



Desarrollo de Aplicaciones Web
Desarrollo de Aplicaciones Multiplataforma
CURSO 24-25



Bloque 4

Práctica 2: Gestión de Usuarios y Grupos en Linux (parte 2)

Instrucciones:

- Continúa la práctica en el contenedor de la práctica anterior.
- Duplica este archivo y añade capturas de pantalla que evidencien los pasos realizados.

Tareas:

4. Eliminación de Usuarios y Grupos

4.1. Elimina el usuario `servicio1` junto con todos sus archivos.

Lo primero es usar `id servicio1` para mostrar que existe. Después usamos `userdel -r servicio1` para borrar también su carpeta home. Si volvemos a usar `id servicio1` obtendremos “no such user”.

```
root@practicalbloque4donGuillermo:/# id servicio1 1
getent passwd servicio1
uid=100(servicio1) gid=65534(nogroup) groups=65534(nogroup)
servicio1:x:100:65534::/nonexistent:/usr/sbin/nologin 2
root@practicalbloque4donGuillermo:/# userdel -r servicio1
userdel: servicio1 mail spool (/var/mail/servicio1) not found
userdel: servicio1 home directory (/nonexistent) not found
root@practicalbloque4donGuillermo:/# id servicio1 3
id: 'servicio1': no such user
```

4.2. Elimina el grupo `docentes` solo si está vacío.

Si utilizamos `getent group docentes` veremos que no está vacío. Si quisiéramos borrarlo sería con `groupdel docentes`.

```
id: 'servicio1': no such user
root@practicalbloque4donGuillermo:/# getent group docentes
docentes:x:1002:usuario3,usuario2,usuario4
root@practicalbloque4donGuillermo:/# groupdel docentes
```



4.3. Elimina el usuario **usuario2** haciendo que no quede rastro de él. (Quizás requieras instalar alguna utilidad)

Si usamos **userdel -r usuario2** borraremos la carpeta junto con el contenido de su carpeta home. Pero tenemos que asegurarnos de que no quede rastro.

```
root@practicalbloque4donGuillermo:/# userdel -r usuario2
userdel: group usuario2 not removed because it is not the primary group of user usuario2.
userdel: usuario2 mail spool (/var/mail/usuario2) not found
userdel: usuario2 home directory (/home/usuario2) not found
```

Usando **grep usuario2 /etc/passwd /etc/shadow /etc/group** buscaremos entre tres distintas carpetas dentro de **etc** que podrían contener rastro de usuario2, y veremos que queda mención de usuario2 en la carpeta **group**:

```
root@practicalbloque4donGuillermo:/# grep usuario2 /etc/passwd /etc/shadow /etc/group
/etc/group:usuario2:x:1001:
```

Usamos **nano /etc/group** para modificar la carpeta **group**:

```
root@practicalbloque4donGuillermo:/# nano /etc/group
```

Veremos que existe una mención a **usuario2**, la cual eliminaremos.

```
users:x:100:usuario3,usuario4,usuarioNuevo1
nogroup:x:65534:
usuario1:x:1000:
usuario2:x:1001:
usuario3:x:1003:
usuario4:x:2025:
estudiantes:x:2026:usuarioNuevo1
```



4.4. Realiza un script que realice una verificación del paso anterior y compruebe que efectivamente no queda rastro de **usuario2**.

Usamos **nano script.sh**.

```
GNU nano 7.2 script.sh *
#!/bin/bash

# Verificar si el usuario existe en /etc/passwd
if grep -q "^usuario2:" /etc/passwd; then
    echo "Error: La entrada de usuario2 sigue existiendo en /etc/passwd."
    exit 1
fi

# Verificar si el usuario existe en /etc/shadow
if grep -q "^usuario2:" /etc/shadow; then
    echo "Error: La entrada de usuario2 sigue existiendo en /etc/shadow."
    exit 1
fi

# Verificar si el usuario existe en /etc/group
if grep -q ".*:usuario2:" /etc/group; then
    echo "Error: El usuario2 todav a pertenece a alg n grupo en /etc/group."
    exit 1
fi

# Verificar si hay archivos propiedad de usuario2
if find / -user usuario2 2>/dev/null | grep -q .; then
    echo "Error: Hay archivos propiedad de usuario2 en el sistema."
    exit 1
fi

# Verificar si hay procesos en ejecucion del usuario2
if pgrep -u usuario2 > /dev/null; then
    echo "Error: Hay procesos en ejecuci n del usuario2."
    exit 1
fi

echo "Todo correcto: No queda rastro del usuario2."
```

Ahora usamos **chmod +x script.sh** para obtener los permisos para ejecutarlo.

Usamos **./script.sh** para ejecutarlo, y vemos que el mensaje programado en caso de no quedar rastro de **usuario2** se activa.

```
root@practica1bloque4donGuillermo:/# chmod +x script.sh
root@practica1bloque4donGuillermo:/# ./script.sh
./script.sh: line 28: pgrep: command not found
Todo correcto: No queda rastro del usuario2.
root@practica1bloque4donGuillermo:/#
```




5. Información de Usuarios y Grupos

5.1. Muestra el UID y los grupos a los que pertenece **usuario4**.

Podemos usar **id usuario4** para ver su UID y los grupos a los que pertenece o **id -u usuario4 && id -nG usuario4** para mostrar solamente los grupos.

```
root@practicalbloque4donGuillermo:/# id usuario4
uid=2025(usuario4) gid=2025(usuario4) groups=2025(usuario4),27(sudo),100(users),2027(docentes)
root@practicalbloque4donGuillermo:/# id -u usuario4 && id -nG usuario4
2025
usuario4 sudo users docentes
```

5.2. Muestra únicamente los nombres de los grupos a los que pertenece **usuario3**.

Usamos **id -nG usuario3 | awk 1{\$1=""; print \$0}' | xargs** para mostrar SOLAMENTE los nombres de los grupos. Si no usamos la función **awk** nos aparecerá primero el nombre de **usuario3**.

```
root@practicalbloque4donGuillermo:/# id -nG usuario3 | awk '1{$1=""; print $0}' | xargs
users docentes
```

5.3. Identifica qué usuario está ejecutando los comandos.

Para esto usamos **whoami**. El usuario es **root**.

```
root@practicalbloque4donGuillermo:/# whoami
root
```

6. Propietarios y Permisos

6.1. Crea un archivo llamado **tarea2bloque4.txt** y cambia su propietario a **usuarioNuevo1** y su grupo a **docentes**.

touch tarea2bloque4.txt

chown usuarioNuevo1 tarea2bloque4.txt

chgrp docentes tarea2bloque4.txt



6.2. Crea la siguiente estructura de directorios:

- /home/estudiantes/dir_1
- /home/estudiantes/dir_2
- /home/estudiantes/dir_1/dir_3

Cambia el grupo propietario del directorio `/home/estudiantes` y sus subdirectorios a `docentes`.

`mkdir -p /home/estudiantes/dir_1`

`/home/estudiantes/dir_2`

`/home/estudiantes/dir_1/dir_3`

6.3. Cambia los permisos del archivo `tarea2bloque4.txt` para que solo el propietario pueda leer y escribir en él.

`chgrp -R docentes /home/estudiantes`

`chmod 600 tarea2bloque4.txt`

Usamos `ls -l tarea2bloque4.txt` para comprobar que solo el propietario puede leer y escribir (`-rw----- 1`)

6.4. Verifica quién es el propietario y el grupo del archivo `tarea2bloque4.txt`.

Usamos **`stat tarea2bloque4.txt`** para ver toda la información completa, el propietario es **`usuarioNuevo1`** y el grupo es **`docentes`**

```
root@practicalbloque4donGuillermo:/# touch tarea2bloque4.txt
root@practicalbloque4donGuillermo:/# chown usuarioNuevo1 tarea2bloque4.txt
root@practicalbloque4donGuillermo:/# chgrp docentes tarea2bloque4.txt
root@practicalbloque4donGuillermo:/# mkdir -p /home/estudiantes/dir_1 /home/estudiantes/dir_2 /home/estudiantes/dir_1/dir_3
root@practicalbloque4donGuillermo:/# chgrp -R docentes /home/estudiantes
root@practicalbloque4donGuillermo:/# chmod 600 tarea2bloque4.txt
root@practicalbloque4donGuillermo:/# ls -l tarea2bloque4.txt
-rw----- 1 usuarioNuevo1 docentes 0 Feb 18 20:34 tarea2bloque4.txt
root@practicalbloque4donGuillermo:/# ls -l tarea2bloque4.txt
-rw----- 1 usuarioNuevo1 docentes 0 Feb 18 20:34 tarea2bloque4.txt
root@practicalbloque4donGuillermo:/# stat tarea2bloque4.txt
  File: tarea2bloque4.txt
  Size: 0          Blocks: 0          IO Block: 4096   regular empty file
Device: 0,101   Inode: 11118         Links: 1
Access: (0600/-rw-----)  Uid: ( 1000/usuarioNuevo1)   Gid: ( 2027/docentes)
Access: 2025-02-18 20:34:09.453974467 +0000
Modify: 2025-02-18 20:34:09.453974467 +0000
Change: 2025-02-18 20:34:55.936938531 +0000
Birth: 2025-02-18 20:34:09.443973564 +0000
root@practicalbloque4donGuillermo:/#
```