

Sébastien Tricaud
toady chez gscore point org

JM2L – Mai 2005

Plan

- Introduction
 - Histoire, Vocabulaire, L'authentification en action, qu'est-ce que Linux PAM, Avantages
- Configurations
- Modules
- Programmation
- Support Prelude

Histoire

- **Début 1995** : Démarrage du projet chez SunSoft
- **Octobre 1995** : Sortie de la RFC 86.0
- **Janvier 1996** : Écriture de l'implémentation libre 'Linux-PAM'
- **Mars 1996** : Première release officielle (0.2)
- **Décembre 1996** : Sortie de Sun Solaris 2.6 'PAM inside'

Histoire

- **Mars 1997** : Standard XSSO, X/Open Single Sign-On Service
- **Août 2004** : Début de ma carrière solo (berlios)
 - Intégration de patchs non appliqués
 - Maintenance de ma propre branche
- **Octobre 2004** : Nouveau souffle au projet, 3 nouveaux mainteneurs

Vocabulaire

- **Token d'authentification**
 - L'authentification n'est pas toujours un mot de passe
- **Credentials** : qualifications
 - En cas d'authentification réussie
 - Caractéristiques et attributs établis par PAM à l'utilisateur

L'authentification en action

- Les utilisateurs
 - N'aiment pas les mots de passe
 - Choisissent des mot de passe du type :
 - toto
 - geantvert
 - biere

L'authentification en action

- Les Administrateurs (avec un grand A)
 - N'aiment pas faire du travail d'esclave
 - Ajouter TOUS les utilisateurs sur TOUTES les machines
 - Changer le mot de passe d'un utilisateur sur TOUTES les machines
 - Coder des scripts non sécurisés pour monter un répertoire distant SMB
 - Recompiler, recoder TOUS les programmes dès que le schéma d'authentification change

Linux PAM, c'est :

- Bibliothèques (bibliothèques en langage nerd)
- Programmes
 - xlock, ssh, wzdfcftpd, apache
 - ... (tout programme sérieux)
- Modules
 - pam_unix, pam_tally, pam_mount, pam_cracklib, pam_time
 - ...

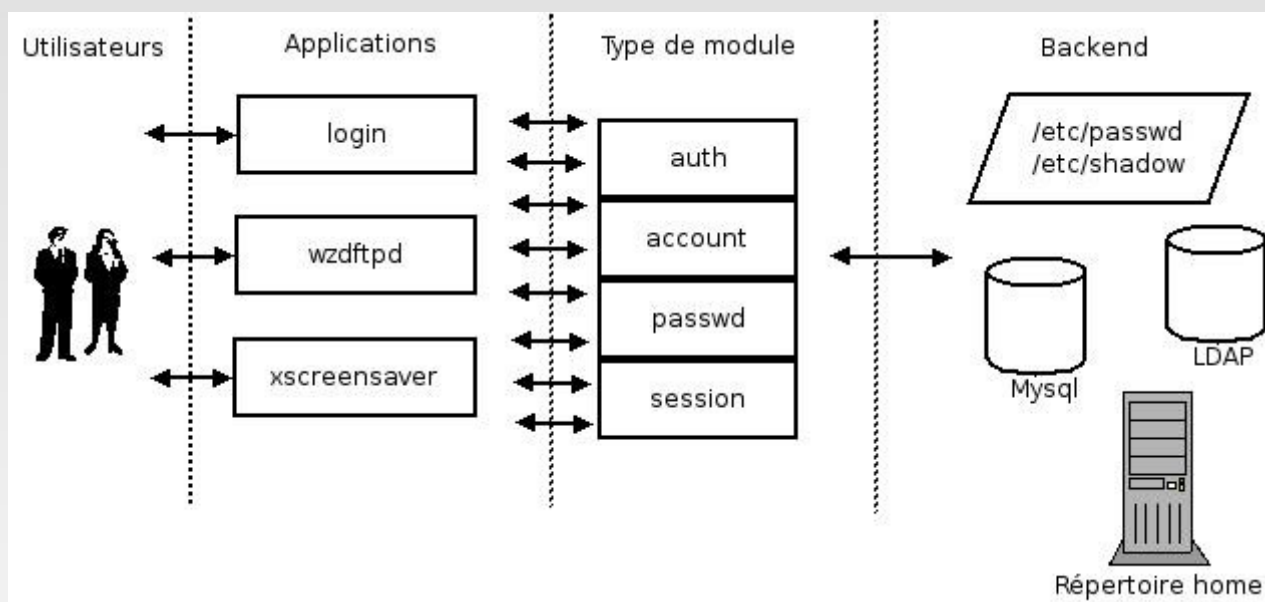
Avantages de PAM

- Type d'authentification modifiable
 - Mot de passe
 - Certificat
 - Clef USB
 - Empreintes digitales
 - Voix

Avantages de PAM

- Unification de l'authentification par rapport aux applications
 - Schéma d'authentification modifiable
 - Pas besoin de toucher à la configuration des programmes
- Séparation des privilèges
 - Sur une machine offrant ssh/imap/ftp.. : on peut choisir que tel ou tel groupe ne puisse pas avoir accès à ssh entre 12h et 14h

Schéma global



Exemple de configuration

- /etc/pam.conf

<nom_du_service> <type> <drapeau> <nom_du_module> options du module..

login auth	requisite	pam_securetty.so
login auth	required	pam_unix.so
login auth	optional	pam_group.so
login account	requisite	pam_time.so
login account	required	pam_unix.so
login password	required	pam_cracklib.so retry=3
login password	required	pam_unix.so shadow md5 use_authtok
login session	required	pam_unix.so

Exemple de configuration

- /etc/pam.d/nomduservice

<type> <drapeau> <nom_du_module> options du module...

```
$ cat /etc/pam.d/login
```

```
auth      requisite pam_securetty.so
auth      required pam_unix.so
auth      optional pam_group.so
account   requisite pam_time.so
account   required pam_unix.so
password  required pam_cracklib.so retry=3
password  required pam_unix.so shadow md5 use_authok
session   required pam_unix.so
```

Configuration : type de module

- **auth** : authentication
- **account** : gestion du compte
- **session** : gestion de la session
- **passwd** : lors de la mise a jour du mot de passe

Configuration : drapeaux de contrôle

- **required**
 - nécessaire pour continuer
 - autres modules de la série encore exécutés
 - l'application ignore de quel module provient l'échec
- **requisite**
 - nécessaire pour continuer
 - n'exécute pas les autres modules de la série
 - renvoie l'erreur à l'application

Configuration : drapeaux de contrôle

- **optional**
 - non nécessaire pour continuer
 - s'il est tout seul, agit comme “required”
- **suffisant**
 - non nécessaire pour continuer
 - n'exécute pas les autres modules de la série : renvoie la notification de réussite a l'application
 - s'il est tout seul, agit comme “required”

Modules utilitaires

- **pam_mount** : monter les répertoires distants SMB et NCP
- **pam_mkhome** : créer un répertoire home pour l'utilisateur
- **pam_motd** : afficher un message d'accueil
- **pam_mail** : afficher le nombre de nouveaux courriels
- ...

Modules pour sécuriser

- **pam_limits** : pour limiter le temps CPU, le nombre de processus, le nombre de fichier ouverts, ...
- **pam_time** : pour restreindre l'accès à un instant donné
- **pam_cracklib** : force les utilisateurs à choisir un bon mot de passe
 - refuse les palindromes (BOB, RADAR, ...), les mots du dictionnaire, les mots de passe trop petits

Étapes de l'authentification

- login démarre une conversation avec l'utilisateur
- affichage de "Password:"
- l'utilisateur tape son mot de passe
- le token d'authentification est envoyé à PAM
- PAM lit le fichier `/etc/pam.conf` OU `/etc/pam.d/login`

Étapes de l'authentification

- les informations envoyées par login sont envoyées aux modules
- les modules vérifient si le token d'authentification est valide
- renvoi de l'état de validité du token à l'application
- login autorise ou non l'utilisateur en fonction de l'état du token renvoyé

Programmation

Après avoir vu les slides qui suivent vous saurez :

- Programmer le support PAM dans une application
- Programmer un module PAM

Linux PAM : librairies

- libpam :
 - authentication
 - conversation
 - interface pour les modules
 - sonde prelude ids
- libpamc :
 - gère l'authentification via un programme client
 - défini un protocole de communication client/serveur (bin/txt)
- libpam_misc : faciliter la vie des développeurs
 - fonction de conversation par texte toute prête
 - dialogue avec les variables d'environnement amélioré

- Permet aux applications de ne pas gérer l'authentification
- Gestion de la communication avec les applications
- Gestion de la communication avec les modules
- Interface de conversation avec l'utilisateur
- Gestion des délais
- Fonctions sécurisées : chaînes de

Programmes

```
/* compiler: gcc plop.c -o plop -lpam -lpam_misc */

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

static struct pam_conv conv = {
    misc_conv,
    NULL
};

...
pam_handle_t *pamh      = NULL;
int retval;

...
pam_start("nomapp", "nomutilisateur", &conv, &pamh);
retval = pam_authenticate(pamh, 0);
if ( retval != PAM_SUCCESS ) {
    /* faire ce que l'application
     * doit faire en cas d'echec
     * de l'authentification*/
    return -1;
}
pam_end(pamh, PAM_SUCCESS);
```

Développement des mo(d)ules : 1/6

Définir les types utilisables par le module

```
#define PAM_SM_AUTH  
#define PAM_SM_ACCOUNT  
#define PAM_SM_SESSION  
#define PAM_SM_PASSWORD
```

Développement des modules : 2/6

Insérer les bonnes en-tetes

```
#include <security/pam_modules.h>  
#include <security/_pam_macros.h>
```

Développement des modules : 3/6

API PAM Modules

`pam_sm_authenticate`

authentification des utilisateurs

`pam_sm_setcred`

pour changer les désignations de l'utilisateur

`pam_sm_acct_mgmt`

gestion du compte

`pam_sm_open_session`

pour démarrer une session

`pam_sm_close_session`

pour fermer une session

`pam_sm_chauthok`

mise à jour de l'authentification

Développement des modules : 5/6

Exemple : enregistrer les connexions des utilisateurs

```
PAM_EXTERN int pam_sm_open_session(pam_handle_t *pamh, int flags,
                                   int argc, const char **argv)
{
    const char *username = NULL;
    int retval = PAM_SESSION_ERR;

    retval = pam_get_user(pamh, &username, NULL);

    if ( ( retval != PAM_SUCCESS ) || ( ! username ) ) {
        pam_syslog(pamh, LOG_WARNING, "Ne peut pas ouvrir la session");
        return PAM_SESSION_ERR;
    }

    pam_syslog(pamh, LOG_WARNING, "L'utilisateur %s ouvre la session", username);

    return PAM_SUCCESS;
}
```

Développement des modules : 6/6

Déclarations pour la compilation statique

```
#ifdef PAM_STATIC

struct pam_module _pam_monmodule_modstruct =
{
    "pam_monmodule",
    pam_sm_authenticate,
    pam_sm_setcred,
    pam_sm_acct_mgmt,
    pam_sm_open_session,
    pam_sm_close_session,
    pam_sm_chauthtok
};

#endif
```

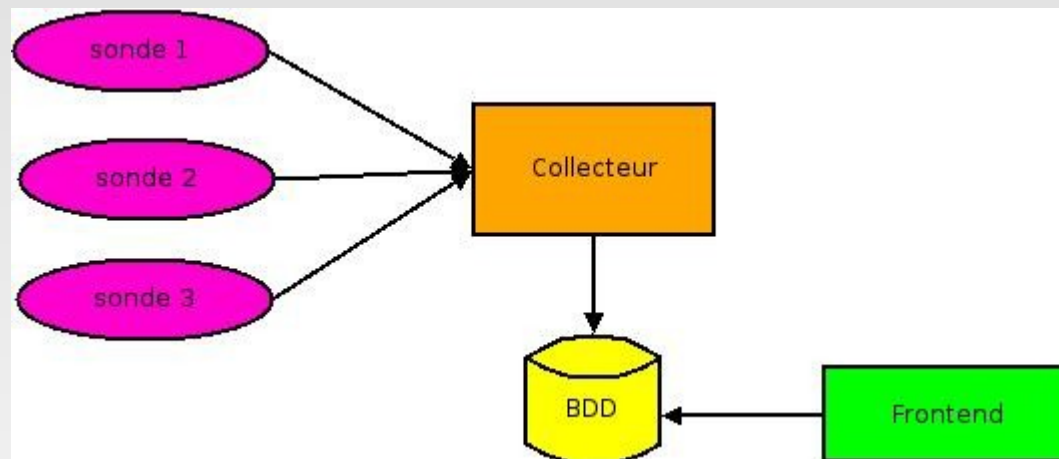
Bindings

- Perl
- Ruby
- Python
- ...

Support de prelude ids

- Framework prelude
- IDMEF
- Collecte des informations

Framework prelude



Message IDMEF (1/2)

alert:

analyzer(0):

analyzerid: 579774451772381

name: PAM

manufacturer: Sebastien Tricaud <http://www.kernel.org/pub/linux/libs/pam/>

model: PAM

version: CVS 0.79

class: pam

ostype: Linux

osversion: 2.6.11-1-686-smp

process:

name:

pid: 22906

create_time: 00:44:10 23/03/2005 (1111535050.50461)

classification:

text: Authentication Failure

Message IDMEF (2/2)

```
source(0):
  spoofed: unknown (0)
  node:
    category: hosts (6)
    name: evil.cracker.org
  user:
    category: application (1)
    user_id(0):
      type: original-user (0)
      tty: ssh
  process:
    name: ssh
    pid: 22906
target(0):
  decoy: unknown (0)
  interface:
  user:
    category: application (1)
    user_id(0):
      type: target-user (2)
      name: GROS_NAIN_DES_BOIS
assessment:
  impact:
    type: other (0)
    description: User not known to the underlying authentication module
```

Collecte d'informations

- Authentification réussie
- Authentification échouée
- Utilisateurs en jeu
- Programme client
- Message d'erreur très précis
 - dlopen() failure when dynamically loading a service module
 - Memory buffer error
 - User not known to the underlying authentication module
 - User account has expired

Installation

Pam ≥ 0.79 et ≤ 0.81 option “--enable-prelude”.

Pam > 0.81 détecte la libprelude et compile le support si elle est présente.

- Client
 - prelude-adduser register PAM “idmef:w”
localhost
- Manager
 - prelude-adduser registration-server
prelude-manager

Futur

- Bugfixing
- Ajouts/Amméliorations basées sur XSSO
- Bindings

Remerciements

- Le mythique Pierre Chifflier
- Andrew Morgan
- Thorsten Kukuk
- Tomas Mraz
- Yoann Vandoorselaere
- Frédéric Motte
- Jean-Philippe Guérard
- Linux Azur

Plus d'infos

- <http://www.kernel.org/pub/linux/libs/pam>
- <http://www.sourceforge.net/projects/pam>
- <http://www.sun.com/software/solaris/pam>
- <http://www.wallinfire.net/pam>
- toady chez gscore point org

Questions ?

