LECOEUR Elise IUP ASR2 12/04/05

PAM - Modules d'authentifications enfichables.

(De l'anglais Pluggable Authentication Modules).

Définition

C'est un projet qui permet, entre autres, de décentraliser l'authentification, de s'authentifier autrement qu'avec les classiques /etc/passwd/group/shadow : sans mot de passe, avec votre voix, votre clef USB, vos empreintes digitales, sur une base SQL, sur un annuaire LDAP, etc. Il permet aussi de définir toutes sortes de contraintes (comme par exemple l'interdiction d'accès à une personne ou un groupe à une certaine heure), ou encore de monter un répertoire distant SMB sur votre home local, etc. Les exemples sont nombreux.

Les programmes qui donnent aux utilisateurs un accès à des privilèges, doivent être capables de les identifier. Lorsque vous vous connectez sur le système, vous indiquez votre nom et votre mot de passe. Le processus de connexion vérifie que vous êtes bien la personne que vous prétendez être. Il existe d'autres formes d'identification que l'utilisation des mots de passe, qui peuvent d'ailleurs êtres stockés sous différentes formes.

PAM permet à l'administrateur de définir une politique d'identification sans avoir à recompiler les programmes concernés. Le contrôle des modules se fait à l'aide d'un fichier de configuration que vous devez éditer.

Avantages des PAM

PAM offre entre autres les avantages suivants :

- Il fournit un système d'authentification commun qui peut être utilisé avec un vaste éventail d'applications.
- Il offre un haut degré de flexibilité et de contrôle en ce qui concerne l'authentification aussi bien au niveau de l'administrateur système qu'au niveau du développeur d'applications.
- Il permet aux développeurs d'applications de concevoir des programmes sans avoir à créer leur propre système d'authentification.

Fichiers de configuration PAM

Le répertoire /etc/pam.d est utilisé pour configurer toutes les applications PAM (/etc/pam.conf dans les anciennes versions, le fichier pam.conf est toujours lu s'il n'y a pas d'entrée /etc/pam.d/).

Chaque application ou *service* prenant en charge les PAM possède un fichier dans le répertoire /etc/pam.d/. Chacun de ces fichiers est nommé en fonction du service dont il contrôle l'accès.

Il appartient au programme prenant en charge les PAM de définir le nom de ses services et d'installer son fichier de configuration PAM dans le répertoire /etc/pam.d/. Par exemple, le programme login attribue le nom login à son service et installe le fichier de configuration PAM /etc/pam.d/login.

Chaque fichier de configuration PAM comprend un ensemble de directives établies selon format suivant :

<Interface du module> <indicateurs de contrôle> <nom du module> <arguments des modules>

Interface du module PAM

Quatre types de modules sont définis par la norme PAM.

- Les modules auth assurent l'authentification réelle, éventuellement en demandant et en vérifiant un mot de passe, et ils définissent des "certificats d'identité" tels que l'appartenance à un groupe ou des "tickets" kerberos.
- Les modules account vérifient si l'authentification est autorisée (si le compte n'est pas arrivé à expiration, si l'utilisateur est autorisé à se connecter à cette heure de la journée, etc.).
- Les modules password sont utilisés pour définir des mots de passe.
- Les modules session sont utilisés une fois qu'un utilisateur a été authentifié pour lui permettre d'utiliser son compte, éventuellement en montant le répertoire personnel de l'utilisateur ou en rendant sa boîte aux lettres disponible.

Indicateurs de contrôle

Lorsqu'ils sont appelés, tous les modules PAM donnent un résultat indiquant soit la réussite, soit l'échec. Les indicateurs de contrôle indiquent aux PAM comment traiter ce résultat. Étant donné que les modules peuvent être empilés dans un ordre bien précis, les indicateurs de contrôle décident de l'importance de la réussite ou de l'échec d'un module spécifique par rapport au but général d'authentification d'un utilisateur pour un service donné.

Il existe quatre types d'indicateurs de contrôle prédéfinis, à savoir :

- required — Le module doit être vérifié avec succès pour que l'authentification puise se poursuivre. Si la vérification d'un module portant l'indication required échoue, l'utilisateur n'en est pas averti tant que tous les modules associés à cette interface n'ont pas été vérifiés.

- requisite — Le module doit être vérifié avec succès pour que l'authentification puisse se poursuivre. Cependant, si la vérification d'un module requisite échoue, l'utilisateur en est averti immédiatement par le biais d'un message lui indiquant l'échec du premier module required *ou* requisite.

- sufficient — En cas d'échec, les vérifications de modules sont ignorées. Toutefois, si la vérification d'un module portant l'indication sufficient est réussie *et* qu'aucun module précédent portant l'indicateur required n'a échoué, aucun autre module de ce type n'est nécessaire et l'utilisateur sera authentifié auprès du service.

- optional — Les vérifications de modules sont ignorées. Un module portant l'indication optional devient nécessaire pour la réussite d'une authentification lorsque aucun autre module ne fait référence à cette interface.

Nom du module

Le nom de module donne à PAM le nom du module enfichable contenant l'interface de module spécifiée. Toutefois, depuis l'arrivée de systèmes multilib qui stockent des modules PAM de 64-octets dans le répertoire /lib64/security/, le nom du répertoire est omis vu que les applications sont liées à la version appropriée de libpam, qui peut trouver la correcte version du module.

Arguments des modules

PAM utilise des arguments pour transmettre des informations à un module enfichable lors du processus d'authentification de certains modules.

Par exemple, le module pam_userdb.so utilise des indications secrètes stockées dans un fichier de la base de données Berkeley pour authentifier les utilisateurs. La base de données Berkeley est une base de données Open Source intégrée dans de nombreuses applications. Le module

nécessite un argument db pour spécifier à la base de données Berkeley quelle base de données précise doit être utilisée pour le service demandé.

Une ligne pam_userdb.so typique d'un fichier de configuration PAM ressemble à l'extrait suivant :

```
auth      required  pam_userdb.so db=<Chemin du fichier>
```

Dans l'exemple précédent, remplacez *<chemin du fichier>* par le chemin d'accès complet au fichier de la base de données Berkeley DB.

Exemples de fichiers de configuration PAM

Ci-dessous figure un exemple de fichier de configuration PAM :

```
#%PAM-1.0
auth      required  pam_securetty.so
auth      required  pam_unix.so shadow nullok
auth      required  pam_nologin.so
account   required  pam_unix.so
password  required  pam_cracklib.so retry=3
password  required  pam_unix.so shadow nullok use_authtok
session   required  pam_unix.so
```

La première ligne est un commentaire, comme l'indique le caractère dièse (#) placé au début de la ligne.

Les lignes deux à quatre empilent trois modules à utiliser pour l'authentification de connexion.

```
auth      required  pam_securetty.so
```

Ce module sert à s'assurer que, *si* l'utilisateur essaie de se connecter en tant que super-utilisateur (ou root), le terminal tty sur lequel il se connecte fait bien partie de la liste se trouvant dans le fichier /etc/securetty, *si* ce fichier existe.

```
auth      required  pam_unix.so shadow nullok
```

Ce module invite l'utilisateur à fournir un mot de passe, puis le vérifie à l'aide des informations stockées dans /etc/passwd et vérifie s'il existe dans /etc/shadow. Le module pam_unix.so détecte et utilise automatiquement les mots de passe masqués pour authentifier les utilisateurs.

L'argument nullok donne l'instruction au module pam_unix.so d'autoriser un mot de passe vide.

```
auth      required  pam_nologin.so
```

Il s'agit de la dernière phase du processus d'authentification. Elle vérifie l'existence du fichier /etc/nologin. Si nologin existe et que l'utilisateur n'est pas un super-utilisateur (ou root), l'authentification échoue.

Ce module effectue toute vérification de compte lorsque cela est nécessaire. Par exemple, si des mots de passe masqués ont été activés, l'élément compte du module pam_unix.so vérifiera si le compte a expiré ou si l'utilisateur a changé son mot de passe pendant le délai de grâce alloué.

```
password  required  pam_cracklib.so retry=3
```

Si un mot de passe n'est plus valable, l'élément mot de passe du module pam_cracklib.so invite l'utilisateur à en fournir un nouveau. Il vérifie ensuite le mot de passe créé afin de déterminer s'il peut être facilement retrouvé par un programme de craquage de mots de passe basé sur des dictionnaires. Si le test du mot de passe échoue, le programme donne à l'utilisateur deux autres possibilités de créer un mot de passe sûr, comme il l'est précisé dans l'argument retry=3.

```
password  required  pam_unix.so shadow nullok use_authtok
```

Cette ligne spécifie que, si le programme change le mot de passe de l'utilisateur, il doit le faire en utilisant l'élément password du module pam_unix.so. Ceci se produit uniquement si la partie auth du module pam_unix.so détermine que le mot de passe doit être changé.

L'argument shadow donne l'instruction au module de créer des mots de passe masqués lors de la mise à jour du mot de passe d'un utilisateur.

L'argument nullok donne l'instruction au module d'autoriser l'utilisateur à changer son mot de passe *à partir d'un* mot de passe vide ; sinon, un mot de passe non-valide est traité comme un verrouillage de compte.

Le dernier argument de cette ligne, use_authtok, est un exemple illustrant bien l'importance de l'ordre lors de l'empilage de modules PAM. Cet argument indique au module de ne pas demander à l'utilisateur un nouveau mot de passe. Au lieu de cela, il accepte tous les mots de passe qui ayant été enregistrés dans le précédent module de mots de passe. De cette façon, tous les nouveaux mots de passe doivent passer le test de sécurité pam_cracklib.so avant d'être acceptés.

```
session required pam_unix.so
```

La dernière ligne spécifie que l'élément session du module pam_unix.so gérera la session. Ce module enregistre dans /var/log/messages le nom d'utilisateur ainsi que le type de service au début et à la fin de chaque session. Il peut être complété en l'empilant avec d'autres modules de session si vous désirez obtenir une fonctionnalité supplémentaire.

Sites Web utiles

- http://www.kernel.org/pub/linux/libs/pam/ — Le site Web de distribution principal pour le projet Linux PAM contenant des informations sur différents modules PAM, un Forum Aux Questions (FAQ) ainsi que de la documentation supplémentaire sur PAM.
- http://www.linux-kheops.com/doc/redhat70/ref-guide-fr/s1-sysadmin-auth.html
- http://www.startcom.org/docs/fr/Guide%20de%20reference%20StartCom%20Enterprise%20Linux%203.0.x/ch-pam.html