PAM

Pluggable Authentication Modules

Linux-PAM est un système de librairies qui manipulent les tâches d'authentification des services dans le système. La librairie fournis une API qui fournis un système d'authentification déféré.

La fonctionnalité principal de **PAM** est la nature de l'authentification qui est configurable dynamiquement. L'administrateur est libre de choisir comment les applications vont authentifier un utilisateur. La configuration individuelle peut être définie dans le fichier de configuration **/etc/pam.conf**. Alternativement, la configuration peut être définie dans des fichiers individuels dans **/etc/pam.d/**.

Linux-PAM sépare les tâches d'authentification en 4 groupes indépendants : gestion des comptes ; gestion de l'authentification ; gestion des mots de passe ; gestion de session.

Le mécanisme **account** fournit une seule primitive : il vérifie si le compte demandé est disponible (si le compte n'est pas arrivé à expiration, si l'utilisateur est autorisé à se connecter à cette heure de la journée, etc.).

Le mécanisme **auth** fournit deux primitives ; il assure l'authentification réelle, éventuellement en demandant et en vérifiant un mot de passe, et il définit des "certificats d'identité" tels que l'appartenance à un groupe ou des "tickets" kerberos.

Le mécanisme **password** fournit une seule primitive : il permet de mettre à jour le jeton d'authentification (en général un mot de passe), soit parce qu'il a expiré, soit parce que l'utilisateur souhaite le modifier.

Le mécanisme **session** fournit deux primitives : mise en place et fermeture de la session. Il est activé une fois qu'un utilisateur a été autorisé afin de lui permettre d'utiliser son compte. Il lui fournit certaines ressources et certains services, par exemple en montant son répertoire personnel, en rendant sa boîte aux lettres disponible, en lançant un agent ssh, etc.

Syntaxe du fichier de configuration

Le format de chaque rêgle est une collection de tokens séparés par un espace :

service type control module-path modeul-arguments

service est typiquement le nom de l'application. Le service '**other**' est réservé pour définir les règles par défaut.

type est le gestionnaire auquel la règle correspond (account, auth, password, session). Si le type est précédé par un '**-**', la librairie PAM ne loggera rien s'il n'est pas possible de charger le module s'il n'est pas présent dans le système.

control indique le comportement de PAM-API si le module échoue dans sa tâche d'authentification. Il y'a 2 types de syntaxe : un simple mot clé, et une paire [value=action].

Les valeurs possibles

required Retourne un échec mais seulement après que les modules stackés restant aient été invoqués.

requisite identique à **required**, mais retourne le contrôle. La valeur de retour est associée avec le premier **required** ou **requisite** qui a échoué. Ce flag peut être utilisé comme protection contre la possibilité qu'un utilisateur ait l'opportunité d'entrer un mot de passe dans un environnement non sécurisé.

sufficient Le succès d'un tel module est suffisant pour satisfaire aux requis de la pile de module (si un précédent **required** a échoué, celui-ci est ignoré). Si le module réussit, retourne le succès immédiatement sans essayer les autres modules.

optional Le succès ou l'échec n'est pas important si c'est le seul module dans la pile.

include inclue toutes les lignes d'un type donné depuis un fichier de configuration spécifié en argument.

substack Identique à **include** mais diffère dans les actions **done** et **die** dans une sous pile qui n'ignorent pas le reste de la pile, mais seulement sur la sous pile.

[value=action1 value2=action2 ...] où **valueN** correspond au code de retour de la fonction invoquée dans le module. peut être :**success, open_err, symbol_err, service_err, system_err, buf_err, perm_denied, auth_err, cred_insufficient, authinfo_unavail, user_unknown, maxtries, new_authtok_reqd, acct_expired, session_err, cred_unavail, cred_expired, cred_err, no_module_data, conv_err, authtok_err, authtok_recover_err, authtok_lock_busy, authtok_disable_aging, try_again, ignore, abort, authtok_expired, module_unknown, bad_item, conv_again, incomplete, et default**.

default implique toutes les valueN non mentionnées explicitement.

actionN prend une des valeurs suivante :

ignore Le status de retour ne contribut pas à retourner un code que l'application obtient.

bad le code de retour indique que le module a échoué. Si c'est le premier module de la pile à échouer, ce status est utilisé pour toute la pile.

die Idem à bad mais la pile de termine immédiatement.

ok Ce code devrait contribuer directement au code de retour de toute la pile.

done Idem à ok mais termine immédiatement la pile

N (entier non signé) Idem à ok mais saute immédiatement au N-ème module suivant dans la pile

reset Efface toute trace de l'état du module et commence avec le module suivant.

required [success=ok new_authtok_reqd=ok ignore=ignore default=bad]

requisite [success=ok new_authtok_reqd=ok ignore=ignore default=die]

sufficient [success=done new_authtok_reqd=done default=ignore]

optional [success=done new_authtok_reqd=ok default=ignore]

module-path est soit le nom de fichier utilisé par l'application (comence par un '/'), soit un chemin relatif à l'emplacement des modules : **/lib/security** ou **/lib64/security/**

module-arguments est une liste de tokens séparés par un espace ou entre [] si des arguments contiennent des espaces, qui peut être utilisé pour modifier le fonctionnement d'un PAM.

Exemple de configuration

Un système considéré sécurisé a une entrée other raisonnablement sécurisé :
other auth required pam_deny.so
other account required pam_deny.so
other password required pam_deny.so
other session required pam_deny.so
l'exemple suivant fournit des alertes utiles pour les administrateurs :
other auth required pam_warn.so
other password required pam_warn.so
exemple classique sur les machines moins sensible dans /etc/pam.d/other :
auth required pam_unix.so
account required pam_unix.so
password required pam_unix.so
session required pam_unix.so
Il est conseillé d'avoir une entrée other sécurisée :
auth required pam_deny.so
auth required pam_warn.so
account required pam_deny.so
account required pam_warn.so
password required pam_deny.so

password required pam_warn.so
session required pam_deny.so
session required pam_warn.so