Authentification Linux / MacOSX

Mardi 25 mai 2010

Emmanuel Blindauer
e.blindauer@unistra.fr
Direction Informatique
Université de Strasbourg

Plan

- Introduction
- Mécanisme d'authentification sous Linux
- Les paramétrages possibles dans MacOSX
- Conclusion

Introduction 1/2

- Parc hétérogène
  - Multiples distributions
  - Multiples systèmes d'exploitations
  - Volumétrie faible
  - Problématique de gestion de parc minimale
- Souplesse de la configuration
- Culture de l'interopérabilité

Introduction 2/2

- Système basé sur un unix
- Réutilisation de solutions libres
- Verrouillage des possibilités

Plan

- Introduction
- Mécanisme d'authentification sous Linux
- Les paramétrages possibles dans MacOSX
- Conclusion

PAM : Pluggable Authentication Module

- « auth » : Vérification de l'identité
- « account » : Vérification des droits d'accès
- « session » : pré-requis à l'accès à un service
- « password » : Mises à jour des mots de passe

PAM : Pluggable Authentication Module

- Répertoire : /etc/pam.d/
- Un nom de fichier par service (sauf ssh)
- *`type control module module-args`*
- *`type`*`: auth account session password`
- *`control`*`: required requisite sufficient optional` (Valeurs prédéfinies)
- Pour un type donné : passage de chaque module jusqu'à un retour OK

PAM : Pluggable Authentication Module

- Les modules d'authentification
  - pam_unix : flat file
  - pam_krb5 : Kerberos
  - pam_ldap : vers un serveur LDAP
  - pam_winbind : vers un Active Directory
  - pam_sso : vers un Active Directory (Microsoft)
  - pam_supplicant : réseau
  - pam_ccreds : Mode déconnecté
  - pam_justwriteit … pam_cas ...

PAM : Pluggable Authentication Module

- Les modules session / account
  - pam_mkhomedir
  - pam_tally
  - pam_time
  - pam_mount

PAM : Pluggable Authentication Module

- `auth required pam_env.so`
- `auth sufficient pam_krb5.so realm=UNISTRA.FR`
- `auth sufficient pam_krb5.so realm=DPTINFO.URS.LOCAL use_first_pass`
- `auth sufficient pam_tcb.so shadow nullok prefix=$2a$ count=8 try_first_pass`
- `auth sufficient pam_winbind.so use_first_pass cached_login`
- `auth required pam_deny.so`
- Auth vers deux domaines Kerberos, /etc/passwd ou un serveur Active Directory, au choix
- Ne pas oublier pam_deny!

PAM : Pluggable Authentication Module

- `auth sufficient pam_unix.so`
- `auth [`*`authinfo_unavail=ignore`* `success=1 default=die] pam_ldap.so use_first_pass`
- `auth [default=done] pam_ccreds.so action=`*`validate`* `use_first_pass`
- `auth [default=done] pam_ccreds.so action=`*`store`* `use_first_pass`
- `auth [default=done] pam_ccreds.so action=`*`update`* `use_first_pass`
- Authentification LDAP sur un portable

Les informations fonctionnelles : NSS

- LDAP
- Winbind
- NSS-DB
- Flat file

NSS : LDAP

- Stockage dans serveur ldap au format rfc2307bis (mapping personnalisé disponible)
- Objets : posixAccount, shadowAccount, posixGroup...
- Configuration : /etc/ldap.conf
- Serveurs multiples
- Mécanisme de cache : nscd
- A compléter pour le mode déconnecté

NSS : LDAP

- Acces anonyme, par configuration ou par Kerberos
- Ecriture : ldapvi, Gosa, ldapbrower, scripts
- Solution très éprouvée

NSS : winbind

- Clients linux membre d'un domaine AD
- Besoin d'avoir les informations fonctionnelles
- Stockage idmap_ldap, idmap_rid sans modification de l'AD
- Stockage idmap_ad si écriture aisée dans l'AD
- Solution jeune et active
- Ne pas utiliser nscd

Une meilleure solution :)

- Authentifier avec un service SSO l'utilisateur
- Utiliser pam_ccreds
- Chercher les informations fonctionnelles sur un serveur central et garder un cache
- Utiliser partout où c'est possible le SSO

Exemple 1

- Authentification Kerberos pam_krb5
- Utilisation de pam_ldap
- Utilisation de pam_mount pour monter un module cifs, qui utilisera le ticket Kerberos

Exemple 2

- Solution pour Portable sur acces wifi EAP/TTLS
- Utilisation de Kerberos pam_supplicant + pam_krb5 + pam_ccreds
- Utilisation de /etc/passwd (configuration manuelle)
- Montage non automatique d'un lecteur réseau

Plan

- Introduction
- Mécanisme d'authentification sous Linux
- Les paramétrages possibles dans MacOSX
- Conclusion

MacOSX

- Pas de PAM
- Fichier XML : /etc/authorization
- Directory Service : Client ligne de commande : dscl
- Changer l'authentification : « Utilitaire d'annuaire »
  - Active Directory (dsconfigad)
  - Open Directory (dsconfigldap)
- A la main pour Kerberos : /etc/authorization + kerberosautoconfig (/Library/Preferences/edu.mit.Kerberos)

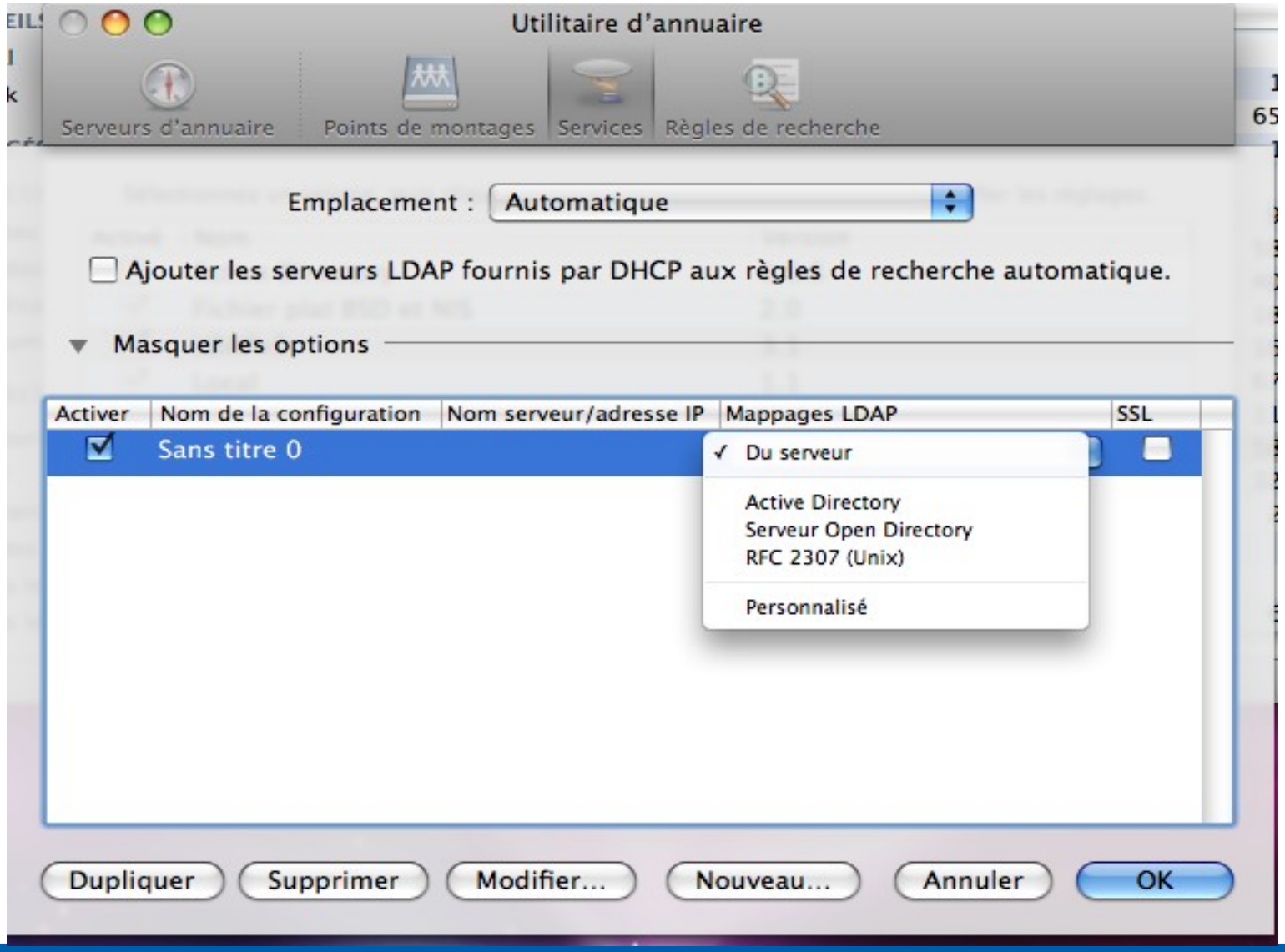
MacOSX : données fonctionnelles

- Intégration Active Directory :
  - Intégration en tant que client
  - Authentification AD
  - Utilisation de samba
  - En particulier DNS Active Directory
- Outils : dscl /Active\ Directory/All\ Domains -read /Users/bart

MacOSX : données fonctionnelles

- LDAP RFC 2307 : Utilisable que via l'onglet « Services » de « Utilitaire d'annuaire »
  - Mapping possible d'attributs
  - Etat identique à un Linux

MacOSX : données fonctionnelles



MacOSX : Open Directory

- Service LDAP + Kerberos clef en main
- Intégration de nouveaux postes MacOSX aisée
- Accès LDAP
- Réplication possible
- Utilisation d'automount
- Configuration des clients via Workgroup Manager (WGM) et MCX
- Augmented Record (Leopard, ~overlay LDAP

MacOSX : Open Directory

- Service de synchronisation des données « Portable Home Directory »
- Énormément de services publiés :
  - DHCP pour les serveurs LDAP OpenDirectory
  - LDAP pour les points de montage (« HOME »)
  - LDAP pour les imprimantes

MacOSX : Kerberos

- Approbation Active Directory / Open Directory (Magic triangle)
- Safari, iChat, iMail, AFP, VPN, Webdav … compatible Kerberos
- 801.1x : Authentification préalable : utiliser « Profils ouverture de session »

Mais...

- NFS sans Kerberos
- Samba 3.0.25 !?

Conclusion

- Linux : Philosophie de souplesse et d'interoperabilité
- Presque tout peut être fait
- Pas d'outil graphique
- MacOSX : Intégration possible dans Active Directory
- OpenDirectory : openLDAP + Kerberos
- Des outils bien pensés mais complexes