GESTIÓN DE USUARIOS

Linux es un sistema multiusuario, por lo tanto, la tarea de añadir, modificar, eliminar y en general administrar usuarios se convierte en algo no solo rutinario, sino importante, además de ser un elemento de seguridad que mal administrado o tomado a la ligera, puede convertirse en un enorme hoyo de seguridad.

Los usuarios en Unix/Linux se identifican por un número único de usuario, User ID, UID. Y pertenecen a un grupo principal de usuario, identificado también por un número único de grupo, Group ID, GID. El usuario puede pertenecer a más grupos además del principal.

Aunque sujeto a cierta polémica, es posible identificar tres tipos de usuarios en Linux:

Usuario root, también llamado superusuario o administrador. Se caracteriza por:

- Su UID (User ID) es 0 (cero).
- Es la única cuenta de usuario con privilegios sobre todo el sistema.
- Acceso total a todos los archivos y directorios con independencia de propietarios y permisos.
- Controla la administración de cuentas de usuarios.
- Ejecuta tareas de mantenimiento del sistema.
- Puede detener el sistema.
- Instala software en el sistema.
- Puede modificar o reconfigurar el kernel, controladores, etc.

Usuarios especiales. Por ejemplos bin, daemon, adm, lp, sync, shutdown, mail, operator, squid, apache, etc. Se caracterizan por:

- Se les llama también cuentas del sistema.
- No tiene todos los privilegios del usuario root, pero dependiendo de la cuenta asumen distintos privilegios de root.
- Lo anterior para proteger al sistema de posibles formas de vulnerar la seguridad.
- No tienen contraseñas pues son cuentas que no están diseñadas para iniciar sesiones con ellas.
- También se les conoce como cuentas de "no inicio de sesión" (nologin).
- Se crean (generalmente) automáticamente al momento de la instalación de Linux o de la aplicación.
- Generalmente se les asigna un UID entre 1 y 100 (definido en /etc/login.defs)

Usuarios normales. Se usan para usuarios individuales y se caracterizan por:

- Cada usuario dispone de un directorio de trabajo, ubicado generalmente en /home.
- Cada usuario puede personalizar su entorno de trabajo.
- Tienen solo privilegios completos en su directorio de trabajo o home.

- Por seguridad, es siempre mejor trabajar como un usuario normal en vez del usuario root, y cuando se requiera hacer uso de comandos solo de root, utilizar el comando su.
- En las distros actuales de Linux se les asigna generalmente un UID superior a 500.

Superusuario en Ubuntu

El superusuario o root tiene todos los permisos por lo que podríamos loguearnos como superusuario para poder realizar todas las operaciones que necesitemos.

Podemos hacerlo, pero lo normal es no *loguearnos* como usuario root, sino emplear nuestro usuario (con el que hemos realizado la instalación) y preceder las instrucciones con el comando sudo (de super user do).

La primera vez que empleemos sudo en cada sesión de la terminal nos solicitará nuestra contraseña de usuario.

Si por algún motivo especial queremos emplear el usuario root en primer lugar debemos establecer su contraseña, para lo cual escribimos:

```
sudo passwd
>introducimos la contraseña de nuestro usuario
>introducimos la nueva contraseña de root
>confirmamos la nueva contraseña de root
```

Ahora para iniciar sesión como usuario root tecleamos

```
su root
```

y tecleamos su contraseña

Para salir de root usamos

```
exit
```

Si queremos deshabilitar root (mejor dicho su contraseña) empleamos:

```
sudo passwd -dl root
```

Otra opción en Ubuntu para trabajar como root es utilizar el comando:

```
sudo bash
```

Creación de usuarios.

adduser es el comando más sencillo para añadir nuevos usuarios al sistema desde la línea de comandos.

Ahora bien, realmente no hay prácticamente necesidad de indicar ninguna opción ya que si hacemos lo siguiente:

```
sudo adduser luis
```

E introducimos su contraseña.

Se creará el usuario y su grupo (con el mismo nombre), así como las entradas correspondientes en /etc/passwd, /etc/shadow y /etc/group.

También se creará el directorio de inicio o de trabajo: /home/luis y los archivos de configuración que van dentro de este directorio y que más adelante se detallan.

Nota: En realidad adduser es un enlace simbólico a useradd, comando de bajo nivel para crear usuarios. Si usamos useradd, debemos añadir la opción –m para que cree su directorio en home y añadir a posteriori su contraseña con passwd. (para ver todas las opciones de useradd, teclear man useradd).

Para cambiar de usuario desde la terminal usamos:

```
su luis
```

y vemos cómo cambia el prompt, aunque en el entorno gráfico el usuario se mantiene.

Para ver todos los usuarios podemos ver el archivo:

```
cat /etc/passwd
```

De la misma forma podemos ver los grupos con:

```
cat /etc/group
```

Nota: Una forma sencilla para ver todos los usuarios es lanzar un ls –l a home:

```
cd /home/
ls –l
```

la tercera y cuarta columna del listado serán el usuario propietario del directorio y el grupo.

Para saber el grupo de un usuario, usamos groups, por ejemplo:

groups luis

Crear usuario con otro grupo

sudo adduser nombreusuario --ingroup nombregrupo

Y para crear grupos usamos:

sudo addgroup nombregrupo

por ejemplo:

sudo addgroup usuarios

Por tanto:

sudo adduser lucia --ingroup usuarios

Cambiar contraseñas con passwd

sudo passwd luis

Las fechas de expiración de contraseña, etc. se quedan lo más amplias posibles así que no hay problema que la cuenta caduque.

El superusuario es el único que puede indicar el cambio o asignación de contraseñas de cualquier usuario. Los usuarios normales pueden cambiar su contraseña en cualquier momento con tan solo invocar passwd sin argumentos, y podrá de esta manera cambiar la contraseña cuantas veces lo requiera, pero solo de sí mismo.

passwd tiene integrado validación de contraseñas comunes, cortas, de diccionario, etc.

Modificación de usuarios

usermod permite modificar o actualizar un usuario o cuenta ya existente. Sus opciones más comunes o importantes son las siguientes:

-c añade o modifica el comentario
-d modifica el directorio de trabajo o home del usuario
-e cambia o establece la fecha de expiración de la cuenta, formato AAAA-MM-DD
-g cambia el número de grupo principal del usuario (GID)
-G establece otros grupos a los que puede pertenecer el usuario
-l cambia el login o nombre del usuario
-L bloquea la cuenta del usuario, no permitiéndole que ingrese al sistema. No borra ni cambia nada del usuario, solo lo deshabilita.
-s cambia el shell por defecto del usuario cuando ingrese al sistema.

-u cambia el UID del usuario.
-U desbloquea una cuenta previamente bloqueada con la opción -L.

Por ejemplo podemos cambiar el nombre de un usuario:

usermod luis –l luigi // Exige estar logueado como root

Cambiar su directorio home;

mkdir usuarioluigi
usermod –d usuarioluigi -m luigi

Cambiar usuario de grupo

sudo usermod -g usuarios luis

Si quisiéramos que tuviera permisos de superusuario lo añadimos al grupo sudo:

sudo usermod –g sudo luis

Para modificar el nombre de un grupo usamos:

sudo groupmod -n users usuarios

Eliminación de usuarios

Como su nombre lo indica, userdel elimina una cuenta del sistema, userdel puede ser invocado de tres maneras:

sudo userdel luis

Sin opciones elimina la cuenta del usuario de /etc/passwd y de /etc/shadow, pero no elimina su directorio de trabajo ni archivos contenidos en el mismo, esta es la mejor opción, ya que elimina la cuenta pero no la información de la misma.

sudo userdel -r luis

Al igual que lo anterior elimina la cuenta totalmente, pero con la opción -r además elimina su directorio de trabajo y archivos y directorios contenidos en el mismo, así como su buzón de correo, si es que estuvieran configuradas las opciones de correo. La cuenta no se podrá eliminar si el usuario está logueado o en el sistema al momento de ejecutar el comando.

sudo userdel -f luis

La opción -f es igual que la opción -r, elimina todo lo del usuario, cuenta, directorios y archivos del usuario, pero además lo hace sin importar si el usuario está actualmente en el

sistema trabajando. Es una opción muy radical, además de que podría causar inestabilidad en el sistema, así que hay que usarla solo en casos muy extremos.

Para eliminar un grupo empleamos:

delgroup nombredelgrupo

IDENTIDADES DE FICHERO

En Linux, los ficheros o directorios tienen 3 tipos de identidades y por tanto tres perfiles.

Perfil propietario
Perfil grupo o grupo asociado
Perfil sistema

Permisos de ficheros y directorios

A cada uno de estos perfiles se les puede asociar unas normas de utilización que son conocidas como los permisos.

Los permisos tienen tres elementos lectura, escritura y ejecución, de tal forma que con la combinación de perfiles y permisos se puede establecer un sistema de administración y control de ficheros y directorios muy eficiente.

En el caso de los ficheros:

- Permiso de lectura (read)
  Si tienes permiso de lectura de un archivo, puedes ver su contenido.

- Permiso de escritura (write)
  Si tienes permiso de escritura de un archivo, puedes modificar el archivo. Puedes agregar, sobrescribir o borrar su contenido.

- Permiso de ejecución (execute)
  Si el archivo tiene permiso de ejecución, entonces puedes decirle al sistema operativo que lo ejecute como si fuera un programa.

En el caso de los directorios:

- Permiso de lectura en un directorio.
  Si un directorio tiene permiso de lectura, puedes ver los archivos que este contiene. Puedes usar un ls para ver su contenido, que tengas permiso de lectura en un directorio no quiere decir que puedas leer el contenido de sus archivos si no tienes permiso de lectura en esos.

- Permiso de escritura en un directorio.

Con el permiso de escritura puedes agregar, remover o mover archivos al directorio

· Permiso de ejecución en un directorio.
Ejecución permite usar el nombre del directorio cuando se acce a archivos en ese directorio, es decir este permiso lo hace que se tome en cuenta en búsquedas realizadas por un programa, por ejemplo, un directorio sin permiso de ejecución no sería revisado por el comando find

Los permisos de lectura, escritura y ejecución se representan con r, w y x respectivamente.

Veámoslo de manera práctica. Cuando en un directorio tecleamos
ls –l

obtenemos para cada directorio o archivo una primera columna con una serie de caracteres relacionados con los permisos:

-rw-rw-r--

El primer carácter representa el tipo de elemento y cómo se interpreta en Linux

d	directorio
l	enlace simbólico
c	dispositivo especial de caracteres
p	canal
s	socket
-	fichero estándar

Después tenemos un primer bloque de 3 caracteres. Este primer grupo contiene los permisos asociados al propietario del fichero.

El siguiente bloque de 3 caracteres contiene los permisos asociados al grupo y el último y tercer bloque de 3 caracteres los permisos asociados al sistema (por tanto a cualquier usuario).

Cada bloque tiene 3 caracteres que determinan la ausencia o presencia del permiso con la secuencia

rwx todos los permisos

cuando no tenga alguno de los permisos o ninguno de ellos se sustituye cada uno de ellos por guión medio, por ejemplo:

r-- solo permiso de escritura

--- ningún permiso

Para cambiar los permisos empleamos el comando chmod (change mode), que pueden agregar o remover permisos a uno o más archivos con + (mas), – (menos) e = (igual).

Por ejemplo, creamos un archivo:

```
touch saludo
nano saludo
echo “¡Hola Mundo!”
```

Y quitarle permisos de escritura con

```
chmod –w saludo
```

le quitará el permiso de escritura a ese archivo. Comprobamos cómo al cambiar datos mediante el editor nano no nos deja guardar los cambios.

Podemos ahora añadir permisos de ejecución con:

```
chmod +x saludo
```

Ahora comprobaremos como se puede ejecutar.

```
./saludo
```

El comando chmod también incluye la opción = para establecer los permisos en combinación exacta, esto es, eliminará los que no se incluyan, por ejemplo:

```
chmod =w archivo
```

Dejará solamente permisos de lectura.

En los casos anteriores sin especificar los perfiles, si queremos modificar los de grupo o los de sistema, emplearemos las opciones u (usuario), g (grupo) y o (otros).

Por ejemplo, podemos dejar que solo el usuario dueño del archivo pueda ejecutar. Primero eliminamos el permiso a todos:

```
chmod -x saludo
```

Y añadimos al usuario

```
chmod u+x saludo
```

Ahora podemos comprobar como otro usuario, aunque esté en el mismo grupo no puede ejecutar el archivo.

Vamos a crear 1 grupo y 2 usuarios en el mismo:

sudo addgroup ventas
sudo adduser fernando --ingroup ventas
sudo adduser gema --ingroup ventas

Creamos de nuevo un archivo similar al anterior y si ahora le damos permisos de ejecución al grupo podrán ejecutarlos los usuarios del grupo pero no los que no pertenezcan a éste:

chmod g+x saludo

También podemos emplear con chmod el modo octal.

Como resultado de la combinación de los tres tipos de permisos (lectura, escritura y ejecución), con las tres clases de usuarios (propietario, grupo y otros), se obtiene 2^3= permisos en total que pueden ser asignados o denegados de forma independiente.

La base 8 se utiliza habitualmente para que exista un dígito por cada combinación de permisos (un bit a modo de bandera por cada permiso, con valor 1 ó 0 según el permiso esté concedido o denegado).

rwx	7	Lectura, escritura y ejecución
rw-	6	Lectura, escritura
r-x	5	Lectura y ejecución
r--	4	Lectura
-wx	3	Escritura y ejecución
-w-	2	Escritura
--x	1	Ejecución
---	0	Sin permisos

Así, las posibles combinaciones se resumen en números octales de tres dígitos del 000 al 777, cada uno de los cuales permite establecer un tipo de permiso distinto a cada clase de usuario.

El primer dígito establece el tipo de permiso deseado al usuario; el segundo al grupo; y el tercero al resto de los usuarios.

De tal manera que podemos establecer los permisos con chmod seguido de la secuencia de números para cada perfil.

Por ejemplo:

chmod u=rwx,g=rwx,o=rx	chmod 775
chmod u=rwx,g=rx,o=	chmod 760
chmod u=rw,g=r,o=r	chmod 644
chmod u=rw,g=r,o=	chmod 640
chmod u=rw,go=	chmod 600
chmod u=rwx,go=	chmod 700

El comando chmod puede usar el flag -R para realizar los cambios de manera recursiva a archivos y directorios.

Podemos crear un conjunto de directorios y archivos en un usuario para cambiar sus permisos recursivamente:

```
mkdir ventas && cd ventas
mkdir previsiones && cd previsiones
touch trimestre.txt
cd ..
cd ..
chmod 700 -R ventas
```

Cambio de propietario de fichero

Como lo que determina el acceso a un fichero o directorio en Linux son sus permisos respecto a su propietario, grupo de propietario o sistema en general (cualquier usuario), puede ser necesario cambiar los propietarios de un archivo o directorio.

Para ello empleamos:

chown opciones nuevopropietario:nuevogrupo fichero

Por ejemplo

```
sudo chown gema:ventas ventas/previsiones/trimestre
```

O de manera recursiva,

```
sudo chown –R gema ventas
```