

Curso Básico de Linux - Fundatec

Alberto Sierra Reales

Tarea de investigación 1

Contenido

Parte I: Grupos y usuarios.....	3
1.1 Creación de cuentas de usuario	3
1.2 Elección de números para el uid y el guid	3
1.3 El ambiente por defecto: /etc/skel	4
1.4 Creación manual de cuentas de usuario	4
1.5 Cambio de las propiedades del usuario.....	4
1.6 Eliminación de cuentas de usuario	5
1.7 Bloqueo temporal	5
Parte II: Permisos y Propiedades de archivos	6
2.1 Interpretación de la línea de comando.	6
2.2 Asignación de permisos	7
2.3 Asignación de propiedad	8
Parte III: Recuperación de la contraseña de root.....	9
3.1 Inicializar la maquina	9
3.2 Montar la partición.....	9
3.3 Editar el archivo de contraseñas.....	9
3.4 Desmontar la partición	10
3.5 Reiniciar en el sistema original.....	10

Parte I: Grupos y usuarios

1.1 Creación de cuentas de usuario

El kernel de Linux trata a los usuarios como simples números. Cada usuario es identificado por un único número entero llamado *uid* (user id), esto porque los números son mucho más rápidos de leer por la computadora que una cadena de caracteres. Esta información se almacena en una base de datos por aparte, relacionando el *username* con el *uid*, además de almacenar otra información sobre el usuario.

En casi todas las distribuciones de Linux se incluyen comandos para la creación de cuentas de usuario, como *useradd* o *adduser*, las cuales no requieren ningún conocimiento extra, a continuación se explica como crear cuentas de usuario manualmente.

La base de datos de usuario básica en un sistema Linux es el archivo de texto */etc/passwd*, el cual contiene todos los usuarios válidos y su información asociada. Este archivo utiliza una línea por cada usuario y cada línea es dividida en 7 campos separados por dos puntos (:):

- Nombre de usuario
- Contraseña (cifrada)
- Identificador de usuario (uid)
- Identificador de grupo (guid)
- Nombre completo o comentario
- Directorio del usuario
- Interpretador de comandos (Shell)

El formato es explicado en detalle en el manual de *passwd*.

Cualquier usuario en el sistema puede leer al archivo *passwd* para saber, por ejemplo, el nombre de otro usuario, esto significa que cualquier usuario puede ver la contraseña, la cual, por estar cifrada, teóricamente no sería problema. Sin embargo, la cifrado es violable, especialmente si el la contraseña es débil (si es corta o se puede encontrar en un diccionario, etc.), por lo cual no es buena idea guardar las contraseñas en este archivo.

Muchos de los sistemas Linux utilizan el método de *shadow passwords* el cual guarda las contraseñas cifradas en el archivo */etc/shadow*, el cual solo el usuario *root* puede leer. En ese caso, el archivo */etc/passwd* contiene un carácter especial en el campo de la contraseña. Cualquier programa que necesite verificar una contraseña, utiliza el *setuid* para tener acceso al archivo, de otra manera, se niega el acceso.

1.2 Elección de números para el uid y el guid

En la mayoría de sistemas Linux, no importa cual sea el número elegido, siempre y cuando no se repita y no se use un sistema de archivos de red (Network File System, NFS), esto porque el NFS también utiliza estos números para la identificación de los usuarios, si no se está usando el NFS, puede crearlos aleatoriamente.

De todas maneras, se debe evitar reutilizar los *uids* (o mismos nombres textuales) ya que el nuevo usuario podría tener acceso a los archivos del usuario anterior de ese *uid*.

1.3 El ambiente por defecto: /etc/skel

Cuando el directorio del usuario es creado, este es inicializado con los archivos en /etc/skel. El administrador del sistema puede crear archivos en /etc/skel para crear un ambiente por defecto en el directorio del usuario, por ejemplo, puede crear el archivo /etc/skel/.profile con la variable EDITOR para definir el editor por defecto para los usuarios nuevos.

Se debe tratar de mantener /etc/skel tan pequeño como sea posible, ya que es muy difícil actualizar los perfiles de usuario existentes, y es preferible usar un *script* para llevar a cabo esta tarea, sin embargo, podría corromper alguna configuración de los perfiles existentes, así que por lo general se recomienda que la configuración global se almacene en archivos de configuración como /etc/profile.

1.4 Creación manual de cuentas de usuario

Para crear una cuenta de usuario manualmente, se deben seguir los siguientes pasos:

- Editar el archivo /etc/passwd con el comando **vipw**, y agregue una línea nueva para la cuenta. Sea cuidadoso con la sintaxis. NO TRATE DE EDITAR DIRECTAMENTE CON UN EDITOR NORMAL, *vipw* bloquea el archivo para que otros comandos no actualicen el archivo al mismo tiempo. El campo de la contraseña debe ser "*" para que sea imposible ingresar al sistema.
- Similarmente, editar el archivo /etc/group con **vigr**, en caso de necesitar también un grupo nuevo.
- Crear el directorio del usuario con el comando *mkdir*.
- Copiar los archivos de /etc/skel al nuevo directorio.
- Cambiar los permisos y la propiedad del nuevo directorio con los comandos *chmod* y *chown*, respectivamente. La opción -R es de mucha utilidad para la recursividad del comando. Los permisos correctos varían un poco entre distribuciones, pero los siguientes comandos deberían funcionar:

```
#cd /home/nuevousuario
#chown -R usuario.grupo .
#chmod -R go=u-w .
#chmod go= .
```

- Crear la contraseña con el comando *passwd*. Una vez creada la contraseña en el paso anterior, la cuenta estará activa. Se recomienda no activar la cuenta hasta que todo lo demás se haya terminado, así se evita que un usuario ingrese al sistema cuando el administrador todavía esta copiando archivos.

1.5 Cambio de las propiedades del usuario

Existen algunos comandos para cambiar las propiedades de las cuentas del archivo /etc/passwd:

chfn: Cambia el campo del nombre o descripción.

chsh: Cambia el campo del shell.

passwd: Cambia la contraseña del usuario.

El usuario root puede cambiar las propiedades de cualquier usuario del sistema, los usuarios normales solamente pueden cambiar las propiedades de su misma cuenta.

Estas y otras propiedades pueden ser editadas manualmente en el archivo /etc/passwd.

1.6 Eliminación de cuentas de usuario

Para eliminar una cuenta de usuario, primero elimine todos sus archivos, y todas las referencias a ese usuario, después se eliminan las líneas necesarias de los archivos `/etc/passwd` y `/etc/group`. Una buena práctica es bloquear la cuenta temporalmente antes de proceder a eliminarla (el bloqueo de cuentas se explica en el punto 1.7)

Hay que recordar que los usuarios pueden tener archivos fuera de su directorio personal, el comando `find` puede encontrarlos:

```
#find / -user nombre_de_usuario
```

Tenga en cuenta que este comando tomará algo de tiempo dependiendo del espacio de sus discos y puede inclusive afectar los servicios de red.

Algunas distribuciones de Linux poseen comandos para la eliminación de cuentas tales como `userdel` o `deluser`, pero estos comandos pueden no hacer todo.

1.7 Bloqueo temporal

En algunas ocasiones, es necesario deshabilitar o bloquear una cuenta sin removerla, en caso de mantenimiento del servidor o de abusos por parte de un usuario.

La mejor manera de bloquear una cuenta es cambiar el *shell* por un programa que despliegue un mensaje informativo al usuario, de esta forma, el usuario sabrá la razón del bloqueo y podrá contactar al administrador, un ejemplo de esos programas sería un *"tail script"*:

```
#!/usr/bin/tail +2
Esta cuenta esta temporalmente suspendida.
Contacte a su administrador de red para más detalles.
```

Donde `"#!"` le indican al kernel que el resto de la línea necesita ejecutarse para interpretar el archivo. En este caso, el comando `tail` envía todo excepto la primera línea a la salida estándar (`stdout`).

Parte II: Permisos y Propiedades de archivos

Linux es un ambiente multiusuario. En un ambiente multiusuario la seguridad tanto de los archivos de sistema como de los archivos de usuario es muy importante, por esto el acceso a los archivos debe ser otorgado al usuario que lo necesita.

2.1 Interpretación de la línea de comando.

Primero veamos de qué se tratan los permisos. Los permisos se definen para usuarios, grupos y otros. El usuario podría ser el nombre que se usa para ingresar al sistema, además, los usuarios pueden ser organizados en grupos para mayor control y facilidad de administración. Cada usuario pertenece a por lo menos un grupo por defecto. Otros incluye todo lo que ha sido excluido de los dos anteriores.

A continuación el resultado del comando “*ls -l*”

```
#ls -l
drwxr-x--- 2 max admin 4096 Dec 28 04:09 tmp
-rw-r--r-- 1 max admin 969 Dec 21 02:32 foo
-rwxr-xr-x 1 max admin 345 Sep 1 04:12 mi_archivo
```

La primera columna se refiere al tipo de archivo seguido por los permisos, la tercera y cuarta, el usuario y el grupo al que pertenecen.

El primer archivo, *tmp*, con una “d” en la primera columna indica que es un directorio, para los otros dos archivos, el valor cambia a “-” lo que significa que es un archivo corriente. Se descompone así:

d	rwX	r-x	---
Tipo	Usuario	Grupo	Otros

Los siguientes nueve caracteres después del tipo de archivo, indican los permisos, los cuales están divididos en grupos de tres. Los primeros tres, son para el dueño del archivo, seguidos de los del grupo al que pertenece el archivo y por último para los que no pertenecen al grupo (otros). Los permisos están definidos por tres tipos de atributos diferentes:

r: Permiso de lectura (Read), en el caso de un directorio, permite listar el contenido del mismo.

w: Permiso de escritura (Write), permite que el archivo sea modificado, en un directorio, permite crear, borrar o renombrar archivos dentro del mismo.

x: Permiso de ejecución (eXecute), permite la ejecución de archivos, en el caso de un directorio, permite entrar al directorio, realizar una búsqueda o ejecutar un archivo desde el mismo.

Tomando los permisos de *tmp* como ejemplo, podemos entonces afirmar que el dueño de este directorio es *max* y el grupo al que pertenece es *admin*. Los primeros tres atributos son *rwX*, esto permite acceso ilimitado al usuario *max* para leer, modificar, o ejecutar sobre el directorio, los permisos para el grupo son *r-x* lo que significa que los usuarios de ese grupo pueden entrar al directorio y listar los contenidos pero no pueden crear, borrar o modificar ningún archivo en el directorio, el resto de los usuarios no tiene acceso, definido por ---.

Para el archivo *foo* el usuario *max* tiene permisos para leer y modificar, el grupo puede leer sin hacer modificaciones al igual que los otros.

2.2 Asignación de permisos

Para asignar o modificar las propiedades de un archivo se utiliza el comando *chmod*. Para cambiar los permisos de un archivo, se debe ser el dueño del archivo o root. La sintaxis del comando *chmod* es relativamente sencilla y los permisos se asignan para usuarios (u), grupos (g), u otros (o).

Un ejemplo del comando *chmod* sería:

```
#chmod u-x,g+w,o+rw mi_archivo
```

En el anterior comando, se retiran los permisos de ejecución al usuario, se le agregan los de escritura al grupo y le da permisos de lectura y escritura a todos los demás. Antes de ejecutar este comando, los permisos eran *-rwxr-xr-x*, después del comando, los permisos son: *-rw-rwxrwx*. Primero, se elije usar "u", "g" u "o" ó los tres, seguido de "+" para asignar un atributo, "-" para eliminar un atributo ó "=" para eliminar cualquier otro anterior. También se puede utilizar "a" para designar a todos.

```
#chmod a+rw mi_archivo
```

La segunda forma para asignar permisos es la forma numérica, cada atributo tiene un valor numérico, de esta forma:

```
r=4  
w=2  
x=1
```

Así, rwx será $4+2+1=7$, r-x sera $4+1=5$, por lo cual entonces usamos el comando *chmod* de la forma:

```
#chmod ugo mi_archivo
```

Donde u,g,o representan los permisos para el usuario, grupo y otros, respectivamente. Por ejemplo:

```
#chmod 644 mi_archivo
```

```
6 = 4 + 2 = rw  
4 = 4 = r  
4 = 4 = r
```

En este caso los permisos asignados serian *-rwr--r--*, una forma mucho mas sencilla y rapida, a continuación una tabla de referencia:

```
0 = ---  
1 = --x  
2 = -w-  
3 = -wx  
4 = r--  
5 = r-x  
6 = rw-  
7 = rwx
```

2.3 Asignación de propiedad

Además de los permisos se puede cambiar el usuario y el grupo al que pertenece un archivo, esto se logra mediante los comandos *chown* y *chgrp* respectivamente, para ejecutar estos comandos es necesario ser el dueño del archivo o root.

```
#chown max foo  
#chgrp admin foo
```

Parte III: Recuperación de la contraseña de root

Este procedimiento es algo avanzado para un usuario nuevo en Linux, consta de 5 pasos los cuales son:

3.1 Inicializar la maquina

En este paso tratamos de inicializar la maquina con un Linux al que tengamos acceso de root, puede ser un disco vivo u otra instalación en el mismo disco.

Muchos de los discos de instalación de Linux traen algún tipo de interpretador de comandos con herramientas básicas.

3.2 Montar la partición

Montar una partición significa hacerla visible y disponible para el sistema.

La primera tarea, es identificar la partición que vamos a montar. En algunas instalaciones como redhat, hay muchas particiones con directorios diferentes, como `/boot`, `/var`, `/home`, etc., en este caso necesitamos la partición donde esta localizado el directorio `/etc`, para esto, hay que tener un conocimiento previo del sistema o por tanteo hasta encontrar la partición. Tambien se puede usar el comando `fdisk` para ayudar en el proceso.

Una vez localizada la partición, la montamos desde la consola con el comando `mount`:

```
#mount /dev/hda3 /mnt
```

Donde `/dev/hda3` es la partición que queremos montar y `/mnt` es el directorio en que vamos a montar la partición en nuestro sistema de rescate. Si el comando se ejecutó satisfactoriamente, la consola le mostrará una nueva línea, si no, le mostrará un mensaje de error. En algunas ocasiones el comando `mount` no podrá adivinar el tipo de sistema de archivos, en estos casos el comando debe ir acompañado por esa variable:

```
#mount -t ext3 /dev/hda3 /mnt
```

Donde la opción `-t` indica el tipo de sistema de archivos en la variable siguiente.

3.3 Editar el archivo de contraseñas

En este paso utilizaremos un editor para editar el archivo de contraseñas, normalmente existirá un clon de `vi`.

Primero determinamos el archivo a editar, comúnmente es `/etc/shadow` pero en algunos sistemas antiguos puede ser todavía `/etc/passwd`. Si ninguno de los anteriores aplica, tal vez la distribución almacena las contraseñas de otra manera, busque el manual de `passwd` para esa distribución. Puede utilizar el comando `cat` para verificar si el archivo es el correcto, si contiene una línea parecida a:

```
root:fx2SDLhg40vDU:11803:0:99999:7:::
```

Una vez encontrado el archivo, ábralo con el editor y elimine los caracteres entre los dos primeros dos puntos (:) para que se vea de esta forma:

```
root::11803:0:99999:7:::
```

Salve y salga del editor (en `vi`, use `:x`).

3.4 Desmontar la partición

Esto se logra con el comando *umount*:

```
#umount /dev/hda3
```

3.5 Reiniciar en el sistema original

Usar el comando *reboot* o *halt*, una vez el sistema operativo original cargue, el sistema no pedirá contraseña para root. Una vez ingresado, recuerde cambiar la contraseña con el comando *passwd*.