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 binaris (o un/s altre/s, segons el que interessi) 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 desitgés, 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 *nobody*)
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_tally2.so	Conta el nº seguits d'intents d'autenticació per tal de bloquejar l'accés si s'excedeix. Un mòdul que implementa una funcionalitat similar (però afegint la possibilitat de compte els intents d'autenticació repetits només dins d'un determinat periode de temps) és **pam_faillock.so**
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 *man systemd.exec*) 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'inclós 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 sinò 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 *figlet* a mode de captcha. La seva pàgina oficial és https://github.com/jordansissel/pam_captcha Atenció: No 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 **pam_mysql.so** **pam_pgsql.so** **pam_sqlite3.so**	Realitza l'autenticació contra usuaris/contrasenyes guardats a un servidor MariaDB.La seva pagina oficial és https://mariadb.com/kb/en/pam-authentication-plugin. Realitza l'autentiació 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) Realitza l'autentiació contra usuaris/contrasenyes guardats a un servidor PostgreSQL. La seva pàgina és https://github.com/pam-pgsql/pam-pgsql Realitza l'autentiació 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'aprofiti 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):

1r 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:nºmàxim 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 1r 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 1r 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 1r 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'efectui. 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 3r 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 dreceres 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 palabra "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 1r 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 intenta 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_authtok
#fa que no es demani cap contrasenya en canviar l'antiga sinò 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_authtok
```
#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
```