Práctica 4: Administración de Usuarios y Grupos

Objetivos

Un sistema operativo habitualmente es multi-usuario, lo que permite compartir los recursos que expone el sistema entre diferentes *cuentas*. Además de compartir, esta característica permite aislar y compartimentar el acceso y determina una jerarquía de privilegios para operar el sistema. En esta práctica veremos cómo administrar cuentas de usuario y grupos, así como la gestión de los permisos que implica.

Preparando el entorno...

En esta práctica necesitaremos una máquina virtual con el sistema CentOS 7 instalado. Según se explica en la práctica 3, haced un clon enlazado de la instalación base disponible en el laboratorio.

Las cuentas de usuario

Después de la instalación se definen varios usuarios. La mayor parte de ellos se utilizan para ejecutar servicios específicos. Hay, sin embargo, una cuenta especial root, que está destinada a realizar tareas de administración. En esta primera parte empezaremos viendo el manejo básico de estas cuentas.

Ejercicio 1. *Información de la cuenta*. Entrar en el sistema con el usuario cursoasr. Obtener la información de la cuenta mediante el comando id (identificador numérico de usuario UID, y grupo GID; así como los grupos a los que pertenece).

id

uid=1000(cursoasr) gid=1000(cursoasr) groups=1000(cursoasr),10(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

Ejercicio 2. *Cambiar de cuenta de usuario.* Para cambiar de cuenta de usuario se utiliza el comando su:

- Consultar su página de manual especialmente la opción -l (ó el equivalente -) y -c
- Cambiar al usuario root. Comprobar la información de este usuario con id
- Volver al usuario cursoasr saliendo de la shell. Comprobar la diferencia en el entorno si se usa su y su - (ó su -l) para cambiar a root.

sudo - Te permite acceder a cualquier usuario que le indiques
sudo -l Solo puedes acceder a root

---

root es el usuario padre por lo tanto su id=0 y no pertenece a ningun grupo mientras que cursoasr tiene un id y esta asociado a varios grupos

Ejercicio 3. *Otros comandos*. Para ver qué usuarios están en el sistema tenemos el comando w, además hay algunas variantes de id como whoami. Probar estos comandos.

w Nos permite ver que usuario esta en el sistema y las horas de conexion

id Te dice la informacion precisa de un usuarioad
whoami Te dice quien es el usuario con el que estas loggueado

Permisos

Cada usuario está definido por su identificador y el grupo o grupos a los que pertenece. El sistema de control de acceso básico del sistema se establece en función de estos dos parámetros, el usuario y el grupo. En esta sección veremos los comandos básicos para gestionarlos: chmod, chown, chgrp, y umask.

Permisos de archivos y directorios

Ejercicio 1. Comprobar los atributos de los ficheros del directorio home del usuario cursoasr, con la orden **ls -la**. Las propiedades son <tipo><rwx_propietario><rwx_grupo><rwx_resto>:

- **Tipo:** - fichero; d directorio; l enlace; c dispositivo carácter; b dispositivo bloque; p FIFO; s socket
- **Permiso:** r: lectura (4); w:escritura (2); x:ejecución (1)

Comprobar los permisos del directorio /etc/sudoers.d (ls -ld) e intentar cambiar a ese directorio como usuario cursoasr.

ls -la

total 8
drwxr-xr-x. 3 root root 21 Feb 10 2016 .
dr-xr-xr-x. 17 root root 4096 Feb 10 2016 ..
drwx------. 14 cursoasr cursoasr 4096 Aug 30 06:27 cursoasr

ls -la | grep sudoers.d permisos drwxr-x--- root root …. Indica que lo creo root por lo que no podemos modificarlo con cursoasr

Ejercicio 2. Escribir un script que imprima la frase (“Curso Administración”) que llamaremos mi_echo. Para poder ejecutarlo añadir permisos de ejecución con chmod +x mi_echo.

mi_echo:

```
!#/bin/bash
echo “Curso Administracion”
```

ejecutar: **chmod +x mi_echo.sh**
./mi_echo.sh

Curso Administracion

Ejercicio 3. Los permisos se pueden otorgar de forma selectiva usando la notación octal o la simbólica. Ejemplo, probar las siguientes órdenes (equivalentes):

- chmod 540 mi_echo
- chmod u+rx,g+r-wx,o-wxr mi_echo

¿Cómo se podrían fijar los permisos rw--w--wx, usando ambas notaciones?

ls -la | grep "mi_echo"

-rwxrwxr-x. 1 cursoasr cursoasr 40 Aug 30 07:12 mi_echo.sh

chmod 540 mi_echo.sh
ls -la | grep "mi_echo"

-r-xr-----. 1 cursoasr cursoasr 40 Aug 30 07:12 mi_echo.sh

chmod 623 mi_echo.sh para fijar los permisos rw—w--wx

chmod 623 mi_echo.sh para fijar los permisos rw—w--wx

chmod u+rwx,g+rw,o+w scrip.sh

Ejercicio 4. Crear un directorio y quitar los permisos de ejecución para usuario, grupo y otros. Intentar cambiar al directorio. Para que un usuario pueda cambiar un directorio tiene que tener permisos de ejecución.

Sin permisos no se pueden hacer nada en cursoasr , solo el root puede acceder a el .

Permisos especiales SUID, SGID y sticky

Ejercicio 1. Hay dos permisos de ejecución especiales: set uid, SUID y set gid, SGID. Si un fichero tiene activados esos permisos se ejecutan con la identidad del propietario (o grupo propietario) en lugar del usuario que invoca la ejecución:

- Listar las propiedades de la utilidad /usr/bin/passwd

ls -l | grep passwd

- Los permisos SUID se pueden añadir con +s o en el byte más significativo un 4. Ejemplo añadir los siguientes permisos al script (u+rws,g+rx ó 4750).

chmod 4750 /usr/bin/passwd
chmod u+rws /usr/bin/passwd
chmod g+rx /usr/bin/passwd

Nota: Aunque los permisos se fijan el kernel de Linux no permite la ejecución de scripts con SUID y propietario root por defecto.

Ejercicio 2. El permiso SGID sobre directorio matiene un significado especial, los archivos creados heredan la propiedad del grupo:

- Crear un directorio y dar los permisos SGID (g+wrxs, 2770), un 2 en el byte más significativo
- Cambiar a root, crear un fichero y ver sus atributos
- Volver al usuario cursoasr, ¿puede modificar los contenidos, y borrar el fichero?

mkdir directorio
chmod g+wrxs directorio
ls -la
su -
cd home/cursoasr/directorio/
touch fichero.txt
ls -la

total 4
drwxrwsr-x. 2 cursoasr cursoasr 24 Aug 30 19:28 .
drwx------. 15 cursoasr cursoasr 4096 Aug 30 19:25 ..
-rw-r--r--. 1 root cursoasr 0 Aug 30 19:28 fichero.txt

exit
ls -la |grep directorio

-rw-r--r--. 1 root cursoasr 0 Aug 30 19:28 fichero.txt

Ejercicio 3. Finalmente el *sticky bit* (1 en el byte más significativo, ó chmod +t) sirve para permitir unicamente al propietario eliminar un fichero. Suele emplearse en directorios compartidos, p. ej. /tmp. Comprobar que a pesar de poder escribir en el directorio /tmp no podemos borrar ficheros de otros usuarios.

Al crear el fichero y ejecutar el comando chmod +t fichero solo permite que el propietario y el root puedan borrar esos ficheros.

Permisos por defecto

Ejercicio 1. La orden umask muestra los permisos que **no** se otorgan a un fichero o directorio cuando se crea. Comprobar la máscara por defecto del usuario, crear un archivo y comprobar los permisos con los que se crea.

```
[cursoasr@asrserver ~]$ umask
        0002
[cursoasr@asrserver ~]$ su -
[root@asrserver ~]# umask
        0022
```

Ejercicio 2. Modificar la máscara de forma que no se de ningún permiso a "otros" ni permisos de modificación al propio grupo. Comprobar el resultado.

Propietario

Ejercicio 1. El superusuario puede cambiar el propietario de un fichero (chown) y del grupo propietario (chgrp):

- Cambiar a root y crear el directorio /home/prueba
- Fijar el propietario y grupo propietario a cursoasr

```
chown cursoasr ./prueba ; chgrp cursoasr ./prueba
```

- Comprobar el funcionamiento

Nota: con chown se puede fijar ambos usando <usuario>:<grupo>, e.g. chown root:root /tmp

Administración básica de usuarios y grupos

La definición de los usuarios y grupos puede ser local (reside completamente en el servidor) o remota, si se encuentra en un servicio de directorio especializados como NIS o más comunmente LDAP, o Active Directory. En esta sección nos ocuparemos de la configuración local.

Archivos de configuración

Los archivos para la administración de usuarios y grupos son cuatro: /etc/passwd, /etc/shadow; /etc/group y /etc/gshadow.

Ejercicio 1. Abrir el fichero /etc/passwd y observar su estructura:

Listado 1. Estructura del fichero /etc/passwd
nombre_usuario:x:uid:gid:información:home:shell

El campo x, sirve para indicar que la información de la contraseña está en el fichero shadow. Usando el contenido del fichero password y las utilidades de unix (práctica 1):

- Listar el nombre (sólo el nombre de la cuenta) de los usuarios definidos
- Determinar el número total de usuarios en el sistema

cat -n passwd | grep -o '.*:x'

Ejercicio 2. *Deshabilitar cuenta, método 1.* Observar la shell especial nologin:

- Determinar su ubicación en el sistema

etc/passwd es donde se conecta cuando se abre el sistema

- Ejecutar directamente ese comando en un terminal.
- Copiar el fichero /etc/passwd a /etc/passwd.bck

cp passwd passwd.bck

- Cambiar la cuenta cursoasr para que tenga como shell wnologin. Usar la orden vipw

vipw
/sbin/nologin sustituirlo por /bin/bash

- Intentar entrar en otro terminal (Ctrl_Dcho + F2).

Intentamos acceder a ella y nos dice que la cuenta no esta disponible

- Restaurar la copia del fichero passwd.

cp passwd.bck passwd

Ejercicio 3. Por defecto en CentOS/RHEL cada usuario se asigna a un grupo propio. Abrir el fichero /etc/group y observar su estructura:

Listado 2. Estructura del fichero /etc/group
nombre_grupo:x:gid:miembros separados por ,

cat group

Ejercicio 4. Cada grupo (usando el sistema de permisos que veremos) implementa un rol. Por ejemplo, el grupo wheel tradicionalmente se asocia al grupo de administradores (por su acceso privilegiado a la orden su y configuraciones de sudo, más adelante), o el grupo disk para la gestión de discos:

- Añadir nuestro usuario cursoasr al grupo disk. Usar la orden vigr

cat -n group | grep disk

7 disk:x:6:

vigr

agregar cursoasr en el fichero donde esta el grupo disk

- Abrir un nuevo terminal y comprobar el cambio con el comando id

Ejercicio 5. Comprobar los permisos de los ficheros /etc/passwd y /etc/shadow. ¿Por qué está separada la información en dos ficheros?

/etc/passwd es donde se registra los usuarios del sistema mientras que /etc/chadow almacena las contraseña cifradas y nos da informacion de caducidad y validez

Ejercicio 6. Abrir el fichero /etc/shadow y observar su estructura:

Listado 3. Estructura del fichero /etc/shadow
nombre:6sal$hash:ultimo_cambio:min:max:inactiva:deshabilitada

cat shadow

Nota: 1 usa MD5, 5 usa SHA-256 en RHEL 5 y 6 usa SHA-512 en RHEL6/7. La "sal" se añade a la contraseña antes de encriptarla para dificultar diversos ataques.

Nota: min es el mínimo número de días que debe conservarse la contraseña, max el máximo sin cambiar y deshabilitada el número de días en los que se deshabilitará la cuenta después de que caduque la contraseña.

Ejercicio 7. *Deshabilitar cuenta, método 2.* El campo contraseña puede tener algunos significados especiales:

- En blanco (::), sin contraseña
- Resultado de encriptación nó valido (no en el conjunto de caracteres válido, p.ej. !,*) o si empieza por (!) cuenta bloqueada.

Hacer una copia de seguridad del fichero /etc/shadow y probar las combinaciones anteriores. Una vez terminado restaurar su contenido.

vipw -s

Ejercicio 8. La configuración y valores por defecto para el mecanismo shadow, se configura en el fichero /etc/login.defs. Abrir el fichero y observar su contenido, especialmente:

- FAIL_DELAY, LOGIN_RETRIES (Número máximo de intentos de inicio de sesión si la contraseña es mala)
- PASS_MAX_DAYS, PASS_MIN_DAYS Número máximo de días en que se puede utilizar una contraseña.
- UID_MIN, UID_MAX (valores automáticos cuando se añade un usuario)
- GID_MIN, GID_MAX (valores automáticos cuando se crea un grupo)

Este fichero configura las opciones del login de usuarios, es un fichero de texto en ASCII

Para mostrar el archivo por partes | more

- FAIL_DELAY,
- LOGIN_RETRIES(Número máximo de intentos de inicio de sesión si la contraseña es mala)
- PASS_MAX_DAYS Número máximo de días en que se puede utilizar una contraseña.
- PASS_MIN_DAYS Número mínimo de días permitidos entre los cambios de contraseña
- UID_MIN:1000 UID_MAX:60000 (valores automáticos cuando se añade un usuario
- GID_MIN:1000 GID_MAX:60000 (valores automáticos cuando se crea un grupo)

Comandos de administración

Ejercicio 1. El comando useradd crea una cuenta y añade las entradas necesarias en passwd, shadow y group, además del directorio de usuario. Algunas opciones importantes para definir la cuenta (ver man useradd) son:

- -c comentario (sección información)
- -e fecha de expiración <aa/mm/dd>
- -f días para que se bloquee la cuenta después de que caduque la contraseña **<dias> y va de la mano de -e**
- -g grupo principal (por defecto creará uno, ver USERGROUPS_ENAB en login.defs)
- -G grupos adicionales
- -m crea el directorio home del usuario
- -s cambio de shell
- -**M sin directorio inicio**
- **-d crea un usuaio con dirctorio de inicio diferente**
- **-u <uid> -g <gid> crea un usuario con UID y GID especificos**

Crear varias cuentas de usuario con diferentes opciones. Comprobar el contenido de passwd, groups y shadow.

Ejercicio 2. Las contraseñas se pueden asignar con el comando passwd. Un usuario puede cambiar su propia contraseña:

- cambiar la contraseña de cursoasr con ese mismo usuario (passwd, sin opciones)
- poner una contraseña a las cuentas creadas en el ejercicio anterior. Comprobar los cambios en el fichero shadow

```
su -
passwd test
```

Ejercicio 3. El comando groupadd crea nuevos grupos. Crear un par de grupos uno de ellos con el GID 60002.

```
groupadd <name> -g 60002
```

Ejercicio 4. Para modificar una cuenta de usuario se usa el comando usermod:

- Deshabilitar una de las cuentas creada cambiando su shell
- Añadir una de las cuentas creadas a uno de los nuevos grupos (notar la diferencia entre -g y -G y la opción -a)

De la misma forma se puede modificar un grupo con groupmod (consultar su página de manual).

```
usermod silviaa -s /sbin/nologin
usermod nombreUsuario -g nombreGrupo
```

Ejercicio 5. Se pueden borrar las cuentas con userdel y groupdel, consultar las opciones (especialmente -

r para userdel). Probar estos comandos con algunos de los nuevos usuarios y grupos.

userdel -r <usuario> te permite borrar automaticamente el directorio home de ese usuario

Control de Acceso

Nunca, ni un administrador, debe usar root para el funcionamiento normal. Los administradores usan cuentas personales y adquieren permisos de root únicamente cuando es necesario. En esta sección veremos cómo controlar y definir este proceso y que herramientas utilizar.

Ejercicio 1. Puede ser necesario permitir el acceso a root al sistema, aunque se puede restringir los terminales desde los que se puede hacer login. El fichero /etc/securetty especifica que terminales son seguros para root:

- Hacer una copia del fichero

cp securetty securetty.bck

- Dejar solo tty3 y probar su comportamiento.

Ejercicio 2. Además de /etc/securetty para root, está el fichero /etc/security/access.conf que configura que usuarios y en que terminales pueden entrar al sistema. Cada entrada determina (+/-) habilita/deshabilita el acceso de un grupo o conjunto de usuarios al sistema desde una terminal o host (-:ALL EXCEPT root:tty1). Observar el contenido del fichero.

Vemos donde esta el root con cat access.conf | grep root

Ejercicio 3. El comando su, permite cambiar de usuario y requiere conocer la contraseña de la cuenta destino. Normalmente se usa la orden sudo, que permite acceder a los usuarios a comandos de administración **con su propia contraseña**:

- El fichero de configuración es /etc/sudoers y se edita con visudo
- Observar el fichero y estudiar la sintaxis empleada:

Listado 4. Sintaxis para la definición de acciones en /etc/sudoers
usuario máquina=(identidades de ejecución) comandos

¿Qué significan las entradas?:

- root ALL=(ALL) ALL

Sudo solicitará la contraseña del usuario especificado por la opción -u (por defecto, root) en lugar de la contraseña del usuario que invoca al ejecutar un comando o editar un archivo.

- %sys ALL = NETWORKING, SOFTWARE
- %wheel ALL=(ALL) NOPASSWD: ALL

Para permitir que todos los usuarios ejecuten todos los comandos sin una contraseña.

- Dar permisos al cursoasr para ejecutar cualquier comando sin contraseña

los vemos cat /etc/sudoers
para editar "i" para salir esc y :w y :q!
cursoasr ALL=(ALL) ALL y ahora hacemos sudo cat /etc/sudoers y no nos pide la contraseña
debajo de la linea donde pone root *ALL=(ALL:ALL) ALL*, añadimos
nombreUsuario ALL=(ALL:ALL) ALL y sudo funiona con nuestro nombre de usuario

Ejercicio 3. La orden sudo permite ejecutar un comando con la entidad de otro usuario, por defecto

root aunque se puede especificar cualquier otro con la opción -u.

- Usar sudo para reiniciar el servicio sshd (práctica 2) como usuario cursoasr
- Como usuario cursoasr cambiar al usuario root usando sudo y la opción -i. Una vez que podemos cambiar a root como cursoasr, deshabilitar el acceso con contraseña a root.
  Sudo -i (y nos pide la contraseña para ponernos como root es igual que si hacemos su)

* Prompt (texto por defecto antes del "$"), esta variable de entorno es actualizada y nos muestra el nuevo usuario y el home del usuario, donde nos encontramos

Configuración de las cuentas

La configuración de la cuenta de usuario hace referencia a las variables de entorno, máscara por defecto y comandos específicos que se ejecutan cuando un usuario hace login.

Ejercicio 1. Consultar el contenido del directorio /etc/skel, que contiene los archivos que se copian cuando se crea una cuenta de usuario (ej. .bashrc, .bash_profile o .bash_logout)

/etc/skel no tiene archivos

Ejercicio 2. El fichero /etc/bashrc contiene definiciones y configuraciones globales, se carga desde la configuración de usuario (.bashrc); estudiar su comportamiento.

Se carga cuando ejecutamos la consola el cual podemos editar para hacer un sin fin de cosas, desde mostrar un simple texto, hasta funciones que nos simplifiquen el uso mediante este Terminal.

Ejercicio 3. Añadir al fichero .bashrc las líneas necesarias para que por defecto:

- No se den permisos de escritura ni para el grupo ni otros a los archivos creados

Agregar al final del archivo umask 022

- Se muestre el mensaje "Hola <nombre_de_usuario> son las <hora>", tanto la hora como el nombre del usuario deben obtenerse con los comandos adecuados.

Inicio del archivo, después de las líneas comentadas, y escribimos algún texto, echo "hola nombreUsuario son las". y nos saldrá cuando volvamos a abrir una consola.

HOY=$(date) echo "comentario $HOY"

Ejercicio 4. Finalmente /etc/profile y /etc/profile.d contienen la configuración global del entorno. Observar el contenido del fichero profile (PATH, USER, HOSTNAME...) y el contenido de algunos de los ficheros en /etc/profile.d (e.g. colorls.sh).

SHELL=/bin/bash **(el tipo de shell en uso)**

TERM=xterm **(el programa de terminal por defecto)**

USER=pepito **(el nombre de usuario)**

PWD=/home/pepito **(la ruta por defecto del usuario)**

LANG**=es_ES.utf8** **(el juego de caracteres de idioma)**

DESKTOP_SESSION**=xfce** **(el entorno de escritorio)**

PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin **(informa al shell en la mayoría de los casos** BASH**) dónde se encuentran los programas binarios que puedo ejecutar en el sistema, sin tener que llamarlos por su ruta absoluta).**

Para saber más...

- En algunas circunstancias la gestión basada sólo en usuario y grupo no es suficiente (por

ejemplo queremos dar permiso de lectura a dos grupos a un mismo fichero). Se pueden fijar esos atributos con setfacl y getfacl

- El sistema de autenticación estudiado (passwd/shadow) es configurable usando los Linux Pluggable Authentication Modules (PAM) , directorio /etc/pam.d.