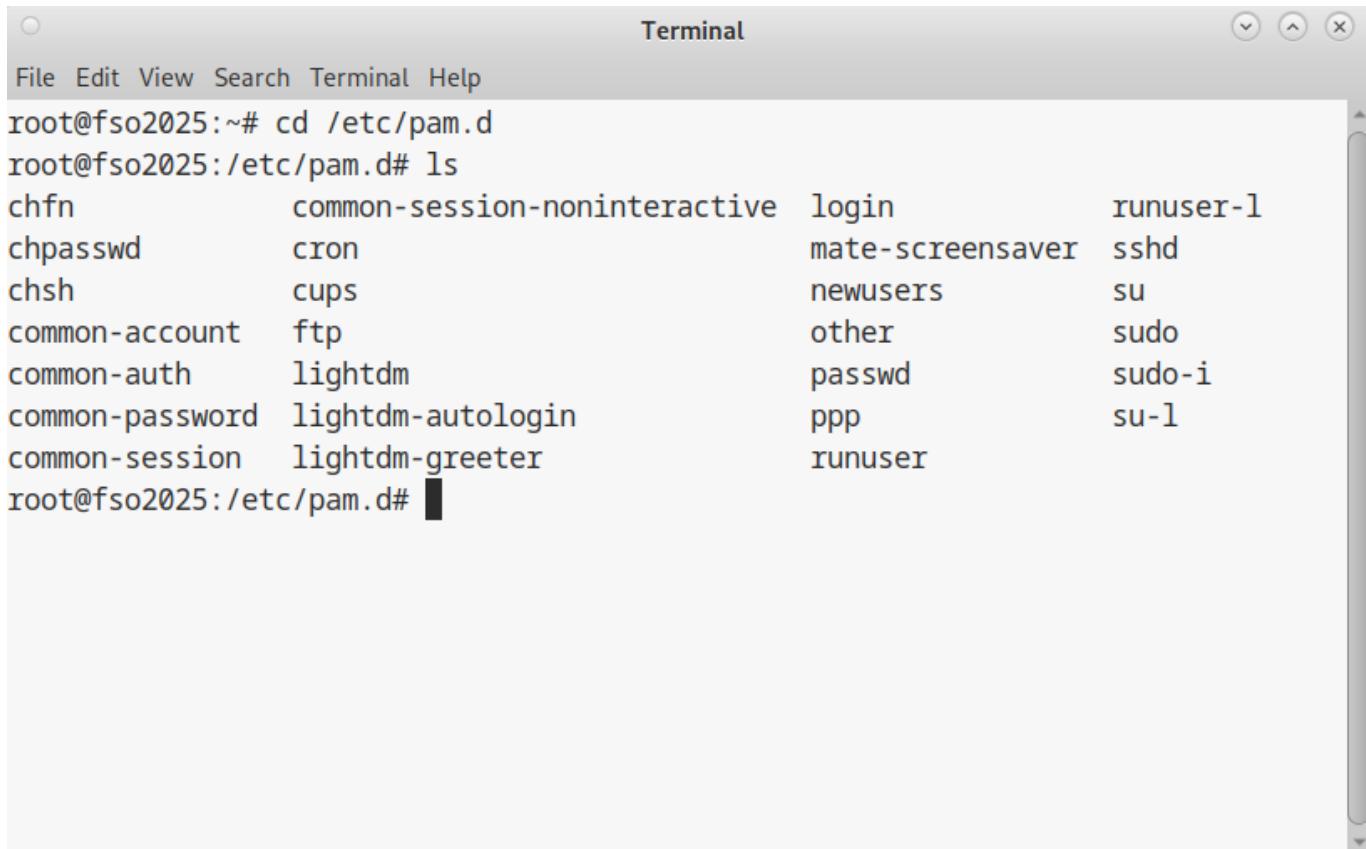


[FORT] Práctica 4: Securizando las cuentas de usuario

1. Deshabilita el login a root, tanto en el Display Manager como en las Terminales Virtuales excepto tty3

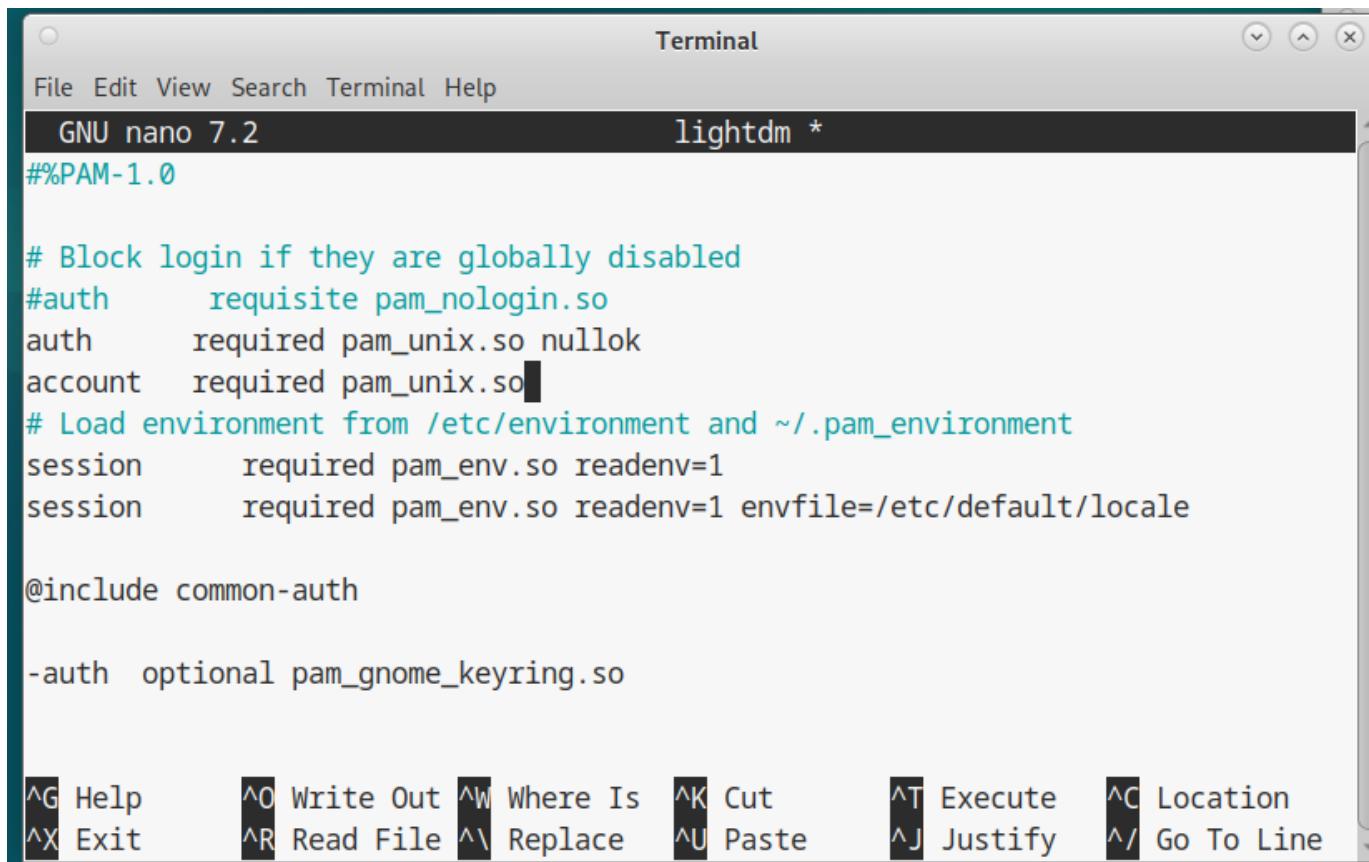


The screenshot shows a terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the following command and its output:

```
root@fso2025:~# cd /etc/pam.d
root@fso2025:/etc/pam.d# ls
chfn          common-session-noninteractive  login          runuser-1
chpasswd       cron                         mate-screensaver sshd
chsh          cups                         newusers       su
common-account  ftp                         other          sudo
common-auth    lightdm                      passwd        sudo-i
common-password lightdm-autologin          ppp           su-1
common-session  lightdm-greeter            runuser
root@fso2025:/etc/pam.d#
```

Para deshabilitar estos permisos vamos a modificar los archivos de PAM que se encuentran en /etc/pam.d. Para deshabilitar el acceso a root en la interfaz gráfica modificamos el archivo lightdm con las siguientes líneas:

```
auth required pam_unix.so nullok
account required pam_unix.so
```



```

Terminal
File Edit View Search Terminal Help
GNU nano 7.2          lightdm *
#%PAM-1.0

# Block login if they are globally disabled
#auth      requisite pam_nologin.so
auth      required pam_unix.so nullok
account  required pam_unix.so
# Load environment from /etc/environment and ~/.pam_environment
session   required pam_env.so readenv=1
session   required pam_env.so readenv=1 envfile=/etc/default/locale

@include common-auth

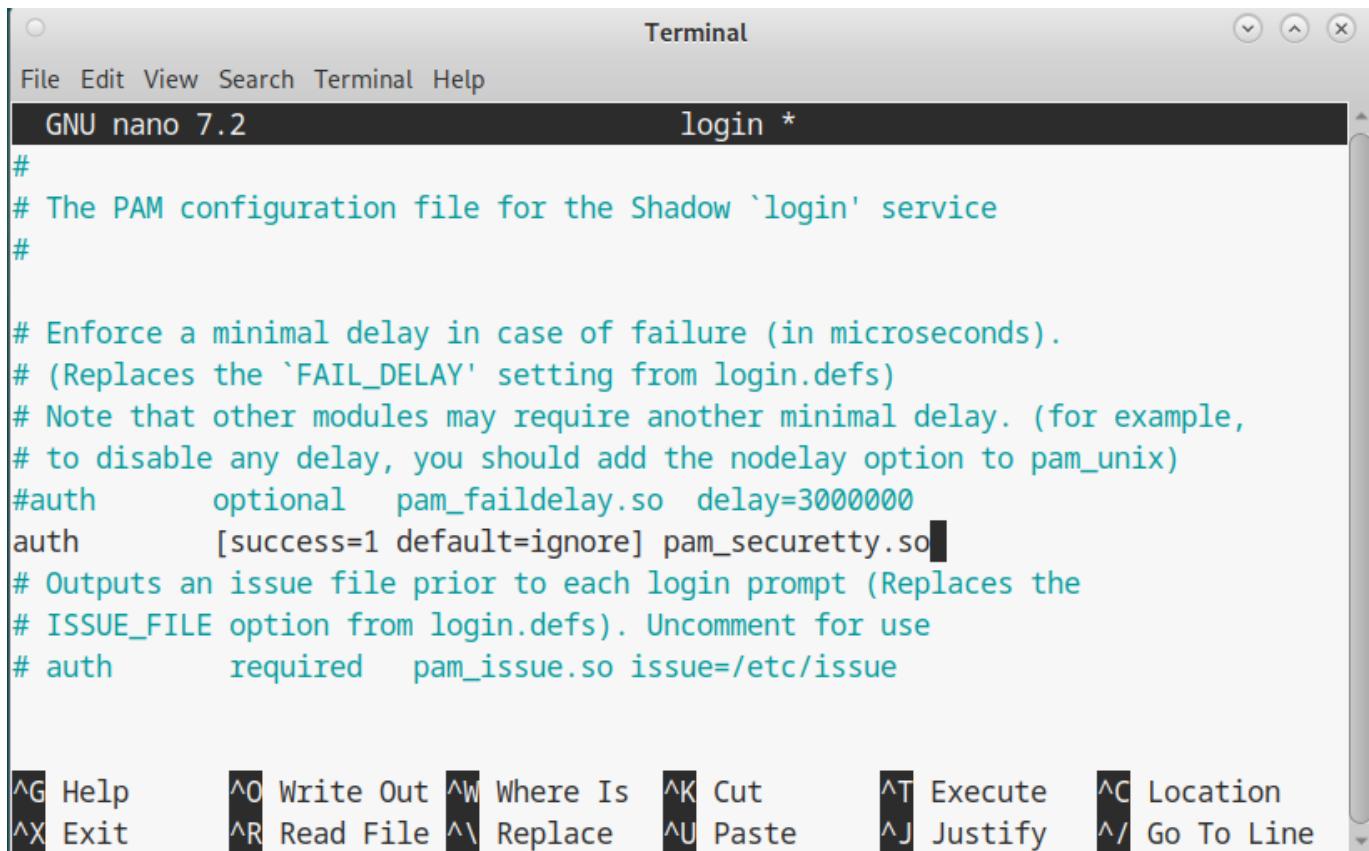
-auth optional pam_gnome_keyring.so

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Para deshabilitar los permisos en las terminales virtuales modificamos el archivo login con lo siguiente:

```
auth [success=1 default=ignore] pam_securetty.so
```



```

Terminal
File Edit View Search Terminal Help
GNU nano 7.2          login *
#
# The PAM configuration file for the Shadow 'login' service
#
# Enforce a minimal delay in case of failure (in microseconds).
# (Replaces the 'FAIL_DELAY' setting from login.defs)
# Note that other modules may require another minimal delay. (for example,
# to disable any delay, you should add the nodelay option to pam_unix)
#auth      optional  pam_faildelay.so  delay=3000000
auth      [success=1 default=ignore] pam_securetty.so
# Outputs an issue file prior to each login prompt (Replaces the
# ISSUE_FILE option from login.defs). Uncomment for use
# auth      required   pam_issue.so issue=/etc/issue

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

```

2. Usando el módulo pam_wheel.so haz que solo usuario, user001, user002, user003 y user004 puedan volverse root con SU.

Usuario no necesita saber la contraseña, mientras que los usuarios del 001 al 004 la necesitan para convertirse en root. Al resto de usuarios no se les preguntará por la contraseña.

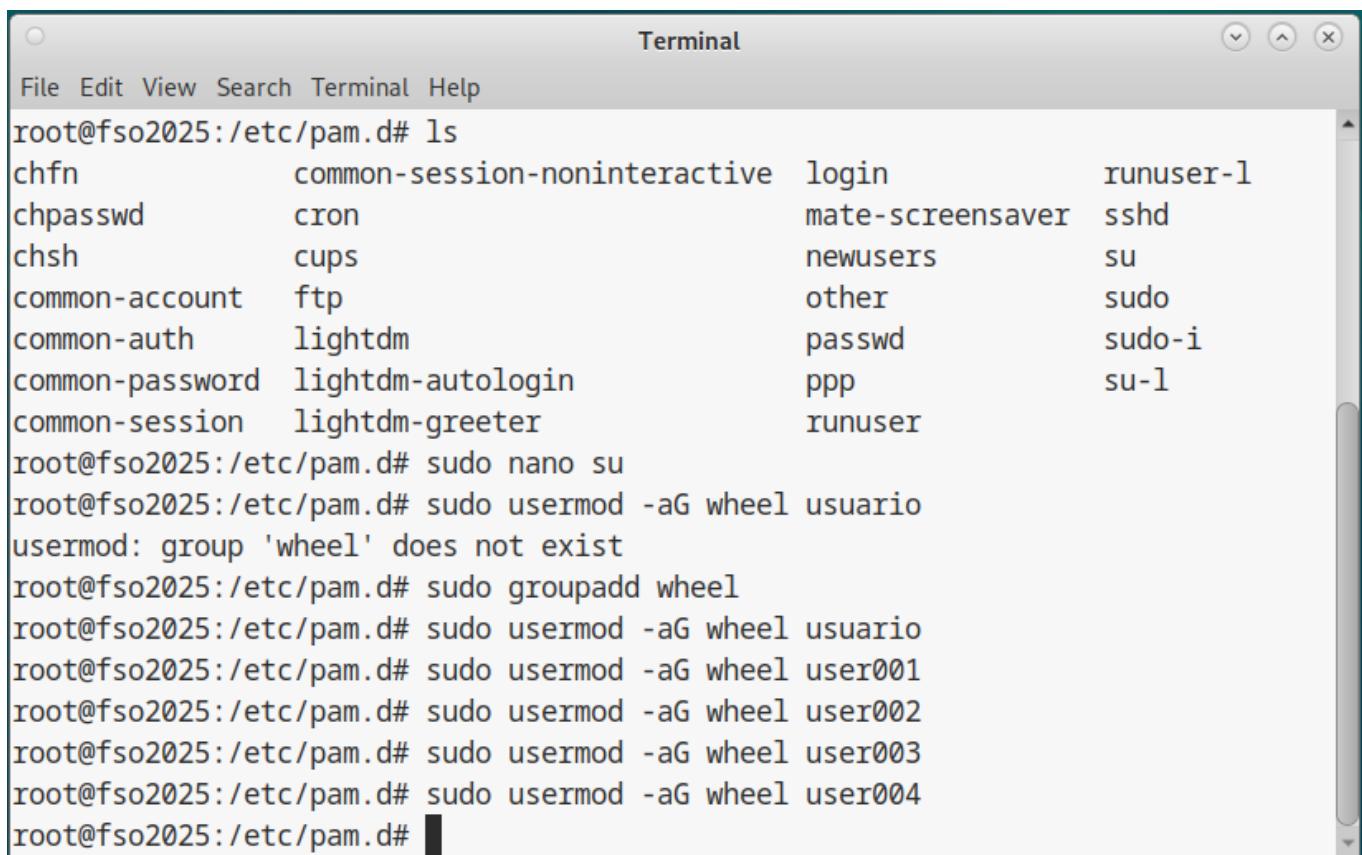
Para aplicar los ajustes debemos modificar el archivo /etc/pam.d/su con las siguientes líneas:

```
auth required pam_wheel.so group=wheel #Permite a los usuario del grupo
Wheel a usar su sin contraseña
auth [success=1 default=ignore] pam_succeed_if.so user = usuario #Permite al
usuario "usuario" usar su sin contraseña
auth required pam_unix.so #se configura para permitir que los usuarios
indicados tengan que poner la contraseña
```

```
# The PAM configuration file for the Shadow `su' service
#
# This allows root to su without passwords (normal operation)
#auth      sufficient pam_rootok.so
auth      required pam_wheel.so group=wheel
auth      [success=1 default=ignore] pam_succeed_if.so user = usuario
auth      required pam_unix.so
# Uncomment this to force users to be a member of group wheel
# before they can use `su'. You can also add "group=foo"
# to the end of this line if you want to use a group other
# than the default "wheel" (but this may have side effect of
# denying "root" user, unless she's a member of "foo" or explicitly
```

También debemos añadir a los usuarios en cuestión al grupo wheel con el siguiente comando:

```
sudo usermod -aG wheel usuario
sudo usermod -aG wheel user001
sudo usermod -aG wheel user002
sudo usermod -aG wheel user003
sudo usermod -aG wheel user004
```

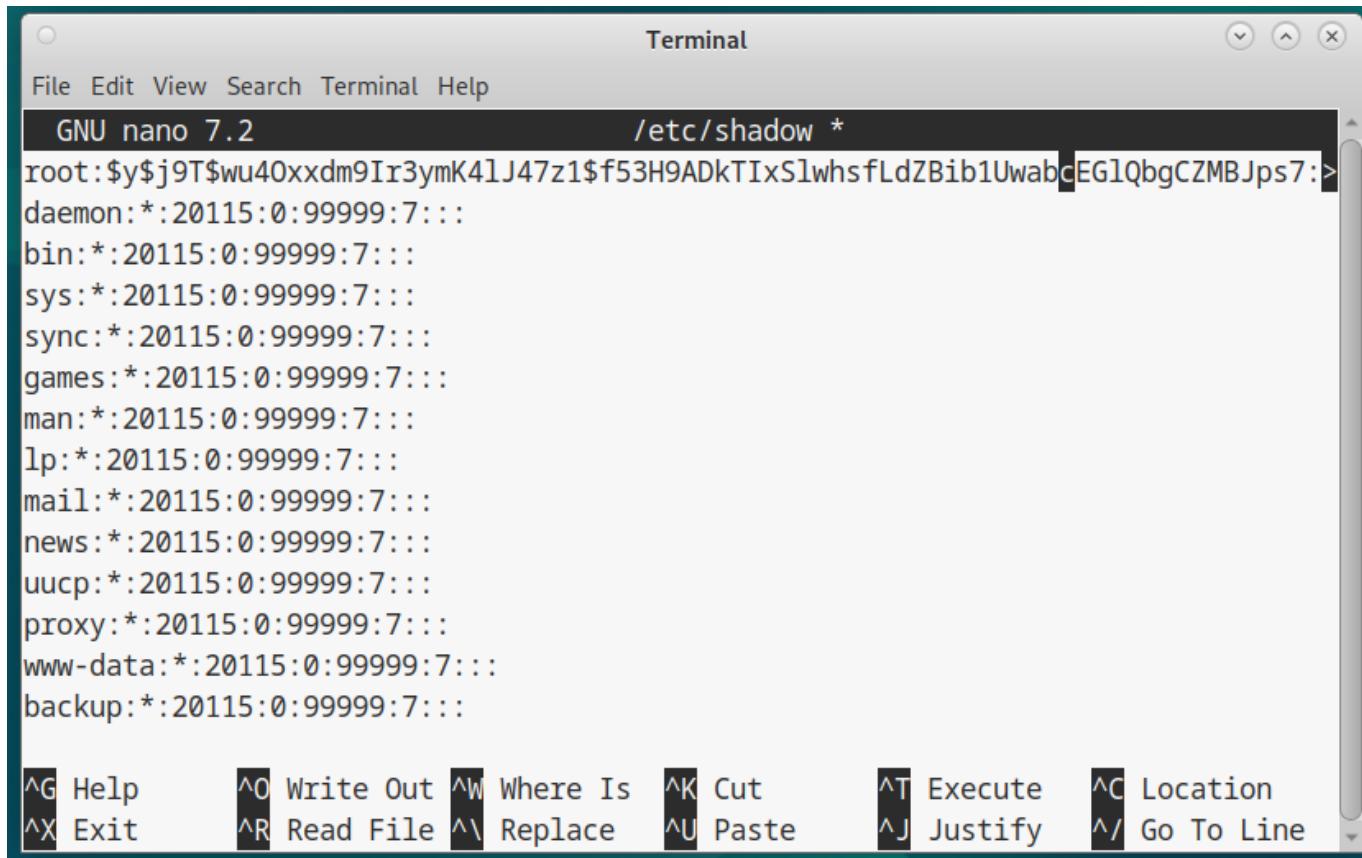


```
Terminal
File Edit View Search Terminal Help
root@fso2025:/etc/pam.d# ls
chfn          common-session-noninteractive  login          runuser-1
chpasswd       cron                         mate-screensaver sshd
chsh          cups                         newusers       su
common-account  ftp                         other          sudo
common-auth    lightdm                      passwd         sudo-i
common-password lightdm-autologin          ppp            su-l
common-session  lightdm-greeter            runuser
root@fso2025:/etc/pam.d# sudo nano su
root@fso2025:/etc/pam.d# sudo usermod -aG wheel usuario
usermod: group 'wheel' does not exist
root@fso2025:/etc/pam.d# sudo groupadd wheel
root@fso2025:/etc/pam.d# sudo usermod -aG wheel usuario
root@fso2025:/etc/pam.d# sudo usermod -aG wheel user001
root@fso2025:/etc/pam.d# sudo usermod -aG wheel user002
root@fso2025:/etc/pam.d# sudo usermod -aG wheel user003
root@fso2025:/etc/pam.d# sudo usermod -aG wheel user004
root@fso2025:/etc/pam.d#
```

3. ¿Que método de cifrado se usa para las contraseñas? ¿Ha sido usado el mismo método para todas las contraseñas en el sistema?

Deberíamos usar SHA256 y cambiar las contraseñas de todos los usuarios a SHA256

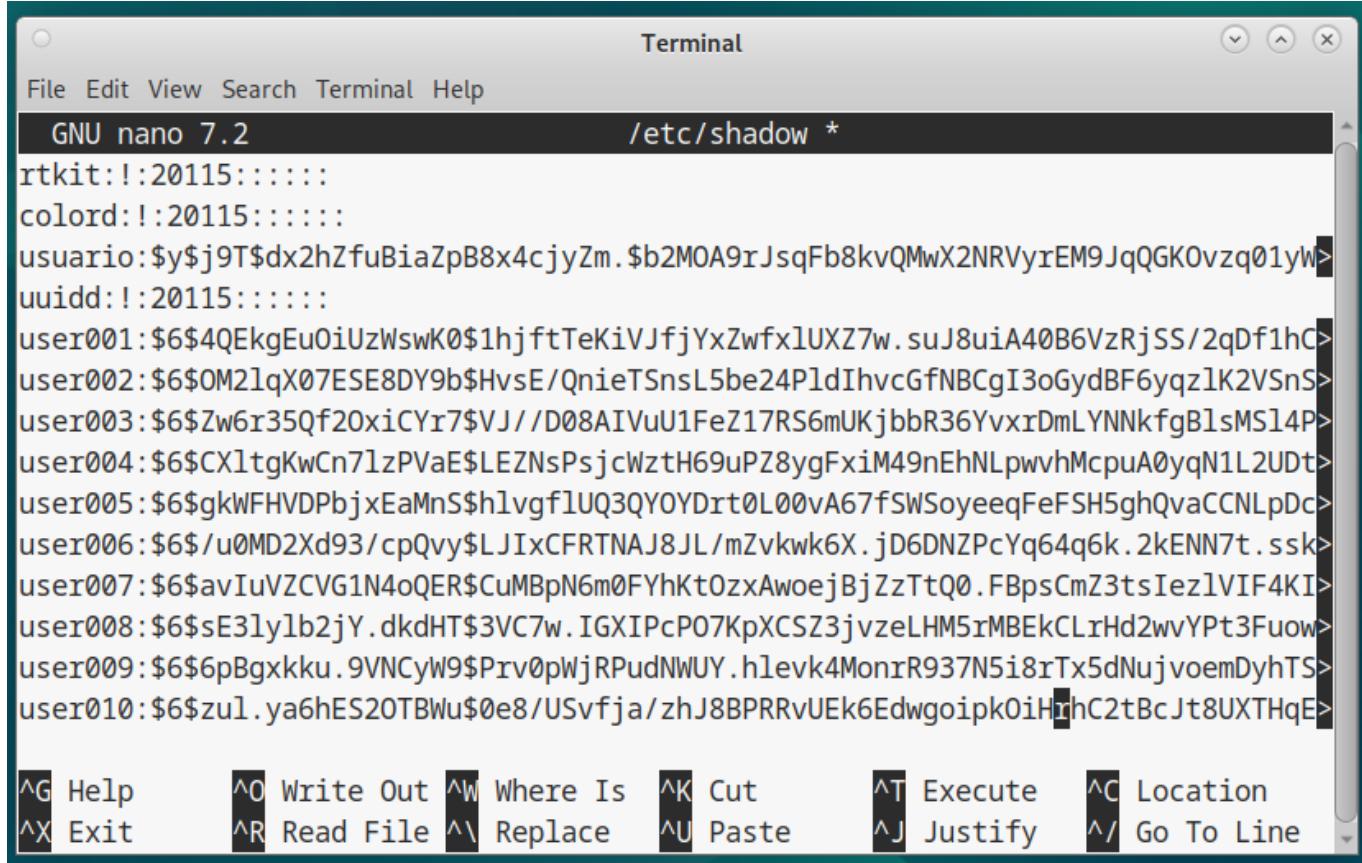
¿Como deberíamos hacer? Echando un vistazo a /etc/shadow podemos ver las contraseñas cifradas por un lado para root:



```
GNU nano 7.2 /etc/shadow *
root:$y$j9T$wu40xxdm9Ir3ymK4lJ47z1$f53H9ADkTIxSlwhsfLdZBib1UwabcEGlQbgCZMBJps7:>
daemon:*:20115:0:99999:7:::
bin:*:20115:0:99999:7:::
sys:*:20115:0:99999:7:::
sync:*:20115:0:99999:7:::
games:*:20115:0:99999:7:::
man:*:20115:0:99999:7:::
lp:*:20115:0:99999:7:::
mail:*:20115:0:99999:7:::
news:*:20115:0:99999:7:::
uucp:*:20115:0:99999:7:::
proxy:*:20115:0:99999:7:::
www-data:*:20115:0:99999:7:::
backup:*:20115:0:99999:7:::
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

y por otro lado para los demás usuarios:



```
GNU nano 7.2 /etc/shadow *
rtkit:!:20115::::::
colord:!:20115::::::
usuario:$y$j9T$dx2hZfuBiaZpB8x4cjyZm.$b2M0A9rJsqFb8kvQMwX2NRVyrEM9JqQGK0vzq01yW>
uuidd:!:20115::::::
user001:$6$4QEkgEu0iUzWswK0$1hjftTeKivJfjYxZwfxlUXZ7w.suJ8uiA40B6VzRjSS/2qDf1hC>
user002:$6$0M2lqX07ESE8DY9b$HvsE/QnieTSnsL5be24PldIhvcGfNBCgI3oGydBF6yqzlK2VSns>
user003:$6$Zw6r35Qf20xiCYr7$VJ//D08AVuU1FeZ17RS6mUKjbbR36YvxrDmLYNNkfgBlsMSl4P>
user004:$6$CXltgKwCn7lzPVaE$LEZNsPsjcWztH69uPZ8ygFxiM49nEhNLpwvhMcpuA0yqN1L2UDt>
user005:$6$gkWFHVDPbjxEaMnS$hlvgf1UQ3QY0YDrt0L00vA67fSWSoyeeqFeFSH5ghQvaCCNLpDc>
user006:$6$/u0MD2Xd93/cpQvy$LJIXCFRTNAJ8JL/mZvkwk6X.jD6DNZPcYq64q6k.2kENN7t.ssk>
user007:$6$avIuVZCVG1N4oQER$CuMBpN6m0FYhKt0zxAwoejBjZzTtQ0.FBpsCmZ3tsIezlVIF4KI>
user008:$6$sE3lylb2jY.dkdHT$3VC7w.IGXIPcP07KpXCSZ3jvzeLHM5rMBEkCLrHd2wvYPt3Fuow>
user009:$6$6pBgxkku.9VNCyW9$Prv0pWjRPudNWUY.hlevk4MonrR937N5i8rTx5dNujvoemDyhTS>
user010:$6$zul.ya6hES20TBWu$0e8/USvfja/zhJ8BPRRvUEk6Edwgoipk0iHrhC2tBcJt8UXTHqE>
```

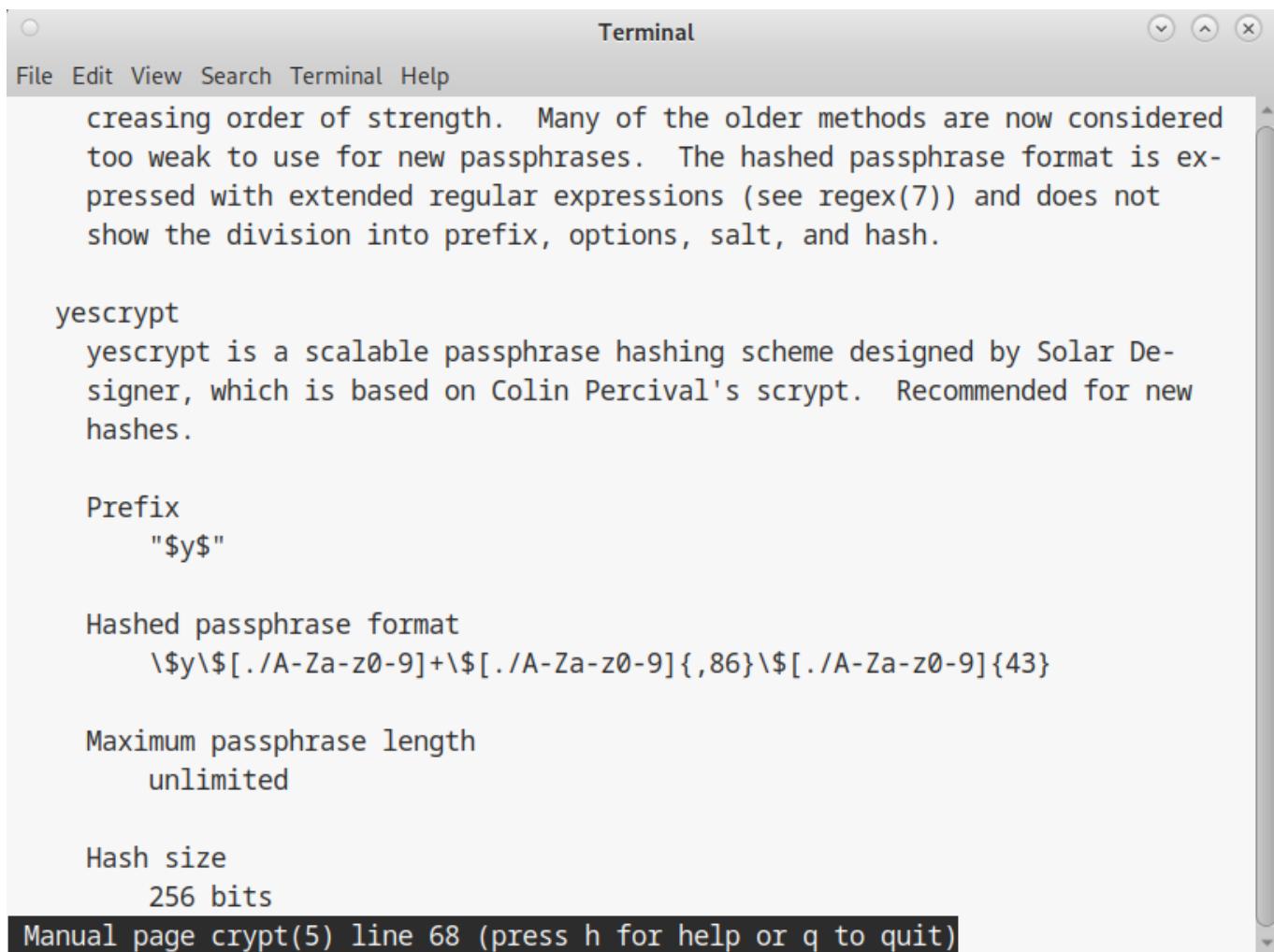
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Las contraseñas en general comienzan con una cadena `\$<caracter>\$`, estos primeros 3 caracteres

señalan en que cifrado viene cada contraseña, por lo que sabemos que:

- Root y usuario comienzan con `\y`, por lo que sabemos que usa yescript
- El resto de usuarios comienzan con `\6`, por lo que sabemos que usan SHA-512

Esto lo sabemos gracias a las salidas del comando `man 5 crypt`, donde podemos ver que es cada cifrado, por ejemplo, el de root y usuario sería el siguiente:



```
Terminal
File Edit View Search Terminal Help
creasing order of strength. Many of the older methods are now considered
too weak to use for new passphrases. The hashed passphrase format is ex-
pressed with extended regular expressions (see regex(7)) and does not
show the division into prefix, options, salt, and hash.

yescript
yescript is a scalable passphrase hashing scheme designed by Solar De-
signer, which is based on Colin Percival's scrypt. Recommended for new
hashes.

Prefix
"$y$"

Hashed passphrase format
\${$y\$[./A-Za-z0-9]+\$[./A-Za-z0-9]{,86}\$[./A-Za-z0-9]{43}\$}

Maximum passphrase length
unlimited

Hash size
256 bits
Manual page crypt(5) line 68 (press h for help or q to quit)
```

Y el del resto de usuarios sería el siguiente:

```
Terminal
File Edit View Search Terminal Help
historical reasons only. The alternative prefixes "$2a$" and "$2x$" provide bug-compatibility with crypt_blowfish 1.0.4 and earlier, which incorrectly processed characters with the 8th bit set.

sha512crypt
A hash based on SHA-2 with 512-bit output, originally developed by Ulrich Drepper for GNU libc. Supported on Linux but not common elsewhere. Acceptable for new hashes. The default CPU time cost parameter is 5000, which is too low for modern hardware.

Prefix
"$6$"

Hashed passphrase format
\$6\$(rounds=[1-9] [0-9]+\$)?[^$:\n]{1,16}\$[./0-9A-Za-z]{86}

Maximum passphrase length
unlimited

Hash size
512 bits
Manual page crypt(5) line 174 (press h for help or q to quit)
```

From:

<https://knoppia.net/> - Knoppia



Permanent link:

https://knoppia.net/doku.php?id=master_cs:fortificacion:p4&rev=1740494999

Last update: **2025/02/25 14:49**