

PAM

Introducció

Existeixen moltes maneres d'autenticar usuaris: consultant els logins/passwords locals dins la parella d'arxius `/etc/passwd` i `/etc/shadow`, o bé consultant-los en un servidor LDAP, o en una base de dades relacional convencional, o bé a través de tarjetes hardware, o via reconeixement d'empremtes dactilars, etc.

Això pels desenvolupadors és un problema, perquè per a què els seus programes (com ara *login*, *gdm*, *sudo*, *su*, *ftp*, *nfs*, *ssh*,...) suportin tots aquests diferents mètodes d'autenticació, han de programar explícitament cada mètode per separat (a més de què si aparegués un altre mètode nou, s'hauria de recompilar el programa per a què el suportés -en el cas que fos possible-, o bé reescriure de nou el programa sencer.

PAM és un conjunt de llibreries (bàsicament `libpam.so` però no només) que fan d'intermediàries entre els mètodes d'autenticació i els programes que els fan servir, permetent la possibilitat de desenvolupar programes de forma independent a l'esquema d'autenticació a utilitzar. La idea és que un programa, gràcies a haver estat desenvolupat mitjançant les llibreries PAM, pugui fer servir un/s determinat/s mòdul/s binari/s (o un/s altre/s, segons el que interressi) que s'encarregui/n de la “feina bruta” d'autenticació. El/s mòdul/s concret/s a utilitzar pel programa en un determinat moment l'escollirà l'administrador del sistema, el qual podrà canviar de mòdul (de `/etc/shadow` a LDAP per exemple), si així ho desitja, sense que el programa corresponent notés la diferència.

Resumint: PAM facilita la vida als desenvolupadors perquè en comptes d'escriure una complexa capa d'autenticació pels seus programes simplement han d'escriure un “hook” a PAM, i també facilita la vida als administradors perquè poden configurar l'autenticació dels programes que la necessiten d'una forma centralitzada i comú sense necessitat de recordar cada model separat per cada programa.

Els fitxers relacionats amb PAM són:

- 1.-Els mòduls, ubicats a `/lib/security` (a Fedora) o a `/lib/x64_64-linux-gnu/security` (a Ubuntu). Molts s'instal·len "de sèrie" juntament amb les llibreries PAM però també poden ser "de tercers". En qualsevol cas, cada mòdul implementa un mètode d'autenticació diferent i bàsicament pot retornar dos valors: "èxit" (`PAM_SUCCESS`) o "fallada" (`PAM_PERM_DENIED` o altres equivalents).
- 2.-Els fitxers de configuració dels mòduls anteriors (si és que en tenen), ubicats a `/etc/security`.
- 3.-Els fitxers de configuració proporcionats per cada aplicació, encarregats d'establir quins mètodes d'autenticació utilitzarà aquesta. Són fitxers de text que tenen com a nom el nom de l'aplicació associada i estan ubicats dins del directori `/etc/pam.d`.

Mòduls

Respecte el punt 1. , dir que alguns dels mòduls oficials més importants són els següents (consulteu <http://www.linux-pam.org/Linux-PAM-html/sag-module-reference.html> o bé la pàgina del manual anomenada "pam_xxx" per saber-ne més):

NOTA: A Ubuntu aquests mòduls -i molts altres- es troben dins del paquet "libpam-modules" (instal·lat per defecte al sistema). A Fedora es troben al paquet "pam" (també instal·lat per defecte).

pam_unix.so	Realitza l'autenticació mitjançant el mètode tradicional ("/etc/passwd" i "/etc/shadow"). Si es vol realitzar l'autenticació contra usuaris/contrasenyes guardats a bases de dades BerkeleyDB es pot fer servir llavors el mòdul pam_userdb.so
pam_deny.so	Sempre retorna fallada. Generalment s'executa si cap altre mòdul ha tingut èxit
pam_permit.so	Sempre retorna èxit (per tant, no hi ha autenticació, usant-se llavors l'usuari <i>nobody</i>)
pam_cracklib.so	Només és un mòdul de tipus "password" (veure més avall). S'assegura que la contrasenya emprada tingui un nivell de seguretat suficient segons les regles establertes (longitud, variabilitat, etc). Hi ha un altre mòdul similar anomenat pam_pwquality.so una mica més modern (però no ve dins del paquet "oficial"). Un altre encara és pam_passwdqc.so (programat per la gent del John The Ripper).
pam_pwhistory.so	Comprova que, en canviar una contrasenya, no hagi sigut usada abans algun cop
pam_succeed_if.so	Dóna per vàlida una autenticació (o no) segons el valor d'una o més característiques especificades (uid, nom, ruta de carpeta personal, shell per defecte, grup al que pertany...) que tingui el compte d'usuari autenticat
pam_access.so	Dóna per vàlida una autenticació (o no) segons si un/s determinat/s usuari/s o grup/s accedeixen des d'un/s determinat/s terminal/s local/s o bé determinada/es màquina/es remota/es -comprovant el seu nom o la seva IP. Un mòdul que implementa una funcionalitat similar (encara que es configura diferent) és pam_listfile.so
pam_time.so	Dóna per vàlida una autenticació (o no) segons el moment (hora, dia...) en que s'ha fet
pam_faillock.so	Conta el nº seguits d'intents d'autenticació fallits durant un cert període de temps per tal de bloquejar l'accés. Antigament hi havia un altre mòdul similar anomenat "pam_tally2" que ha sigut retirat per obsolet (https://src.fedoraproject.org/rpms/pam/blob/f30/f/pam.spec#_398)
pam_lastlog.so	Bloqueja el logueig d'un usuari depenent de la data del seu darrer login
pam_faildelay.so	Estableix el temps d'espera mínim entre introduccions seguides de contrasenyes errònies (menys del qual sempre retornarà "fallada" sigui quina sigui la contr. escrita)
pam_limits.so	Només és un mòdul de sessió (veure més avall). Assigna determinades limitacions als usuaris (nº màxim de processos permesos, nº màxim de fitxers oberts, etc), segons la configuració indicada a l'arxiu "/etc/security/limits.conf". Actualment, no obstant, aquesta funcionalitat també ve incorporada dins de Systemd (veure les directives "LimitXXX" a <i>man systemd.exec</i>) així que tots els serveis gestionats per Systemd no fan servir aquest mòdul per res.
pam_securetty.so	Excepte en els terminals llistats a l'arxiu /etc/securetty, prohibeix l'autenticació al root. Si aquest arxiu no existeix, es permetrà l'autenticació de root des de qualsevol terminal
pam_nologin.so	Si existeix l'arxiu /etc/nologin (el contingut del qual es mostrarà a pantalla), prohibeix l'autenticació a tothom excepte root
pam_wheel.so	Només permet actuar com a root (mitjançant su/sudo) als usuaris del grup "wheel"
pam_rootok.so	Permet que l'usuari "root" es pugui autenticar de forma directa (és a dir, sense que se li demani cap tipus d'autenticació)

pam_env.so	Permet fer servir variables d'entorn definides dins de /etc/security/pam_env.conf (o a l'arxiu indicat amb "envfile=") i que poden ser utilitzades per altres mòduls posteriors o també pel sistema.
pam_echo.so pam_motd.so	Mostra missatges personalitzats a pantalla. Mostra un missatge (per defecte l'inclòs a /etc/motd) un cop l'usuari s'ha loguejat. Moltes vegades, no obstant, per fer això s'usa directament l'arxiu /etc/profile o similar
pam_exec.so	Executa una comanda externa
pam_mkhomedir.so	Crea la carpeta personal d'un usuari quan es logueja per primera vegada en el sistema. Útil per quan l'usuari no és local sinó que es troba en alguna BD remota o serv. LDAP
pam_systemd.so	Només és un mòdul de sessió (veure més avall). Dóna suport al servei systemd-login

També existeixen altres mòduls no oficials però molt interessants que poden instal·lar al nostre sistema com paquets estàndar (per trobar aquests i molts d'altres, podeu executar *apt search libpam -a Ubuntu-* o bé *dnf search pam -a Fedora-*):

pam_captcha.so	Mostra un text gràfic mitjançant <i>figlet</i> a mode de captcha. La seva pàgina oficial és https://github.com/jordansissel/pam_captcha Atenció: <u>No</u> està empaquetat a les distribucions més importants
pam_oath.so	Implementa sistema d'autenticació amb contrasenyes d'un sol ús (OTP). La seva pàgina oficial és http://www.nongnu.org/oath-toolkit
pam_google_authenticator.so	Utilitza els servidors de Google para realitzar el logueig en dos passos (amb contrasenyes d'un sol ús, les OTP). La seva pàgina oficial és https://github.com/google/google-authenticator-libpam
pam_usb.so DEPRECATED	Només permet l'inici de sessió si hi ha un llapis Usb enxufat a l'ordinador. La seva pàgina oficial és https://github.com/aluzzardi/pam_usb
pam_ldap.so	Realitza l'autenticació contra usuaris/contrasenyes guardats a un servidor Ldap. La seva pàgina oficial és http://www.padl.com/OSS/pam_ldap.html
pam_mariadb.so	Realitza l'autenticació contra usuaris/contrasenyes guardats a un servidor MariaDB. La seva pàgina oficial és https://mariadb.com/kb/en/pam-authentication-plugin .
pam_mysql.so	Realitza l'autenticació contra usuaris/contrasenyes guardats a un servidor MySQL. La seva pàgina oficial és https://github.com/NigelCunningham/pam-MySQL . D'altra banda, dir que existeix un mòdul PAM de MySQL oficial, però és comercial (https://dev.mysql.com/doc/refman/5.7/en/pam-pluggable-authentication.html)
pam_pgsqll.so	Realitza l'autenticació contra usuaris/contrasenyes guardats a un servidor PostgreSQL. La seva pàgina és https://github.com/pam-pgsqll/pam-pgsqll
pam_sqlite3.so	Realitza l'autenticació contra usuaris/contrasenyes guardats a un servidor SQLite3. La seva pàgina oficial és https://github.com/HormyAJP/pam_sqlite3
pam_abl.so	Bloqueja noms d'usuari/IPs remotes que intenten loguejar al sistema (similar en intenció a Fail2ban). La seva pàgina oficial és https://github.com/deksai/pam_abl

NOTA: En el cas de voler desenvolupar una aplicació pròpia (en C) que faci ús d'algun dels mòduls anteriors i se n'aprofita de la seva funcionalitat, un petit tutorial el podreu trobar aquí: <https://fedetask.com/writing-a-linux-pam-aware-application>. En el cas de voler desenvolupar un mòdul PAM pròpiament dit, un petit tutorial el podreu trobar aquí: <https://fedetask.com/write-linux-pam-module>. En qualsevol cas, la guia oficial del desenvolupador PAM es troba a http://www.linux-pam.org/Linux-PAM-html/Linux-PAM_ADG.html i http://www.linux-pam.org/Linux-PAM-html/Linux-PAM_MWG.html, respectivament.

Fitxers de configuració de les aplicacions

Al punt 3. ja hem dit que dins de /etc/pam.d trobem els fitxers de configuració que controlen el comportament de les aplicacions homònimes que utilitzen els mòduls PAM (*login, gdm, ssh, sudo,...*). Aquests fitxers consten de línies formades per tres camps (explicats a continuació), on cal tenir en compte que l'ordre de les línies és molt important (es llegeixen de dalt a baix):

1^r camp: Defineix el tipus d'acció que realitza el mòdul PAM. Bàsicament hi ha quatre tipus, tres dels quals sempre se solen executar amb el següent ordre: "auth", després "account" i després "session"; l'acció "password" s'executa sota demanda:

auth : autentica l'usuari. És a dir, comprova si és un usuari vàlid i reconegut pel sistema i que tingui una contrasenya (o el mètode d'autenticació emprat) igual a la proporcionada per l'aplicació.

account : autoritza l'usuari. És a dir, un cop autenticat, li dóna accés -o no- a certs recursos del sistema seguint les restriccions indicades: n^o màxim d'usuaris, localització de l'usuari, horari, expiració de la contrasenya, etc

password : acció activada en actualitzar contrasenyes

session : aplica accions sobre l'usuari ja autoritzat que estan relacionades amb l'inici i final de sessió (com muntar carpetes, utilitzar el correu del sistema, habilitar els logs, executar algun script, definir variables d'entorn, crear carpetes personals, etc)

2ⁿ camp: Indica com ha de reaccionar el mòdul davant l'èxit (valor retornat PAM_SUCCESS) o l'error (diferents valors possibles depenent del tipus d'error) en l'autenticació. En concret pot valer:

requisite : es necessita l'èxit d'aquest mòdul. Si falla no se segueix llegint (i es notifica 'error'); si s'obté èxit, es continuen llegint les següents línies "requisite" corresponents a la mateixa acció (valor 1^r camp). En conclusió: només hi ha èxit si tots els mòduls "requisite" de la mateixa acció han obtingut èxit

required : igual que amb *requisite*, es necessita l'èxit d'aquest mòdul. La diferència està en que tant si s'obté èxit com si falla, se segueix llegint la resta de línies "required" corresponents a la mateixa acció (valor 1^r camp) i no es notifica a l'aplicació el resultat final fins llegir l'última d'aquestes línies. Això es fa per seguretat, per a què un "hacker" no sàpiga quin mòdul ha fallat. En conclusió: només hi ha èxit si tots els mòduls "required" de la mateixa acció han obtingut èxit

sufficient : si el seu resultat és èxit (i els required/requisite anteriors també), no se segueix llegint. Si falla, sí se segueix llegint la resta de línies corresponents a la mateixa acció (valor 1^r camp). En conclusió: hi ha èxit simplement amb què un sol mòdul "sufficient" d'una determinada acció hagi obtingut èxit.

optional : el seu valor de retorn (èxit o fallada) no influeix en l'èxit per a què l'autenticació/autorització/... s'efectuï. Simplement s'executa el mòdul per realitzar una determinada tasca i ja està. Aquest valor se sol fer servir amb mòduls de tipus "session".

include : inclou el contingut d'un altre fitxer, el nom del qual s'ha d'indicar al 3^r camp (i la seva ubicació serà igualment /etc/pam.d). Un altre valor similar seria substack

[opcio1=valor2 opcio2=valor2] : permet un control més granular de les condicions d'èxit i fracàs. Les "opcions" corresponen als diferents valors de retorn del mòdul en qüestió i els "valors" representen l'acció que s'executarà en detectar aquest retorn concret. Per exemple, d'entre les "opcions" podem tenir:

success : el mòdul retorna PAM_SUCCESS
ignore : el mòdul diu que s'ignori qualsevol línia "account" posterior
... : multitud d'errors diferents retornats pel mòdul (veure <http://www.linux-pam.org/Linux-PAM-html/sag-configuration-file.html>)
default : qualsevol valor de retorn no indicat explícitament mitjançant una "opció"

I d'entre els "valors" podem tenir:

Un nº : indica el número de línies dins de l'arxiu que s'hauran de saltar per continuar llegint la configuració a la línia just després del salt
bad : indica que el mòdul retornarà "fallada" a l'aplicació i es continuarà llegint
die : indica que el mòdul retornarà "fallada" a l'aplicació i es deixarà de llegir
ok : indica que el mòdul retornarà "èxit" a l'aplicació i es continuarà llegint
done : indica que el mòdul retornarà "èxit" a l'aplicació i es deixarà de llegir
ignore : indica que l'"opció" corresponent no es té en compte (com si no hagués passat)

NOTA: En aquest sentit, podem concloure que les paraules "requisite", "required", "sufficient" i "optional" no són més que drecceres dels següents valors, respectivament:

```
success=ok    new_authok_reqd=ok    default=die    ignore=ignore
success=ok    new_authok_reqd=ok    default=bad    ignore=ignore
success=done  new_authok_reqd=done  default=ignore
success=ok    new_authok_reqd=ok    default=ignore
```

NOTA: La diferència entre el valor "include" i el valor "substack" és que en aquest darrer el fitxer "mare" veu les línies incloses en ell com una única línia amb un únic resultat "èxit" o "fallada".

3r camp: Nom (o ruta absoluta si és diferent de /lib/security) del fitxer corresponent al mòdul PAM utilitzat, seguit dels possibles paràmetres que pot rebre. En el cas de què el 2ⁿ camp sigui la paraula "include", llavors serà el nom del fitxer de configuració ubicat a /etc/pam.d el contingut del qual es vol "copiar-pegar".

NOTA: Si es vol saber quines aplicacions fan servir un mòdul PAM concret, es pot executar simplement la comanda *grep "nomModul" /etc/pam.d/** i això ja ho mostra

NOTA: Si es vol veure el contingut d'algun d'aquests fitxers de configuració sense les línies de comentaris ni les línies buides (és a dir, només les línies efectives), un truc és executar, en comptes de cat o less, la comanda *grep -vE "(^#|^\$)" /etc/pam.d/nomFitxer*

NOTA: L'arxiu "/etc/pam.d/other" serveix per definir les directives "auth", "account", "password" i "session" de qualsevol aplicació que no incorpori cap fitxer de configuració PAM propi. Normalment, aquest arxiu ho denega tot...és recomanable la seva consulta.

Hi ha moltes aplicacions que comparteixen el mateix esquema d'autenticació. Les distribucions més populars faciliten la tasca d'administrar de forma centralitzada aquestes aplicacions (*login,su,ssh...*) concentrant la configuració comuna en els arxius "common-auth", "common-account", "common-password", "common-session" (Debian/Ubuntu) o "system-auth" i "password-auth" (Fedora/Suse). Per tant, moltes vegades només caldrà modificar aquests arxius per canviar el mode d'autenticació de moltes de les aplicacions del sistema de cop. Per saber quines aplicacions fan servir aquests fitxers comuns (concretament, per exemple el "common-auth") basta amb fer *grep common-auth /etc/pam.d/**.

NOTA: La diferència entre els arxius "system-auth" i "password-auth" a Fedora/Suse és molt poca: de fet, a penes hi ha una o dues línies diferents. En principi, el primer fitxer està pensat per ser utilitzat en autenticacions locals (per *su, sudo*, etc) i el segon està pensat per ser utilitzat en autenticacions remotes (per *ssh, vsftpd*, etc)

NOTA: A Debian/Ubuntu també existeix l'arxiu "common-session-noninteractive", el qual serveix a totes aquelles aplicacions que no proporcionen interactivitat (com ara cron, cups, samba, ppp, etc)

NOTA: Aquesta configuració comuna present en els arxius "common-*" o "*-auth" (segons sigui Debian/Ubuntu o Fedora/Suse, respectivament) es pot modificar mitjançant aplicacions específiques que eviten el haver d'editar directament els fitxers de text. Concretament, a Debian/Ubuntu aquesta aplicació s'anomena *pam-auth-config*, a Fedora s'anomena *authselect* i a Suse s'anomena *pam-config*. No obstant, no les estudiarem.

Tal com ja hem dit, hi ha mòduls que poden utilitzar-se pels quatre valors possibles del 1^r camp i altres que només poden utilitzar-se per algun valor determinat. En concret tenim que (aquesta informació es pot obtenir de la pàgina del manual de cada mòdul):

*Els mòduls de les llistes anteriors que poden usar-se amb els quatre valors són:

pam_unix, pam_listfile, pam_succeed_if, pam_permit/deny, pam_echo, pam_rootok, pam_exec

*Els mòduls de les llistes anteriors que poden usar-se només amb els valors "auth" i "account" són:

pam_userdb, pam_tally2, pam_lastlog, pam_nologin, pam_wheel

*Els mòduls de les llistes anteriors que poden usar-se només amb els valors "auth" i "session" són:

pam_env

*Els mòduls de les llistes anteriors que poden usar-se només amb el valor "auth" són:

pam_securetty, pam_faildelay, pam_captcha, pam_google_authenticator, pam_usb, pam_oath

*Els mòduls de les llistes anteriors que poden usar-se només amb el valor "account" són:

pam_time, pam_access

*Els mòduls de les llistes anteriors que poden usar-se només amb el valor "session" són:

pam_limits, pam_motd, pam_mkhomedir, pam_systemd

*Els mòduls de les llistes anteriors que poden usar-se només amb el valor "password" són:

pam_pwquality, pam_pwhistory

Per més ajuda sobre PAM, veure *man pam* i *man pam.d*

Exemples

Un exemple pot ser el següent fitxer (un `/etc/pam.d/login` simplificat), associat a la comanda *login* (comanda responsable de demanar usuari i contrasenya als terminals virtuals):

```
auth      required      pam_securetty.so
auth      required      pam_env.so
auth      sufficient    pam_ldap.so
auth      required      pam_unix.so try_first_pass
```

Primer de tot, el mòdul “pam_securetty” comprova l'arxiu `/etc/securetty` i mira si el terminal virtual utilitzat està llistat en aquest fitxer; si no és així, els logins amb l'usuari `root` no estaran permesos perquè aquest mòdul fallarà. Notar que un arxiu `/etc/securetty` buit farà que l'usuari “root” no pugui iniciar sessió en cap terminal (i, pel contrari, la inexistència de l'arxiu `/etc/securetty` farà que l'usuari “root” pugui iniciar sessió a qualsevol terminal).

Seguidament, el mòdul “pam_env” establirà variables d'entorn basades en el què l'administrador ha establert a `/etc/security/pam_env.conf`. Aquest mòdul en circumstàncies normals sempre hauria de retornar “èxit” (és per això que s'estableix com “required”).

A continuació, el mòdul “pam_ldap” (no oficial) demanarà un usuari i contrasenya i els compararà amb els emmagatzemats en un servidor de tipus Ldap convenientment configurat; si l'autenticació té èxit, ja s'ha acabat el procés; en canvi, si aquest pas falla, l'autenticació encara pot tenir èxit perquè es continuarà llegint.

Concretament, la darrera línia fa intent de que el mòdul “pam_unix” aconsegueixi autenticar l'usuari en local via `/etc/passwd` i `/etc/shadow`; això és molt útil per poder iniciar sessió a la màquina encara que el servidor Ldap no estigui funcionant. El paràmetre “try_first_pass” indica al mòdul “pam_unix” que utilitzi la contrasenya donada al mòdul anterior (en aquest cas, “pam_ldap”) per no haver de demanar-la un altre cop. Si, finalment, “pam_unix” falla, l'usuari no podrà iniciar sessió.

Un altre exemple de `/etc/pam.d/login` una mica més sofisticat podria ser aquest:

```
#Comprova l'existència de l'arxiu /etc/nologin per, si és el cas, prohibir l'accés a tots els usuaris excepte root
auth      required      pam_nologin.so
#El paràmetre nullok indica que es permeten contrasenyes buides. Un altre paràmetre similar és nullok_secure
auth      required      pam_unix.so nullok
#Declarant-se pam_unix de tipus "account", aquesta aplicació concreta (login) podrà usar pam_unix per obtenir
#informació sobre els comptes d'usuaris (com per exemple si la contrasenya ha expirat)
account   required      pam_unix.so
#Si la contrasenya ha expirat, pam_pwquality demana una nova contrasenya i l'evalua per veure si pot ser fàcilment
#determinada mitjançant diccionaris. El paràmetre retry=3 especifica que si la prova falla la primera vegada, l'usuari té
#dues opcions més per crear una contrasenya millor. El paràmetre minlen indica la longitud mínima obligatòria.
password  required      pam_pwquality.so retry=3 minlen=6
#Per canviar la contrasenya de l'usuari, s'ha d'utilitzar pam_unix.so. El paràmetre shadow indica que creï contrasenyes
#de tipus shadow (les que estan a /etc/shadow), nullok indica que es permet indicar contrasenyes buides i use_authok
#fa que no es demani cap contrasenya en canviar l'antiga sinó que s'utilitzi la que s'hagi introduït en mòduls anteriors
#(és a dir, que hagin superat pam_pwquality)
password  required      pam_unix.so shadow nullok use_authok
#Declarant-se pam_env de tipus "session", s'assegura la seva execució (i, per tant, la declaració de variables) a la
#sessió de l'usuari. El paràmetre envfile indica la ruta d'un fitxer que conté un conjunt de parelles clau->valor per
#definir com a variables i el paràmetre readenv activa la capacitat de llegir aquest fitxer
session   required      pam_env.so readenv=1 envfile=/etc/default/locale
```

EXERCICIS:

Utilitza una màquina virtual Ubuntu o Fedora per fer els exercicis

1.-a) Si una aplicació qualsevol tingués el seu arxiu de configuració dins de `/etc/pam.d` escrit de la següent manera, ¿en quins casos l'accés estaria permès? (és a dir, ¿quines combinacions d'"èxit" entre els diferents mòduls haurien d'ocórrer per permetre l'accés?)

```
auth      required      pam_modulA
auth      sufficient     pam_modulB
auth      required      pam_modulC
```

b) ¿I si estigués escrit d'aquesta altra manera? (fixeu-vos que només ha canviat l'ordre de les línies)

```
auth      sufficient     pam_modulB
auth      required      pam_modulA
auth      required      pam_modulC
```

2.-a) Observa el contingut del mòdul `pam_nologin` dins del fitxer `/etc/pam.d/login` i decideix si has de fer algun canvi (i si sí, quin) per a què ningú (excepte root) pugui iniciar sessió en terminals virtuals. ¡Atenció: abans de provar-ho, si estàs a Ubuntu dóna-li una contrasenya a l'usuari root (amb `sudo passwd root`), que per defecte no en té!

b) Observa l'ús del mòdul `pam_rootok` dins del fitxer `/etc/pam.d/su` i decideix si has de fer algun canvi (i si sí, quin) per a què s'obligui a l'usuari "root" a escriure la contrasenya quan es vulgui convertir en un altre usuari qualsevol

c) Deixa el sistema un altre cop com estava abans d'haver realitzat els apartats anteriors. Seguidament, escriu un línia al principi del fitxer `/etc/pam.d/login` que utilitzi el mòdul `pam_permit` per permetre l'accés des de terminals virtuals a qualsevol usuari sense demanar cap contrasenya. Un cop comprovat aquest canvi, desfes-lo un altre cop.

3.-a) Digues què passa si comentes la línia *auth required pam_succeed_if.so user != root quiet_success* de l'arxiu */etc/pam.d/gdm-password* a Ubuntu (o la línia *auth required pam_succeed_if.so uid >=1000 quiet* de l'arxiu */etc/pam.d/systemd-auth* a Fedora, que seria l'equivalent) i per què. A partir d'aquí, ¿com se t'acudiria prohibir l'inici de sessió a Gnome només per l'usuari "usuari"?

Les condicions que el mòdul *pam_succeed_if* ha de comprovar s'escriuen en forma de paràmetres amb una sintaxi detallada a continuació. Es poden indicar més d'una condició (una rera l'altra, entre espais) però l'autenticació llavors només tindrà èxit si es compleixen totes les condicions sense excepció. Les condicions possibles són les següents (on "camp" pot ser algun dels següents valors: **user**, **uid**, **gid**, **shell**, **home** o **tty**):

```
camp < número
camp <= número
camp eq número
camp ne número
camp >= número
camp > número
camp = cadena
camp != cadena
camp =~ expr. Reg.
camp !~ expr. Reg.
camp in item:item:...
camp notin item:item:...
user ingroup undeterminatgrup
user notingroup undeterminatgrup
```

b) Observa la llista de mòduls "auth" que apareixen a l'arxiu */etc/pam.d/gdm-autologin* i dedueix quins usuaris poden entrar sense contrasenya al sistema si es configura aquesta opció al GDM. ¿Quina diferència hi ha amb la llista de mòduls "auth" que apareixen a */etc/pam.d/gdm-password*?

Per tal d'activar l'ús del mòdul *pam_faillock*, hem d'afegir la línia

```
auth    required    pam_faillock.so preauth ...
```

al principi de l'arxiu */etc/pam.d/common-auth* (a Ubuntu) o dels arxius */etc/pam.d/system-auth* i */etc/pam.d/password-auth* (a Fedora) o de l'arxiu corresponent a l'aplicació concreta que volguem configurar, la línia

```
auth    [default=die] pam_faillock.so authfail ...
```

en el mateix fitxer anterior però després de la línia "auth" responsable de realitzar l'autenticació que volem monitoritzar (normalment, serà la que faci servir el mòdul *pam_unix.so*, que serà de tipus sufficient) i la línia

```
account required    pam_faillock.so authsucc
```

al principi de l'arxiu */etc/pam.d/common-account* (a Ubuntu) o de la secció de línies "account" dins dels arxius */etc/pam.d/system-auth* i */etc/pam.d/password-auth* (a Fedora) o de l'arxiu corresponent a l'aplicació concreta que volguem configurar. Els paràmetres indicats a les línies anteriors serveixen per indicar en quin moment cal comprovar/resetejar el comptador d'inicis fallits i són:

preauth: Es realitza una comprovació del comptador abans d'intentar cap tipus d'autenticació; si l'usuari resulta que està bloquejat, doncs l'autenticació ja no segueix; si l'usuari no ho està, es continua.

authfail: Es realitza una actualització del comptador després d'haver-se produït una autenticació fallida

authsucc: Es reseteja el comptador després d'haver-se produït una autenticació exitosa. Opció per defecte.

NOTA: Es recomana afegir sempre al paràmetre *preauth* un altre paràmetre anomenat *silent* per tal de no donar informació al possible "hacker" sobre si els comptes d'usuaris provats existeixen o no

Els punts suspensius s'han de substituir pels paràmetres necessaris (els quals seran iguals a les dues línies) que serviran per configurar el número d'intents fallits d'inici de sessió, el temps de bloqueig, etc. Concretament, alguns d'aquests paràmetres poden ser:

deny=n° : Indica el número màxim d'intents de logueig permesos. Si se supera, el compte d'usuari es bloqueja
fail_interval=n°: Indica el número de segons, començats a comptar a partir del primer intent fallit, durant els quals s'activarà el comptador d'intents fallits. És a dir, si per exemple *deny=3* i *fail_interval=300*, per bloquejar el compte s'han de fer 3 intents fallits durant un interval de 5 minuts.

unlock_time=n°: Indica el número de segons després dels quals el compte d'usuari bloquejat es desbloquejarà. El comptador comença a comptar a partir de l'inici del bloqueig (i no es resetja). Si el valor indicat és 0, significa temps de bloqueig infinit; en aquest cas, per desbloquejar l'usuari en qüestió caldrà fer servir explícitament la comanda de terminal *faillock*

even_deny_root: El compte de root pot ser bloquejat com qualsevol altre compte (si no es posa aquest paràmetre a l'usuari root no l'afecta aquest mòdul). Si es vol especificar un temps de bloqueig diferent que el de la resta d'usuaris, es pot afegir opcionalment el paràmetre específic *root_unlock_time=n°*

La informació sobre quins usuaris estan bloquejats en un moment donat es troben en sengles arxius dins de la carpeta *"/var/run/faillock"* (podria ser una altra si afegim dir=/ruta/carpeta com a paràmetre del mòdul). Cal tenir en compte que el contingut de la carpeta per defecte es resetja si el sistema es reinicia (perquè de fet, tot el que hi ha a *"/var/run"* és volàtil per definició), així que si es vol que el bloqueig duri més enllà d'un eventual reinici de la màquina caldrà indicar una carpeta permanent.

Per exemple, per bloquejar durant 10 minuts els inicis de sessió amb tres intents fallits perpetrats durant 5 minuts, caldria tenir unes línies similars a:

```
auth    required    pam_faillock.so preauth deny=3 fail_interval=300 unlock_time=600 silent
...
auth    sufficient  pam_unix.so nullok try_first_pass
auth    [default=die] pam_faillock.so authfail deny=3 fail_interval=300 unlock_time=600
...
account required pam_faillock.so authsucc
```

NOTA: El mòdul *pam_faillock* no discrimina per usuaris: els bloqueja tots. Si volem tenir algun usuari que no estigui afectat per aquest mòdul (és a dir, que no tingui límit d'inicis de sessió invàlids), un truc és afegir una línia similar a la següent just abans de la línia *pam_faillock.so preauth...* : *auth [success=1 default=ignore] pam_succeed_if.so user in nomusuari:unaltre:unaltre2*

4.-Crea primer un nou usuari anomenat "pepe" amb la contrasenya que vulguis (això es pot fer ràpidament amb les comandes *sudo useradd -m pepe && sudo passwd pepe*). A partir d'aquí:

a) Configura el mòdul *pam_faillock* per bloquejar qualsevol programa d'autenticació del sistema (*login*, *su*, *gdm*, *ssh*...) durant 50 minuts a qualsevol usuari si aquest s'equivoca tres cops en introduir la contrasenya en un període de 30 segons. Provoca els intents fallits necessaris per tal de bloquejar l'usuari "pepe" (això ho pots fer per exemple executant *su -l pepe* varies vegades amb una contrasenya dolenta). Finalment, prova d'iniciar sessió (un altre cop amb *su -l pepe*) però ara escrivint la contrasenya bona per tal de veure que, efectivament, ja no pots entrar.

b) ¿Per a què serveix la comanda *sudo faillock --user pepe* ? ¿I la comanda *sudo faillock --user pepe --reset*? Executa'ls i comprova si has desbloquejat l'usuari pepe tornant a intentar iniciar sessió amb ell.

El mòdul `pam_time` permet restringir l'accés a un determinat servei del sistema (o al sistema en sí) en base a un horari. És a dir, permet denegar, depenent del dia i de la hora, l'accés a determinats programes a determinats usuaris. La configuració de tot això es realitza a l'arxiu `/etc/security/time.conf`.

Cada línia (que ha d'acabar obligatòriament amb un salt de línia) és una regla (a no ser que comenci amb #, que serà un comentari). La sintaxi d'una regla és la següent: *aplicacions;ttys;usuaris;temps* on “aplicacions” és una llista d'aplicacions PAM (les existents a `/etc/pam.d`) -separades per “|”- a les que afectarà aquest mòdul, “ttys” és la llista de terminals virtuals on tindrà efecte aquest mòdul, “usuaris” és la llista d'usuaris als que els afectarà aquest mòdul -separats per “|”- i “temps” especifica els horaris -separats per “|”- en què les aplicacions indicades pels usuaris indicats (i als terminals indicats) estaran **PERMESES** -prohibint-les a la resta de l'horari que quedi fora-. Per indicar el contrari (l'horari en què estaran prohibides, permetent-les a la resta de l'horari), s'ha d'afegir el símbol “!” al començament de l'horari. El format de l'horari és especificar primer el/s dia/es de la setmana (així: Mo, Tu, We, Th, Fr, Sa, Su) i després l'hora d'inici i final així: (hhmm-hhmm). Per exemple, la línia...:

```
login ; * ; !pepe ; MoTuWeThFr0800-2030 #“!” significa “tots excepte”
```

...permet a tots els usuaris excepte l'usuari pepe loguejar-se a qualsevol terminal virtual entre les 08:00 i 20:30. Un altre exemple: la línia...:

```
* ; * ; pepe ; Al1320-1520|Al1600-2030 #“Al” significa “tots els dies”
```

...permet a l'usuari pepe autenticar-se a la màquina (amb qualsevol mètode: login, gdm, etc) tots els dies però només entre 13:20-15:20 i 16:00-20:30. Un altre exemple és...:

```
* ; * ; pepe ; !Al0000-2400
* ; * ; pepa ; We1300-1600
```

...on pepe mai pot connectar amb la màquina i pepa pot connectar només els dimecres de 13:00 a 16:00.

Hi ha un detall, no obstant, que cal tenir en compte: el rang d'hores especifica l'horari en què l'usuari pot ser autenticat, però un cop ho sigui, no hi hauria limitació temporal en el sentit que podria estar connectat tot el temps que volgués. Si volem tallar la connexió un cop arribat a una hora concreta, haurem de fer servir algun altre mètode, com ara una tasca programada que matés tots els programes oberts per l'usuari en concret (i així, obligar-lo automàticament a tancar sessió); és a dir, executar: *kill -s 9 -u pepe*

Un cop configurat, per tal d'activar el mòdul `pam_time`, hem d'afegir la línia

```
account    required    pam_time.so
```

al principi de l'arxiu `/etc/pam.d/common-account` (o a Fedora, al principi de la secció “account” de l'arxiu `/etc/pam.d/system-auth`) o bé a l'arxiu corresponent a l'aplicació concreta que volem configurar.

NOTA: Si algun usuari del sistema no apareix mencionat dins d'aquest arxiu, no serà afectat per aquest mòdul (és a dir, tindrà accés al sistema a qualsevol hora)

5.-a) Fes que l'usuari “pepe” no pugui iniciar sessió en cap terminal virtual (*login*) durant l'hora actual

b) Canvia la regla anterior per a què ara sí pugui iniciar sessió en els terminals virtuals però que ara no pugui loguejar-se gràficament (*gdm*) durant l'hora actual

El mòdul pam_pwquality (si no està instal·lat caldrà instal·lar el paquet corresponent: a Fedora es diu "libpwquality" i a Ubuntu "libpam-pwquality") permet establir la "qualitat" desitjada per les contrasenyes del sistema. Per tal d'activar-lo, hem d'afegir (si no ho està ja) la línia

```
password    required    pam_pwquality.so ...
```

al principi de l'arxiu "/etc/pam.d/common-password" (o a Fedora, al principi de la secció "password" de l'arxiu "/etc/pam.d/system-auth") o bé a l'arxiu corresponent a l'aplicació concreta que volguem configurar. Els punts suspensius d'han de substituir pels paràmetres adients a cada cas, els quals poden ser alguns com:

local_users_only : Ignora usuaris que no estiguin a "/etc/passwd" (per exemple a un servidor LDAP o altres)

retry=n : Estableix el número d'intents per establir contrasenya abans de deixar-ho córrer

authtok_type : Al missatge que es mostra en canviar la contrasenya s'afegirà allò escrit després de l'igual

El mòdul pam_pwquality realitza una sèrie de comprovacions per defecte com són que la nova contrasenya sigui igual a l'anterior -indiferentment de majúscules/minúscules-, o que estigui revertida ("asdf"->"fdsa") o que estigui rotada ("asdf"->"fasd") a més de comprovar que no estigui en un diccionari propi de paraules comunes (gestionat per la llibreria cracklib). Però pot fer moltes altres coses si s'hi indiquen els paràmetres adients, com ara:

minlen=n : Longitud mínima admissible per la contrasenya.

ucredit=-n : El número indicat és el número mínim de lletres majúscules que ha de tenir la contrasenya

lcredit=-n : El número indicat és el número mínim de lletres minúscules que ha de tenir la contrasenya

dcredit=-n : El número indicat és el número mínim de dígitos que ha de tenir la contrasenya

ocredit=-n : El número indicat és el número mínim d'altres caràcters (símbols) que ha de tenir la "

maxrepeat=n : Rebutja contrasenyes que tinguin més de n vegades seguides el mateix caràcter

maxclassrepeat=n : Rebutja " que tinguin més de n vegades seguides n o mínus o majus o símbols

difok=n : Indica el número de caràcters mínims que han de ser diferents entre la contrasenya nova i l'antiga

dictpath=/ruta/carpeta : Estableix una ruta diferent de l'estàndar pels diccionaris cracklib

És important tenir en compte que el mòdul pam_pwquality només comprova la qualitat de la contrasenya però no fa cap tasca d'actualització real. És per això que el mòdul pam_unix.so se sol escriure just després d'ell (també emprant l'acció "password") per tal de, si és el cas, gravar la nova contrasenya si aquesta compleix els requisits. D'aquesta manera, una configuració típica pot ser la següent:

```
password    requisite    pam_pwquality.so local_users_only retry=3
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
```

Com es pot veure, en aquest cas el mòdul pam_unix també té uns paràmetres propis complementaris, com:

sha512 : Indica que quan es canviï la contrasenya s'haurà de fer servir l'algoritme d'encryptació SHA512 (si no s'indica res es farà servir el que està indicat a /etc/login.defs)

shadow : Indica que la nova contrasenya s'emmagatzemarà a l'arxiu /etc/shadow (i no a /etc/passwd).

nullok : Indica que la nova contrasenya pot estar buida. Un paràmetre similar és "nullok_secure"

try_first_pass : Indica que es reutilitzi la contrasenya que es va introduir en alguna línia anterior (en aquest cas, la demanada per pam_pwquality). Això es fa per a què no es demani a l'usuari la contrasenya dues vegades. El paràmetre "use_authtok" és similar...de fet són redundants

La "qualitat" definida pels paràmetres de pam_pwquality es pot definir per defecte dins de l'arxiu /etc/security/pwquality.conf (per no haver així d'escriure paràmetres específics). Cada línia d'aquest arxiu equival a un dels paràmetres ja vistos.

6.-a) Obliga a què la contrasenya de "pepe" tingui a partir d'ara com a mínim una lletra minúscula, una majúscula, un número i al menys 8 caràcters i tot seguit intenta canviar-la

b) Utilitza la comanda *pwscore* per comprovar si una determinada contrasenya compleix amb els requisits definits a /etc/security/pwquality.conf (o no) . D'altra banda, ¿per a què serveix la comanda *pwmake*?

NOTA: La comanda *cracklib-check* del paquet "cracklib-runtime" és similar a *pwscore* i pot ser una alternativa

c) Podem afegir el mòdul "pam_pwhistory" per afegir més dificultat a l'hora d'establir una nova contrasenya. Concretament, si tinguéssim un sistema Ubuntu amb l'arxiu "common-password" contenint les següents línies...:

password	requisite	pam_pwquality.so	local_users_only	retry=3
password	required	pam_pwhistory.so	remember=400	use_authtok
password	sufficient	pam_unix.so	sha512	shadow use_authtok

...digues, a partir de la consulta pertinent del manual d'aquest mòdul, quantes contrasenyes hauran de canviar-se fins que es pugui tornar a establir una contrasenya repetida. En aquest sentit, ¿quin és el contingut de l'arxiu */etc/security/opasswd*?

Si es crea un usuari amb la comanda *useradd*, dins de l'arxiu */etc/shadow* s'estableixen uns determinats valors relacionats amb la caducitat de la contrasenya, els quals per defecte venen definits dins de */etc/login.defs*. Concretament són *PASS_MAX_DAYS* (número de dies màxim abans d'haver de canviar la contrasenya), *PASS_MIN_DAYS* (número de dies mínim abans de poder canviar la contrasenya un altre cop), *PASS_WARN_AGE* (número de dies abans de la data de caducitat durant els quals s'avisarà a l'usuari). Aquests valors, no obstant, es poden canviar en qualsevol moment o bé editant directament l'arxiu */etc/shadow* o bé, més còmodament, mitjançant la comanda *chage -M dies -m dies -W dies nomUsuari* (on els paràmetres *-M*, *-m* i *-W* es corresponen a les opcions respectives). Sense paràmetres es canviaran aquests valors de forma interactiva. El paràmetre *-l* serveix per obtenir les dates actuals per l'usuari indicat (vaja, el que hi ha gravat a */etc/shadow*).

d) Canvia la data de caducitat de la contrasenya de l'usuari "pepe" per tal de què sigui avui mateix (*-M 0*). Intenta iniciar sessió amb ell (via *login*, *su*, *gdm*...el que vulguis) i comprova que el sistema et demana canviar-la

El mòdul *pam_access* permet restringir l'accés al sistema a determinats usuaris/grups des de determinats terminals/hosts. La configuració es realitza a l'arxiu */etc/security/access.conf*.

Cada línia és una regla (a no ser que comenci amb #, que serà un comentari) i la seva sintaxi és la següent : *permis:usuaris/grups:origen* on "permis" pot ser un "+" (per permetre) o un "-" (per denegar), "usuaris/grups" ha de ser o bé la paraula ALL o bé una llista d'usuaris i/o grups (separats per espais i el nom dels grups escrits entre parèntesi) als quals els afectarà el permís i "origen" ha de ser o bé la paraula ALL o bé una llista de terminals (*tty1,tty2, :0, :1...*) i/o direccionsIP/mask i/o noms de màquines, etc separats per espais des dels quals s'hauran de connectar els usuaris/grups indicats per a què els afecti el permís.

També es pot fer servir tant a "usuaris/grups" com a "origen" l'expressió ALL EXCEPT ...

L'arxiu es deixa de llegir quan es troba la primera ocurrència de usuari<->origen coincident amb l'intent detectat, aplicant-se llavors el permís existent en aquesta ocurrència. És per això que normalment, la darrera línia d'aquest arxiu és *":-:ALL:ALL"* per denegar l'accés a qualsevol usuari que no hagi coincidit amb cap línia anterior.

Un cop configurat, per tal d'activar el mòdul *pam_access*, hem d'afegir (si no ho està ja) la línia

```
account    required    pam_access.so
```

al principi de l'arxiu *"/etc/pam.d/common-account"* (o a Fedora, al principi de la secció "account" de l'arxiu *"/etc/pam.d/system-auth"*) o bé a l'arxiu corresponent a l'aplicació concreta que volguem configurar.

7.-a) Denega l'accés via *login* a l'usuari "pepe" però només si entra des del terminal *tty3* . Un cop provat, desfés el canvi

- b) Denega l'accés via *gdm* a l'usuari "pepe" . Un cop provat, desfés el canvi
- c) ¿Cam faries per denegar l'accés (via *ssh* , per exemple) a l'usuari "pepe" quan es connecti des d'una màquina que no sigui la 192.168.18.10?
- d) Vés a http://ankh-morp.org/code/pam_geiop i digues per a què serveix aquest mòdul PAM

El mòdul *pam_listfile* permet restringir l'ús d'una determinada aplicació autenticadora (o totes si s'empra a l'arxiu "common-auth" o "system-auth") a determinats usuaris/grups i/o des de determinats terminals. La idea és tenir un fitxer (anomenat de qualsevol manera) que contingui la llista d'usuaris/grups/terminals que volem o bé denegar (llista negra) o bé acceptar (llista blanca). A partir d'aquí, per activar l'ús d'aquest mòdul només caldria afegir una línia similar a la següent al principi de l'arxiu ubicat dins de "/etc/pam.d" associat a l'aplicació que volguem gestionar:

```
auth    required    pam_listfile.so item=user sense=deny file=/ruta/fitxer onerr=succeed
```

El paràmetre "item" identifica el tipus d'element que hi ha llistat a "/ruta/fitxer": pot valer "user", "group" o "tty", entre d'altres (cada element de la llista ha d'estar escrit en una línia diferent dins de "/ruta/fitxer"). El paràmetre "sense" indica si es farà servir la llista com a llista negra ("deny") o llista blanca ("allow"). El paràmetre "onerr" indica què fer quan passi algun error (com ara que "/ruta/fitxer" no existeixi o hi hagi algun paràmetre escrit malament, per exemple); si val "succeed" es permetrà l'autenticació, si val "fail" no.

8.-Denega l'accés via *login* a l'usuari "pepe". Prova-ho. Tot seguit, permet l'accés via *gdm* només a l'usuari "pepe" . Prova-ho també. Desfes tots aquests canvis.

El mòdul *pam_limits* permet restringir fins un màxim l'ús de determinats recursos de la màquina (nombre de processos en marxa, número de fitxers oberts, etc) als usuaris ja autenticats via PAM. La configuració d'aquests límits es realitza a l'arxiu "/etc/security/limits.conf".

Cada línia és una regla (a no ser que comenci amb #, que serà un comentari) i la seva sintaxi és la següent (l'espai separador és el tabulador): *usuari**afectats* *tipuslimit* *itemalimitar* *valorlimit* on "usuari*afectats*" pot ser el nom d'un usuari, el nom d'un grup (en aquest cas, hauria d'anar precedit del símbol @), el símbol * per especificar el valor per defecte, un rang d'uids amb la sintaxi *uidmínim:uidmàxim* o un rang de gids amb la sintaxi *@gidmínim:@gidmàxim*; "tipuslimit" pot ser "hard" (indicant que el límit no pot ser superat), "soft" (indicant que el límit pot ser superat però no és recomanable) o "-" (tots dos); "itemalimitar" pot ser "fsize" (tamany màxim permès pels fitxers -en KB-), "nofile" (nº màxim de fitxers oberts), "nproc" (nº màxim de processos), "maxlogins" (nº màxim de logins concurrents), etc;i, finalment, "valorlimit" és el valor màxim de l'item corresponent, el qual és un número amb el significat que correspongui (si val -1, no hi ha límit). Els límits individuals tenen preferència sobre els de grup, així que si s'imposa un límit més estricte a un usuari i un altre més laxa al seu grup, prevaldrà el de l'usuari. Un exemple de fitxer *limits.conf* podria ser:

*	hard	nofile	512
@alumnes	hard	nproc	20
pepe	hard	nproc	0
@alumnes	hard	maxlogins	4

Un cop configurat, per tal d'activar el mòdul *pam_limits*, hem d'afegir (si no ho està ja) la línia

```
session    required    pam_limits.so
```

al principi de l'arxiu "/etc/pam.d/common-session" (o a Fedora, al principi de la secció "session" de l'arxiu "/etc/pam.d/system-auth") o bé a l'arxiu corresponent a l'aplicació concreta que volguem configurar.

9.-a) Fes que l'usuari "pepe" no tingui cap límit a l'hora de generar un número infinit de processos. Seguidament, executa `:(){ :|: &};:` (una alternativa seria `echo "\$0&\$0">_;&chmod +x _;./_`). ¿Què passa? ¿Per què?

NOTA: Els processos iniciats per Systemd ignoren l'arxiu `limits.conf`. Per establir límits similars cal introduir-los dins dels fitxers `.service` pertinents (veure man `systemd.exec`).

aII) Torna a establir per "pepe" un límit per generar un número màxim de processos (per exemple, 20) i torna a executar alguna de les comandes indicades a l'apartat anterior. ¿Què passa ara?

b) Estableix un límit de tamany de fitxer màxim en 10MB per tothom. A continuació, intenta generar un fitxer més gran (així, per exemple: `dd if=/dev/urandom of=fitxer.iso bs=20M count=1`) i explica què passa i per què. Finalment, desfés aquest límit

El mòdul `pam_motd`, serveix per mostrar, a l'inici de sessió en mode text oferit per una determinada aplicació autenticadora (`login`, `su`, `sudo`, `ssh`...) o per totes en general ("`common-auth`" / "`system-auth`"), el contingut estàtic de l'arxiu `/etc/motd` i de tots els arxius (per ordre lexicogràfic) que estiguin dins de la carpeta `/etc/motd.d`. Per utilitzar-lo només cal afegir la línia

```
session optional pam_motd.so
```

al principi de les línies de tipus "session" existents dins de l'arxiu `/etc/pam.d/*` adient.

10.- Crea un fitxer anomenat com vulguis dins de la carpeta `/etc/motd.d` que contingui el següent missatge: "Hola amic". Activa el mòdul `pam_motd` només per l'aplicació `login` i prova'l.

NOTA: Un efecte similar el podries haver obtingut afegint la línia `echo "Hola amic"` a l'arxiu `/etc/profile`, el qual és un script que s'executa a cada inici de sessió en un terminal virtual.

A Ubuntu (només) el mòdul `pam_motd` també és capaç de concatenar els scripts ubicats a `/etc/update-motd.d` (per ordre segons el número indicat al començament del nom de cadascun d'aquests scripts) en un de sol anomenat `/run/motd.dynamic` per seguidament executar-lo; la sortida per pantalla d'aquesta execució es veurà igualment a l'inici de sessió. En el cas de què estiguis treballant en Ubuntu, fes el següent apartat.

b) Instal·la el paquet "weather-util" i crea un script (ha de ser executable, si no no funcionarà!) dins de `/etc/update-motd.d` anomenat "temps.sh" amb el següent contingut ...

```
#!/bin/bash
weather 41.3851n,2.1734w 2>/dev/null | sed -n '/^ /p'
```

... i inicia sessió en un terminal virtual. ¿Què veus i per què?

NOTA: A Fedora es podria simular aquesta funcionalitat però per això caldria instal·lar programes de tercers, com ara aquest <https://github.com/rtnpro/motdgen> o, aquest <https://gist.github.com/cha55son/6042560>

El mòdul `pam_env` carrega a memòria les variables d'entorn que s'hagin definit al fitxer `/etc/security/pam_env.conf` i, si el paràmetre `readenv` d'aquest mòdul val 1, també al fitxer `/etc/environment` (o bé un altre fitxer si a més de `readenv=1` s'especifica la seva ruta completa el paràmetre `envfile=/ruta/altre/fitxer`). Les línies del primer fitxer tenen el format `VARIABLE=valor` i les del segon tenen el format `VARIABLE=valor`. En qualsevol cas, aquests fitxers es llegeixen abans de tots els altres (`/etc/bash.bashrc`, `~/profile`, etc). Aquest mòdul és de tipus "session" i normalment s'escriu com a "required" perquè no hauria de fallar mai

NOTA: Si es vol indicar el valor d'una variable definida prèviament en el fitxer `pam_env.conf` (i fer-lo servir en la definició d'una altra variable posterior, per exemple) s'ha d'emprar la notació `${VARIABLE}`. Existeixen, d'altra banda, dos variables ja predefinides especials, que són `@{HOME}` i `@{SHELL}`, les quals tenen el valor corresponent present dins de l'arxiu `/etc/passwd`.

11.-Fes que només iniciant sessió mitjançant gdm estigui predefinida una variable d'entorn anomenada PEPE amb el valor 1234 . Comprova que així sigui executant `echo $PEPE` en un terminal on prèviament hagi iniciat sessió via gdm

12.-Consulta les següents pàgines de documentació i digues per a què serveixen els següents mòduls PAM oficials, i quina línia concreta auth/account/password/session hauries d'escriure per tal de fer-los servir (fixa't en els exemples):

- a) http://www.linux-pam.org/Linux-PAM-html/sag-pam_echo.html
- b) http://www.linux-pam.org/Linux-PAM-html/sag-pam_lastlog.html
- c) http://www.linux-pam.org/Linux-PAM-html/sag-pam_mkhomedir.html
- d) http://www.linux-pam.org/Linux-PAM-html/sag-pam_wheel.html