Linux Server Administration (First level full-time studies)

Mgr inz Patient ZIHISIRE MUKE
Department of Applied Informatics
Faculty of Computer Science and Management
Wrocław University of Technology
Patient.zihisire@pwr.edu.pl

LAB 4: Practical management of accounts and user groups

Version 1.0-250321

Introduction

Use these exercises to test your knowledge of adding and managing user and group accounts in Linux. For questions that involve adding and removing user accounts, you can use the Users window, the User Manager window, or command-line tools such as useradd and usermod. The point is to make sure that you get the correct results shown in the answers that follow, not necessarily do it exactly the same way I did. There are multiple ways you can achieve the same results. The answers here show how to complete the exercises from the command line. (Become root user when you see a # prompt.)

Important notes

- **Create user 1 with your own names in order "first name then second name separated by the low dash symbole (_)", then give the username as the initial of your first name plus the second name in full but in one word.**
  Example: "John_Baxter" as full name and "jboxter" as username.
- **Create user 2 with your own names in order "second name then first names separated by the low dash symbol (_)", then give the username as the initial of your second name plus the first name in full but in one word.**
  Example: "Baxter_ John" as full name and "bjohn" as username.
- **For the group created to store user 1 use your own second name in lower case.**
- **For the group created to store user 2 use your own first name in lower case.**
- **For the group ID, give the group ID the last 3 digits (numbers) of your student number.**
- **For password use : "P@ssw0rd"**

Task 1. Add a local user account to your Linux system (0.5 points)

1. Add a local user account to your Linux system that has a username of jbaxter and a full name of John Baxter and that uses /bin/sh as its default shell. Let the UID be assigned by default. Set the password for jbaxter to: My1N1te0ut!

To add a local user account to your Linux system that has a username of jbaxter and a full name of John Baxter, which uses /bin/sh as its default shell and is the next available UID (yours may differ from the one shown here), enter the following. You can use the grep command to check the new user account. Then set the password for jbaxter to: My1N1te0ut!

$ **su -**
# **useradd -m -c "John Baxter" -s /bin/sh jbaxter**
# **grep jbaxter /etc/passwd**
jbaxter:x:1001:1001:John Baxter:/home/jbaxter:/bin/sh
passwd jbaxter
Changing password for user jbaxter
New password: **My1N1te0ut!**
Retype new password: **My1N1te0ut!**
passwd: all authentication tokens updated successfully

2.C reate a group account named testing that uses group ID 315.

To create a group account named testing that uses group ID 315, enter the following:

```
# groupadd -g 315 testing
# grep testing /etc/group
testing:x:315:
```

3.A dd jbaxter to the testing group and the bin group.

To add jbaxter to the testing group and the bin group, enter the following:

```
# usermod -aG testing,bin jbaxter
# grep jbaxter /etc/group
bin:x:1:bin,daemon,jbaxter
jbaxter:x:1001:
testing:x:315:jbaxter
```

4. Open a shell as jbaxter (either a new login session or using a current shell) and temporarily have the testing group be your default group so that when you type touch /home/jbaxter/file.txt, the testing group is assigned as the file's group.

To become jbaxter and temporarily have the testing group be jbaxter's default group, run touch /home/jbaxter/file.txt so that the testing group is assigned as the file's group, and do the following:

```
$ su - jbaxter
Password: My1N1te0ut!
sh-4.2$ newgrp testing
sh-4.2$ touch /home/jbaxter/file.txt
sh-4.2$ ls -l /home/baxter/file.txt
-rw-rw-r--. 1 jbaxter testing 0 Jan 25 06:42 /home/jbaxter/file.

txt
sh-4.2$ exit ; exit
```

5. Note what user ID has been assigned to jbaxter, and then delete the user account without deleting the home directory assigned to jbaxter.

```
$ su -

# userdel jbaxter
```

Task 2. Customise user account (0.5 points)

6. Find any files in the /home directory (and any subdirectories) that are assigned to the user ID that recently belonged to the user named jbaxter.

Use the following command to find any files in the /home directory (and any subdirectories) that are assigned to the user ID that recently belonged to the user named jbaxter. (When I did it, the UID/GID were both 1001; yours may differ.) Notice that the username jbaxter is no longer assigned on the system, so any files that user created are listed as belonging to UID 1001 and GID 1001, except for a couple of files that were assigned to the testing group because of the newgrp command run earlier:

```
# find /home -uid 1001 -ls
262184 4 drwx------ 4 1001 1001 4096 Jan 25 08:00 /home/jbaxter
262193 4 -rw-r--r-- 1 1001 1001 176 Jan 27 2011 /home/jbaxter/.bash_profile
262196 4 -rw------- 1 13602 testing 93 Jan 25 08:00 /home/jbaxter/.bash_history
262194 0 -rw-rw-r-- 1 13602 testing 0 Jan 25 07:59 /home/jbaxter/file.txt
...
```

7. Copy the /etc/services file to the default skeleton directory so that it shows up in the home directory of any new user. Then add a new user to the system named mjones, with a full name of Mary Jones and a home directory of /home/maryjones.

Run these commands to copy the /etc/services file to the /etc/skel/ directory; then add a new user to the system named mjones, with a full name of Mary Jones and a home directory of /home/maryjones. List her home directory to make sure that the services file is there.

```
# cp /etc/services /etc/skel/
# useradd -m -d /home/maryjones -c "Mary Jones" mjones

# ls -l /home/maryjones
total 628
-rw-r--r--. 1 mjones mjones 640999 Jan 25 06:27 services
```

8. Find all files under the /home directory that belong to mjones. Are there any files owned by mjones that you didn't expect to see?

Run the following command to find all files under the /home directory that belong to mjones. If you did the exercises in order, notice that after you deleted the user with the highest user ID and group ID, those numbers were assigned to mjones. As a result, any files left on the system by jbaxter now belong to mjones. (For this reason, you should remove or change ownership of files left behind when you delete a user.)

```
# find /home -user mjones -ls
262184 4 drwx------ 4 mjones mjones 4096 Jan 25 08:00 /home/jbaxter
262193 4 -rw-r--r-- 1 mjones mjones 176 Jan 27 2011 /home/jbaxter/.bash_profile
262189 4 -rw-r--r-- 1 mjones mjones 18 Jan 27 2011 /home/jbaxter/.bash_logout
262194 0 -rw-rw-r-- 1 mjones testing 0 Jan 25 07:59 /home/jbaxter/file.txt
262188 4 -rw-r--r-- 1 mjones mjones 124 Jan 27 2011 /home/jbaxter/.bashrc
262197 4 drwx------ 4 mjones mjones 4096 Jan 25 08:27 /home/maryjones
262207 4 -rw-r--r-- 1 mjones mjones 176 Jan 27 2011 /home/maryjones/.bash_profile
262202 4 -rw-r--r-- 1 mjones mjones 18 Jan 27 2011 /home/maryjones/.bash_logout
262206 628 -rw-r--r-- 1 mjones mjones 640999 Jan 25 08:27 /home/maryjones/services
262201 4 -rw-r--r-- 1 mjones mjones 124 Jan 27 2011 /home/maryjones/.bashrc
```

9. Log in as mjones, and create a file called /tmp/maryfile.txt. Using ACLs, assign the bin user read/write permission to that file. Then assign the lp group read/write permission to that file.

As the user mjones, you can use the following to create a file called /tmp/maryfile.txt, and use ACLs to assign the bin user read/write permission and the lp group read/write permission to that file.

```
[mjones]$ touch /tmp/maryfile.txt
[mjones]$ setfacl -m u:bin:rw /tmp/maryfile.txt
[mjones]$ setfacl -m g:lp:rw /tmp/maryfile.txt
[mjones]$ getfacl /tmp/maryfile.txt
# file: tmp/maryfile.txt
# owner: mjones
# group: mjones
user::rw-

user:bin:rw-

group::rw-

group:lp:rw-

mask::rw-

other::r& —
```

10. Still as mjones, create a directory named /tmp/mydir. Using ACLs, assign default permissions to that directory so that the adm user has read/write/execute permission to that directory and any files or directories created in it. Create the /tmp/mydir/testing/ directory and /tmp/mydir/newfile.txt file, and make sure that the adm user was also assigned full read/write/execute permissions. (Note that despite rwx permission being assigned to the adm user, the effective permission on newfile.txt is only rw. What could you do to make sure that adm gets execute permission as well?)

Run this set of commands (as mjones) to create a directory named /tmp/mydir, and use ACLs to assign default permissions to it so that the adm user has read/ write/execute permission to that directory and any files or directories created in it. Test that it worked by creating the /tmp/mydir/testing/ directory and /tmp/mydir/newfile.txt.

```
[mary]$ mkdir /tmp/mydir
[mary]$ setfacl -m d:u:adm:rwx /tmp/mydir
[mjones]$ getfacl /tmp/mydir
# file: tmp/mydir
# owner: mjones
# group: mjones
user::rwx
group::rwx
other::r-x
default:user::rwx
default:user:adm:rwx
default:group::rwx
default:mask::rwx
default:other::r-x
[mjones]$ mkdir /tmp/mydir/testing
[mjones]$ touch /tmp/mydir/newfile.txt
[mjones]$ getfacl /tmp/mydir/testing/
# file: tmp/mydir/testing/
# owner: mjones
# group: mjones
user::rwx
user:adm:rwx
group::rwx
mask::rwx
other::r-x
default:user::rwx
default:user:adm:rwx
default:group::rwx
default:mask::rwx
default:other::r-x
[mjones]$ getfacl /tmp/mydir/newfile.txt
# file: tmp/mydir/newfile.txt
# owner: mjones
# group: mjones
user::rw-

user:adm:rwx        #effective:rw-

group::rwx          #effective:rw-

mask::rw-

other::r—
```

Notice that the adm user effectively has only rw- permission. To remedy that, you need to expand the permissions of the mask. One way to do that is with the chmod command, as follows:

[mjones]$ **chmod 775 /tmp/mydir/newfile.txt**
[mjones]$ **getfacl /tmp/mydir/newfile.txt**
file: tmp/mydir/newfile.txt
owner: mjones# group: mjonesuser::rwxuser:adm:rwxgroup::rwxmask::rwxother::r-x

Task 3. Writing Simple Shell Scripts (3 points)

11. Create a simple shell Script that will allow you to add a user with a password to the system.
    NB: The script has to output the message "User was added successfully to the system".
    Log out and log in with the new user before coming back to the initial account and proceed.
    After verification with the lecturer delete the created user
12. Create a simple shell Script that will allow to add multiple users (up to 200) with a password and append those users to groups (up to 4 groups, 50 users each) with Bash Script.

    NB: 2 important steps for this task:
    1. add multiple users and groups at the same time. And add all users in different groups at the same time Also Assign the passwords to all the users of all the groups. Give the permission to created file script to execute using the chmod command.

    2. After verification by the lecturer modify the script to delete all 200 created users and all 4 groups.