TD3

PAM, Pluggable authentication module

Cette manipulation vous permet d'appréhender PAM en tant qu'administrateur système et en tant que développeur.

Vous travaillez seul, n'oubliez pas de compléter vos notes.

Lectures

Les lectures suivantes pourront être utiles;

- ✗ Voir Morgan, A.G. and Kukuk T. (2005). The linux-pam guides. http://www.kernel.org/pub/linux/libs/pam/
  - The System Administrators' Guide, http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html
  - The Application developers' Manual, http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam_appl.html
- ✗ La page de manuel (*man pam*)

Un peu de prudence …

Organisation

Pour cette manipulation, vous travaillez seul. PAM gère toutes les authentifications de votre système… soyez prudent. Voici quelques précautions élémentaires à prendre avant de se mettre au travail :

- ✗ copier[1] le répertoire `pam.d` vers `pam.d.original` ;
- ✗ (vérifier que l'on sait *rebooter* en *single user)*[2] *;*
- ✗ réfléchir avant d'agir ;

1 J'ai bien dit **copier** pas déplacer, sinon vous êtes déjà foutu !
2 Ça, c'est pour les plus paranos. Les rebelles peuvent s'en passer ;-)

En cas de problèmes

Voici l'extrait de réponse de Peter WOOD sur http://kernel.org

> What the hell do I do now?

OK, don't panic. The first thing you have to realize is that this happens to 50% of users who ever do anything with PAM. It happened here, not once, not twice, but three times, all different, and in the end, the solution was the same every time.

First, I hope you installed LILO with a delay. If you can, reboot, hit shift or tab or something and type:

```
LILO boot: linux single
```

(Replace 'linux' with 'name\-of\-your\-normal\-linux\-image'). This will let you in without logging in. Ever wondered how easy it is to break into a linux machine from the console? Now you know.

If you can't do that, then get yourself a bootkernel floppy and a root disk a-la slackware's rescue.gz. (Red Hat's installation disks can be used in this mode too.)

In either case, the point is to get back your root prompt.

Second, I'm going to assume that you haven't completely nuked your pam installation - just your configuration files. Here's how you make your configs nice again:

```
cd /etc
mv pam.conf pam.conf.orig
mv pam.d pam.d.orig
mkdir pam.d
cd pam.d
```

and then use vi to create a file called "other" in this directory. It should contain the following four lines:

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

Now you have the simplest possible PAM configuration that will work the way you're used to. Everything should magically start to work again. Try it out by hitting ALT-F2 and logging in on another virtual console. If it doesn't work, you have bigger problems, or you've mistyped something. One of the wonders of this system (seriously, perhaps) is that if you mistype anything in the conf files, you usually get no error reporting of any kind on the console - just some entries in the log file. So look there! (Try 'tail /var/log/messages'.)
From here you can go back and get a real configuration going, hopefully after you've tested it first on a machine you don't care about screwing up. :/>

Manipulations

Manipulation administrateur

Commande su

Modifier le fichier `/etc/pam.d/su` afin que l'utilisateur *root* doive entrer son mot de passe lors de l'utilisation de la commande.

Remodifier le fichier pour qu'aucun utilisateur ne doive entrer de mot de passe lors de l'utilisation de su.

Tester également :

- ✗ les restrictions d'accès "temporelle", pas de su entre 10h15' et 10h30' ;
- ✗ les restrictions de lieu (pas de su d'un terminal, uniquement d'une console) ;
- ✗ … (l'un ou l'autre paramètre au choix).

… et ensuite, tout remettre en ordre.

Commande ssh

Apprenons comment configurer PAM afin qu'il limite l'accès à certains utilisateurs. Nous utiliserons *ssh* pour ce faire. Vérifiez d'abord que vous pouvez vous connecter en *ssh* à *localhost*[3].

Nous allons utiliser le module `pam_listfile.so` (dont vous pouvez lire la doc dans le manuel de l'administrateur PAM). Il faudra donc utiliser cette directive… avec les paramètres qui vont bien.

```
auth required pam_listfile.so
```

Les paramètres ont une signification. Par exemple, si je ne veux autoriser que les utilisateurs se trouvant listés dans le fichier — que je devrais créer bine sûr — `/etc/pam.d/list-auth-users`, je peux écrire;

```
auth required pam_listfile.so onerr=succeed item=user \
     sense=allow file=/etc/pam.d/list-auth-users
```

Exercice 1 Configurez PAM afin que seuls les utilisateurs faisant partie du groupe `ssh` puissent se connecter en ssh à votre machine.

Exercice 2 Trouver un module intéressant dans la documentation de PAM et le mettre en œuvre.

3 Si ce n'est pas le cas, installez le paquet *openssh-server*.
Dès que c'est fait, prenez la bonne habitude d'aller modifier la ligne *PermitRootLogin yes* par ***PermitRootLogin no*** dans le fichier de configuration du dæmon *ssh*, `/etc/ssh/sshd_config`.

Manipulation dévelopeur

Objectif Tester le programme proposé dans *The Application developpers'Manual*, le comprendre, l'utiliser et, éventuellement, l'améliorer.

```
$ cat check_user.c
/*
 * This program was contributed by Shane Watts
 * [modifications by AGM]
* You need to update directory /etc/pam.d
 */

#include <security/pam_appl.h>
#include <security/pam_misc.h>
#include <stdio.h>

static struct pam_conv conv = {
     misc_conv,
     NULL
  };

int main(int argc, char *argv[]) {
  pam_handle_t *pamh=NULL;
  int retval;
  const char *user="nobody";

  if(argc == 2) {
     user = argv[1];
  }

  if(argc > 2) {
     fprintf(stderr, "Usage: check_user [username]\n");
     exit(1);
  }

  retval = pam_start("check_user", user, &conv, &pamh);

  if (retval == PAM_SUCCESS)
     retval = pam_authenticate(pamh, 0);  /* is user really user? */

  if (retval == PAM_SUCCESS)
     retval = pam_acct_mgmt(pamh, 0);
     /* permitted access? */

  /* This is where we have been authorized or not. */

  if (retval == PAM_SUCCESS) {
     fprintf(stdout, "Authenticated\n");
  } else {
     fprintf(stdout, "Not Authenticated\n");
  }

 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 */
}
```

Pour rappel, vous devez lier les librairies PAM lors de la compilation… ce qui vous donne une commande du style

```
gcc check_user.c -o check_user -lpam -lpam_misc -ldl
```

Lorsque ce programme compile, ajoutez son fichier de configuration PAM dans le répertoire `pam.d` et testez-le.