

REPASO DE LINUX. CAPÍTULO 4: GRUPOS, USUARIOS Y PERMISOS

REPASO DE LINUX. CAPÍTULO 4: GRUPOS, USUARIOS Y PERMISOS

¿Qué es el superusuario?

Permisos

¿Quiénes somos?

Comando `whoami`

Comando `groups`

Comandos para gestión de grupos (`groupadd`, `groupdel`, `groupmod`)

Comandos para gestión de usuarios (`adduser`, `userdel`, `usermod`)

Comandos para cambio de grupo y de dueño (`chown`, `chgrp`)

Cambio de privilegios (comando `chmod`)

Ejercicios

¿Qué es el superusuario?

El **superusuario**, **administrador del sistema** o simplemente el `root` es un usuario especial que tiene privilegios para cambiar la configuración, borrar y crear ficheros en cualquier directorio, crear nuevos grupos y usuarios, etc.

ES PELIGROSO TRABAJAR COMO SUPERUSUARIO porque se puede dañar el sistema de forma irreversible. El lector debe estar seguro de lo que hace cuando trabaje como superusuario.

```
$ sudo touch /etc/prueba.txt
```

```
alejandra@cliente24:~$ touch /etc/prueba.txt
touch: no se puede efectuar `touch' sobre '/etc/prueba.txt': Permiso denegado
alejandra@cliente24:~$ sudo touch /etc/prueba.txt
[sudo] contraseña para alejandra:
alejandra@cliente24:~$
```

Se intenta crear el fichero `prueba.txt` en `/etc` como usuario normal y se obtiene un error de "Permiso denegado", lo que quiere decir que un usuario sin privilegios no puede hacer eso. A continuación, se intenta como administrador usando el comando `sudo`, esta vez sí se consigue.

Permisos

La información sobre grupos, usuarios y permisos se obtiene mediante el comando `ls` junto con la opción `-l`. Vamos a ver los permisos que tiene establecidos el fichero `whatis` que se encuentra en el directorio `/usr/bin`:

```
$ ls -l /usr/bin/what
```

```
alejandra@cliente24:~$ ls -l /usr/bin/what
-rwxr-xr-x 1 root root 48416 abr  8 2024 /usr/bin/what
```

En la primera columna aparecen los permisos, en la tercera se indica el usuario y en la cuarta aparece el nombre del grupo. A continuación se muestra qué significan los caracteres de la primera columna:

-	r	w	x	r	-	x	r	-	x
Tipo de fichero.	Permisos para el dueño del fichero.			Permisos para el grupo al que pertenece el fichero.			Permisos para el resto de usuarios		

r	Permiso de lectura .
w	Permiso de escritura .
x	Permiso de ejecución .

El tipo de fichero se indica en la siguiente tabla:

Tipo de fichero	
l	Enlace simbólico.
c	Dispositivo especial de caracteres.
b	Dispositivo especial de bloques.
p	FIFO (estructura de datos).
s	Socket (comunicaciones).
-	Ninguno de los anteriores. Puede ser un fichero de texto, un binario, etc.

¿Quiénes somos?

Comando `whoami`

Antes de empezar a crear usuarios, grupos y cambiar permisos, debemos saber quiénes somos y a qué grupo/grupos pertenecemos. En principio entraremos al sistema como un determinado usuario, pero podemos utilizar `su` para ejecutar comandos como otro usuario distinto, siempre y cuando sepamos la contraseña de ese otro usuario:

```
$ whoami
$ su usuario
```

```
alejandra@cliente24:~$ whoami
alejandra
```

Para volver al usuario original basta con utilizar el comando `exit`.

```
$ exit
```

Comando `groups`

Se puede ver a qué grupo pertenecemos.

```
$ groups
```

```
alejandra@cliente24:~$ groups
alejandra adm cdrom sudo dip plugdev users lpadmin vboxsf
```

Se pueden especificar uno o más usuarios detrás de `groups`. Eso nos dirá a qué grupos pertenece cada uno de ellos.

```
$ groups usuario root
```

Comandos para gestión de grupos (`groupadd`, `groupdel`, `groupmod`)

Los comandos `groupadd`, `groupdel` y `groupmod` permiten crear, borrar y modificar grupos respectivamente.

Vamos a crear los grupos `oficina_malaga`, `oficina_jaen` y `oficina_madrid`:

```
$ sudo groupadd oficina_malaga
$ sudo groupadd oficina_jaen
$ sudo groupadd oficina_madrid
```

```
alejandra@cliente24:~$ sudo groupadd oficina_malaga
[sudo] contraseña para alejandra:
alejandra@cliente24:~$ sudo groupadd oficina_jaen
alejandra@cliente24:~$ sudo groupadd oficina_madrid
```

Si se escribiese mal el nombre de un grupo, se puede solucionar con el comando `groupmod`:

```
$ sudo groupmod -n oficina_madrit oficina_madrid
```

```
alejandra@cliente24:~$ sudo groupmod -n oficina_madrit oficina_madrid
```

Se puede borrar con `groupdel`:

```
$ sudo groupdel oficina_madrit
```

```
alejandra@cliente24:~$ sudo groupdel oficina_madrit
```

Comandos para gestión de usuarios (`adduser`, `userdel`, `usermod`)

Igual que en la gestión de grupos se exige que los comandos se ejecuten con privilegios de administrador del sistema. Se puede escribir `sudo` antes de cada comando o:

```
$ sudo bash
```

Ahora se mostrará un carácter `#` en lugar de `$`. Para volver al usuario inicial se hace con el comando `exit`.

```
# adduser pedro --ingroup oficina_malaga
# adduser ana --ingroup oficina_malaga
# adduser berta --ingroup oficina_jaen
# adduser laura --ingroup oficina_malaga
# adduser laura oficina_jaen
```

Se han creado los usuarios y se han incluido dentro de sus grupos correspondientes al mismo tiempo. **Laura** pertenece a dos grupos. Al crear los usuarios se piden las claves, éstas se pueden cambiar con el comando **passwd**:

```
# passwd pedro
# passwd ana
# passwd laura
```

Comandos para cambio de grupo y de dueño (**chown**, **chgrp**)

Tanto **chown** como **chgrp** se pueden usar con la opción **-R** para cambiar el dueño o grupo en un directorio completo, de forma recursiva.

Cambio de privilegios (comando **chmod**)

Sirve para cambiar los permisos de uno o varios ficheros. Esos permisos se pueden ver con **ls -l**.

```
$ chmod +x hola_mundo.rb
```

En la siguiente tabla se muestran de forma esquemática los parámetros del comando **chmod**:

u	g	o	+ -	r	w	x
(user) dueño del fichero	(group) usuarios que pertenecen al mismo grupo	(others) el resto de usuarios	dar permiso quitar permiso	(read) lectura	(write) escritura	(execution) ejecución

Ahora vamos a quitar el permiso de ejecución para el resto de usuarios (others) y daremos permiso de escritura (write) a los usuarios del mismo grupo (group).

```
$ chmod o-x hola_mundo.rb
$ chmod g+w hola_mundo.rb
```

A este método, que utiliza los caracteres `rwX` se le denomina método simbólico. Podemos utilizar de forma análoga el método numérico.

4	2	1	Total
r	w	x	4 + 2 + 1 = 7
r	w	-	4 + 2 + 0 = 6
r	-	x	4 + 0 + 1 = 5
r	-	-	4 + 0 + 0 = 4
-	w	x	0 + 2 + 1 = 3
-	w	-	0 + 2 + 0 = 2
-	-	x	0 + 0 + 1 = 1

De esta forma, la siguiente línea:

```
$ chmod 755 hola_mundo.rb
```

Equivale a estas tres:

```
$ chmod u+rx hola_mundo.rb
$ chmod g+rx-w hola_mundo.rb
$ chmod o+rx-w hola_mundo.rb
```

Ejercicios

1. Completa la siguiente tabla:

Método numérico	Método simbólico
654	rw-r-xr--
766	rw-rw-rw-
777	rw-rw-rwx
520	r-x-w----
764	rw-rw-r--
440	r--r-----

2. Crea los grupos `oficina1` y `oficina2`.

```
$ sudo groupadd oficina1
$ sudo groupadd oficina2
```

```
alejandra@cliente24:~$ sudo groupadd oficina1
[sudo] contraseña para alejandra:
alejandra@cliente24:~$ sudo groupadd oficina2
```

3. Crea los usuarios `paco` y `pablo`. Estos usuarios deben pertenecer únicamente al grupo `oficina1`.

```
$ sudo useradd -m -g oficina1 paco
$ sudo useradd -m -g oficina1 pablo
```

```
alejandra@cliente24:~$ sudo useradd -m -g oficina1 paco
alejandra@cliente24:~$ sudo useradd -m -g oficina1 pablo
```

4. Crea los usuarios `alba` y `nerea`. Estos usuarios deben pertenecer únicamente al grupo `oficina2`.

```
$ sudo useradd -m -g oficina2 alba
$ sudo useradd -m -g oficina2 nerea
```

```
alejandra@cliente24:~$ sudo useradd -m -g oficina2 alba
alejandra@cliente24:~$ sudo useradd -m -g oficina2 nerea
```

5. Como usuario `paco` crea un fichero con nombre `topsecret.txt` en su directorio de trabajo al que únicamente él tenga acceso, tanto de lectura como de escritura.

```
$ su paco
$ cd
$ touch top_secret.txt
$ chmod 600 top_secret.txt
```

6. Crea otro fichero, también como usuario `paco`, con nombre `ventas_trimestre.txt` al que tengan acceso, tanto para leer como para escribir todos los usuarios que pertenezcan al mismo grupo. Se deben dejar los permisos que haya por defecto para el dueño y para el resto de usuarios. Comprueba como usuario `pablo` que puedes modificar el fichero.

```
$ touch ventas_trimestre.txt
$ chmod g+rw ventas_trimestre.txt
$ exit
$ su pablo
$ vi /home/paco/ventas_trimestre.txt
```

7. Como usuario `alba`, crea un fichero con nombre `empleados.txt` al que pueda acceder cualquier usuario para leer su contenido, y cualquier usuario del mismo grupo para leer o escribir.

```
$ exit
$ su alba
$ cd
$ touch empleados.txt
$ chmod 664 empleados.txt
```

8. Copia el fichero `empleados.txt` al directorio de trabajo de `alumno` (crea también el usuario `alumno` si no está creado). Cambia el propietario y el grupo al que pertenece el fichero, ahora debe ser `alumno`.

```
$ exit
$ sudo cp /home/alba/empleados.txt /home/alumno/
$ sudo chown alumno /home/alumno/empleados.txt
$ sudo chgrp alumno /home/alumno/empleados.txt
```


11. Como usuario `pablo`, copia un programa del directorio `/usr/bin` al directorio de trabajo con un nombre diferente. Por ejemplo `xclock` se puede copiar como `reloj`. Mira los permisos de este programa. Comprueba que se pueda ejecutar. Puede que sea necesario dar permiso para que otros usuarios distintos al actual puedan ejecutar aplicaciones en el entorno gráfico, basta con ejecutar como administrador: `xhost +`.

12. Cambia los permisos de `reloj` de tal forma que sólo lo pueda ejecutar el propietario del archivo.

```
$ chmod go-x reloj
```

13. Crea el usuario `modesto`, perteneciente a `oficina2`. Dentro de su directorio de trabajo, crea un directorio de nombre `compartido_con_todos`.

```
$ exit
$ sudo adduser modesto --ingroup oficina2
$ su modesto
$ cd
$ mkdir compartido_con_todos
```

14. Cambia de usuario en el entorno gráfico (botón **salir** y botón **cambiar de usuario**) y entra como `modesto`. Crea con **OpenOffice.org Calc** los ficheros `telefono_contactos.ods`, `gastos_marzo.ods` y `compartido_con_todos`. Se puede acceder al programa **Calc** mediante Aplicaciones -> Oficina -> OpenOffice.org Cal Hoja de cálculo

15. Da permiso de lectura a la carpeta `compartido_con_todos` y a todos los ficheros que contenga para todos los usuarios.

```
$ chmod -R a+r compartido_con_todos
```

16. Restringe el acceso de escritura sobre el fichero `telefono_contactos` para que sólo lo puedan modificar los usuarios del grupo al que pertenece su propietario.

```
$ cd compartido_con_todos
$ chmod g+w telefono_contactos.ods
$ chmod o-w telefono_contactos.ods (en realidad esta línea sería redundante)
```

17. Cambia los permisos de `gastos_marzo` para que sólo pueda modificarlo su propietario y leerlo cualquiera del mismo grupo.

```
$ chmod 640 gastos_marzo.ods
```

18. Cambia los permisos de `sueldos` para que sólo su dueño tenga acceso a él, tanto para lectura como para escritura.

```
$ chmod 600 sueldos.ods
```

19. Si un usuario tiene permiso de lectura sobre un fichero pero ese fichero se encuentra dentro de un directorio sobre el que no tiene permiso de lectura, ¿podría leer el fichero?, haz la prueba. No. Un usuario sin privilegios de lectura sobre un directorio no puede acceder a ficheros contenidos en ese directorio, aunque esos ficheros tengan todos los permisos activados.