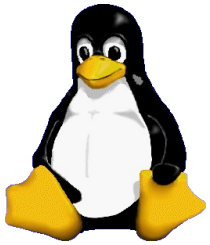


Sécurité GNU/Linux

PAM

By sharevb

Sommaire

I.Introduction....1
II.Fichiers de configuration....1
III.Format des lignes de configuration /etc/pam.d....2
a)Type de module : module-type....2
b)Importance du module : control-flag....2
c)Le chemin et le nom du module : module-path....2
d)Le format de fichier pour /etc/pam.conf....2
1Nom du service : service-name....3
IV.Modules usuels....3
V.Fonctionnement dans les programmes....4
VI.Implémentation dans les programmes C....5
VII.Bibliographie....6

I. Introduction

PAM est un système permettant gérant l'authentification pour les applications individuellement. On peut régler individuellement le type d'authentification pour chaque application. 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/`. A ces fichiers, s'ajoute le fichier `/etc/pam.conf`.

Un des principaux avantages de PAM est que l'authentification est centralisé pour toutes les applications d'un système. Ainsi, il n'est plus nécessaire de réécrire à chaque fois tous les programmes lorsqu'une nouvelle méthode d'authentification apparaît. Il suffit juste de se lier à PAM et laisser la configuration de celui-ci à l'administrateur du systèmes. Ensuite, dans l'application on obtient simplement une résultat « oui/non » pour l'authentification configurée.

PAM sert à faire l'interface entre le moyen d'authentification et l'application demandeuse.

II. Fichiers de configuration

Il existe deux types de configuration de PAM :

- la configuration par service dans des fichiers de même que le service dans `/etc/pam.d/`
- la configuration par service dans le fichier `/etc/pam.conf`
- le fichier `/etc/pam.d/other` permet de configurer les actions par défaut si aucun fichier de configuration n'est présent pour un service

On configure généralement au minimum 4 lignes par service (les 4 types de modules). Ceci est nécessaire pour une authentification totale.

III. Format des lignes de configuration /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`

a) Type de module : module-type

L'authentification est divisée en quatre catégories (`module-type`) :

- `account` : ce service est chargé de vérifier si l'utilisateur qui veut se connecter en a le droit (liste, expiration de compte...) d'après son nom
- `auth`(entication) : ce service se charge de vérifier que l'utilisateur qui veut se connecter est bien celui qu'il prétend être. Ceci passe par tout moyen d'authentification comme :
  - un mot de passe
  - une phrase de passe
  - un accès matériel type carte à puce
  - un accès matériel type biométrique
- `password` : ce service permet d'utilisateur à modifier ce qui lui sert pour authentification (mot de passe, carte à puce...)
- `session` : ce service permet d'effectuer des actions avant/après l'accès au service par l'utilisateur (logging, montage, script de de connexion, script de déconnexion...)

b) Importance du module : control-flag

On peut choisir le type de réaction au succès ou à l'échec du module parmi :

- `required` : indique que le succès d'un des modules marqués « `required` » est requis. Cela signifie que le premier module marqués « `required` » qui indique un succès annule l'utilisation des modules qui suivent dans la configuration
- `requisite` : indique que le succès de tous les modules marqués « `requisite` » est requis. Cela signifie que tous les modules doivent marqués « `requisite` » doivent indiquer une succès pour que l'authentification soit validée.
- `sufficient` : indique que le succès suffit à arrêter d'exécuter les modules du même type suivants celui-ci
- `optional` : indique de ne pas tenir compte du résultat de ce module

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

`module-path` indique le nom du module (.so) relativement à `/lib/security` ou `/usr/lib/security` si ce nom ne commence pas par `/`.

d) Le format de fichier pour /etc/pam.conf

Le format de chaque ligne des fichiers de configuration dans les fichiers de `/etc/pam.conf` est :
`service-name  module-type  control-flag  module-path  arguments`

1 Nom du service : service-name

`service-name :` indique le nom du service pour lequel la configuration s'applique.
Pour les autres parties voir ci-dessus.

IV. Modules usuels

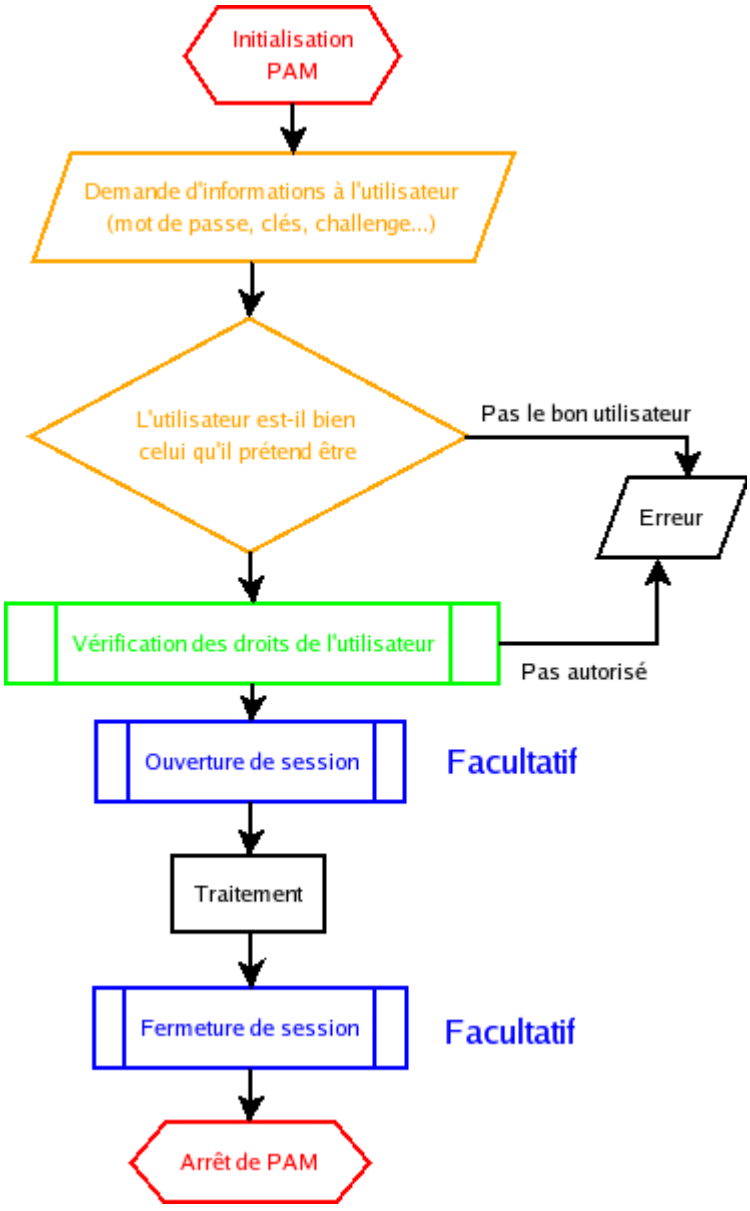
Voici quelques modules :

Module	***Arguments***	***Description***
pam_cracklib.so	*Vérifie la robustesse des mots de passe (modtype : password)*	
	retry=*n*	[fac] Nombre d'essais infructueux autorisés
	minlen=*n*	[fac] Taille minimum du mot de passe
pam_deny.so	*Interdit le type du module (modtype : n'importe) : authentification, connexion, changement de mot de passe...*	
pam_listfile.so	*Indique une liste d'utilisateurs autorisés ou interdits à se connecter (modtype: auth) à partir d'un fichier.*	
	onerr=succeed|fail	[obl] indique si en cas d'erreur (fichier inexistant), on renvoie succès ou échec
	sense=allow|deny	[obl] indique si la liste est autorisante ou interdisante
	file=*filename*	[obl] indique le fichier contenant la liste d'utilisateurs (un par ligne)
pam_motd.so	*Affiche le contenu de /etc/motd à la connexion (modtype: session)*	
	motd=*filename*	Utilise le fichier *filename* à la place de motd
pam_unix.so	*Utilise l'authentification UNIX /etc/passwd (modtype: tous)*	
pam_warn.so	*Loggue les connexions (modtype : auth, password)*	
pam_loginuid.so	*Indique de changer l'uid du propriétaire du service vers celui de l'utilisateur authentifié.*	
pam_nologin.so	*Indique que le compte n'existe pas (modtype: auth,account)*	
pam_stack.so	*Indique un fichier de configuration PAM à inclure au fichier en cours (modtype : tous)*	
	service=*nom_service*	Nom du service du fichier de configuration PAM à inclure

Module	*Arguments*	*Description*
pam_limits.so	*Utilise le fichier `/etc/security/limits.conf` et le support noyau de la mesure des ressources à l'authentification (modtype : session)*	
	conf=*filename*	Indique le chemin du fichier *limits.conf*
pam_time.so	*Utilise le fichier `/etc/security/time.conf` pour indique des restrictions horaires à l'authentification(modtype : account)*	

V. Fonctionnement dans les programmes

Voici comment se passe l'authentification PAM dans les programmes :



- *pam_start* initialise PAM pour authentifier suivant la configuration du service dont le nom lui est passé en paramètre. Si l'on connaît déjà le nom de l'utilisateur que l'on va authentifier on peut aussi le passer en paramètre.
- *pam_authenticate* permet de vérifier l'identité d'un utilisateur (`auth`)
- *pam_acct_mgmt* permet de vérifier les droits de l'utilisateur dont on vient de vérifier l'identité (`account`)
- *pam_open_session* et *pam_close_session* permettent de signaler une ouverture et une fermeture de session (`session`)
- *pam_end* termine PAM
- *pam_chauthtok* permet de changer les informations d'authentification de l'utilisateur authentifié suivant la configuration du service (`password`)

VI. Implémentation dans les programmes C

Les programmes C Unix sont au sens de PAM : les services. Le programmeur a donc les choses suivantes à faire :

- inclure un fichier `/etc/pam.d/`*`nom_programme`* contenant une configuration par défaut pour votre programmes avec au moins un `required` pour `auth` et `account`.

Voici un exemple de programme C qui utilise PAM. Il provient de l'aide de PAM sur le site de kernel.org :

- Dans un fichier `/etc/pam.d/`*`nom_service`* :

```
# configuration pour nom_service
nom_service   auth       required     pam_unix_auth.so
nom_service   account    required     pam_unix_acct.so
```

- Dans une fichier *`nom_service`*`.c` :

```
/*
  This program was contributed by Shane Watts
  [modifications by AGM]
 */

//pour PAM
#include <security/pam_appl.h>
#include <security/pam_misc.h>

//pour printf
#include <stdio.h>

//indique une fonction de conversation par défaut
static struct pam_conv conv = {
    misc_conv,
    NULL
};

int main(int argc, char *argv[])
{
    //handle de l'instance de PAM
    pam_handle_t *pamh=NULL;
```

```
    //résultat de l'authentification
    int retval;
    //nom de l'utilisateur à authentifier
    const char *user="nobody";

    //si un nom d'utilisateur est spécifié
    if(argc == 2) {
        user = argv[1];
    }

    //controle du nombre d'argument du programme
    if(argc > 2) {
        fprintf(stderr, "Usage: nom_service [username]\n");
        exit(1);
    }

    //initialisation de PAM pour l'utilisateur user
    retval = pam_start("nom_service", user, &conv, &pamh);

    //on essaie d'authentifier l'utilisateur passé en argument (ou nobody)
    if (retval == PAM_SUCCESS)
        retval = pam_authenticate(pamh, 0);    /* is user really user? */

    //on vérifie les droits de l'utilisateur pour notre programme
    if (retval == PAM_SUCCESS)
        retval = pam_acct_mgmt(pamh, 0);

    /* A partir de maintenant, on va pouvoir savoir
                        si l'utilisateur est autorisé ou pas */

    //c'est le résultat de  pam_acct_mgmt qui indique l'authentification
    if (retval == PAM_SUCCESS) {
        fprintf(stdout, "OK\n");
    } else {
        fprintf(stdout, "Pas OK\n");
    }

    //on libère PAM
    if (pam_end(pamh,retval) != PAM_SUCCESS) {     /* close Linux-PAM */
        pamh = NULL;
        fprintf(stderr, "check_user: failed to release authenticator\n");
        exit(1);
    }

    return ( retval == PAM_SUCCESS ? 0:1 );       /* indicate success */
}
```

VII. Bibliographie

ftp://ftp.kernel.org/pub/linux/libs/pam/ : bibliothèques et programmes PAM pour Linux.
ftp://ftp.kernel.org/pub/linux/libs/pam/Linux-PAM-html/ :

- The Linux-PAM System Administrators' Guide <ftp://ftp.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html> par Andrew G. Morgan <morgan@transmeta.com>
- The Linux-PAM Application Developers' Guide <ftp://ftp.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_appl.html> par Andrew G. Morgan <morgan@transmeta.com>
- The Linux-PAM Module Writers' Guide <ftp://ftp.kernel.org/pub/linux/libs/pam/Linux-

PAM-html/pam_modules.html>
par Andrew G. Morgan <morgan@transmeta.com>
Unified Login With Pluggable Authentication Modules
<http://www.pilgrim.umass.edu/pub/osf_dce/RFC/rfc86.0.txt> :
Open Software Foundation Request For Comments 86.0
par V. Samar and R. Schemers (SunSoft)

PAM

Configuration de **PAM**