**Lab 3 User and group accounts**

1. **Overview**

In this lab, you will practice creating user and group accounts. You will also start using file permissions to control access to files and directories.

**Multiple user accounts will be used for this lab**

Creating accounts and groups requires superuser privilege, so those commands will require the use of sudo. Once you have created users, you will be perorming tasks as those users, so for those tasks, you will be required to login as those users. For this purpose, it is recommended that in a terminal window, you use

login username

to start a login shell as the desired user in that window. When you are done running commands as that user, you can use the

logout

command to return to the shell you were using before you entered the login username command.

1. **Tutorial**

**2.1 Create a group for our users to use for collaboration, and a group directory for shared files**

1. Create a group named **developers**.
2. sudo addgroup developers
3. Create a directory named **/projects/bigdeal** owned by **root** and the **developers** group.
4. The directory must be full access for the owner and the group, with no access for any other users and it must have the **set-group-id**bit turned **on**.
5. Create a subdirectory in **/projects/bigdeal** named **incoming**, with the same ownership and permissions as **/projects/bigdeal**.
6. sudo mkdir -p /projects/bigdeal/incoming
7. sudo chgrp -R developers /projects/bigdeal
8. sudo chmod -R 2770 /projects/bigdeal

**2.2 Create accounts for our users**

1. Create a user named **harry** with a home directory.
2. sudo adduser harry
3. Create a user named **sally** with a home directory.
4. sudo adduser sally
5. Create a user named **cupid** with a home directory.
6. sudo adduser cupid
7. Add **harry**, **sally** and **cupid** to the **developers** group as a supplementary group for each of them.
8. sudo adduser harry developers
9. sudo adduser sally developers
10. sudo adduser cupid developers

**2.3 Log into the harry account and perform the following tasks:**

login harry

1. Create a file named **myclothes.txt** in **harry**’s home directory with 5 lines of text in it, each line containing some item of clothing.
2. nano myclothes.txt
3. Copy that file to a file named **harry** in **/projects/bigdeal/incoming**.
4. cp myclothes.txt /projects/bigdeal/incoming/harry
5. logout

**2.4 Log into the sally account and perform the following tasks:**

login sally

1. Create a file named **myfurniture.txt** in **sally**’s home directory with 5 lines of text in it, each line containing some kind of furniture.
2. nano myfurniture.txt
3. Copy that file to a file named **sally** in **/projects/bigdeal/incoming**.
4. cp myfurniture.txt /projects/bigdeal/incoming/sally
5. logout

**2.5 Log into the cupid account and perform the following tasks:**

login cupid

1. Using the paste command, create a file called **when-harry-met-sally.txt** in **/projects/bigdeal** by pasting together the two files **harry** and **sally** and use redirection to put the output into the **incoming** directory.
2. cd /projects/bigdeal/incoming
3. paste harry sally > ../when-harry-met-sally.txt
4. Remove the two files from the incoming directory.
5. rm harry sally
6. Use cat to display the contents of the **when-harry-met-sally.txt** file.
7. cd ..
8. cat when-harry-met-sally.txt
9. logout
10. Find all files on the system belonged to the **developers** group, and then show the files owned by user **harry.**
11. cd ~/
12. find ./ -group {developers}
13. find ./ -user {harry}
14. Find all \*.txt files on the system belonged to the developers group.
15. find ./ -group {developers} –name “\*.txt”
16. **Exercise**
17. Create some users for your friends as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **username** | **GECOS name** | **password** | **shell** |
| dangle | phone number, home address | jim | /bin/bash |
| junior | phone number, home address | travis | /bin/sh |
| wiegel | phone number, home address | trudy | /bin/sh |
| kimball | phone number, home address | cheresa | /bin/sh |

(a) The user must change the password during the first system login.

(b) Password must be changed every 3 months.

(c) Account expires 1st January 2021.

(d) Add phone number, mail, etc... in GECOS

1. Create groups and assign their members according to the following table:

|  |  |
| --- | --- |
| **group name** | **members** |
| reno | dangle, wiegel, junior, kimball |
| miami | dangle, wiegel, junior, kimball |
| commanding | dangle |
| competent | kimball |
| SWAT | dangle, junior |

1. For all the users you created above, ensure that only they have permission to view or modify the following files: .bash\_profile .bashrc
2. Create the directory /home/reno, setgid active and owned by group reno. Do the same for the directory /home/miami. Try to figure out what setgit is for. (Please refer https://blog.csdn.net/frank\_jb/article/details/40192003)
3. Enable the cracklib module in PAM for the command passwd, configure it to force the size of new passwords to at least 8 characters and check its utility.

(Please refer to:

[1] <https://www.ibm.com/developerworks/cn/linux/l-pam/index.html>

[2] <https://blog.csdn.net/linfanhehe/article/details/50072127>

[3] <https://www.serverwatch.com/tutorials/article.php/3771431/Setting-Password-Policy-With-PAM.htm>

[4] <https://www.cyberciti.biz/faq/securing-passwords-libpam-cracklib-on-debian-ubuntu-linux/>

[5] <http://www.deer-run.com/~hal/sysadmin/pam_cracklib.html>

[6] https://www.cyberciti.biz/security/linux-password-strength-checker/

)

**Submission of Results**

Capture screenshots of the terminal window showing your prompt, commands, and as much of the output for each command as fits in the window with your command still showing. Submit a single PDF file to blackboard containing your screenshots in the order they were performed, along with your responses to the questions asked in the lab.