Cours 6 : Annuaire LDAP

Rabii El Ghorfi

Plan

Plan

Objectifs

Permet de fusionner plusieurs bases de données en un unique annuaire informatique

- base Microsoft Excel du personnel administratif
- base Microsoft Access du personnel enseignant
- base `/etc/passwd` des comptes unix
- base `/etc/aliases` (ou Sympa) des listes de diffusion
- base Samba des utilisateurs windows
- autres bases, MySQL, ...
- ...

Exemple:

Comment envoyer un e-mail a l'ensemble du personnel administratif en sachant que l'administrateur recevra uniquement une liste de noms/prénoms?

Le concept d'annuaire

Un annuaire est comme une base de données. . .

→ on peut y mettre des information et les consulter

Cependant un annuaire est spécialisé :

→ Dédié à la lecture plus qu'à l'écriture

→ L'accès aux données se fait par des recherches multi-critères.

Son objectif est de maintenir de façon cohérente et contrôlée une grande quantité de données.

Exemples d'annuaire :

- carnet d'adresses
- annuaire téléphonique
- répertoire des rues
- . . .

Le concept d'annuaire

Un annuaire est comme une base de données...

→ on peut y mettre des information et les consulter

Cependant un annuaire est spécialisé :

→ Dédié à la lecture plus qu'à l'écriture

→ L'accès aux données se fait par des recherches multi-critères.

Son objectif est de maintenir de façon cohérente et contrôlée une grande quantité de données.

Différences annuaires/SGBD : Dans un annuaire :

- ⋆ pas de dépendances entre les objets stockés
- ⋆ les objets peuvent être distribués sur plusieurs annuaires pour assurer une meilleure disponibilité
- ⋆ les applications de l'annuaire n'ont pas besoin de connaître la structure interne des données stockées

Plan

L'annuaire LDAP

LDAP → **L**ightweight **D**irectory **A**ccess **P**rotocol
Héritier de l'annuaire X500 (proposé par l'ISO)

- ⋆ standard conçu par les opérateurs télécom pour interconnecter leurs annuaires téléphoniques
- ⋆ X500 adapté à l'internet → LDAP (même modèle de schéma, ...)

LDAP a été proposé en 1995 :

- ⋆ Standard d'annuaire au dessus de TCP/IP
  - ▸ Le protocole ne concerne pas le contrôle d'accès aux données de l'annuaire
- ⋆ version 3 actuellement (RFC 2251)
- ⋆ aussi RFC 2252 à 2256, RFC 2829 à 2830, RFC 2849

Objectifs

- fournir aux utilisateurs des informations fiables, facilement accessibles
- permettre aux utilisateurs de mettre à jour eux-même leurs informations personnelles
- rendre les informations accessibles de façon contrôlée
- éviter la redondance d'informations : un seul annuaire pour l'ensemble des services
- faciliter la gestion (administration) des postes de travail, des équipements réseau

Tout ceci est fait sans remettre en cause les applications existantes

Concepts

LDAP définit :

un protocole. accéder à l'information contenue dans l'annuaire,

un modèle d'information. le type des informations contenues dans l'annuaire,

un modèle de nommage. comment l'information est organisée et référencée,

un modèle fonctionnel. comment accéder à l'information (syntaxe des requêtes,etc...),

un modèle de sécurité. comment données et accès sont protégés,

un modèle de duplication. comment la base est répartie entre serveurs,

des API. pour développer des applications clientes,

LDIF. un format d'échange de données.

Protocole LDAP

Le protocole définit :

- Comment s'établit la communication client-serveur :
  - → commandes pour se connecter ou se déconnecter, pour rechercher, comparer, créer, modifier ou effacer des entrées.
- Comment s'établit la communication serveur-serveur :
  - → échanger leur contenu et le synchroniser (réplication service)
  - → créer des liens permettant de relier des annuaires les uns aux autres (referral service).
- Le format de transport de données :
  - → pas l'ASCII (comme pour HTTP, SMTP...) mais le Basic Encoding Rules (BER), sous une forme allégée (appelée LBER : Lightweight BER)

Protocole LDAP

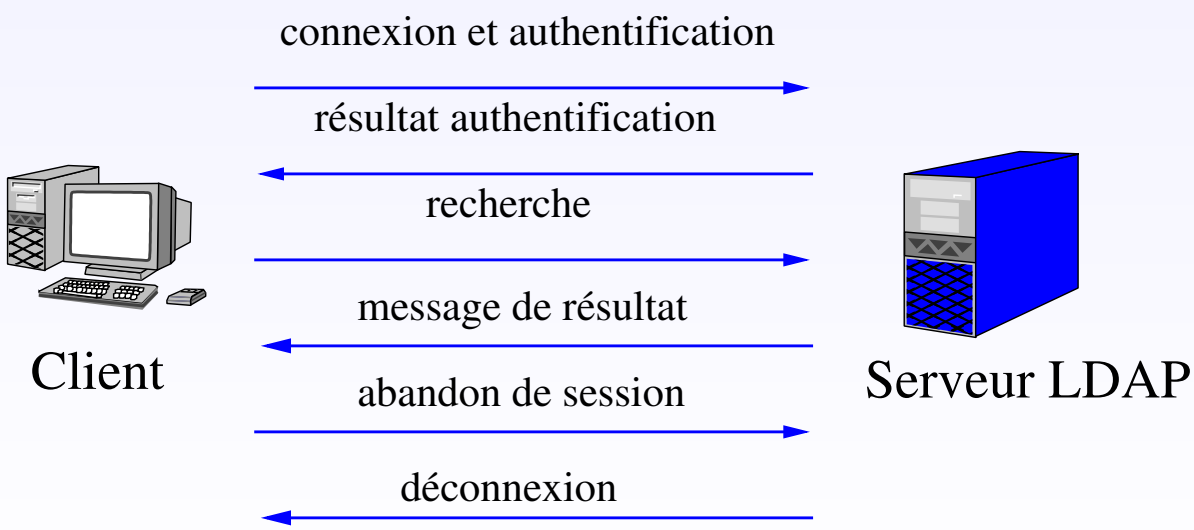
Le protocole définit (suite) :

- Les mécanismes de sécurité :
  - → méthodes de chiffrement et d'authentification
  - → mécanismes de règles d'accès aux données.
- Les opérations de base :

  interrogation : search, compare

  mise à jour : add, delete, modify, rename

  connexion au service : bind, unbind, abandon



Plan

Le modèle d'information

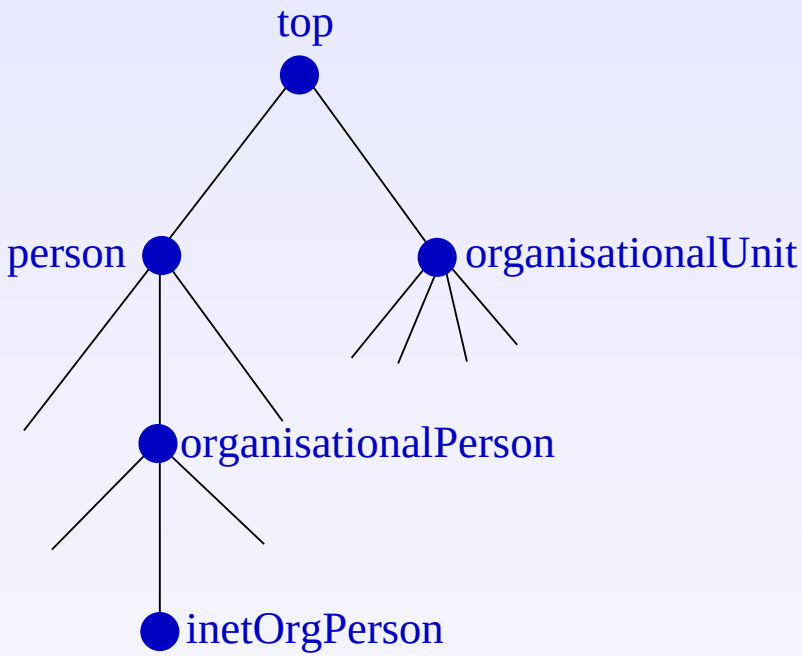
Le modèle d'information définit le type des données pouvant être stockées dans l'annuaire

- L'entrée (Entry) = élément de base de l'annuaire. Elle contient les informations sur un objet de l'annuaire.
- Ces informations sont représentées sous la forme d'attributs décrivant les caractéristiques de l'objet.
- Toute sorte de classe d'objet (réel ou abstrait) peut être représentée.
- Le *schéma* de l'annuaire définit la liste des classes d'objets qu'il connaît. Le *Directory schema* est la « charte » qui donne, pour le serveur, l'ensemble des définitions relatives aux objets qu'il sait gérer.
  - Le schéma décrit les classes d'objets, leurs types d'attribut et leur syntaxe.
  - Chaque entrée de l'annuaire fait obligatoirement référence à une *classe d'objet* du *schéma* et ne doit contenir que des attributs qui sont rattachés au type d'objet en question.

Le modèle d'information

- Un attribut est défini par :
  - un nom, un identifiant unique (OID), mono/multi valué, une syntaxe et des règles de comparaison (*matching rules*), une valeur (format+taille limite), modifiable ou non
- Les classes d'objet modélisent
  - des objets réels : Un compte UNIX (`posixAccount`), une organisation (`o`), un département (`ou`), un personnel (`organizationPerson`), une imprimante (`device`),...
  - ou abstraits : l'objet père de tous les autres (`top`),...
- Une classe d'objet est définie par
  - Un nom, OID, des attributs obligatoires, des attributs optionnels, un type (structurel, auxiliaire ou abstrait)

Le modèle d'information



- Chaque objet hérite des propriétés (attributs) de l'objet dont il est le fils.
- On précise la classe d'objet d'une entrée à l'aide de l'attribut objectClass.
- Il faut obligatoirement indiquer la parenté de la classe d'objet en partant de l'objet top et en passant par chaque ancêtre de l'objet.

Le modèle d'information (exemple)

L'objet `inetOrgPerson` à la filiation suivante :

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

L'objet `person` a comme attributs : `commonName, surname, description, seeAlso, telephoneNumber, userPassword`

L'objet fils `organizationalPerson` ajoute des attributs comme : `organizationUnitName, title, postalAddress...`

L'objet petit-fils `inetOrgPerson` lui rajoute des attributs comme : `mail, labeledURI, uid (userID), photo...`

Remarques :

- Une entrée peut appartenir à un nombre non limité de classes d'objets.
- Les attributs obligatoires sont la réunion des attributs obligatoires de chaque classe.

Le modèle de nommage

Il définit comment les entrées de l'annuaire sont organisées et comment elles sont référencées.
Structure arborescente contenant deux catégories d'objets :

les conteneurs : départ d'une nouvelle branche (nœud intermédiaire de l'arbre)

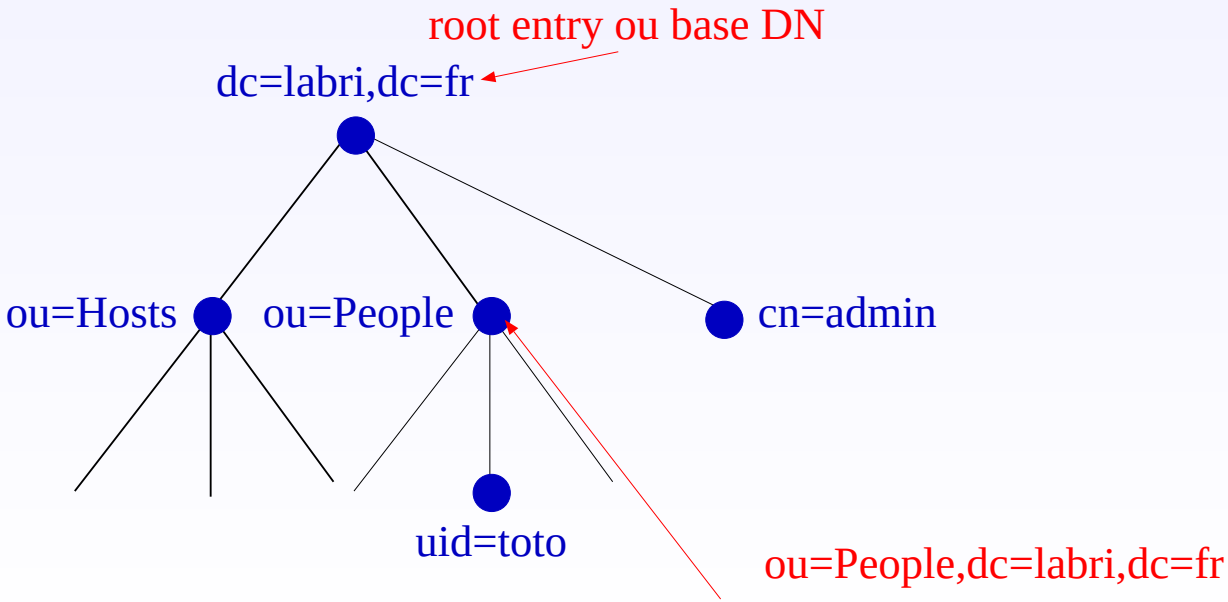
- peuvent contenir des conteneurs ou des feuilles
- généralement, une sous-organisation de l'organisation (zone géographique,...)

les feuilles : elles représentent les données (généralement les machines, les utilisateurs,...)

Le modèle de nommage

Il définit comment les entrées de l'annuaire sont organisées et comment elles sont référencées.
Structure arborescente contenant deux catégories d'objets :

- Structure logique hiérarchique : le DIT (Directory Information Tree)
- Une entrée est identifiée par un nom unique : le DN (Distinguish Name)
- RDN(Relative Distinguish Name)



Le format LDIF

LDIF → **L**DAP **I**nterchange **F**ormat

- ★ Standard de représentation des entrées sous format texte.
- ★ Permet de :
  - ▶ faire des *imports/exports* de la base ou d'une partie de la base
  - ▶ créer, ajouter, modifier, . . . un grand nombre d'entrées de manière automatisée

```
dn: uid=toto, ou=People, dc=labri, dc=fr
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
uid: toto
uidNumber: 44321
gidNumber: 200
homeDirectory: /home/toto
cn: toto titi
loginShell: /bin/bash
```

Le modèle fonctionnel

Il décrit le moyen d'accéder aux données (syntaxe des requêtes) et les requêtes que l'on peut leur appliquer.
Rappel des opérations de consultation/mise-à-jour

- ⋆ opérations d'interrogation : recherche (`search`) et comparaison (`compare`) d'entrées
- ⋆ opérations de mise-à-jour des entrées de l'annuaire : `add`, `delete`, `modify`, `rename`

Il n'y a pas d'opération de lecture d'une entrée

→ pour connaître le contenu d'une entrée, il est nécessaire d'écrire une requête qui pointe sur cette entrée.

Le modèle de réplication

Il définit comment dupliquer l'annuaire sur plusieurs serveurs.

- ★ améliorer le temps de réponse
- ★ être tolérant aux pannes

Deux types de serveurs LDAP

supplier server: fournit les données

consumer server: reçoit les données du maître

Possibilité de partitionner l'annuaire (éclatement sur plusieurs serveurs)

- ★ liens virtuels entre les différentes partitions (*referral service*)

Le modèle de sécurité

Authentification pour se connecter au service

- Anonymous authentication, Root DN/passwd authentication (administrateur), User DN/passwd

Contrôle de l'accès aux données

- droits d'accès aux données (fonctions de l'utilisateur authentifié)
- règles définies sous forme d'ACL (*Access Control List*) au niveau du sommet d'un sous-arbre ou d'une entrée.

Chiffrement des transactions (LDAP+SSL, ...)

Plan

Mettre en place un annuaire LDAP

Il faut bien choisir les schémas

- Quelles informations veut-on stocker dans l'annuaire?
- Quelles sont les applications qui vont utiliser l'annuaire?

Il faut réfléchir a l'organisation du DIT

- impact sur la performance, les droits d'accès, ...

Puis dans un deuxième temps

- gestion centralisée sur un seul serveur?
- nombre de serveurs redondants? Emplacement?

OpenLDAP

- Logiciel LDAP du domaine public
- le démon `slapd`
  → traite les requêtes LDAP
- le démon `slurpd`
  → permet la réplication
- des librairies LDAP
  - par exemple pour authentifier les logins via LDAP : `libpamldap, libnssldap`
- des utilitaires :
  - `ldapadd, ldapdelete, ldapmodify, ldapmodrdn, ldappasswd, ldapsearch`

Configuration du serveur ldap(1/2)

Le fichier `/etc/ldap/slapd.conf` permet de configurer le démon `sladp`

- définition des schémas utilisés
  `include inetorgperson.schema`
- définition du *backend* (type de la base de données utilisée)
  `backend bdb`
- définition de la base, de l'annuaire et de l'administrateur
  - le suffixe (racine de l'arbre)
    `suffix "dc=labri,dc=fr"`
  - l'administrateur et son mot de passe
    `rootdn "cn=Manager,dc=labri,dc=fr"`
    `rootpw MD5x0dg9sP0uUf+NRm0MIPz7Q==`
  - le répertoire où la base est stockée
    `directory "/var/lib/ldap"`

Configuration du serveur ldap(1/2)

Définition des ACLs (`man slapd.access`)

```
 # par défaut
access to attrs=userPassword
            by dn="cn=admin,dc=com" write # l'admin
            by anonymous auth # droit de lecture lors du
            bind
            by self write # le propriétaire
            by * none
access to dn.base="" by * read
# L'administrateur a un accès total en écriture, tous
# les autres utilisateurs peuvent tout lire.
access to *
            by dn="cn=admin,dc=com" write
            by * read
```

configuration du client LDAP

La configuration se fait grâce au fichier `/etc/ldap/ldap.conf`

- `man ldap.conf`
- peut aussi se faire dans `/.ldaprc`
- exemple de fichier `ldap.conf`

```
# base par défaut à contacter pour les opérations LDAP
BASE dc=labri, dc=fr
# en tant que qui le client va se connecter
# à la base
BINDDN uid=toto,ou=People,dc=labri,dc=fr
# le serveur auquel se connecter
URI ldap://147.210.20.21:389/
```

Authentification Unix via LDAP

- PAM (Pluggable Authentication Modules)
  - permet de gérer la politique d'authentification sans recompilation
  - pour authentifier via LDAP, il faut ajouter la ligne `auth sufficient pam_ldap.so` (qui signifie que l'authentification LDAP est suffisante) dans le fichier `/etc/pam.d/common-auth`. Il faut faire de même pour tous les autres fichiers `/etc/pam.d/common-*`.
  - Modifier éventuellement `/etc/pam.d/ssh`,...
- Configurer l'accès à la base dans `/etc/libnss-ldap.conf` et `/etc/pam_ldap.conf` (voir pages man)
- Indiquer dans `/etc/nsswitch.conf` l'ordre d'interrogation pour l'authentification

  → toujours laisser `files` en premier !