Universidad Politécnica de Cartagena

Escuela Técnica Superior de Ingeniería de Telecomunicación

SEGURIDAD EN REDES

Práctica 2: Seguridad en el Sistema Operativo (I)

Creación de usuarios
Permisos
Listas de Control de Acceso (ACL)

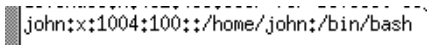
Autores: Josemaría Malgosa Sanahuja
María Dolores Cano Baños

SEGURIDAD EN REDES

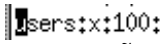
Seguridad en el Sistema Operativo PRÁCTICA 2 (primera parte)

IMPORTANTE: Para la realización de esta práctica será necesario trabajar con el usuario root (recordar que en la máquina virtual, la contraseña del usuario root es root)

CREACIÓN DE USUARIOS

1. Crear el usuario john con el comando `useradd -m john`. Verificar que el usuario se ha creado mirando el contenido del fichero /etc/passwd ¿Cuál es su user ID? Verificar que se ha creado el directorio /home/john 

ID= 1004

2. ¿Qué hace el comando `groups john`? Inspeccionando el contenido del archivo /etc/group determinar el group ID del usuario john ¿Para qué sirve el comando `id john`?
groups nos dice en que grupos está un usuario 

id muestra todas las ID relacionadas con ese usuario
3. Mediante el comando `passwd john` asignarle la contraseña `elloco`

4. De la misma forma, crear los usuarios paul y george y asignarles las contraseñas `elguapo` y `elmalo` respectivamente. Utilizando el comando `su <usuario>`, acceder al menos una vez al sistema utilizando el login de los tres usuarios creados ¿tienen algún tipo de archivos –aunque sean del tipo oculto- en su home? ¿A qué grupo pertenecen los usuarios paul y george? ¿Cuáles son los *group* ID de cada uno? Pertenecen a users con ID 100

5. ¿Qué hace el comando `groupadd cantantes`? Verificarlo inspeccionando el contenido del archivo /etc/group. Usando el comando `usermod -g`, asignar el grupo cantantes como grupo primario a los usuarios `john`, `paul` y `george`. Añade un grupo "cantantes"

6. Un usuario puede pertenecer a varios grupos. Crear los grupos `ritmica`, `bajo` y `solo`. Con el comando `usermod -a -G`, añadir el grupo secundario `ritmica` al usuario `john`, el grupo secundario `bajo` al usuario `paul`, y el grupo secundario `solo` al usuario `george` ¿Cómo podemos ver todos los grupos a los que pertenece un usuario? Con el comando groups "nombre"

7. Eliminar los usuarios john, paul y george con el comando `userdel -r`. Eliminar los grupos cantantes, bajo, ritmica y solo con el comando `groupdel`

PERMISOS

En Linux cada usuario tiene los siguientes atributos:

- Nombre de usuario
- UID: identificador único del usuario. Este identificador es asignado por el sistema operativo
- GID: identificador del grupo al que pertenece ese usuario. En Linux cada usuario puede pertenecer a uno o más grupos
- Contraseña: clave de acceso al sistema
- Directorio personal: ubicado dentro del directorio */home*

Los identificadores del usuario los usa el S.O. para decidir los privilegios o permisos que posee ese usuario a la hora de realizar una acción. Cada elemento del sistema de archivos, directorio o fichero tiene asociados unos permisos que describen quién puede utilizarlos. Sobre un fichero o directorio del sistema se puede realizar una o más de estas tres acciones:

- *read*: Leer el contenido del fichero o del directorio
- *write*: Modificar un fichero o escribir sobre un directorio
- *exec*: Ejecutar un archivo (si es posible), o acceder a un directorio (mediante *cd*)

Los ficheros y directorios tienen permisos concretos para cada una de estas acciones, para el propietario del archivo (a quién pertenece el archivo), para el grupo del fichero (grupo al que pertenece el propietario) y el resto de usuarios.

En Linux sólo permite que los usuarios realicen las acciones que los permisos atribuidos al archivo autorizan. Estos permisos definen la palabra de protección, que consiste en nueve bits repartidos en tres grupos de tres caracteres. Se visualiza al ejecutar un **`ls -l`** de la siguiente manera:

```
rwx rwx rwx
```

Empezando por la izquierda, el primer grupo de tres bits corresponde a los permisos del propietario del fichero. El siguiente a los permisos del grupo y el último al resto de usuarios. Si el bit está a uno, significa que está autorizado para hacer esa acción y si está a cero, no. Por ejemplo, si un archivo tiene la siguiente palabra:

```
rw-r-----
```

Equivale a 110-100-000, es decir, el propietario tiene permiso de lectura y escritura del archivo, el grupo solo de lectura y el resto de usuarios no tiene ningún permiso.

En la máquina remota existen los usuarios alberto, beatriz y carmen (sus contraseñas coinciden con el nombre de usuario). Conectarse como usuario `alberto` y responder a las siguientes preguntas: su alberto[o]

8. ¿A qué grupo pertenecen los usuarios `alberto`, `beatriz` y `carmen`? profes

9. ¿Quién es el propietario de los archivos ubicados en `~/doc`? ¿A qué grupo pertenecen? ¿Cuáles son los permisos de los archivos doc_priv_alberto.txt y doc_pub_alberto.txt? ¿le parecen adecuados? El propietario es Alberto y pertenece al grupo profes Publico sí, privado debería ser rw-------

10. ¿Para qué sirve el comando `chmod`? En el directorio ~/bin ejecutar los siguientes comandos y mediante el comando `ls -l`, comprobar cuáles son los permisos que se modifican:
chmod sirve para cambiar los permisos de un archivo -rw-rw-r-- 1 alberto profes 142 Sep 14 11:00 exe_alb

```
chmod g+w exe_alb
ls -l exe_alb
```
-rw-rw-r-- 1 alberto profes 142 Sep 14 11:00 exe_alb

```
chmod o+w exe_alb
ls -l exe_alb
```
-rw-rw-rw- 1 alberto profes 142 Sep 14 11:00 exe_alb

```
chmod u="rw",g="rw",o="r" exe_alb
ls -l exe_alb
```
-rw-rw-r-- 1 alberto profes 142 Sep 14 11:00 exe_alb

```
chmod 644 exe_alb
ls -l exe_alb
```
-rw-r--r-- 1 alberto profes 142 Sep 14 11:00 exe_alb

```
chmod 751 exe_alb
ls -l exe_alb
```
-rwxr-x--x 1 alberto profes 142 Sep 14 11:00 exe_alb

11. ¿Qué indican los permisos cuando se refieren a un directorio? Establecer para cada directorio y archivo los permisos que le parezcan más idóneos

1. **Lectura (r):** En archivos Puede listar, copiar o visualizarlo. En Directorios pueden ver el contenido, se pueden listar a través del comando ls
2. **Escritura (w):** En archivos significa que se puede modificar o borrar el contenido, incluso puede modificar los permisos. En Directorios significa que puede crear, eliminar archivos y directorios dentro de ese directorio.
3. **Ejecución (x):** En archivos significa que se puede ejecutar el contenido. En Directorios significa que podemos entrar en la carpeta (comando cd).

LISTAS DE CONTROL DE ACCESO (ACL)

El sistema tradicional es insuficiente en una gran cantidad de situaciones, ya que solo permite que un fichero pertenezca a un único propietario. Por otra parte, aunque sí permite que un fichero tenga asignado varios grupos, solo se le pueden asignar permisos al grupo principal. Sin embargo, en muchas ocasiones es necesario que un archivo pertenezca a varios usuarios y grupos simultáneamente, cada uno con sus propios permisos de lectura (r), escritura (w) y ejecución (x).

Las ACL (*Access Control List*) permiten especificar los permisos de acceso a los ficheros con mucha mayor granularidad. Con las ACL, un fichero puede pertenecer a tantos usuarios y grupos como se desee. Para cada fichero el sistema asocia una lista con todos los usuarios y grupos a los que pertenece y los permisos asociados con ellos (`rwx`). De esta forma, se solventan muchas de las restricciones del sistema anterior

12. Verificar con el comando `tune2fs -l /dev/sda2` de la máquina virtual que la partición `/dev/sda2` ha sido formateada y montada para poder usar ACL

13. Cambiar al usuario `alberto` y situarse en su *home*. Tomar nota del propietario y grupo al que pertenece el archivo `acl.bin`, así como sus permisos asociados ¿Qué hace el comando `getfacl acl.bin`? En lo referente a los permisos de usuario ¿aporta la misma información que el comando `ls -l acl.bin`? `-rw-rw---- 1 alberto profes 71 Oct 19 20:23 acl.bin`

```
alberto@localhost:~> getfacl acl.bin
# file: acl.bin
# owner: alberto
# group: profes
user::rw-
group::rw-
other::---

alberto@localhost:~> ls -l acl.bin
-rw-rw---- 1 alberto profes 71 Oct 19 20:23 acl.bin
```

14. Ejecutar `setfacl -m u:beatriz:rw acl.bin` Volver a ejecutar el comando `getfacl acl.bin` Ejecutar de nuevo el comando `ls -l` ¿Cómo se aprecia el hecho de que ahora hay más usuarios que los tres tradicionales de Linux? A la vista del resultado del comando `ls`, ¿han cambiado los permisos del grupo *profes*? ¿y del comando `getfacl`? ¿Cuál de los dos cree que proporciona la información correcta?

La información correcta la proporciona "getfacl"

```
alberto@localhost:~> ls -l acl.bin
-rw-rw---- 1 alberto profes 71 Oct 19 20:23 acl.bin
alberto@localhost:~> setfacl -m u:beatriz:rw acl.bin
alberto@localhost:~> getfacl acl.bin
# file: acl.bin
# owner: alberto
# group: profes
user::rw-
user:beatriz:rw-
group::rw-
mask::rw-
other::---

alberto@localhost:~> ls -l acl.bin
-rw-rw----+ 1 alberto profes 71 Oct 19 20:23 acl.bin
```

Las ACL -por cuestiones de compatibilidad con aplicaciones muy antiguas- definen el concepto de máscara. La máscara representa los permisos que todos los usuarios y grupos añadidos podrán tener como máximo. Cuando se utilicen ACL, los 3 bits que antes indicaban los permisos del grupo por defecto, ahora indicarán la máscara del fichero (o directorio).

15. ¿Cuál es la máscara asociada al archivo? ¿Puede ahora el usuario beatriz modificar el contenido del archivo acl.bin? Verificarlo haciendo que beatriz ejecute el siguiente comando: `echo "nueva linea al final" >> acl.bin` `rw-`

16. Ejecutar `setfacl -m u:beatriz:rwx acl.bin` y ejecutar el comando `getfacl acl.bin`; a la vista del resultado ¿ha cambiado el valor de la máscara? ¿puede ahora el usuario beatriz ejecutar el archivo acl.bin? Ejecutar el comando `chmod 664 acl.bin` ¿ha cambiado el valor de la máscara? ¿puede ahora el usuario beatriz ejecutar el archivo acl.bin? ¿de qué otra forma se puede cambiar la máscara? el valor de la máscara es rwx no puede ejecutarlo

17. Ejecutar `setfacl -m u:carmen:--- acl.bin` y ejecutar el comando `getfacl acl.bin`; a la vista del resultado ¿puede ahora el usuario carmen (que pertenece al grupo *profes*) ver el contenido del archivo acl.bin? No puede ver el archivo

18. Ejecutar `setfacl -x u:carmen acl.bin` ¿Qué ha ocurrido? Se borra la configuración de Carmen

19. Ejecutar `setfacl --set u::rw,g::rw,o::-,u:alumno:r acl.bin` ¿Qué ha ocurrido?

```
alberto@localhost:~> getfacl acl.bin
# file: acl.bin
# owner: alberto
# group: profes
user::rw-
user:alumno:r--
group::rw-
mask::rw-
other::---
```

20. Ejecutar `setfacl -b acl.bin` ¿Qué ha ocurrido?
Se borran los ajustes.