

# SSO Práctica 1

Andrés Merlo Trujillo

## Índice

<b>1. Primer ejercicio</b>	<b>1</b>
1.1. /etc/passwd . . . . .	1
1.2. /etc/group . . . . .	2
1.3. /etc/shadow . . . . .	2
1.4. /etc/gshadow . . . . .	3
<b>2. Segundo ejercicio</b>	<b>3</b>
<b>3. Tercer ejercicio</b>	<b>6</b>
<b>4. Ejercicio 4</b>	<b>7</b>
4.1. /etc/pam.d/chfn . . . . .	7
4.2. /etc/pam.d/chsh . . . . .	8
<b>5. Quinto ejercicio</b>	<b>9</b>
5.1. Apartado a . . . . .	9
5.2. Apartado b . . . . .	9
<b>6. Sexto ejercicio</b>	<b>11</b>
<b>7. Séptimo ejercicio</b>	<b>11</b>
<b>8. Octavo ejercicio</b>	<b>12</b>
8.1. /var/log/lastlog . . . . .	13
8.2. /var/log/wtmp . . . . .	13
8.3. /var/log/utmp . . . . .	14
8.4. /var/log/btmp . . . . .	14
8.5. /var/log/sudo . . . . .	14
8.6. /var/log/messages . . . . .	14
8.7. Noveno ejercicio . . . . .	14
8.8. PC de mi casa . . . . .	14
8.9. PC de prácticas (máquina virtual) . . . . .	14

## 1. Primer ejercicio

A continuación, voy a explicar el formato y el significado de cada uno de los campos. Para ello, voy a dividir cada archivo en subsecciones:

### 1.1. /etc/passwd

Este fichero está formado por líneas de 7 campos separados por “:”. Los campos y sus significados son los siguientes:

1. **Nombre de login:** Nombre de usuario.
2. **Contraseña encriptada opcional:** Contraseña encriptada del usuario. Si este campo tiene la letra “x” minúscula, significa que la contraseña se almacena en “/etc/shadow”.  
Si se encuentra vacío, significa que no hace falta contraseña para autenticar.  
Si comienza en exclamación, significa que la contraseña ha sido bloqueada.

Ademas, si contiene una exclamacion o un asterisco, significa que el usuario no podra usar la contraseña para iniciar sesion (pero puede usar otro medio).

3. **User ID numerico:** ID del usuario.
4. **Group ID numerico:** ID del grupo al que pertenece.
5. **Nombre de usuario o campo de comentario:** Este campo sirve para poder poner un comentario sobre el usuario (por ejemplo: accion que realiza, para evitar confusion con dos usuarios similares, etc).
6. **Directorio home del usuario:** Directorio que será el home privado del usuario. Además sirve para poner la variable de entorno “\$HOME”
7. **Interprete opcional de comando de usuario:** Shell que usará el usuario por defecto (bash, sh, zsh, fish, etc). Ademas, pondra la variable de entorno “\$SHELL” a este valor.

## 1.2. /etc/group

Este fichero está formado por 4 campos separados por “:”. EL significado de cada campo es el siguiente:

1. **Nombre del grupo:** Nombre del grupo. Este nombre debe ser único en el sistema.
2. **Contraseña:** Contraseña del grupo. Si es una letra “x” minuscúla significa que la contraseña encriptada se encuentra en “/etc/gpasswd”.
3. **Group ID:** Indica el ID del grupo. Este valor debe ser único en el sistema.
4. **Usuarios:** Lista de usuarios separados por coma (“,”) los cuales son miembros del grupo.

## 1.3. /etc/shadow

Este fichero está formado por líneas de 9 campos separados por “:”. Los campos y sus significados son los siguientes:

1. **login name (nombre de login):** Nombre de la cuenta del usuario. Debe existir en el sistema.
2. **encrypted password (contraseña encriptada):** Contraseña encriptada del usuario especificado en “login name”. Si este campo está vacío, significa que ese usuario no requiere contraseña para iniciar sesion.

Además, en caso de que la contraseña comience con una exclamacion (“!”), significa que la contraseña ha sido bloqueada.

Por último, si la contraseña contiene el carácter de exclamacion mencionado anteriormente o asterisco (“\*”), significa que no puede iniciar sesión (si es exclamación también se cumple lo de arriba).

3. **date of last password change (fecha del ultimo cambio de contraseña):** El último cambio de contraseña, expresado como el numero de dias desde el epoch (1 de enero de 1970). Además, si el valor es 0 significa que el usuario debe cambiar la contraseña en el proximo login. En cambio, si el campo está vacío signica que las contraseñas no tienen edad (y por tanto no se cumplen estas restricciones).
4. **minimum password age (edad minima de la contraseña):** Numero de dias que el usuario tiene que esperar antes de poder cambiar la contraseña de nuevo. Un valor 0 ó vacío indica que no hay un minimo de dias.
5. **maximum password age (edad maxima de la contraseña):** Numero maximo de dias en los cuales la contraseña “caduca” (tiene que cambiarla). Al pasar este numero de dias, el sistema pedira al usuario que cambie la contraseña.

Si el valor maximo es mayor que el del campo anterior, el usuario no podra cambiar su contraseña.

Por último, si el campo está vacío, se deshabilitara este servicio junto con “password warning period” y “password inactivity period”.

6. **password warning period (periodo de advertencia de la contraseña):** El numero de dias antes de que la contraseña “caduque” durante los cuales se le advierte al usuario.  
Un valor 0 o cadena vacia indica que no habra advertencias.
7. **password inactivity period (periodo de inactividad de la contraseña):** Numero de dias despues de que la contraseña haya “caducado” en el cual eberia ser aceptada. Al pasar este eriodo, el usuario no podra iniciar sesion.  
Un campo vacio indica que no se cumple esta regla.
8. **account expiration date (fecha de expiracion de la cuenta):** La fecha en la que la cuenta expira. Esta fecha se exresa como el numero de dias desde el epoch.  
La diferencia con la expiracion de una contraseña es que, si la cuenta expira, no podra iniciar sesion de ninguna forma, mientras que si la contraseña expira, tendrá otros medios para iniciar sesion.  
El campo vacio indica que la cuenta no expira. Ademas, no se debe usar el valor 0 ya que se puede interpretar como que la cuenta expira en el epoch o que no expira.
9. **reserved field (campo reservado):** Este campo está reservado para usos futuros.

#### 1.4. /etc/gshadow

Este fichero está tambien formado por 4 campos separados por el símbolo “:”. El significado de cada campo es el siguiente:

1. **Nombre del grupo:** Nombre del grupo. Debe existir en el sistema.
2. **Contraseña encriptada:** Contraseña encriptada que sirve para que un usuario que no es miembro del grupo obtenga los permisos.  
Si el campo está vacio, entonces cualquier usuario puede obtener los privilegios del grupo.  
Si la contraseña comienza por una exclamacion, significa que esta está bloqueada.  
Si contiene una exclamacion o asterisco, los usuarios no podran acceder al grupo si no estan en el.
3. **Administradores:** Lista de usuarios separados por coma que puede realizar operaciones como cambiar la contraseña del grupo o administrar los usuarios del mismo.
4. **Miembros:** Lista de usaurios separados por coma. Los miembros del grupo pueden acceder al mismo sin necesitar la contraseña.

## 2. Segundo ejercicio

En este ejercicio se pide modificar el valor de la variable “LOGIN\_TIMEOUT” y comprobar sus efectos con un usuario nuevo que se haya creado manualmente.

Para ello, modifiko la variable, que estaba por defecto a 60 segundos:

```
GNU nano 6.2 /etc/login.defs
#SYS_GID_MIN      100
#SYS_GID_MAX      999
#
# Max number of login retries if password is bad. This will most likely be
# overridden by PAM, since the default pam_unix module has it's own built
# in of 3 retries. However, this is a safe fallback in case you are using
# an authentication module that does not enforce PAM_MAXTRIES.
#
LOGIN_RETRIES      5
#
# Max time in seconds for login
#
LOGIN_TIMEOUT      60
#
# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone). If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
#
CHFN_RESTRICT      rwh
#
# Should login be allowed if we can't cd to the home directory?
# Default in no.
#
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación  M-U Deshacer  M-A Poner marca
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^/ Ir a línea  M-E Rehacer   M-G Copiar
```

Figura 1: Valor por defecto.

Y lo cambio a otro valor, por ejemplo, 5 segundos:

```
GNU nano 6.2 /etc/login.defs
#
# Max number of login retries if password is bad. This will most likely be
# overridden by PAM, since the default pam_unix module has it's own built
# in of 3 retries. However, this is a safe fallback in case you are using
# an authentication module that does not enforce PAM_MAXTRIES.
#
LOGIN_RETRIES      5
#
# Max time in seconds for login
#
LOGIN_TIMEOUT      5
#
# Which fields may be changed by regular users using chfn - use
# any combination of letters "frwh" (full name, room number, work
# phone, home phone). If not defined, no changes are allowed.
# For backward compatibility, "yes" = "rwh" and "no" = "frwh".
#
CHFN_RESTRICT      rwh
#
# Should login be allowed if we can't cd to the home directory?
#
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación  M-U Deshacer  M-A Poner marca
^X Salir      ^R Leer fich. ^_ Reemplazar  ^U Pegar      ^J Justificar ^/ Ir a línea  M-E Rehacer   M-G Copiar
```

Figura 2: Valor cambiado a 5 segundos.

A continuacion, creo el usuario llamado “prueba”, le cambio la contraseña y hago login con él desde la terminal. Cuando se encuentre en la parte de pedir la contraseña de este usuario nuevo, se espera un tiempo hasta que la terminal devuelva un mensaje:

```

root@andres-kvm:~# useradd prueba
root@andres-kvm:~# passwd prueba
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
root@andres-kvm:~# login prueba
Contraseña:
El acceso caducó después de 5 segundos.
root@andres-kvm:~# █

```

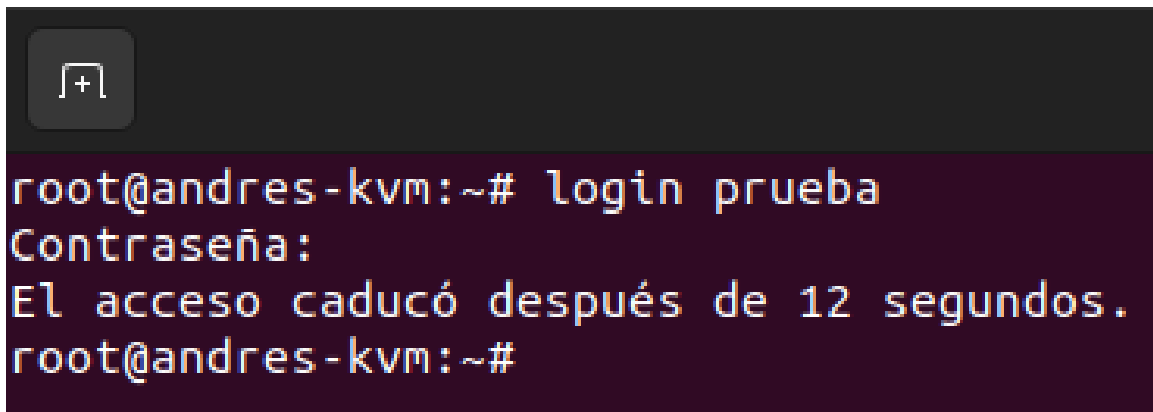
Como se puede ver, pone que han pasado 5 segundos y el acceso ha caducado. Ahora, pruebo con otro valor, por ejemplo **12** segundos:

```

GNU nano 6.2
UID_MAX                60000
# System accounts
#SYS_UID_MIN           100
#SYS_UID_MAX           999
#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                1000
GID_MAX                60000
# System accounts
#SYS_GID_MIN           100
#SYS_GID_MAX           999
#
# Max number of login retries if password is bad. This will most likely be
# overridden by PAM, since the default pam_unix module has it's own built
# in of 3 retries. However, this is a safe fallback in case you are using
# an authentication module that does not enforce PAM_MAXTRIES.
#
LOGIN_RETRIES           5
#
# Max time in seconds for login
#
LOGIN_TIMEOUT           12

```

El intento iniciar sesión de nuevo con el usuario “prueba” y espero en la parte de la contraseña.



```
root@andres-kvm:~# login prueba
Contraseña:
El acceso caducó después de 12 segundos.
root@andres-kvm:~#
```

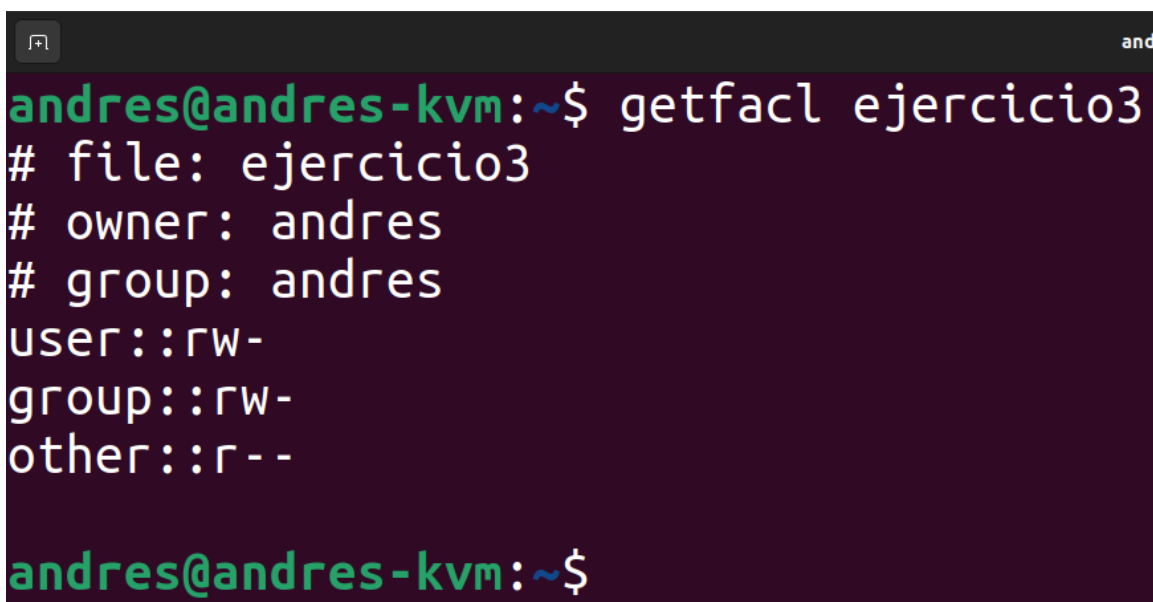
Como se puede ver, el timeout ahora es distinto.

### 3. Tercer ejercicio

En este ejercicio se pide crear un archivo y darle, mediante un ACL, permisos de lectura y escritura al usuario creado (en mi caso sigue siendo “prueba”).

Para ello, mediante la orden “touch” creo el archivo denominado “ejercicio3”.

Ahora bien, al menos en Ubuntu 22.04 no están las ordenes “getacl/setacl”, sino que se llaman “getfacl/setfacl”. El resultado no varía y tienen las mismas sintaxis.



```
andres@andres-kvm:~$ getfacl ejercicio3
# file: ejercicio3
# owner: andres
# group: andres
user::rw-
group::rw-
other::r--

andres@andres-kvm:~$
```

Figura 3: se puede ver que solo el usuario “andres” tiene los permisos

Ahora, con la orden `setfacl -m u:prueba:rw ejercicio3` se le dará al usuario “prueba” permisos “rw”. Y de nuevo mostramos con “getfacl” el archivo anterior:

```
andres@andres-kvm: ~$ setfacl -m u:prueba:rw ejercicio3
andres@andres-kvm: ~$ getfacl ejercicio3
# file: ejercicio3
# owner: andres
# group: andres
user::rw-
user:prueba:rw-
group::rw-
mask::rw-
other::r--

andres@andres-kvm: ~$
```

Figura 4: se puede ver que solo el usuario “andres” tiene los permisos

Como se puede observar, ahora aparece una línea que indica que el usuario “prueba” tiene permisos “rw”.

## 4. Ejercicio 4

Con el comando “ls” muestro los archivos que se encuentran en el directorio “/etc/pam.d”:

```
andres@andres-kvm: ~$ ls /etc/pam.d
chfn          common-session-noninteractive  gdm-smartcard          passwd         sudo
chpasswd      cron                          gdm-smartcard-pkcs11-exclusive  polkit-1       sudo-i
chsh          cups                          gdm-smartcard-sssd-exclusive    ppp            su-l
common-account  gdm-autologin                gdm-smartcard-sssd-or-password  runuser
common-auth     gdm-fingerprint              login                     runuser-l
common-password gdm-launch-environment        newusers                  sshd
common-session  gdm-password                  other                     su
```

A continuación explicare dos archivos:

### 4.1. /etc/pam.d/chfn

Permite cambiar la información personal de un usuario tales como: el nombre, el número de teléfono, de habitación, etc. Estos datos luego pueden ser leídos por comandos como “finger”.

El contenido del archivo es:

```
root@andres-kvm:/etc/pam.d# batcat chfn

File: chfn

1  #
2  # The PAM configuration file for the Shadow `chfn' service
3  #
4
5  # This allows root to change user information without being
6  # prompted for a password
7  auth      sufficient  pam_rootok.so
8
9
10 # The standard Unix authentication modules, used with
11 # NIS (man nsswitch) as well as normal /etc/passwd and
12 # /etc/shadow entries.
13 @include common-auth
14 @include common-account
15 @include common-session
16

root@andres-kvm:/etc/pam.d#
```

La funcion de la linea 7 es para no pedir la contraseña al usuario root cuando esté usando este comando. Para ello, hace uso del campo de control `sufficient`, que hace que si tiene éxito retorne sin ejecutar mas modulos. Ademas, hace uso del modulo “`pam_rootok.so`” que hace que solo tenga éxito si el usuario tiene el UID a 0 (es el usuario root).

## 4.2. /etc/pam.d/chsh

El comando “`chsh`” permite cambiar la shell por defecto del usuario que lo invoca. Si no se le pasa ningun parametro se activa el modo interactivo para realizar el cambio de shell.

El contenido del archivo es el siguiente:

```
root@andres-kvm:~# batcat /etc/pam.d/chsh

File: /etc/pam.d/chsh

1  #
2  # The PAM configuration file for the Shadow `chsh' service
3  #
4
5  # This will not allow a user to change their shell unless
6  # their current one is listed in /etc/shells. This keeps
7  # accounts with special shells from changing them.
8  auth      required    pam_shells.so
9
10 # This allows root to change user shell without being
11 # prompted for a password
12 auth      sufficient  pam_rootok.so
13
14 # The standard Unix authentication modules, used with
15 # NIS (man nsswitch) as well as normal /etc/passwd and
16 # /etc/shadow entries.
17 @include common-auth
18 @include common-account
19 @include common-session
20

root@andres-kvm:~#
```

Como se puede ver en la linea 8, esta llamada lo que hace es prohibir el cambio de shell a no ser que se encuentre listada en “`/etc/shells`”. Esto se consigue mediante el campo de control “`required`”, que provocará un fallo de autenticacion en el sistema (ejecutará la linea siguiente, pero al ser irrelevante, no pasa nada) si el modulo falla. Tambien se consigue mediante la llamada al modulo “`pam_shells.so`”, que hace que si la shell pasada como parametro no se encuentra en “`/etc/shells`” dé un fallo.

La funcion de la linea 12 es de permitir al superusuario cambiar la shell sin ser necesario introducir la contraseña. Esto se realiza mediante el campo de control “`sufficient`” y el modulo “`pam_rootok.so`”. Con “`sufficient`”, cuando la orden tiene éxito retorna sin ejecutar los demas modulos **COMPROBAR AFIRMACION: (Como es el ultimo puede retornar sin problema).**



Ademas, con el modulo “pam\_rootok.so” autoriza al usuario con el UID 0 (root).

## 5. Quinto ejercicio

### 5.1. Apartado a

Es necesario modificar el archivo PAM “common-password” y en Ubuntu 22.04 ya como primera linea aparece el uso del modulo “pam\_pwquality”.

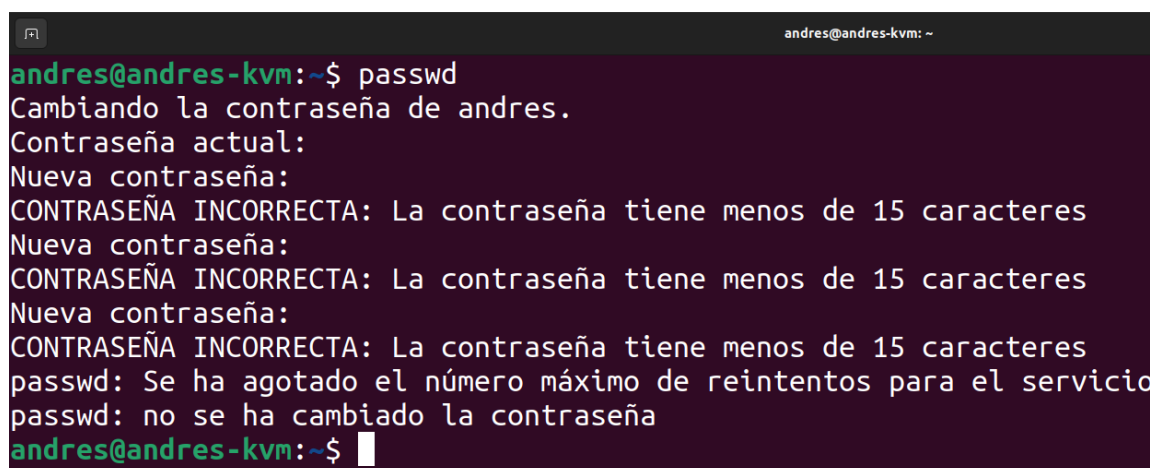
Ahora bien si leemos el manual de este modulo con “man 8 pam\_pwquality” se puede ver que hay un argumento denominado “minlen” y que valor por defecto es 8. No obstante, no se puede bajar del valor 4, ya que es un limite que tiene “Cracklib” y mostrara que la contraseña es muy corta. Por eso, voy a poner el limite a **15 caracteres**.



```
GNU nano 6.2 /etc/pam.d/common-password
## To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password            requisite pam_pwquality.so retry=3 minlen=15
```

Y ahora al usar el comando “passwd” y poner una contraseña con menos de 15 palabras, muestra un error:



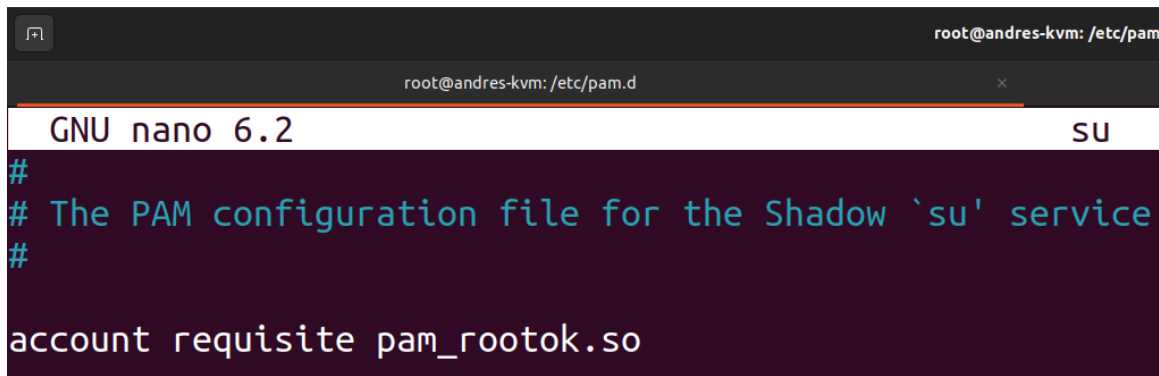
```
andres@andres-kvm:~$ passwd
Cambiando la contraseña de andres.
Contraseña actual:
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 15 caracteres
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 15 caracteres
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 15 caracteres
passwd: Se ha agotado el número máximo de reintentos para el servicio
passwd: no se ha cambiado la contraseña
andres@andres-kvm:~$
```

Y al agotarse los intentos (que son 3) se sale del programa.

### 5.2. Apartado b

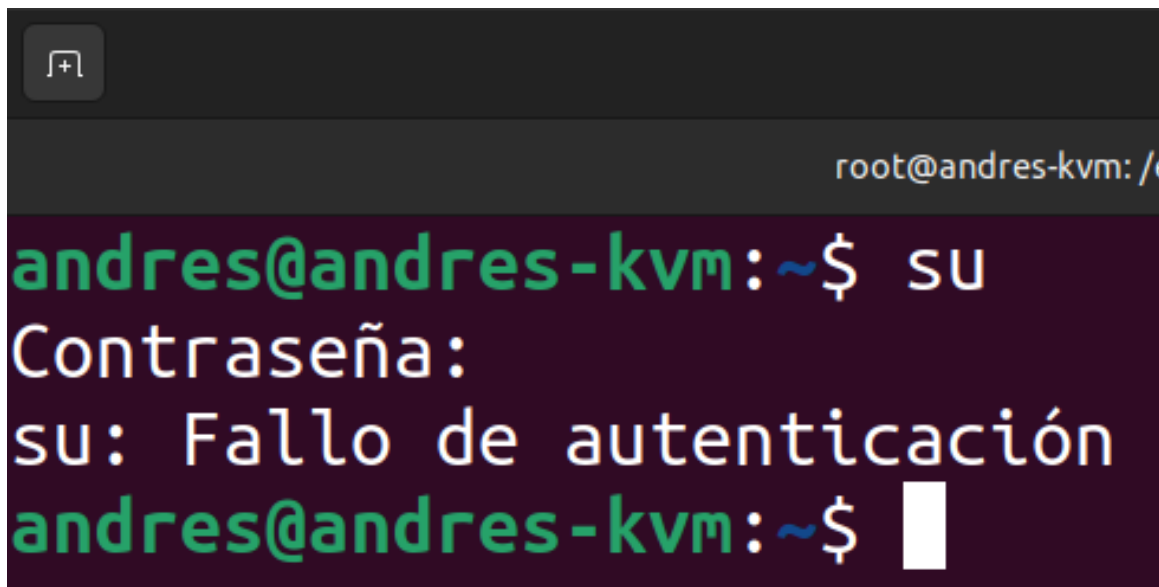
En esta parte he restringido el acceso al comando “su” para así evitar que un usuario que ponga el comando sin “sudo” pueda entrar. Si ponen “sudo su” sí van a poder entrar, pero esto es así ya que son usuarios administradores (y es una decisión de diseño, ya que en otro caso no podría usar nadie “sudo”), en ese caso lo recomendable es deshabilitar el acceso al grupo “sudo” (en el caso de Ubuntu) para que no lo pueda usar (editando el archivo sudoers mediante el comando “visudo”).

Para conseguir esto, es necesario modificar el archivo “/etc/pam.d/su” y añadir la siguiente línea al principio:



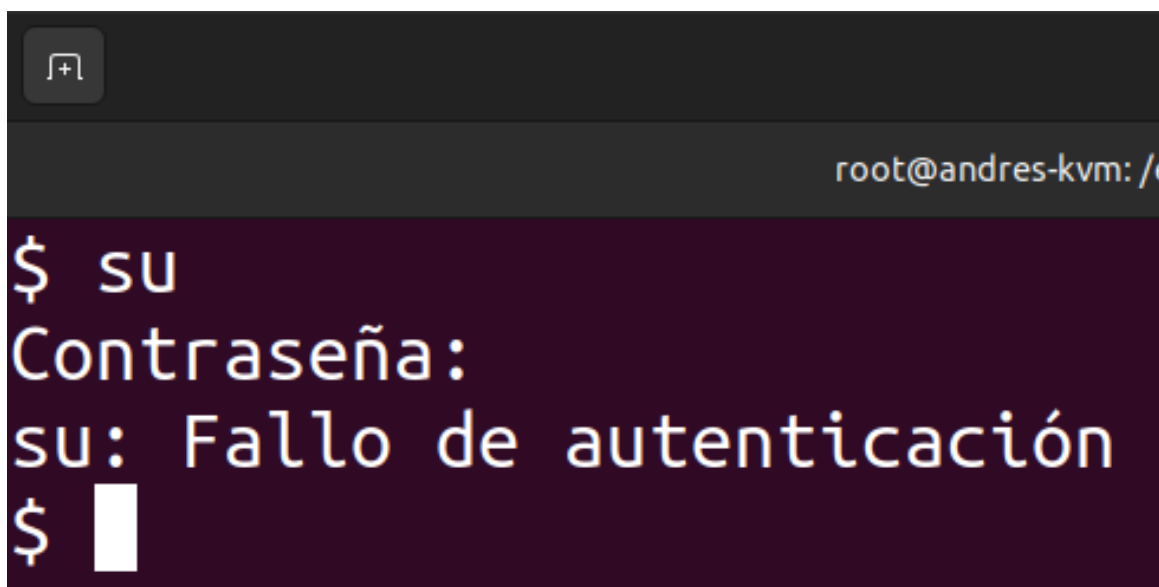
```
root@andres-kvm: /etc/pam.d
GNU nano 6.2 SU
#
# The PAM configuration file for the Shadow `su' service
#
account requisite pam_rootok.so
```

Ahora, al ejecutar el comando “su” con un usuario normal aparece lo siguiente:



```
root@andres-kvm: /etc/pam.d
andres@andres-kvm: ~$ su
Contraseña:
su: Fallo de autenticación
andres@andres-kvm: ~$
```

En cambio, si el usuario puede usar “sudo”, si puede acceder.



```
root@andres-kvm: /etc/pam.d
$ su
Contraseña:
su: Fallo de autenticación
$
```

Figura 5: el usuario que se ha creado en esta practica no tiene permisos de sudo y por tanto no puede entrar de ninguna manera

La línea que he añadido lo que hace es comprobar que la cuenta sea root (UID=0) y en caso de no serlo, no sigue ejecutando el archivo provocando un error de autenticación.

## 6. Sexto ejercicio

En Ubuntu 22.04 los cambios de contraseña no se almacenan en “/var/log/messages”, sino en “/var/log/auth.log”. [Enlace](#) a la guía.

Voy a crear el usuario “ejercicio6” y le voy a cambiar la contraseña:

```
root@andres-kvm:~# useradd ejercicio6
root@andres-kvm:~# passwd ejercicio6
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
root@andres-kvm:~# su ejercicio6
$ passwd
Cambiando la contraseña de ejercicio6.
Contraseña actual:
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
$
```

Y ahora, al mostrar el archivo “/var/log/auth.log” aparecen las siguientes líneas:

```
41 Sep 21 20:05:12 andres-kvm useradd[2779]: new group: name=ejercicio6, GID=1002
42 Sep 21 20:05:12 andres-kvm useradd[2779]: new user: name=ejercicio6, UID=1002, GID=1002, home=/home/ejercicio6, s
hell=/bin/sh, from=/dev/pts/1
43 Sep 21 20:05:30 andres-kvm passwd[2788]: pam_unix(passwd:chauthtok): password changed for ejercicio6
44 Sep 21 20:05:30 andres-kvm passwd[2788]: gkr-pam: couldn't update the login keyring password: no old password was
entered
45 Sep 21 20:05:34 andres-kvm su: (to ejercicio6) root on pts/1
46 Sep 21 20:05:34 andres-kvm su: pam_unix(su:session): session opened for user ejercicio6(uid=1002) by andres(uid=0
)
47 Sep 21 20:05:46 andres-kvm passwd[2792]: pam_unix(passwd:chauthtok): password changed for ejercicio6
48 Sep 21 20:05:46 andres-kvm passwd[2792]: gkr-pam: unable to locate daemon control file
49 Sep 21 20:06:39 andres-kvm su: pam_unix(su:session): session closed for user ejercicio6
(END)
```

## 7. Séptimo ejercicio

Para empezar, el propio archívo de sudoers recomienda usar la orden “visudo”. Por tanto, es necesario usar “visudo” para que no haya problemas después. Además, por defecto usa el editor “vi”, esto se puede cambiar usando el comando siguiente:

```
EDITOR=nano visudo
```

Ahora, voy a asignarle permisos para usar sudo al usuario “prueba” que no se encuentra en el grupo “sudo”, que es el que usa Ubuntu para dar permisos.

```

andres@andres-kvm: ~
$ whoami
prueba
$ groups
prueba
$ sudo su
[sudo] contraseña para prueba:
prueba no está en el archivo sudoers. Se informará de este incidente.
$

```

Si añadimos la siguiente línea en el archivo “sudoers” tendremos acceso con sudo:

```

root@andres-kvm: /home/andres
GNU nano 6.2 /etc/sudoers.tmp
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
prueba  ALL=(ALL:ALL) ALL

```

Y ahora al hacer una prueba se puede ver que ya funciona:

```

$ whoami
prueba
$ groups
prueba
$ sudo su
[sudo] contraseña para prueba:
root@andres-kvm: /# whoami
root
root@andres-kvm: /#

```

## 8. Octavo ejercicio

Voy a examinar cada uno de los archivos y comprobar que se registran eventos que realizaré. Para ello, voy a dividir la explicación en subsecciones para cada uno de los archivos.

## 8.1. /var/log/lastlog

Este archivo almacena informacion sobre el ultimo inicio de sesion de los usuarios. Para acceder a la informacion es necesario utilizar el comando “lastlog”:

```
root@andres-kvm: ~  
tss **Nunca ha accedido**  
uidd **Nunca ha accedido**  
systemd-oom **Nunca ha accedido**  
tcpdump **Nunca ha accedido**  
avahi-autoipd **Nunca ha accedido**  
usbmux **Nunca ha accedido**  
dnsmasq **Nunca ha accedido**  
kernoops **Nunca ha accedido**  
avahi **Nunca ha accedido**  
cups-pk-helper **Nunca ha accedido**  
rtkit **Nunca ha accedido**  
whoopsie **Nunca ha accedido**  
sssd **Nunca ha accedido**  
speech-dispatcher **Nunca ha accedido**  
nm-openvpn **Nunca ha accedido**  
saned **Nunca ha accedido**  
colord **Nunca ha accedido**  
geoclue **Nunca ha accedido**  
pulse **Nunca ha accedido**  
gnome-initial-setup **Nunca ha accedido**  
hplip **Nunca ha accedido**  
gdm **Nunca ha accedido**  
andres **Nunca ha accedido**  
prueba pts/1 vie sep 23 19:56:14 +0200 2022  
ssh **Nunca ha accedido**  
ejercicio6 **Nunca ha accedido**  
root@andres-kvm:~#
```

Como se puede ver, el usuario “ejercicio6” nunca ha iniciado sesion en el sistema. Ahora si inicio sesion y vuelvo a usar la orden “lastlog” aparece lo siguiente:

```
root@andres-kvm: ~  
tss **Nunca ha accedido**  
uidd **Nunca ha accedido**  
systemd-oom **Nunca ha accedido**  
tcpdump **Nunca ha accedido**  
avahi-autoipd **Nunca ha accedido**  
usbmux **Nunca ha accedido**  
dnsmasq **Nunca ha accedido**  
kernoops **Nunca ha accedido**  
avahi **Nunca ha accedido**  
cups-pk-helper **Nunca ha accedido**  
rtkit **Nunca ha accedido**  
whoopsie **Nunca ha accedido**  
sssd **Nunca ha accedido**  
speech-dispatcher **Nunca ha accedido**  
nm-openvpn **Nunca ha accedido**  
saned **Nunca ha accedido**  
colord **Nunca ha accedido**  
geoclue **Nunca ha accedido**  
pulse **Nunca ha accedido**  
gnome-initial-setup **Nunca ha accedido**  
hplip **Nunca ha accedido**  
gdm **Nunca ha accedido**  
andres **Nunca ha accedido**  
prueba pts/1 vie sep 23 19:56:14 +0200 2022  
ssh **Nunca ha accedido**  
ejercicio6 pts/1 vie sep 23 21:03:38 +0200 2022  
root@andres-kvm:~#
```

Ahora como se puede ver aparece la fecha del ultimo inicio de sesion del usuario “ejercicio6”.

## 8.2. /var/log/wtmp

Almacena los login y logout de los distintos usuarios del sistema. Se accede con el comando “last”.

Ahora con el comando “last –since today” muestra solo la informacion de hoy.

Y ahora voy a iniciar sesion con el usuario “prueba” y lo logout para ver como se almacena la informacion:

### 8.3. /var/log/utmp

Muestra los usuarios que estan *loggeados* en el sistema. Se puede obtener esta informacion con la orden “who”.

Ahora si inicio sesion con el usuario “prueba” deberia aparecer con el comando anterior:

### 8.4. /var/log/btmp

Muestra los intentos fallidos de inicio de sesion en el sistema. Se puede obtener con la orden “lastb”.

Ahora, si hago que fallo el inicio de sesion del usuario “prueba” deberia de salir:

### 8.5. /var/log/sudo

Este archivo, al menos en Ubuntu, no existe, por tanto no puedo mostrar informacion al respecto sobre la actividad del uso de “sudo”.

### 8.6. /var/log/messages

En Ubuntu (no sé si en otras distribuciones también) ya no existe este archivo porque duplicaba informacion con “/var/log/syslog”. Por tanto, voy a mostrar este archivo:

### 8.7. Noveno ejercicio

#### 8.8. PC de mi casa

Para comprobar los intentos de inicio de sesion fallidos en el ordenador de mi casa, voy a usar “lastb”:

Como se puede observar, el unico intento fallido de inicio de sesion que he tenido ha sido provocado por mi en el momento de hacer esta parte del ejercicio.

Ahora, muestro con el comando “last” los login y logout del sistema:

Por lo que puedo ver, no ha habido ningun inicio en el sistema por ssh (se mostraria en la tercera columna la direccion IP del que lo ha intentado).

Por ultimo, con la orden “lastlog” muestro los inicios de sesion producidos en el sistema.

Como se puede ver, no hay ninguno en el que aparezca ssh y los inicios de sesion me concuerdan, lo cual me puede indicar que el sistema no ha sido (a simple vista) comprometido.

#### 8.9. PC de prácticas (máquina virtual)

Voy a usar los mismo comandos para la maquina virtual y comprobar su seguridad. Para suponer que el sistema ha sido comprometido, he usado SSH desde el ordenador de mi casa (host) hacia la maquina virtual.

Como se puede ver, la direccion IP “192.168.122.1” ha intentado conectarse a la maquina con el usuario “andres” por SSH.

Se puede observar que alguien ha entrado al sistema con SSH usando el usuario “andres”.

Aqui tambien se puede observar que alguien ha accedido con la misma direccion IP anterior mediante SSH y ha conseguido iniciar sesion en el sistema.

Segun estos datos, suponiendo que no hubiera sido yo, se podria decir que alguien ha intentado acceder al sistema y ha conseguido iniciar sesion como el usuario “andres”. Una vez sacada esta conclusion, lo recomendable es detectar los cambios que ha realizado en el sistema y el trafico de red para ver que posibles datos se ha podido llevar.