Grado en Ingeniería Informática

Seguridad y Riesgos en Sistemas de Información

Práctica 1 - Usuarios

Laboratorio

Contenidos

Introducción

El eslabón más débil en una infraestructura de TI es el usuario. Es por ello que en esta primera práctica se trabaja cómo se asegura un entorno Linux desde la perspectiva de los usuarios, mediante la asignación de permisos y configurando directorios con permisos especiales.

Usuarios

Los usuarios en *Unix* tienen un nombre único, teniendo un grupo *Unix* primario y pudiendo tener más grupos *Unix* asignados (suplementarios). Pero hay que tener en cuenta que la forma de identificarlo realmente es por el identificador de usuario (*UID*). En *Linux* se sigue la misma filosofía. Las cuentas de usuario son una forma de asegurar los archivos, además de proporcionar otras características, como ofrecer una interfaz de usuario personalizada.

Los usuario se crean mediante el comando **useradd**. Al crear un usuario podemos dar múltiples opciones, de las que se destacan las siguientes:

Opción	Acción
-b *directorio*	El directorio base por defecto si no se utiliza la opción *-d*. Este directorio se concatena con el nombre de usuario para definir el directorio *home*
-c “comentario”	Proporciona una descripción para la nueva cuenta de usuario, típicamente el nombre completo
-d directorio_home	Establece el directorio *home* para la cuenta. Por defecto corresponde con el *login* y se ubica en la carpeta */home*
-D	En lugar de crear una nueva cuenta, guarda la información suministrada como la nueva configuración por defecto para cualquier nueva cuenta que se cree
-e “fecha_expiración”	Establece una fecha de expiración para la cuenta
-f *-1*	Establece el número de días desde que la contraseña expira hasta que la cuenta es eliminada de forma permanente. Por defecto, -1, deshabilita esta opción. Si se establece a 0, la cuenta se deshabilita inmediatamente después de que la contraseña expire. Reemplazar -1 con el número a utilizar
-g grupo	Establece el grupo primario. Sin esta opción, se crea un nuevo grupo con el nombre de usuario y se asigna como grupo primario

Opción	Acción
-G lista	Añade el usuario a la lista, separada por comas, de grupos suplementarios. Si se utiliza esta opción después de crear el usuario hay que añadir -aG en lugar de -G para añadir grupos suplementarios
-k *skel_dir*	Establece el directorio esqueleto (por defecto */etc/skel*) que contiene los archivos de configuración inicial y *scripts* de *login* que deben de ser copiados al nuevo directorio de usuario. Este parámetro hay que utilizarlo con la opción -m
-m	Crea de forma automática el directorio de usuario y copia los archivos del directorio esqueleto
-M	No crea el directorio de usuario, aunque el comportamiento por defecto así lo indique
-o	Utilizar junto a la opción -u para crear una cuenta de usuario que tenga el mismo *UID* que otro nombre de usuario. De esta forma, se pueden tener dos nombres de usuario distintos con la misma autoridad sobre el mismo conjunto de archivos y directorios
-p *contraseña*	Permite añadir la contraseña para la cuenta que se añade. Tiene que ser una contraseña encriptada. Se puede generar una contraseña encriptada en *MD5* utilizando el comando *openssl passwd*
-s *shell*	Especifica el *shell* a utilizar por la cuenta
-u *user_id*	Especifica el número de *ID* de usuario para la cuenta. Por defecto se asigna el próximo número disponible

Al crear un usuario se crea una entrada en el archivo */etc/passwd*, mientras que el nuevo grupo se añade al archivo */etc/group*. Las contraseñas se guardan, de forma encriptada en los ficheros */etc/shadow* y */etc/gshadow*.

Opciones por Defecto

El comando *useradd* determina los valores por defecto para las nuevas cuentas a partir de los ficheros */etc/login.defs* y */etc/default/useradd*. Se pueden modificar estos valores por defecto editando de forma manual los archivos. Algunas opciones pueden variar entre sistemas. Como ejemplo, puede contener las siguientes opciones:

```
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
UID_MIN       1000
```

```
UID_MAX       60000
GID_MIN        1000
GID_MAX       60000
```

Para visualizar la configuración por defecto se puede utilizar la opción *-D* con el comando *useradd*:

```
$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
$
```

Se puede utilizar esta opción para cambiar los valores por defecto. No todas las opciones pueden ser modificadas, sólo cinco de ellas están permitidas (-b, -e, -f, -g y -s). Para establecer cualquiera de los valores por defecto, hay que utilizar el parámetro *-D* primero seguido de la opción por defecto que se quiera establecer, con la opción pertinente.

Modificar Usuarios

El comando para cambiar los parámetros de una cuenta es *usermod*. Muchas de las opciones disponibles son las mismas que las definidas en *useradd*. Algunas de las opciones a destacar, de entre las nuevas opciones con respecto a *useradd*, son las siguientes:

Opción	Acción
-l *login_name*	Cambia el nombre de *login* para la cuenta
-L	Bloquea la cuenta
-U	Desbloquea la cuenta

Borrar Usuarios

Para borrar un usuario se utiliza el comando *userdel*. Si se añade la opción -r también se elimina su directorio home.

Grupos

Cuando queremos compartir un conjunto de archivos con múltiples usuarios es donde entran en juego los grupos. El proceso consiste en crear un grupo y asignarle un conjunto de archivos. Sólo el usuario *root* puede realizar este proceso.

A la hora de crear un archivo, el grupo se establece con el grupo primario del usuario que lo crea.

La creación de grupos la realiza el usuario *root* a través del comando *groupadd*. El identificador de los grupos se asigna de forma automática, a partir del *ID* 1000, a no ser que se utilice el parámetro -g.

También se puede modificar tanto el *ID* como el nombre de un grupo, con los parámetros *-g* y *-n* respectivamente.

Pertenencia Temporal a un Grupo

Es posible permitir de forma temporal que un usuario sea miembro de un grupo utilizando el comando *newgrp* sin añadir el usuario al grupo. Para ello, con permisos de *root*, se utiliza el comando *gpasswd* para establecer una contraseña de grupo. A partir de este momento, cualquier usuario puede utilizar el comando *newgrp* seguido del grupo al que se quiere pertenecer de forma temporal. Esto hace que se abra una *shell* en la que el usuario utiliza el grupo especificado como grupo primario una vez proporcionada la contraseña del grupo.

Permisos Avanzados

La forma en que se gestionan las cuentas de usuario y grupos en Linux ha ido evolucionando con el tiempo. Se han ido añadiendo más funcionalidades para permitir formas más complejas de gestionar usuarios, grupos y sus permisos asociados. Pero la gestión de usuarios y grupos en este modelo básico tiene problemas de flexibilidad.

En el modelo básico, sólo un usuario y un grupo puede ser asignado a un archivo. Además, los usuarios normales no tienen la habilidad de asignar permisos específicos a múltiples usuarios y grupos, además de ofrecer muy poca flexibilidad para configurar archivos de forma colaborativa.

Con el tiempo se han incorporado mejoras que permiten que en este modelo los usuarios normales configuren directorios especiales para la colaboración utilizando el denominado *sticky bit* y el *bit GID* en directorios. Además, utilizando *Access Control Lists* (*ACL*), cualquier usuario puede asignar permisos específicos a los archivos para cualquier usuario y grupo.

ACL

La *ACL* permite a los usuarios normales compartir sus archivos y directorios de forma selectiva con otros usuarios y grupos. Se pueden asignar permisos de lectura, escritura y ejecución a archivos y directorios sin dejar estos elementos abiertos para todos y sin requerir que el usuario *root* tenga que realizar la operación.

Para poder utilizar esta funcionalidad el sistema de archivos tiene que tener habilitada la funcionalidad de *ACL*. En Ubuntu 22.04 basta con instalar el paquete *acl*:

```
$ sudo apt installa acl
```

Para gestionar los permisos *ACL* se utiliza el comando *setfacl*, por ejemplo:

```
$ setfacl -m u:usuario:rwx archivo
```

En este ejemplo se utiliza la opción *-m* para añadir permisos. A continuación se utiliza *u:* para indicar que se modifican los permisos a nivel de usuario. Se puede utiliza *g:* para indicar que se quieren cambiar los permisos de grupo. A continuación se indica el nombre de usuario (grupo) seguido de los permisos que se asignan. En este caso se indican todos (*rwx*), pero se puede indicar cualquier combinación de los tres.

Si se utiliza la opción *-x*, en lugar de *-m*, se eliminan los permisos indicados.

Para ver los permisos *ACL* se utiliza el comando *getfacl*:

```
$ getfacl archivo
```

También se puede añadir *d:* antes de la designación de usuario (*u:*) o grupo (*g:*) para establecer la configuración por defecto de *ACL* en un directorio:

```
$ setfacl -m d:g:grupo:rwx /tmp/directorio/
```

Cuando se asignan permisos *ACL* junto a los permisos básicos (*ugo*), se tienen que tener también en cuenta los permisos efectivos. Esta combinación hace que los permisos básicos que se establecen en un archivo establezcan una máscara con el permiso máximo que un usuario/grupo *ACL* puede tener sobre un archivo, por ejemplo:

```
[mary]$ touch /tmp/memo.txt
[mary]$ ls -l /tmp/memo.txt
-rw-rw-r--. 1 mary mary 0 Jan 21 09:27 /tmp/memo.txt
[mary]$ setfacl -m u:bill:rw /tmp/memo.txt
[mary]$ setfacl -m g:sales:rw /tmp/memo.txt
[mary]$ ls -l /tmp/memo.txt
-rw-rw-r--+ 1 mary mary 0 Jan 21 09:27 /tmp/memo.txt
[mary]$ getfacl /tmp/memo.txt
# file: tmp/memo.txt
# owner: mary
# group: mary
user::rw-
user:bill:rw-
group::rw-
group:sales:rw-
mask::rw-
other::r--
```

```
$ chmod 644 /tmp/test.txt
$ getfacl /tmp/memo.txt
# file: tmp/memo.txt
# owner: mary
# group: mary
user::rw-
user:bill:rw- #effective:r--
group::rw- #effective:r--
group:sales:rw- #effective:r--
mask::r--
other::r--
```

Creación de un Grupo para Colaboración

Se puede añadir un permiso que típicamente es ignorado al utilizar el comando *chmod* para cambiar los permisos del sistema de archivos. Este bit establece permisos especiales en directorios, entre otros. El valor para establecer el directorio colaborativo es 2 y se añade antes de los permisos de usuario, grupo y otros, por ejemplo:

```
$ chmod 2644 /tmp/directorioCompartido
```

Para crear un directorio para colaboración en grupo (*set GID bit*) tendremos que realizar los siguientes pasos:

1. Crear el grupo a utilizar para colaborar
2. Añadir usuarios al grupo para que puedan compartir archivos
3. Crear el directorio colaborativo
4. Asignar el grupo al directorio
5. Cambiar el permiso utilizando el *GID bit* a 2

Directorios Restringidos

Un directorio de borrado restringido se crea habilitando el *sticky bit*. La diferencia en este directorio es que, normalmente, si el permiso de escritura está habilitado para un usuario en un archivo o directorio, ese usuario puede borrar cualquier archivo o directorio. Sin embargo, en un directorio de borrado restringido, a no ser que seas el usuario *root* o el propietario del directorio, nunca podrás borrar los archivos de otros usuarios.

Para restringir el borrado en directorios hay que habilitar el *sticky bit*. Para ello cambiamos los permisos pero el primer bit se establece a 1, por ejemplo:

```
$ chmod 1775 /tmp/directorio
```

Acceso Administrativo

Los usuarios particulares pueden tener permisos administrativos para tareas particulares utilizando *sudo*. Para ello se antepone este comando a la tarea que se quiera realizar sin tener que conocer la contraseña de *root*. Por ejemplo:

```
$ sudo ls
```

Para conceder este privilegio se puede utilizar la utilidad *sudoers*, que permite que un usuario pueda utilizar el comando *sudo* y, por ejemplo, tener que introducir, o no, su contraseña (no la de *root*).

Para ello hay que modificar el fichero */etc/sudoers* y definir los privilegios que se desea que el usuario tenga. Para editar este fichero se utiliza, como usuario *root*, el comando *visudo*.

```
$ sudo visudo
```

Se muestra el fichero de configuración y se puede añadir la línea correspondiente. Por ejemplo, para conceder todos los privilegios se puede añadir la siguiente línea:

```
usuario ALL=(ALL) ALL
```

También se puede configurar para que se solicite la contraseña:

```
usuario ALL=(ALL) NOPASSWD: ALL
```

Ejercicios

- (e01) Añadir varios usuarios y comprobar los ficheros de configuración donde se encuentra la información de usuarios, grupos, contraseñas …
- (e02) Al crear una cuenta con *useradd*, ¿qué ficheros se tienen en cuenta y en qué orden?
- (e03) ¿Qué comando y opción permite ver la configuración utilizada por defecto al crear un nuevo usuario?
- (e04) ¿En qué archivo se encuentra especificada la shell por defecto al crear un nuevo usuario? ¿Qué valor tiene en tu sistema Linux?
  - Cámbiala y comprueba que funciona creando nuevos usuarios
- (e05) Crear varios usuarios combinando las opciones *-b*, *-d* y *-m*
  - ¿Bajo qué circunstancias tiene que existir (o no) el directorio previamente a la creación del usuario?
- (e06) Cambiar el contenido del directorio esqueleto y crear varios usuarios para comprobar su funcionamiento
- (e07) Comprueba las diferentes formas para añadir o sustituir grupos a un usuario
- (e08) ¿Qué opciones son iguales para los comandos *useradd* y *usermod*

- (e09) Buscar el shell por defecto al crear una cuenta en el archivo de configuración correspondiente
  - Establecer, con el comando correspondiente, el shell por defecto a /bin/bash
  - Comprobar el cambio en el fichero pertinente
  - Comprobar que el cambio se hace efectivo al crear nuevas cuentas de usuario
- (e10) ¿Cuando se deshabilita una cuenta, qué cambios se producen en los ficheros de configuración? Investiga el fichero de contraseñas de la cuenta al ser desactivada
- (e11) Crear una cuenta de usuario que tenga un directorio *home*. A continuación borrar el usuario.
  - Utilizar el comando *find* con la opción correspondiente para listar sus archivos y también borrarlos
  - ¿Qué opción tenemos que utilizar con el comando *ls* para ver el *uid*
- (e12) Crear una cuenta de usuario que tenga un directorio *home*, a continuación realiza los siguientes pasos:
  - Mostrar el *ID* del usuario en el fichero correspondiente
  - Listar los archivos de su directorio *home*
  - Borrar el usuario recién creado
  - Crear nuevo usuario (con nombre distinto al anteriormente borrado) que tenga un directorio *home*
  - Mostrar el *ID* del usuario en el fichero correspondiente
    - ¿Existe algún problema con el *ID* de este usuario y el del anterior?
  - ¿Qué sucede con el usuario recién creado con respecto al directorio *home* del usuario borrado anteriormente?
    - ¿Existe algún problema de seguridad en el sistema en estos casos?
- (e13) ¿Dónde se almacenan contraseñas de los grupos?
- (e14) Crear varios usuarios y grupos y asignar de forma temporal un grupo primario a un usuario
  - Experimentar creando/borrando archivos
- (e15) ¿Los grupos son creados exclusivamente por el usuario *root* a través del comando *groupadd*?
- (e16) Crear un grupo y ver que *ID* se le ha asignado. Luego cambiar el *ID* por otro y también su nombre
- (e17) Crear un archivo y modificar permisos ACL añadiendo varios permisos para otros usuarios/grupos
  - ¿Cambia la forma en que se muestran los permisos del archivo con el comando *ls*?
- (e18) Establecer los permisos *ACL* por defecto en un directorio y crear archivos en este directorio
  - ¿Se heredan los permisos *ACL*?
- (e19) Experimenta con los permisos *ACL* y los básicos y explica las diferentes configuraciones de permisos efectivos que se alcanzan
- (e20) Crear un directorio para colaboración en el que puedan colaborar varios usuarios que pertenezcan a un grupo determinado. Explica su fun-

cionamiento proporcionando ejemplos
  - ¿Cambia la forma en que se muestran los permisos del directorio con el comando *ls*?
- (e21) Crear un directorio de borrado restringido y crear/borrar archivos con diferentes usuarios. Explicar su funcionamiento proporcionando ejemplos
  - ¿Cambia la forma en que se muestran los permisos del directorio con el comando *ls*?

Evaluación

La evaluación de la práctica se realizará mediante una evaluación en clase y la entrega de una memoria de prácticas.

Memoria de prácticas

Durante la elaboración de la práctica se realiza un documento de memoria de práctica. Este documento es único para cada grupo y tiene que tener un formato y desarrollo adecuado, respetando los convenios básicos para la elaboración de documentos académicos. Por lo tanto, se espera que tenga su portada, índices, figuras, tablas, apartados, conclusiones, bibliografía, ..., por nombrar algunos de los elementos que debe de contener.

No se evaluarán las memorias que no especifiquen claramente los enunciados de cada actividad realizada, junto a una descripción adecuada del trabajo realizado.

La memoria recogerá todo el trabajo que ha llevado a cabo el grupo, destacando los ejercicios propuestos en la práctica (**exy**). Se evaluará la completitud de todas las actividades propuestas en esta práctica, así como los ejercicios adicionales realizados para trabajar la materia.

La fecha límite para entregar la memoria de prácticas es el 10 de marzo de 2024 a las 23:59 horas a través del Campus Virtual.

No se aceptará trabajos que no tengan un contenido y estructura adecuados y/o cuyas respuestas no sean suficientemente detalladas o no estén lo suficientemente justificadas. Se incluirán imágenes o figuras de texto que describan la salida del terminal, cuando así lo requiera el ejercicio.

Consideraciones Finales

- Se considera que se conocen los permisos básicos *ugo* del sistema de archivos *Linux*
- Algunos comandos de utilidad para el desarrollo de esta práctica (se aconseja utilizar el manual para ver las opciones que ofrece):
  - *ls*
  - *find*

– *chown*

Bibliografía

[1] Negus, C. 2020. *Linux Bible*. Standards Information Network.

[2] Peek, J. et al. 2002. *Unix Power Tools*. O'Reilly Media.

[3] Troncone, P. y Albing, C. 2019. *Cybersecurity Ops with bash*. O'Reilly Media.