

Administración 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. En este manual aprenderás todo lo necesario para administrar completamente tus usuarios en GNU/Linux.

Tipos de usuarios

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.
- 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

- Ejemplos: bin, daemon, adm, lp, sync, shutdown, mail, operator, squid, apache, etc.
- 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.
- 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.

/etc/passwd

Cualquiera que sea el tipo de usuario, todas las cuentas se encuentran definidas en el archivo de configuración 'passwd', ubicado dentro del directorio /etc. Este archivo es de texto tipo ASCII, se crea al momento de la instalación con el usuario root y las cuentas especiales, más las cuentas de usuarios normales que se hayan indicado al momento de la instalación.

El archivo /etc/passwd contiene una línea para cada usuario, similar a las siguientes:

```
root:x:0:0:root:/root:/bin/bash
sergio:x:501:500:Sergio González:/home/sergio:/bin/bash
```

La información de cada usuario está dividida en 7 campos delimitados cada uno por ':' dos puntos.

/etc/passwd	
Campo 1	Es el nombre del usuario, identificador de inicio de sesión (login). Tiene que ser único.
Campo 2	La 'x' indica la contraseña encriptada del usuario, además también indica que se está haciendo uso del archivo /etc/shadow, si no se hace uso de este archivo, este campo se vería algo así como: 'ghy675gjuXCc12r5gt78uuu6R'.
Campo 3	Número de identificación del usuario (UID). Tiene que ser único. 0 para root, generalmente las cuentas o usuarios especiales se numeran del 1 al 100 y las de usuario normal del 101 en adelante, en las distribuciones mas recientes esta numeración comienza a partir del 500.
Campo 4	Numeración de identificación del grupo (GID). El que aparece es el número de grupo principal del usuario, pero puede pertenecer a otros, esto se configura en /etc/groups.
Campo 5	Comentarios o el nombre completo del usuario.
Campo 6	Directorio de trabajo (Home) donde se sitúa al usuario después del inicio de sesión.
Campo 7	Shell que va a utilizar el usuario de forma predeterminada.

/etc/shadow

Anteriormente (en sistemas Unix) las contraseñas cifradas se almacenaban en el mismo /etc/passwd. El problema es que 'passwd' es un archivo que puede ser leído por cualquier usuario del sistema, aunque solo puede ser modificado por root. Con cualquier computadora potente de hoy en día, un buen programa de descifrado de contraseñas y paciencia es posible "crackear" contraseñas débiles (por eso la conveniencia de cambiar periódicamente la contraseña de root y de otras cuentas importantes). El archivo 'shadow', resuelve el problema ya que solo puede ser leído por root. Considérese a 'shadow' como una extensión de 'passwd' ya que no solo almacena la contraseña encriptada, sino que tiene otros campos de control de contraseñas.

El archivo /etc/shadow contiene una línea para cada usuario, similar a las siguientes:

```
root:ghy675gjuXCc12r5gt78uuu6R:10568:0:99999:7:7:-1::
sergio:rfgf886DG778sDFFDRRu78asd:10568:0:-1:9:-1:-1::
```

La información de cada usuario está dividida en 9 campos delimitados cada uno por ':' dos puntos.

/etc/shadow	
Campo 1	Nombre de la cuenta del usuario.
Campo 2	Contraseña cifrada o encriptada, un '*' indica cuenta de 'nologin'.
Campo 3	Días transcurridos desde el 1/ene/1970 hasta la fecha en que la contraseña fue cambiada por última vez.
Campo 4	Número de días que deben transcurrir hasta que la contraseña se pueda volver a cambiar.
Campo 5	Número de días tras los cuales hay que cambiar la contraseña. (-1 significa nunca). A partir de este dato se obtiene la fecha de expiración de la contraseña.
Campo 6	Número de días antes de la expiración de la contraseña en que se le avisará al usuario al inicio de la sesión.
Campo 7	Días después de la expiración en que la contraseña se inhabilitara, si es que no se cambio.
Campo 8	Fecha de caducidad de la cuenta. Se expresa en días transcurridos desde el 1/Enero/1970 (epoch).
Campo 9	Reservado.

/etc/group

Este archivo guarda la relación de los grupos a los que pertenecen los usuarios del sistema, contiene una línea para cada usuario con tres o cuatro campos por usuario:

```
root:x:0:root
ana:x:501:
sergio:x:502:ventas,supervisores,produccion
cristina:x:503:ventas,sergio
```

- El campo 1 indica el usuario.
- El campo 2 'x' indica la contraseña del grupo, que no existe, si hubiera se mostraría un 'hash' encriptado.
- El campo 3 es el Group ID (GID) o identificación del grupo.
- El campo 4 es opcional e indica la lista de grupos a los que pertenece el usuario

Actualmente al crear al usuario con useradd se crea también automáticamente su grupo principal de trabajo GID, con el mismo nombre del usuario. Es decir, si se añade el usuario 'sergio' también se crea el /etc/group el grupo 'sergio'. Aun así, existen comandos de administración de grupos que se explicarán más adelante.

/etc/login.defs

En el archivo de configuración /etc/login.defs están definidas las variables que controlan los aspectos de la creación de usuarios y de los campos de shadow usadas por defecto. Algunos de los aspectos que controlan estas variables son:

- Número máximo de días que una contraseña es válida PASS_MAX_DAYS
- El número mínimo de caracteres en la contraseña PASS_MIN_LEN
- Valor mínimo para usuarios normales cuando se usa useradd UID_MIN
- El valor umask por defecto UMASK
- Si el comando useradd debe crear el directorio home por defecto CREATE_HOME

Basta con leer este archivo para conocer el resto de las variables que son autodescriptivas y ajustarlas al gusto. Recuérdese que se usaran principalmente al momento de crear o modificar usuarios con los comandos useradd y usermod que en breve se explicaran.

Añadir usuarios con useradd

useradd o adduser es el comando que permite añadir nuevos usuarios al sistema desde la línea de comandos. Sus opciones más comunes o importantes son las siguientes:

- -c añade un comentario al momento de crear al usuario, campo 5 de /etc/passwd
- -d directorio de trabajo o home del usuario, campo 6 de /etc/passwd
- -e fecha de expiración de la cuenta, formato AAAA-MM-DD, campo 8 de /etc/shadow
- -g número de grupo principal del usuario (GID), campo 4 de /etc/passwd
- -G otros grupos a los que puede pertenecer el usuario, separados por comas.
- -r crea una cuenta del sistema o especial, su UID será menor al definido en /etc/login.defs en la variable UID_MIN, además no se crea el directorio de inicio.
- -s shell por defecto del usuario cuando ingrese al sistema. Si no se especifica, bash, es el que queda establecido.
- -u UID del usuario, si no se indica esta opción, automáticamente se establece el siguiente número disponible a partir del último usuario creado.

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

```
#> useradd juan
```

Se creará el usuario y su grupo, así como las entradas correspondientes en /etc/passwd, /etc/shadow y /etc/group. También se creará el directorio de inicio o de trabajo: /home/juan y los archivos de configuración que van dentro de este directorio y que más adelante se detallan.

Las fechas de expiración de contraseña, etc. Quedan lo más amplias posibles así que no hay problema que la cuenta caduque, así que prácticamente lo único que faltaría sería añadir la contraseña del usuario y algún comentario o identificación de la cuenta. Como añadir el password o contraseña se estudiara en un momento y viendo las opciones con '-c' es posible establecer el comentario, campo 5 de /etc/passwd:

```
#> useradd -c "Juan Perez Hernandez" juan
```

Siempre el nombre del usuario es el último parámetro del comando. Así por ejemplo, si queremos salirnos del default, podemos establecer algo como lo siguiente:

```
#> useradd -d /usr/juan -s /bin/csh -u 800 -c "Juan Perez Hernandez" juan
```

Con lo anterior estamos cambiando su directorio de inicio, su shell por defaultl sera csh y su UID será el 800 en vez de que el sistema tome el siguiente número disponible.

Modificar usuarios con `usermod`

Como su nombre lo indica, `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, campo 5 de `/etc/passwd`
- `-d` modifica el directorio de trabajo o home del usuario, campo 6 de `/etc/passwd`
- `-e` cambia o establece la fecha de expiración de la cuenta, formato AAAA-MM-DD, campo 8 de `/etc/shadow`
- `-g` cambia el número de grupo principal del usuario (GID), campo 4 de `/etc/passwd`
- `-G` establece otros grupos a los que puede pertenecer el usuario, separados por comas.
- `-l` cambia el login o nombre del usuario, campo 1 de `/etc/passwd` y de `/etc/shadow`
- `-L` bloque 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`.

Si quisiéramos cambiar el nombre de usuario de 'sergio' a 'sego':

```
#> usermod -l sego sergio
```

Casi seguro también cambiará el nombre del directorio de inicio o HOME en `/home`, pero si no fuera así, entonces:

```
#> usermod -d /home/sego sego
```

Otros cambios o modificaciones en la misma cuenta:

```
#> usermod -c "supervisor de area" -s /bin/ksh -g 505 sego
```

Lo anterior modifica el comentario de la cuenta, su shell por defecto que ahora sera Korn shell y su grupo principal de usuario quedó establecido al GID 505 y todo esto se aplicó al usuario 'sego' que como se observa debe ser el último argumento del comando.

El usuario 'sego' salió de vacaciones y nos aseguramos de que nadie use su cuenta:

```
#> usermod -L sego
```

Eliminar usuarios con `userdel`

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

```
#> userdel sergio
```

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.

```
#> userdel -r sergio
```

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.

```
#> userdel -f sergio
```

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.

Cambiar contraseñas con `passwd`

Crear al usuario con `useradd` es el primer paso, el segundo es asignarle una contraseña a ese usuario. Esto se logra con el comando `passwd` que permitirá ingresar la contraseña y su verificación:

```
#> passwd sergio

Changing password for user prueba.

New UNIX password:

Retype new UNIX password:

passwd: all authentication tokens updated successfully.

#>
```

El usuario `root` es el único que puede indicar el cambio o asignación de contraseñas de cualquier usuario. 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.

`passwd` tiene integrado validación de contraseñas comunes, cortas, de diccionario, etc. así que si por ejemplo intento como usuario normal cambiar mi contraseña a 'qwerty' el sistema me mostrará lo siguiente:

```
$> passwd

Changing password for user prueba.

New UNIX password:
```

```
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
$>
```

Nótese que al ingresar 'qwerty' como contraseña se detectó que es una secuencia ya conocida como contraseña y me manda la advertencia: "BAD PASSWORD: it is based on a dictionary word", sin embargo me permite continuar, al ingresar la verificación. Es decir, passwd avisa de malas o débiles contraseñas pero permite establecerlas si realmente se desea.

Resumiendo entonces, se podría decir que todo este tutorial se reduce a dos líneas de comandos para crear y dejar listo para trabajar a un usuario en Linux:

```
#> useradd ana
#> passwd ana
```

Se crea el usuario 'ana', useradd hace todo el trabajo de establecer el shell, directorio de inicio, copiar archivos iniciales de configuración de la cuenta, etc. y después passwd establece la contraseña. Así de simple.

passwd tiene varias opciones que permiten bloquear la cuenta '-l', desbloquearla '-u', y varias opciones más que controlan la vigencia de la contraseña, es decir, es otro modo de establecer los valores de la cuenta en /etc/shadow. Para más información consulta las páginas del manual:

```
$> man passwd
```

Resumen de comandos y archivos de administración de usuarios

Existen varios comandos más que se usan muy poco en la administración de usuarios, que sin embargo permiten administrar aun más a detalle a tus usuarios de Linux. Algunos de estos comandos permiten hacer lo mismo que los comandos previamente vistos, solo que de otra manera, y otros como chpasswd y newusers resultan muy útiles y prácticos cuando de dar de alta a múltiples usuarios se trata.

A continuación te presento un resumen de los comandos y archivos vistos en este tutorial más otros que un poco de investigación

Comandos de administración y control de usuarios	
adduser	Ver useradd
chage	Permite cambiar o establecer parámetros de las fechas de control de la contraseña.
id	Muestra la identidad del usuario (UID) y los grupos a los que pertenece.
groupdel	Elimina grupos del sistema.
groupmod	Modifica grupos del sistema.
groups	Muestra los grupos a los que pertenece el usuario.
useradd	Añade usuarios al sistema (/etc/passwd).
userdel	Elimina usuarios del sistema.
usermod	Modifica usuarios.

Archivos de administración y control de usuarios	
/etc/group	Usuarios y sus grupos.
/etc/login.defs	Variables que controlan los aspectos de la creación de usuarios.
/etc/passwd	Usuarios del sistema.
/etc/shadow	Contraseñas encriptadas y control de fechas de usuarios del sistema.

El comando chown

El comando **chown** en linux (change owner) nos permite cambiar de propietario en archivos y directorios de linux. Hay diferentes formas de usar el comando. La más básica es:

chown *nuevousuario archivo1*

Por ejemplo:

chown *root /var/home/musica.mp3*

Establece como propietario del archivo musica.mp3 al usuario root. Para cambiar recursivamente el propietario a todos los archivos y subcarpetas, podemos usar:

chown *-R root /var/home*

Aclarar que el comando chown en linux, usado de forma recursiva modifica el propietario de los archivos y subdirectorios, dejando el directorio principal sin cambios de propietario. Si añadimos el modificador -c nos informará acerca de los cambios que haga, por ejemplo

chown *-R -c root /var/home*

Para cambiar el grupo además del propietario, podemos poner dos puntos despues del owner y a continuación añadir el grupo. Por ejemplo para cambiar de usuario y grupo a un archivo lo haríamos de la siguiente manera:

chown *web1:client1 /var/www/clients/client1/web1/robots.txt* En éste ejemplo el archivo

robots.txt pasará a tener como propietario *web1* y como grupo *client1*.

Para cambiar el usuario de los archivos y carpetas del directorio donde nos encontramos podemos usar:

chown *nuevousuario **

Si además queremos que lo haga de forma recursiva bajo todos los subdirectorios y sus archivos usaremos -R:

chown *-R nuevousuario **

También podemos además de usar el nombre de usuario y el nombre de grupo usar el formato numérico mediante el UID (identificador numérico del usuario) y el GID (identificador numérico de grupo) en el comando `chown`, por ejemplo:

```
chown -R 0:0 /etc
```

Asignará recursivamente a /etc y a sus subdirectorios y archivos el usuario con UID 0 y el grupo con GID 0 (usuario 0 = root y grupo 0=root)

El modificador `-v` dentro del comando `chown` nos dará información de los permisos aplicados.

El comando `fdisk` (y `mkfs`)

Fdisk (válido para fixed disk o format disk), es uno de los comandos más importantes que deberíamos conocer, puesto que en caso de no contar con interfaz gráfica de por medio (cosa muy habitual en las distribuciones destinadas a servidores por el mejor aprovechamiento de los recursos), nos ayudará enormemente en la **gestión y administración de nuestro espacio en disco**.

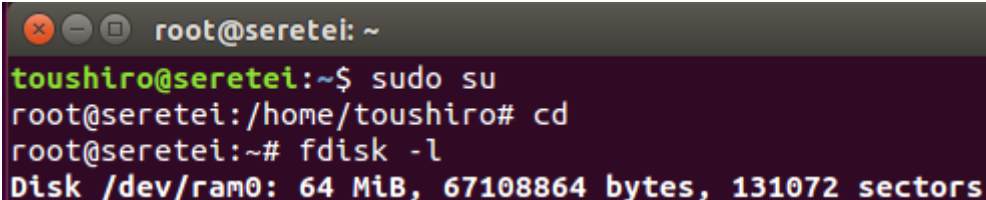
Con esta herramienta podremos **crear, eliminar, redimensionar, cambiar o copiar y mover particiones** usando el sencillo menú que ofrece. El límite que existe en esta herramienta está en **4 particiones primarias como máximo por disco**, y un número de particiones extendidas o lógicas que será variable en función del tamaño de nuestro disco duro.

A continuación veremos **algunos de los comandos más usados** para gestionar la tabla de particiones de un sistema **Linux**. Recordad que deberemos estar con **el usuario root o con algún usuario con permisos similares**, para no encontrarnos continuamente con errores de permisos o comandos no encontrados.

1. Ver todas las particiones.

Para listar todas las particiones existentes en nuestro sistema pasaremos el argumento `-l`, que hará que se listen ordenadas por el nombre del dispositivo.

```
fdisk -l
```



```
root@seretei: ~  
toushiro@seretei:~$ sudo su  
root@seretei:/home/toushiro# cd  
root@seretei:~# fdisk -l  
Disk /dev/ram0: 64 MiB, 67108864 bytes, 131072 sectors
```