CENTRE ESPECÍFIC
D´EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

UNIDAD 10.LINUX - PARTE 2

Sistemas informáticos
CFGS DAW

Alfredo Oltra
alfredo.oltra @ ceedcv.e s

2019/2020

Versión: 190927.1219

Licencia

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

- Importante

- Atención

- Interesante

ÍNDICE

UD010. LINUX - PARTE 2

1) USUARIOS EN LINUX

Linux es un sistema operativo multiusuario.

Los usuarios en Linux tienen un nombre asociado, pero internamente se identifican por un número. Este identificador se llama UID. Si dos usuarios tienen un nombre diferente pero el mismo UID, internamente son el mismo usuario. Más información en
https://en.wikipedia.org/wiki/User_identifier

Básicamente hay dos tipos de usuarios: usuarios normales y root.

- Un usuario normal es un usuario con UID mayor que 0 y puede realizar operaciones limitadas y solo acceder / modificar recursos a los que tiene permiso de acceso.

- El usuario raíz es un usuario con UID = 0. Es el administrador principal del sistema y prácticamente puede hacer casi todo (cambiar la configuración, instalar programas, instalar controladores, ejecutar servidores, leer / eliminar cualquier archivo, ...).

> • Hacer operaciones siendo usuario root es muy peligroso (puede cometer un error y romper su sistema). Si ingresa en un sistema como root, debe saber muy bien qué está haciendo.

1.1 Archivos "/ etc / passwd" y "/ etc / shadow"

La lista de usuarios se almacena en un archivo llamado "/ etc / passwd". Almacena varios atributos como UID, directorio de inicio, si el usuario está habilitado o no, ... Si ejecutamos "cat / etc / passwd" podemos ver su contenido. Se puede encontrar más información sobre el archivo "/ etc / passwd" en

https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/

También la contraseña cifrada se puede almacenar en "/ etc / passwd", pero no se recomienda por razones de seguridad ("todo el mundo podría leer" / etc / passwd "). Por esta razón, hay otro archivo para las contraseñas de la tienda llamado "/ etc / shadow" que solo el usuario root puede leer y modificar.

Puede encontrar más información sobre el archivo "/ etc / shadow" en
https://www.cyberciti.biz/faq/understanding-etcshadow-file/

> • Resumiendo, "/ etc / passwd" almacena información general de los usuarios y "/ etc / shadow" almacena contraseñas cifradas.

1.2 Comando "sudo" y lista de sudoers

Hace unas pocas líneas, hemos dicho que hay 2 tipos de usuarios: usuarios root y usuarios normales. Es una forma ineficiente e insegura de administrar cuentas de administrador. Por esta razón, las distribuciones modernas de Linux como Ubuntu o Mint:

- De forma predeterminada, la cuenta raíz está desactivada (no puede iniciar sesión como root).
- Hay una lista llamada "sudoers". En esta lista, puede otorgar varios privilegios a los usuarios normales.
- El privilegio más común (y útil) es "convertirse en root" temporalmente usando un comando llamado "sudo" antes de la instrucción para realizar. Con esta herramienta y esta configuración, el sistema puede tener más de un administrador (cada usuario que está en la lista de sudoers puede realizar operaciones de raíz). También es obligatorio usar el comando "sudo" antes de que el comando se ejecute como root. Aumenta la seguridad porque se supone que si usas "sudo" sabes lo que estás haciendo.

Ejemplo:

Si el usuario pepe (UID = 1001) está en la lista de sudoer y se ejecuta

"Sudo cat archivo.txt"

Ejecuta el comando "cat archivo.txt" siendo root (UID = 0).

- Cuando ejecuta por primera vez en su sesión un comando sudo (o su último comando sudo fue hace mucho tiempo), el sistema le solicita su propio inicio de sesión por razones de seguridad.

Más información en https://en.wikipedia.org/wiki/Sudo

1.3 Comando "su"

El comando "su" es una abreviatura de "Cambiar usuario". Este comando se puede llamar:

- Sin parámetros: en este caso, intenta iniciar sesión como root (UID = 0). Funciona incluso si la cuenta raíz está deshabilitada.
- Con parámetro: tiene un parámetro que es el nombre de usuario en el que desea iniciar sesión.

Si ejecuta el comando como root, se registra automáticamente como el usuario. Si es un usuario normal, le pedirá la contraseña de usuario.

Ejemplo:

“Su pepe”

El sistema intentará iniciar sesión como el usuario "pepe". “Sudo su”

El sistema intentará iniciar sesión como root (UID = 0). Más información en https://en.wikipedia.org/wiki/Su_(Unix)

1.4 Crear usuarios en Linux

En esta página, puede leer información sobre cómo crear usuarios (por línea de comando) y, si lo desea, otorgarles privilegios de "sudo": https://www.digitalocean.com/ community / tutorials / how-to-add-and-delete-users-on-ubuntu-16-04

También puedes ver un ejemplo con interfaz gráfica en este video https://www.youtube.com/watch?v=DQHS1tQ2Xt8

- Cuando crea un usuario en Linux, el contenido predeterminado de su nuevo directorio de inicio se obtiene del directorio "/ etc / skel". Funciona como una "plantilla". Más información en http://linuxg.net/skeleton-directory-etcskel /

2) GRUPOS EN LINUX

Linux te permite crear grupos de usuarios. Es útil otorgar permisos o privilegios (como la lista de sudoers) a un grupo completo (por ejemplo, puede otorgar privilegios de "sudo" a un grupo y cada miembro de este grupo podría ejecutar el comando sudo para convertirse en root).

Un usuario puede ser miembro de varios grupos a la vez.

Al igual que los usuarios, los grupos tienen un nombre, pero internamente se identifican mediante un GID entero. Si dos grupos comparten el mismo GID, internamente son el mismo grupo.

2.1 Archivo "/ etc / group"

Hay un archivo "/ etc / group" donde se enumeran los grupos. Cada línea es un grupo y almacena información como nombre, GID y el valor más importante: la lista completa de usuarios que son miembros de ese grupo. Más información sobre "/ etc / group" en

https://www.cyberciti.biz/faq/understanding-etcgroup-file/

2.2 Crear grupos en Linux

En este enlace puede ver cómo crear un grupo y agregar un nombre de usuario existente a ese grupo usando la consola http://www.omnisecu.com/gnu-linux/redhat-certified- engineer-rhce / how-to-create-a-new-group-in-linux-using-groupadd- command.php

También puedes ver cómo hacerlo gráficamente en este video https://www.youtube.com/watch?v=ZNeWntArcOg

3) ARCHIVOS Y DIRECTORIOS EN LINUX

3.1 Tipos de archivos

En Linux hay esos tipos de archivos:

- Archivos regulares: contiene información. Son archivos normales, como los que usamos todos los días.
- Directorios: son archivos especiales con referencias a otros directorios y archivos.
- Enlaces
  - Enlaces simbólicos: es un archivo que contiene la ruta a otro archivo. Es similar a los accesos directos de Windows. Si elimina el archivo original, el enlace simbólico permanece, pero apunta a un archivo inexistente.
  - Enlaces duros: no es un tipo de archivo, es un segundo nombre para un archivo. Si crea un enlace duro de un archivo, para el sistema de archivos son el mismo archivo y no hay forma de saber cuál es el original. Si un archivo tiene más de una referencia, solo se elimina cuando se eliminan todas las referencias.
- Archivos especiales: son archivos que generalmente representan dispositivos físicos, como unidades de almacenamiento, impresoras ...

3.2 archivos ocultos

En Linux, los archivos ocultos son archivos que comienzan con "." como ".bash". Cuando enumera un directorio, no aparecen, a menos que utilice el parámetro "-a". Puedes verlos usando "ls -a".

4) PERMISOS EN LINUX

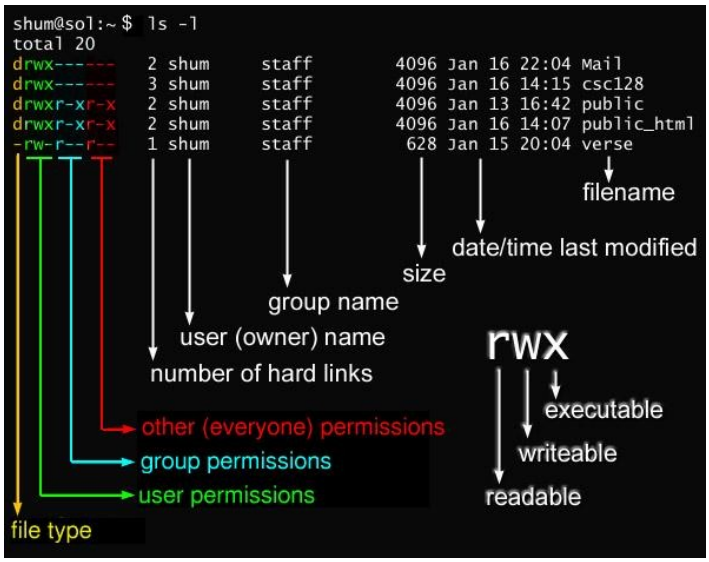
En Linux usando el comando de línea de comando "ls -l" puede ver información detallada sobre archivos y directorios. Esta información contiene permisos de cada archivo o directorio.

Los principales tipos de permisos en Linux son:

- Leer
  - En un archivo: permite leer su contenido.
  - En un directorio: permite enumerar sus archivos, nombres de directorios y atributos (comando ls).
- Escribir
  - En un archivo: puede modificar el contenido del archivo.
  - En un directorio: puede eliminar o crear archivos y directorios en ese directorio.
- Ejecutar
  - En un archivo: puede ejecutar el archivo (como Windows ".exe").

- En un directorio: puede ingresar el directorio (comando cd). Estos permisos principales deben definirse en 3 grupos: propietario (afecta al propietario del archivo), grupo (afecta al miembro del grupo) y otros (afecta a otros usuarios).

Un ejemplo del comando "ls -l" aplicado a los permisos:



4.1 Algoritmo de concesión de permisos

Para determinar si se otorga un permiso o no, sigue el siguiente algoritmo:

1) Primero verifique si el usuario es root (UID = 0). Si es cierto, se otorga permiso.

2) En segundo lugar, compruebe si el usuario es el propietario. Si es el propietario, se aplican los "permisos de propietario".

3) En tercer lugar, si el usuario no es root o el propietario, pero es un miembro del grupo asociado al archivo, se aplican los "permisos de grupo".

4) Por último, el usuario no es root, ni el propietario ni el miembro del grupo; se aplican "otros permisos".

- yo Es posible encontrar contradicciones como "otros" tienen más permisos que "propietario". Si "otros" pueden escribir y el propietario no puede, aunque es extraño, es una configuración válida.

4.2 Uso del comando chmod para establecer permisos

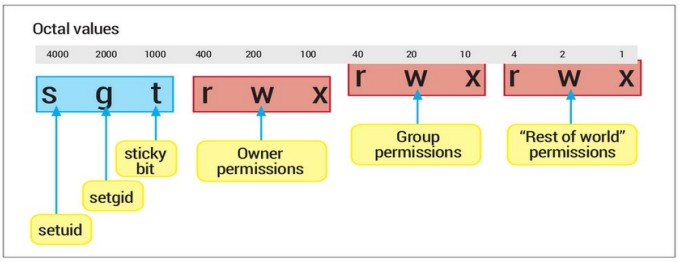
El comando chmod se usa para establecer permisos. Solo la raíz y el propietario del recurso pueden cambiar los permisos. Chmod tiene principalmente dos notaciones:

- **Notación alfa:**
  - Ejemplo: chmod u = rwx, g = rx, o = - myFile.txt # Pone todos los permisos al propietario, lee una ejecución al grupo y nada a los demás.
- **Notación octal:**
  - Utiliza el "valor binario" de un valor octal para establecer permisos. Por ejemplo, 5 es 101 en binario y es equivalente en rwx leer y ejecutar permisos.
  - Ejemplo: chmod 750 myFile.txt Pone los mismos permisos que el último ejemplo

Más información al respecto en http://www.perlfect.com/articles/chmod.shtml

4.3 Permisos especiales

Hemos hablado de 9 bits de permisos (rwx para propietario, rwx para grupos y rwx para otros). Pero hay 3 bits más: setUID, setGID y Sticky bit:



- **setUID:** https://en.wikipedia.org/wiki/Setuid
  - En archivos: si el permiso setUID está activado, cuando ejecuta ese archivo, no lo ejecuta con su propio UID, lo ejecuta con el UID del propietario.
  - En directorios: si el permiso setUID está activado, si crea un archivo o un directorio, el propietario no es usted, es el propietario del directorio principal donde se encuentra.
- **setGID:** https://en.wikipedia.org/wiki/Setuid
  - Lo mismo que setUID, pero con ID de grupo en lugar de ID de usuario.
- **Poco pegajoso:** https://en.wikipedia.org/wiki/Sticky_bit
  - Hoy en día se usa principalmente en directorios. Si alguien tiene permiso de escritura en un directorio, puede crear archivos y directorios, pero también puede eliminar cualquier archivo o directorio. Si el bit fijo está activado en un directorio, cualquier persona con permisos de escritura puede crear archivos y directorios, pero solo puede eliminar archivos y directorios que son de su propiedad.
    - La única excepción son la raíz y el propietario del directorio principal.

Más información sobre esos permisos en http://www.unixrock.com/2013/09/how-to-use-setuid-setgid-and-stickybit.html

5) COMANDOS PRINCIPALES

En esta sección vamos a describir los comandos principales de la consola en los sistemas Linux. Si desea obtener información detallada sobre cada uno de ellos, puede utilizar el "comando man".

Mando	Que hace	Ejemplo
Comandos para administrar la interfaz		
hombre	Muestra ayuda de un comando man ls	
claro	Pantalla clara	Claro
eco	Mostrar un literal texto en pantalla.	echo "Hola Mundo"
salida	Cierra La sesión en consola	salida

Mando	Que hace	Ejemplo
Comandos para configurar el sistema		
fecha	Establecer fecha del sistema	fecha # Muestra fecha fecha -s # Establece fecha
California	Muestra el calendario	California
apagar	Apagar el sistema	apagar
reiniciar	Reiniciar el sistema	reiniciar

Mando	Que hace	Ejemplo
Comandos para obtener información sobre discos		
du	Muestra el uso del disco para cada archivo.	du -h ##Formato legible por humanos
df	Muestra información sobre sistemas de archivos.	df -h ##Formato legible por humanos

Mando	Que hace	Ejemplo
Comandos para administrar archivos y directorios		
toque	Crea un archivo vacío	toque myfile.txt
vi / nano	Crea / edita un archivo de texto	nano myfile.txt vi myfile.txt
mkdir	Crea un directorio	hacer mydir
gato más	Muestra el contenido de un archivo de texto.	cat myfile.txt más myfile.txt
grep	Busca un usuario de texto en un archivo de texto	grep root / etc / contraseña
ls	Muestra contenidos de un directorio	ls ls -la
discos compactos	Directorio de cambios	cd / home # Ruta absoluta cd ../myDir # Ruta relativa
pwd	Muestra la ruta actual	pwd
rm	"Rm" elimina archivos "Rm -r" elimina un directorio de forma recursiva	rm myfile rm -r myDirectory
cp	"Cp" copia un archivo "Cp -r" copia un directorio de forma recursiva	cp myFile / home / admin cp -r myDir / home / admin

mv	Mueve / renombra un archivo o un directorio	mv myFileOldName / home / myNewName
En	"Ln" crea un enlace duro. "Ln -s" crea un enlace simbólico (como los accesos directos de Windows.	En mi archivo hardLinkMyFile ln -s myFile shortcutMyFile
montar	Montar un dispositivo en una carpeta.	mount / dev / sda1 / media / myDisk

Mando	**Que hace**	**Ejemplo**
Comandos relacionados con permisos		
chmod	Cambia los permisos de un archivo o directorio	chmod 750 myFile
chown	Cambia el propietario / grupo de un archivo o directorio	chown newuser: newgroupt my file

6) MATERIAL ADICIONAL

[1] Glosario. [2] Ejercicios

7) BIBLIOGRAFÍA

[1] "Los Linux mando línea" Creativo Los comunes libro http://linuxcommand.org/tlcl.php