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.

El concepto de usuario en Linux permite separar entornos de ejecución para diferentes propósitos. Dos personas pueden trabajar simultáneamente en el mismo sistema, teniendo cada uno un usuario diferente, y un directorio personal diferente.

También es muy común que muchos servicios internos del sistema tengan su propio usuario para restringir el acceso de ese servicio como mecanismo de seguridad. De este modo, si un servicio ve su seguridad comprometida por un ataque, el acceso que tenga el usuario de ese servicio servirá como contención del ataque, y no podrá acceder a ficheros pertenecientes a otro usuario (de persona o servicio).

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.

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)
- Demonio (daemon) es el término usado en Linux para referirse al proceso de un servicio que se ejecuta en segundo plano de forma no interactiva. En general acaban por la letra d, como httpd o ftpd.

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.

Archivo /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.

Archivo /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 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. Considerar a 'shadow' como una extensión de 'passwd' ya que no solo almacena la contraseña encriptada, sino que tiene otros campos de control de contraseñas.

Grupos

Los grupos permiten conceder permisos a un conjunto de usuarios simultáneamente.
En Linux un usuario tiene los siguientes grupos:

- Grupo primario: es el que consta como su *GID* en /etc/passwd. Sólo puede haber un grupo primario.
- Grupos secundarios o suplementarios: son los gestionados en el fichero /etc/groups, donde se puede añadir un usuario a más grupos.

Además, durante la sesión de usuario se puede cambiar temporalmente el grupo al que pertenece el usuario:

- Grupo real: es su grupo primario que consta en /etc/passwd. Es el grupo al que pertenece un usuario cuando inicia sesión.
- Grupo efectivo: mediante el comando newgrp se puede cambiar el grupo primario al que pertenece el usuario, y la configuración es efectiva hasta que cierre la sesión o vuelva a cambiar de grupo efectivo.

Archivo /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'.

Permisos

Una de las cosas más importantes en lo que a seguridad de la información se refiere son los permisos que otorgamos a nuestros ficheros. En ellos, decidimos quién puede y quién no puede acceder a la información almacenada por nosotros mismos o por el resto de los usuarios que acceden a nuestro sistema.

En una instalación por defecto en un sistema GNU/Linux, el sistema de ficheros cumple con el estándar Posix (con algunas diferencias). Este nos permite modificar los permisos sobre los ficheros y directorios con herramientas comunes para todas las distribuciones.

Cada uno de los elementos del sistema de ficheros de Linux posee permisos de acceso de acuerdo a tres tipos de usuarios:

1. Su dueño (casi siempre el creador) representado por la letra "u" (user)
2. Su grupo representado por la letra "g" (group)
3.El resto de los usuarios que no son el dueño ni pertenecen al grupo. Se representan con "o" (other).

Para cada uno de estos tres grupos de usuarios existen tres tipos de permisos fundamentales:

1. "r": read (lectura). El usuario que tenga este permiso podrá si es un directorio, listar los recursos almacenados en él, y si es cualquier otro tipo de fichero podrá leer su contenido.

2. "w" write (escritura). Todo usuario que posea este permiso para un fichero podrá modificarlo. Si se posee para un directorio se podrán crear y borrar ficheros en su interior.

3. "x" executive (ejecución). Este permiso para el caso de los ficheros permitirá ejecutarlos desde la línea de comandos y para los directorios, el usuario que lo posea tendrá acceso para realizar el resto de las funciones permitidas mediante los otros permisos (lectura y/o escritura).

Para determinar los permisos finales siempre se deben tener en cuenta los siguientes aspectos:

- Para poder realizar operaciones sobre cualquier directorio (leer o escribir) será necesario siempre, tener otorgado además el permiso de ejecución.
- Para acceder a un recurso de cualquier forma (ejecución, lectura o escritura) se deben tener permisos de ejecución para todos los directorios que contienen al recurso directa o indirectamente.

Los tres tipos de permisos mencionados poseen una representación numérica basada en el sistema octal que parte de representar como "1" los *bits* de los permisos otorgados y "0" para los negados. Luego se transforma la representación binaria así obtenida en octal. De esta forma se obtiene para cada tipo de permiso los siguientes valores:

r - - = 100 (4 en octal)

- w - = 010 (2 en octal)

- - x = 001 (1 en octal)

La combinación de los tres tipos de permisos para un tipo de usuario oscila desde cero (ningún permiso) hasta siete (todos los permisos)

Ejemplos:

r w - = 110 (6 en octal)

r w x = 111 (7 en octal)

r - x = 101 (5 en octal)

Los permisos "totales" de un recurso consta de nueve indicadores, donde los tres primeros indican los permisos asociados al dueño, los otros tres, al grupo y los últimos, al resto de los usuarios.

La notación octal consiste de valores de tres a cuatro dígitos en base-8. Con la notación octal de tres dígitos cada número representa un componente diferente de permisos a establecer: clase de usuario, clase de grupo y clase de otros (resto del mundo) respectivamente. Cada uno de estos dígitos es la suma de sus bits que lo componen (en el sistema numeral binario). Como resultado, bits específicos se añaden a la suma conforme son representados por un numeral:

- El Bit de ejecución (acceso en el caso de directorios) añade **1** a la suma.
- El bit de escritura añade 2 a la suma
- El bit de lectura añade 4 a la suma.

Cabe señalar que el permiso 3 (wx) es el resultado de 1+2 (w+x), que el permiso 5 (rx) es el resultado de 4+1 (r+x), que el permiso 6 (rw) es el resultado de 4+2 (r+w) y que el permiso 7 (rwx) es el resultado de 4+2+1 (r+w+x).

El formato octal puede ser utilizado por el comando ‘chmod’ que es el encargado de modificar los permisos de ficheros y directorios.

Ejemplos con notación octal:

Solo lectura para el dueño del archivo

chmod 400 sample.txt

Solo lectura para el grupo del archivo

chmod 040 sample.txt

Permisos de lectura para todos

chmod 004 sample.txt

Solo puede escribir el dueño del archivo

chmod 200 sample.txt

Solo puede escribir el grupo del archivo

chmod 020 sample.txt

Cualquiera puede escribir el archivo

chmod 002 sample.txt

Solo lo puede ejecutar el dueño

chmod 100 sample.txt

Solo lo puede ejecutar el grupo

chmod 010 sample.txt

Todos los pueden ejecutar

chmod 001 sample.txt

Permitir la lectura para todos

chmod 444 sample.txt

Permite que todos puedan leer, escribir y ejecutar

chmod 777 sample.txt

Ejemplos en modo simbólico:

Nadie puede ejecutar el archivo

chmod a-x sample.txt

Todos pueden leer el archivo

chmod a+r sample.txt

El archivo puede ser leído por el grupo y otros

chmod go+rw sample.txt

Realizar un script por shell ejecutable por el dueño solamente

chmod u+x samplescript.sh