

**Índex**

[1. Introducció 3](#_gjdgxs)

[1.1. Objectius 3](#_30j0zll)

[2. Profile i entorn d'usuari 3](#_1fob9te)

[La variable d'entorn que defineix el prompt en la majoria dels shells de Unix com Bash és PS1. Aquesta és la variable primària que s'utilitza per definir l'aparença del prompt de la línia d'ordres quan s'utilitza un shell interactiu. 4](#_sk7sbr9jb9vf)

[3. Creació manual d'usuaris 4](#_3znysh7)

[4. Creació automàtica d'usuaris 10](#_2et92p0)

[Trieu i justifiqueu el lloc més adequat per als home de tots els usuaris. 11](#_tyjcwt)

[5. Connexió remota d'usuaris 14](#_3dy6vkm)

[6. Eliminació i des-activació d'usuaris 17](#_1t3h5sf)

[7. Usuari especial asosh 25](#_4d34og8)

[8. Sudo i control d'execució d'aplicacions 27](#_2s8eyo1)

# Introducció

Al sistema cada usuari té un compte associat. Un compte són tots els fitxers, recursos i informació que pertanyen a cada usuari. Els comptes d'usuari permeten al sistema diferenciar les dades i processos de cada usuari i permeten als usuaris protegir la seva informació.

Per al kernel els usuaris s'identifiquen amb un nombre enter conegut com l'identificador d'usuari (*user identifier o UID*). A més hi ha una base de dades que associa el UID amb un nom textual: el *username.* Aquest *username é*s l'utilitzat per l'usuari per fer *login*. La base de dades d'usuaris inclou altra informació relativa a l 'usuari com la ruta del directori *home,* el nom complet de l'usuari i l'interpret de comandes (shell).

La creació de un nou usuari inclou l'assignació d'un UID i la modificació de la base de dades d'usuaris per assignar els paràmetres propis de l'usuari. A més és necessari associar almenys un grup a l'usuari i finalment copiar els fitxers de configuració i personalització al directori *home* de cada usuari.

Opcionalment es pot assignar l'usuari a més d'un grup, la qual cosa permet a l'administrador del sistema dividir els usuaris en grups amb diferents permisos i privilegis. D'aquesta manera podem mantenir un millor control sobre què poden fer el usuaris.

## Objectius

Gestionar els usuaris del sistema: realitzar l'alta i baixa d'usuaris i modificar les propietats dels comptes d'usuari.

# Profile i entorn d'usuari

Quant s'inicia un *login* interactiu, el *shell* automàticament executa un o més fitxers predefinits. Cada *shell* executa fitxers diferents. El shell **bash** executa el fitxer */etc/profile* i a més a més executa el fitxer *.profile, .bash\_profile* o .*bashrc* del *home* de cada usuari. El fitxer */etc/profile* permet a l'administrador del sistema definir un entorn comú per a tots els usuaris, especialment definint la variable **PATH**. Per altra banda *.bash\_profile* o *.bashrc* permet a cada usuari definir el seu propi entorn adequant el *PATH*, el *prompt*, etc.

Quan es crea el directori home d'un usuari s'han de copiar els fitxers del directori */etc/skel*. L'administrador del sistema pot posar fitxers a */etc/skel* que donin un entorn inicial pels usuaris. Per exemple, com administradors creeu un fitxer */etc/skel/.bashrc* (si no està ja creat) amb unes definicions bàsiques que després l'usuari podria canviar.

Comproveu que al PATH de tots els usuaris hi sigui el directori */usr/local/bin* i, si cal, feu que el .*bashrc* modifiqui el PATH per incloure un directori bin situat en el directori *home* de cada usuari (**$HOME/bin**).

Executem la comanda:

echo PATH

Si /usr/local/bin no apareix editem /etc/profile i afegim:

export PATH="/usr/local/bin:$PATH"

**Volem que el *prompt* sigui el *username* seguit de la data actual i finalment “> “ (per exemple, el de l'usuari xavim seria "xavim (Tue April 10) >” )**



**Quina variable d'entorn té la definició del prompt?**

# La variable d'entorn que defineix el prompt en la majoria dels shells de Unix com Bash és PS1. Aquesta és la variable primària que s'utilitza per definir l'aparença del prompt de la línia d'ordres quan s'utilitza un shell interactiu.

Per verificar la configuració actual del prompt, s’utilitza el comandament echo $PS1 en el terminal.

# Creació manual d'usuaris

**Ara volem donar d'alta un compte d'usuari per a dos usuaris. Abans de començar trieu els paràmetres de cada usuari. Els usuaris han de formar part del grup *admin*.**

**Omple la següent taula:**

**Editeu la base de dades d'usuaris per afegir els nous usuaris. Utilitzeu la comanda vipw per editar aquest fitxer.**

Assignació UID:

* Ha de ser major o igual a 1000, ja que els que són menors s’utilitzen per als usuaris del sistema.
* No podem assignar un UID si ja està assignat a un altre usuari prèviament:
  + Ho comprovem amb la comanda grep “UID” /etc/passwd

A Usuari 1 li donarem el UID 1001, comprovem que no hi ha un usuari amb aquest UID assignat:

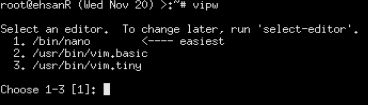


A Usuari 2 li assignem el UID 1002, fem la mateixa comprovació:



| paràmetres /Usuari | Usuari 1 | Usuari 2 |
| --- | --- | --- |
| UID | 1001 | 1002 |
| *Username* | user1 | user2 |
| Directori home | /home/user1 | /home/user2 |
| *Shell* | /bin/bash | /bin/bash |
| Grups | admin | admin |

Entrem a usuari root i escrivim la comanda vipw i escollim la primera opció:



Afegim les següents línies al final per als nous usuaris:

user1:x:1001:1001::/home/user1:/bin/bash

user2:x:1002:1002::/home/user2:/bin/bash



Desem amb Ctrl + o i sortim amb Ctrl + x.

**Quina és la diferencia en usar vipw o editar directament el fitxer de passwd amb vi?(pista: obriu dos vipw en sessions diferents)**

Si fem servir vipw, ens assegurem que es bloquegi l’accés al fitxer /etc/passwd mentre s’està editant. El seu ús ens permet evitar que altres processos o usuaris facin canvis al fitxer de forma simultània, el que podria causar inconsistències.

D’altra banda, si ho editem directament amb vi, això no passaria, de manera que un altre usuari o procés podria modificar el fitxer al mateix temps.

**De la mateixa manera, utilitzeu la comanda vigr per crear un grup per a cada usuari i definir els altres grups que siguin necessaris.**

Per afegir un grup per a cada usuari amb un GID únic:

* Comprovem que en el fitxer /etc/group no hi hagi un grup amb el mateix GID

Fem servir la comanda grep “UID” /etc/group. Farem servir el GID 1003 per admin



* Escrivim la comanda vigr i afegim la següent línia:

admin:x:1003:



* Per últim afegim els usuaris al grup admin escrivint les comandes:

usermod -aG admin user1

usermod -aG admin user2



Això és el que hauria de sortir si fem la comanda id nomUsuari:



**Com es pot desactivar un compte de forma que l'usuari no pugui fer *login?***

Es pot de dues maneres:

* Bloquejant el compte amb usermod -L nomUsuari
  + Això afegeix el símbol ! davant de la contrasenya xifrada al fitxer /etc/shadow, per tant, l’usuari no podrà iniciar sessió perquè la contrasenya queda inhabilitada, però el compte segueix existent.
* Canviant el shell de l’usuari a /usr/sbin/nologin: *sudo usermod -s /usr/sbin/nologin nomUsuari*
  + Canvia la shell de l'usuari a una shell especial que mostra un missatge indicant que el compte està desactivat.

**Desactiveu els comptes nous fins que no hagi finalitzat de donar d'alta els usuaris.**

Escrivim la comanda usermod -L nomUsuari:



**Creeu el directori *home* de cada usuari, copieu els fitxers que estiguin a /etc/skel i assigneu el propietari i permisos adequats per al directori *home* i per a tots els fitxers que estiguin dintre del directori.**

Creem els directoris amb les comandes:

mkdir /home/user1

mkdir /home/user2



Seguidament escrivim les comandes per copiar els fitxers del directori /etc/skel a /home/nomUsuari:

cp -r /etc/skel/. /home/user1/

cp -r /etc/skel/. /home/user2/



I assignem els propietaris escrivint:

chown -R user1:admin /home/user1

chown -R user2:admin /home/user2



Assignem permisos de lectura, escriptura i d’execució al propietari, als altres usuaris i a la resta només de lectura i execució.

chmod 755 /home/user1

chmod 755 /home/user2



**Ara assigneu una clau (password) per a cada usuari nou.**

**Per raons de seguretat la clau no es posa directament al fitxer /etc/passwd. Per això hi ha un altre fitxer anomenat /etc/shadow que només té permisos de lectura per al superusuari. En aquest fitxer es posa la clau xifrada i altres paràmetres associats a la vigència de la clau.**

**Amb quina comanda es pot editar de manera segura el fitxer de *shadow?***

Utilitzem la comanda passwd per establir una contrasenya per a cada usuari:

passwd user1



El sistema ens demanarà que introduïm la nova contrasenya, ho fem i confirmem-la. La contrasenya es xifrarà automàticament i es desarà al fitxer /etc/shadow. Repetim el procés per user2.

**Quin es el significat dels altres paràmetres que es poden definir al fitxer de shadow?**





Aquestes imatges corresponen a l’estructura següent:

1:2:3:4:5:6:7:8

1. Nom de l'usuari al qual es refereix aquesta entrada.
2. Conté la contrasenya xifrada de l'usuari.
3. Data del darrer canvi de contrasenya, expressada en dies des del 1 de gener de 1970 (epoch time).
4. Nombre mínim de dies que han de passar abans que l'usuari pugui canviar la contrasenya.
5. Nombre màxim de dies durant els quals una contrasenya pot ser utilitzada abans que caduqui i es requereixi un canvi.
6. Nombre de dies abans de la caducitat en què l'usuari serà avisat que ha de canviar la contrasenya.
7. Nombre de dies després de la caducitat de la contrasenya durant els quals l'usuari pot seguir iniciant sessió.
8. Data exacta en què el compte serà desactivat, expressada en dies des del 1 de gener de 1970.

**Amb quina comanda es poden modificar aquests paràmetres?**

Per canviar aquests paràmetres, es fa servir la comanda chage → Uns quants exemples:

* chage -M X nomUsuari (establir nombre màxim de dies abans de forçar un canvi de contrasenya)
* chage -m X nomUsuari (establir el nombre mínim de dies entre canvis de contrasenya)
* chage -W X nomUsuari (definir dies d'inactivitat abans de bloquejar el compte)

Per consultar la configuració actual, escrivim chage -l nomUsuari:

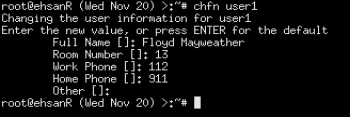


**Per editar altres paràmetres del compte d'usuari es poden utilitzar les comandes: chfn i chsh. Utilitzeu aquestes comandes per assignar valors adequats als comptes creats.**

Si executem chfn nomUsuari (que modifica camps com el nom complet, número de telèfon, etc., que es troben al fitxer /etc/passwd), ens apareix això:



Afegim la nostra informació:



Si fem cat /etc/passwd podrem observar la nostra informació:



D’altra banda, chsh permet modificar el shell predeterminat d’un usuari. Aquesta configuració es troba al camp final de cada entrada al fitxer /etc/passwd.

Si escrivim la comanda chsh -s /bin/sh user1, canvia el shell predeterminat:



Podem veure-ho si fem cat /etc/passwd:



# Creació automàtica d'usuaris

**La majoria de les distribucions de Linux inclouen programes per automatitzar les tasques de creació i modificació de dades d'usuaris. Unes d'aquestes aplicacions son useradd i adduser, que permeten crear usuaris i assignar els diferents paràmetres necessaris per donar d'alta cada compte.**

**Utilitzeu aquestes comandes per donar d'alta els usuaris següents:**

**Product Owners: PO1, PO2, PO3**

**Scrum Master: SM1, SM2**

**Equip de Desenvolupament (ED): El nom d'usuari del compte serà: nomX, on nom és el vostre nom i X la primera lletra del vostre cognom en minúscules.**

## Trieu i justifiqueu el lloc més adequat per als home de tots els usuaris.

El lloc més adequat es /home/ ja que és la ubicació estàndard per norma. Una de les raons es que els permisos del directori /home garanteixen la privadesa de cada usuari i permeten un control segur de l'accés als seus fitxers. L’estructura també separa clarament els fitxers del sistema dels fitxers dels usuaris, evitant problemes quan s'actualitza o modifica el sistema.

**Els permisos de cadascun d'aquests grups d'usuaris (POs, SMs i ED) venen definits de la següent forma:**



**Mostra tot el procés de creació indicant pas a pas que s'ha fet**

Creem directoris a home per tenir ordenats els usuaris

# 

Ara creem els grups on després crearem els usuaris, ho fem amb groupadd nom



En cas que ens hàgim equivocat o vulguem esborrar-ho, hem de fer: groupdel nomgrup

comprovem que tot hagi sortit correctament fent nano /etc/group

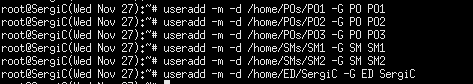




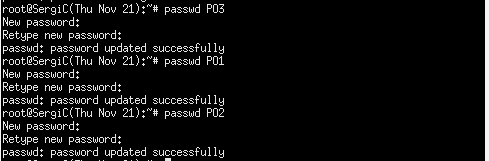
Creem els usuaris dins del grup pertinent, ho podem fer amb

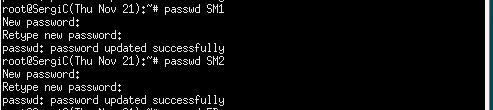
useradd -m -d /home/directori/nomusuari -s /bin/bash -G nomgrup nomusuari

on directori és el que hem creat abans



Creem les contrasenyes per a cada usuari amb passwd nomusuari

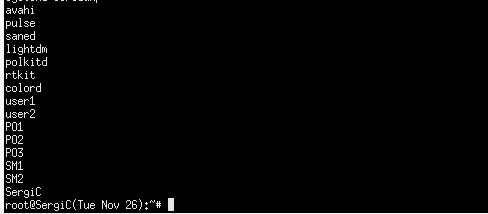


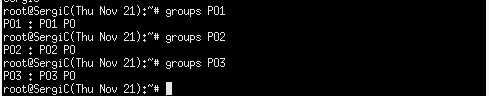




Si volem que el usuari canvi la contrasenya al entrar fem: chage -d 0 nomusuari llistem els usuaris del sistema per comprovar que tot ha sortit correctament



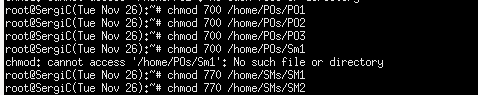




I donem els permisos necessaris:

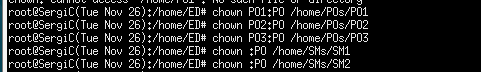
700 restringim l’accés només al propietari del directori

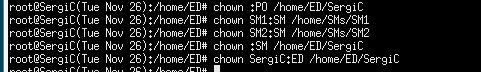
770 el propietari i el grup tenen accés complet al directori



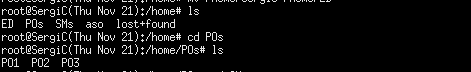


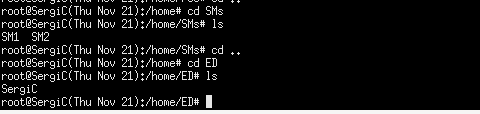
I posem els propietaris pertinents amb la comanda chown





I comprovem que tot hagi sortit correctament



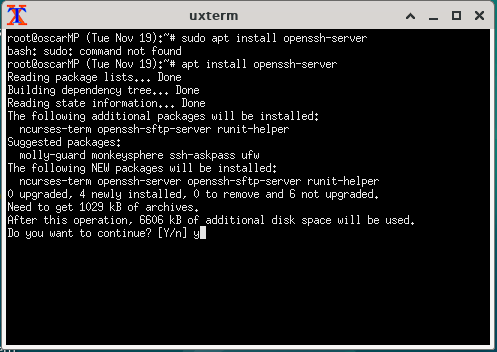


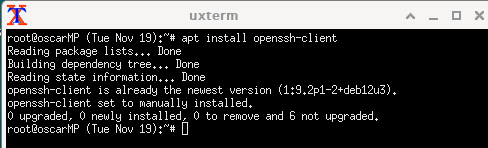
# Connexió remota d'usuaris

**Els usuaris de la nostra màquina han de tenir l'opció de poder connectar-se en remot de una forma segura.**

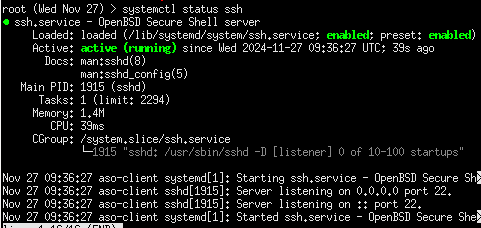
**instal·leu el paquet *openssh-server i openssh-client (si cal)***

**Amb la comanda *sudo apt install openssh-server* instal·lem el paquet per poder exercir de servidor, i amb *sudo apt install openssh-client*, el paquet per exercir de client**

**

**

Ara fem us de la comanda **sudo systemctl enable ssh,** per activar la opció d’ssh sempre que la maquina estigui activa. Després **sudo systemctl start ssh**, i per acabar, **sudo systemctl status ssh**, per comprovar que funciona correctament.



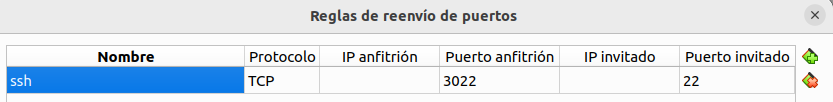
Comproveu que us podeu connectar remotament a un altra màquina.

Per comprovar que funciona fem la següent comanda: ***ssh usuari\_local@ip\_local***.

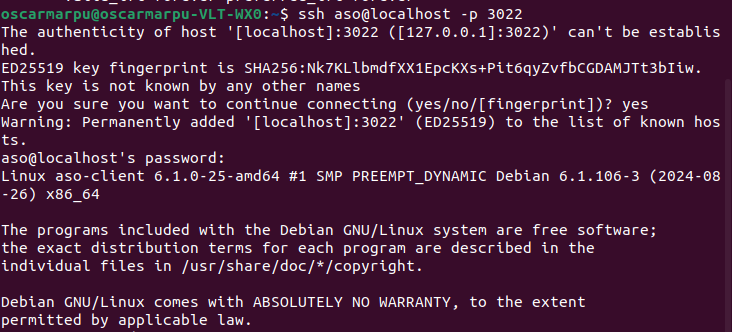
# 

En cas que ens deixi accedir remotament, significarà que funciona correctament com a client.

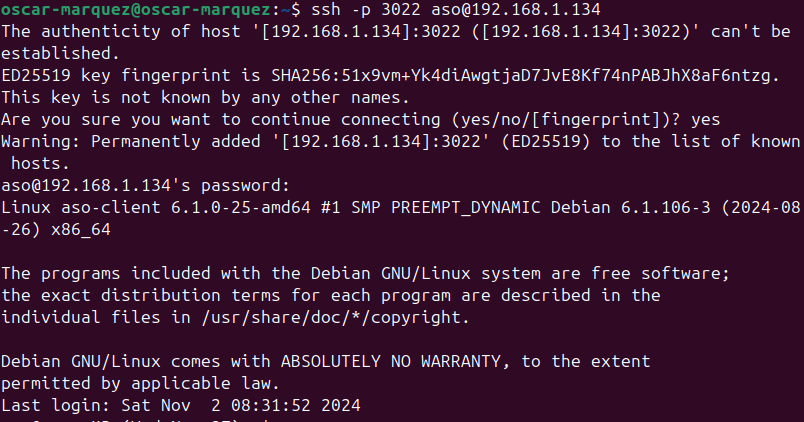
Ara per poder connectar amb ssh a la maquina virtual, hem d’apagar la maquina, i anar a la configuració de la MV. Un cop allà, anem a l’apartat Red, i cliquem a la opció “reenvío de puertos”. Afegim una nova regla, i la posem de la següent manera:



Comprovem el correcte funcionament desde el PC local: (Abans de fer la comanda ens hem d’assegurar que ssh es trobi instal·lat en el nostre ordinador local)



Prova des d’un altre ordinador:



# Eliminació i des-activació d'usuaris

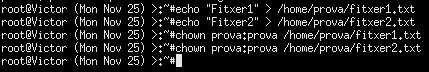
**Per donar de baixa un usuari és necessari eliminar tots els seus fitxers, les bústies de correu, treballs d'impressió, treballs cron i at i totes les referències a l'usuari. Després d'això es poden esborrar les línies associades a l'usuari al fitxer de passwd i de grups. Com un usuari pot tenir fitxers fora del seu directori home es necessari buscar per tot l'arbre de directoris el fitxers que pertanyen l'usuari i esborrar-los.**

**Crea un usuari de prova (o escolleix un existent) i afegeix fitxers al seu home.**

Creem l’usuari de prova:



I afegim fitxers qualsevols al seu home



**És una bona practica de seguretat primer desactivar el compte del usuari abans de començar el procés de donar-lo de baixa.**

**Una manera de desactivar un compte, a banda d'invalidar el password, consisteix en canviar el *shell* de l'usuari per un un programa senzill que només escriu a la pantalla un missatge i dóna informació a l'usuari de les raons per les quals el seu compte d'usuari ha estat desactivat. Per això es pot crear un 'tail script'. Per exemple:**

**Aquest script es pot posar com shell de l'usuari usant la comanda chsh i es pot guardar en un directori separat, per exemple /usr/local/lib/no-login.**

**Utilitzeu la comanda chsh per posar un *tail script* per desactivar el compte de l'usuari creat .**

Creem el script al directori /usr/local/lib:



Y el modifiquem. Hem de posar un sleep per a que la terminal no es tanqui i l’usuari pugui llegir el missatge que hem configurat. El sleep 10 fará que el terminal es mostri 10 segons i després es tanqui.



I utilitzem la comanda chsh per desactivar el usuari prova



**Com es pot comprovar que el compte ha quedat desactivat?**

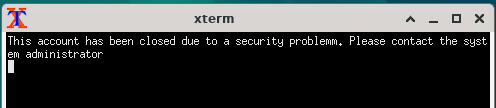
Es pot comprovar de diferents formes, primer podem comprobar al fitxer passswd com s’ha posat al directori no-login.



I a més, podem intentar iniciar sessió amb l’usuari prova:

  
I podem veure com ens surt el missatge que hem configurat al script

També podem entrar directament amb l’usuari nou i veure la terminal:



**Fes un backup amb tots els fitxers de l'usuari (tingueu en compte que potser una llista molt llarga de fitxers. Pista: feu servir xargs)**

Utilitzem la comanda per trobar tots els fitxer de l’usuari prova, i la comanda per comprimir tots aquests fitxers en un backup.



Els errors que ens surten del directori proc són perquè aquest directori té archius dinamics que poden canviar amb el temps, encara això aquest error no afecta a la comanda i es poden ometre.

Comprovem que al backup estiguin tots els fitxers



**Quin problema hi ha amb els fitxers que tinguin espais al seu nom? Com es pot resoldre això? (veure les opcions de la comanda xargs o la opció -exec de find)**

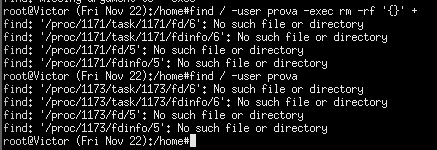
Si hi han fitxers que tinguin espais al nom poden haver problemes al ser processats per la comanda xargs, ja que l’espai es interpretat com un separador d’arguments i pot partir el fixter en dos, per exemple el el fixter “Fitxer 1.txt” la comanda xargs només llegirá fins Fitxer, i la resta ho llegirá com un altre document.

Per solucionar-ho hem d’executar la comanda find amb el parametre “-print0”, i la comanda xargs amb el parametre “-0”.

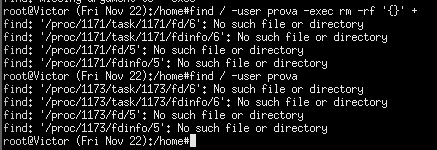


**Busca tots els fitxers de l'usuari i esborrar-los.**

Utilitzem la comanda per trobar tots els fitxer de l’usuari prova i per a cada fixter fem per eliminar els arxius



I comprovem que no estiguin els fixters.



**Ara crea un script que donat el nom d'usuari, faci un backup del seu directori home, esborri tots el fitxers que l'usuari tingui al sistema i canviï el shell per un *tail script* que avisi a l'usuari que el seu compte ha estat esborrat.**

#!/bin/bash

# Comprovar que s'ha proporcionat un usuari

if [ -z "$1" ]; then

echo "Ús: $0 nom\_usuari"

exit 1

fi

# Variables

USUARI=$1

BACKUP\_DIR="/root/backups"

BACKUP\_FILE="${BACKUP\_DIR}/${USUARI}\_home\_backup.tar.gz"

TAIL\_SCRIPT="/usr/local/lib/no-login.sh"

# 1. Comprovar si l'usuari existeix

if ! id "$USUARI" &>/dev/null; then

echo "Error: L'usuari '$USUARI' no existeix."

exit 2

fi

# 2. Crear el directori de backups si no existeix

if [ ! -d "$BACKUP\_DIR" ]; then

mkdir -p "$BACKUP\_DIR"

echo "Directori de backups creat a $BACKUP\_DIR."

fi

# 3. Fer un backup del directori home de l'usuari

echo "Fent backup del directori home de l'usuari..."

tar -czvf "$BACKUP\_FILE" "/home/$USUARI" 2>/dev/null

if [ $? -eq 0 ]; then

echo "Backup completat: $BACKUP\_FILE"

else

echo "Error en fer el backup del directori home."

exit 3

fi

# 4. Eliminar tots els fitxers de l'usuari al sistema

echo "Eliminant tots els fitxers de l'usuari..."

find / -user "$USUARI" -exec rm -rf '{}' + 2>/dev/null

if [ $? -eq 0 ]; then

echo "Tots els fitxers de l'usuari han estat eliminats."

fi

# 5. Crear un script de no-login

echo "Creant el tail script per desactivar l'usuari..."

cat <<EOT > "$TAIL\_SCRIPT"

#!/usr/bin/tail -n 2

Aquest compte ha estat tancat per l'administrador del sistema.

Si us plau, contacteu amb l'administrador.

EOT

# Fer executable el script

chmod +x "$TAIL\_SCRIPT"

if [ $? -eq 0 ]; then

echo "Tail script creat a $TAIL\_SCRIPT."

else

echo "Error en crear el tail script."

exit 5

fi

# 6. Canviar el shell de l'usuari pel tail script

echo "Canviant el shell de l'usuari..."

chsh -s "$TAIL\_SCRIPT" "$USUARI"

if [ $? -eq 0 ]; then

echo "Shell canviat correctament per a l'usuari '$USUARI'."

else

echo "Error en canviar el shell de l'usuari."

exit 6

fi

# 7. Finalització

echo "Procés completat. L'usuari '$USUARI' ha estat desactivat i el seu backup es troba a $BACKUP\_FILE."

exit 0



**Comprova que s'ha fet correctament**

Primer creem una altra vegada l’usuari prova amb els pasos anteriors, i executem el scrpit.



Comprovem:

1. Els fitxers estan borrats



1. El backup conté tots els fitxers



1. L’usuari prova no pot iniciar sessió:



# **Usuari especial asosh**

**A Unix hi ha comandes com el shutdown per apagar la màquina que només pot executar l'usuari root. En moltes ocasions pot ser interessant que algun altre usuari pugui apagar també la màquina però sense que tingui accés als privilegis de root.**

**Per aconseguir-ho es demana que creeu un compte especial que serveixi per executar un shell simplificat que permetrà fer shutdown i altres tasques especials amb permisos de superusuari. L'username corresponent serà asosh, i el password que decidiu. Quan algú faci un login en aquest compte s'executarà l'script asosh que hauríeu de tenir instal·lat de la pràctica anterior d'aplicacions. Per raons de seguretat cal que us assegureu que quan s'entra amb aquest compte no s'executa cap shell script. Quins permisos posaríeu a aquesta aplicació perquè no pugui ser executat per cap usuari directament?**

**useradd -m asosh:** Crear l'usuari asosh amb un directori /home propi.

**passwd asosh:** Assignar una contrasenya a l'usuari asosh.

**nano /home/asosh/asosh.sh:** Crear un script (asosh.sh) per executar com a shell de l'usuari asosh.

#!/bin/bash

# Mostrar un menú simple al usuario

echo "1. reboot"

echo "2. shutdown"

echo "e. exit"

read -p "Insereix un nombre: " command

# Evaluar la opción ingresada

if [ "$command" -eq 1 ]; then

echo "Reiniciant el sistema..."

# Ejecutar el reinicio con permisos adecuados

exec sudo /usr/sbin/shutdown -r now

elif [ "$command" -eq 2 ]; then

echo "Apagant el sistema..."

# Ejecutar el apagado con permisos adecuados

exec sudo /usr/sbin/shutdown now

elif [ "$command" == "e" ]; then

echo "Sortint de l'usuari ASOSH..."

exit 0

else

echo "Nombre no valid"

exit 1

fi

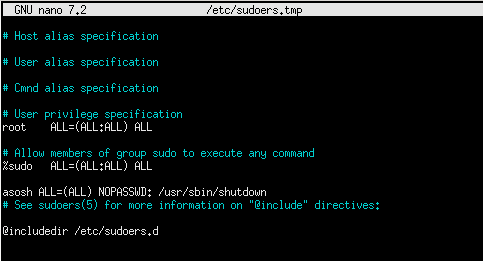


**chown asosh:asosh /home/asosh/asosh.sh:** Assignar la propietat del script asosh.sh a l'usuari asosh.ch

**chmod u+s /home/asosh/asosh.sh:** Establir el bit SetUID perquè el script s'executi amb permisos del propietari (no és necessari en aquest cas, pots ometre-ho).

**chmod 700 /home/asosh/asosh.sh:** Donar permisos per llegir, escriure i executar només a l'usuari asosh.

**visudo (**en cas que no es trobi instal·lat fem **apt install sudo):** Afegir la línia asosh ALL=(ALL) NOPASSWD: /usr/sbin/shutdown per permetre a asosh executar el comando shutdown sense contrasenya.

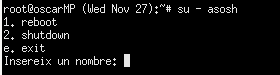


**nano /etc/passwd:** Modificar la línia de l'usuari asosh per utilitzar el script asosh.sh com a shell:

**asosh:x:1001:1001::/home/asosh:/home/asosh/asosh.sh**

****

Finalment executem **su - asosh** per entrar a l’usuari i comprovem que entra correctament a l’usuari



Com queda finalment l'entrada de la base de dades d'usuaris per a l'usuari **asosh?**

Executem nano /etc/passwd i veiem l’usuari asosh d’aquesta manera:

# **Sudo i control d'execució d'aplicacions**

**Com el shutdown hi ha altres comandes d'administració que només poden ser executades per el superusuari. És una mala pràctica de seguretat utilitzar el compte del superusuari per executar aquestes comandes. Per resoldre això es pot utilitzar la comanda *sudo. Sudo* permet executar una comanda a un usuari autoritzat com superusuari o un altre usuari. L'especificació de quines aplicacions pot executar un determinat usuari es defineix al fitxer /etc/sudoers. Aquest fitxer es pot editar de forma segura fent servir la comanda visudo.**

**Feu els canvis necessaris perquè els membres del grup admin puguin executar qualsevol comanda amb privilegis de superusuari.**

Primer instal·lem sudo fent:

apt update

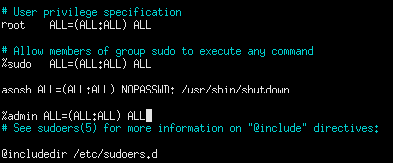
apt install sudo

Obrir /etc/sudoers usant:

visudo /etc/sudoers

I afegim:

%admin ALL=(ALL:ALL) ALL



**Feu els canvis necessaris perquè els usuaris PO puguin executar l'script per esborrar els usuaris que heu creat abans i tots els binaris que siguin al directori /usr/local/PO/bin.**

Creem la carpeta /usr/local/PO/bin fent:

mkdir -p /usr/local/PO/bin



Per a que el grup PO sigui propietari de la carpeta fem:

chown root:PO /usr/local/PO/bin



I donem permisos per executar binaris amb la comanda:

chmod 750 /usr/local/PO/bin/gestio\_usuaris.sh



Copiem l’script a la carpeta /usr/local/PO/bin amb la comanda:

cp <ruta al directori on hem guardat l’script> /usr/local/PO/bin

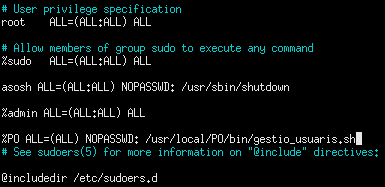


Obrim /etc/sudoers usant:

visudo /etc/sudoers

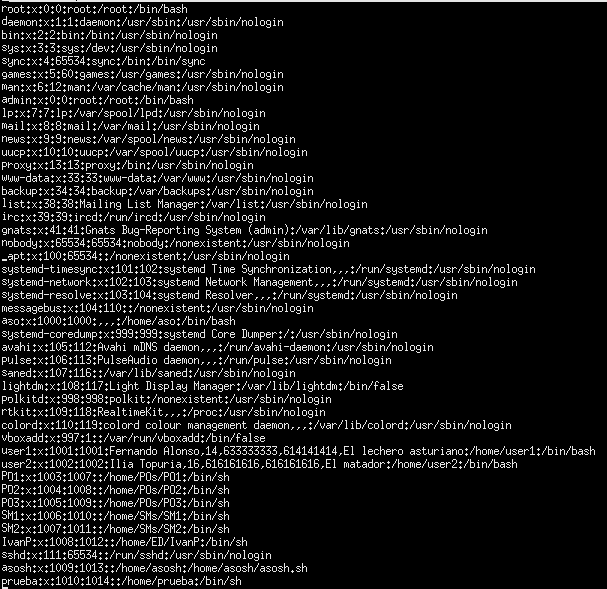
I afegim:

%PO ALL=(ALL) NOPASSWD: /usr/local/PO/bin/gestio\_usuaris.sh

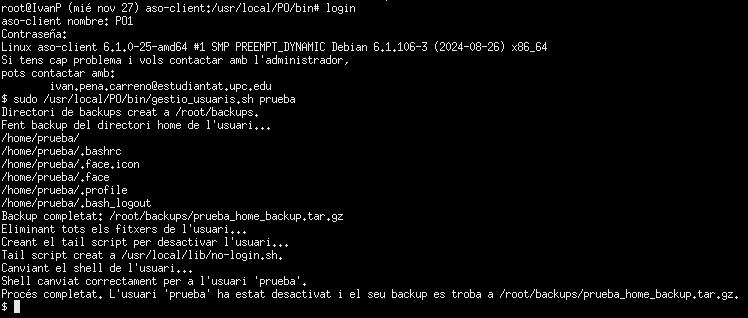


Comproveu que això funciona executant la comanda **vipw.**

Executem vipw i comprovem que l’usuari està actiu:



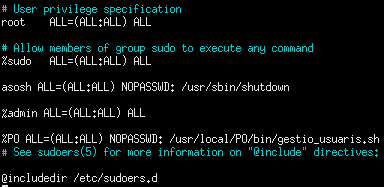
Entrem a l’usuari PO1 i executem l’script:



I comprovem amb vipw el resultat:



Quins canvis heu fet al fitxer /etc/sudoers per activar els controls anteriors?



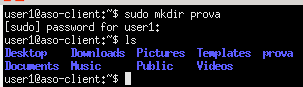
Hem afegit les línies asosh, %admin i %PO.

**Finalment desactiveu el compte del root de tal forma no es pugui fer *login* com superusuari. Les comandes d'administració es podran fer només des dels comptes del grup admin o fent ús de l'usuari asosh. Assegureu-vos que podeu fer comandes des d'un usuari administrador abans de desactivar-ho.**

Abans de fer res amb l’usuari root, hem de comprovar que els usuaris del grup admin puguin executar comandes amb sudo.

Per fer això primer hem d’activar aquests usuaris en cas que no els haguem activat abans amb la comanda

I comprovem que funcioni el sudo.



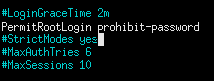
Fem la següent comanda:

usermod -s /sbin/nologin root



Després desactivem ssh obrint /etc/ssh/sshd\_config i descomentant la linia:

PermitRootLogin prhibit-password



Un cop desats els canvis, reiniciem el servei ssh amb sudo systemctl restart sshd.



Finalment verifiquem que podem entrar a root fent su.

