EPAM University Programs

DevOps external course

Module 4 Linux & Bash Essentials

TASK 4.6

1. *User management*. Here we suppose there are at least two users, namely, root and guest.

(i) Create a new user *user*

groupadd user

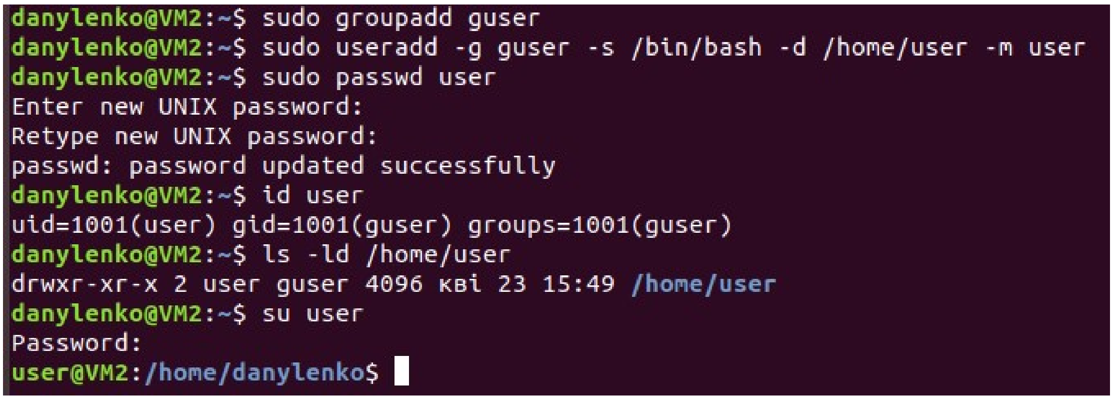
useradd -g user -s /bin/bash -d /home/user -m user

passwd user

id user

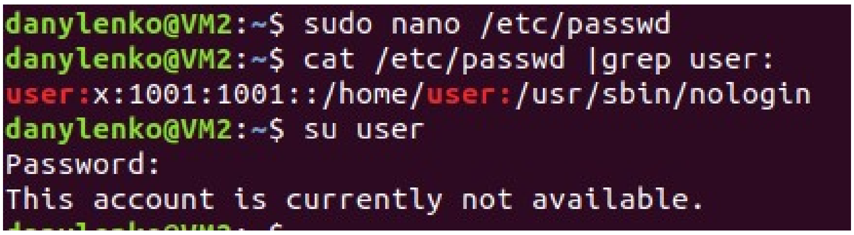
ls -ld /home/user

(ii) Log in to the system as "user" (hint use **su**).



```
danylenko@VM2:~$ sudo groupadd guser
danylenko@VM2:~$ sudo useradd -g guser -s /bin/bash -d /home/user -m user
danylenko@VM2:~$ sudo passwd user
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
danylenko@VM2:~$ id user
uid=1001(user) gid=1001(guser) groups=1001(guser)
danylenko@VM2:~$ ls -ld /home/user
drwxr-xr-x 2 user guser 4096 кві 23 15:49 /home/user
danylenko@VM2:~$ su user
Password:
user@VM2:/home/danylenko$
```

(ii) Edit **/etc/passwd** to prevent user *user* from logging in to the system.



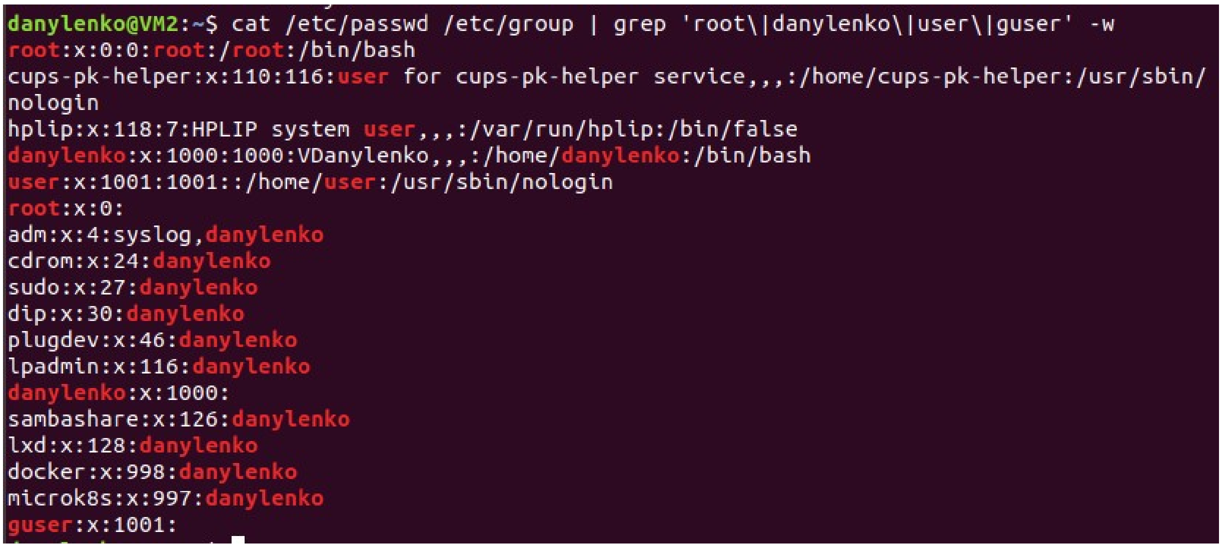
```
danylenko@VM2:~$ sudo nano /etc/passwd
danylenko@VM2:~$ cat /etc/passwd |grep user:
user:x:1001:1001::/home/user:/usr/sbin/nologin
danylenko@VM2:~$ su user
Password:
This account is currently not available.
```

2. *Content of* **/etc/passwd** and **/etc/group**.

(i) Look through **/etc/passwd** and **/etc/group** (hint: use **less** or **cat**).

Looked through it, decided there is no need for screenshot

(ii) Get data from **/etc/passwd** and **/etc/group** about users: *root*, *guest*, *user* (hint: filter by **grep**).



```
danylenko@VM2:~$ cat /etc/passwd /etc/group | grep 'root\|danylenko\|user\|guser' -w
root:x:0:0:root:/root:/bin/bash
cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/
nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
danylenko:x:1000:1000:VDanylenko,,,:/home/danylenko:/bin/bash
user:x:1001:1001::/home/user:/usr/sbin/nologin
root:x:0:
adm:x:4:syslog,danylenko
cdrom:x:24:danylenko
sudo:x:27:danylenko
dip:x:30:danylenko
plugdev:x:46:danylenko
lpadmin:x:116:danylenko
danylenko:x:1000:
sambashare:x:126:danylenko
lxd:x:128:danylenko
docker:x:998:danylenko
microk8s:x:997:danylenko
guser:x:1001:
```

(iii)Parse **/etc/passwd** and **/etc/group** with cut.

Her we cut all lines by using delimiter “:” means we get list of parameters from columns of file

cut -f1 -d: /etc/passwd list of users

cut -f1,2 -d: /etc/passwd list of user:x x –means shadowing enabled and encrypted passwords stored in /etc/shadow

cut -f1,7 -d: /etc/passwd list of user:shell

cut -f1 -d: /etc/group list of groups

cut -f1,2 -d: /etc/group list of groups:x

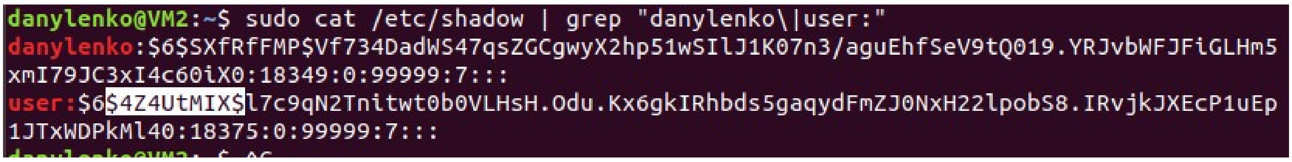
(iv) Try to call **less** on **/etc/shadow** and invoke

sudo less /etc/shadow

man -k shadow

man 5 shadow

Analyse content of **/etc/shadow** based on what you’ve found in **man** 5 shadow.



```
danylenko@VM2:~$ sudo cat /etc/shadow | grep "danylenko\|user:"
danylenko:$6$SXfRfFMP$Vf734DadWS47qsZGCgwyX2hp51wSIlJ1K07n3/aguEhfSeV9tQ019.YRJvbWFJFiGLHm5
xmI79JC3xI4c60iX0:18349:0:99999:7:::
user:$6$4Z4UtMIX$l7c9qN2Tnitwt0b0VLHsH.Odu.Kx6gkIRhbds5gaqydFmZJ0NxH22lpobS8.IRvjkJXEcP1uEp
1JTxWDPkMl40:18375:0:99999:7:::
```

This file contains encrypted password info. Its stored in fields with “:” delimiter

1. Username
2. Hash of decrypted password in 3 parts each part starts from $
   $ “hash_algorithm”$”hash_salt”$”hash_data”
   $6 – means SHA-512 Hash Algoritm
3. Date of password change as example :18375 days after Jan 1, 1970 for «user:»
4. 0 –(days) minimum password age, when it’s possible to start changing password
5. 99999 (days) maximum password age

6. 7 (days) – length of user warning period before pwd expire date from field #5
7. Empty – number of days expired pwd will be still active. Empty means will be expired in 99999 days without inactivity period
8. Empty – account expiration date, also number of days after Jan 1, 1970

3. Dealing with **chmod**.

(i) An executable script. Open your favorite editor and put these lines into a file

#!/bin/bash

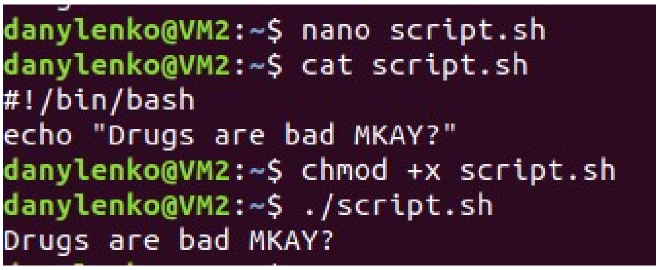
echo "Drugs are bad MKAY?"

Give name "script.sh" to the script and call to

chmod +x script.sh

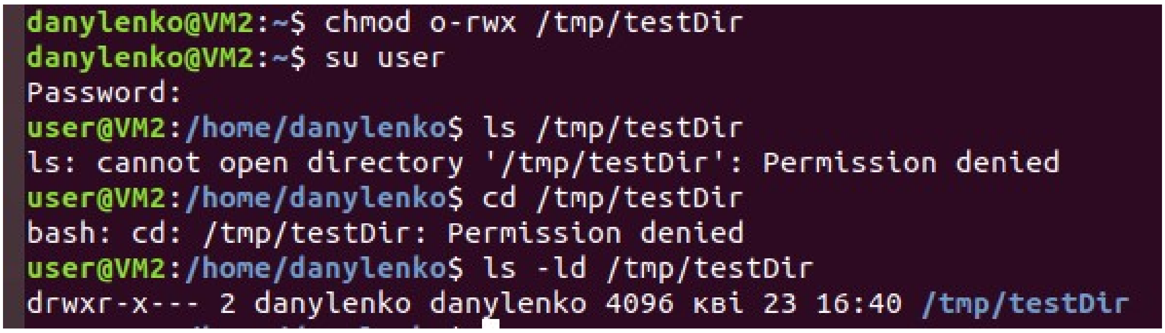
Then you are ready to execute the script:

./**script.sh**



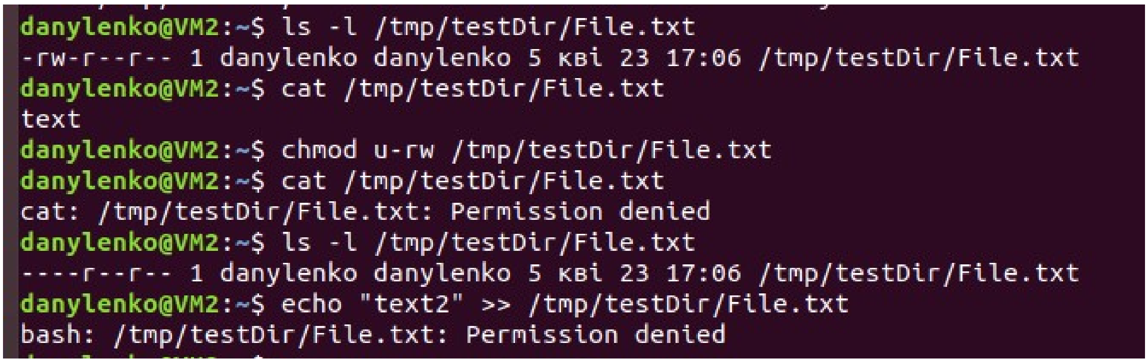
```
danylenko@VM2:~$ nano script.sh
danylenko@VM2:~$ cat script.sh
#!/bin/bash
echo "Drugs are bad MKAY?"
danylenko@VM2:~$ chmod +x script.sh
danylenko@VM2:~$ ./script.sh
Drugs are bad MKAY?
```

(ii) Suppose, you have logged in to the system as *guest*. Create directory "testDir" in the **/tmp**; put some file into testDir and prohibit user *user* from visiting this directory (i.e. "testDir").



```
danylenko@VM2:~$ chmod o-rwx /tmp/testDir
danylenko@VM2:~$ su user
Password:
user@VM2:/home/danylenko$ ls /tmp/testDir
ls: cannot open directory '/tmp/testDir': Permission denied
user@VM2:/home/danylenko$ cd /tmp/testDir
bash: cd: /tmp/testDir: Permission denied
user@VM2:/home/danylenko$ ls -ld /tmp/testDir
drwxr-x--- 2 danylenko danylenko 4096 кві 23 16:40 /tmp/testDir
```

(iii) Test, if it possible to forbid an owner of some file to read to or write from this file.



```
danylenko@VM2:~$ ls -l /tmp/testDir/File.txt
-rw-r--r-- 1 danylenko danylenko 5 кві 23 17:06 /tmp/testDir/File.txt
danylenko@VM2:~$ cat /tmp/testDir/File.txt
text
danylenko@VM2:~$ chmod u-rw /tmp/testDir/File.txt
danylenko@VM2:~$ cat /tmp/testDir/File.txt
cat: /tmp/testDir/File.txt: Permission denied
danylenko@VM2:~$ ls -l /tmp/testDir/File.txt
----r--r-- 1 danylenko danylenko 5 кві 23 17:06 /tmp/testDir/File.txt
danylenko@VM2:~$ echo "text2" >> /tmp/testDir/File.txt
bash: /tmp/testDir/File.txt: Permission denied
```