Guide pratique de l'authentification des utilisateurs

Version française de : **User Authentication HOWTO**

Peter Hernberg

Raphaël Semeteys

<raphael semeteys CHEZ wanadoo POINT fr>

Adaptation française

Version : 0.9.fr.0.1

17 février 2005

Historique des versions		
Version 0.9.fr.0.1	2005-02-17	rcs
Adaptation française. Passage de SGML DocBook à XML DocBook.		
Version 0.9	2004-04-03	fl
Mise à jour des liens externes. *Updated external links*		
Version 0.8	2003-02-20	fl
Modifications de langage et diverses corrections mineures. *Language changes, various small fixes*		
Version 0.5	2000-05-15	ph
Ajout d'une section sur la sécurisation de PAM et d'une autre sur les ressources. *Added section on securing pam, added resources section*		
Version 0.1	2000-05-02	ph
Version initiale. *Initial version*		

Résumé

Ce guide pratique explique comment l'information sur les utilisateurs et les groupes est stockée, comment les utilisateurs sont authentifiés sur un système Linux (PAM) et comment sécuriser l'authentification des utilisateurs sur votre système.

Table des matières

Introduction

Comment ce document a vu le jour

Alors que je tentais d'ajouter des services réseaux (pour la plupart inutiles :) à mon réseau personnel, j'ai été confronté à de nombreuses reprises au problème de l'authentification. J'ai donc décidé de comprendre comment l'authentification fonctionne sur Linux, d'écrire un guide pratique et d'en faire mon projet de fin d'études. J'espère que ce document vous aidera à comprendre cet aspect de l'administration système souvent oublié mais néanmoins très important.

Nouvelles versions de ce document

Vous trouverez la plus récente version française de ce document à l'adresse : http://www.traduc.org/docs/howto/lecture/User-Authentication-HOWTO.html.

La plus récente version de la version originale de ce document est disponible à l'adresse : http://www.linuxdoc.org.

Commentaires et corrections

Merci de faire parvenir en anglais à l'auteur vos questions et commentaires relatifs à la version originale de ce document à l'adresse <petehern CHEZ yahoo POINT com>.

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à <commentaires CHEZ traduc POINT org> en précisant sont titre, sa date et sa version.

Droits d'utilisation

Remerciements

Merci à ma famille de me supporter depuis 18 ans. Merci aux gens de Debian de fournir une distribution si sympa avec laquelle je peux jouer. Merci à CGR de me payer à être un *geek*. Merci à Sandy Harris pour ses utiles suggestions. Et finalement j'aimerai remercier les fabricants de nouilles chinoises, car je ne sais comment je vivrais sans.

Pré requis du lecteur

Je considère dans la suite de ce document que le lecteur est à l'aise avec tout ce qui touche à l'exécution en ligne de commandes et l'édition de fichiers de configuration.

Comment l'information des utilisateurs est stockée sur votre système

`/etc/passwd`

Dans pratiquement toutes les distributions Linux (ainsi que les versions commerciales d'Unix), l'information sur les utilisateurs est stockée dans `/etc/passwd`, un fichier texte contenant le *login* de chaque utilisateur, son mot de passe encrypté, un identifiant numérique unique d'utilisateur (appelé l'uid), un identifiant numérique unique de groupe (appelé le gid), un champ optionnel de commentaire (contenant généralement des choses comme le nom, le numéro de téléphone, etc.), le répertoire personnel et le shell préféré. Une entrée typique de `/etc/passwd` ressemble à ceci :

```
pete:K3xcO1Qnx8LFN:1000:1000:Peter Hernberg,,,1-800-FOOBAR:/home/pete:/bin/bash
```

Comme vous pouvez le voir, c'est plutôt simple. Chaque entrée contient les six champs décrits plus haut, avec ":" comme séparateur. Si l'authentification des utilisateurs était aussi simple que cela, il n'y aurait pas besoin de ce guide.

Mots de passes Shadow

En observant votre `/etc/passwd`, il y a des chances que vous ayez vu quelque chose comme ceci :

```
pete:x:1000:1000:Peter Hernberg,,,1-800-FOOBAR:/home/pete:/bin/bash
```

Où est donc passé le mot de passe encrypté ? Avant de répondre, une petite explication s'impose.

Le ficher `/etc/passwd`, qui contient l'information sur tous les utilisateurs y compris leurs mots de passes encryptés, est consultable par tous, ce qui rend capable n'importe quel utilisateur d'obtenir le mot de passe encrypté de n'importe qui sur le système. Bien que les mots de passes soient encryptés, des programmes de piratage sont largement disponibles. Les mots de passes Shadow ont été développés afin de combattre cette menace grandissante.

Lorsque les mots de passes Shadow sont activés sur un système, le champ correspondant au mot de passe dans `/etc/passwd` est remplacé par un "x" et le véritable mot de passe encrypté est stocké dans `/etc/shadow`. Comme `/etc/shadow` est uniquement consultable par l'utilisateur root, les utilisateurs malicieux ne peuvent pas pirater les mots de passes des autres utilisateurs. Chaque entrée de `/etc/shadow` contient le login de l'utilisateur, son mot de passe crypté et un certain nombre de champs liés à l'expiration du mot de passe. Une entrée typique ressemble à ceci :

```
pete:/3GJllg1o4152:11009:0:99999:7:::
```

`/etc/group` et `/etc/gshadow`

L'information sur les groupes est stockée dans `/etc/group`. Le format est similaire à celui de `/etc/passwd`, avec chaque entrée contenant des champs pour le nom du groupe, le mot de passe, l'identifiant numérique (gid) et une liste de membres du groupe séparés par des virgules. Une entrée typique de `/etc/group` ressemble à ceci :

```
pasta:x:103:spagetti,fettucini,linguine,vermicelli
```

Comme vous pouvez le voir, il est également possible d'utiliser des mots de passes Shadow. Bien que les groupes ne disposent pratiquement jamais de mots de passes propres, cela ne coûte rien de stocker les mots de passes Shadow des groupes dans `/etc/gshadow`.

Mots de passes encryptés par MD5

Traditionnellement, les mots de passes Unix sont encryptés avec la fonction standard crypt(). (Pour plus d'information sur la fonction crypt(), se reporter à la page man crypt(3).) Au fur et à mesure que les ordinateurs sont devenus plus puissants, les mots de passes encryptés sont devenus plus faciles à pirater. Et avec l'émergence d'Internet, des outils de piratage utilisant la puissance de traitement de plusieurs ordinateurs en réseau ont fait leur apparition. De nombreuses "nouvelles" distributions offrent la possibilité d'encrypter les mots de passes avec l'algorithme de hashage MD5, qui est plus puissant. (Pour plus d'information sur l'algorithme de hashage MD5, se reporter au RFC 1321.) Bien que les mots de passes MD5 n'éliminent pas la menace du piratage, ils rendent cette tâche beaucoup plus difficile.

Organiser le désordre

Comme vous pouvez le voir, il existe plusieurs manières différentes de stocker l'information d'authentification des utilisateurs sur votre système (mots de passes Shadow sans encryptage MD5, mots de passes `/etc/passwd` avec encryptage MD5, etc.). Comment des programmes comme login et su savent-ils comment vérifier votre mot de passe ? Pire encore, et si vous vouliez modifier la manière dont les mots de passes sont stockés dans votre système ? Comment les programmes nécessitant votre mot de passe sauront-ils que les mots de passes sont désormais stockés différemment ? La solution est PAM.

PAM (Modules enfichables d'authentification ou *Pluggable Authentication Modules*)

Les modules enfichables d'authentification sont au cœur de toute distribution moderne de Linux.

Le pourquoi

Au tout début de Linux, si un programme comme su, passwd, login ou xlock, avait besoin d'authentifier un utilisateur, il lisait simplement l'information nécessaire dans `/etc/passwd`. Pour modifier le mot de passe il éditait simplement `/etc/passwd`. Cette méthode simple mais maladroite posait de nombreux problèmes aux administrateurs systèmes et aux développeurs d'applications. Une fois MD5 et les mots de passes Shadow devenus incroyablement populaires, chaque programme nécessitant l'authentification d'un utilisateur devait savoir récupérer l'information appropriée, tout en étant capable de gérer de nombreuses configurations différentes. Lorsque vous vouliez modifier la configuration de l'authentification des utilisateurs, tous ces programmes devaient être recompilés. PAM élimine ce désordre en permettant aux programmes d'authentifier les utilisateurs de manière transparente, indépendamment de la méthode de stockage de l'information.

Le quoi

Voici un extrait du Guide administrateur système Linux-PAM (version anglaise) : "Le projet Linux-PAM a pour objet de séparer le développement de logiciels d'autorisation de celui de méthodes d'authentification sécurisées. Ceci est réalisé en fournissant une bibliothèque de fonctions utilisables par une application pour demander qu'un utilisateur soit authentifié". Avec PAM, peu importe que votre mot de passe soit stocké dans `/etc/passwd` ou sur un serveur à Hong Kong. Lorsqu'un programme a besoin d'authentifier un utilisateur, PAM fournit une bibliothèque contenant les fonctions appropriées à la méthode d'authentification utilisée. Comme cette bibliothèque est chargée dynamiquement, le changement de méthode d'authentification peut être réalisé par simple modification d'un fichier de configuration.

La souplesse est un des plus grands atouts de PAM. PAM peut être utilisé, par exemple, pour refuser à certains programmes le droit d'authentifier les utilisateurs, pour n'autoriser que certains utilisateurs à être authentifiés, pour émettre des notifications lorsque certains programmes tentent de réaliser une authentification ou même pour priver tous les utilisateurs du droit de se connecter. La conception modulaire de PAM vous donne le contrôle total sur comment sont authentifiés les utilisateurs.

Les distributions qui supportent PAM

A peu près toutes les distributions supportent PAM depuis longtemps. En voici une liste non exhaustive :

- Redhat depuis la version 5.0
- Mandrake depuis la version 5.2
- Debian depuis la version 2.1 (support partiel dans la 2.1 -- support total depuis la 2.2)
- Caldera depuis la version 1.3
- Turbolinux depuis la version 3.6
- SuSE depuis la version 6.2

Cette liste est certainement incomplète et sans doute erronée. J'apprécierai que vous envoyiez toute correction ou ajout à cette liste à l'adresse <petehern CHEZ yahoo POINT com>.

Installer PAM

Installer PAM en partant de zéro est un long processus hors du périmètre de ce guide pratique. Si PAM n'est pas installé sur votre système, vous utilisez probablement une version si ancienne de votre distribution qu'il y a de nombreuses autres raisons de la mettre à jour. Si vous voulez vraiment faire l'installation vous-même, alors vous n'êtes certainement pas le genre de personne que je peux aider. Pour toutes ces raisons, je considèrerai que PAM est déjà installé sur votre système.

Le comment

Assez parlé, rentrons dans le vif du sujet.

Fichiers de configuration de PAM

Les fichiers de configuration de PAM sont stockés dans le répertoire `/etc/pam.d/`. (Si vous n'avez pas de répertoire `/etc/pam.d/`, ne vous en faites pas, je traiterai ce cas dans la section suivante.) Jetons-y un coup d'œil.

```
~$ cd /etc/pam.d
/etc/pam.d/$ ls
chfn  chsh    login   other   passwd  su      xlock
/etc/pam.d/$
```

Le nombre de fichiers que vous trouverez dans ce répertoire varie en fonction de ce qui est installé sur votre système. Quoi qu'il en soit, vous avez remarqué l'existence d'un fichier pour chaque programme de votre système qui authentifie les utilisateurs. Comme vous l'avez probablement deviné, chaque fichier contient la configuration PAM pour le programme dont il porte le nom (sauf pour le fichier `other` dont nous allons parler dans un petit moment). Regardons le fichier de configuration PAM pour login (j'ai condensé le fichier par souci de simplicité) :

```
/etc/pam.d/$ cat login
# Configuration PAM pour login
auth       requisite  pam_securetty.so
auth       required   pam_nologin.so
auth       required   pam_env.so
auth       required   pam_unix.so nullok
account    required   pam_unix.so
session    required   pam_unix.so
session    optional   pam_lastlog.so
password   required   pam_unix.so nullok obscure min=4 max=8
```

Mais avant d'analyser ce fichier, je dois mentionner une petite chose.

Remarque

Un petit pourcentage des lecteurs doit probablement se dire "Oh non ! Je n'ai pas de répertoire /etc/pam.d ! Votre liste affirme que ma distribution intègre PAM, mais je ne retrouve pas ce répertoire. Sans PAM, ma vie n'a plus de sens ! Que puis-je faire ?". Ne vous inquiétez pas, tout n'est pas perdu. Si vous savez que votre distribution intègre PAM mais que vous n'avez aucun répertoire `/etc/pam.d/`, alors votre configuration PAM est stockée dans `/etc/pam.conf`. Plutôt que d'être répartie sur plusieurs fichiers, l'intégralité de votre configuration PAM est stockée dans un fichier unique. Cela modifie légèrement la configuration de PAM mais les ajustements adéquats sont expliqués dans la section 3.3.4.

Syntaxe de configuration

Les fichiers de configuration PAM ont la syntaxe suivante :

```
type  contrôle  chemin-vers-le-module   arguments-du-module
```

En prenant comme exemple le fichier de configuration de login (voir plus haut), regardons de plus près cette syntaxe :

Eléments de configuration PAM

type

L'élément *type* précise à PAM quel type d'authentification utiliser pour un module donné. Les modules de même types peuvent être chaînés, imposant ainsi à l'utilisateur de satisfaire à de multiples exigences pour pouvoir être authentifié. PAM reconnaît quatre types :

account

Détermine si l'utilisateur est autorisé à accéder au service, si son mot de passe a expiré, etc.

auth

Détermine si l'utilisateur est bien celui qu'il prétend être, généralement via un mot de passe, mais éventuellement via un moyen plus sophistiqué comme la biométrie.

password

Fournit à l'utilisateur un mécanisme pour modifier son authentification. Une fois encore il s'agit généralement de son mot de passe.

session

Les actions à réaliser avant et/ou après l'authentification de l'utilisateur. Cela peut inclure des choses comme monter/démonter le répertoire personnel de l'utilisateur, journaliser la connexion/déconnexion et restreindre les services mis à disposition de l'utilisateur.

Dans le fichier de configuration de login, on trouve au moins une entrée pour chacun de ces types. Comme il s'agit du programme qui permet à l'utilisateur de se connecter, il est compréhensible qu'il nécessite d'accéder à tous les types d'authentification.

contrôle

L'élément *contrôle* indique à PAM ce qu'il faut faire lorsque l'authentification réalisée par ce module échoue. PAM reconnaît quatre types de contrôles :

requisite

Tout échec retourné par le module implique le refus immédiat de l'authentification.

required

L'échec implique également le refus de l'authentification, mais PAM passe quand même par tous les autres modules listés pour ce service avant de refuser définitivement l'authentification.

sufficient

Si le retour du module est un succès, PAM accordera l'authentification, même si un module obligatoire (contrôle *required*) précédent a échoué.

optional

Le fait que le module échoue ou non n'a d'impact uniquement dans le cas où il s'agit du seul module de ce type configuré pour le service.

Dans le fichier de configuration de login, on retrouve pratiquement tous les types de contrôles. `pam_unix.so` (le module principal d'authentification) est majoritaire parmi les modules requis (*required*), `pam_securetty.so` (qui s'assure que l'utilisateur se connecte) est le seul module obligatoire et `pam_lastlog.so` (qui récupère les informations sur la connexion le plus récente de l'utilisateur) est le seul module optionnel.

chemin-vers-le-module

Cet élément indique à PAM quel module utiliser et (éventuellement) où le trouver. La plupart des configurations contiennent uniquement le nom du module, comme c'est le cas dans notre fichier de configuration de login. Dans ce cas, PAM recherche les modules dans le répertoire par défaut des modules PAM, normalement `/usr/lib/security`. Cependant, si votre distribution Linux est conforme au standard FHS (*Filesystem Hierarchy Standard*), les modules PAM peuvent être trouvés dans `/lib/security`.

arguments-du-module

Il s'agit des arguments à transmettre au module. Chaque module possède ses propres arguments. Par exemple, dans notre fichier de configuration de login, l'argument "nullok" ("null ok") passé au module `pam_unix.so` indique qu'un mot de passe vide ("null") est acceptable ("ok").

Fichier de configuration `pam.conf`

Si votre configuration PAM est stockée dans `/etc/pam.conf` plutôt que dans `/etc/pam.d/`, les lignes de configuration sont un peu différentes. Plutôt que d'avoir un fichier de configuration par service, toutes les configurations sont stockées dans `/etc/pam.conf` avec le nom du service comme premier élément de la ligne. Par exemple, la ligne suivante dans `/etc/pam.d/login`:

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

deviendra celle-ci dans `/etc/pam.conf` :

```
login   auth     required    pam_unix.so nullok
```

A part cette différence mineure, le reste de la syntaxe de configuration PAM s'applique.

Obtenir de l'information complémentaire

Pour plus d'information concernant la configuration de PAM et la référence complète des modules PAM, consultez le Guide administrateur système Linux-PAM (version anglaise). Ce guide fait office de référence exhaustive et à jour au sujet de la configuration de PAM.

Sécuriser l'authentification des utilisateurs

De nombreuses distributions intègrent une authentification des utilisateurs qui n'est pas sécurisée de manière adéquate. Cette section présente certaines manières de sécuriser cette authentification sur votre système. Bien que faire ce qui est décrit plus bas rende votre système plus sécurisé, n'ayez cependant pas la naïveté de penser que cela vous rende pour autant invulnérable.

Un fichier `/etc/pam.d/other` sécurisé

Chaque fichier placé dans `/etc/pam.d/` contient la configuration relative à un service donné. L'exception à la règle est le fichier `/etc/pam.d/other`. Ce fichier contient la configuration à utiliser pour tout service ne disposant pas de son propre fichier de configuration. Par exemple, si le service (imaginaire) xyz tentait une authentification, PAM chercherait un fichier `/etc/pam.d/xyz`. N'en trouvant aucun, l'authentification pour xyz serait déterminée par le fichier `/etc/pam.d/other`. Comme `/etc/pam.d/other` est la configuration utilisée en dernier recours, il est important qu'elle soit sécurisée. Nous allons étudier deux configurations de `/etc/pam.d/other`, l'une quasi-paranoïaque et l'autre plus permissive.

Une configuration paranoïaque

Voici une configuration paranoïaque de `/etc/pam.d/other` :

```
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
```

Dans cette configuration, lorsqu'un service inconnu tente d'accéder à n'importe lequel des quatre types de configuration, PAM refuse d'abord l'authentification (via le module `pam_deny.so`) et journalise ensuite une alerte dans syslog (via le module `pam_warn.so`). A moins d'un bogue dans PAM, cette configuration est fortement sécurisée. Le seul problème est que cela peut causer des problèmes si jamais vous effacez par accident le fichier de configuration d'un autre service. Si votre fichier `/etc/pam.d/login` est effacé par erreur, plus personne ne pourra plus se connecter !

Une configuration plus permissive

Voici une configuration moins vicieuse :

```
auth            required    pam_unix.so
auth            required    pam_warn.so
account         required    pam_unix.so
account         required    pam_warn.so
password    required        pam_deny.so
password    required        pam_warn.so
session         required    pam_unix.so
session         required    pam_warn.so
```

Cette configuration autorisera un service inconnu à authentifier (via le module `pam_unix.so`), sans pour autant l'autoriser à modifier le mot de passe de l'utilisateur. Bien que cela permette l'authentification par des services inconnus, une alerte est générée dans le journal syslog à chaque fois qu'un tel service tente une authentification.

Choix de la configuration `/etc/pam.d/other`

A moins que vous ayez une très bonne raison de ne pas le faire, je vous recommande fortement d'implémenter la première configuration de `/etc/pam.d/other`. C'est un bon réflexe d'être "sécurisé par défaut". Si jamais vous avez vraiment besoin d'accorder à un nouveau service le droit d'authentification, vous pouvez tout simplement créer un nouveau fichier de configuration PAM pour ce service.

Désactiver la connexion des utilisateurs avec des mots de passes vides

Sur la plupart des systèmes Linux il existe un certain nombre de comptes utilisateurs "factices" utilisés pour attribuer des droits à des services systèmes comme serveurs ftp, Web ou les passerelles de messagerie. Disposer de ces comptes permet à votre système d'être plus sécurisé car si ces services sont compromis, un utilisateur malveillant n'obtiendra que les droits limités du compte factice, plutôt que les droits étendus d'un service système tournant sous root. Cependant, donner le droit de connexion à ces comptes constitue une faille de sécurité car ils ont généralement des mots de passes vides. L'option de configuration qui active les mots de passes vides est l'argument "nullok" transmis au module. Vous préférerez enlever cet argument de tous les modules de type "auth" des services autorisant les connexions. Il s'agit habituellement du service login, mais cela peut également inclure des services comme rlogin et ssh. Ceci implique que la ligne suivante dans `/etc/pam.d/login` :

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

doive être transformée en :

```
auth        required        pam_unix.so
```

Désactiver les services inutilisés

En regardant les fichiers situés dans `/etc/pam.d/`, vous remarquerez probablement la présence de fichiers de configuration pour un certain nombre de programmes que vous n'utilisez pas, voire même certains dont vous n'avez jamais entendu parler. Bien que le fait d'accorder l'authentification à ces services ne crée probablement pas d'énorme trou de sécurité, vous feriez mieux de leur refuser ce droit. Le meilleur moyen de désactiver l'authentification PAM pour ces programmes est de renommer ces fichiers. Ne trouvant pas le fichier nommé d'après le service, PAM appliquera la configuration par défaut `/etc/pam.d/other` qui est, a priori, très sécurisée. Si plus tard vous vous rendez compte avoir besoin d'un de ces programmes, vous pouvez tout simplement redonner au fichier son nom initial et tout fonctionnera comme c'est sensé le faire.

Outils de piratage de mots de passes

Les outils de piratage de mots de passes peuvent certes être utilisés par des utilisateurs malveillants pour mettre à mal un système, mais peuvent aussi être utilisés par les administrateurs systèmes pour vérifier le niveau de sécurité des mots de passes sur leurs systèmes. Les deux outils de piratage de mots de passes les plus utilisés sont "crack" et "John the Ripper". "Crack" est probablement déjà intégré à votre distribution favorite. "John the Ripper" peut être téléchargé sur http://www.openwall.com/john/. Lancez les outils sur votre base de mots de passes et vous serez probablement surpris par le résultat.

En outre, il existe un module PAM qui utilise la bibliothèque de "crack" pour vérifier le niveau de sécurité du mot de passe d'un utilisateur lorsqu'il est modifié. Lorsque ce module est installé, l'utilisateur peut modifier son mot de passe uniquement lorsqu'il satisfait aux exigences minimales de sécurité.

Mots de passes Shadow et MD5

Comme déjà dit dans la première section de ce document, les mots de passes Shadow et MD5 vous permettent de sécuriser votre système. Lors de la procédure d'installation, la plupart des distributions modernes vous demanderont si vous désirez utiliser les mots de passes MD5 et/ou Shadow. A moins que vous ayez une bonne raison de ne pas le faire, vous devriez activer ces options. Le processus de conversion des mots de passe non MD5 et non Shadow est compliqué et hors du périmètre de ce document. Le Guide pratique des mots de passes Shadow (version anglaise) n'est plus vraiment à jour mais peut tout de même s'avérer utile sur le sujet.

Exemple d'utilisation

Dans cette section, je vais illustrer ce qui a été présenté dans la section précédente à l'aide d'un exemple simple d'utilisation.

Apache + mod_auth_pam

A titre d'exemple, nous allons installer et configurer mod_auth_pam, un module Apache permet d'authentifier les utilisateurs de votre serveur Web en passant par PAM. Dans la suite de cet exemple je considèrerai qu'Apache est déjà installé. Si ce n'est pas le cas, vous devriez être capables de trouver les paquetages d'installation auprès de votre distributeur.

Notre exemple

L'objectif est de configurer sur notre serveur Web une zone privée, le répertoire `famille/`, nécessitant une authentification des utilisateurs via PAM. Ce répertoire contient des informations familiales réservées uniquement aux membres du groupe "famille".

Installation de mod_auth_pam

Tout d'abord, vous devez télécharger mod_auth_pam depuis http://pam.sourceforge.net/mod_auth_pam/. Utilisez les commandes suivantes pour compiler mod_auth_pam (vous devez être connectés en tant que root) :

```
~# tar xzf mod_auth_pam.tar.gz
~# cd mod_auth_pam-1.0a
~/mod_auth_pam-1.0a# make
~/mod_auth_pam-1.0a# make install
```

Si vous avec des problèmes lors de l'installation du module mod_auth_pam, assurez-vous d'avoir bien installé le paquetage apache-dev de votre distribution. Une fois le module installé vous devrez redémarrer Apache. Cela est habituellement fait par la commande suivante (encore une fois vous devez être root) :

```
~# /etc/init.d/apache restart
```

Configuration de PAM

La configuration PAM pour Apache est stockée dans `/etc/pam.d/httpd`. La configuration par défaut (installée en même temps que mod_auth_pam) est sécurisée mais utilise un module (`pam_pwdb.so`) qui est indisponible sur de nombreux systèmes. (En outre, configurer ce module à partir de zéro sera une vraie partie de plaisir !) En conclusion, effacez le fichier `/etc/pam.d/httpd` et créez en un nouveau.

Décider comment configurer PAM

Si nous désirons configurer comment PAM traitera les requêtes d'authentification d'Apache, nous devons déterminer exactement ce que PAM va vérifier. Tout d'abord, nous voulons que PAM vérifie que le mot de passe de l'utilisateur est identique à celui stocké au niveau du système. Cela ressemble au type "auth" et au module `pam_unix.so`. Nous positionnerons le contrôle du module à "*required*", ainsi l'authentification échouera en cas de mot de passe incorrect. Voici à quoi ressemblera la première ligne de notre fichier `/etc/pam.d/httpd` :

```
auth       required       pam_unix.so
```

Ensuite, nous devons nous assurer que le compte de l'utilisateur est valide (i.e. son mot de passe n'a pas expiré ou tout autre problème de ce genre). Il s'agit du type "account", qui est également fourni par le module `pam_unix.so`. De même, nous positionnerons le contrôle du module à "*required*". Après avoir ajouté cette ligne, notre fichier de configuration

`/etc/pam.d/httpd` ressemble à ceci :

```
auth        required        pam_unix.so
account     required        pam_unix.so
```

Ce n'est pas très sophistiqué mais cela fonctionne. C'est une bonne base de départ pour configurer des services PAM.

Configuration d'Apache

Maintenant que PAM est configuré pour authentifier les requêtes d'Apache, nous allons configurer Apache afin d'utiliser PAM pour gérer l'accès au répertoire `famille/`. Pour ce faire, ajoutez les lignes suivantes à votre fichier `httpd.conf` (habituellement stocké dans le répertoire `/etc/apache/` ou `/etc/httpd`) :

```
<Directory /var/www/famille>
AuthPAM_Enabled on
AllowOverride None
AuthName "Secrets de famille"
AuthType "basic"
require group famille
</Directory>
```

Il se peut que vous ayez à remplacer `/var/www/` par l'emplacement par défaut de vos documents Web, qui est souvent `/home/httpd/`. Quel que soit le répertoire concerné, vous devez y créer le sous répertoire `famille`.

Avant de tester notre configuration, je vais prendre quelques minutes pour expliquer la configuration Apache que vous venez de réaliser. La directive <Directory> est utilisée pour encapsuler des données de configuration pour ce répertoire. A l'intérieur de cette directive, nous avons activé l'authentification PAM ("AuthPAM_enabled on"), désactivé la surcharge de la configuration ("AllowOverride none"), nommé la zone d'authentification "Secrets de famille" (AuthName "Secrets de famille"), positionné le type de l'authentification HTTP (et non l'authentification PAM) à la valeur par défaut ("AuthType "basic"") et restreint l'accès au groupe "famille" ("require group famille").

Test de notre configuration

Maintenant que tout est correctement configuré, il est temps d'admirer notre succès. Lancez votre navigateur favori et rendez vous à http://votre-domaine/famille/ (en remplaçant votre-domaine par votre nom de domaine). Ca y est, vous disposez désormais d'un système d'authentification des utilisateurs !

Ressources

Il existe un certain nombre de ressources, en ligne ou pas, où vous pourrez trouver de l'information concernant l'authentification des utilisateurs. Si vous connaissez des ressources susceptibles d'être ajoutées à cette liste, envoyez moi un message à l'adresse <petehern CHEZ yahoo POINT com>.

PAM

- Guide administrateur système Linux-PAM (version anglaise)
- Manuel de développement de module Linux-PAM (version anglaise)

- Manuel de développement d'application Linux-PAM (version anglaise)

Sécurité en général

- linuxsecurity.com
- securitywatch.com
- Guide pratique de la sécurité
- Packetstorm

Pages man

Beaucoup d'information peut être recueillie dans les pages du manuel de votre système. Ci suivent quelques pages man relatives à l'authentification des utilisateurs. Le nombre entre parenthèses se réfère à la section concernée de la page man. Par exemple, pour visualiser la page man passwd(5), vous devez entrer **`man 5 passwd`**.

- passwd(5)
- crypt(3)
- pam.d(5)
- group(5)
- shadow(5)

Conclusion

J'espère que vous avez trouvé ce guide utile. Faites moi part de toute question, commentaire ou suggestion. Vous pouvez me contacter à l'adresse <petehern CHEZ yahoo POINT com>.