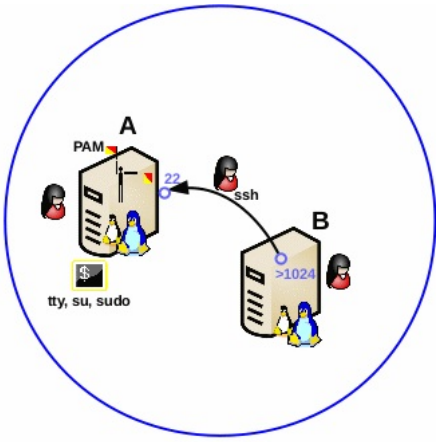
# Práctica Seguridade Informática: PAM

**ESCENARIO**

**Máquinas virtuais ou físicas:**
RAM ≤ 2048MB  CPU ≤ 2  PAE/NX habilitado
Rede: 192.168.120.0/24
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

**Máquina A:**
Rede Interna(eth0) e NAT(eth1)
IP/MS: 192.168.120.100/24
Servidor SSH
SO: Debian amd64 xfce instalado
sda: SO instalado
User: usuario
Passwd: abc123.
Groups usuario: usuario sudo
usuario ➜ sudo su - ➜ root

**Máquina B:**
Rede Interna(eth0)
IP/MS: 192.168.120.101/24
Cliente SSH
SO: Live GNU/Linux




**LIMITACIÓN DE RESPONSABILIDADE** O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

**NOTA**:

- Documentación de interese:

  **Linux-PAM**  **Fedora - Guía de Seguridad: PAM**  **Arquitectura PAM**

  **Pluggable Authentication Modules (PAM)**

- Configuración: **/etc/pam.d/ (man pam.d)**. Antigamente: /etc/pam.conf
  - Se non existe o directorio /etc/pam.d/ o ficheiro /etc/pam.conf segue vixente.
  - Se existe o directorio /etc/pam.d/ o ficheiro /etc/pam.conf ignórase.

  Funcionan mediante regras.

  - /etc/pam.d/ contén ficheiros os cales son os servizos PAM
  - /etc/pam.conf identifica en cada liña un servizo PAM
  - Nos ficheiros dentro de /etc/pam.d/ e en /etc/pam.conf:
    - Comentarios: Liñas en branco e liñas que comezan co caracter #
    - Se non é comentario ➜ Cada liña é unha regra:
      - As regras dentro de cada ficheiro /etc/pam.d non inclúen o nome do servizo, xa que este e considerado co propio nome do ficheiro (servizo).
      - As regras dentro de /etc/pam.conf deben incluír o nome do servizo para identificar a que servizo se lle aplica cada regra.
      - Calquera erro nunha regra pode provocar calquera cousa inesperada na autenticación do usuario: problema de acceso, acceso indebido...
      - As regras vanse lendo de forma secuencial de arriba a abaixo. Así, se por exemplo nun ficheiro temos 2 liñas e en cada liña unha regra, primeiro lese a regra da liña 1 e logo, se é o caso (por non ser suficiente coa regra da liña 1), lese a regra da liña 2.

| Configuración | Descrición |
|---|---|
| **/etc/pam.conf** | **Estrutura regra**<br>service type control module-path module-argument<br><br>**service** ⇒ Nome do servizo a configurar con PAM. O conxunto de regras dun servizo forman o que denomina pila. A pila lese de arriba a abaixo secuencialmente.<br><br>**tipo** ⇒ Indicar o tipo PAM que emprega o servizo. Existen 4 e cada un indica un aspecto no proceso de autorización.<br><br>**control** ⇒ Indicar que facer no caso de éxito ou fracaso na regra establecida<br><br>**module-path** ⇒ Indicar en que ruta existe o módulo no que ten efecto a regra establecida<br><br>**module-argument** ⇒ Indicar, se é o caso, opcións sobre a chamada ao módulo indicado no campo *module-path* |
| **/etc/pam.d/** | **Estrutura regra**<br>type control module-path module-argument<br><br>O nome do ficheiro indica xa o nome do servizo.<br><br>**Campos equivalentes a /etc/pam.conf** |
| **Tipos** | **Existen catro tipos (asociados a módulos co mesmo nome)**<br>auth<br>account<br>password<br>session<br><br>**auth** ⇒ Validar que o usuario é quen di ser (Non repudio) e unha ver verificado poder otorgar permisos. Exemplo: pedir/validar contrasinal<br><br>**account** ⇒ Verificar que o acceso está permitido. Exemplo: Usuario non ten a conta deshabilitada.<br><br>**password** ⇒ Actualizar token do usuario. Exemplo: validar ao modificar un contrasinal que posúa un número mínimo de caracteres.<br><br>**session** ⇒ Tarefas a facer antes/despois de conceder ao usuario acceso a un servizo. Exemplo: montaxe de directorios |
| **Control** | **Sintaxe antiga: Bandeiras(flag) de control**<br>required<br>requisite<br>sufficient<br>optional<br>include<br>substack<br><br>**Sintaxe moderna: [value1=action1 value2=action2 ...]**<br>**Existen pares de valores equivalentes a bandeiras de control da sintaxe antiga**<br>required ⇔ [success=ok new_authtok_reqd=ok ignore=ignore default=bad]<br>requisite ⇔ [success=ok new_authtok_reqd=ok ignore=ignore default=die]<br>sufficient ⇔ [success=done new_authtok_reqd=done default=ignore]<br>optional ⇔ [success=ok new_authtok_reqd=ok default=ignore]<br><br>**required** ⇒ Indicar que na regra que existe o módulo debe ter éxito. Se dá erro este non será visible ata que se executen o resto de módulos (regras) para este servizo e mesmo tipo, é dicir, se o erro prodúcese no tipo auth éste non será visible ata que se executen todos os tipos auth deste módulo (servizo). Importante dende o punto de vista de seguridade xa que ninguén pode prever de que módulo da pila ven este erro, o cal daría información valiosa sobre o funcionamento do aplicativo<br><br>**requisite** ⇒ Equivalente ao anterior pero se dá erro este será visible inmediatamente, sen esperar a que se executen o resto de módulos (regras), enviando o control directamente á aplicación que o chamou. Importante dende o punto de vista de seguridade xa ao amosara información inmediata do fallo dáse información valiosa sobre o funcionamento do aplicativo, pero polo contra pode interromper inmediatamente o acceso ao aplicativo. |

**sufficient** ⇒ Se ten éxito e ningún módulo previo *required* tivo fallo xa é suficiente para devolver o éxito á aplicación e non seguir revisando módulos. Se ten fallo ignórase e séguese revisando a pila de módulos.

**optional** ⇒ O éxito ou fracaso deste módulo só é importante se é o único módulo na pila asociada a este servizo + tipo.

**include** ⇒ Incluír todas as liñas do ficheiro de configuración especificado como argumento na súa chamada.

**substack** ⇒ Similar a include:

- Include equivale a copiar o código no lugar da chamada.
- Substack equivale a chamar dende ese punto a unha función e devolve a resposta ao lugar da chamada, o cal permite que se poida ignorar.

## Servizo login

$ cat /etc/pam.d/login | sed '/^$/d' | grep -v '#' #Amosar do contido do ficheiro /etc/pam.d/login soamente as regras, evitando liñas en branco e comentarios.

```
auth optional pam_faildelay.so delay=3000000
auth [success=ok new_authtok_reqd=ok ignore=ignore user_unknown=bad default=die] pam_securetty.so
auth requisite pam_nologin.so
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so close
session required pam_loginuid.so
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
session required pam_env.so readenv=1
session required pam_env.so readenv=1 envfile=/etc/default/locale
@include common-auth
auth optional pam_group.so
session required pam_limits.so
session optional pam_lastlog.so
session optional pam_motd.so motd=/run/motd.dynamic
session optional pam_motd.so noupdate
session optional pam_mail.so standard
session optional pam_keyinit.so force revoke
@include common-account
@include common-session
@include common-password
```

- No **Exemplo1. Módulo pam_nologin.so** imos verificar o funcionamento da regra existente no servizo login correspondente ao módulo pam_nologin.
- No **Exemplo2. Módulo pam_limits.so** imos verificar o funcionamento da regra existente no servizo login correspondente ao módulo pam_limits.
- No **Exemplo3. Módulo pam_lastlog.so** imos verificar o funcionamento da regra existente no servizo login correspondente ao módulo pam_lastlog.
- No **Exemplo4. Módulo pam_motd.so** imos verificar o funcionamento da regra existente no servizo login correspondente ao módulo pam_motd.
- No **Exemplo5. Módulo pam_unix.so** imos verificar o funcionamento das regras existentes no servizo login correspondentes ao módulo pam_unix. Estas regras inclúense no servizo login mediante os ficheiros correspondentes:

```
/etc/pam.d/login → @include common-auth → /etc/pam.d/common-auth → regra pam_unix
/etc/pam.d/login → @include common-account → /etc/pam.d/common-account → regra pam_unix
/etc/pam.d/login → @include common-session → /etc/pam.d/common-session → regra pam_unix
/etc/pam.d/login → @include common-password → /etc/pam.d/common-password → regra pam_unix
```

### Máquina virtual A: Debian amd64

1. Na contorna gráfica abrir un terminal e executar:
   usuario@debian:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   usuario@debian:~$ passwd usuario #Cambiar o contrasinal do usuario usuario. Por como contrasinal **abc123.** (Ollo que o contrasinal ten un caracter punto final).

2. Cambiar hostname da máquina virtual A. Por debianA como hostname:
   usuario@debian:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@debian:~# echo 'debianA' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@debian:~# echo 'kernel.hostname=debianA' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@debian:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@debian:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.
   usuario@debian:~$ exit #Pechar o terminal saíndo da consola local do usuario **usuario**.

3. Configurar a rede:
   Na contorna gráfica abrir un terminal e executar:

   usuario@debianA:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@debianA:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@debianA:~# systemctl disable avahi-daemon #Impide que o servizo avahi-daemon sexa iniciado no arranque xerando os links K* nos runlevels (/etc/rcX.d)
   root@debianA:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a configuración de rede e non ter conflicto con este xestor.
   root@debianA:~# systemctl disable network-manager #Impide que o servizo network-manager sexa iniciado no arranque xerando os links K* nos runlevels (/etc/rcX.d)
   root@debianA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), interna(enp0s3) e NAT(enp0s8).

   ```
   $ man interfaces #Ver ás páxinas de manual referente ao ficheiro de configuración de rede /etc/network/interfaces
   $ cat /etc/network/interfaces #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces
   $ ls -l /etc/network/interfaces.d #Listar de forma extendida o contido do directorio /etc/network/interfaces/setup
   $ cat /etc/network/interfaces.d/setup #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces/setup
   ```

   root@debianA:~# cat > /etc/network/interfaces.d/setup <<EOF #Comezo do ficheiro a crear /etc/network/interfaces.d/setup

   ```
   auto lo
   iface lo inet loopback

   auto enp0s3
   iface enp0s3 inet static
     address 192.168.120.100/24

   auto enp0s8
   iface enp0s8 inet dhcp
   ```

   EOF #Fin do ficheiro a crear /etc/network/interfaces.d/setup
   root@debianA:~# /etc/init.d/networking status #Comprobar o estado do demo networking, é dicir, comprobar se está activa a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
   root@debianA:~# /etc/init.d/networking start #Arrancar o demo networking, é dicir, activar a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
   root@debianA:~# /etc/init.d/networking status #Comprobar o estado do demo networking, é dicir, comprobar se está activa a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
   root@debianA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), interna(enp0s3) e NAT(enp0s8).
   root@debianA:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local enp0s3

4. Comprobar estado do Servidor SSH:

   ```
   # apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
   # apt -y install netcat #Instalar o paquete netcat, é dicir, instalar o paquete que integra o comando nc. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
   # dpkg -l net-tools ; [ $(echo $?) -eq '1' ] && apt update && apt -y install net-tools #Verificar se o paquete net-tools está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase. O paquete net-tools é necesario para poder empregar comandos coma: ifconfig, netstat, route e arp.
   # dpkg -l openssh-server ; [ $(echo $?) -eq '1' ] && apt update && apt -y install openssh-server #Verificar se o paquete openssh-server está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase.
   ```

   root@debianA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.
   root@debianA:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   root@debianA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en

estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.

root@debianA:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.

root@debianA:~# /etc/init.d/ssh start #Arrancar o servidor SSH.

root@debianA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.

root@debianA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*

root@debianA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)

root@debianA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*

root@debianA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled

root@debianA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# ssh -v usuario@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario **usuario** e o seu contrasinal. Se é a primeira ver que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.

usuario@debianA:~$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$

### Máquina virtual B: Kali amd64

5. Configuración da rede. Na contorna gráfica abrir un terminal e executar:
   kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
   root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflicto con este demo.
   root@kali:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflicto con este xestor.
   root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
   root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
   root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
   root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
   root@kali:~# echo '192.168.120.100 debianA' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome debianA, para que atenda á IP 192.168.120.100
   root@kali:~# ping -c4 debianA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A

6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:
   root@kali:~# echo 'kaliB' > /etc/hostname #Indicar ao sistema o valor do hostname.
   root@kali:~# echo 'kernel.hostname=kaliB' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
   root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
   root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.
   kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario **kali**.

7. **B** $\overset{\text{SSH}}{\rightarrow}$ **A** Acceder mediante SSH dende a máquina virtual B á máquina virtual A. Dende agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:
   Na contorna gráfica abrir un terminal e executar:

   kali@kaliB:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
   kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   kali@kaliB:~$ nc -vz debianA 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
   kali@kaliB:~$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario **usuario** a través da conexión cifrada SSH.
      usuario@debianA:~$

8. **Exemplo1. Módulo pam_nologin.so**

Imos verificar o funcionamento da **regra** existente no **servizo login** correspondente ao **módulo pam_nologin**.

**pam_nologin** é un módulo PAM que impide aos usuarios iniciar sesión no sistema cando /var/run/nologin ou /etc/nologin existen (verificados nesa orde). O contido do ficheiro móstrase ao usuario, e no caso que existan os 2 ficheiros amósase o contido de /var/run/nologin. O módulo pam_nologin non ten efecto na capacidade do usuario root para iniciar sesión.
**Está asociado aos tipos auth e session.**

A. Executar:
usuario@debianA:~$ ls -ld /var/run #Listar soamente os permisos do cartafol /var/run, é dicir, listar os permisos do propio cartafol pero non os do seu contido. Nesta caso como se pode observar é unha ligazón ao directorio /run
```
lrwxrwxrwx 1 root root 4 Nov 16 2019 /var/run -> /run
```

usuario@debianA:~$ grep pam_nologin /etc/pam.d/login #Buscar o patrón pam_nologin no ficheiro /etc/pam.d/login
```
auth requisite pam_nologin.so
```

> Esta regra existe antes que a regra password (@include common-password) e como o flag é requisite o erro amosarase antes de poder introducir un contrasinal.

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# echo 'Acceso non autorizado para usuarios sen privilexios' > /etc/nologin #Xerar o ficheiro /etc/nologin con contido. Ese contido será amosado logo que o usuario intente acceder ao sistema.

B. En debianA acceder á consola tty1 mediante o usuario **usuario**. Que é o que acontece?

> O usuario *usuario*, así como calquera usuario que non sexa **root** non pode acceder ao sistema amosando un erro (o contido do ficheiro /etc/nologin)

C. En debianA acceder á consola gráfica mediante o usuario **usuario**. Que é o que acontece?

> O usuario *usuario*, así como calquera usuario que non sexa **root** non pode acceder ao sistema amosando un erro (que non é o contido do ficheiro /etc/nologin)

D. En debianA acceder mediante ssh co usuario **usuario**. Que é o que acontece?

> O usuario *usuario*, así como calquera usuario que non sexa **root** non pode acceder ao sistema amosando un erro (o contido do ficheiro /etc/nologin)

E. Executar:
root@debianA:~# echo '1-Acceso non autorizado para usuarios sen privilexios' > /var/run/nologin #Xerar o ficheiro /var/run/nologin con contido. Ese contido será amosado logo que o usuario intente acceder ao sistema.

F. En debianA acceder de novo á consola tty1, consola gráfica e ssh mediante o usuario **usuario**. Que é o que acontece?

> Pois acontece o mesmo, é dicir, o usuario *usuario*, así como calquera usuario que non sexa **root** non pode acceder ao sistema amosando un erro (o contido do ficheiro /var/run/nologin ou no caso da consola gráfica outro erro)

G. Executar:
root@debianA:~# echo '2-Acceso non autorizado para usuarios sen privilexios' > /etc/nologin #Xerar o ficheiro /etc/nologin con contido. Ese contido será amosado logo que o usuario intente acceder ao sistema.

H. En debianA acceder de novo á consola tty1, consola gráfica e ssh mediante o usuario **usuario**. Que é o que acontece?

> Pois acontece o mesmo, é dicir, o usuario *usuario*, así como calquera usuario que non sexa **root** non pode acceder ao sistema amosando un erro (o contido do ficheiro /var/run/nologin ou no caso da consola gráfica outro erro). Así, independentemente de cal sexa o último ficheiro xerado(/var/run/nologin e /etc/nologin) sempre prevalece o contido do ficheiro /var/run/nologin porque é o primeiro ficheiro a verificar a súa existencia.

I. Executar:
root@debianA:~# reboot #Reiniciar o sistema operativo.

> **De interese**: Verificar o que acontece co comando *reboot* pois unha vez reiniciado o sistema os ficheiros /var/run/nologin e /etc/nologin elimínanse no caso de existir.
> **man systemd-user-sessions**: systemd-user-sessions.service é un servizo que controla os inicios de sesión dos usuarios a través de pam_nologin (8). Despois de completar a inicialización básica do sistema, elimina /run/nologin, permitíndo así inicios de sesión. Antes do apagado do sistema, crea /run/nologin, prohibindo así máis inicios de sesión.

9. **Exemplo2. Módulo pam_limits.so**

Imos verificar o funcionamento da **regra** existente no **servizo login** correspondente ao **módulo pam_limits**.

**pam_limits** é un módulo PAM que establece límites nos recursos do sistema que se poden obter nunha sesión de usuario. Os usuarios de uid=0 (root) tamén se ven afectados por estes límites. Por defecto, os límites tómanse do ficheiro de configuración /etc/security/limits.conf e logo de /etc/security/limits.d/*.conf. Os ficheiros analízanse un despois doutro na orde de configuración local "C". O efecto dos ficheiros individuais é o mesmo que se todos os ficheiros concatenáronse xuntos na orde de análise. Se hai un ficheiro de configuración especificado explícitamente cunha opción de módulo, entón os ficheiros no directorio anterior non son analizados.
**Só está asociado ao tipo session.**

A. Executar:

usuario@debianA:~$ man limits.conf #Ver ás páxinas de manual referente ao ficheiro de configuración /etc/security/limits.conf. A sintaxe deste ficheiro aplícase tamén nos ficheiros /etc/security/limits.d/*.conf

usuario@debianA:~$ help ulimit #Ver a axuda do comando ulimit, o cal permite modificar os límites de recursos dispoñibles para o shell e os procesos que crea.

usuario@debianA:~$ grep pam_limits /etc/pam.d/login #Buscar o patrón pam_limits no ficheiro /etc/pam.d/login

```
session required pam_limits.so
```

Esta regra posúe o flag required polo que en caso de erro non se amosará ata que se executen o resto de módulos.

usuario@debianA:~$ ls -l /etc/security/limits.conf #Listar de forma extendida o ficheiro /etc/security/limits.conf

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# echo -e 'usuario\t\t -\t maxlogins\t 1' >> /etc/security/limits.conf #Engadir ao ficheiro /etc/security/limits.conf unha nova entrada, onde cada campo irá separado por un tabulado horizontal. Os campos indican:

- usuario → Nome do usuario a quen lle afecta o límite xerado.
- - → Sen límite de recurso soft ou hard.
- maxlogins → Limitar o número máximo de logins permitidos (neste caso para o usuario de nome *usuario*)
- 1 → Número máximo de intentos permitidos.

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$ w usuario #O comando **w** amosa información sobre que usuarios están conectados e que están a facer no momento de execución deste comando (w). Como estamos indicándolle o nome do usuario no comando pois soamente amosa información dese usuario (neste caso do usuario de nome *usuario*)

usuario@debianA:~$ su - usuario #Acceder como usuario *usuario*. Unha vez insertado o contrasinal *abc123*. non é posible o acceso debido ao límite que acabamos de configurar, posto que soamente é permitido 1 login co usuario *usuario* e este xa ten acceso, pois é co usuario que estamos a traballar.

```
Password:
Too many logins for 'usuario'.
su: cannot open session: Permission denied
```

B. En debianA acceder á consola tty1 mediante o usuario **usuario**. Que é o que acontece?

O mesmo que antes. Unha vez insertado o contrasinal *abc123*. non é posible o acceso debido ao límite que acabamos de configurar, posto que soamente é permitido 1 login co usuario *usuario* e este xa ten acceso, pois é co usuario que estamos a traballar.

C. En debianA acceder mediante ssh co usuario **usuario**. Que é o que acontece?

O mesmo que antes. Unha vez insertado o contrasinal *abc123*. non é posible o acceso debido ao límite que acabamos de configurar, posto que soamente é permitido 1 login co usuario *usuario* e este xa ten acceso, pois é co usuario que estamos a traballar.

D. En debianA voltar a realizar o procedemento anterior pero agora para o usuario **root**. Que é o que acontece?

O mesmo que antes, xa que **pam_limits tamén afecta ao uid=0**.

E. Executar:

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# sed -i 's/^usuario/#usuario/' /etc/security/limits.conf #Comentar no ficheiro /etc/security/limits.conf a entrada correspondente ao usuario *usuario*, de tal forma que agora o usuario poida iniciar máis de 1 vez sesión no sistema operativo.

10. **Exemplo3. Módulo pam_lastlog.so**

Imos verificar o funcionamento da **regra** existente no **servizo login** correspondente ao **módulo pam_lastlog**.

**pam_lastlog** é un módulo PAM que amosa unha liña de información sobre o último inicio de sesión do usuario e permite bloquear contas según a súa inactividade de inicio de sesión. Ademais, o módulo mantén o ficheiro /var/log/lastlog.
**Está asociado aos tipos auth, account e session:**

- auth e account: Permiten bloquear as contas de usuarios (uid !=0) se a conta estivo inactiva(sen iniciar sesión) durante un número de días(por defecto 90). A comprobación non se realiza para a conta root polo que root nunca é bloqueado.
- session: Permite amosar unha liña de información sobre o último inicio de sesión do usuario.

A. Executar:
usuario@debianA:~$ grep pam_lastlog /etc/pam.d/login #Buscar o patrón pam_lastlog no ficheiro /etc/pam.d/login
```
session optional pam_lastlog.so
```
Esta regra posúe o flag optional polo que en caso de acerto/erro non inflúe no resto dos módulos.

usuario@debianA:~$ su - usuario #Acceder como usuario *usuario*. Unha vez insertado o contrasinal *abc123.* e antes de aparecer o prompt tal e como comentamos debería amosarse unha liña con información sobre o último inicio de sesión, pero non se amosa. Por que? Porque o comando **su** ten o seu propio servizo en /etc/pam.d/su e polo tanto **su** non se rixe polo servizo /etc/pam.d/login

$ grep pam_lastlog /etc/pam.d/su #Non amosa resultados debido a que dentro do servizo /etc/pam.d/su non existe ningunha referencia ao módulo pam_lastlog

usuario@debianA:~$ exit #Saír da consola local de **usuario** á que acabamos de acceder para voltar á consola local de **usuario**.

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo). Unha vez insertado o contrasinal *abc123.* e antes de aparecer o prompt tal e como comentamos debería amosarse unha liña con información sobre o último inicio de sesión, pero non se amosa. Por que? Pois, polo mesmo que no comando **su**, o comando **sudo** ten o seu propio servizo en /etc/pam.d/sudo e polo tanto non se rixe polo servizo /etc/pam.d/login

$ grep pam_lastlog /etc/pam.d/sudo #Non amosa resultados debido a que dentro do servizo /etc/pam.d/sudo non existe ningunha referencia ao módulo pam_lastlog

B. En debianA acceder á consola tty1 mediante o usuario **usuario**. Que é o que acontece?

Ao usuario *usuario*, logo de insertar o contrasinal e antes de aparecer o prompt amósase unha liña con información sobre o último inicio de sesión.

C. En debianA acceder mediante ssh co usuario **usuario**. Que é o que acontece?

Ao usuario *usuario*, logo de insertar o contrasinal e antes de aparecer o prompt amósase unha liña con información sobre o último inicio de sesión.

D. Executar:
root@debianA:~# A=$(grep -n 'pam_lastlog' /etc/pam.d/login | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (pam_lastlog) no ficheiro /etc/pam.d/login e gardalo na variable A

root@debianA:~# sed -i "${A}s/^session/#session/" /etc/pam.d/login #Comentar no ficheiro /etc/pam.d/login a entrada correspondente á regra do módulo *pam_lastlog*

E. En debianA voltar a realizar o procedemento anterior, é dicir, voltar a acceder co usuario *usuario* dende tty1 e ssh. Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login comentando o módulo pam_lastlog, en tty1 non se amosará ningunha liña informativa sobre o último inicio de sesión, pero no acceso por ssh SI, xa que ssh posúe o seu propio servizo en /etc/pam.d/sshd e polo tanto non se rixe polo servizo /etc/pam.d/login. Así, existe unha entrada no arquivo de configuración **/etc/ssh/sshd_config** que amosa unha liña informativa co último inicio de sesión:

**PrintLastLog yes**

F. En debianA voltar a activar a regra pam_lastlog no servizo login:

root@debianA:~# A=$(grep -n 'pam_lastlog' /etc/pam.d/login | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (pam_lastlog) no ficheiro /etc/pam.d/login e gardalo na variable A

root@debianA:~# sed -i "${A}s/^#session/session/" /etc/pam.d/login #Comentar no ficheiro /etc/pam.d/login a entrada correspondente á regra do módulo *pam_lastlog*

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$

11. **Exemplo4. Módulo pam_motd.so**

Imos verificar o funcionamento da **regra** existente no **servizo login** correspondente ao **módulo pam_motd**.

**pam_motd** é un módulo PAM que amosa un ficheiro (mensaxe do día), por defecto /etc/motd, e/ou un conxunto de ficheiros contidos nun directorio, por defecto /etc/motd.d/. O tamaño da mensaxe está limitada a 64KB. A mensaxe é amosada antes do prompt do sistema unha vez iniciada a sesión..
**Só está asociado ao tipo session.**

A. Executar:

usuario@debianA:~$ grep pam_motd /etc/pam.d/login #Buscar o patrón pam_motd no ficheiro /etc/pam.d/login

```
session optional pam_motd.so motd=/run/motd.dynamic
session optional pam_motd.so noupdate
```

Estas regras posúen o flag optional polo que en caso de acerto/erro non inflúen no resto dos módulos.
- A primeira regra amosa o contido do ficheiro **/run/motd.dynamic**, se éste existe.
- A segunda regra como non especifica ficheiro amosa o contido do ficheiro **/etc/motd**, se éste existe, xa que é o ficheiro por defecto a amosar. A opción **noupdate** non executa os scripts existentes en **/etc/update-motd.d** para refrescar o ficheiro motd.

usuario@debianA:~$ su - usuario #Acceder como usuario *usuario*. Unha vez insertado o contrasinal *abc123*. e antes de aparecer o prompt tal e como comentamos debería amosarse o contido da mensaxe do día, pero non se amosa. Por que? Porque o comando **su** ten o seu propio servizo en /etc/pam.d/su e polo tanto **su** non se rixe polo servizo /etc/pam.d/login

$ grep pam_motd /etc/pam.d/su #Non amosa resultados debido a que dentro do servizo /etc/pam.d/su non existe ningunha referencia ao módulo pam_motd

usuario@debianA:~$ exit #Saír da consola local de **usuario** á que acabamos de acceder para voltar á consola local de **usuario**.

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo). Unha vez insertado o contrasinal *abc123*. e antes de aparecer o prompt tal e como comentamos debería amosarse a mensaxe do día, pero non se amosa. Por que? Pois, polo mesmo que no comando **su**, o comando **sudo** ten o seu propio servizo en /etc/pam.d/sudo e polo tanto non se rixe polo servizo /etc/pam.d/login

$ grep pam_motd /etc/pam.d/sudo #Non amosa resultados debido a que dentro do servizo /etc/pam.d/sudo non existe ningunha referencia ao módulo pam_motd

B. En debianA acceder á consola tty1 mediante o usuario **usuario**. Que é o que acontece?

Ao usuario *usuario*, logo de insertar o contrasinal e antes de aparecer o prompt amósase unha liña coa mensaxe do día referenciada en /run/motd.dynamic e /etc/motd.

C. En debianA acceder mediante ssh co usuario **usuario**. Que é o que acontece?

Ao usuario *usuario*, logo de insertar o contrasinal e antes de aparecer o prompt amósase unha liña coa mensaxe do día referenciada en /run/motd.dynamic e /etc/motd.

D. Executar:

root@debianA:~# A=$(grep -n 'pam_motd' /etc/pam.d/login | cut -d':' -f1 | xargs | tr ' ' ',') #Atopar as liñas onde aparece o patrón buscado (pam_motd) no ficheiro /etc/pam.d/login e gardalo na variable A, tal que o número de liñas atopadas estarán separadas por un caracter coma

root@debianA:~# sed -i "${A}s/^session/#session/" /etc/pam.d/login #Comentar no ficheiro /etc/pam.d/login as entradas correspondentes ás regras do módulo *pam_motd*

E. En debianA voltar a realizar o procedemento anterior, é dicir, voltar a acceder co usuario *usuario* dende tty1 e ssh. Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login comentando o módulo pam_motd, en tty1 non se amosará ningunha mensaxe do día, pero no acceso por ssh SI, xa que ssh posúe o seu propio servizo en /etc/pam.d/sshd e polo tanto non se rixe polo servizo /etc/pam.d/login

$ grep pam_motd /etc/pam.d/sshd #Amosa resultados debido a que dentro do servizo /etc/pam.d/sshd si existen referencias ao módulo pam_motd

```
session optional pam_motd.so motd=/run/motd.dynamic
session optional pam_motd.so noupdate
```

F. En debianA editamos de novo as regras pam_motd no servizo login:

root@debianA:~# echo 'BENVIDO a '$(hostname)'!!!' > /etc/motd2 #Crear o ficheiro /etc/motd2 cunha liña que amose: BENVIDO a debianA!!!.

root@debianA:~# A=$(grep -n 'pam_motd' /etc/pam.d/login | grep noupdate | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (noupdate) no ficheiro /etc/pam.d/login e gardalo na variable A

root@debianA:~# sed -i -e "${A}s/^#session/session/" -e "${A}s|noupdate|motd=/etc/motd2 noupdate|" /etc/pam.d/login #Descomentar no ficheiro /etc/pam.d/login a entrada correspondente a *noupdate* na regra do módulo *pam_motd* e activar que se amose como mensaxe do día o contido do ficheiro */etc/motd2*

G. En debianA voltar a realizar o procedemento anterior, é dicir, voltar a acceder co usuario *usuario* dende tty1 e ssh. Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login cambiando no módulo pam_motd o ficheiro mensaxe do día, en tty1 amosarase o contido dese ficheiro (/etc/motd2), pero no acceso por ssh NON, xa que ssh posúe o seu propio servizo en /etc/pam.d/sshd e polo tanto non se rixe polo servizo /etc/pam.d/login

H. En debianA voltar a restaurar as regras pam_motd no servizo login:

root@debianA:~# A=$(grep -n 'pam_motd' /etc/pam.d/login | cut -d':' -f1 | xargs | tr ' ' ',') #Atopar as liñas onde aparece o patrón buscado (pam_motd) no ficheiro /etc/pam.d/login e gardalo na variable A, tal que o número de liñas atopadas estarán separadas por un caracter coma

root@debianA:~# sed -i -e "${A}s/^#session/session/" -e "${A}s|motd=/etc/motd2 noupdate|noupdate|" /etc/pam.d/login #Descomentar no ficheiro /etc/pam.d/login as entradas correspondentes a pam_motd e modificar *noupdate* na regra do módulo *pam_motd* para deixar como mensaxe do día o contido do ficheiro por defecto */etc/motd*

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$

12. **Exemplo5. Módulo pam_unix.so**

Imos verificar o funcionamento das regras existentes no servizo login correspondentes ao módulo pam_unix. Estas regras inclúense no servizo login mediante os ficheiros correspondentes:

```
/etc/pam.d/login → @include common-auth → /etc/pam.d/common-auth → regra pam_unix
/etc/pam.d/login → @include common-account → /etc/pam.d/common-account → regra pam_unix
/etc/pam.d/login → @include common-session → /etc/pam.d/common-session → regra pam_unix
/etc/pam.d/login → @include common-password → /etc/pam.d/common-password → regra pam_unix
```

**pam_unix** é o módulo PAM de autenticación estándar Unix. Utiliza chamadas estándar das librarías do sistema para recuperar e configurar a información da conta, así como a autenticación. Normalmente isto é obtido a partir dos ficheiros **/etc/passwd** e **/etc/shadow**. A acción predeterminada do módulo pam_unix é non permitir ao usuario o acceso a un servizo (login neste caso) se é o seu contrasinal está en branco.
**Está asociado a todos os tipos: account, auth, password e session.**

**@include common-auth → /etc/pam.d/common-auth**

Imos verificar o funcionamento da **inclusión** do ficheiro **/etc/pam.d/common-auth** para o **servizo login** referente ao módulo **pam_unix**, é dicir, imos verificar a regra que se inclúe para pam_unix dende o ficheiro /etc/pam.d/common-auth no servizo login.

**common-auth** é un arquivo existente en /etc/pam.d que incorpora regras de tipo auth ao servizo que o chama, neste caso o servizo login.
**Só está asociado ao tipo auth.**

A. Executar:

usuario@debianA:~$ grep common-auth /etc/pam.d/login #Buscar o patrón common-auth no ficheiro /etc/pam.d/login

```
@include common-auth
```

usuario@debianA:~$ cat /etc/pam.d/common-auth | sed '/^$/d' | grep -v '#' #Amosar do contido do ficheiro /etc/pam.d/common-auth soamente as regras, evitando liñas en branco e comentarios.

```
auth [success=1 default=ignore] pam_unix.so nullok_secure
auth requisite pam_deny.so
auth required pam_permit.so
```

- A primeira regra refírese ao módulo pam_unix.
- A segunda regra refírese ao módulo pam_deny.
- A terceira regra refírese ao módulo pam_permit.

usuario@debianA:~$ grep pam_unix /etc/pam.d/common-auth #Buscar o patrón pam_unix no ficheiro /etc/pam.d/common-auth

```
auth [success=1 default=ignore] pam_unix.so nullok_secure
```

- nullok_secure → Anula o valor predeterminado e permite a calquera usuario cun contrasinal en branco acceder ao servizo sempre que o valor de PAM_TTY sexa establecido nun dos valores atopados no ficheiro /etc/securetty.

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo).

root@debianA:~# useradd -m -d /home/alumno -s /bin/bash -p '' alumno #Crear o usuario *alumno* co comando *useradd*, onde:
-d /home/alumno → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/alumno
-m → Copia na casa do usuario o que exista no cartafol /etc/skel
-s /bin/bash → Establece como shell de traballo para o usuario a shell bash
-p '' → Establece como contrasinal un contrasinal en branco
alumno → Establece como nome de autenticación de usuario o nome alumno

root@debianA:~# exit #Saír da consola local de **usuario** á que acabamos de acceder para voltar á consola local de **usuario**.

B. En debianA acceder á consola tty1 mediante o usuario **alumno**. Que é o que acontece?

O usuario *alumno*, inicia sesión sen contrasinal, é dicir, unha vez introducido *alumno* o servizo login non pregunta polo contrasinal e xa aparece o prompt do sistema.

C. En debianA acceder como alumno mediante o comando **su**. Que é o que acontece?

usuario@debianA:~$ su - alumno #Acceder como usuario *alumno*. Este usuario inicia sesión sen contrasinal xa que o servizo **/etc/pam.d/su** tamén fai unha chamada ao arquivo **/etc/pam.d/common-auth**

alumno@debianA:~$ exit #Saír da consola local de **alumno** á que acabamos de acceder para voltar á consola local de **usuario**.

D. En debianA acceder mediante ssh co usuario **alumno**. Que é o que acontece?

Ao usuario *alumno*, solicítaselle o contrasinal, xa que aínda que o servizo **/etc/pam.d/sshd** tamén fai unha chamada ao arquivo **/etc/pam.d/common-auth** resulta que existe unha entrada no arquivo de configuración **/etc/ssh/sshd_config** que evita o acceso con contrasinais en branco:

**PermitEmptyPasswords no**

E ademais aínda que activemos esta entrada (PermitEmptyPasswords yes), como estamos a empregar **nullok_secure** e non **nullok**, debemos engadir a consola **ssh** en **/etc/securetty**

```
# echo 'ssh' >> /etc/securetty
```

E. Executar:

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo).

root@debianA:~# sed -i "s/nullok_secure//" /etc/pam.d/common-auth #Eliminar o parámetro *nullok_secure* na regra existente ao módulo pam_unix en /etc/pam.d/common-auth

F. En debianA voltar a realizar o procedemento anterior, é dicir, voltar a acceder co usuario *alumno* dende tty1, su e ssh. Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login cambiando na regra de pam_unix existente en /etc/pam.d/common-auth (eliminando nullok_secure) o usuario *alumno* non pode iniciar sesión xa que non se permite iniciar sesión con contrasinais baleiros (en branco).

G. En debianA voltar a restaurar a regra modificada en pam_unix no arquivo /etc/pam.d/common-auth:

root@debianA:~# sed -i "s|pam_unix|pam_unix.so nullok_secure|" /etc/pam.d/common-auth #Modificar o ficheiro /etc/pam.d/common-auth para engadir na regra do módulo pam_unix o parámetro *nullok_secure*

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$

## @include common-account → /etc/pam.d/common-account

Imos verificar o funcionamento da **inclusión** do ficheiro **/etc/pam.d/common-account** para o **servizo login** referente ao módulo **pam_unix**, é dicir, imos verificar a regra que se inclúe para pam_unix dende o ficheiro /etc/pam.d/common-account no servizo login.

**common-account** é un arquivo existente en /etc/pam.d que incorpora regras de tipo account ao servizo que o chama, neste caso o servizo login.
**Só está asociado ao tipo account.**

A. Executar:

usuario@debianA:~$ grep common-account /etc/pam.d/login #Buscar o patrón common-account no ficheiro /etc/pam.d/login

```
@include common-account
```

usuario@debianA:~$ cat /etc/pam.d/common-account | sed '/^$/d' | grep -v '#' #Amosar do contido do ficheiro /etc/pam.d/common-account soamente as regras, evitando liñas en branco e comentarios.

```
account [success=1 new_authtok_reqd=done default=ignore] pam_unix.so
account requisite pam_deny.so
account required pam_permit.so
```

- A primeira regra refírese ao módulo pam_unix.
- A segunda regra refírese ao módulo pam_deny.
- A terceira regra refírese ao módulo pam_permit.

usuario@debianA:~$ grep pam_unix /etc/pam.d/common-account #Buscar o patrón pam_unix no ficheiro /etc/pam.d/common-account

```
account [success=1 new_authtok_reqd=done default=ignore] pam_unix.so
```

O tipo account realiza a tarefa de establecer o estado da conta e contrasinal do usuario baseado nos seguintes elementos de /etc/shadow: expire(oitavo campo), last_change(terceiro campo), max_change(quinto campo), min_change(cuarto campo), warn_change(sexto campo). No caso deste último, pode ofrecer consellos ao usuario sobre o cambio do seu contrasinal ou, a través da devolución de PAM_AUTHTOKEN_REQD, demorar a prestación do servizo ao usuario ata que estableza un novo contrasinal. As entradas enumeradas anteriormente están documentadas na páxina manual de shadow(5). Se o rexistro do usuario non conten unha ou máis destas entradas, a correspondente comprobación de shadow non é realizada.

B. Executar:

usuario@debianA:~$ man shadow #Ver ás páxinas de manual referente ao ficheiro de configuración /etc/shadow

usuario@debianA:~$ su - -c "grep usuario /etc/shadow" #Executar coma root o comando *grep usuario /etc/shadow*, é dicir, amosar a liña do ficheiro que ser refire ao usuario de nome *usuario*

```
usuario:$6$4IFjgmY/8C1.nOMQ$n2MJge3UTmxK5Mz1OJZa2vrZH9/syEupPQwAlSulQvruueEN89f4p/c3yfd
wq4awXt6wmidvW7SRJO.d8VomJ0:18659:0:99999:7:::
```

Cada liña do ficheiro **/etc/shadow** está formada por **9** campos separados pola caracter dous puntos ':', tal que:

- **Primeiro campo →** identifica o nome do usuario, o que se emprega para facer login.
- **Segundo campo →** identifica o contrasinal cifrado. Se este campo:
  1. Posúe soamente o valor ! ou * o usuario estará bloqueado e non poderá iniciar unha sesión do sistema.
  2. Está en branco, o usuario poder iniciar sesión sen contrasinal.
  3. Comeza por ! e a continuación posúe o contrasinal cifrado indica que o usuario está bloqueado e non pode iniciar sesión no sistema.
- **Terceiro campo →** identifica a data do último cambio de contrasinal, expresado como o número de días dende o 1 de xaneiro de 1970.
- **Cuarto campo →** identifica o número de días que o usuario ten que agardar antes que poida cambiar o contrasinal de novo. Se toma o valor 0 non existe esta condición.
- **Quinto campo →** identifica o número de días despois dos cales o usuario terá que cambiar o seu contrasinal. Se este campo:
  1. É un campo baleiro significa que non hai idade máxima do contrasinal, nin período de advertencia do contrasinal, e sen período de inactividade do contrasinal (ver máis abaixo).
  2. É inferior á idade mínima do contrasinal(cuarto campo), o usuario non pode cambiar o seu contrasinal.
- **Sexto campo →** identifica o número de días antes de que caduque un contrasinal (consulte a idade máxima do contrasinal arriba (quinto campo)) durante o cal se debe advertir ao usuario. Un campo baleiro e o valor 0 significan que non hai período de aviso de contrasinal.
- **Sétimo campo →** identifica o número de días despois de que caducou un contrasinal (consulte a idade máxima do contrasinal máis arriba (quinto campo)) durante o cal aínda se debería aceptar o contrasinal (e o usuario debería actualizalo durante o seguinte inicio de sesión).
  Despois de caducar o contrasinal e transcorrido este período de caducidade, non hai inicio de sesión posible usando o contrasinal do usuario actual. O usuario debe poñerse en contacto co seu administrador.
  Un campo baleiro significa que non hai aplicación dun período de inactividade.
- **Oitavo campo →** identifica a data de caducidade da conta, expresada como o número de días desde o 1 de xaneiro de 1970. Teña en conta que a caducidade da conta é diferente da caducidade do contrasinal. No caso da caducidade dunha conta, o usuario non poderá iniciar sesión. En caso de caducidade do contrasinal, non se permite ao usuario iniciar sesión usando o seu contrasinal.
  Un campo baleiro significa que a conta nunca caducará.
  Non se debe empregar o valor 0 xa que se interpreta como unha conta sen caducidade ou como caducidade o 1 de xaneiro de 1970.
- **Noveno campo →** está reservado para un uso futuro.

Neste caso para o usuario *usuario* os campos indican:

- Primeiro campo: Login → usuario
- Segundo campo: Contrasinal cifrado mediante algoritmo SHA-512
- Terceiro campo: O último cambio produciuse 18569 días despois do 1 de xañeiro de 1970.
- Cuarto campo: Valor 0, co cal o usuario non ten que agardar ningún número de días antes de cambiar o contrasinal.
- Quinto campo: 99999 son o número de días despois dos cales o usuario estará obrigado a cambiar o contrasinal.
- Sexto campo: avisarase ao usuario 7 días antes que vaia a caducar o contrasinal.
- Sétimo campo: Campo baleiro, entón non existe periodo de inactividade.
- Oitavo campo: Campo baleiro, a conta non caduca.
- Noveno campo: Reservado para un futuro.

Mediante o comando **chage** podemos de forma sinxela amosar esa información relativa a datas:

usuario@debianA:~$ chage -l alumno #Amosar a información da idade da conta.

```
Último cambio de contrasinal                                        : Feb 01, 2021
O contrasinal caduca                                                : nunca
Contrasinal inactivo                                                : nunca
A conta caduca                                                      : nunca
Número mínimo de días entre cambios de contrasinal                  : 0
Número máximo de días entre cambios de contrasinal                  : 99999
Número de días de aviso antes de que caduque o contrasinal          : 7
```

C. Executar:

usuario@debianA:~$ su - -c 'chage -E 2021-02-01 usuario' #Executar coma root o comando chage para pór como data de caducidade da conta (expire) o 1 de febreiro de 2021

usuario@debianA:~$ chage -l alumno #Amosar a información da idade da conta.

```
Último cambio de contrasinal                                        : Feb 01, 2021
O contrasinal caduca                                                : nunca
Contrasinal inactivo                                                : nunca
A conta caduca                                                      : Feb 01, 2021
Número mínimo de días entre cambios de contrasinal                  : 0
Número máximo de días entre cambios de contrasinal                  : 99999
Número de días de aviso antes de que caduque o contrasinal          : 7
```

usuario@debianA:~$ date +%F #Amosar a data en formato YYYY-mm-dd (ano-mes-día)

```
2020-02-02
```

D. En debianA acceder á consola tty1 mediante o usuario **usuario**. Que é o que acontece?

O usuario *usuario,* non pode iniciar sesión porque a conta caducou

E. En debianA acceder como *usuario* mediante o comando **su**. Que é o que acontece?

usuario@debianA:~$ su - usuario #Acceder como usuario *usuario*. Este usuario non poder iniciar sesión porque a conta caducou, xa que o servizo **/etc/pam.d/su** tamén fai unha chamada ao arquivo **/etc/pam.d/common-account**

```
$ grep common-account /etc/pam.d/su
@include common-account
```

F. En debianA acceder mediante ssh co usuario **usuario**. Que é o que acontece?

O usuario *usuario*, non pode iniciar sesión porque a conta caducou, xa que o servizo **/etc/pam.d/sshd** tamén fai unha chamada ao arquivo **/etc/pam.d/common-account**

```
$ grep common-account /etc/pam.d/sshd
@include common-account
```

G. En debianA desactivar a data de caducidade da conta *usuario*:

usuario@debianA:~$ su - -c 'chage -E -1 usuario' #Executar coma root o comando chage para desactivar a data de caducidade da conta (expire)

usuario@debianA:~$ chage -l alumno #Amosar a información da idade da conta.

```
Último cambio de contrasinal                                        : Feb 01, 2021
O contrasinal caduca                                                : nunca
Contrasinal inactivo                                                : nunca
A conta caduca                                                      : nunca
Número mínimo de días entre cambios de contrasinal                  : 0
Número máximo de días entre cambios de contrasinal                  : 99999
Número de días de aviso antes de que caduque o contrasinal          : 7
```

## @include common-session → /etc/pam.d/common-session

Imos verificar o funcionamento da **inclusión** do ficheiro **/etc/pam.d/common-session** para o **servizo login** referente ao módulo **pam_unix**, é dicir, imos verificar a regra que se inclúe para pam_unix dende o ficheiro /etc/pam.d/common-session no servizo login.

**common-session** é un arquivo existente en /etc/pam.d que incorpora regras de tipo session ao servizo que o chama, neste caso o servizo login.
**Só está asociado ao tipo session.**

A. Executar:

usuario@debianA:~$ grep common-session /etc/pam.d/login #Buscar o patrón common-session no ficheiro /etc/pam.d/login

```
@include common-session
```

usuario@debianA:~$ cat /etc/pam.d/common-session | sed '/^$/d' | grep -v '#' #Amosar do contido do ficheiro /etc/pam.d/common-session soamente as regras, evitando liñas en branco e comentarios.

```
session [default=1]                     pam_permit.so
session requisite                       pam_deny.so
session required                        pam_permit.so
session required        pam_unix.so
session optional        pam_systemd.so
```

- A primeira e terceira regra referense ao módulo pam_permit.
- A segunda regra referese ao módulo pam_deny.
- A cuarta regra referese ao módulo pam_unix.
- A quinta regra referese ao módulo pam_systemd.

usuario@debianA:~$ grep pam_unix /etc/pam.d/common-session #Buscar o patrón pam_unix no ficheiro /etc/pam.d/common-session

```
session required pam_unix.so
```

O tipo session deste módulo rexistra cando un usuario inicia sesión ou sae do sistema.
- quiet → Desactivar as mensaxes referentes a session, é dicir, non se rexistran mensaxes a través de syslog(3).

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo).

root@debianA:~# userdel -r alumno #Eliminar o usuario alumno e coa opción -r eliminar tamén o seu cartafol de usuario (/home/alumno) e se existe o seu cartafol de correo (var/mail/alumno ou /var/spool/mail/alumno; depende da variable MAIL_DIR definida no ficheiro /etc/login.defs)

root@debianA:~# dpkg -l | grep whois; [ $(echo $?) -eq '1' ] && apt update && apt -y install whois
#Verificar se o paquete whois está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase, é dicir, instálase o paquete que integra o comando mkpasswd.

root@debianA:~# useradd -m -d /home/alumno -s /bin/bash -p $(mkpasswd -m sha-512 '123') alumno
#Crear o usuario *alumno* co comando *useradd*, onde:
-d /home/alumno → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/alumno
-m → Copia na casa do usuario o que exista no cartafol /etc/skel
-s /bin/bash → Establece como shell de traballo para o usuario a shell bash
-p '123' → Establece como contrasinal *123* cifrado mediante SHA-512.
alumno → Establece como nome de autenticación de usuario o nome alumno

root@debianA:~# id alumno #Amosa información de usuario e grupo do usuario de nome alumno

```
uid=1021(alumno) gid=1021(alumno) grupos=1021(alumno)
```

root@debianA:~# tail -n 2 -f /var/log/auth.log #Deixar aberto o ficheiro /var/log/auth.log para lectura, comenzando a ver polas 2 últimas liñas. Moi empregado na revisión de logs.

B. En debianA acceder á consola tty1 mediante o usuario **alumno**. Que é o que acontece?

O usuario *alumno*, inicia sesión sen problema co contrasinal *123* e observanse rexistros no ficheiro */var/log/auth.log*, é dicir ao acceder co usuario *alumno* no inicio de sesión envíanse mensaxes rexistrados con syslog ao ficheiro */var/log/auth.log*:

```
...
Feb  1 17:31:00 debianA login[1121]: pam_unix(login:session): session opened for user alumno by LOGIN(uid=0)
```

E que acontece se pechamos a sesión co usuario *alumno*?:

O usuario *alumno*, pecha a sesión e observanse rexistros no ficheiro */var/log/auth.log*, é dicir ao pechar a sesión do usuario *alumno* seguen enviándose mensaxes rexistrados con syslog no ficheiro */var/log/auth.log*:

```
...
Feb  1 17:31:15 debianA login[1121]: pam_unix(login:session): session closed for user alumno
```

C. En debianA acceder como alumno mediante o comando **su**. Que é o que acontece?

usuario@debianA:~$ su - alumno #Acceder como usuario *alumno*. Este usuario inicia sesión sen problema co contrasinal *123*

O usuario *alumno*, inicia sesión sen problema co contrasinal *123* e observanse rexistros no ficheiro */var/log/auth.log*, é dicir ao acceder co usuario *alumno* no inicio de sesión envíanse mensaxes rexistrados con syslog ao ficheiro */var/log/auth.log*:

```
...
Feb  1 17:31:00 debianA login[1121]: pam_unix(login:session): session opened for user alumno by LOGIN(uid=0)
Feb  1 17:33:51 debianA su: (to alumno) usuario on pts/2
Feb  1 17:33:51 debianA su: pam_unix(su-l:session): session opened for user alumno by (uid=1000)
```

E que acontece se pechamos a sesión co usuario *alumno*?:

alumno@debianA:~$ exit #Saír da consola local de **alumno** á que acabamos de acceder para voltar á consola local de **usuario**.

O usuario *alumno*, pecha a sesión e observanse rexistros no ficheiro */var/log/auth.log*, é dicir ao pechar a sesión do usuario *alumno* seguen enviándose mensaxes rexistrados con syslog no ficheiro */var/log/auth.log*:

```
...
Feb  1 17:34:51 debianA su: pam_unix(su-l:session): session closed for user alumno
```

D. En debianA acceder mediante ssh co usuario **alumno**. Que é o que acontece?

Ao usuario *alumno*, rexístranselle as mensaxes de acceso, xa que o servizo **/etc/pam.d/sshd** tamén fai unha chamada ao arquivo **/etc/pam.d/common-session**, co cal accede sen problema co contrasinal *123*

```
$ grep common-session /etc/pam.d/sshd
@include common-session
```

Así, o usuario *alumno*, inicia sesión sen problema co contrasinal *123* e observanse rexistros no ficheiro */var/log/auth.log*, é dicir ao acceder co usuario *alumno* no inicio de sesión envíanse mensaxes rexistrados con syslog ao ficheiro */var/log/auth.log*:

```
...
Feb  1 17:52:54 debianA sshd[9337]: Accepted password for alumno from 192.168.120.100 port 40824 ssh2
Feb  1 17:52:54 debianA sshd[9337]: pam_unix(sshd:session): session opened for user alumno by (uid=0)
```

E que acontece se pechamos a sesión co usuario *alumno*?:

Pois acontece o mesmo, o usuario *alumno*, pecha a sesión e observanse rexistros no ficheiro */var/log/auth.log*, é dicir ao pechar a sesión do usuario *alumno* seguen enviándose mensaxes rexistrados con syslog no ficheiro */var/log/auth.log*:

```
...
Feb  1 17:57:15 debianA sshd[9343]: Received disconnect from 192.168.120.100 port 40824:11: disconnected by user
Feb  1 17:57:15 debianA sshd[9343]: Disconnected from user alumno 192.168.120.100 port 40824
Feb  1 17:57:15 debianA sshd[9337]: pam_unix(sshd:session): session closed for user alumno
```

E. Executar:

usuario@debianA:~$ ^C #Abortar execución do comando anterior, é dicir, abortar a lectura do ficheiro /var/log/auth.log, enviando o sinal 2 (SIGNINT 2)(kill -l) ao sistema.

root@debianA:~# sed -i "s/pam_unix.so/pam_unix.so quiet/" /etc/pam.d/common-session #Engadir o parámetro *quiet* na regra existente ao módulo pam_unix en /etc/pam.d/common-session

root@debianA:~# tail -n 2 -f /var/log/auth.log #Deixar aberto o ficheiro /var/log/auth.log para lectura, comenzando a ver polas 2 últimas liñas. Moi empregado na revisión de logs.

F. En debianA acceder de novo co usuario **alumno** na consola tty1, mediante **su** e a través dunha sesión **ssh**. Que é o que acontece?

Pois, o usuario *alumno*, inicia sesión sen problema co contrasinal *123* e observanse que non se amosan rexistros no ficheiro */var/log/auth.log*, é dicir ao acceder co usuario *alumno* no inicio de sesión xa non se envían mensaxes con syslog ao ficheiro */var/log/syslog*:

E que acontece se pechamos a sesión co usuario *alumno*, con calquera dos 3 accesos anteriores?:

Pois o mesmo, o usuario *alumno*, pecha a sesión e non se observan rexistros no ficheiro */var/log/auth.log*, é dicir ao pechar a sesión do usuario *alumno* non se envián mensaxes con syslog no ficheiro */var/log/auth.log*:

G. En debianA voltar a restaurar a regra modificada en pam_unix no arquivo /etc/pam.d/common-session:

root@debianA:~# sed -i "s/pam_unix.so quiet/pam_unix.so/" /etc/pam.d/common-session #Modificar o ficheiro /etc/pam.d/common-session para restaurar a regra do módulo pam_unix

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$

## @include common-password → /etc/pam.d/common-password

Imos verificar o funcionamento da **inclusión** do ficheiro **/etc/pam.d/common-password** para o **servizo login** referente ao módulo **pam_unix**, é dicir, imos verificar a regra que se inclúe para pam_unix dende o ficheiro /etc/pam.d/common-password no servizo login.

**common-password** é un arquivo existente en /etc/pam.d que incorpora regras de tipo password ao servizo que o chama, neste caso o servizo login.
**Só está asociado ao tipo password.**

A. Executar:

usuario@debianA:~$ grep common-password /etc/pam.d/login #Buscar o patrón common-password no ficheiro /etc/pam.d/login

```
@include common-password
```

usuario@debianA:~$ cat /etc/pam.d/common-password | sed '/^$/d' | grep -v '#' #Amosar do contido do ficheiro /etc/pam.d/common-password soamente as regras, evitando liñas en branco e comentarios.

```
password [success=1 default=ignore] pam_unix.so obscure sha512
password requisite pam_deny.so
password required pam_permit.so
password optional pam_gnome_keyring.so
```

- A primeira regra refírese ao módulo pam_unix.
- A segunda regra refírese ao módulo pam_deny.
- A terceira regra refírese ao módulo pam_permit.
- A cuarta regra refírese ao módulo pam_gnome_keyring.

usuario@debianA:~$ grep pam_unix /etc/pam.d/common-password #Buscar o patrón pam_unix no ficheiro /etc/pam.d/common-password

```
password [success=1 default=ignore] pam_unix.so obscure sha512
```

O tipo password realiza a tarefa de actualizar o contrasinal do usuario. O hash de cifrado predeterminado tómase da variable ENCRYPT_METHOD de /etc/login.defs

- obscure → Activar algunhas comprobacións adicionais sobre a forza do contrasinal. Estes controis baséanse sobre o "obscure" comprobacións no paquete orixinal shadow (ver man pam_unix). Esta opción substitúe á antiga opción OBSCURE_CHECKS_ENAB en /etc/login.defs.
- sha512 → Cando un usuario cambia o contrasinal farase un cifrado SHA512 para o novo contrasinal. Se a función crypt(3) non coñece o algoritmo SHA512 farase o cifrado mediante o algoritmo MD5.
- remember = n → Os últimos n contrasinais para cada usuario gárdanse en /etc /security/opasswd para forzar o cambio de contrasinal revisando o historial e evitar que o usuario alterne tamén entre o mesmo contrasinal con frecuencia. O algoritmo de hash de contrasinal MD5 úsase para almacenar os contrasinais antigos. No canto desta opción debería usarse o módulo pam_pwhistory.
- minlen = n → Establecer unha lonxitude mínima de contrasinal de n caracteres. O valor predeterminado é 6. O máximo para os contrasinais cifrados mediante DES son 8 caracteres.

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo).

root@debianA:~# userdel -r alumno #Eliminar o usuario alumno e coa opción -r eliminar tamén o seu cartafol de usuario (/home/alumno) e se existe o seu cartafol de correo (var/mail/alumno ou /var/spool/mail/alumno; depende da variable MAIL_DIR definida no ficheiro /etc/login.defs)

root@debianA:~# dpkg -l | grep whois; [ $(echo $?) -eq '1' ] && apt update && apt -y install whois #Verificar se o paquete whois está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase, é dicir, instálase o paquete que integra o comando mkpasswd.

root@debianA:~# useradd -m -d /home/alumno -s /bin/bash -p $(mkpasswd -m sha-512 '123') alumno #Crear o usuario *alumno* co comando *useradd*, onde:
-d /home/alumno → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/alumno
-m → Copia na casa do usuario o que exista no cartafol /etc/skel
-s /bin/bash → Establece como shell de traballo para o usuario a shell bash
-p '123' → Establece como contrasinal *123* cifrado mediante SHA-512.
alumno → Establece como nome de autenticación de usuario o nome alumno

root@debianA:~# exit #Saír da consola local sudo á que acabamos de acceder para voltar á consola local de **usuario**.

B. En debianA acceder á consola tty1 mediante o usuario **alumno**. Que é o que acontece?

O usuario *alumno*, inicia sesión sen problema co contrasinal *123*

C. En debianA acceder como alumno mediante o comando **su**. Que é o que acontece?

usuario@debianA:~$ su - alumno #Acceder como usuario *alumno*. Este usuario inicia sesión sen problema co contrasinal *123*

alumno@debianA:~$ exit #Saír da consola local de **alumno** á que acabamos de acceder para voltar á consola local de **usuario**.

D. En debianA acceder mediante ssh co usuario **alumno**. Que é o que acontece?

Ao usuario *alumno*, solicítaselle o contrasinal, xa que que o servizo **/etc/pam.d/sshd** tamén fai unha chamada ao arquivo **/etc/pam.d/common-password**, co cal accede sen problema co contrasinal *123*

```
$ grep common-password /etc/pam.d/sshd
@include common-password
```

E. Executar:

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo).

root@debianA:~# sed -i "s/pam_unix.so obscure sha512/pam_unix.so obscure sha512 minlen=12/" /etc/pam.d/common-password #Engadir o parámetro *minlen* co valor *12* na regra existente ao módulo pam_unix en /etc/pam.d/common-password

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$ su - alumno #Acceder como usuario *alumno*. Este usuario inicia sesión sen problema co contrasinal *123*

alumno@debianA:~$ passwd alumno || (echo -e '123\n123456789zx\n123456789zx' | passwd) #Cambiar o contrasinal do usuario alumno. Por como contrasinal **123456789zx**

Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login cambiando na regra de pam_unix existente en /etc/pam.d/common-password (engadindo minlen=12) o usuario *alumno* non pode cambiar o contrasinal a non ser que posúa como mínimo 12 caracteres, e resulta que *abc123456789zx* son 11 caracteres.

```
alumno@debianA:~$ echo -e '123\n123456789zx\n123456789zx' | passwd
Changing password for alumno.
Current password: New password: Retype new password: You must choose a longer password
New password: Password change aborted.
passwd: Authentication token manipulation error
passwd: password unchanged
```

F. Executar:

alumno@debianA:~$ passwd alumno || (echo -e '123\n123456789zxc\n123456789zxc' | passwd) #Cambiar o contrasinal do usuario alumno. Por como contrasinal **123456789zxc**

Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login cambiando na regra de pam_unix existente en /etc/pam.d/common-password (engadindo minlen=12) o usuario *alumno* si pode cambiar o contrasinal xa que *abc123456789zxc* son 12 caracteres.

```
alumno@debianA:~$ echo -e '123\n123456789zx\n123456789zx' | passwd
echo -e '123\n123456789zxc\n123456789zxc' | passwd
Changing password for alumno.
Current password: New password: Retype new password: passwd: o contrasinal actualizouse con éxito
```

alumno@debianA:~$ exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

G. Executar:

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo).

root@debianA:~# sed -i "s/pam_unix.so obscure sha512 minlen=12/pam_unix.so obscure sha512 minlen=12 remember=3/" /etc/pam.d/common-password #Engadir o parámetro *remember* co valor *3* na regra existente ao módulo pam_unix en /etc/pam.d/common-password

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$ su - alumno #Acceder como usuario *alumno*. Este usuario inicia sesión sen problema co contrasinal *123456789zxc*

alumno@debianA:~$ ls -l /etc/security/opasswd #Listar de forma extendida o ficheiro /etc/security/opasswd, o cal contén os últimos contrasinais gardados (ver parámetro *remember* do módulo *pam_unix*)

```
-rw------- 1 root root 0 Nov 16 2019 /etc/security/opasswd
```

Podemos comprobar que non existe contido no ficheiro /etc/security/opasswd

alumno@debianA:~$ PASS=$(mktemp -u XXXXXXXXXXXXXXXX) && echo -e "123456789zxc\n${PASS}\n${PASS}" | passwd #Cambiar o contrasinal do usuario alumno. Por como contrasinal o valor contido dentro da variable **PASS**. A variable **PASS** contén un valor aleatorio de **16** caracteres.

alumno@debianA:~$ ls -l /etc/security/opasswd #Listar de novo de forma extendida o ficheiro /etc/security/opasswd.

```
--rw------- 1 root root 49 Feb 1 19:39 /etc/security/opasswd
```

Podemos comprobar que agora SI existe contido no ficheiro /etc/security/opasswd

alumno@debianA:~$ su - -c "cat /etc/security/opasswd" #Executar coma root o comando *cat /etc/security/opasswd*, é dicir, amosar o contido do ficheiro */etc/security/opasswd*

```
alumno:1021:1:$1$YJAPwpRU$4glGepbclE94Zjpte1h7w0
```

Temos gardado o último contrasinal.

alumno@debianA:~$ PASSOLD=${PASS} && PASS=$(mktemp -u XXXXXXXXXXXXXXXX) && echo -e "${PASSOLD}\n${PASS}\n${PASS}" | passwd #Cambiar o contrasinal do usuario alumno. Por como contrasinal o valor contido dentro da variable **PASS**. A variable **PASS** contén un valor aleatorio de **16** caracteres.

alumno@debianA:~$ su - -c "cat /etc/security/opasswd" #Executar coma root o comando *cat /etc/security/opasswd*, é dicir, amosar o contido do ficheiro */etc/security/opasswd*

```
alumno:1021:2:$1$YJAPwpRU$4glGepbclE94Zjpte1h7w0,$1$R06ypTw/$BYkAwkUxolGj4inB/Wnpd1
```

Temos gardados os 2 últimos contrasinais. Os contrasinais están separados polo caracter coma ','

alumno@debianA:~$ for i in $(seq 1 10); do PASSOLD=${PASS} && PASS=$(mktemp -u XXXXXXXXXXXXXXXX) && echo -e "${PASSOLD}\n${PASS}\n${PASS}" | passwd;done #Equivale a executar o comando anterior 10 veces máis.

Que é o que acontece?

Pois agora, debido ao cambio que efectuamos no servizo login cambiando na regra de pam_unix existente en /etc/pam.d/common-password (engadindo remember=3) soamente gárdanse os últimos 3 contrasinais

alumno@debianA:~$ su - -c "cat /etc/security/opasswd" #Executar coma root o comando *cat /etc/security/opasswd*, é dicir, amosar o contido do ficheiro */etc/security/opasswd*

```
alumno:1021:3:$1$d9/85RLS$7OU2mlTkmbgOFqUlgHOQ.1,$1$9/LPJFrq$CFvY/XJfpfzWQ79AS53XI0,
$1$iDa2Wo.d$XK3FKz7ytOLnRw79fcban/
```

H. En debianA voltar a restaurar a regra modificada en pam_unix no arquivo /etc/pam.d/common-password:

alumno@debianA:~$ exit #Saír da consola local de **alumno** á que acabamos de acceder para voltar á consola local de **usuario**.

usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

root@debianA:~# sed -i "s/pam_unix.so obscure sha512 minlen=12 remember=3/pam_unix.so obscure sha512/" /etc/pam.d/common-password #Modificar o ficheiro /etc/pam.d/common-password para restaurar a regra do módulo pam_unix

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **usuario**.

usuario@debianA:~$

---

Ricardo Feijoo Costa

This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**