

# SEGURIDAD EN SISTEMAS OPERATIVOS

4º Grado en Informática – Complementos de Ing. del Software  
Curso 2019-20

---

**Práctica [1].** Administración de la seguridad en Linux.

**Sesión [1].** Seguridad básica en Linux: privilegios de usuario y permisos.

**Autor<sup>1</sup>:** Nazaret Román Guerrero

---

## Ejercicio 1.

---

Indicar los formatos de los archivos `/etc/passwd`, `/etc/group`, `/etc/shadow` y `/etc/gshadow`.

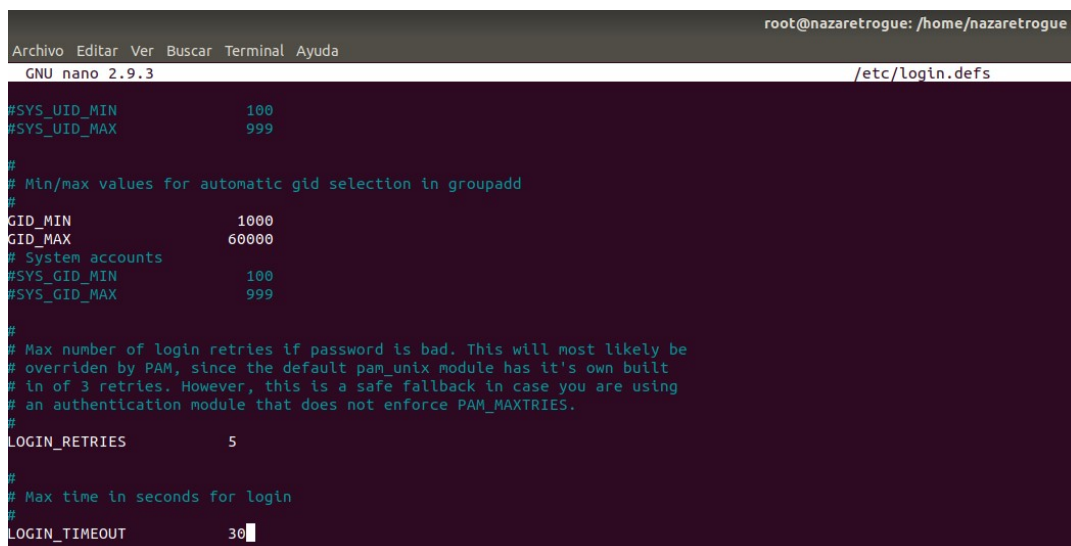
Todos los ficheros (`/etc/passwd`, `/etc/group`, `/etc/shadow` y `/etc/gshadow`) no tienen extensión. Cuando se utiliza el comando `file`, comprobamos que los ficheros de texto ascii, sin más.

## Ejercicio 2.

---

Modificar el archivo `/etc/login.defs` para que los usuarios creados a partir de ese momento tengan un valor asignado para la directiva `LOGIN_TIMEOUT`. Crear un usuario y comprobar que tiene efecto la citada directiva.

La directiva `LOGIN_TIMEOUT` viene activa por defecto con un tiempo de 60 segundos. Yo he cambiado su valor a 30 para comprobar que, en efecto, se estaba ejecutando (he puesto un tiempo menor para tardar menos tiempo).



```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/login.defs
#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     30
```

---

1 Como autor declaro que los contenidos del presente documento son originales y elaborados por mi. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la “Normativa de evaluación y de calificaciones de los estudiantes de la Universidad de Granada” esto “conllevará la calificación numérica de cero ... independientemente del resto de calificaciones que el estudiante hubiera obtenido ...”

Una vez cambiado el archivo, he creado un usuario, user\_prueba, y he intentado cambiar de cuenta de usuario mediante el comando login user\_prueba.

Tras esperar 30 segundos a que se introdujera la contraseña, el sistema ha dado por no válido el logueo y se ha salido del modo, tal y como se comprueba en las imágenes.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# useradd user_prueba
root@nazaretroque:/home/nazaretroque# passwd user_prueba
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
root@nazaretroque:/home/nazaretroque# login user_prueba
Contraseña:
El acceso caducó después de 30 segundos.
root@nazaretroque:/home/nazaretroque#
```

### Ejercicio 3.

Crear un ACL para un archivo de vuestro sistema de forma que el usuario creado en el ejercicio 2 tenga acceso de lectura y escritura.

Para hacer la prueba, he creado el archivo archivo\_ACL.txt, que pertenece a mi usuario, y tiene permisos de lectura y escritura para el propietario, y de lectura solamente para el grupo y otros, como se puede ver en la imagen.

```
nazaretroque@nazaretroque: ~
Archivo Editar Ver Buscar Terminal Ayuda
nazaretroque@nazaretroque:~$ touch archivo_ACL.txt
nazaretroque@nazaretroque:~$ ls -l
total 44
-rw-r--r-- 1 nazaretroque nazaretroque  0 sep 23 18:48 archivo_ACL.txt
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Descargas
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Documentos
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Escritorio
-rw-r--r-- 1 nazaretroque nazaretroque 8980 sep 20 13:02 examples.desktop
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Imágenes
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Música
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Plantillas
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Público
drwxr-xr-x 2 nazaretroque nazaretroque 4096 sep 20 13:18 Vídeos
nazaretroque@nazaretroque:~$ setfacl -m u:user_prueba:rw archivo_ACL.txt
nazaretroque@nazaretroque:~$
```

Para que el usuario creado en el ejercicio 2 tenga permisos de lectura y escritura, hay que añadirlo manualmente. Para ello, usando el comando setfacl, modificamos la lista de control de acceso para el usuario user\_prueba dándole permisos de lectura y escritura sobre el archivo archivo\_ACL.txt

Para comprobar si de verdad el usuario tiene los permisos con los que lo hemos dotado, mostramos la lista de control de acceso del archivo en cuestión, y, como podemos ver, el usuario user\_prueba

tiene los permisos que se piden.

```
nazaretroque@nazaretroque: ~
Archivo Editar Ver Buscar Terminal Ayuda
nazaretroque@nazaretroque:~$ getfacl archivo_ACL.txt
# file: archivo_ACL.txt
# owner: nazaretroque
# group: nazaretroque
user::rw-
user:user_prueba:rw-
group::r--
mask::rw-
other::r--
```

#### Ejercicio 4.

En el sistema que tenemos en uso, indicar los archivos de configuración existentes y comentar la misión de un par de ellos y cómo lo hacen.

El sistema que hay activo es Ubuntu 18.04 LTS, donde hay presentes los archivos de configuración que se observan en la captura.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# ls -la /etc/pam.d/
total 120
drwxr-xr-x  2 root root  4096 sep 24 19:31 .
drwxr-xr-x 123 root root 12288 sep 24 19:32 ..
-rw-r--r--  1 root root   384 ene 25 2018 chfn
-rw-r--r--  1 root root    92 ene 25 2018 chpasswd
-rw-r--r--  1 root root   581 ene 25 2018 chsh
-rw-r--r--  1 root root  1208 sep 24 19:30 common-account
-rw-r--r--  1 root root  1249 sep 24 19:30 common-auth
-rw-r--r--  1 root root  1480 sep 24 19:30 common-password
-rw-r--r--  1 root root  1470 sep 24 19:30 common-session
-rw-r--r--  1 root root  1435 sep 24 19:30 common-session-noninteractive
-rw-r--r--  1 root root   606 nov 16 2017 cron
-rw-r--r--  1 root root    69 mar 27 2018 cups
-rw-r--r--  1 root root  1192 oct  9 2018 gdm-autologin
-rw-r--r--  1 root root   1182 oct  9 2018 gdm-fingerprint
-rw-r--r--  1 root root   383 oct  9 2018 gdm-launch-environment
-rw-r--r--  1 root root   1160 oct  9 2018 gdm-password
-rw-r--r--  1 root root  4945 ene 25 2018 login
-rw-r--r--  1 root root    92 ene 25 2018 newusers
-rw-r--r--  1 root root   520 abr  4 2018 other
-rw-r--r--  1 root root    92 ene 25 2018 passwd
-rw-r--r--  1 root root   270 ene 15 2019 polkit-1
-rw-r--r--  1 root root   168 feb 26 2018 ppp
-rw-r--r--  1 root root   143 feb 14 2018 runuser
-rw-r--r--  1 root root   138 feb 14 2018 runuser-l
-rw-r--r--  1 root root  2257 ene 25 2018 su
-rw-r--r--  1 root root   239 ene 18 2018 sudo
-rw-r--r--  1 root root   317 abr 20 2018 systemd-user
root@nazaretroque:/home/nazaretroque#
```

Hablaré de dos de ellos: el archivo de cron y el archivo de sudo.

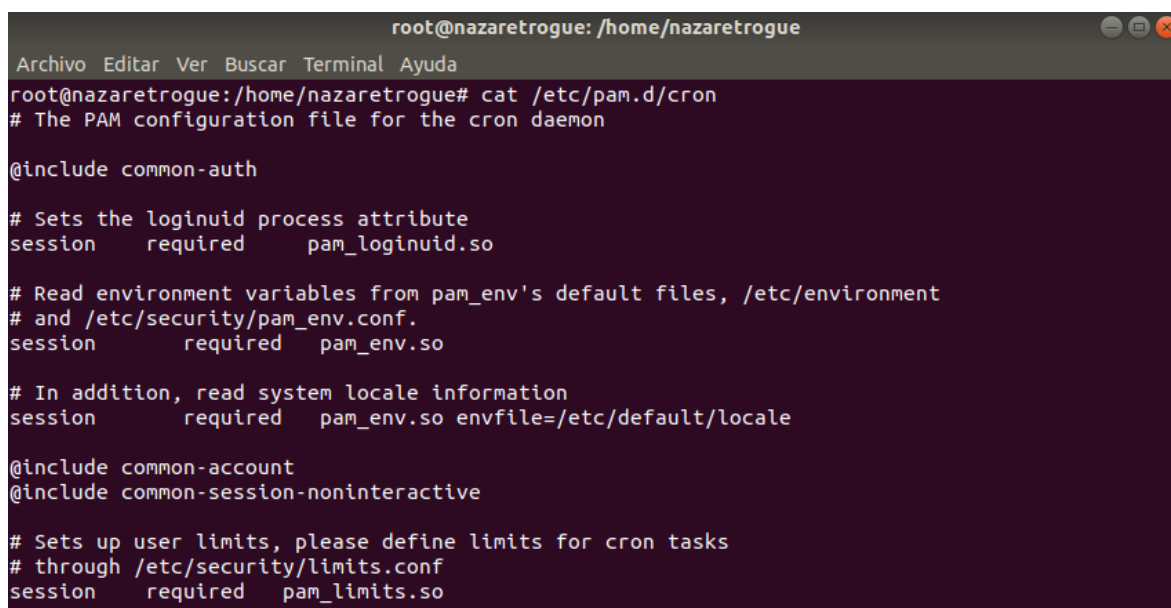
- Este archivo es utilizado por el demonio cron, utilizado para programar tareas en el sistema. Como se puede observar, la primera línea (ignorando el comentario) hace referencia a un archivo que debe consultarse, common-auth (que trata sobre la autenticación en el sistema); la siguiente línea llama al módulo loginuid que establece un identificador para el proceso y aunque falle la carga de dicho módulo, se llamará al resto de módulos.

La siguiente línea lee las variables de entorno, y, al igual que en caso anterior, aunque falle se continúa la carga de los demás módulos.

La siguiente línea establece el valor del fichero donde se almacenan las variables de entorno; en el caso de que falle la carga de este módulo, se siguen cargando los restantes.

Las dos siguientes líneas referencian a dos archivos que deben comprobarse, que son common-account y common-session-noninteractive.

Por último, la línea del final establece el número de tareas máximas que puede llevar a cabo el demonio. Si falla este módulo, se sigue adelante, aunque, en este caso, finaliza el archivo y por tanto finaliza la carga de módulos.

A terminal window titled 'root@nazaretroque: /home/nazaretroque' with a menu bar 'Archivo Editar Ver Buscar Terminal Ayuda'. The prompt is 'root@nazaretroque:/home/nazaretroque#'. The command 'cat /etc/pam.d/cron' has been executed, displaying the following PAM configuration for the cron daemon:

```
# The PAM configuration file for the cron daemon

@include common-auth

# Sets the loginuid process attribute
session    required    pam_loginuid.so

# Read environment variables from pam_env's default files, /etc/environment
# and /etc/security/pam_env.conf.
session    required    pam_env.so

# In addition, read system locale information
session    required    pam_env.so envfile=/etc/default/locale

@include common-account
@include common-session-noninteractive

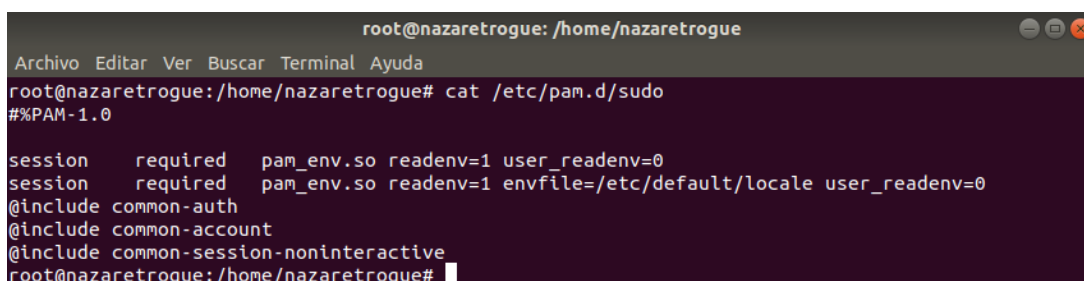
# Sets up user limits, please define limits for cron tasks
# through /etc/security/limits.conf
session    required    pam_limits.so
```

- El segundo archivo es el que usa la orden sudo, encargada de cambiar de usuario para ejecutar ciertas órdenes con permisos especiales. Este archivo es más corto que el anterior. Veamos lo que hace.

La primera línea carga el módulo de variables de entorno, activa la lectura de las variables de entorno pero desactiva la lectura de todas aquellas que sean específicas de un usuario. Si este módulo falla, se sigue adelante.

La siguiente línea establece el archivo donde se guardarán las variables de entorno, con la lectura de todas las variables activa excepto aquellas que pertenecen a un usuario concreto, igual que se hizo en la línea anterior. Si falla este módulo, se continua.

Las tres siguientes líneas se encargan de comprobar otros archivos que hacen falta, como son common-auth, common-account y common-session-noninteractive.

A terminal window titled 'root@nazaretroque: /home/nazaretroque' with a menu bar 'Archivo Editar Ver Buscar Terminal Ayuda'. The prompt is 'root@nazaretroque:/home/nazaretroque#'. The command 'cat /etc/pam.d/sudo' has been executed, displaying the following PAM configuration for the sudo command:

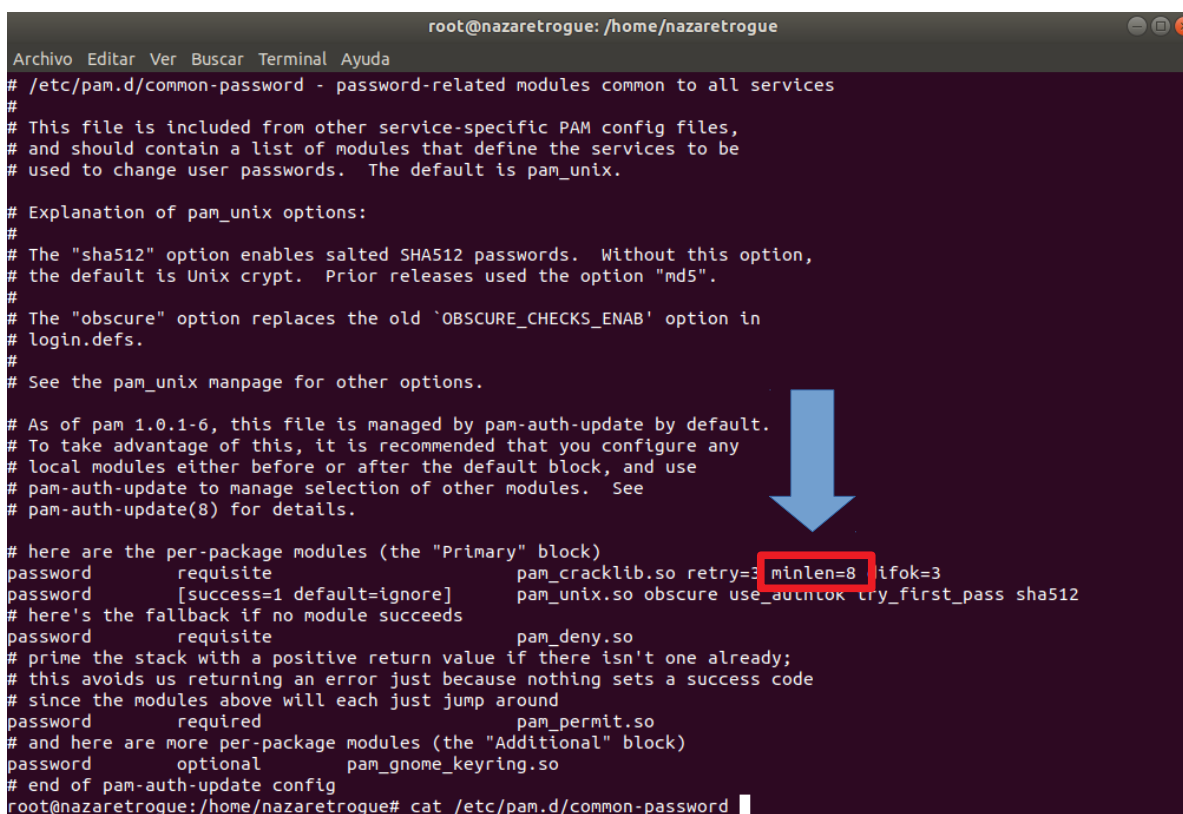
```
##PAM-1.0

session    required    pam_env.so readenv=1 user_readenv=0
session    required    pam_env.so readenv=1 envfile=/etc/default/locale user_readenv=0
@include common-auth
@include common-account
@include common-session-noninteractive
root@nazaretroque:/home/nazaretroque#
```

## Ejercicio 5.

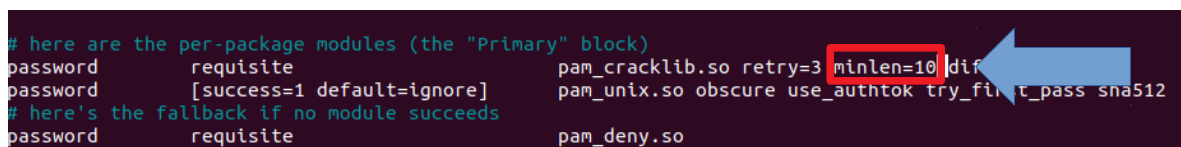
- Modificar la configuración para que la autenticación exija que la clave de un usuario tenga una longitud mínima. Debemos utilizar el módulo `pam_cracklib`. ¡Cuidado! Pues modificaciones inadecuadas pueden dejar sin acceso a usuarios que existen en el sistema.
  - Piensa otra modificación de tu preferencia e impleméntala. Por ejemplo, deshabilitar el acceso a root directo por consola, evitar que un usuario que no es root tire el sistema, etc.
- a) Para empezar, es necesario instalar el módulo de `pam_cracklib`, ya que no viene instalado por defecto. Para ello se usa la orden `apt-get install libpam-cracklib -y`.

Una vez instalado, accedemos al archivo `common-password`, situado en `/etc/pam.d/`, donde podemos observar que la longitud mínima establecida en la contraseña es de 8 caracteres (recuadro rojo):



```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.
#
# Explanation of pam_unix options:
#
# The "sha512" option enables salted SHA512 passwords. Without this option,
# the default is Unix crypt. Prior releases used the option "md5".
#
# The "obscure" option replaces the old 'OBSOLETE_CHECKS_ENAB' option in
# login.defs.
#
# See the pam_unix manpage for other options.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# 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_cracklib.so retry=3 minlen=8 ifok=3
password      [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass sha512
# here's the fallback if no module succeeds
password      requisite      pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password      required       pam_permit.so
# and here are more per-package modules (the "Additional" block)
password      optional       pam_gnome_keyring.so
# end of pam-auth-update config
root@nazaretroque:/home/nazaretroque# cat /etc/pam.d/common-password
```

Una vez localizada dicha longitud, la cambiamos y establecemos un mínimo de 10 caracteres para que la contraseña sea más segura.



```
# here are the per-package modules (the "Primary" block)
password      requisite      pam_cracklib.so retry=3 minlen=10 difok=3
password      [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass sha512
# here's the fallback if no module succeeds
password      requisite      pam_deny.so
```

Ahora, probamos a crear un usuario con una contraseña corta, por ejemplo, "1234". La salida es la siguiente:

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# useradd user_password
root@nazaretroque:/home/nazaretroque# passwd user_password
Nueva contraseña:
CONTRASEÑA INCORRECTA: Es demasiado corta.
CONTRASEÑA INCORRECTA: es demasiado sencilla
```

Como hemos comprobado, el sistema nos dice que la contraseña es demasiado corta.

- b) Vamos a implementar una medida que denegará el uso del comando `su` para usuarios que no sean parte de un grupo concreto con permiso para usarlo.

Para ello, creamos un grupo, `administrad`, donde meteré mi usuario personal. Una vez hecho esto, creamos un archivo en `/etc/security` para dar permiso de administración a los usuarios que hayamos metido en el grupo.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# groupadd administrad
root@nazaretroque:/home/nazaretroque# usermod -G administrad nazaretroque
root@nazaretroque:/home/nazaretroque# touch /etc/security/su-administrad-access
root@nazaretroque:/home/nazaretroque# nano /etc/security/su-administrad-access
```

Para ello escribimos `admin` en el archivo.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# cat /etc/security/su-administrad-access
admin
```

Una vez hecho esto, modificamos el archivo de configuración del comando `su`, añadiendo 2 líneas nuevas. La primera línea invoca al módulo `pam_wheel`, que permite la escalada a privilegios de `root` durante la ejecución, pero si falla, llamará a los demás módulos igualmente. Establece un `uid` de usuario a todos aquellos que formen parte del grupo `administrad`, y se establece la depuración para que pudiese comprobar qué pasaba si fallaba.

La segunda línea llama al módulo `pam_listfile`, un archivo que contiene una lista de *items*, en este caso, usuarios a los que se les permite (sense=allow) hacer algo, en este caso, usar `su`. El fichero que debe buscar está situado en el *path* dado por `file`, y, en el caso de que haya algún error, automáticamente el programa (comando en este caso) dará error y denegará el uso de éste.

```
# Solo los usuarios en el grupo administrad pueden cambiar con su
auth    required      pam_wheel.so        use_uid group=administrad    debug
auth    required      pam_listfile.so     item=user             sense=allow             onerr=fail file=/etc/security/su-administrad-access
```



## Ejercicio 6.

---

Crear en el sistema un usuario con las características que deseéis, entrando como ese usuario cambiar la contraseña y analizar los archivos de log para ver el mensaje correspondiente.

Para este ejercicio, creamos un usuario `user_log` y establecemos una contraseña.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
$ whoami
user_log
$ passwd
Cambiando la contraseña de user_log.
(actual) contraseña de UNIX:
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
$
```

Tras crearlo, entramos en la sesión, comprobamos que, en efecto, somos `user_log`, y cambiamos la contraseña desde nuestra cuenta.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# useradd user_log
root@nazaretroque:/home/nazaretroque# passwd user_log
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
root@nazaretroque:/home/nazaretroque#
```

Ahora, comprobamos el fichero de logs, `/var/log/auth.log`, donde aparece la información sobre el login en una cuenta `user_log` y el cambio de contraseña que se ha hecho.

```
Sep 28 18:57:43 nazaretroque login[2780]: pam_unix(login:session): session opened for user user_log by (uid=0)
Sep 28 18:57:43 nazaretroque login[2780]: pam_systemd(login:session): Cannot create session: Already running in a session
Sep 28 18:58:19 nazaretroque passwd[2988]: pam_unix(passwd:chauthtok): password changed for user_log
```

## Ejercicio 7.

---

Modificar el archivo `sudoers` para que un usuario determinado tenga acceso a todas las órdenes del `root`.

Para este ejercicio, utilizaremos el mismo usuario que se creó en el segundo ejercicio, el usuario `user_prueba`. Para probar que desde el inicio no tiene permisos para ejecutar comandos `sudo`, hacemos la prueba de mostrar el archivo que tenemos que modificar. Sin el comando en modo `sudo`, nos deniega el acceso por no tener permiso, con el comando `sudo` nos deniega el acceso y además

dice que se informará por haber intentado acceder con un comando que no tenemos permitido, tal y como se ve en la imagen.

```
nazaretroque@nazaretroque: ~
Archivo Editar Ver Buscar Terminal Ayuda
$ whoami
user_prueba
$ cat /etc/sudoers
cat: /etc/sudoers: Permiso denegado
$ sudo cat /etc/sudoers
[sudo] contraseña para user_prueba:
user_prueba no está en el archivo sudoers. Se informará de este incidente.
$
```

Tras comprobar que en efecto no tenemos permisos, añadimos el usuario al archivo (desde un usuario que sí pueda utilizar comandos sudo, como es el mío personal que tiene permisos).

```
nazaretroque@nazaretroque: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/sudoers.tmp Modificado

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include /etc/sudoers.d
user_prueba ALL=(ALL:ALL) ALL
```

Tras añadirlo, volvemos a comprobar si es posible ahora abrir el archivo el comando sudo:

```
nazaretroque@nazaretroque: ~
Archivo Editar Ver Buscar Terminal Ayuda
$ sudo cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

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

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include /etc/sudoers.d
user_prueba ALL=(ALL:ALL) ALL
$
```



Tal y como hemos comprobado, ahora no dice que se vaya a dar un aviso por utilizar un comando que no tenía permitido utilizar. Esto demuestra que el usuario `user_prueba` ahora tiene acceso al uso de los comandos que requieren privilegios de `root`.

## Ejercicio 8.

Analiza el contenido de estos archivos del registro del sistema de prácticas y comprueba que efectivamente se registran los eventos indicados.

- `/var/log/lastlog`: tal y como se puede observar, se muestra la última conexión que alguien ha hecho a través del comando `login`.

```
hplip                **Nunca ha accedido**
geoclue              **Nunca ha accedido**
gnome-initial-setup  **Nunca ha accedido**
gdm                  **Nunca ha accedido**
nazaretroque         **Nunca ha accedido**
user_prueba pts/0    sáb sep 28 20:48:22 +0200 2019
root@nazaretroque:/home/nazaretroque#
```

- `/var/log/wtmp`: se puede ver que muestra todos los usuarios que han hecho `login` y `logout`. La última línea del archivo indica el momento en que el archivo fue creado.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque:/home/nazaretroque# last
user_pru pts/0          Sat Sep 28 20:48 - 20:48 (00:00)
nazaretr :0              Sat Sep 28 20:45 - still logged in
reboot    system boot    5.0.0-29-generic Sat Sep 28 20:44 - still running
reboot    system boot    5.0.0-29-generic Sat Sep 28 20:31 - still running
nazaretr  :0              Tue Sep 24 21:45 - 21:45 (00:00)
reboot    system boot    5.0.0-29-generic Tue Sep 24 21:44 - 21:45 (00:00)
nazaretr  :0              Tue Sep 24 20:16 - crash (01:28)
reboot    system boot    5.0.0-29-generic Tue Sep 24 18:55 - 21:45 (02:49)
nazaretr  :0              Mon Sep 23 18:25 - 19:11 (00:45)
reboot    system boot    5.0.0-29-generic Mon Sep 23 18:01 - 19:11 (01:09)
nazaretr  :0              Fri Sep 20 14:01 - 14:23 (00:21)
reboot    system boot    5.0.0-29-generic Fri Sep 20 14:00 - 14:23 (00:22)
nazaretr  :0              Fri Sep 20 13:18 - 13:22 (00:04)
reboot    system boot    5.0.0-29-generic Fri Sep 20 13:17 - 13:22 (00:05)
reboot    system boot    5.0.0-29-generic Fri Sep 20 13:15 - 13:22 (00:07)

wtmp empieza Fri Sep 20 13:15:30 2019
root@nazaretroque:/home/nazaretroque#
```

- `/var/run/utmp`: muestra los usuarios conectados al sistema. Entre los distintos caracteres no reconocidos, se puede ver que los usuarios conectados son `nazaretroque` (mi usuario) y `user_prueba`.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque: /home/nazaretroque# cat /var/run/utmp
[00]~~reboot5.0.0-29-generic*00]00[00]~~runlevel5.0.0-29-genericH00]b0
w[00]0nazaretroque:0e00]P|pts/0/0user_pruebav[00]00[00]root@nazaretroque: /home/nazaretroque#
```

- /var/log/btmp: muestra las conexiones fallidas de usuarios. En este caso, está vacío.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque: /home/nazaretroque# lastb
btmp empieza Fri Sep 20 12:59:19 2019
root@nazaretroque: /home/nazaretroque#
```

- /var/log/sudo y /var/log/messages: estos ficheros como tal no existen. En la imagen se han listado todos los archivos del directorio /var/log, y como se puede apreciar, dichos ficheros no están, por lo que, o esos logs pertenecen a versiones más antiguas del sistema operativo (estoy trabajando con Ubuntu 18.04 LTS), o bien han cambiado de nombres.

```
root@nazaretroque: /home/nazaretroque
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaretroque: /home/nazaretroque# ls -l /var/log
total 3072
-rw-r--r-- 1 root root 34933 sep 24 19:31 alternatives.log
drwxr-xr-x 2 root root 4096 sep 28 20:32 apt
-rw-r----- 1 syslog adm 10618 sep 28 21:13 auth.log
-rw-r----- 1 syslog adm 17133 sep 23 18:50 auth.log.1
-rw----- 1 root root 8862 sep 28 20:32 boot.log
-rw-r--r-- 1 root root 56751 feb 10 2019 bootstrap.log
-rw-rw---- 1 root utmp 0 feb 10 2019 btmp
drwxr-xr-x 2 root root 4096 sep 28 20:37 cups
drwxr-xr-x 2 root root 4096 ene 17 2019 dist-upgrade
-rw-r--r-- 1 root root 1532629 sep 28 20:33 dpkg.log
-rw-r--r-- 1 root root 32064 sep 20 14:19 faillog
-rw-r--r-- 1 root root 5784 sep 24 19:20 fontconfig.log
drwx--x--x 2 root gdm 4096 oct 9 2018 gdm3
-rw-r--r-- 1 root root 1175 sep 28 20:45 gpu-manager.log
drwxr-xr-x 3 root root 4096 feb 10 2019 hp
drwxrwxr-x 2 root root 4096 sep 20 13:14 installer
drwxr-sr-x+ 3 root systemd-journal 4096 sep 20 13:15 journal
-rw-r----- 1 syslog adm 242797 sep 28 20:46 kern.log
-rw-r----- 1 syslog adm 259876 sep 23 18:25 kern.log.1
-rw-rw-r-- 1 root utmp 292584 sep 28 20:48 lastlog
drwx----- 2 speech-dispatcher root 4096 abr 23 2018 speech-dispatcher
-rw-r----- 1 syslog adm 224268 sep 28 21:13 syslog
-rw-r----- 1 syslog adm 444777 sep 28 20:37 syslog.1
-rw-r----- 1 syslog adm 22297 sep 24 19:00 syslog.2.gz
-rw-r----- 1 syslog adm 130327 sep 23 19:07 syslog.3.gz
-rw----- 1 root root 64128 sep 20 14:19 tallylog
drwxr-x-- 2 root adm 4096 sep 24 19:16 unattended-upgrades
-rw-rw-r-- 1 root utmp 12288 sep 28 20:48 wtmp
root@nazaretroque: /home/nazaretroque#
```

## Ejercicio 9.

---

Analizar las conexiones al sistema de prácticas y al de casa. ¿Hay o ha habido alguna conexión ajena al equipo?

Utilizando las órdenes que se han usado en el ejercicio anterior (`lastlog`, `last` y `lastb`), podemos comprobar quienes son los que han accedido al sistema (las imágenes de la ejecución de dichos comandos están en el ejercicio anterior). La mayoría de los usuarios que han accedido son demonios del propio sistema que llevan a cabo funciones como el correo, hacer *backups* o llevar a cabo ciertas funciones de forma periódica (como el demonio `cron`, del que se ha hablado en el ejercicio 4).

Por tanto, que se pueda apreciar no ha habido ningún acceso externo al sistema, ya que, de otro modo, se habría visto reflejado en el contenido de dichos archivos de `log`.