A Objective

This lab develops the skills to add and manage users and groups with the Linux operating systems using shell commands. Further, the lab will look behind the scenes of user management in Linux by analyzing the content of the corresponding configuration files. In the second part of this lab, as a more real-world example, we will create the users of a small company called *Gotham City*!

B Platform and process

- You will need a Linux-system where you have superuser (root) privileges (with `sudo` or `su`) – employ one of the prepared VMs running e.g. Kali Linux (the privileged user is called kali or junioradmin)!
- Execute all commands in a terminal window using the standard shell and don't forget to elevate your privileges first using e.g. `sudo su`!

⟶ Be sure to provide answers to all questions and to document **each and every** shell command line in your lab notes (lab report)!

C Existing users

(1) A *regular* linux user belongs to many groups.

a. Which group(s) are you in the (virtual machine) in? *Hint*: use the `id` command.

b. Why are there so many groups?

D Creating a new user

(2) Create a new group called `heroes`*`NN`* ! Note: replace *`NN`* by the last two digits of your login-number! Which (simple) command line did you use to create the new user group?

(3) Changes

a. Which (text) file in `/etc` received a new entry as a consequence to step (2)?

b. List the corresponding line from the `group` file using the command `grep`!

Command:

Result:

c. What is the `group id` of the group `heroes`*`NN`*?

d. Try the following command lines – what is the purpose of each?

```
getent passwd
getent group
```

(4) Create a new user with the following specs – use only *one single* shell command line:

- User name: `max`
- Full name: `Max Headroom`
- Home directory:`/home/max`
- Shell: `/bin/bash`
- "Main" group (= *Login Group*): `heroes`*`NN`*

a. What command line did you use to create the user[1]

b. What (text) system files consequently received new entries - enter the absolute path? List the corresponding lines from the passwd file using the command grep!

c. What is the (numerical) user id of user max?

d. Is it already possible for the user max to login?

(5) Configure a *secure* password for user max!

a. How did you do that?

b. What text file in /etc was changed when adding the password? List the corresponding line with grep! Which command did you use?

(6) Try to login with user name max – does it work? *(If not, what error message do you receive? See (7))*

a. … with the command su - *username*. What does the command su stand for?

Does it work?

b. … via the network using ssh[2]: ssh *username*@localhost

Does it work?

c. If a) and/or b) work, try to login by using the graphical user interface (GUI)!

Does it work?

(7) **Only if (6) did not work (couldn't login):** Check if perhaps the home directory of max does not exist at all *(how do you check this?)* or if maybe the password has not been set *(where would you check this?)*

- If the home directory does not exist (you may have missed to pass the correct option to useradd), you will need to create it using mkdir. The *owner* should be max – and max should have all permissions to his own directory – does the login work now?
- So what was the cause of the problem?

Does it work?

E Creating multiple users

(8) Create the following users, groups and directories on the system:

a. … with the help of a simple *Shell Script* (include it in your lab report – you will need it again later!):

Hint 1: It is easiest to create the script file with an *editor* (vi, nano, gedit…). For each new user, enter a useradd command per line. You can execute the script with sudo sh scriptfile, you don't need sudo in front of each line in your script!

Hint 2: It is a good idea to create both the groups (friends, heroes*NN*, …) and the higher-level folders (/home/friends, /home/heroes*NN*, …) *first*, before adding the users!

Hint 3: You may want to automate adding passwords by entering them directly in the script file *(what is the potential problem here?)* – to achieve this, you *cannot* use the passwd command, as it expects input directly from the console. Use the command chpasswd (which reads from standard input, which in turn can be redirected from a pipe) instead, e.g.:

[1] note the mnemonic trick for the important options of useradd: **G**etting **s**omething **d**one **m**akes **c**harts **g**reen ;-)

[2] you may need to install & activate an SSH-Server first, you can do this using sudo apt-get install openssh-server and sudo systemctl start sshd

```
echo  username:password  | chpasswd
```

Hint 4: It would be smart to make backup copies of all three (important) user management files in `/etc` before creating the groups and users – *how do you do that?*

b. List of users: Note: replace NN by the last two digits of your login-number!

Username	Name	Login Group	Home Directory	Password
bossNN	Your Name	hereosNN	/home/hereos_NN/*boss_NN*	I!love!linux!
batman	Bruce Wayne	hereosNN	/home/hereos_NN/batman	darkknight
robin	Dick Grayson	friends	/home/friends/robin	nightwing
catwoman	Selina Kyle	friends	/home/friends/catwoman	meowmeow
gordon	James Gordon	police	/home/police/gordon	law-n-order
joker	The Joker	villains	/home/villians/joker	whysoserious
penguin	Oswald Cobblepot	villians	/home/villains/penguin	fish!

(9) a. Ensure that all users (including their home directories) have been successfully created – you can test this by logging in via the GUI or directly in the terminal using `su - **username**`!

b. How secure are above passwords?

c. Your script to create the users

d. Write and test a second script file that *deletes* all the users, groups and directories again! Insert your script as text here:

(10) *Bonus* ;-): Who is/was Max Headroom? Where does his name come from?

Have fun!