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.

Contenidos

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).

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.

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.

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`.

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`.

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?

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.

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`
- 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).

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 directorios tiene 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?

Ejercicio 3. Finalmente el *sticky bit* (1 en el byte más significativo, ó `chmod +t`) sirve para permitir únicamente 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.

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.

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
- 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

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

- Determinar su ubicación en el sistema
- Ejecutar directamente ese comando en un terminal.
- Copiar el fichero `/etc/passwd` a `/etc/passwd.bck`
- Cambiar la cuenta cursoasr para que tenga como shell nologin. Usar la orden `vipw`
- Intentar entrar en otro terminal (Ctrl_Dcho + F2).
- Restaurar la copia del fichero passwd.

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

Listado 2. Estructura del fichero `/etc/groups`

```
nombre_grupo:x:gid:miembros separados por ,
```

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`
- 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?

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

Listado 3. Estructura del fichero `/etc/shadow`

```
nombre:$6$sal$hash:ultimo_cambio:min:max:inactiva:deshabilitada
```

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.

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`
- `PASS_MAX_DAYS, PASS_MIN_DAYS`
- `UID_MIN, UID_MAX`
- `GID_MIN, GID_MAX`

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
- -f días para que se bloquee la cuenta después de que caduque la contraseña
- -g grupo principal (por defecto creará uno, ver `USERGROUPS_ENAB` en `login.defs`)
- -G grupos adicionales
- -m crea el directorio home del usuario
- -s shell

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`

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

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.

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
- 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.

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

- Dar permisos al `cursoasr` para ejecutar cualquier comando sin contraseña

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.

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`)

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

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
- 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.

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).

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`.