Network Administration Lab
Experiment#3
User Management

Dr. Ahmed Awad

September 29, 2020

1 Objectives

This experiment aims to practice the basic tasks for users provisioning and management during system administration. This includes adding users, removing users, and managing account-related files.

2 Overview

User management forms a crucial task for network and system administrator. With the continuous growth in the number of users within a network system of different organizations, manual management for different accounts gets more and more impractical. Therefore, effective management of user accounts gains extra attention within the administration process. Besides, user management is a key dominant factor of system security.

User management typically includes adding new user accounts, removing accounts, and any account-related management task. However, as having centralized servers with all accounts in a company is required to be effectively managed, managing several user accounts on a single Linux machine provides insights about the basic user management skills which can be easily scaled later for a large number of users within a network.

In Linux machine, a list of users recognized by the system is found in the file **/etc/passwd**. Each line in this file represents one user and contains several fields that provide information about that user. This information include:

- Login name

- Encrypted password

- User ID.

- Group ID.

- Home directory.

- GECOS Information.

- Login shell.

It is important to mention that managing users through manipulating the content of **/etc/passwd** file forms the basis to user management. Adding, removing, and even disabling accounts can be achieved when having the proper permissions for such operations.

3 Procedure

3.1 Users List

a. Check the list of the current user accounts created on your machine. **What command do you need to use for this purpose?**

b. From the list, explain the information about a randomly chosen user.

c. Write a bash shell script that gets all the information about the currently logging user in **/etc/passwd** file and writes them to a file whose name is the ID of that user.

d. What is the purpose the command **id**. Show its output.

3.2 Adding User Accounts

a. Explain the objective for each of the following commands:

   1. useradd
   2. passwd
   3. su

b. Create a user (with useradd command) whose login name is **skywalk** and the login password is **skywalk** as well.

c. Switch the login to the new user and confirm that you have successfully switched to that user. **What command would you use for this purpose?**

d. Once you have switched to the new user, verify the current group. **What command would you use for this purpose?**

e. Switch back to your original account.

f. Write down the encrypted password for the user **skywalk**. **What encryption algorithm has been used?**

g. Remove the account you have created.

h. Is it possible to create two accounts with the same login name?. **Demonstrate an experiment to prove your answer?**

i. Is the login name case sensitive? **Demonstrate an experiment to prove your answer?**

j. Is it possible to create a user whose ID is 0? **What is the drawback for such behavior? Demonstrate an experiment to prove your answer?**

k. Is it possible to delete a user whose ID is 0? **Why? How can you delete it?**

3.3 Modifying Existing User Accounts

a. Create a user whose name is **test-user**.

b. Create a home directory for this user.

c. Make sure that you can login with the GUI to that user.

d. Logout and return to your account.

e. For the following tasks, you are allowed only to use the commands **usermod** and **groupmod** unless you are explicitly asked to use some other commands:

1. Modify the login name of the user **test-user** to be **test-user-modified**.
2. Modify the group name of the user to be **user-group-modified**.
3. Modify the home directory of the new user to match with its new login name.
4. Switch to the user **test-user-modified** and use the commands **pwd** and **id** to confirm that your modifications have been successfully executed.
5. Switch back to your user account.
6. Change the shell of the user **test-user-modified** to be one that prevent logins (**no logins**.
7. Now switch again to the user **test-user-modified**. Can you do that? Why?

f. Remove the account **test-user-modified**.

3

3.4   Removing User Accounts

a. Briefly explain what steps do you need to successfully remove an account?

b. Create an account named **user-to-delete**.

c. Disable the account temporarily. **Provide two different ways to disable that account?**

d. Enable the account again.

e. Remove the account.

f. Make sure that the account has been successfully removed.

3.5   Managing Users with Scripts

a. Write a bash shell script that asks the user to enter user-name, password, and GECOS information. Then, it creates a user with the specified input.

b. Write a bash shell script that receives a file name from the command line. This file contains a number of user names and their corresponding passwords in clear text. Each line contains a user-name and the password. Your script should successfully create all the users stated in the file.