6. Prácticas de gestión de usuarios

Entrega: 1) Mostrar al profesor la máquina cliente con los dos usuarios configurados y el correcto funcionamiento de los scripts en la máquina cliente. 2) Además, enviar los scripts por la USC virtual empaquetarlos en un fichero `tar.gz` sin subdirectorios.

6.1. Claves OTPW (1 pto)

En la máquina cliente crear un usuario y configurar el sistema y a este usuario para que pueda utilizar las claves de un solo uso en la conexiones por SSH.

NOTA: En *buster* no es necesaria la opción `UsePrivilegeSeparation`.

6.2. Cuotas (1 pto)

En la máquina cliente crear un usuario y configurar el sistema para fijar las cuotas de este usuario en la partición `/home`.

NOTA: antes de poner las cuotas al usuario con `edquota` es necesario reiniciar la máquina, puesto si no lo hacemos así se borrarían en el siguiente reinicio.

Asignarle al usuario una cuota de disco de 100 kB soft y 125 kB hard y una cuota de ficheros de 10 soft y 15 hard. Comprobar que los límites funcionan creando más de 15 ficheros (ya hay ficheros ocultos de configuración de la cuenta) o con el comando `dd if=/dev/zero of=`*`fichero`* `bs=`*`bytes`* `count=1` para crear ficheros de un tamaño concreto.

6.3. Creación de grupos y directorios en bash (4 puntos)

Realiza este y el siguiente ejercicio en la máquina virtual cliente.

NOTA: Realiza un backup de los archivos `/etc/passwd`, `/etc/shadow`, `/etc/group` y `/etc/gshadow` para poder restaurar el sistema después de cada prueba, esto es,

```
cp /etc/passwd /etc/passwd.ori
cp /etc/shadow /etc/shadow.ori
cp /etc/group /etc/group.ori
cp /etc/gshadow /etc/gshadow.ori
```

Los directorios creados se pueden borrar a mano.

Esquema de la empresa

La empresa ACME acaba de instalar un servidor Linux y desea dotar de cuentas de usuario a sus jefes y empleados, que se organizan en 7 departamentos: `gerencia`, `administracion`, `programacion`, `sistemas`, `comercial`, `rrhh` y `contabilidad`. Escribir un script en `bash` que haga lo siguiente:

1. Crear un grupo por departamento (mediante un lazo que recorra el conjunto de departamentos). Crear dos grupos adicionales: uno para los `jefes` y otro para los `empleados`.

2. Crear un directorio `/home/{departamento}/` para cada departamento. Dentro de estos directorios se crearan los directorios personales de los usuarios. Los usuarios solo podrán entrar en el directorio de su departamento y no en los otros. Ver propietarios y permisos en el fichero `arbol_grupos.txt`.

3. Dentro de cada directorio `/home/{departamento}/` coexistirá con los directorios personales de los usuarios otro directorio llamado `work` donde todos los usuarios del departamento (y sólo los de ese departamento) podrán intercambiar información (todos pueden crear y borrar archivos). Los usuarios de los otros departamentos no deben poder acceder (ni leer ni escribir) en ese directorio. Estos directorios (con los permisos adecuados) deben crearse en el script.

4. Crear también un directorio `/home/comun` donde todos los usuarios de cualquier departamento pueden intercambiar información. Cualquier usuario podrá crear ficheros en ese directorio, pero sólo el propietario (y el root) podrá borrar un fichero.

   Dentro de dicho directorio común, existirá otro directorio `comunJefes`, exclusivo para los jefes. En este directorio, sólo los jefes de cualquier departamento pueden intercambiar información. `comunJefes` no será accesible para ningún empleado no-jefe. Los ficheros creados en ese directorio sólo los podrán borrar el que los creó y el root. Estos directorios (con los permisos adecuados) deben crearse en el script.

5. Puede comprobarse el árbol del directorios resultante comparándolo con este fichero que puede descargarse: `arbol_grupos.txt`

6.4. Creación de usuarios en python (4 puntos)

Crear un script en `python` para añadir usuarios al sistema. Las características del script será las siguientes:

1. La versión de `python` será la `3.x`, por lo que se ha de indicar en la primera línea del script: `#!/usr/bin/env python3`. Si python3 no está instalado en nuestra maquina lo instalamos con: `apt-get install python3`.

2. El script debe aceptar los siguientes parámetros en línea (usar `argparse`):

   - `-u` ***usuario*** (parámetro obligatorio)
   - `-d` ***departamento*** (parámetro obligatorio)

- `-j/-e`: indican si es jefe o empleado. No pueden usarse los dos simultaneamente, por lo que deberá detectarse este caso mediante la creación de un grupo exclusivo en `argparse`. Si no se especifica ninguno, se considerará un empleado.
- `-p` *`clave`*: usa la clave indicada, si este parámetro falta se creará una contraseña aleatoria.

3. Si no se especifican los parámetros obligatorios (*usuario* y *departamento*) o se usan a la vez los dos parámetros exclusivos (*empleado* y *jefe*) el script mostrará el mensaje de ayuda y saldrá.

4. Si el usuario ya existe en el sistema, el script indicará este hecho y saldrá. Para comprobar si el usuario existe, en `python` disponemos del módulo `pwd` (para detectar un grupo tenemos `grp`). Para evitar que este módulo muestre error en pantalla, usar la construcción *try* y *except*.

5. Para crear los usuarios utilizad el comando `useradd` del sistema operativo llamado desde `python` usando el módulo `subprocess`.

   - Este comando requiere que la contraseña se almacene cifrada (como un hash), el cual se puede generar en `python` usando el módulo `crypt` (el parámetro *salt* funciona como semilla, puede usarse cualquier cadena).
   - El directorio del usuario será `/home/{departamento}/{usuario}`. Al comando hay que indicarle dos opciones, uno primero para que cree el directorio y otro segundo para que lo haga en el directorio especificado.
   - A partir del departamento, asignar al usuario al grupo correspondiente.
   - Dependiendo si el usuario es jefe o empleado, asignarlo al grupo secundario correspondiente.
   - El campo GECOS debe contener el nombre del usuario y el departamento al que pertenece, esto es, "usuario,departamento".
   - Indicar el shell `/bin/bash`.

6. La contraseña debe tener una validez de 30 días y empezar a avisar de la caducidad 10 días antes.

7. Dentro del directorio home del usuario crear enlaces simbólicos al directorio `work` de su departamento y al directorio `/home/comun`. En el caso de tratarse de un jefe, crear además un enlace simbólico adicional al directorio `/home/comun/comunJefes`.

8. Al finalizar el script, debe mostrar información sobre el usuario creado: nombre de login y contraseña. Comprobad que el usuario puede entrar en la cuenta con la clave generada y que puede crear archivos solo en los directorios que tiene asignados.

9. Ejecutar el script proporcionado `crea_usuarios.sh`. Comprobar la ejecución con el fichero `ejecucion_usuarios.txt` y que el árbol de directorios es similar al indicado en `arbol_usuarios.txt`. Se proporciona también el script `recupera.sh` para recuperar el estado original, borrando los usuarios y directorios creados.