Politique d'authentification

L'authentification sur un système Linux ne concerne pas uniquement une personne physique. On trouve à la fois des utilisateurs et des applications, chacun avec sa propre méthode de demande d'authentification. Les bibliothèques PAM (*Pluggable Authentication Modules*) apportent un mécanisme d'authentification simple, souple et unifié.

1. Modules PAM

a. Principes

PAM décrit la manière de développer des programmes indépendamment de la vérification de l'identité. Ces programmes utilisent pour cela des modules qui se chargent de la demande d'exécution. Une politique d'authentification peut, par exemple, autoriser un simple utilisateur à exécuter une commande en local, mais pas à distance.

L'identification se rapporte au login, l'authentification se vérifie par la saisie du mot de passe de l'utilisateur. En cas de changement de type d'authentification, tous les programmes s'y rapportant doivent être modifiés pour le nouveau fonctionnement. C'est ce que simplifie PAM : les directives des interfaces de modules s'additionnent avec un ordonnancement suivant l'ordre de déclaration.

Modifier sans précaution le mécanisme d'authentification sur un système aboutit parfois à son blocage. Soyez prudent et attentif, sinon la seule ressource sera de vous connecter en mode `rescue` (voir le chapitre Maintenance de base du système) afin de remettre les bons fichiers de configuration PAM.

La source de documentation principale pour PAM se trouve à l'adresse :

http://www.kernel.org/pub/linux/libs/pam

Par contre, la multiplicité des modules fait qu'il vaut mieux rechercher directement les informations s'y rapportant. Exemple pour le module `pam_mount` (il permet le montage de volumes pour une session utilisateur) :

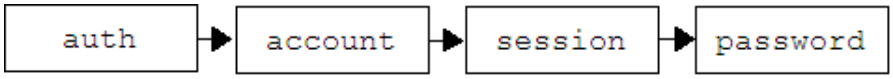
http://pam-mount.sourceforge.net/

b. Configuration et structure des fichiers

Les nombreux modules PAM se situent dans le répertoire `/lib/security`, le fichier de configuration générale dans `/etc/pam.conf` et les fichiers de configuration des services dans `/etc/pam.d`. Le fichier de configuration générale sous Ubuntu est vide et utilisé uniquement si le répertoire `/etc/pam.d` n'existe pas.

Les quatre primitives PAM

Le processus de connexion répond à la suite logique de quatre étapes :



Avec :

- **`auth`** : pour l'identification du compte.
- **`account`** : pour la vérification de l'autorisation.
- **`session`** : pour le contrôle des ressources liées au compte.
- **`password`** : pour la vérification de l'authentification.

Syntaxe d'une ligne d'un fichier PAM

Chaque fichier possède des lignes applicables à des modules (l'ensemble constitue une requête) avec :

- soit une inclusion :

  `@include` *`fichier`*

- soit une structure de la forme :

  `primitive  contrôle  module  arguments`

Le contrôle se décrit par les directives :

- **`required`** : le contrôle doit réussir mais l'enchaînement est vérifié.
- **`requisite`** : le contrôle doit réussir pour continuer l'enchaînement.
- **`sufficient`** : la validation du contrôle suffit pour valider la requête.
- **`optional`** : le contrôle doit réussir seulement dans le cas où c'est le seul.

Les modules sont nombreux (nous avons déjà vu `pam_mount.so`). Une authentification normale Unix utilise le module `pam_unix.so` ; une authentification LDAP (*Lightweight Directory Access Protocol* ou service d'annuaire) utilise le module `pam_ldap.so`. La commande `aptitude search libpam` vous donne la liste non exhaustive mais appliquée à Ubuntu, des modules PAM possibles sur ce système.

Les arguments dépendent du module concerné, il faut se reporter à la documentation propre à chaque module. Par exemple, l'argument `use_first_pass`, commun à plusieurs modules, permet de ne pas retaper le mot de passe quand deux modules l'exigent :

```
...
auth required pam_unix.so nullok_secure
auth required pam_ldap.so use_first_pass
...
```

c. Exemple du fichier /etc/pam.d/login

Certainement le fichier le plus utilisé, il montre les règles de connexion à une session en mode console. Remarquez les quatre inclusions de fichiers, communes à tous les services.

```
#
# The PAM configuration file for the Shadow `login' service
#

# Outputs an issue file prior to each login prompt (Replaces the
# ISSUE_FILE option from login.defs). Uncomment for use
# auth     required   pam_issue.so issue=/etc/issue

# Disallows root logins except on tty's listed in /etc/securetty
# (Replaces the `CONSOLE' setting from login.defs)
auth       requisite  pam_securetty.so

# Disallows other than root logins when /etc/nologin exists
# (Replaces the `NOLOGINS_FILE' option from login.defs)
auth       requisite  pam_nologin.so

# SELinux needs to be the first session rule. This ensures that any
# lingering context has been cleared. Without out this it is
# possible that a module could execute code in the wrong domain.
# (When SELinux is disabled, this returns success.)
session    required   pam_selinux.so close

# This module parses environment configuration file(s)
# and also allows you to use an extended config
# file /etc/security/pam_env.conf.
#
# parsing /etc/environment needs "readenv=1"
```

```
session       required   pam_env.so readenv=1
# locale variables are also kept into /etc/default/locale in etch
# reading this file *in addition to /etc/environment* does not hurt
session       required   pam_env.so readenv=1 envfile=/etc/default/locale

# Standard Un*x authentication.
@include common-auth

# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please edit /etc/security/group.conf to fit your needs
# (Replaces the `CONSOLE_GROUPS' option in login.defs)
auth       optional   pam_group.so

# Uncomment and edit /etc/security/time.conf if you need to set
# time restrainst on logins.
# (Replaces the `PORTTIME_CHECKS_ENAB' option from login.defs
# as well as /etc/porttime)
# account    requisite  pam_time.so

# Uncomment and edit /etc/security/access.conf if you need to
# set access limits.
# (Replaces /etc/login.access file)
# account  required       pam_access.so

# Sets up user limits according to /etc/security/limits.conf
# (Replaces the use of /etc/limits in old login)
session    required   pam_limits.so

# Prints the last login info upon succesful login
# (Replaces the `LASTLOG_ENAB' option from login.defs)
session    optional   pam_lastlog.so

# Prints the motd upon succesful login
# (Replaces the `MOTD_FILE' option in login.defs)
session    optional   pam_motd.so

# Prints the status of the user's mailbox upon succesful login
# (Replaces the `MAIL_CHECK_ENAB' option from login.defs).
#
# This also defines the MAIL environment variable
# However, userdel also needs MAIL_DIR and MAIL_FILE variables
# in /etc/login.defs to make sure that removing a user
# also removes the user's mail spool file.
# See comments in /etc/login.defs
session    optional   pam_mail.so standard

# Standard Un*x account and session
@include common-account
@include common-session
@include common-password

# SELinux needs to intervene at login time to ensure that the
# process starts in the proper default security context. Only
# sessions which are intended to run in the user's context should be
# run after this.  (When SELinux is disabled, this returns success.)
session required pam_selinux.so open
```

Exemple de modification

Vous désirez filtrer la connexion du `root` suivant la console. La toute première ligne du fichier `login` nous montre que ce filtrage se règle au niveau du module `pam_securretty.so` et de son fichier de configuration `/etc/securetty`.

- Ouvrez le fichier par VIM et supprimez par exemple la ligne `tty4`, sauvegardez.
- Ouvrez la quatrième console, tapez comme login le mot `root` : un message indiquant `login incorrect` apparaît.

Autre exemple : `/etc/pam.d/gdm`

Voici ce fichier chargé de la connexion pour une session graphique (type GNOME), expurgé des commentaires :

```
#%PAM-1.0
auth    requisite       pam_nologin.so
auth    required        pam_env.so readenv=1
auth    required        pam_env.so readenv=1 envfile=/etc/default/locale
auth    sufficient      pam_succeed_if.so user ingroup nopasswdlogin
@include common-auth
auth    optional        pam_gnome_keyring.so
@include common-account
session [success=ok ignore=ignore module_unknown=ignore
default=bad] pam_selinux.so close
session required        pam_limits.so
@include common-session
session [success=ok ignore=ignore module_unknown=ignore
default=bad] pam_selinux.so open
session optional        pam_gnome_keyring.so auto_start
@include common-password
```

Remarquez les quatre inclusions identiques à celles du fichier `/etc/pam.d/login`.

Tout le mécanisme d'authentification ne passe pas forcément par PAM. Par exemple, si vous désirez vous connecter en `root` à partir de `gdm` (bloqué par défaut), vous cocherez la case **Autoriser la connexion locale de l'administrateur système** dans l'onglet **Sécurité** du programme **gdmsetup** (**Système - Administration - Fenêtre de connexion**). Cela revient à mettre manuellement la directive `AllowRoot=true` dans le fichier `/etc/gdm/gdm.conf-custom` au niveau de la section `[security]`.

À l'inverse, notez que la connexion automatique par `gdm` sans introduction de mot de passe utilise les modules PAM dans `/etc/pam.d/gdm-autologin`.

2. Utilisation de PAM pour une configuration à un annuaire

En entreprise, le mécanisme d'authentification repose naturellement sur un service d'annuaire de type **LDAP** (*Lightweight Directory Access Protocol*) (Linux) ou **Active Directory** (Windows). On intègre dans cette situation des modules PAM complémentaires, nécessaires à la connexion utilisateur.

a. Connexion à un serveur LDAP

L'exemple traité utilise une version Ubuntu sans session graphique.

Depuis la version Gutsy l'authentification LDAP, plus simple, s'effectue par un outil (optionnel) gérant des profils de connexion centralisés. L'installation du méta-paquetage suivant suffit pour toutes les dépendances :

```
aptitude install ldap-auth-client
```

Questions posées lors de l'installation :

- **l'URL du serveur LDAP** : comme par exemple `ldap://ldap.virtualix.fr` ; mettre l'IP du serveur résout parfois des problèmes de traduction DNS.
- **le nom qualifiant la base** : à partir de notre exemple `dc=virtualix,dc=fr` ; classiquement déduit du nom DNS.
- **la version LDAP utilisée** : usuellement on répond la 3.
- **l'administrateur local comme administrateur de la base** : par défaut sur oui (inutile, voire dangereux), répondre **non**.
- **une obligation de connexion de la base** : usuellement, répondre non.

En cas d'erreur, ces paramètres sont modifiables à partir du fichier /etc/ldap.conf.

Exemples de modification :

Certains serveurs LDAP nécessitent l'emploi :

- de la directive host comme host ldap.virtualix.fr,
- de la directive soft comme bind_policy soft.

Ces directives (et d'autres) sont données par l'administrateur réseau, gestionnaire du serveur LDAP.

Trois profils sont fournis par défaut dans le répertoire /etc/auth-client-config/profile.d/ :

- acc-cracklib : pour forcer des mots de passe plus complexes ;
- acc-default : pour une authentification utilisant le protocole Kerberos ;
- ldap-auth-config : modèle pour une configuration de base utilisant un serveur LDAP.

Le script auth-client-config, en langage Python, positionne les différents fichiers avec une section décrite dans un fichier présent dans /etc/auth-client-config/profile.d/ (voir le modèle). L'installation se réalise alors par :

```
auth-client-config -a -p votre_profil
```

Je lui préfère cependant la configuration manuelle, plus didactique, avec tout d'abord la modification du fichier de configuration des bases de données du système en lui ajoutant la possibilité LDAP pour passwd et group :

```
passwd:     compat ldap
group:      compat ldap
...
```

Vous pouvez dès à présent tester la bonne réponse du serveur LDAP par la commande getent passwd ou getent group, qui doit retourner normalement les comptes de l'annuaire.

Ensuite, suivant l'enchaînement des inclusions PAM, modifiez (ne mettez que ces lignes) :

- le fichier /etc/pam.d/common-auth

```
auth          sufficient         pam_ldap.so
auth          sufficient         pam_unix.so nullok_secure use_first_pass
```

- le fichier /etc/pam.d/common-account

```
account     sufficient     pam_ldap.so
account     required       pam_unix.so
```

- le fichier /etc/pam.d/common-session

```
session     required     pam_mkhomedir.so skel=/etc/skel
session     required     pam_unix.so
session     optional     pam_ldap.so
session     optional     pam_foreground.so
```

- le fichier /etc/pam.d/common-password

```
password     requisite     pam_unix.so nullok obscure min=4 max=8 md5
```

Les modules libpam-mkhomedir et libpam-foreground doivent être, au préalable, installés. La connexion avec un

utilisateur de l'annuaire entraîne, la première fois, la création de son répertoire personnel :

```
root@fai:~# login fougeron
Password:
Linux fai 2.6.24-19-server #1 SMP Wed Jun 18 15:18:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
Creating directory '/home/fougeron'.
fougeron@fai:~$ id
uid=4000(fougeron) gid=1000(users) groups=1000(users),2038(Classe_IG12),5001(Eleves)
fougeron@fai:~$
```

Pour le montage automatique d'un répertoire utilisateur avec `pam_mount`, veuillez vous reporter à un ouvrage traitant d'Ubuntu et de l'administration réseau avec l'installation d'un serveur LDAP.

b. Connexion à un serveur Active Directory

Je prends comme exemple un serveur Windows 2003 édition entreprise avec comme domaine `virtualix.local`. Ce type de serveur utilise **Kerberos** (http://web.mit.edu/Kerberos/) comme protocole d'authentification réseau par défaut.

Vous devez commencer par installer les paquetages liés au protocole Kerberos :

```
aptitude install krb5-user libpam-krb5
```

Renseignez dans le fichier `/etc/hosts` le nom de votre serveur Active Directory (le mien se nomme `win2k3see` ayant pour adresse IP 192.168.3.101) :

```
127.0.0.1          desktop.virtualix.local       desktop
192.168.3.101      win2k3see.virtualix.local     win2k3see
```

Vérifiez par un `ping` la bonne réponse de votre serveur.

Un point important : synchronisez les heures systèmes entre votre serveur et votre client, que ce soit le deuxième par rapport au premier ou les deux sur l'heure universelle.

La configuration de Kerberos se fait de préférence manuellement à partir de son fichier. Ce protocole fait appel à la notion de royaume (REALM) confondu généralement avec le domaine DNS mais toujours exprimé **en majuscules**. Le fichier résultant, en fonction de notre exemple, donne ceci :

Fichier **/etc/krb5.conf**

```
[login]
     default = FILE666:/var/log/krb5lib.log

[libdefaults]
     default_realm = VIRTUALIX.LOCAL
     kdc_timesync = 1
     ccache_type = 4
     forwardable = true
     proxiable = true

[realms]
     VIRTUALIX.LOCAL = {
          admin_server = win2k3see.virtualix.local
          default_domain = VIRTUALIX.LOCAL
          kdc = win2k3see.virtualix.local
     }

[domain_realm]
```

```
    virtualix.local = VIRTUALIX.LOCAL
    .virtualix.local = VIRTUALIX.LOCAL
```

D'ores et déjà vous pouvez obtenir votre "ticket" (Kerberos fonctionne ainsi en lieu et place d'un échange de mots de passe) pour un utilisateur présent dans la base Active Directory du serveur. Voici mon exemple pour un utilisateur nommé **filochard** :

```
root@desktop:~# kinit filochard@VIRTUALIX.LOCAL
Password for filochard@VIRTUALIX.LOCAL:
root@desktop:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: filochard@VIRTUALIX.LOCAL

Valid starting     Expires            Service principal
06/30/08 16:57:25  07/01/08 02:57:26  krbtgt/VIRTUALIX.LOCAL@VIRTUALIX.LOCAL
        renew until 07/01/08 16:57:25


Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
root@desktop:~#
```

La commande `klist` montre les tickets obtenus et en cours.

La démarche suivante concerne l'authentification en utilisant les paquetages `winbind` et SAMBA (LE logiciel d'intégration des plates-formes Linux/WINDOWS) :

```
aptitude install winbind samba
```

La configuration du (long) fichier SAMBA diffère de celle d'un serveur :

Fichier **`/etc/samba/smb.conf`**

```
[global]
 security = ads
 realm = VIRTUALIX.LOCAL
 password server = 192.168.3.101
 workgroup = VIRTUALIX
 encrypt passwords = true
 domain master = no
 local master = no
 winbind use default domain = yes
 idmap uid = 10000-20000
 idmap gid = 10000-20000
 template shell = /bin/bash
 template homedir = /home/%D/%U
 winbind enum groups = yes
 winbind enum users = yes
 client use spnego = yes
 client ntlmv2 auth = yes
```

Il faut ensuite relancer, dans l'ordre, les services :

```
/etc/init.d/samba restart
```

```
/etc/init.d/winbind restart
```

Le domaine Active Directory est désormais joignable pour l'utilisateur. Vous validez la demande de mot de passe à blanc car vous disposez du ticket correspondant :

```
root@desktop:/etc/samba# net ads join
Password:
Using short domain name -- VIRTUALIX
DNS update failed!
Joined 'DESKTOP' to realm 'VIRTUALIX.LOCAL'
root@desktop:/etc/samba#
```

Et PAM dans tout ça ?

De la même façon pour une connexion utilisateur, vous modifiez le fichier `/etc/nsswitch.conf`, cette fois pour `winbind` au lieu de `ldap` :

```
passwd:     compat winbind
group:      compat winbind
...
```

La commande `getent group`, par exemple, retourne alors les groupes de la machine locale et ceux du serveur Active Directory :

```
haldaemon:x:123:
max:x:1000:
winbindd_priv:x:124:
sambashare:x:125:max
ordinateurs du domaine:x:10002:
contrôleurs de domaine:x:10003:
administrateurs du schéma:x:10004:administrateur
administrateurs de l'entreprise:x:10005:administrateur
admins du domaine:x:10006:administrateur
utilisa. du domaine:x:10000:
invités du domaine:x:10001:
propriétaires créateurs de la stratégie de groupe:x:10007:administrateur
dnsupdateproxy:x:10008:
root@desktop:/etc/samba#
```

En cas de non-fonctionnement, n'hésitez pas à relancez les services `samba` et `winbind`.

Suivant la même procédure vue plus haut, remplacez `pam_ldap.so` par `pam_winbind.so` dans les fichiers `common-*` là ou il y est fait référence. La recherche des utilisateurs et des groupes se fait par la commande `wbinfo` :

```
root@desktop:~# wbinfo -u
administrateur
invité
krbtgt
filochard
root@desktop:~# wbinfo -g
ordinateurs du domaine
contrôleurs de domaine
administrateurs du schéma
administrateurs de l'entreprise
admins du domaine
utilisa. du domaine
invités du domaine
propriétaires créateurs de la stratégie de groupe
dnsupdateproxy
root@desktop:~#
```

La connexion à l'utilisateur `filochard` est possible par la commande `login` :

```
root@desktop:~# login filochard
Password:
Dernière connexion : lundi 30 juin 2008 à 18:40:46 CEST sur pts/0
Linux desktop 2.6.24-19-generic #1 SMP Wed Jun 18 14:43:41 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
filochard@desktop:~$ pwd
/home/VIRTUALIX/filochard
filochard@desktop:~$
```

Comme GDM utilise les modules inclus, la connexion par l'interface graphique est tout à fait possible.

La même remarque vue pour LDAP s'impose avec Active Directory : le montage du profil et de son répertoire utilisateur à la connexion passe par le module `pam_mount`. Sa mise en œuvre est de l'ordre de l'administration réseau.

---

3. Plus de sécurité avec PAM

a. Restriction horaire

L'administrateur système applique la politique de sécurité de l'entreprise. Elle porte notamment sur la restriction d'accès des utilisateurs. Dans le cadre normal d'une utilisation des comptes, on peut envisager naturellement la non-possibilité de connexion des utilisateurs en dehors des heures de bureau et donc l'usurpation éventuelle d'un compte. PAM dispose d'un module `pam_time` pour cela, couplé avec l'utilitaire `cron`.

La modification du fichier `/etc/pam.d/gdm` ne suffit pas car elle n'interdira pas une connexion à partir d'une console en mode texte. Par le biais des inclusions la place de `pam_time` se situe dans le fichier `common-account`.

Ajoutez cette ligne à la fin du fichier `/etc/pam.d/common-account` :

```
account required     pam_time.so
```

La modification suivante passe par le fichier `/etc/security/time.conf`. Le fichier contient des explications en commentaires très claires sur le format d'une entrée. Dans notre exemple, nous ne voulons autoriser que la période entre 08h00 et 20h00 à tous les utilisateurs et du lundi au vendredi :

```
login;*;*;Wk0800-2000
```

Le premier champ traitant du service `login`, une mesure de sécurité plus sévère aurait été de mettre une étoile (willcard) afin d'interdire tous les services, comme `ssh` par exemple.

Cas d'une connexion non fermée

Tout le monde ne quitte pas proprement sa session de travail : il arrive fréquemment que le poste reste en l'état au départ de l'employé. Aussi, un mécanisme de déconnexion doit être couplé à cette mesure de sécurité (et d'économie en électricité). La mise en place de l'arrêt du poste se fait par `cron` (voir le chapitre Maintenance de base).

b. Mots de passe renforcés

Longtemps considérée comme accessoire, la politique des mots de passe a été le parent pauvre de la sécurité, partant du faux principe - surtout sur Linux/Unix - que le système est difficilement inviolable à partir du moment où l'administrateur gère et surveille les comptes des utilisateurs.

Même si les dégâts d'un piratage restent circonscrit à un compte utilisateur par le biais de la suppression du compte

administrateur et de la gestion des droits par profils, la perte de donnée (ou le vol) pour un utilisateur n'est pas acceptable. La politique à mettre en place pour les mots de passe consiste en quelques règles :

- forcer le changement à intervalle régulier,
- augmenter la complexité et la longueur,
- garder un historique dans le but d'éviter une réutilisation.

C'est le but du module PAM `libpam-cracklib`, non installé par défaut (il devrait pourtant l'être) :

```
aptitude install libpam-cracklib
```

Le fichier `/etc/pam.d/common-password` donne en exemple les deux lignes essentielles à mettre pour positionner un renforcement de la sécurité avec `pam_cracklib.so` :

```
password required pam_cracklib.so retry=3 minlen=6 difok=3
password required pam_unix.so use_authtok nullok md5
```

La première ligne

Les options du module `pam_cracklib` s'expliquent de la façon suivante :

- **`retry=3`** : trois essais possibles avant une relance du programme `passwd`. Cette option est facultative car la demande de connexion n'étant pas bloquée, rien n'empêche l'utilisateur de recommencer.
- **`minlen=6`** : impose un nombre minimum de caractères pour la longueur d'un mot de passe.
- **`difok=3`** : impose un nombre minimum de caractères différents lors d'un changement de mot de passe.

Par défaut, `pam_cracklib` vérifie si le mot de passe se base trop sur un mot courant ou si certains caractères se retrouvent trop souvent (en liaison aussi avec le paramètre `difok`). Attention toutefois, il arrive que le programme accepte certains mots en se trompant avec des mots d'écriture identique suivant les langues anglaise et française.

Les options supplémentaires `dcredit`, `ucredit`, `lcredit` et `ocredit` expriment respectivement l'obligation de lettres en minuscules, majuscules, chiffres et autres caractères alphanumériques.

La deuxième ligne

Les options de la deuxième ligne montrent l'utilisation du codage classique MD5, `use_authok` indiquant au module `pam_unix` de ne pas utiliser ses propres contrôles (rôle dédié à `pam_cracklib`) et `nullok` d'autoriser (contradiction évidente) les mots de passe vides !

Au niveau de ce module, l'option `remember` sauvegarde un nombre d'anciens mots de passe (et interdit la réutilisation) dans le fichier `/etc/security/opasswd` uniquement accessible à l'administrateur.

Un résultat possible

En fonction de ces remarques, vous pouvez modifier plus clairement ces deux lignes en ceci :

```
password required pam_cracklib.so retry=3 minlen=8 difok=3 ucredit=2
password required pam_unix.so use_authtok md5 remember=24
```

N'oubliez pas que la durée de vie d'un mot de passe se règle dans le fichier `/etc/login.defs` au niveau de la variable PASS_MAX_DAYS.