|  |  |
| --- | --- |
| **DOCUMENT RULES:** | |
| **Task name** | **Work with users** |
| **Task name & column name should be written:** | **Bold (CTRL+B)** |
| **Commands should be written in the after # sign:** | *Italic (CTRL+I) #hostname* |
| **Output photo should be cropped or compressed:**  **Photo could be more than one:**  **If you need extra lines, add the line next after it:** | ***Description photo should be with title bar (CTRL + I + B)*** |
| **All other text should be written:** | Standard |
| **Font name and text size:** | Calibri and 9 |
| **Group name:** | Dev\_ops\_1 |
| **Student name and surname:** | Ehtiram Mustafayev |
| **E-mail:** | ehtiramst@gmail.com |
| **WhatsApp number:** | **+994507492393** |

|  |  |
| --- | --- |
| **Task names** | **Command steps and outputs** |
| **The** whoami **command shows the user we are logged in.** | whoami |
| **We have here a few information. What is the User ID (UID), primary group ID (GID) and other groups to which user belongs to.** | id |
| **We wish to learn more about user** ubuntu | id ubuntuText  Description automatically generated |
| Passwd file **The first file is the passwd file. It is located, like most of config files, in** /etc. **Let's have a look on it.** | clear && cat /etc/passwd  **The structure of this file is as follows:**  username:password:UID:GID:description:homedir:shell |
| shadow **This file contains sensitive information about the user, like password, and other configurations.** | * **if password is set for the user, it is hashed and not retrievable from the file (well, let's say "not retrievable"... by staring on it. There are tools which allow us to crack passwords).** * \* **there is no password set (and very good, as we want to use passwordless approach with access keys, no passwords).** * ! **password was never set**. |
| group **The file** /etc/group **contains information about groups.** | clear && cat /etc/group  **The structure is as follows:**   * **group name** * **password** * **GID** * **users belong to the group** |
| gshadow **The last file** **is** /etc/gshadow | clear && cat /etc/gshadow  **This file contains encrypted passwords for groups**. |
| **to create user we can use one of two ways.** | whatis useradd whatis adduser |
| **Command** which **show us where the executable is located.** | which useradd  which adduser |
| useradd **is a binary, compiled with the system**. adduser **is a Perl script, which uses** useradd **in its backend. Another words,** adduser **uses** useradd v | file $(which useradd) file $(which adduser) |
| **we will use** adduser **to create final user.**  **It asks for password for user and other information.**  usedadd create user default | useradd testuser1  adduser testuser2 |
| ls -l /home  **testuser1 has the home directory defined, but it is not created.** |  |
| **we should consider to use during create of new user.** | -d - **create home directory in specified location, if we want to change**  -m - **create the home directory**  -p - **password**  -s - **provide shell**  -c - **comments, or description of the account** |
| **let's create our final user.** | useradd testuser3 -s /bin/bash -m |
| **Let's see all three accounts now.**  grep testuser /etc/passwd  grep testuser /etc/shadow  ls -l /home |  |
| **We will modify** testuser3 | grep testuser3 /etc/shadow  passwd testuser3 |
| **to modify the user.** | usermod testuser3 -g testuser2  grep testuser3 /etc/passwd  grep testuser3 /etc/group |
| **We can use GID also, not just name:** | usermod testuser3 -g 1001  grep testuser3 /etc/passwd  grep testuser3 /etc/group |
| **Let's come back to original setting** usermod testuser3 -g 1003 |  |
| **We just confirmed,** testuser3**is attached to** testuser3 **group only.** | usermod testuser3 -G testuser1  grep testuser3 /etc/group |
| **let's**add**one**more**.** | usermod testuser3 -G testuser2  grep testuser3 /etc/group |
| **We should have**two**secondary groups, not one!**  **let's correct this.** | usermod testuser3 -aG 1001  grep testuser3 /etc/group |
| **With** usermod -d **we can create a home directory** | usermod testuser3 -d /home/anotherdir  grep testuser3 /etc/passwd |
| ls -l /home **Hmm... There is no directory...**  **All right, let's come back** | usermod testuser3 -d /home/testuser3 |
|  | usermod testuser3 -d /home/newdir -m  grep testuser3 /etc/passwd  ls -l /home ok, we have a directory!  ls -l /home/newdir Yep, it works now. |
| **Let's modify the shell.** | usermod testuser3 -s /bin/sh  grep testuser3 /etc/passwd |
| **it is time to remove our user** | userdel testuser1 . |
| **The user itself, however, was removed.**  **In order to remove secondary groups, we can run** | grep testuser /etc/passwd  grep testuser /etc/group  usermod testuser3 -G ""  grep testuser /etc/group |
| **We can delele user now** | userdel testuser2 |
| ls -l /home  **hm... the home directory wasn't removed...** | In order to remove the files is a good practice to add two arguments:   * r - remove files * f - force removal (in case if files don't belongs to the user) |
| **So**,  userdel -rf testuser3  grep testuser /etc/passwd  ls -l /home |  |