

Administración / Gestión de usuarios y grupos en LINUX

Javier Fernández Rivera - www.aurea.es

Linux para la administración de usuarios maneja tres ficheros que se encuentran en el subdirectorio /etc:

/etc/passwd ----> Fichero en el que se encuentran las cuentas de usuarios y passwords.

/etc/group -----> Fichero que almacena los grupos.

/etc/shadow ----> Fichero (sombra) con función de seguridad para guardar las passwords de los usuarios.

Linux es un sistema multiusuario y como tal debe poder trabajar con distintos usuarios.

En Linux podemos manejar la cuenta de usuario root (en caso de saber la pass), es la cuenta llamada "superusuario". Con esta cuenta podemos tener acceso completo a todo el sistema operativo para su administración, configuración y gestión. Aunque lo lógico sería en linux trabajar con una cuenta aparte de usuario.

Resumiendo, en linux aunque seamos un solo usuario usando este sistema lo lógico y bueno sería usar una cuenta de usuario creada anteriormente con el root (superusuario) y solo acceder a la cuenta del root cuando se precise importante para realizar algún cambio.

GESTION DE USUARIOS

Información sobre las cuentas de usuario

Nombre de usuario: es el nombre de usuario que nosotros le damos a la cuenta (por ejemplo: Quasi pukii, loco_bob), este campo es mas bien orientado o dedicado al usuario o administrador, para facilitar a estos la tarea de identificación de un usuario. El SO no contempla este campo (es mera información implementada), identifica a los usuarios mediante el USER ID (identificador de usuario).

USER ID: (GUI="GROUP-ID"); es la identificación (un numero) del grupo por defecto del usuario.

Clave: (UID="USER- ID"); como ya anticipaba, este es un numero identificativo a cada usuario. Mediante este numero el SO sigue la pista al usuario. Cada usuario tiene un numero distinto.

Group ID: Linux como buen sistema operativo de red y seguro mantiene la clave de cada usuario. Esta clave se encuentra encriptada. Con el comando "passwd" podemos modificar dicha clave.

Nombre completo: Sería el nombre completo y real del usuario. Nombre y apellidos. Este campo también es manejado por el SO como mera información llana de un usuario. Ejemplo: Javier Fernández Rivera.

Directorio inicial: este campo almacena el directorio donde el SO debería colocar al usuario nada mas conectar. Generalmente se mantiene la estructura de un directorio llamado "home" y de este cuelgan los directorios de cada usuario (cada directorio suele llamarse de la misma forma que el nombre de usuario en la cuenta).

Interprete de inicio: este campo guarda la información del interprete que debe ejecutar el SO linux cuando el usuario se conecte.

Un usuario al conectarse debe aparecerle una interface para poder interactuar con el sistema operativo, MSDOS (shell de DOS), Linux (shell de Linux), windows (GUI windows). Podemos especificar en este campo: /bin/bash y /bin/tcsh.

Dentro del directorio bin (binario) se encuentra los ejecutables del SO Linux, en el ejecutamos el bash (interprete de comandos usual).

En el fichero passwd localizado en el directorio etc (/etc/passwd) se encuentra toda esta información almacenada.

Cada línea pertenece a un usuario, y cada línea posee estos campos de información.

El formato que se sigue es:

nombre:clave encriptada:UID:GID:nombrecompleto:dir.inicio:intérprete

Un ejemplo sería:

Quasii:Xv8Q981g71oKK:105:100:Javier_Fernandez_Rivera:/home/quasi:/bin/bash

Fijémonos que entre campo y campo los caracteres de separación son los dos puntos.

El primer campo (Quasi) es el nombre de la cuenta de usuario, luego se encuentra la clave (encriptada), en algunas distribuciones de linux se amplia la seguridad de la clave con el uso de claves en sombra, podríamos observarlo en el fichero (/etc/shadow), a continuación el identificador del usuario (con el trabaja el SO) y so identificador de grupo por defecto, mas

adelante tenemos al nombre completo (Javier_Fernandez_Rivera), y ya por ultimo el directorio quasi dentro del home, sera su directorio por defecto, y el interprete de comandos de linux.

Añadiendo usuarios

Comando: useradd

Etimología: Añadir usuario

Sintaxis: useradd [-c][d][e][g/G][m][p][s][u]opciones <nombreCuentaUsuario>

-c: comentario o nombre completo

-d: directorio home por defecto.

-e: fecha en la que se deshabilita la cuenta YYYY-MM-DD

-g: grupo inicial , tiene que haber sido creado ya.

-G: Lista de grupos suplementarios separados por comas y sin espacios en blanco a la que también pertenece el usuario.

-m: Crea el directorio home y copia los ficheros contenidos en /etc/skel (ficheros de inicialización).

-p: clave de entrada

-s: nombre y ruta del shell

-u: nombre numérico que identifica al usuario. Debe de ser único si no se especifica la opción o. Del 0-99 están reservados.

Ejemplo:

useradd -c Javier_Fernández_Rivera -d /home/quasi -u 505 -g 105 -p kak0ta -s /bin/bash Quasi

En este ejemplo vemos como se crea una cuenta como (Quasi).

Se ha especificado el parámetro -c que incluye un comentario, en este caso seria el nombre y los apellidos de Quasi.

A continuación -d especifica el directorio home por defecto para la cuenta de Quasi.

El argumento -u especifica el identificador de usuario, para que linux trate e identifique a Quasi por el ID 505

La opción -g marca el grupo inicial (por defecto) del usuario, en este caso seria el grupo con ID 105.

Mas adelante, tenemos la opción -p que debe especificar la contraseña para la cuenta de Quasi. Con la que este usuario podrá tener acceso al sistema (kak0ta).

Luego con la opción -s, especificamos la ruta del bash que se ejecutara una vez sea validada la clave del usuario, y tenga acceso al sistema. El sistema shell a cargar.

Y por ultimo, debemos especificar el nombre de la cuenta, en este caso Quasi.

En algunos casos podemos olvidarnos la especificación de algún parámetro. De ser así podemos escribir el comando

"useradd", tal cual y este ya nos va pidiendo los datos y parámetros de forma secuencial y automatizada.

useradd -D: visualiza los parámetros de usuario o permite cambiarlos.

-b: directorio home

-e: fecha deshabilitada

-g: grupo

-s: shell

Modificando usuarios

Comando: usermod

Etimología: Modificar usuario

Sintaxis: usermod [-c][d][e][g/G][m][p][s][u]opciones <nombreCuentaUsuario>

-c: comentario o nombre completo

-d: directorio home por defecto.

-e: fecha en la que se deshabilita la cuenta YYYY-MM-DD

-g: grupo inicial , tiene que haber sido creado ya.

-G: Lista de grupos suplementarios separados por comas y sin espacios en blanco a la que también pertenece el usuario.

-m: Crea el directorio home y copia los ficheros contenidos en /etc/skel (ficheros de inicialización).

-p: clave de entrada

-s: nombre y ruta del shell

-u: nombre numérico que identifica al usuario. Debe de ser único si no se especifica la opción o. Del 0-99 están reservados.

-f: nombre completo o comentario

-r: extensión

Si deseamos modificar la clave de un usuario, lo que deberíamos hacer sería entrando como superusuario (root), pondríamos: <passwd Quasi>, de esta forma cambiaríamos la password del usuario Quasi.

usermod -G game, public, video pepe

Esto añadiría pepe a los grupos: video, game, public.

groups pepe

Con este comando veremos los grupos a los que pertenece pepe

last: Permite visualizar las entradas del usuario al sistema.

Comando chfn: Permite modificar los siguientes datos del usuario.

-f Comentario o nombre de usuario.

-r Se usa para la extensión

-w Telefono de trabajo

-h Telefono personal

-o Otros

Comando chsh: Nos permite cambiar de shell inicial. La única restricción es que solo pueden ser los citados en el fichero /etc/shell/

Sintaxis: chsh [-s shell] [nombreUser]

Borrando usuarios

Comando: userdel

Etimología: Eliminar usuario

Sintaxis: userdel [-r] <nombreDeCuentaUsuario>

Donde el <nombreDeCuentaUsuario> sería para el ejemplo ya hecho anteriormente "Quasi". Eliminaríamos a este usuario de nuestro sistema de usuarios.

Con la opción -r será borrada la dirección y ficheros. Los ficheros localizados en otro sistema de ficheros deberán ser borrados manualmente.

Para borrar un usuario no debe estar en el sistema. De ser así el superusuario deberá matar sus procesos y echarlo fuera. Por ej matando su shell.

Posibilidades para dar de baja a un usuario:

1. **No permitirle el acceso:** Editar el archivo /etc/passwd y poner un * donde está la clave. Si se utiliza un sistema de claves hacer esto en el /etc/shadow.

2. **Dar de baja en passwd:** pero guardando los archivos, esto es útil si los archivos los utilizan otros usuarios. Simplemente habría que teclear userdel nombre de usuario y con los comandos chown y mv se cambian la propiedad y ubicación de los archivos.

3. **Dar de baja completamente:** Es decir eliminar al usuario de los ficheros passwd y shadow y sus ficheros del home, bastaría con teclear userdel -r nombre de usuario.

userdel borraría a un usuario. Si lo que deseamos es betar (impedir que entre) a un usuario pero no eliminar su cuenta podemos preceder en su clave el carácter "*" como ya dije. Para ello editamos con vi el fichero /etc/passwd. Por ejemplo: Quasii:*Xv8Q981g71oKK:105:100:Javier_Fernandez_Rivera:/home/quasi:/bin/bash

Este sistema evitaría que Quasi se pudiera conectar, lo que estamos haciendo internamente es engañando al sistema con la clave. Es un truco que paso a estandarizarse como método de vetar sin eliminación de cuenta.

GESTION DE GRUPOS

Los grupos tiene la función de facilitar la tarea del administrador. Supongamos que creamos un grupo, dicho grupo tiene asignados unos permisos, a la hora de meter los usuarios en ese grupo por defecto estaríamos a su vez dando a los usuarios los permisos del grupo (los heredarían) de tal forma que el administrador no tendría que andar siempre poniendo permisos a cada usuario.

Por ejemplo, si eres administrador Linux de un colegio, podríamos crear grupos llamados: claustro, direccion, alumnos...

De esta forma a la hora de añadir un usuario a claustro, direccion o alumnos heredaría un permiso u otro y no tendría que modificar permisos uno a uno.

El SO a la hora de la instalación crea ya sus propios grupos.

El fichero "/etc/group" contiene la información (de forma secuencial lineal como passwd) de los grupos existentes.

Se sigue la siguiente sintaxis: nombre de grupo:clave:GID:otros miembros

Ejemplo:

root:*:0:

alumnos:*:100:Quasi,pukii

direccion:*:200:

otros:*:250:Quasi

En este ejemplo vemos como ya existe el grupo del superusuario root con el GID (0), en la línea siguiente se halla el grupo "alumnos", al que pueden acceder Quasi y pukii, y posee el GID (100), etc.

Añadiendo grupos

Comando: groupadd

Etimología: Añadir grupo

Sintaxis: groupadd -g gid [-o] nombre

gid: valor numérico identificador del grupo. Del numero 0-99 están reservados por el sistema. Debe ser valor único y positivo. o: se puede crear un grupo con el gid de otro. Este comando solo lo puede utilizar el administrador, si no se especifica la opción g: asigna el sistema al siguiente.

Ejemplos:

groupadd pruebas----- > pruebas:x:101:

En este ejemplo no especificamos parametros, el sistema tomara al grupo con el identificador libre que siga en su lista interna de grupos.

groupadd -g 150 prueba ---> prueba:x:150:

Aquí, si estamos predefiniendo el identificador del grupo, con la opción -g n, donde n sería el numero ID, en este caso 150.

groupadd -g 150 -o prueb -> prueb:x:150:

En este ejemplo hacemos que el grupo prueba presente el mismo ID, con la opción -o.
groupadd -g 150 prue-----> No te deja, debido a que no se especifica la opción (-o).

Modificando grupos

Comando: groupmod

Etimología: Modificando grupo

Sintaxis: groupmod [-g gid [-o]] [-n grupo] nombre

Nos permite cambiar el identificador de usuario o el nombre del grupo.

Ej. Cambiaremos de Prueba con el ID(101) a pruebalinux con ID(105)

groupmod -g 105 -n pruebalinux prueba

Eliminando grupos

Comando: groupdel

Etimología: Eliminar grupo

Sintaxis: groupdel <nombreGrupo>

Elimina un grupo en el fichero etc/group/ siempre que no tenga usuarios en: /etc/passwd

COMANDOS VARIOS

Uso del comando passwd

Comando: passwd

Etimología: password == palabra de paso == contraseña.

Sintaxis: passwd [-opciones] <cuentaUser>

Opciones:

Como ya sabemos el comando passwd sirve para cambiar la password (palabra de paso "contraseña"). Pero este comando a nivel de super-usuario (o sea si entramos con la cuenta de "root"), permite realizar ciertas operaciones especiales.

Tales operaciones son enumeradas, mediante ejemplos:

passwd Quasi -----> No pide la pass antigua simplemente pulsando intro, el administrador anula la pass del usuario.

passwd -d pukii -----> Tiene el mismo efecto que el caso anterior.

passwd -f Andera -----> Permite modificar los datos personales del usuario.

passwd -d -x 30 -n 20 TeMpEsT -> Este ejemplo, anula la pass del user y le pide que la cambie cada 30 días e impide que la modifique antes de 20 días.

passwd -l ipy -----> Bloquea la cuenta del usuario

passwd -a loco_bob -----> Desbloquea la cuenta del usuario (en caso de estar bloqueada "lógicamente").

passwd -S MoAsT -----> Nos muestra la siguiente información

```
pepe PS 03/23/2001 0 7000 7 0
```

```
pepe LK
```

```
pepe NP
```

Donde el primer dato es el nombre del usuario.

El segundo nos muestra el código de estado de la cuenta de ese usuario. El estado puede ser:

PS: tiene password.-

LK: bloqueado con la opción "-l".

NP: no tiene password.

El siguiente dato, representa la fecha de cambio de la password.

Los dos siguientes representan el número mínimo y máximo de días durante los cuales es validado el password. Ambos dos definidos mediante la opción x y n.

El siguiente dato (7), representa los días de advertencia del cambio.

Y el (0) es el periodo de inactividad.

Cambiando de cuenta de usuario.

Comando: su

Sintaxis: su <cuentaUser>

Donde cuentaUser es la cuenta a la que queremos acceder.

Si nos encontramos en la cuenta del root y queremos acceder a la cuenta de Quasi (por poner un ejemplo) debemos introducir en el shell de linux el comando: su Quasi Lógicamente el sistema no nos pedirá password para entrar en Quasi, esto es debido a que lo hacemos desde root a Quasi. De ser a la inversa (su root), el sistema si nos pediría la password del root.

COMANDO PASSWD

Consideremos el comando "passwd" para cambiar contraseñas. La mayoría de los sistemas linux almacenan las contraseñas en el fichero "/etc/passwd". Este fichero es legible para todos los users pero solo puede escribir en el root. Al ejecutar en el comando:

```
"ls -l"
```

Nos muestra:

```
_rw_r_r_ 1 root root 891 fecha
```

Para poder modificar este fichero debemos entrar como root. El programa passwd debe ser propiedad del usuarios root ya que solo root tiene permisos de escritura sobre el fichero. Sin embargo cualquier usuario puede ejecutar el comando passwd. Con lo visto hasta el momento un usuario cualquier no podría modificar el "/etc/passwd". La solución a este dilema consiste en definir programas llamadas "seguid raiz".

El comando passwd es uno de ellos, es decir cuando se ejecuta su "uid efectivo", se establece como "uid real" del proceso, permitiéndole actualizar el fichero "/etc/passwd". Si ejecutamos el comando "ls -l /usr/bin/passwd" nos muestra:

```
_r_sr_xr_x 1 root root 935 fecha /usr/bin/passwd
```

Donde la "s" del propietario indica con que el comando passwd se ejecutara como un programa "setuid raiz". De la misma forma una "s" en el bit ejecutable del grupo significa que si el programa se ejecutara como un programa setguid.

Funciones para la obtención de los atributos reales y efectivos en C

```
getuid()      identificador usuario
geteuid()     identificador efectivo usuario
getgid()      identificador grupo
getegid()     identificador efectivo grupo
getpid()
getppid()
```

Para probarlo se realiza el siguiente código en C:

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
main () {
    printf("%d",getuid());
}
```