**LES UTILISATEURS**

# CREATION D'UN UTILISATEUR

**1 - ) UTILISATEUR**

Un utilisateur est identifié au niveau du système, par :

* son nom de **login**
* son mot de passe (commande **passwd**)
* son numéro d**'UID** (identificateur de l'utilisateur)
* son numéro de **GID** (identificateur de groupe)
* son répertoire d'accueil (**HOME DIRECTORY**)
* le programme lancé au moment de la connexion (en général, le **shell**)

1. **Le fichier /etc/passwd**

On déclare un utilisateur en créant une entrée dans le fichier **/etc/passwd**. Ce fichier peut être modifié :

* par n'importe quel éditeur de texte.
* par la commande **vipw**.
* géré à l'aide de procédures de commandes.

**/etc/passwd** contient une ligne par utilisateur. Cette ligne comporte sept champs séparés par le caractère : (deux points). La syntaxe de chaque ligne est la suivante :

**logname:password:UID:GID:libre:répertoire de connexion:shell**

* **logname** : nom de login

- il doit toujours commencer par un caractère alphanumérique.

- il est limité à 32 caractères.

- il ne doit pas contenir d'espace.

- il doit être unique dans le système.

* **password** : mot de passe.

Il est crypté. Il peut être remplacé par le caractère "\*" ou le caractère "x" si le système est en **shadow password**. Dans ce cas, le mot de passe est crypté dans un autre fichier /etc/shadow

- **UID** : User Identifier.

Nombre entier unique dont se sert le système pour identifier l'utilisateur. L'UID O est réservé au super-utilisateur "**root**". Par convention, les numéros compris entre 0 et 499 sont réservés au système. L’UID, en général codé sur 16 bits, ne devra pas dépasser 65535 (216 – 1). Aujourd’hui, on commence à voir certains systèmes qui codent les UID sur 32 bits, dans ce cas, la limite passera à 4294967295 (232 – 1).

* **GID** : Group Identifier

Nombre entier qui détermine le groupe principal de l'utilisateur. Le groupe est déclaré dans le fichier /etc/group. Comme pour les UID, le GID maximum ne peut dépasser 65535 s’il est codé en 16 bits sinon il peut aller jusqu’à 4294967295.

* **libre** :

Zone libre dans laquelle on fait figurer des renseignements concernant l'utilisateur (Prénom Nom, Service, N° de téléphone). Attention, ces champs peuvent être utilisés par certaines commandes : le mail, finger etc…

* **répertoire de connexion** :

Appelé **HOME DIRECTORY**. Il renseigne la variable **HOME**. Il n'est pas nécessaire que le répertoire existe au moment de la création dans **/etc/passwd**. Par contre, il doit être créé pour que l'utilisateur puisse ouvrir une session.

* **shell** : interpréteur de commande.

C'est lui qui permettra de lancer des commandes et programmes à l'utilisateur.

## 2 - ) GROUPE

Un groupe est un ensemble d'utilisateurs ayant des points communs. Il est identifié par :

* + son nom
  + son numéro d'identification (**GID**)
  + la liste des utilisateurs qui le compose.

Si l'utilisateur appartient à un groupe, alors on modifie le fichier **/etc/group**, modifié par un éditeur de texte ou une procédure de commandes. Il comprend une ligne par groupe. Cette ligne est formée de quatre champ séparés par le caractère ":" (deux points)

**groupname::gid:logname1,logname2,…**

- **groupname** : nom du groupe

- **gid** : numéro d'identification du groupe.

- **logname1** liste des noms de login appartenant à ce groupe.

**Remarques** :

- Le deuxième champ n'est pas utilisé.

* La liste des noms de login n'est pas obligatoire. L'utilisateur appartiendra obligatoirement au groupe principal correspondant à son GID décrit dans son entrée du fichier **/etc/passwd**. Par contre un utilisateur peut appartenir à plusieurs groupes secondaires et dans ce cas-là, il devra figurer dans la liste.

Lorsque les fichiers **/etc/passwd** et **/etc/group** sont mis à jour, il faut exécuter les étapes suivantes :

* + création du répertoire de connexion : **mkdir nom\_de\_login**
  + création des variables d'environnement.

**3° ) Etude des commandes et autres fichiers se rapportant aux utilisateurs**

**a) La commande useradd.**

La commande **useradd** peut être utilisée pour ajouter des utilisateurs dans votre système.

Avec la commande **useradd**, vous pouvez indiquer des paramètres :

exemple : **useradd –u 502 –g 300 –d /home/etu/jmm –s /bin/bash jmm**

Dans ce cas, l'utilisateur que nous venons de créer aura les caractéristiques suivantes :

UID : 502

GID : 300

Répertoire de connexion : /home/etu/jmm

shell : /bin/bash

Lorsqu'on ne précise pas de paramètres à la commande **useradd**, ils sont alors pris par défaut.

Les paramètres par défaut sont situés dans le fichier **/etc/default/useradd**. On peut connaître ces paramètres par défaut en tapant la commande useradd –D.

Si les paramètres par défaut ne sont pas ceux que vous souhaitiez, alors vous pouvez les modifier directement dans le fichier **/etc/default/useradd** ou par la commande **useradd –D**.

**Exemple** : - je veux que le groupe par défaut soit 100

- je veux que le mot de passe expire au bout de 60 jours

- je ne veux pas que le compte soit bloqué lors de l'expiration du mot de passe

- je veux que le shell par défaut soit /bin/bash

- je veux que le répertoire de connexion par défaut soit /home/etu

Pour effectuer ces changements, nous utilisons la commande :

**useradd –D –b /home/etu –g 100 –e 60 –f 0 –s /bin/sh**

Maintenant le fichier /etc/default/useradd sera le suivant :

GROUP=100

HOME=/home/etu

INACTIVE=0

EXPIRE=60

SHELL=/bin/bash

SKEL=/etc/skel

Par exemple pour ajouter un utilisateur fred, on peut utiliser la commande suivante :

useradd -m -u 510 –g 300 –c "Frederic Dupont" –s /bin/bash fred

cela créera l'entrée dans /etc/passwd :

fred:x:510:300:Frederic Dupont:/home/etu/fred:/bin/bash

Nous aurons aussi une entrée dans **/etc/shadow**

**b) le fichier /etc/shadow**

Exemple d'entrée dans **/etc/shadow**: Les paramètres par défaut sont définis dans le fichier **/etc/login.defs**

fred:!!:12368:0:99999:7:::

Le premier champ correspond au login.

Le deuxième champ correspond au mot de passe crypté. Attention, le mot de passe est initialisé par la commande **passwd**.

Le champ suivant correspond au nombre de jours écoulés entre le 1 er Janvier 1970 et la dernière modification du mot de passe.que le mot de passe est initialisé.

Le champ suivant correspond au nombre de jour minimum entre deux changements de mot de passe. ( 0 indique que le mot de passe peut être changé à tout moment )

Le champ suivant correspond au nombre de jour maximum qu'un mot de passe soit validé (99999 correspond à un grand nombre d'année). Si ce champ est inférieur à la valeur du champ précédent, l'utilisateur ne pourra pas changer son mot de passe.

Le champ suivant correspond au nombre de jour pour avertir l'utilisateur de l'expiration de son mot de passe (7 correspond à une semaine pleine)

Le champ suivant correspond au nombre de jour avant que le compte soit bloqué suite à l'expiration du mot de passe

Le nombre de jours depuis le 1 janvier 1970, qu'un compte a été neutralisé.

Le dernier champ n'est pas utilisé.

**Remarque** : Le ***timestamp*** (unix) désigne le nombre de secondes écoulées depuis le 1er janvier 1970 à minuit UTC précise La date du 1er janvier 1970 est considérée comme étant la date de naissance du système Unix, ce qui explique pourquoi toutes les horloges systèmes sous Unix démarrent à partir de cette époque.

**c) le fichier /etc/login.defs**

Le fichier **login.defs** contient des paramètres concernant les mots de passe :

**CHFN\_AUTH** : la valeur à yes indique que les programmes chfn et chsh nécessiteront une authentification avant de procéder à tout changement, sauf si ces commandes sont exécutées par root.

**CHFN\_RESTRICT** : Ce paramètre spécifie quelles valeurs du champ gécos du fichier /etc/passwd peuvent être modifiées par les utilisateurs ordinaires avec le programme chfn. Il est composé d’une combinaison de lettres (f,r,w,h) correspondant respectivement au nom complet, au numero de téléphone de bureau, au numéro de téléphone professionnel, et au numero de téléphone personnel. Pour des raisons de compatibilité avec des versions précédentes,’ *yes*’ est équivalent à ‘rwh’ et ‘*no*’ à ‘frwh’. S’il n’est pas spécifié, seul le root peut réaliser les modifications. Pour des mesures plus restrictives, il sera préférable de ne pas installer ***chfn*** avec le drapeau ***SUID***.

**MAIL\_DIR** (chaine) : Le répertoire d’attente des courreils. Ce paramètre est nécessaire pour manipuler les boîtes aux lettres lorsque le compte d’un utilisateur est modifié ou supprimé..

**PASS\_MAX\_DAYS** 99999 : nombre de jour maximum de validation d'un mot de passe

**PASS\_MIN\_DAYS** 0 : nombre de jour minimum de validation d'un mot de passe

**PASS\_MIN\_LEN** 5 : longueur minimale d'un mot de passe

**PASS\_WARN\_AGE** 7 : nombre de jours pour avertir l'expiration d'un mot de passe

**UID\_MIN** 500 : valeur minimale de l’UID pour créer un groupe par défaut

**UID\_MAX** 6000 : valeur de l’UID maximum

**UMASK (nombre)** : Valeur d’initialisation du masque des permissions. S’il n’est pas précisé, le masque est initialisé à 077.

**USERDEL\_CMD** (chaîne) : Si le paramètre est défini, cette commande est exécutée lors de la suppression d’un utilisateur. Elle poura supprimer tous les travaux d’impression de l’utilisateur, par exemple. :

Les paramètres PASS\_MAX\_DAYS, PASS\_MIN\_DAYS et PASS\_WARN\_AGE ne sont utilisés qu’au moment de la création d’un compte. Les changements n’affecteront pas les comptes existants.

**d) la commande chage**

Cette commande permet de configurer le fichier /etc/shadow. C'est elle qui permet de visualiser avec l'option –l, ou de modifier les attributs qui traitent de la pérennité du mot de passe.

Exemples :

Visualiser les attributs du mot de passe d'un utilisateur

# chage –l Nom\_utilisateur

Fixer la date de validité d’un compte d'un utilisateur:

# chage –E aaaa/mm/jj Nom\_utilisateur (la valeur -1 supprime cette durée)

Modifer la durée d’inactivité, après qu’un mot de passe ait dépassé la date de validité :

# chage –I 3 Nom\_utilisateur (la valeur -1 annule la durée d’inactivité d’un compte)

Modifier le nombre minimum de jours entre chaque changement de mot de passe :

# chage –m 3 Nom\_utilisteur (la valeur -1 annule cette durée)

Modifier le nombre maximum de jours pendant lesquels un mot de passe est valable :

# chage –M 90 Nom\_utilisateur (la valeur -1 supprime cette durée)

Modifier le nombre de jours d’avertissement avant le chagement de mot de passe soit obligatoire :

# chage –W 7 Nom\_utilisateur

**Remarques importantes** :

Le programme chage nécessite l’utilisation d’un fichier de mots de passe cachés (shadow password). Ces fonctionnalités ne sont pas disponibles si les mots de passe sont dans le fichier /etc/passwd.

La commande chage est réservée à root, sauf pour l’option –l qui peut être utilisée par un utilisateur non privilégié pour savoir la date de fin de validité de son mot de passe.

**e) la commande pwconv**

Cette commande permet de passer en "shadow passwd". Elle crée le fichier /etc/shadow à partir de /etc/passwd. La commande **pwunconv** passe du mode *shadow* au mode non sécurisé, c'est-à-dire que les mots de passe apparaitront cryptés dans le fichier /etc/passwd.

**f) la commande usermod**

La commande **usermod** est utilisée pour modifier les informations relatives à un utilisateur. Les options sont les mêmes que pour **useradd**.

**g) la commande userdel**

La commande **userdel** supprime un compte. Si **userdel** est utilisée avec l'option **–r**, alors tous les fichiers et sous-répertoires du HOME DIRECTORY seront supprimés. Les fichiers et répertoires situés en dehors du répertoire d'accueil, devront être recherchés et effacés manuellement.

**h) Programmes de vérification de la structure**.

* **pwck** : cette commande est fournie pour vérifier la cohérence des fichiers /etc/passwd et /etc/shadow. Elle vérifie chaque nom d'utilisateur ainsi que les points suivants :

Le nombre correct de champs

Nom unique

Nom et groupe valide

Groupe primaire valide

Répertoire d'accueil valide

Shell valide

Elle prévient aussi lorsqu'un compte ne possède pas de mot de passe.

* **grpck** : cette commande est fournie pour vérifier la cohérence des fichiers /etc/group. Elle effectue les vérifications suivantes :

Le nombre de champs corrects

L'unicité du nom de groupe

La validité de la liste des membres et des administrateurs.

**La gestion des utilisateurs se fait par l'administrateur du système**.

**i) Modifier le mot de passe par un fichier batch par la commande chpasswd**

La commande «chpasswd» est utilisée pour changer le mot de passe pour les utilisateurs. Ceci est une commande d’admin, seul l’utilisateur racine peut modifier le mot de passe pour les utilisateurs.

Exemples**:**

1. Pour réinitialiser le mot de passe pour les utilisateurs de la ligne commande, saisissez

**chpasswd**

Suivi de la saisie des paires codes d’utilisateur: mot de passe, une paire par ligne. Appuyez les touches ctrl+D lorsque vous avez terminé.

user1:passwd1

user2:passwd2

Ctrl-D

2. Utilisation d’un fichier texte :

Nous avons le fichier texte « texte.txt » avec le contenu suivant :

user1:passwd1

user2:passwd2

....

Ensuite nous tapons la commande suivante :

**cat texte.txt | chpasswd**

On peut utiliser des options, en voici quelques unes :

-c Effacer toutes les alertes de mot de passe.

-e Spécifie que les mots de passe sont en un format crypté.

.

**j) Modifier le mot de passe en mode silencieux**

On peut utiliser l’options –stdin

**echo** *mot\_de\_passe***| passwd –stdin** *user*

**4° ) Etude de l'environnement des utilisateurs.**

Résultat de la commande **env**, en italiques : commentaires

**USERNAME**=stagex *<-- Nom d'utilisateur*  
**HISTSIZE**=1000     *<-- Nombre de commandes enregistrées dans l'historique*  
**HOSTNAME**=localhost.domain.fr **LOGNAME**=stagex  *<-- redondant avec username mais demandé par certaines commandes*  
**MAIL**=/var/spool/mail/stagex  *<-- Répertoire du courrier entrant*  
**TERM**=vt100   *<-- Modèle de terminal*  
**HOSTTYPE**=i386  *<-- Type processeur*

*Ordre de recherche des commandes dans les répertoires*  
**PATH**=/opt/kde/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/usr/local/dislin/bin:/usr/local/lam/bin:/home/phys/enseign/epelboin/bin  
**KDEDIR**=/opt/kde   *<-- Répertoire de configuration de l'environnement KDE*  
**HOME**=/home/phys/enseign/stagex  *<--  Répertoire home (défini dans* [*/etc/passwd)*](http://www.lmcp.jussieu.fr/impmc/Enseignement/ye/informatique/unix/documents/passwd.html)  
**SHELL**=/bin/bash   *<--  Shell )par défaut (défini dans* [*/etc/passwd*](http://www.lmcp.jussieu.fr/impmc/Enseignement/ye/informatique/unix/documents/passwd.html)*)*  
**PS1**=[\u@\h \W]\$   *<--  Prompt*  
**USER**=stagex   *<---  Encore une définition de l'utilisateur!*  
**BASH\_ENV**=/home/phys/enseign/stagex/.bashrc *<--  Fichier de configuration du shell*  
**OSTYPE**=Linux

## Script /etc/profile executé en début de session

*Ce script, rangé dans /etc/profile est celui par défaut, employé par tous les utilisateurs.*

# /etc/profile  
# System wide environment and startup programs  
# Functions and aliases go in /etc/bashrc  
PATH="$PATH:/usr/X11R6/bin"  
PS1="[\u@\h \W]\\$ "  
ulimit -c 1000000

*Positionnement du masque de droits pour les fichiers. On distingue le système (id <= 14) et pour les usagers "ordinaires" : rwx propriétaire, rx groupe, rien pour les autres*  
if [ `id -gn` = `id -un` -a `id -u` -gt 14 ]; then  
      umask 007  
else  
      umask 027  
fi  
USER=`id -un`  
LOGNAME=$USER  
MAIL="/var/spool/mail/$USER"  
HOSTNAME=`/bin/hostname`  
HISTSIZE=1000  
export PATH PS1 HOSTNAME HISTSIZE USER LOGNAME MAIL

*S'il existe un script complémentaire dans /etc/profile.d on l'exécute (source) dans le contexte de l'utilisateur*

for i in /etc/profile.d/\*.sh ; do  
      if [ -x $i ]; then  
            . $i  
      fi  
done  
unset i

## Exemple script dans profile.d

*Il existe un script pour initialiser l'environnement kde : kde. sh*

# KDE initialization script (sh)  
if [ -z "$KDEDIR"  -o  "$KDEDIR" != "/opt/kde" ] ; then  
        KDEDIR="/opt/kde"  
        PATH="$KDEDIR/bin:$PATH"  
fi  
export  KDEDIR PATH

## Initialisation du shell bash

*Script général de démarrage de bash, exécuté avant celui propre à l'utilisateur*

/etc/bashrc  
# System wide functions and aliases  
# Environment stuff goes in /etc/profile  
# For some unknown reason bash refuses to inherit  
# PS1 in some circumstances that I can't figure out.  
# Putting PS1 here ensures that it gets loaded every time.

PS1="[\u@\h \W]\\$ "  
alias which="type -path"  
export PATH=$PATH:/usr/local/lam/bin

## Profil particulier utilisateur

*Le profil est séparé en deux parties :*

1. *la définition des alias et l’exécution du script de début de shell*
2. *la définition de l’environnement et du nom du script de début de shell, qui sera lancé, s’il existe, à l’ouverture de chaque shell.*

*Script .bashrc : lance le script général puis les extensions propres à chacun*

# User specific aliases and functions  
# Source global definitions

if [ -f /etc/bashrc ]; then  
        . /etc/bashrc  
fi

alias rm "rm -i"

*Script  .bashrc\_profile : entièrement à définir par l'utilisateur*

# User specific environment and startup programs

PATH=$PATH:$HOME/bin  
BASH\_ENV=$HOME/.bashrc  
USERNAME=""  
export USERNAME BASH\_ENV PATH

#### GESTION DE L’AUTHENTIFICATION D’UN UTILISATEUR AVEC PAM

**1°) Introduction**.

PAM est l’acronyme de Pluggable Authentification Modules (modules d’authentification enfichables).

PAM est le système d’authentification utilisé en standard sous Linux. PAM est un système permettant de gérer individuellement l’authentification pour les applications. Pour cela, il suffit de configurer dans un fichier du même nom que le service ayant besoin d’authentification dans le répertoire /etc/pam.d.

**2°) Format des lignes de configuration des fichiers de /etc/pam.d**

Le format de chaque ligne des fichiers de configuration dans les fichiers de /etc/pam.d est :

**module-type control-flag module-path arguments**

1. **Type de modules** : module-type

* **account** : ce module est chargé de vérifier si l’identification est autorisée (expiration de compte, plage horaire de connexion, …)
* **auth** : ce module est utilisé pour vérifier si l’utilisateur qui veut se connecter est bien celui qu’il prétend être. Ceci passe par un moyen d’authentification comme :

un mot de passe

une phrase de passe

un accès matériel type carte à puce

un accès de type biométrique

* **password** : ce module permet à l’utilisateur de modifier ce qui lui sert pour l’authentification (mot de passe, carte à puce etc…)
* **session** : ce module est utilisé à l’intérieur de la session (après l’identification). Il permet de contrôler, par exemple, l’accès aux répertoires, aux boites mail etc…

1. **Importance du module**: control-flag

* **required**: indique que le succès du module est nécessaire pour autoriser le service. Les autres lignes du fichier sont également lues et ce n’est qu’après que le module est accordé ou non.
* **requisite**: indique que le succès du module est nécessaire pour autoriser le service. Les tests sont stoppés dès le premier échec, ceci permettant de mettre en place une sécurité accrue dans les milieux « hostiles ». Si par exemple, on veut s’assurer que seul tty1 peut être utilisé par le root, on utilise ce drapeau. Ainsi, un utilisateur sur une autre console, n’aura même pas l’occasion d’essayer un mot de passe.
* **sufficient** : indique que le succès suffit à arrêter d’exécuter les modules du même type suivant celui-ci.
* **optional** : indique de ne pas tenir compte du résultat de ce module. C’est uniquement en dernier recours que son ‘avis’ sera pris en compte, lorsque, par exemple, les autres modules renvoie *PAM\_IGNORE*.

#### c ) Le chemin et le nom du module : module-path

#### Indique le nom du module (.so) relatif à /lib/security ou /usr/lib/security si ce nom ne commence pas par un /.

#### d) Les arguments.

C’est une liste de valeur, séparées par un espace,qui peuvent modifier le comportement des modules.

#### 3°) Quelques modules usuels

**pam\_cracklib :** permet d'accepter ou de rejeter un mot de passe, si celui-ci se trouve dans un dictionnaire. Il permet aussi de vérifier que vous ne réutilisez pas le même mot de passe. Vous pouvez le faire suivre de retry=n (le nombre de tentatives) minlen=n (la longueur imposée) difok=n (nombre de caractères qui sont dans le vieux mot de passe et que l'on ne peut pas retrouver dans le nouveau).

**pam\_env :**  permet de spécifier des variables d'environnement spécifiées dans /etc/security/pam\_env.conf à tout utilisateur qui se connecte sur la machine.

**pam\_unix :** Module de base. Gère à la mode unix la politique d'authentification. Il peut être avec les quatre types de modules : account (établi la validité utilisateur/mot de passe et peut forcer la modification de celui là), auth (compare avec la base le mot de passe), password (la politique de changement du mot de passe), session (pour loguer les connexions).  
Vous pouvez associer quelques options dont : nullock pour autoriser un mot de passe vide, md5 pour le type de cryptage, debug pour loguer les informations à syslog, remember=n pour ce souvenir des n derniers mots de passe utilisés.

**pam\_pwdb :** module de base, qui a les mêmes options que pam\_unix.

**pam\_time :** autorise un accès par heure. La configuration se faisant dans le fichier /etc/security/time.conf.

**pam\_wheel :** permet de limiter l'accès à root via la commande su qu'aux seuls membres du groupe wheel. On peut changer le nom du groupe par défaut avec l'option group=mon\_group.

**pam\_limits :** permet de limiter les ressources mis à la disposition d'un utilisateur. Il faut alors configurer le fichier /etc/security/limits.conf. Avec l’argument **conf=filename** indique le chemin du fichier limits.conf s’il n’est pas dans /etc/security

**pam\_nologin :** permet de désactiver les comptes. Il faut alors créer le fichier /etc/nologin et alors il n'y a plus que root qui puisse se connecter.

**pam\_access :** Ce module permet de contrôler les utilisateurs par nom, machine, domaine, adresse IP, terminal. Vous devez alors configurer le fichier /etc/security/access.conf

**pam\_deny :** comme sont nom l'indique. Vous pouvez (devez !) l'utiliser dans /etc/pam.d/other pour auth, account, password et session avec required. Si dans le répertoire /etc/pam.d vous avez des noms d'applications que vous n'utilisez pas, vous pouvez renommer ces fichiers avec un autre nom au cas..! Si quelqu'un cherche à utiliser l'application le other sera alors utilisé par défaut.

**pam\_securetty :** vérifie que le compte root a la possibilité de se connecter sur cette console. Pour cela il faut qu'elle soit indiquée dans le fichier /etc/securetty.

**pam\_warm :** log les informations à syslog

**pam\_console :** permet de spécifier les autorisations d'accès à la console. Il faut alors configurer /etc/security/console.perms.

**pam\_stack :** indique un fichier de configuration PAM à inclure au fichier en cours (modtype : tous). Avec l’argument **service=nom\_service** Nom du service du fichier de configuration PAM à inclure. Généralement suivi de service=system-auth, permet de renvoyer sur system-auth.

**pam\_ldap :** permet d'effectuer l'authentification sur une base ldap. Ce module demande une documentation à lui  tout seul.

**pam\_listfile.so** : indique une liste d’utilisateurs autorisés ou interdits à se connecter (modtype : auth) à partir d’un fichier

Avec l’argument **onerr=succeed|fail** (obligatoire) indique si en cas d’erreur (fichier inexistant), on renvoie succès ou echec.

Avec l’argument **sense=allow|deny** (obligatoire) indique si la liste est autorisante ou interdisante.

Avec l’argument **file=filename** (obligatoire) indique si le fichier contenant la liste d’utilisateur (un par ligne)

**pam\_motd.so** : affiche le contenu de /etc/motd à la connexion (modtype : session)

Avec l’argument **motd=filename** (facultatif) si le fichier motd n’est pas /etc/motd par défaut.

#### pam\_mkhomedir.so : Crée le home directory de l’utilisateur dans le cas où celui-ci ne serait pas créé. Il est chargé dans le module session avec un control-flag à « required ». Il est accompagné par les paramètres skel=/etc/skel ummask=0077

**session required pam\_mkhomedir.so skel=/etc/skel umask=0077**

**1er Cas : Création des utilisateurs par défaut.**

**Modification des fichier /etc/login.defs et /etc/default/useradd**

**A l’origine, nous avons ces deux fichiers avec les paramètres suivants :**

[root@localhost etc]# **more /etc/login.defs**

**# Please note that the parameters in this configuration file control the**

**# behavior of the tools from the shadow-utils component. None of these**

**# tools uses the PAM mechanism, and the utilities that use PAM (such as the**

**# passwd command) should therefore be configured elsewhere. Refer to**

**# /etc/pam.d/system-auth for more information.**

**# \*REQUIRED\***

**# Directory where mailboxes reside, \_or\_ name of file, relative to the**

**# home directory. If you \_do\_ define both, MAIL\_DIR takes precedence.**

**# QMAIL\_DIR is for Qmail**

**#QMAIL\_DIR Maildir**

**MAIL\_DIR /var/spool/mail**

**#MAIL\_FILE .mail**

**# Password aging controls:**

**# PASS\_MAX\_DAYS Maximum number of days a password may be used.**

**# PASS\_MIN\_DAYS Minimum number of days allowed between password changes.**

**# PASS\_MIN\_LEN Minimum acceptable password length.**

**# PASS\_WARN\_AGE Number of days warning given before a password expires.#**

**PASS\_MAX\_DAYS 99999**

**PASS\_MIN\_DAYS 0**

**PASS\_MIN\_LEN 5**

**PASS\_WARN\_AGE 7**

**# Min/max values for automatic uid selection in useradd**

**#**

**# SYS\_GID\_MAX (nombre), SYS\_GID\_MIN (nombre)**

**# Plage d´identifiants numériques de groupes que les commandes useradd, groupadd ou newusers peuvent utiliser pour la création de groupes**

**# système.**

**#**

**# SYS\_UID\_MAX (nombre), SYS\_UID\_MIN (nombre) Plage d´identifiants numériques d´utilisateurs que les commandes useradd ou newusers**

**# peuvent utiliser pour la création d´utilisateurs système.**

**UID\_MIN 1000**

**UID\_MAX 60000**

**# System accounts**

**SYS\_UID\_MIN 201**

**SYS\_UID\_MAX 999**

**# Min/max values for automatic gid selection in groupadd**

**GID\_MIN 1000**

**GID\_MAX 60000**

**# System accounts**

**SYS\_GID\_MIN 201**

**SYS\_GID\_MAX 999**

**# If defined, this command is run when removing a user.**

**# It should remove any at/cron/print jobs etc. owned by**

**# the user to be removed (passed as the first argument).**

**#USERDEL\_CMD /usr/sbin/userdel\_local**

**# If useradd should create home directories for users by default**

**# On RH systems, we do. This option is overridden with the -m flag on**

**# useradd command line.**

**CREATE\_HOME yes**

**# The permission mask is initialized to this value. If not specified,**

**# the permission mask will be initialized to 022.**

**UMASK 077**

**# This enables userdel to remove user groups if no members exist.**

**#**

**USERGROUPS\_ENAB yes**

**# Use SHA512 to encrypt password.**

**ENCRYPT\_METHOD SHA512**

[root@localhost etc]#

[root@localhost /]# **more /etc/default/useradd**

**# useradd defaults file**

**GROUP=100**

**HOME=/home**

**INACTIVE=-1**

**EXPIRE=**

**SHELL=/bin/bash**

**SKEL=/etc/skel**

**CREATE\_MAIL\_SPOOL=yes**

[root@localhost/]#

**Avant modification des fichiers, nous allons créer un utilisateur user1 par la commande useradd en appliquant les paramètres par défaut**

[root@localhost /]# **tail -3 /etc/passwd**

**postfix:x:89:89::/var/spool/postfix:/sbin/nologin**

**sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin**

**tcpdump:x:72:72::/:/sbin/nologin**

[root@localhost /]#

[root@localhost /]# **tail -3 /etc/shadow**

**postfix:!!:16371::::::**

**sshd:!!:16371::::::**

**tcpdump:!!:16371::::::**

[root@localhost /]#

[root@localhost /]# **useradd user1**

[root@localhost /]#

[root@localhost /]# **tail -3 /etc/passwd**

**sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin**

**tcpdump:x:72:72::/:/sbin/nologin**

**user1:x:1000:1000::/home/user1:/bin/bash**

[root@localhost /]#

[root@localhost /]# **tail -3 /etc/shadow**

**sshd:!!:16371::::::**

**tcpdump:!!:16371::::::**

**user1:!!:16388:0:99999:7:::**

[root@localhost /]#

[root@localhost /]# **chage -l user1**

**Dernier changement de mot de passe : nov. 14, 2014**

**Fin de validité du mot de passe : jamais**

**Mot de passe désactivé : jamais**

**Fin de validité du compte : jamais**

**Nombre minimum de jours entre les changements de mot de passe : 0**

**Nombre maximum de jours entre les changements de mot de passe : 99999**

**Nombre de jours d'avertissement avant la fin de validité du mot de passe : 7**

[root@localhost /]#

**Maintenant nous allons modifier le fichier /etc/login.defs.**

**Mettre l’UID minimum à 1500 (variable UID\_MIN=1500)**

**Mettre le GID minimum à 1500 (variable GID\_MIN=1500)**

**Mettre une durée de vie maximale à un mot de passe 100 jours (variable PASS\_MAX\_DAYS=100)**

**Nous allons modifier le fichier /etc/default/useradd**

**Mettre par défaut le HOME\_DIRECTORY à /home/MON\_ENTREPRISE (variable HOME=/home/MON\_ENTRPRISE)**

**Nous allons créer un user2 avec les paramètres par défaut avec la commande useradd**

[root@localhost home]# **mkdir MON\_ENTREPRISE**

[root@localhost home]# **useradd user2**

[root@localhost home]#

[root@localhost home]# **tail -3 /etc/passwd**

**tcpdump:x:72:72::/:/sbin/nologin**

**user1:x:1000:1000::/home/user1:/bin/bash**

**user2:x:1500:1500::/home/MON\_ENTREPRISE/user2:/bin/bash**

[root@localhost home]#

[root@localhost home]# **tail -3 /etc/shadow**

**tcpdump:!!:16371::::::**

**user1:!!:16388:0:99999:7:::**

**user2:!!:16388:0:100:7:::**

[root@localhost home]#

[root@localhost home]# **chage -l user2**

**Dernier changement de mot de passe : nov. 14, 2014**

**Fin de validité du mot de passe : févr. 22, 2015**

**Mot de passe désactivé : jamais**

**Fin de validité du compte : jamais**

**Nombre minimum de jours entre les changements de mot de passe : 0**

**Nombre maximum de jours entre les changements de mot de passe : 100**

**Nombre de jours d'avertissement avant la fin de validité du mot de passe : 7**

[root@localhost home]#

**On voit que les modifications par défaut ont été appliquées.**

**Avec la commande useradd, les groupes correspondants ont été créés.**

[root@localhost /]#

[root@localhost /]# **tail -4 /etc/group**

**sshd:x:74:**

**tcpdump:x:72:**

**user1:x:1000:**

**user2:x:1500:**

[root@localhost /]#

**2 ème cas : création des utilisateurs avec paramètres**

**Nous allons créer :**

* **1 groupe "stage" avec un GID égal à 1710**
* **1 utilisateur "stage1" avec un UID égal à 1510 et faisant parti du groupe "stage". Son HOME\_DIRECTORY sera /home/STAGE/stage1**
* **1 utilisateur "stage2" avec un UID égal à 1950 et faisant parti du groupe "stage". Son HOME\_DIRECTORY sera /home/STAGE/stage2**

**Rappels : les utilisateurs sont décrits dans le fichier /etc/passwd et les groupes dans le fichier /etc/group. Le mot de passe est stocké dans le fichier /etc/shadow.**

* 1. **) Création du groupe STAGE**

[root@localhost home]#

[root@localhost home]# **groupadd -g 1710 stage**

[root@localhost home]#

**On peut vérifier l'ajout du groupe dans le fichier /etc/group par la commande**

[root@localhost home]#

[root@localhost /]# **tail /etc/group**

[root@localhost home]#

**1-2) Création des l'utilisateurs**.

[root@localhost home]#

[root@localhost /]# **cd /home**

[root@localhost home]# **mkdir STAGE**

[root@localhost home]#

[root@localhost home]#

[root@localhost home]# **useradd -u 1510 -g 1710 -d /home/STAGE/stage1 stage1**

[root@localhost home]# **useradd -u 1950 -g 1710 -d /home/STAGE/stage2 stage2**

[root@localhost home]#

**On peut vérifier la présence de l'utilisateur par la commande**

[root@localhost /]#

[root@localhost /]# **tail /etc/passwd**

[root@localhost /]#

**On doit mettre obligatoirement un mot de passe à l'utilisateur.**

[root@localhost home]#

[root@localhost home]# **passwd stage1**

**Changing password for user stage1.**

**New UNIX password:**

**BAD PASSWORD: it is based on a dictionary word**

**Retype new UNIX password:**

**passwd: all authentication tokens updated successfully**.

[root@localhost home]#

**L'utilisateur peut se connecter maintenant. Même chose pour l'utilisateur "stage2"**

**2°) Modification du compte utilisateur**

**2-1) Forcer l’utilisateur à modifier son mot de passe à la connexion**.

**Nous allons utiliser la commande ”chage” avec l’option ”–d0”. Cela a pour effet de modifier le fichier /etc/shadow. Le champ 3 recevra la valeur 0.**

[root@localhost /]#

[root@localhost /]# **chage -d0 stage1**

[root@localhost /]# **tail /etc/shadow**

**tomcat:!!:16371::::::**

**gdm:!!:16371::::::**

**gnome-initial-setup:!!:16371::::::**

**postfix:!!:16371::::::**

**sshd:!!:16371::::::**

**tcpdump:!!:16371::::::**

**user1:$6$rfxs6xpCP3YaTMaM$dKn5ZKPNcDoB70v7J/aAt4vs.SRNfjkKhIFejcuc9vrYYhbhVxYkTwgp7WMbjh0NQWJG2vh5hTZqwstUQ.inK1:16398:0:99999:7:::**

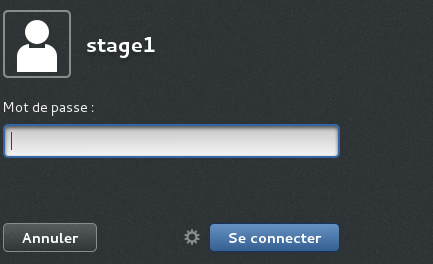
**user2:!!:16398:0:100:7:::**

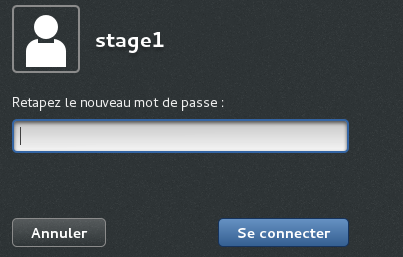
**stage1:$6$IfuJSTRm$jTAdgOZ8r.mPHAXC9IQvHfZZbhqoowQN3y641a62d3atVkHw86N3XPMtgPLt4EvgFqgHiv/CQXzh45iq.NQ6j/:0:0:100:7:::**

**stage2:$6$caAqi.0h$BKUZjV8L5E0HM6wxq9FUHDzqRdk.braeY2RzSAm0WTBV5pb624MOys.vR4E1xBve.yVFYXYodhoc7dgizpvEq0:16398:0:100:7:::**

[root@localhost /]

**Lorsque l’utilisateur stage1 se connectera, il devra changer son mot de passe. On voit que troisième champ est passé à 0.**

**Le fichier /etc/shadow sera ainsi modifié.**

tcpdump:!!:16371::::::

user1:$6$rfxs6xpCP3YaTMaM$dKn5ZKPNcDoB70v7J/aAt4vs.SRNfjkKhIFejcuc9vrYYhbhVxYkTwgp7WMbjh0NQWJG2vh5hTZqwstUQ.inK1:16398:0:99999:7:::

user2:!!:16398:0:100:7:::

stage1:$6$IfuJSTRm$jTAdgOZ8r.mPHAXC9IQvHfZZbhqoowQN3y641a62d3atVkHw86N3XPMtgPLt4EvgFqgHiv/CQXzh45iq.NQ6j/:16398:0:100:7:::

stage2:$6$9i3G.xQH$6mncclPdPbSRapxRkkI5xY8LDK6gAJ6Py1I8zh9vdyASNb.it9Pd741R0/IXqD.LQXKxX8uAXZy6HjesKBAXW0:16398:0:100:7:::

[root@localhost ~]#

**2-2) Changer le délai d’attente après un échec de mot de passe.**

**Le délai par défaut est de deux secondes.**

**Connectez-vous avec l’application ssh, avec l’utilisateur stage1. Donnez un mauvais mot de passe, l’application vous redemande immédiatement le mot de passe**

**Fichier /etc/pam.d/password-auth d’origine**

[root@localhost pam.d]# **more password-auth**

**#%PAM-1.0**

**# This file is auto-generated.**

**# User changes will be destroyed the next time authconfig is run.**

**auth required pam\_env.so**

**auth sufficient pam\_unix.so nullok try\_first\_pass**

**auth requisite pam\_succeed\_if.so uid >= 1000 quiet\_success**

**auth required pam\_deny.so**

**account required pam\_unix.so**

**account sufficient pam\_localuser.so**

**account sufficient pam\_succeed\_if.so uid < 1000 quiet**

**account required pam\_permit.so**

**password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 auth**

**tok\_type=**

**password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok**

**password required pam\_deny.so**

**session optional pam\_keyinit.so revoke**

**session required pam\_limits.so**

**-session optional pam\_systemd.so**

**session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid**

**session required pam\_unix.so**

[root@localhost pam.d]#

**Après modification, nous allons passer ce délai à 10 secondes en modifiant le module ”auth ” du fichier ”/etc/pam.d/password-auth”**

[root@localhost pam.d]# **more password-auth**

**#%PAM-1.0**

**# This file is auto-generated.**

**# User changes will be destroyed the next time authconfig is run.**

**auth required pam\_env.so**

**# on annule le delai de 2 secondes ajoute par pam\_unix**

**auth required pam\_unix.so nullok\_secure nodelay**

**# on ajoute un delai de 10 secondes, temps donne en microsecondes**

**auth optional pam\_faildelay.so delay=10000000**

**auth sufficient pam\_unix.so nullok try\_first\_pass**

**auth requisite pam\_succeed\_if.so uid >= 1000 quiet\_success**

**auth required pam\_deny.so**

**account required pam\_unix.so**

**account sufficient pam\_localuser.so**

**account sufficient pam\_succeed\_if.so uid < 1000 quiet**

**account required pam\_permit.so**

**password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 auth**

**tok\_type=**

**password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok**

**password required pam\_deny.so**

**session optional pam\_keyinit.so revoke**

**session required pam\_limits.so**

**-session optional pam\_systemd.so**

**session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid**

**session required pam\_unix.so**

[root@localhost pam.d]#

**Reconnectez-vous avec stage2 par l’application ssh. Vous voyez que le délai entre la demande du mot de passe est passé à 10 secondes.**

[root@localhost ~]# **ssh localhost -l stage2**

**The authenticity of host 'localhost (::1)' can't be established.**

**ECDSA key fingerprint is 92:be:d5:e1:41:ff:39:b5:06:7d:46:da:3c:be:23:f9.**

**Are you sure you want to continue connecting (yes/no)? yes**

**Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.**

**stage2@localhost's password:**

**# On voit que le temps d’attente après le mot de passé erroné est passé à 10 secondes**

**Permission denied, please try again.**

**stage2@localhost's password:**

**Last failed login: Mon Nov 24 12:16:11 CET 2014 from localhost on ssh:notty**

**There was 1 failed login attempt since the last successful login.**

**Last login: Mon Nov 24 12:13:57 2014**

[stage2@localhost ~]$

**2-3) Historique des mots de passe.**

**Le module pam\_pwhistory sauvegarde les mots de passe des utilisateurs dans le fichier /etc/security/opasswd et force à avoir un mot de passe différents sur n rotations. Par défaut, c’est 10 mot de passe par utilisateur qui sont sauvegardés. On peut changer ce nombre en ajoutant remember=”x” à la directive pam\_pwhistory.**

**A l’origine, le fichier /etc/security/opasswd est vide. Le module password du fichier /etc/pam.d/system -auth est comme suit** :

[root@localhost ~]#

[root@localhost ~]# **more /etc/security/opasswd**

[root@localhost ~]#

**password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 authtok\_type=**

**password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok**

**password required pam\_deny.so**

**Modification du fichier /etc/pam.d/system- auth :**

**password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 authtok\_type=**

**password required pam\_pwhistory.so use\_authtok**

**password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok**

**password required pam\_deny.so**

**Connectez-vous sous l’utilisateur stage2 et modifiez le mot de passe**

[stage2@localhost ~]$

[stage2@localhost ~]$ **passwd**

**Changement de mot de passe pour l'utilisateur stage2.**

**Changement du mot de passe pour stage2.**

**Mot de passe UNIX (actuel) :**

**Nouveau mot de passe :**

**Retapez le nouveau mot de passe :**

**passwd : mise à jour réussie de tous les jetons d'authentification.**

[stage2@localhost ~]$

**Listez le contenu du fichier /etc/security/opasswd**

[root@localhost /]#

[root@localhost /]# **more /etc/security/opasswd**

**stage2:1950:1:$6$9i3G.xQH$6mncclPdPbSRapxRkkI5xY8LDK6gAJ6Py1I8zh9vdyASNb.it9Pd741R0/IXqD.LQXKxX8uAXZy6HjesKBAXW0**

[root@localhost /]#

**2-4) Contrôle des ressources**

**Les différentes ressources du système (nombre de processus, nombre de connexion simultanée, etc … sont gérées par le module pam\_limits.so au niveau de la session dans le fichier /etc/pam.d/system-auth. Les limites sont configurées dans le fichier /etc/security/limits.conf**

[root@localhost /]#

[root@localhost /]# **more /etc/pam.d/system-auth**

**#%PAM-1.0**

**# This file is auto-generated.**

**# User changes will be destroyed the next time authconfig is run.**

**auth required pam\_env.so**

**auth sufficient pam\_fprintd.so**

**auth sufficient pam\_unix.so nullok try\_first\_pass**

**auth requisite pam\_succeed\_if.so uid >= 1000 quiet\_success**

**auth required pam\_deny.so**

**account required pam\_unix.so**

**account sufficient pam\_localuser.so**

**account sufficient pam\_succeed\_if.so uid < 1000 quiet**

**account required pam\_permit.so**

**password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 auth**

**tok\_type=**

**password required pam\_pwhistory.so use\_authtok**

**password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok**

**password required pam\_deny.so**

**session optional pam\_keyinit.so revoke**

**session required pam\_limits.so**

**-session optional pam\_systemd.so**

**session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid**

**session required pam\_unix.so**

[root@localhost /]#

**Modification du fichier /etc/security/limits.conf**

[root@localhost /]#

[root@localhost /]# **tail -8 /etc/security/limits.conf**

# Nombre de connexions simultanees pour l'utilisateur stage1

**stage1 - maxlogins 1**

# Nombre de process pour le groupe stage

**@stage soft nproc 20**

**@stage hard nproc 50**

# End of file

[root@localhost /]#

**A ) Limitation du nombre de connexions**

[root@localhost ~]#

[root@localhost ~]# **ssh localhost -l stage1**

**stage1@localhost's password:**

**Last login: Mon Nov 24 12:54:07 2014 from localhost**

[stage1@localhost ~]$

**Je relance une autre connexion.**

[root@localhost ~]# **ssh localhost -l stage1**

**stage1@localhost's password:**

**Too many logins for 'stage1'.**

**Last login: Mon Nov 24 12:55:57 2014 from localhost**

**Connection to localhost closed.**

[root@localhost ~]#

**B) Limitation du nombre de processus.**

**Je lance une fork bomb, lorsque j’ai atteint le nombre maximum de processus, ma machine ne plante pas mais me renvoie le message suivant**

[stage1@localhost ~]$

[stage1@localhost ~]$ **whoami**

stage1

[stage1@localhost ~]$ **:(){ :|:& };:**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Ressource temporairement non disponible**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**-bash: fork: retry: Aucun processus enfant**

**2-5 ) Limitation de la commande su**.

**Nous allons définir les utilisateurs qui pourront se connecter en root par la commande su.**

- **Modifier le fichier /etc/group sur le groupe wheel et rajouter les utilisateurs autorisés**. **Ici, nous avons rajouté l’utilisateur stage1**

- **Modifier le fichier /etc/pam.d/su en décomentant la ligne**

**auth required /lib/security/pam\_wheel.so use\_uid**

[root@localhost ~]# **more /etc/group**

**root:x:0:**

**bin:x:1:**

**daemon:x:2:**

**sys:x:3:**

**adm:x:4:**

**tty:x:5:**

**disk:x:6:**

**lp:x:7:**

**mem:x:8:**

**kmem:x:9:**

**wheel:x:10:stage1**

**cdrom:x:11:**

[root@localhost ~]# **more /etc/pam.d/su**

**#%PAM-1.0**

**auth sufficient pam\_rootok.so**

**# Uncomment the following line to implicitly trust users in the "wheel" group.**

**#auth sufficient pam\_wheel.so trust use\_uid**

**# Uncomment the following line to require a user to be in the "wheel" group.**

**auth required pam\_wheel.so use\_uid**

**auth substack system-auth**

**auth include postlogin**

**account sufficient pam\_succeed\_if.so uid = 0 use\_uid quiet**

**account include system-auth**

**password include system-auth**

**session include system-auth**

**session include postlogin**

**session optional pam\_xauth.so**

[root@localhost ~]#

**Maintenant, nous allons lancer la commande “ su” sous le user “stage1”, puis “stage2”.**

[stage1@localhost ~]$ **whoami**

**stage1**

[stage1@localhost ~]$ **su -**

**Mot de passe :**

**Dernière connexion : lundi 24 novembre 2014 à 12:14:50 CET sur :0**

[root@localhost ~]#

[stage2@localhost ~]$

[stage2@localhost ~]$ **whoami**

**stage2**

[stage2@localhost ~]$ **su -**

**Mot de passe :**

**su: Autorisation refusée**

[stage2@localhost ~]$

**Nous avons bien le message ‘su : Autorisation refusée’ lié au fait que l’utilisateur stage2 ne fait pas parti du groupe wheel.**

**2-6) Limitation d’un service en fonction des horaires**

**Il faut modifier le fichier /etc/security/time.conf. Ici, nous refusons la connexion à l’utilisateur stage1 le lundi entre 13H00 et 14H00**

[root@localhost ~]#

[root@localhost ~]# **tail /etc/security/time.conf**

# Interdit tous les services au user stage1 le lundi de 13H00 a 14 H00 à partir de tous les termuinaux

**\*;\*;stage1;Mon1300-1400**

**#**

**# End of example file.**

**#**

[root@localhost ~]#

Mon : Lundi Fri : Vendredi

Tue : Mardi Sat : Samedi

Wed : Mercredi Sun : Dimanche

Thu : Jeudi \* ou Al : Tous les Jours

Wd : Samedi/Dimanche

Wk : Lundi/Mardi/Mercredi/Jeudi/Vendredi

**Description de la ligne**

**\* : correspond au champ service. Ici, on interdit tous les services, on auriat pu mettre ‘login’ par exemple.**

**\* : correspond au champ terminal. Ici on interdit tous les services à partir de n’importe quel terminal**

**stage1 : correspond au nom de l’utilisateur à qui s’applique ces restricitions.**

**Mon1400-1700 : correspond au lundi de 13H00 à 14H00**

**On peut mettre plusieurs utilisateurs, dans ce cas on applique l’écriture suivante user1|user2|user3**

**On peut mettre plusieurs terminaux , dans ce cas on applique l’écriture suivante tty1|tty2|tty3**

**Puis modifier le fichier /etc/pam.d/sshd**

[root@localhost ~]#

[root@localhost ~]# **more /etc/pam.d/sshd**

**#%PAM-1.0**

**auth required pam\_sepermit.so**

**auth substack password-auth**

**auth include postlogin**

**account required pam\_nologin.so**

**account required pam\_time.so**

**account include password-auth**

**password include password-auth**

**# pam\_selinux.so close should be the first session rule**

**session required pam\_selinux.so close**

**session required pam\_loginuid.so**

**# pam\_selinux.so open should only be followed by sessions to be executed in the**

**user context**

**session required pam\_selinux.so open env\_params**

**session optional pam\_keyinit.so force revoke**

**session include password-auth**

**session include postlogin**

[root@localhost ~]#

**Voilà ce qui se passe sur la machine. La connexion de stage1 est refusée**

[root@localhost ~]#

[root@localhost ~]# **date**

**lun. nov. 24 13:32:33 CET 2014**

[root@localhost ~]#

[root@localhost ~]# **ssh localhost -l stage1**

**stage1@localhost's password:**

**Connection closed by ::1**

[root@localhost ~]#

**2-7 ) Blocage de compte après plusieurs tentatives de connexions infructueuses**

**Modification du fichier /etc/pam.d/password-auth**

[root@localhost ~]# **more /etc/pam.d/password-auth**

**#%PAM-1.0**

**# This file is auto-generated.**

**# User changes will be destroyed the next time authconfig is run.**

**auth required pam\_env.so**

# on annule le delai de 2 secondes ajoute par pam\_unix

**auth required pam\_unix.so nullok\_secure nodelay**

# on ajoute un delai de 10 secondes, temps donne en microsecondes

**auth optional pam\_faildelay.so delay=10000000**

# on met en place un blocage du compte au bout de deux mot de passe infructueux

**auth required pam\_tally2.so deny=2 even\_deny\_root unlock\_time=120**

**auth sufficient pam\_unix.so nullok try\_first\_pass**

**auth requisite pam\_succeed\_if.so uid >= 1000 quiet\_success**

**auth required pam\_deny.so**

**account required pam\_unix.so**

**account sufficient pam\_localuser.so**

**account sufficient pam\_succeed\_if.so uid < 1000 quiet**

**account required pam\_permit.so**

**password requisite pam\_pwquality.so try\_first\_pass local\_users\_only retry=3 auth**

**tok\_type=**

**password sufficient pam\_unix.so sha512 shadow nullok try\_first\_pass use\_authtok**

**password required pam\_deny.so**

**session optional pam\_keyinit.so revoke**

**session required pam\_limits.so**

**-session optional pam\_systemd.so**

**session [success=1 default=ignore] pam\_succeed\_if.so service in crond quiet use\_uid**

**session required pam\_unix.so**

[root@localhost ~]#

**Nous avons rajouté la librairie pam\_tally2.so avec l’option deny pour indiquer le nombre de tentatives infructueuses à partir duquel le blocage sera effectif (ici, deux tentatives)**

**L’option even\_deny\_root pour indiquer que l’utilisateur root sera aussi bloqué en cas de connexions infructueuses.**

**L’option unlock\_time précise le nombre de secondes que sera bloqué le compte.**

**Exemple :**

[root@localhost ~]# **pam\_tally2**

[root@localhost ~]#

[root@localhost ~]# **ssh localhost -l stage2**

**stage2@localhost's password:**

**Permission denied, please try again.**

**stage2@localhost's password:**

**Permission denied, please try again.**

**stage2@localhost's password:**

Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).

[root@localhost ~]#

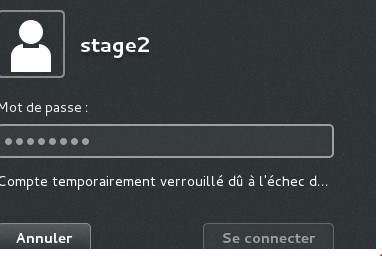
[root@localhost ~]# **pam\_tally2**

**Login Failures Latest failure From**

**stage2 3 11/24/14 13:55:35 localhost**

[root@localhost ~]#

**A la connexion, nous avons le message suivant :**

****

**Attention, le compte est déverrouillé, lorsque le temps de blocage est terminé et que l’utilisateur du compte s’est connecté avec le bon mot de passe.**

* 1. **) Fixer une date de validité d’un compte**

1. **Visualisation des attributs du mot de passe d’un utilisateur**

[root@localhost ~]#

[root@localhost ~]# **chage -l stage2**

**Dernier changement de mot de passe : nov. 24, 2014**

**Fin de validité du mot de passe : mars 04, 2015**

**Mot de passe désactivé : jamais**

**Fin de validité du compte : jamais**

**Nombre minimum de jours entre les changements de mot de passe : 0**

**Nombre maximum de jours entre les changements de mot de passe : 100**

**Nombre de jours d'avertissement avant la fin de validité du mot de passe : 7**

[root@localhost ~]#

1. **Visualisation du fichier /etc/shadow nous montrant les caractéristiques de l’utilisateur stage1**

**tomcat:!!:16371::::::**

**gdm:!!:16371::::::**

**gnome-initial-setup:!!:16371::::::**

**postfix:!!:16371::::::**

**sshd:!!:16371::::::**

**tcpdump:!!:16371::::::**

**user1:$6$rfxs6xpCP3YaTMaM$dKn5ZKPNcDoB70v7J/aAt4vs.SRNfjkKhIFejcuc9vrYYhbhVxYkTwgp7WMbjh0NQWJG2vh5hTZqwstUQ.inK1:16398:0:99999:7:::**

**user2:!!:16398:0:100:7:::**

**stage1:$6$UmJs1GMi$EzFg5xrySl8JD9bRvoe5Dv41QrHvsX/J5Ufkunk8ZoOBGdsq/U6CU5IpIj2/7wKm8GJVBlG3../bqyU.NuQi//:16398:0:100:7:::**

**stage2:$6$tC6JLfoT$QixWYTuUcSvDwULLtr9jVzzXOcQfB23VjeIv7l5c619Sj.prjNa.gPkrFD8J9JJr/eyezUcY0QzeXHU5NCElI.:16398:0:100:7:::**

[root@localhost ~]#

[root@localhost ~]# **date**

**lun. nov. 24 14:03:12 CET 2014**

[root@localhost ~]#

Modification de la date d’expiration du compte (dans l’exemple, le 23 novembre 2014)

[root@localhost ~]#

[root@localhost ~]# **chage -E 2014/11/23 stage2**

[root@localhost ~]#

Cela entrainera la modification du fichier /etc/shadow

[root@localhost ~]#

[root@localhost ~]# **tail /etc/shadow**

**tomcat:!!:16371::::::**

**gdm:!!:16371::::::**

**gnome-initial-setup:!!:16371::::::**

**postfix:!!:16371::::::**

**sshd:!!:16371::::::**

**tcpdump:!!:16371::::::**

**user1:$6$rfxs6xpCP3YaTMaM$dKn5ZKPNcDoB70v7J/aAt4vs.SRNfjkKhIFejcuc9vrYYhbhVxYkTwgp7WMbjh0NQWJG2vh5hTZqwstUQ.inK1:16398:0:99999:7:::**

**user2:!!:16398:0:100:7:::**

**stage1:$6$UmJs1GMi$EzFg5xrySl8JD9bRvoe5Dv41QrHvsX/J5Ufkunk8ZoOBGdsq/U6CU5IpIj2/7wKm8GJVBlG3../bqyU.NuQi//:16398:0:100:7:::**

**stage2:$6$tC6JLfoT$QixWYTuUcSvDwULLtr9jVzzXOcQfB23VjeIv7l5c619Sj.prjNa.gPkrFD8J9JJr/eyezUcY0QzeXHU5NCElI.:16398:0:100:7::16397:**

[root@localhost ~]#

[root@localhost ~]# **chage -l stage2**

**Dernier changement de mot de passe : nov. 24, 2014**

**Fin de validité du mot de passe : mars 04, 2015**

**Mot de passe désactivé : jamais**

**Fin de validité du compte : nov. 23, 2014**

**Nombre minimum de jours entre les changements de mot de passe : 0**

**Nombre maximum de jours entre les changements de mot de passe : 100**

**Nombre de jours d'avertissement avant la fin de validité du mot de passe : 7**

[root@localhost ~]#

**Message à la connexion**

**Par ssh**

[root@localhost ~]#

[root@localhost ~]# **ssh localhost -l stage2**

**stage2@localhost's password:**

**Your account has expired; please contact your system administrator**

**Connection closed by ::1**

[root@localhost ~]#

**Par interface graphique**

