Dernière mise-à-jour : 2020/01/30 03:27

Topic 210 - Gestion de la Sécurité - PAM (3/60)

LPI 210.2 - PAM authentication

Weight: 3

Description: The candidate should be able to configure PAM to support authentication using various available methods. This includes basic SSSD functionality.

Key Knowledge Areas:

- PAM configuration files, terms and utilities
- passwd and shadow passwords
- Use sssd for LDAP authentication

Terms and Utilities:

- /etc/pam.d/
- pam.conf
- nsswitch.conf
- pam_unix, pam_cracklib, pam_limits, pam_listfile, pam_sss
- sssd.conf

RHEL/CentOS 6

PAM (*Pluggable Authentification Modules* ou Modules d'Authentification Enfichables) est une architecture modulaire permettant à l'administrateur système de définir une politique d'authentification pour les logiciels prenant en charge PAM.

Les fichiers de configuration se trouvent dans le répertoire **/etc/pam.d** :

```
[root@centos6 ~]# ls /etc/pam.d
atd                   halt              smartcard-auth-ac
authconfig            ksu               smtp
authconfig-gtk        login             smtp.postfix
authconfig-tui        newrole           sshd
chfn                  other             su
chsh                  passwd            sudo
config-util           password-auth     sudo-i
crond                 password-auth-ac  su-l
cups                  polkit-1          system-auth
cvs                   poweroff          system-auth-ac
eject                 ppp               system-config-authentication
fingerprint-auth      reboot            system-config-date
fingerprint-auth-ac   remote            system-config-kdump
gdm                   run_init          system-config-keyboard
gdm-autologin         runuser           system-config-network
gdm-fingerprint       runuser-l         system-config-network-cmd
gdm-password          setup             system-config-users
gnome-screensaver     smartcard-auth    xserver
```

Ces fichiers ont une structure spécifique et sont nommés d'après le service ou l'application qu'ils contrôlent. Leur contenu fait appel à des modules qui se trouvent dans le répertoire **/lib/security** :

```
[root@centos6 ~]# ls /lib/security
pam_access.so          pam_krb5.so             pam_sepermit.so
pam_cap.so             pam_lastlog.so          pam_shells.so
pam_chroot.so          pam_ldap.so             pam_smbpass.so
pam_ck_connector.so    pam_limits.so           pam_sss.so
pam_console.so         pam_listfile.so         pam_stress.so
pam_cracklib.so        pam_localuser.so        pam_succeed_if.so
pam_debug.so           pam_loginuid.so         pam_tally2.so
pam_deny.so            pam_mail.so             pam_time.so
```

```
pam_echo.so            pam_mkhomedir.so         pam_timestamp.so
pam_env.so             pam_motd.so              pam_tty_audit.so
pam_exec.so            pam_namespace.so         pam_umask.so
pam_faildelay.so       pam_nologin.so           pam_unix_acct.so
pam_filter             pam_oddjob_mkhomedir.so  pam_unix_auth.so
pam_filter.so          pam_passwdqc.so          pam_unix_passwd.so
pam_fprintd.so         pam_permit.so            pam_unix_session.so
pam_ftp.so             pam_postgresok.so        pam_unix.so
pam_gnome_keyring.so   pam_pwhistory.so         pam_userdb.so
pam_group.so           pam_rhosts.so            pam_warn.so
pam_issue.so           pam_rootok.so            pam_wheel.so
pam_keyinit.so         pam_securetty.so         pam_winbind.so
pam_krb5               pam_selinux_permit.so    pam_xauth.so
pam_krb5afs.so         pam_selinux.so
```

Les modules les plus importants sont :

Module	**Description**
pam_listfile.so	Ce module est utilisé pour consulter un fichier spécifique pour vérifier les authiorisations. Par exemple, le service ftp utilise ce module pour consulter le fichier **/etc/ftpusers** qui contient une liste d'utilisateurs qui ne sont **pas** autorisés à se connecter au serveur ftp.
pam_access.so	Ce module est utilisé pour interdire l'accès aux services sécurisés par des hôtes non-autorisés.
pam_nologin.so	Ce module interdit les connexions d'utilisteurs, autre que root, dans le cas où le fichier **/etc/nologin** est présent.
pam_securetty.so	Ce module interdit des connexions de root à partir des périphériques tty qui ne sont pas listés dans le fichier **/etc/securetty**.
pam_cracklib.so	Ce module est utilisé pour vérifier le mot de passe d'un utilisateur
pam_unix.so	Ce module est utilisé pour vérifier les informations suivantes ; expire, last_change, max_change, min_change, warn_change.
pam_limits.so	Ce module implémente les limites des ressources détaillées dans le fichier **/etc/security/limits.conf** et dans les fichiers ***.conf** trouvés dans le répertoire **/etc/security/limits.d/**.
pam_echo.so	Ce module présente le contenu du fichier passé en argument à tout utilisateur lors de sa connexion.

Chaque fichier dans /etc/pam.d contient les règles PAM utilisées pendant l'authentification. Ouvrez le fichier **login** :

```
[root@centos6 ~]# cat /etc/pam.d/login
#%PAM-1.0
```

```
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth       include      system-auth
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
session    optional     pam_console.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      system-auth
-session   optional     pam_ck_connector.so
```

La première ligne de ce fichier est un commentaire qui spécifie que le fichier est conforme à la spécification PAM 1.0.

Ce fichier, tout comme les autres, est ensuite structuré de la façon suivante :

- Une module par ligne,
- Quatre champs séparés par un espace dans chaque règle dont les trois premières sont obligatoires.

Le **premier champs** est le ***type de module***. Il en existe quatre :

Type	**Description**
auth	Utilisé pour authentifier un utilisateur ou les pré-requis système (par exemple /etc/nologin)
account	Utilisé pour vérifier si l'utilisateur peut s'authentifier (par exemple la validité du compte)
password	Utilisé pour vérifier si l'utilisateur dispose des droits pour mettre le mécanisme d'authentification à jour
session	Utilisé pour gérer la session après l'authentification (par exemple monter un répertoire)

Le **deuxième champs** est le ***Control-flag***. Il en existe quatre :

Control-flag	Description
required	La réussite de ce module est indispensable. L'échec d'un module *required* n'est communiqué à l'application qu'après la vérification de tous les modules ayant un *control-flag* de **required**
requisite	La réussite de ce module est indispensable. L'échec d'un module *requisite* est immédiatement communiqué à l'application
sufficient	La réussite de ce module est suffisant pour authoriser l'authentification. Si aucun test *required* précédent est en échec, la vérification s'arrête. Si un test *required* précédent était en échec, le test *sufficient* est ignoré. L'échec d'un test *sufficient* n'a pas de conséquence si tous les tests *required* réussissent.
optional	La réussite ou l'échec de ce module est sans importance, **sauf** s'il s'agit du seul module à exécuter
include	Ce control-flag permet d'inclure toutes les lignes du même *type de module* se trouvant dans le fichier spécifié en argument

Le **troisième champs** stipule le ***module*** associé à la règle. Sans chemin absolu, le fichier est supposé être dans le répertoire **/lib/security**. Pour inclure un module en dehors de ce répertoire il convient donc de stipuler son chemin absolu.

Le **quatrième champs** contient éventuellement les **arguments**.

Ouvrez maintenant le fichier **system-auth** :

```
[root@centos6 ~]# cat /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 >= 500 quiet
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3 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     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
```

Dans ce fichier, si la règle **sufficient** réussit, les modules suivants ne sont pas invoqués.

Utiliser des Mots de Passe Complexe

La complexité des mots de passe est gérée par le module **pam_cracklib.so**. Afin de mettre en place une politique de mots de passe complexe, il convient de modifier la ligne :

```
password    requisite     pam_cracklib.so try_first_pass retry=3 type=
```

en

```
password    requisite     pam_cracklib.so try_first_pass retry=3 minlen=8 lcredit=-1 ucredit=-1 dcredit=-2
ocredit=-1
```

Dans ce cas, le mot de passe doit être long de 8 caractères et doit contenir au moins un caractère minuscule, un caractère majuscule, deux chiffres et un caractère spécial.

Bloquer un Compte après N Echecs de Connexion

Le module PAM **pam_tally.so** permet de bloquer un compte après N échecs de connexion. Afin d'activer ce comportement, il convient d'ajouter dans le fichier **/etc/pam.d/system-auth** la ligne suivante :

```
auth required pam_tally.so onerr=fail deny=3 unlock_time=300
```

Dans ce cas, après trois tentatives infructueuses de connexion, le compte sera bloquer pendant 5 minutes.

Configuration

Certains modules de PAM peuvent être configurés grâce aux fichiers présents dans le répertoire **/etc/security** :

```
[root@centos6 ~]# ls /etc/security
access.conf       console.perms    limits.d        opasswd
chroot.conf       console.perms.d  namespace.conf  pam_env.conf
console.apps      group.conf       namespace.d     sepermit.conf
console.handlers  limits.conf      namespace.init  time.conf
```

Parmi les fichiers cités on note ceux qui peuvent être utilisés pour configurer les modules suivants :

Fichier/*Répertoire*	**Description**
access.conf	Utilisé par le module pam_access.so
console.apps	Utilisés par le module pam_console.so
console.perms	Utilisé par le module pam_console.so
console.perms.d	Utilisé par le module pam_console.so
group.conf	Utilisés par le module pam_group.so
limits.conf	Utilisé par le module pam_limits.so
pam_env.conf	Utilisé par le module pam_env.so
time.conf	Utilisé par le module pam_time.so

A faire : Passez en revue chacun de ces fichiers.

Dernièrement, PAM propose une solution pour toutes les applications ne disposant pas de leurs propres fichiers de configuration PAM. Cette solution prend la forme du fichier **/etc/pam.d/other** :

```
[root@centos6 ~]# cat /etc/pam.d/other
```

```
#%PAM-1.0
auth     required       pam_deny.so
account  required       pam_deny.so
password required       pam_deny.so
session  required       pam_deny.so
```

RHEL/CentOS 7

PAM (*Pluggable Authentification Modules* ou Modules d'Authentification Enfichables) est une architecture modulaire permettant à l'administrateur système de définir une politique d'authentification pour les logiciels prenant en charge PAM.

Les fichiers de configuration se trouvent dans le répertoire **/etc/pam.d** :

```
[root@centos7 ~]# ls /etc/pam.d
atd                     login              smtp
chfn                    other              smtp.postfix
chsh                    passwd             sshd
config-util             password-auth      su
crond                   password-auth-ac   sudo
cups                    pluto              sudo-i
fingerprint-auth        polkit-1           su-l
fingerprint-auth-ac     postlogin          system-auth
gdm-autologin           postlogin-ac       system-auth-ac
gdm-fingerprint         ppp                system-config-language
gdm-launch-environment  remote             systemd-user
gdm-password            runuser            vlock
gdm-pin                 runuser-l          vmtoolsd
gdm-smartcard           setup              xserver
ksu                     smartcard-auth
liveinst                smartcard-auth-ac
```

Ces fichiers ont une structure spécifique et sont nommés d'après le service ou l'application qu'ils contrôlent. Leur contenu fait appel à des modules qui

se trouvent dans le répertoire **/lib64/security** :

```
[root@centos7 ~]# ls /lib64/security
pam_access.so          pam_krb5afs.so           pam_selinux.so
pam_cap.so             pam_krb5.so              pam_sepermit.so
pam_chroot.so          pam_lastlog.so           pam_shells.so
pam_console.so         pam_limits.so            pam_sss.so
pam_cracklib.so        pam_listfile.so          pam_stress.so
pam_debug.so           pam_localuser.so         pam_succeed_if.so
pam_deny.so            pam_loginuid.so          pam_systemd.so
pam_echo.so            pam_mail.so              pam_tally2.so
pam_env.so             pam_mkhomedir.so         pam_time.so
pam_exec.so            pam_motd.so              pam_timestamp.so
pam_faildelay.so       pam_namespace.so         pam_tty_audit.so
pam_faillock.so        pam_nologin.so           pam_umask.so
pam_filter             pam_oddjob_mkhomedir.so  pam_unix_acct.so
pam_filter.so          pam_permit.so            pam_unix_auth.so
pam_fprintd.so         pam_postgresok.so        pam_unix_passwd.so
pam_ftp.so             pam_pwhistory.so         pam_unix_session.so
pam_gnome_keyring.so   pam_pwquality.so         pam_unix.so
pam_group.so           pam_rhosts.so            pam_userdb.so
pam_issue.so           pam_rootok.so            pam_warn.so
pam_keyinit.so         pam_securetty.so         pam_wheel.so
pam_krb5               pam_selinux_permit.so    pam_xauth.so
```

Les modules les plus importants sont :

Module	**Description**
pam_access.so	Ce module est utilisé pour interdire l'accès aux services sécurisés par des hôtes non-autorisés.
pam_echo.so	Ce module présente le contenu du fichier passé en argument à tout utilisateur lors de sa connexion.
pam_limits.so	Ce module implémente les limites des ressources détaillées dans le fichier **/etc/security/limits.conf** et dans les fichiers ***.conf** trouvés dans le répertoire **/etc/security/limits.d/**.

Module	Description
pam_listfile.so	Ce module est utilisé pour consulter un fichier spécifique pour vérifier les authiorisations. Par exemple, le service ftp utilise ce module pour consulter le fichier **/etc/ftpusers** qui contient une liste d'utilisateurs qui ne sont **pas** autorisés à se connecter au serveur ftp.
pam_nologin.so	Ce module interdit les connexions d'utilisteurs, autre que root, dans le cas où le fichier **/etc/nologin** est présent.
pam_pwquality.so	Ce module est utilisé pour vérifier la qualité du mot de passe d'un utilisateur
pam_securetty.so	Ce module interdit des connexions de root à partir des périphériques tty qui ne sont pas listés dans le fichier **/etc/securetty**.
pam_unix.so	Ce module est utilisé pour vérifier les informations suivantes ; expire, last_change, max_change, min_change, warn_change.

Chaque fichier dans /etc/pam.d contient les règles PAM utilisées pendant l'authentification. Ouvrez le fichier **login** :

```
[root@centos7 ~]# cat /etc/pam.d/login
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth       substack     system-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
session    optional     pam_console.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      system-auth
session    include      postlogin
-session   optional     pam_ck_connector.so
```

La première ligne de ce fichier est un commentaire qui spécifie que le fichier est conforme à la spécification PAM 1.0.

Ce fichier, tout comme les autres, est ensuite structuré de la façon suivante :

- Une module par ligne,
- Quatre champs séparés par un espace dans chaque règle dont les trois premières sont obligatoires.

Le **premier champs** est le ***type de module***. Il en existe quatre :

Type	Description
auth	Utilisé pour authentifier un utilisateur ou les pré-requis système (par exemple /etc/nologin)
account	Utilisé pour vérifier si l'utilisateur peut s'authentifier (par exemple la validité du compte)
password	Utilisé pour vérifier si l'utilisateur dispose des droits pour mettre le mécanisme d'authentification à jour
session	Utilisé pour gérer la session après l'authentification (par exemple monter un répertoire)

Le **deuxième champs** est le ***Control-flag***. Il en existe quatre :

Control-flag	Description
required	La réussite de ce module est indispensable. L'échec d'un module *required* n'est communiqué à l'application qu'après la vérification de tous les modules ayant un *control-flag* de **required**
requisite	La réussite de ce module est indispensable. L'échec d'un module *requisite* est immédiatement communiqué à l'application
sufficient	La réussite de ce module est suffisant pour authoriser l'authentification. Si aucun test *required* précédent est en échec, la vérification s'arrête. Si un test *required* précédent était en échec, le test *sufficient* est ignoré. L'échec d'un test *sufficient* n'a pas de conséquence si tous les tests *required* réussissent.
optional	La réussite ou l'échec de ce module est sans importance, **sauf** s'il s'agit du seul module à exécuter
include	Ce control-flag permet d'inclure toutes les lignes du même *type de module* se trouvant dans le fichier spécifié en argument

Le **troisième champs** stipule le ***module*** associé à la règle. Sans chemin absolu, le fichier est supposé être dans le répertoire **/lib/security**. Pour inclure un module en dehors de ce répertoire il convient donc de stipuler son chemin absolu.

Le **quatrième champs** contient éventuellement les **arguments**.

Ouvrez maintenant le fichier **password-auth-ac** :

```
[root@centos7 ~]# cat /etc/pam.d/password-auth-ac
#%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 authtok_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
```

Dans ce fichier, si la règle **sufficient** réussit, les modules suivants ne sont pas invoqués.

Utiliser des Mots de Passe Complexe

La complexité des mots de passe est gérée par le module **pam_pwquality.so**. Afin de mettre en place une politique de mots de passe complexe, il convient de modifier la ligne :

```
password    requisite     pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
```

en

```
password    requisite     pam_pwquality.so try_first_pass local_users_only retry=3 minlen=8 lcredit=-1 ucredit=-1
```

```
dcredit=-2 ocredit=-1
```

Dans ce cas, le mot de passe doit être long de 8 caractères et doit contenir au moins un caractère minuscule, un caractère majuscule, deux chiffres et un caractère spécial.

Bloquer un Compte après N Echecs de Connexion

Le module PAM **pam_tally.so** permet de bloquer un compte après N échecs de connexion. Afin d'activer ce comportement, il convient d'ajouter dans le fichier **/etc/pam.d/system-auth** la ligne suivante :

```
auth required pam_tally.so onerr=fail deny=3 unlock_time=300
```

Dans ce cas, après trois tentatives infructueuses de connexion, le compte sera bloquer pendant 5 minutes.

Configuration

Certains modules de PAM peuvent être configurés grâce aux fichiers présents dans le répertoire **/etc/security** :

```
[root@centos7 ~]# ls /etc/security
access.conf       console.perms    limits.d        opasswd         time.conf
chroot.conf       console.perms.d  namespace.conf  pam_env.conf
console.apps      group.conf       namespace.d     pwquality.conf
console.handlers  limits.conf      namespace.init  sepermit.conf
```

Parmi les fichiers cités on note ceux qui peuvent être utilisés pour configurer les modules suivants :

Fichier/*Répertoire*	**Description**
access.conf	Utilisé par le module pam_access.so
console.apps	Utilisés par le module pam_console.so
console.perms	Utilisé par le module pam_console.so
console.perms.d	Utilisé par le module pam_console.so

Fichier/*Répertoire*	**Description**
group.conf	Utilisés par le module pam_group.so
limits.conf	Utilisé par le module pam_limits.so
pam_env.conf	Utilisé par le module pam_env.so
time.conf	Utilisé par le module pam_time.so

A faire : Passez en revue chacun de ces fichiers.

Dernièrement, PAM propose une solution pour toutes les applications ne disposant pas de leurs propres fichiers de configuration PAM. Cette solution prend la forme du fichier **/etc/pam.d/other** :

```
[root@centos7 ~]# cat /etc/pam.d/other
#%PAM-1.0
auth     required       pam_deny.so
account  required       pam_deny.so
password required       pam_deny.so
session  required       pam_deny.so
```

Dernière mise-à-jour : 2020/01/30 03:27

Debian 8

PAM (*Pluggable Authentification Modules* ou Modules d'Authentification Enfichables) est une architecture modulaire permettant à l'administrateur système de définir une politique d'authentification pour les logiciels prenant en charge PAM.

Les fichiers de configuration se trouvent dans le répertoire **/etc/pam.d** :

```
root@debian8:~# ls /etc/pam.d
atd      common-session            newusers   sshd
```

```
chfn         common-session-noninteractive    other       su
chpasswd     cron                passwd       systemd-user
chsh         lightdm            polkit-1   xscreensaver
common-account   lightdm-autologin        ppp
common-auth  lightdm-greeter        runuser
common-password  login              runuser-l
```

Ces fichiers ont une structure spécifique et sont nommés d'après le service ou l'application qu'ils contrôlent. Leur contenu fait appel à des modules qui se trouvent dans le répertoire **/lib/x86_64-linux-gnu/security/** :

```
root@debian8:~# ls /lib/x86_64-linux-gnu/security
pam_access.so     pam_keyinit.so    pam_permit.so      pam_tally2.so
pam_debug.so      pam_lastlog.so    pam_pwhistory.so   pam_tally.so
pam_deny.so   pam_limits.so     pam_rhosts.so      pam_time.so
pam_echo.so   pam_listfile.so   pam_rootok.so      pam_timestamp.so
pam_env.so    pam_localuser.so  pam_securetty.so   pam_tty_audit.so
pam_exec.so   pam_loginuid.so   pam_selinux.so     pam_umask.so
pam_faildelay.so  pam_mail.so       pam_sepermit.so    pam_unix.so
pam_filter.so     pam_mkhomedir.so  pam_shells.so      pam_userdb.so
pam_ftp.so    pam_motd.so        pam_stress.so      pam_warn.so
pam_group.so      pam_namespace.so  pam_succeed_if.so  pam_wheel.so
pam_issue.so      pam_nologin.so    pam_systemd.so     pam_xauth.so
```

Les modules les plus importants sont :

Module	**Description**
pam_access.so	Ce module est utilisé pour interdire l'accès aux services sécurisés par des hôtes non-autorisés.
pam_cracklib.so	Ce module est utilisé pour vérifier la qualité du mot de passe d'un utilisateur
pam_echo.so	Ce module présente le contenu du fichier passé en argument à tout utilisateur lors de sa connexion.
pam_limits.so	Ce module implémente les limites des ressources détaillées dans le fichier **/etc/security/limits.conf** et dans les fichiers ***.conf** trouvés dans le répertoire **/etc/security/limits.d/**.
pam_listfile.so	Ce module est utilisé pour consulter un fichier spécifique pour vérifier les authiorisations. Par exemple, le service ftp utilise ce module pour consulter le fichier **/etc/ftpusers** qui contient une liste d'utilisateurs qui ne sont **pas** autorisés à se connecter au serveur ftp.

Module	Description
pam_nologin.so	Ce module interdit les connexions d'utilisteurs, autre que root, dans le cas où le fichier **/etc/nologin** est présent.
pam_securetty.so	Ce module interdit des connexions de root à partir des périphériques tty qui ne sont pas listés dans le fichier **/etc/securetty**.
pam_unix.so	Ce module est utilisé pour vérifier les informations suivantes ; expire, last_change, max_change, min_change, warn_change.

Chaque fichier dans /etc/pam.d contient les règles PAM utilisées pendant l'authentification. Ouvrez le fichier **login** :

```
root@debian8:~# cat /etc/pam.d/login
#
# The PAM configuration file for the Shadow `login' service
#

# Enforce a minimal delay in case of failure (in microseconds).
# (Replaces the `FAIL_DELAY' setting from login.defs)
# Note that other modules may require another minimal delay. (for example,
# to disable any delay, you should add the nodelay option to pam_unix)
auth       optional   pam_faildelay.so  delay=3000000

# 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)
#
# With the default control of this module:
#   [success=ok new_authtok_reqd=ok ignore=ignore user_unknown=bad default=die]
# root will not be prompted for a password on insecure lines.
# if an invalid username is entered, a password is prompted (but login
# will eventually be rejected)
#
# You can change it to a "requisite" module if you think root may mis-type
# her login and should not be prompted for a password in that case. But
# this will leave the system as vulnerable to user enumeration attacks.
```

```
#
# You can change it to a "required" module if you think it permits to
# guess valid user names of your system (invalid user names are considered
# as possibly being root on insecure lines), but root passwords may be
# communicated over insecure lines.
auth [success=ok new_authtok_reqd=ok ignore=ignore user_unknown=bad default=die] 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 the module is present, "required" would be sufficient (When SELinux
# is disabled, this returns success.)
session [success=ok ignore=ignore module_unknown=ignore default=bad] 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 message of the day upon succesful login.
# (Replaces the `MOTD_FILE' option in login.defs)
session    optional   pam_exec.so type=open_session stdout /bin/uname -snrvm
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

# Sets the loginuid process attribute
session    required     pam_loginuid.so

# 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.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
# When the module is present, "required" would be sufficient (When SELinux
# is disabled, this returns success.)
```

La première ligne de ce fichier est un commentaire qui spécifie que le fichier est conforme à la spécification PAM 1.0.

Ce fichier, tout comme les autres, est ensuite structuré de la façon suivante :

- Une module par ligne,
- Quatre champs séparés par un espace dans chaque règle dont les trois premières sont obligatoires.

Le **premier champs** est le ***type de module***. Il en existe quatre :

Type	**Description**
auth	Utilisé pour authentifier un utilisateur ou les pré-requis système (par exemple /etc/nologin)
account	Utilisé pour vérifier si l'utilisateur peut s'authentifier (par exemple la validité du compte)
password	Utilisé pour vérifier si l'utilisateur dispose des droits pour mettre le mécanisme d'authentification à jour
session	Utilisé pour gérer la session après l'authentification (par exemple monter un répertoire)

Le **deuxième champs** est le ***Control-flag***. Il en existe quatre :

Control-flag	Description
required	La réussite de ce module est indispensable. L'échec d'un module *required* n'est communiqué à l'application qu'après la vérification de tous les modules ayant un *control-flag* de **required**
requisite	La réussite de ce module est indispensable. L'échec d'un module *requisite* est immédiatement communiqué à l'application
sufficient	La réussite de ce module est suffisant pour authoriser l'authentification. Si aucun test *required* précédent est en échec, la vérification s'arrête. Si un test *required* précédent était en échec, le test *sufficient* est ignoré. L'échec d'un test *sufficient* n'a pas de conséquence si tous les tests *required* réussissent.
optional	La réussite ou l'échec de ce module est sans importance, **sauf** s'il s'agit du seul module à exécuter
include	Ce control-flag permet d'inclure toutes les lignes du même *type de module* se trouvant dans le fichier spécifié en argument

Le **troisième champs** stipule le ***module*** associé à la règle. Sans chemin absolu, le fichier est supposé être dans le répertoire **/lib/security**. Pour inclure un module en dehors de ce répertoire il convient donc de stipuler son chemin absolu.

Le **quatrième champs** contient éventuellement les **arguments**.

Utiliser des Mots de Passe Complexe

La complexité des mots de passe est gérée par le module **pam_cracklib.so**. Commencez par installer **libpam-cracklib** :

```
root@debian8:~# apt-get install libpam-cracklib
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  cracklib-runtime libcrack2
The following NEW packages will be installed:
  cracklib-runtime libcrack2 libpam-cracklib
0 upgraded, 3 newly installed, 0 to remove and 95 not upgraded.
Need to get 289 kB of archives.
After this operation, 1,195 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.fr.debian.org/debian/ jessie/main libcrack2 amd64 2.9.2-1 [54.7 kB]
Get:2 http://ftp.fr.debian.org/debian/ jessie/main cracklib-runtime amd64 2.9.2-1 [148 kB]
```

```
Get:3 http://ftp.fr.debian.org/debian/ jessie/main libpam-cracklib amd64 1.1.8-3.1+deb8u1+b1 [86.0 kB]
Fetched 289 kB in 0s (323 kB/s)
Selecting previously unselected package libcrack2:amd64.
(Reading database ... 82496 files and directories currently installed.)
Preparing to unpack .../libcrack2_2.9.2-1_amd64.deb ...
Unpacking libcrack2:amd64 (2.9.2-1) ...
Selecting previously unselected package cracklib-runtime.
Preparing to unpack .../cracklib-runtime_2.9.2-1_amd64.deb ...
Unpacking cracklib-runtime (2.9.2-1) ...
Selecting previously unselected package libpam-cracklib:amd64.
Preparing to unpack .../libpam-cracklib_1.1.8-3.1+deb8u1+b1_amd64.deb ...
Unpacking libpam-cracklib:amd64 (1.1.8-3.1+deb8u1+b1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libcrack2:amd64 (2.9.2-1) ...
Setting up cracklib-runtime (2.9.2-1) ...
Setting up libpam-cracklib:amd64 (1.1.8-3.1+deb8u1+b1) ...
Processing triggers for libc-bin (2.19-18+deb8u4) ...
```

Afin de mettre en place une politique de mots de passe complexe, il convient de modifier la ligne suivante du fichier **cat /etc/pam.d/common-password** :

```
password    requisite            pam_cracklib.so retry=3 minlen=8 difok=3
```

en

```
password    requisite            pam_cracklib.so retry=3 minlen=8 lcredit=-1 ucredit=-1 dcredit=-2 ocredit=-1
```

```
root@debian8:~# cat /etc/pam.d/common-password
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.
```

```
# Explanation of pam_unix options:
#
# The "sha512" option enables salted SHA512 passwords.  Without this option,
# the default is Unix crypt.  Prior releases used the option "md5".
#
# The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
# login.defs.
#
# See the pam_unix manpage for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password    requisite            pam_cracklib.so retry=3 minlen=8 lcredit=-1 ucredit=-1 dcredit=-2 ocredit=-1
password    [success=1 default=ignore]    pam_unix.so obscure use_authtok try_first_pass sha512
# here's the fallback if no module succeeds
password    requisite            pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password    required            pam_permit.so
# and here are more per-package modules (the "Additional" block)
password    optional    pam_gnome_keyring.so
# end of pam-auth-update config
```

Dans ce cas, le mot de passe doit être long de 8 caractères et doit contenir au moins un caractère minuscule, un caractère majuscule, deux chiffres et un caractère spécial.

Bloquer un Compte après N Echecs de Connexion

Le module PAM **pam_tally.so** permet de bloquer un compte après N échecs de connexion. Afin d'activer ce comportement, il convient d'ajouter dans le fichier **/etc/pam.d/system-auth** la ligne suivante :

```
auth required pam_tally.so onerr=fail deny=3 unlock_time=300
```

Dans ce cas, après trois tentatives infructueuses de connexion, le compte sera bloquer pendant 5 minutes.

Configuration

Certains modules de PAM peuvent être configurés grâce aux fichiers présents dans le répertoire **/etc/security** :

```
root@debian8:~# ls /etc/security
access.conf  group.conf  limits.conf  limits.d  namespace.conf    namespace.d  namespace.init  opasswd
pam_env.conf  sepermit.conf  time.conf
```

Parmi les fichiers cités on note ceux qui peuvent être utilisés pour configurer les modules suivants :

Fichier/*Répertoire*	**Description**
access.conf	Utilisé par le module pam_access.so
group.conf	Utilisés par le module pam_group.so
limits.conf	Utilisé par le module pam_limits.so
pam_env.conf	Utilisé par le module pam_env.so
time.conf	Utilisé par le module pam_time.so

A faire : Passez en revue chacun de ces fichiers.

Dernièrement, PAM propose une solution pour toutes les applications ne disposant pas de leurs propres fichiers de configuration PAM. Cette solution

prend la forme du fichier **/etc/pam.d/other** :

```
root@debian8:~# cat /etc/pam.d/other
#
# /etc/pam.d/other - specify the PAM fallback behaviour
#
# Note that this file is used for any unspecified service; for example
#if /etc/pam.d/cron  specifies no session modules but cron calls
#pam_open_session, the session module out of /etc/pam.d/other is
#used.  If you really want nothing to happen then use pam_permit.so or
#pam_deny.so as appropriate.

# We fall back to the system default in /etc/pam.d/common-*
#

@include common-auth
@include common-account
@include common-password
@include common-session
```
