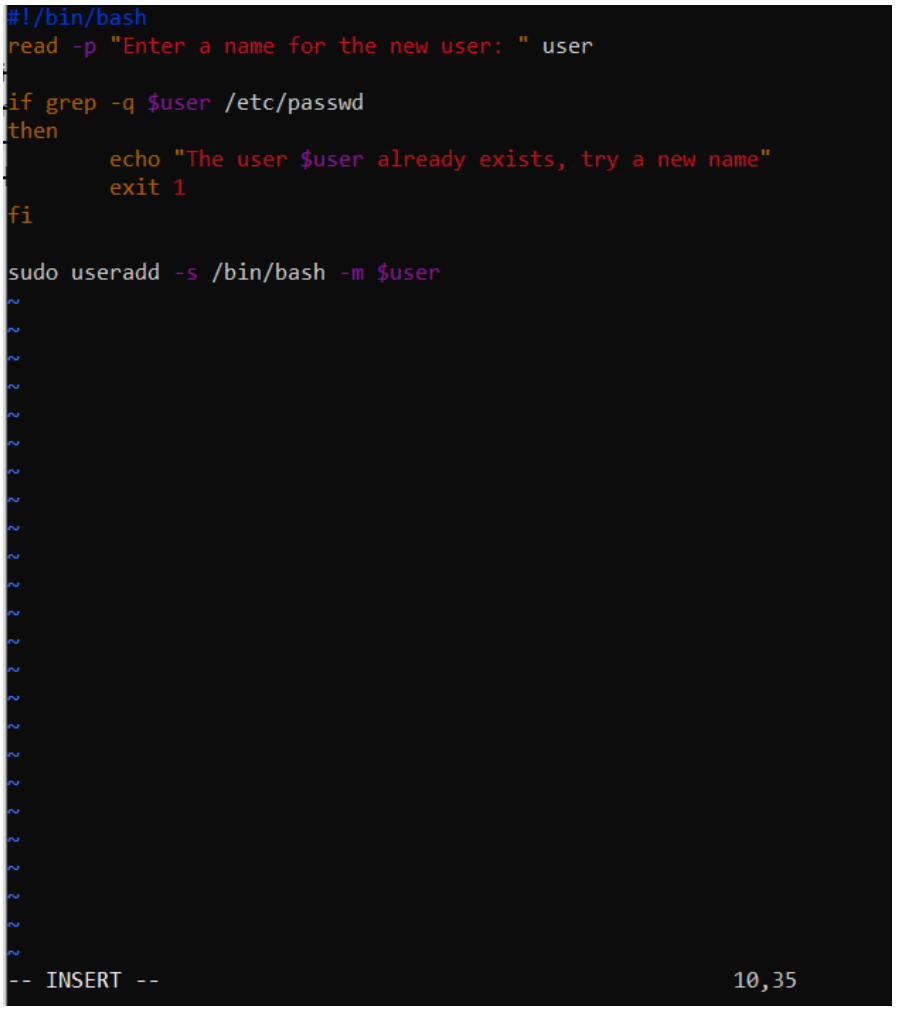
sudo su
vi userscript.sh

#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
	echo "The user $user already exists, try a new name"
	exit 1
fi

sudo useradd -s /bin/bash -m $user

:wq



```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash userscript.sh
Enter a name for the new user: erik-user
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash userscript.sh
Enter a name for the new user: erik-user
The user erik-user already exists, try a new name
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

c. ~~Create a password for each user that is created.~~

sudo su
vi userscript.sh

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m $user
sudo passwd $user
```

:wq

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m $user
sudo passwd $user
```

```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash userscript.sh
Enter a name for the new user: erik1-user
New password:
Retype new password:
passwd: password updated successfully
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

d. ~~Ensure that the new user created is a member of the new group created.~~

```
sudo su
vi userscript.sh

#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m $user
sudo passwd $user

read -p “Enter the group name you want to join: “ group

if grep -q $group /etc/group
then
        echo “You have joined $group!”
        sudo usermod -g $group $user
else
        echo “The group does not exist try again!”
        sudo userdel $user
        exit
fi
:wq
```

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m $user
sudo passwd $user

read -p "Enter the group name you want to join: " group

if grep -q $group /etc/group
then
        echo "You have joined $group!"
        sudo usermod -g $group $user
else
        echo "The group does not exist try again!"
        sudo userdel $user
        exit
fi
~
```

This shows that the script works:

```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash userscript.sh
Enter a name for the new user: erik2-user
New password:
Retype new password:
passwd: password updated successfully
Enter the group name you want to join: erikgroup
You have joined erikgroup!
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

This shows if it were to be a group that does not exist, and the code works as intended:

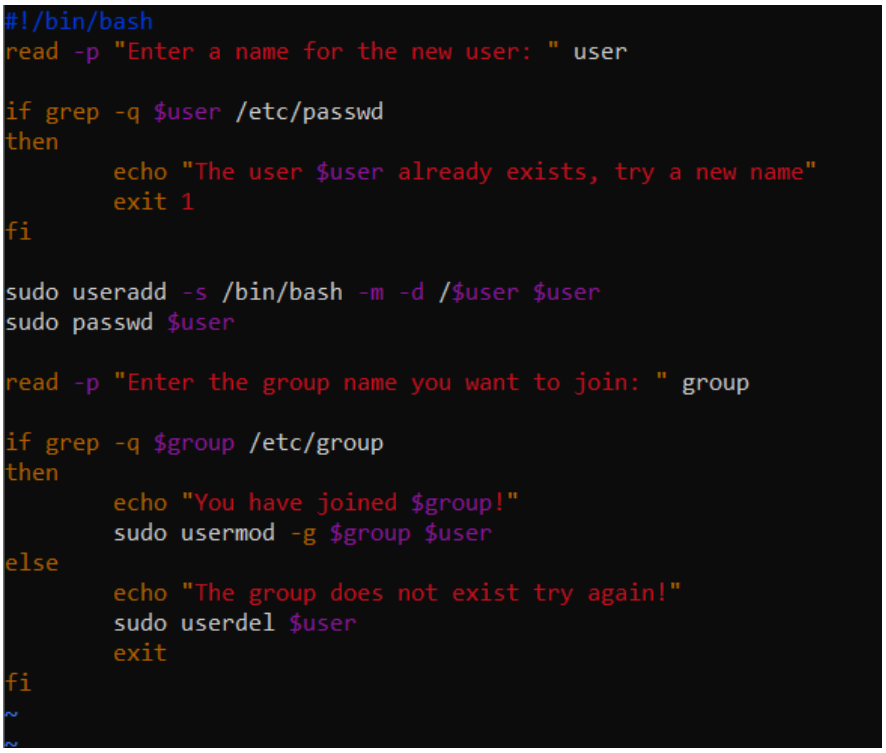
```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash userscript.sh
Enter a name for the new user: erik3-user
New password:
Retype new password:
passwd: password updated successfully
Enter the group name you want to join: asdadasdsadfsdf
The group does not exist try again!
root@LAPTOP-2KJQ3HMA:/home/k-user# id erik3-user
id: ‘erik3-user’: no such user
root@LAPTOP-2KJQ3HMA:/home/k-user#
```

e. ~~Create a directory at the root (/) of the file system with same name as the user created.~~

sudo su

vi userscript.sh

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user <--
sudo passwd $user

read -p “Enter the group name you want to join: “ group

if grep -q $group /etc/group
then
        echo “You have joined $group!”
        sudo usermod -g $group $user
else
        echo “The group does not exist try again!”
        sudo userdel $user
        exit
fi
:wq
```



```
root@LAPTOP-2KJQ3HMA:/home/k-user# bash userscript.sh
Enter a name for the new user: erik4-user
New password:
Retype new password:
passwd: password updated successfully
Enter the group name you want to join: erikgroup
You have joined erikgroup!
root@LAPTOP-2KJQ3HMA:/home/k-user# cd /
root@LAPTOP-2KJQ3HMA:/# ls
Engineering  dev         lib         media  run           sys   var
HR           erik4-user  lib32       mnt    sampletest.sh tmp
Sales        etc         lib64       opt    sbin          user1
bin          home        libx32      proc   snap          user2
boot         init        lost+found  root   srv           usr
root@LAPTOP-2KJQ3HMA:/#
```

f. ~~Set the ownership of the directory to the user and group created.~~

sudo su
vi userscript.sh

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user
sudo passwd $user

read -p “Enter the group name you want to join: “ group

if grep -q $group /etc/group
then
        echo “You have joined $group!”
        sudo usermod -g $group $user
        sudo chgrp $group /$user <--
else
        echo “The group does not exist try again!”
        sudo userdel $user
        sudo rm -rf /$user <--
        exit
fi
:wq
```

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user
sudo passwd $user

read -p "Enter the group name you want to join: " group

if grep -q $group /etc/group
then
        echo "You have joined $group!"
        sudo usermod -g $group $user
        sudo chgrp $group /$user
else
        echo "The group does not exist try again!"
        sudo userdel $user
        sudo rm -rf /$user
        exit
fi
~
~
```

After running the bash script again and making a user named “erik5-user” we can see the ownerships down below:

```
drwxr-xr-x   2 erik5-user erikgroup    4096 Oct 12 22:05 erik5-user
```

g. ~~Set the permissions of the directory to full control for the owner and full control for the group created.~~

sudo su
vi userscript.sh

#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
    echo "The user $user already exists, try a new name"
    exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user
sudo passwd $user

read -p “Enter the group name you want to join: “ group

if grep -q $group /etc/group
then
    echo “You have joined $group!”
    sudo usermod -g $group $user
    sudo chgrp $group /$user

sudo chmod g+rwx /$user <--
else
echo “The group does not exist try again!”
sudo userdel $user
sudo rm -rf /$user
exit
fi
:wq

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user
sudo passwd $user

read -p "Enter the group name you want to join: " group

if grep -q $group /etc/group
then
        echo "You have joined $group!"
        sudo usermod -g $group $user
        sudo chgrp $group /$user
        sudo chmod g+rwx /$user
else
        echo "The group does not exist try again!"
        sudo userdel $user
        sudo rm -rf /$user
        exit
fi
~
```

```
Enter a name for the new user: erik6-user
New password:
Retype new password:
passwd: password updated successfully
Enter the group name you want to join: erikgroup
You have joined erikgroup!
root@LAPTOP-2KJQ3HMA:/home/k-user# cd /
root@LAPTOP-2KJQ3HMA:/# ls -l
total 720
drwxrwx---   2 E-admin    engi-team    4096 Oct 12 21:20 Engineering
drwxrwx---   2 HR-admin   hr-team      4096 Oct 12 17:50 HR
drwxrwx---   2 S-admin    sales-team   4096 Oct 12 17:50 Sales
lrwxrwxrwx   1 root       root            7 Apr 23  2020 bin -> usr/bin
drwxr-xr-x   2 root       root         4096 Apr 23  2020 boot
drwxr-xr-x   8 root       root         2720 Oct 12 17:12 dev
drwxr-xr-x   2 erik4-user erikgroup    4096 Oct 12 21:54 erik4-user
drwxr-xr-x   2 erik5-user erikgroup    4096 Oct 12 22:05 erik5-user
drwxrwxr-x   2 erik6-user erikgroup    4096 Oct 12 22:14 erik6-user
drwxr-xr-x  92 root       root         4096 Oct 12 22:14 etc
```

h. ~~Let the permission to ensure that only the owner of a file can delete it from the directory.~~

vi userscript.sh

#!bin/bash

read -p “Enter a name for the new user: “ user

if grep -q $user /etc/passwd
then
    echo “The user $user already exists, try a new name”
    exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user
sudo passwd $user

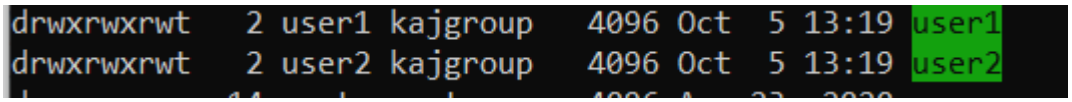
read -p “Enter the group name you want to join: “ group

if grep -q $group /etc/group
then
    echo “You have joined $group!”
    sudo usermod -g $group $user
    sudo chgrp $group /$user
    sudo chmod 1777 /$user <- Sticky bit with the “1”
else
    echo “The group does not exist try again!”
    echo “User and directory removed successfully!”
    sudo userdel $user
    sudo rm -rf /$user
    exit 1
fi

:wq

```
#!/bin/bash
read -p "Enter a name for the new user: " user

if grep -q $user /etc/passwd
then
        echo "The user $user already exists, try a new name"
        exit 1
fi

sudo useradd -s /bin/bash -m -d /$user $user
sudo passwd $user

read -p "Enter the group name you want to join: " group

if grep -q $group /etc/group
then
        echo "You have joined $group!"
        sudo usermod -g $group $user
        sudo chgrp $group /$user
        sudo chmod 1777 /$user
else
        echo "The group does not exist try again!"
        sudo userdel $user
        sudo rm -rf /$user
        exit
fi
```

Sticky bit in other words “1777” is what ensures that only the owner of the file can delete it from the directory if not root.
Sticky bit may also be written like “ +t” and to see if it enabled you can see it in ls -l command, with example here below:



The “t” at the end of all permissions on the left is what proves that sticky bit is enabled. Right now others have access to w and x, but will be changed soon.

i. Ensure that the script is executable.

Yes, everything works!

Explaining the code 1 by 1 using numbers, from logical order from top to bottom.

1. We receive input from the user about what new user he/she wants to create.
2. By using “if grep -q $user” we use the “grep” command to search for the specific user we typed in “/etc/passwd”, so that if the user already exists the script will end and he/she will have to run the script again, until he/she types a name that is unique.
3. “exit 1” and “fi” means we are ending that part.
4. Under “fi” we use the necessary commands to create the user plus its directory and password for the user, also bash login shell.
5. After that part of the script we want the user to enter a specific group he/she wants to join.
6. By using the same command as before “grep” we search for every group in the system in “/etc/group”, if the group exists then the user will join that group. If not then we delete the user and the users directory.
7. Why we delete the user and its directory is because if we do not delete them right now, it will cause issues in the system since you have already created them previously, so by deleting them the user can run the script again just fine, and not worry about deleting the directory and user manually.
8. With the command “usermod” we add the user into the group.
9. With the command “chgrp” the group gets full access to the user file.
10. With the command “chmod 1777” we fix the sticky bit plus everyone gets full access to the specific user file.
11. With “userdel $user” we delete the user.
12. With “rm -rf /$user” we force delete the directory of the specific user.
13. Then the script completely ends.