Sécurité et
Administration des
Systèmes
Informatiques

Administration réseau

LEGOND-AUBRY Fabrice
Fabrice.Legond-Aubry@u-paris10.fr

Section : Section : « Les services réseaux »

Identification / Authentification

NIS

LDAP: Principes

LDAP: Architecture

LDAP: Manipuler les bases

LDAP: Distribution et redondance

Un exemple: OpenLDAP

Définitions (un peu de sécurité)

- Identité/identifiant: Une identité est une abstraction attachée à une entité (personne, groupe, ...) et repérable de façon non ambiguë par un code appelé "identifiant" ; l'identité peut être basique ou au contraire enrichie de certains des attributs de l'entité.
- Créance ("credentials"): Eléments fournis comme preuves de ce qui est avancé
- Authentification: Processus par lequel l'individu/usager prouve à une entité de vérification qu'il a bien l'identité qu'il proclame/revendique en lui présentant ses « créances ».
- Dans le cas de services Internet, aujourd'hui le vérifieur se contente le plus souvent de demander à l'usager de frapper son "mot-de-passe" pour s'authentifier. Il existe aussi de clefs numériques, des créances biométriques.

Historique de l'authentification

- L'authentification nss est la première des méthodes
  - Nss=« Name Service Switch »
  - A la fois un système d'identification et d'authentification unique pour tout le système
  - Un seul fichier de configuration
- L'authentifcation pam.d est la suivante
  - Capable d'utiliser nss
  - Pam=« Pluggable Authentification Module »
  - Authentification différenciée suivant les services
  - Chaque service a son fichier de configuration
- Le démon SASL est le dernier en date (rfc 2222, 2245, 2444, 2831)
  - Sasl=« simple authentification and security layer »
  - Capable d'utiliser nss et pam.d
  - Il offre de nouveaux moyens d'authentification par défi
  - En particulier pour les méthodes orientées connexions distantes

Identification associée à « nss »

- « ***/etc/nsswitch.conf*** » : Il détermine l'ordre dans lequel sont effectuées les recherches d'identité (les infos sur les personnes)

```
passwd:     files nis ldap
shadow:     files nis ldap
group:      files nis ldap
ethers:     nis [NOTFOUND=return, UNAVAIL=return]
```

- A chaque étape le module évalue le résultat:
  - « ***succes*** »: la donnée a été trouvée. Action par défaut: Retour à l'appelant (return).
  - « ***notfound*** »: la donnée n'a pas été trouvée. Action par défaut: on passe à la méthode suivante (continue).
  - « ***unavail*** »: service indisponible. Action par défaut: on passe à la méthode suivante (continue).
  - « ***tryagain*** »: service temporairement indisponible. Action par défaut: on passe à la méthode suivante (continue).

Authentification PAM

- Avant si un service voulait utiliser un service d'authentification spécifique :
  - Il devait être compiler avec les mécanismes d'authentification
- En cas de nouvelles méthodes d'authentification
  - recompilation de l'application
- Création d'une nouvelle interface d'authentification appelée « *Pluggable Authentification Module* »
- Avantage de l'authentification PAM pour un service
  - on modifie un « simple » fichier de configuration.
  - Chaque programme à son fichier de configuration dans « /etc/pam.d/ »

Les fichiers « /etc/pam.d/* »

- Chaque fichier représente un service particulier
- Structure de chaque ligne

  module-type control-flag module-path args

- module-type : Etape dans l'authentification
  - ***Account***: fournit une vérification des types de service du compte utilisateur (expiration du mot de passe, …)
  - ***Auth***(entification): établit la correspondance entre l'utilisateur et celui pour lequel il prétend être.
  - ***Password***: mettre à jour les mécanismes d'authentification (changer de mot passe)
  - ***Session***: tout ce qui doit être fait en priorité pour un service donné et après qu'il soit retiré

Les fichiers « /etc/pam.d/* »

- Structure de chaque ligne

  module-type control-flag module-path args

- control-flag : option pour le module
  - ***Requisite***: succès requis, en cas d'échec retour à l'application
  - ***Required***: succès requis, en cas d'échec retour à l'application après l'exécution des tous les autres modules
  - ***Sufficient***: le succès est suffisant pour réussir l'authentification (sauf si un module requis a déjà échoué)
  - ***Optional***: le module n'influence pas le succès de l'authentification
  - ***Include***: permet l'inclusion d'un fichier
  - Une autre méthode possible est :
    - ✔[value1=action1 value2=action2 ...]
    - ✔Voir le manuel « ***man pam.d*** »

Exemple: « /etc/pam.d/login »

```
#%PAM-1.0
#autorise root sur des console ‘secure’ seulement
auth       required   pam_securetty.so

# utilise l’authentification décrite d ans le fichier
# /etc/pam.d/system-auth
auth       required   pam_stack.so service=system-auth

# verifier la presence de /etc/nologin
# si /etc/nologin existe, seul root peut se logger
auth       required   pam_nologin.so

# utilise l’authentification décrite dans le fichier
# /etc/pam.d/system-auth
account    required   pam_stack.so service=system-auth
password   required   pam_stack.so service=system-auth
session    required   pam_stack.so service=system-auth

# pour la session fixe les droits sur la console
session    optional   pam_console.so
```


Section : « Les services réseaux »

Identification / Authentification

Network Information Service

LDAP: Principes

LDAP: Architecture

LDAP: Manipuler les bases

LDAP: Distribution et redondance

Network Information Service

Le service NIS : Network Information Service

- Le but du service NIS:
  - Partager des informations sur l'ensemble du système
    - ✔Utilisation du réseau
  - Eviter de configurer toutes les machines !
    - ✔Réduire le temps de configuration
- Modèle de Client/serveur
  - Serveurs
    - ✔Le maître est utilisé pour modifier et propager les changement
    - ✔Utilisation de serveurs esclaves pour faire de la tolérance aux fautes
  - Clients
    - ✔Toutes les machines hôtes peuvent accéder aux informations
    - ✔Un serveur peut aussi agir comme un client

Network Information Service

Le domaine NIS

- Le domaine NIS
  - Un serveur sert seulement un domaine
  - Un serveur peut être maître ou esclave
  - Un client appartient à un seul domaine
  - Plusieurs domaines impliquent plusieurs serveurs
- Le domaine NIS ne doit par être un nom trivial !
  - Ne doit pas être l'@ IP
  - Ne doit pas être le nom ou l'acronyme de l'organisation
  - Ne doit pas être le nom du serveur NIS
- Commandes pour fixer ou obtenir le nom de domaine
  - ***nisdomainname***, ***domainname***

Network Information Service

Technologie NIS

- Le serveur peut être n'importe quelle machine du réseau
  - Elle doit accepter les appels RPC (développé par Sun)
  - Le NIS pose des problèmes avec les firewalls
- NIS est basé sur les RPC
  - Commandes « ***rpcinfo –p localhost*** »
  - Commandes « ***rpcinfo -u localhost ypbind*** »
  - Service RPC « ***ypserv*** » sur le serveur
    - ✔Gère les informations
  - Service RPC « ***ypbind*** » sur le client
    - ✔« Cache » les informations et maintient les connexions au serveur NIS
  - Le service « ***rpc.ypxfrd*** » permet un transfert efficace des infos
  - Un client doit interrogé le portmapper (port tcp 111)
  - Le port des services peut VARIER !

Network Information Service

Mise en place du serveur

- Le répertoire « ***/var/yp*** » doit exister
- Il contiendra les bases d'informations que l'administrateur veut publier.
- Initialisation des bases d'informations
  - « ***ypinit [-m] [-s master_name]*** »
  - L'option -m est utilisé pour créer un serveur maître
  - Un et un seul serveur maître est nécessaire pour un domaine
  - L'option -s est utilisé pour créer un serveur esclave
- Un domaine peut fonctionner sans serveur esclave
  - Un serveur esclave est nécessaire pour la redondance
  - Le serveur maître doit connaître les serveurs esclaves
- Exécuter « ***make*** » dans « ***/var/yp*** » pour resynchroniser les informations
  - Lors de la création de nouveaux alias, utilisateurs, …

Network Information Service

Cartes NIS (informations partagées)

- Les cartes NIS sont de petites BdD
  - Le format utilisé est DBM
  - Elles sont crées à l'aide de la commande « ***makedbm*** » invoqués par la commande « ***ypinit*** »
- Les informations partageables sont:
  - Les informations utilisateurs (***/etc/passwd***)
  - Les informations sur les groupes (***/etc/group***)
  - Le noms des machines (***/etc/hosts***)
  - Les alias d'eMail (***/etc/aliases***)
  - Les services offerts (***/etc/services***)
  - Les protocoles (***/etc/protocoles***)
  - Les services RPC (***/etc/rpc***)

Network Information Service

Cartes NIS (informations partagées)

- Possibilité de monté des répertoires en fonction des utilisateurs via le service "***automounter***"
- Les cartes partagées sont, en générale, multi-critères
  - Mot de passe: passwd.byname, passwd.byuid
  - Groupes: group.byname, group.bygid
  - Protocoles: protocols.byname, protocols.bynumber
  - Machines: hosts.byaddr, hosts.byname
  - Alias : mail.aliases, mail.byaddr, mail.revaliases
  - Voir les fichiers générés dans ***/var/yp/domaine/***….
- Obtenir l'ensemble des entrées d'une carte
  - La commande « ***ypcat*** » permet d'afficher le contenu d'une carte
    - ✔Ex: ***ypcat passwd*** ou ***ypcat passwd.byuid***

Network Information Service

Commandes NIS

- Recherche d'une entrée particulière dans une carte
  - Commande ***ypmatch***

```
legond@hebe > ypmatch bonnaire passwd
bonnaire:XXXXXXXXX:358:300:Xavier BONNAIRE:/home/bonnaire:/bin/tcsh

legond@hebe > ypmatch 358 passwd
Can't match key 358 in map passwd.byname. Reason: No such key in map

legond@hebe > ypmatch 358 passwd.byuid
bonnaire:XXXXXXXXX:358:300:Xavier BONNAIRE:/home/bonnaire:/bin/tcsh
```

- Quel est le nom du serveur NIS ?
  - Commande *ypwhich*

```
legond@hebe > ypwhich
zeus
```

- Tester un serveur NIS : *yptest*

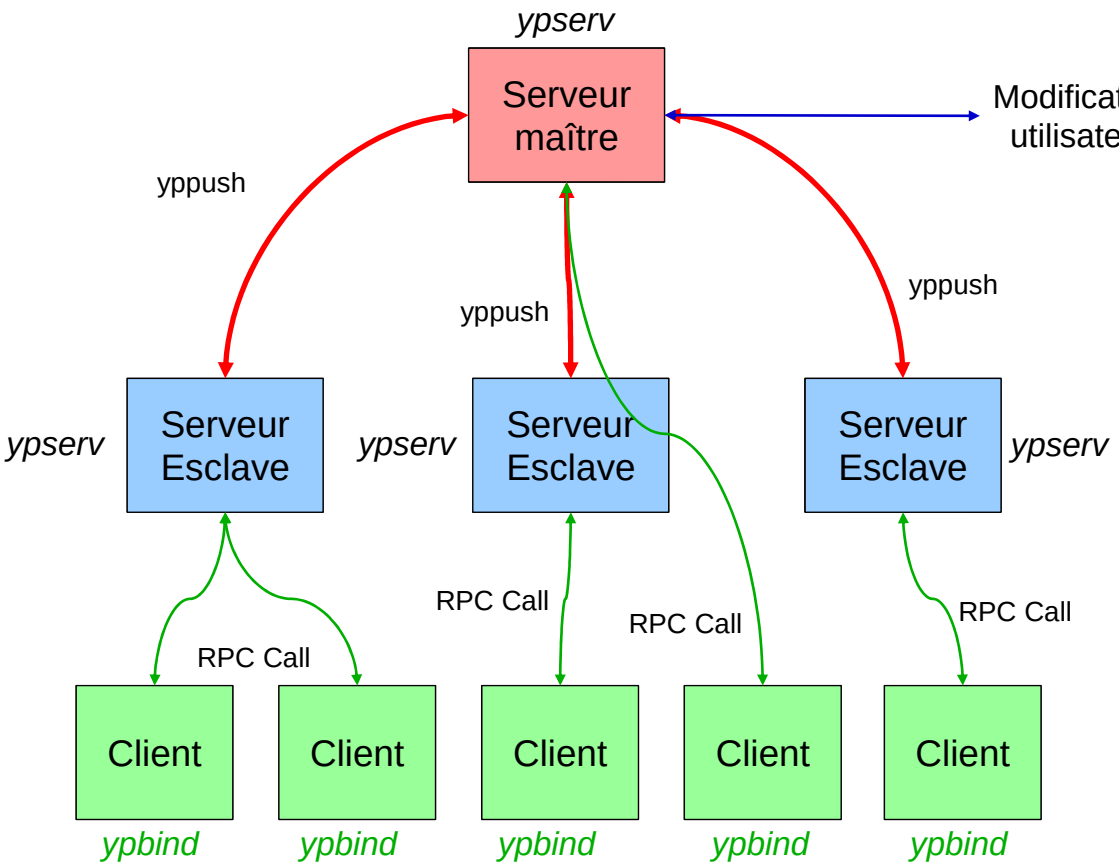
Network Information Service

Commandes NIS

- Changer manuellement de serveur NIS
  - Commande ***ypset <server>***
- Obtenir des informations sur une carte
  - ***yppoll [-h host] [-d domain] <mapname>***
    - ✔Retourne les informations à propos de la carte
    - ✔L'option -d permet de choisir un autre domaine NIS
    - ✔L'option -h permet de choisir le serveur NIS (master or slave)

```
legond@hebe > yppoll passwd.byname
Domain mondomaineamoi is supported.
Map passwd.byname has order number 1098194382. [Tue Oct 19 15:59:42
2004]
The master server is server1.
```

Cohérence des serveurs NIS



Network Information Service

Cohérence des serveurs NIS

- Du serveur maître vers les serveurs esclaves
  - Commande ***ypxfr [-h host] mapname***
    - ✔Transfert de cartes depuis le machine spécifiée
    - ✔Utilisé par ***ypserv*** sur les serveurs esclaves quand un transfert de carte est réquisitionné par le serveur maître
    - ✔Le serveur esclave doit reconstruire ses bases avec l'outil ***makedbm***
  - Démon ***rpc.ypxfrd***
    - ✔Utiliser pour améliorer la transfert de larges cartes
    - ✔Utilisation du protocole TFTP pour le transfert
    - ✔Soulève des problème de sécurité (firewall)

Commandes “utilisateur” du NIS

- Commande ***yppasswd***
  - Similaire à “***passwd***”
  - Modifie le mot de passe sur le serveur maître
    - ✔Seulement sur linux
    - ✔Les autres unix utilisent la commande ***passwd***
- Command ***ypchsh***
  - Similaire à “***chsh***”
  - Modifie le shell sur le serveur primaire
- Command ***ypchfn***
  - Similaire à “***chfn***”
  - Modifie le nom d’utilisateur sur le serveur NIS maître

Section : « Les services réseaux »

Identification / Authentification

Network Information Service

LDAP: Principes

LDAP: Architecture

LDAP: Manipuler les bases

LDAP: Distribution et redondance

LDAP: Principes

Lightweight Directory Access Protocol : LDAP

LDAP c ’est :

- Un protocole standard TCP/IP pour l’accès au contenu qui impose une structuration hiérarchique des données
- Un **modèle d’information** organisées
- Un **modèle fonctionnel** de diffusion de données
- Un **modèle de nommage** permettant une unicité d’appellation des entrées à un niveau mondial, ce qui permet une répartition cohérente des informations sur plusieurs serveurs
- Un **modèle de sécurité** permettant le contrôle d’accès
- Ce protocole est particulièrement adapté :
  - Pour la diffusion à grande échelle de données simples
  - Pour les mises à jour dynamiques
  - Pour la sécurisation et la personnalisation des données
- Les ports utilisés sont 389 (ldap) et 636 (ldaps)

LDAP: Principes

LDAP vs NIS, LDAP vs BdD

- **LDAP** vs base de données
  - Plus de lecture que d'écriture
  - Bases plus facilement extensibles
  - Diffusion à plus large échelle
  - Répartition des données plus éclatée entre serveurs
  - Duplication de l'information
  - Importance des standards
  - Fortes quantités d'enregistrements mais faibles capacités de stockage
- LDAP vs NIS
  - Le traffic peut être encrypté via SSL or TLS
  - Toutes les fonctionnalités NIS peuvent être implantés (shadow)
  - On ne peut obtenir toute la liste des mot de passe
  - Sauvegarde simple par « slapcat »
  - Charge OpenLDAP plus importante / NIS (utilisation de cache)

LDAP vs NIS, LDAP vs BdD

Critère	LDAP	Base de Données
Rapport L/E	optimisé en lecture	équivalent
Extensibilité	Facile	difficile
Distribution des tables	Inhérente (hiérarchique)	complexe
Réplication	possible	possible
Modèle transactionnel	simple	avancé
Standard	oui	non (SGBD)

Critère	LDAP	NIS
Port	spécifique (389/636)	arbitraire (RPC)
Chiffrement	possible	impossible
Contrôle d'accès	oui	non
Distribution des tables	oui	non
Réplication	oui (totale ou partielle)	oui (totale)
Sémantique des recherches	avancée	simple

LDAP : un protocole normalisé !

- Le cœur de LDAP :
  - RFC 2251 : Lightweight Directory Access Protocol (v3)
  - RFC 2252 : Lightweight Directory Access Protocol (v3): Attribute Syntax
  - RFC 2253 : Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
  - RFC 2254 : The String Representation of LDAP Search Filters
  - RFC 2255 : The LDAP URL Format
  - RFC 2256 : A Summary of the X.500(96) User Schema for use with LDAPv3
  - RFC 2829Authentication Methods for LDAP
  - RFC 2830 : Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
  - RFC 3377 : Lightweight Directory Access Protocol (v3): Technical Specification

LDAP : un protocole normalisé !

- Autour de LDAP :
  - RFC 2247 : Using Domains in LDAP/X.500 Distinguished Names
  - RFC 2307 : An Approach for Using LDAP as a Network Information Service
  - RFC 2377 : Naming Plan for Internet Directory-Enabled Applications
  - RFC 2589 : Lightweight Directory Access Protocol (v3): Extensions for Dynamic Directory Services
  - RFC 2596 : Use of Language Codes in LDAP
  - RFC 2891 : LDAP Control Extension for Server Side Sorting of Search Results
  - RFC 3062 : LDAP Password Modify Extended Operation
  - RFC 3112 : LDAP Authentication Password Schema
  - RFC 2044 : UTF-8, a transformation format of Unicode and ISO 10646
  - RFC 2849 : The LDAP Data Interchange Format (LDIF) - Technical Specification
  - RFC 3384 : LDAPv3 Replication Requirements

Section : « Les services réseaux »

LDAP: Architecture

LDAP: Le modèle d'information

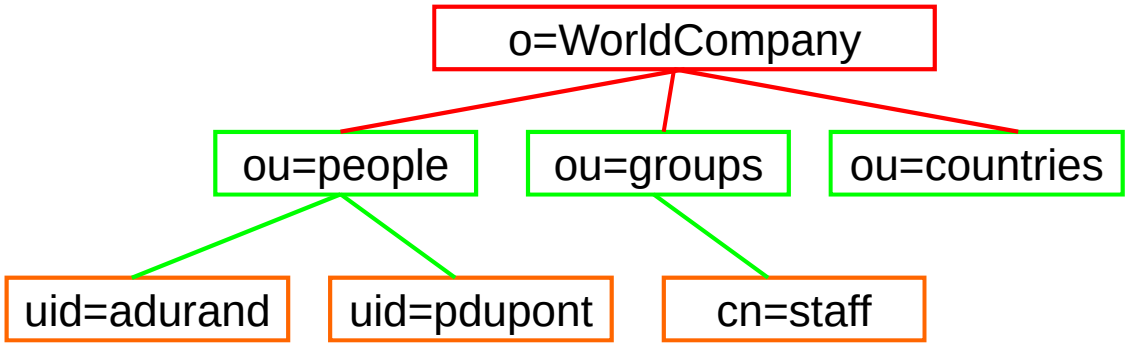
- Le modèle d'information LDAP est basé un « schéma »
- Un schéma c'est une collection de types d'entrées ou classes d'objets que l 'on peut rencontrer
- Chaque entrée fait référence à une classe d'objets, définie par un nom et une liste d'attributs (obligatoires ou optionnels)
- Les attributs sont définis par un nom d'attribut, une syntaxe et des règles de comparaison
- LDAP v3 contient sa propre description:

ldapsearch -x -H ldaps://localhost -x -b 'cn=Subschema' -s base -LLL objectclass=subschema objectClasses

LDAP: Architecture

LDAP: Le modèle de nommage

- LDAP organise les entrées dans une structure logique hiérarchique.
  - « D.I.T. Directory Information Tree »
- L 'identification d 'une entrée se fait à l 'aide du Distinguished Name (DN) :
  - ***uid=pdupont, ou=people,o=WorldCompany,c=com***
- La tendance étant de « calquer » les appellations sur le modèle DNS
  - ***uid=pdupont,ou=people,dc=WordCompany,dc=com***
- Il n 'y a pas de distinction entre nœuds et feuilles de l 'arbre
- Toutes les « intersections » sont des « entrées »



LDAP: Acronymes & définitions

- CN ➔ Common Name
- DN ➔ Distinguished Name
- RDN ➔ Relative Distinguished Name
- OID ➔ Object Identifier
- C ➔ Country (Pays)
- O ➔ Organisation
- OU ➔ Organisation Unit
- Exemple de DIT:

  o=societe.com

  RDN: ou=administratif

  DN: ou=administratif, o=societe.com

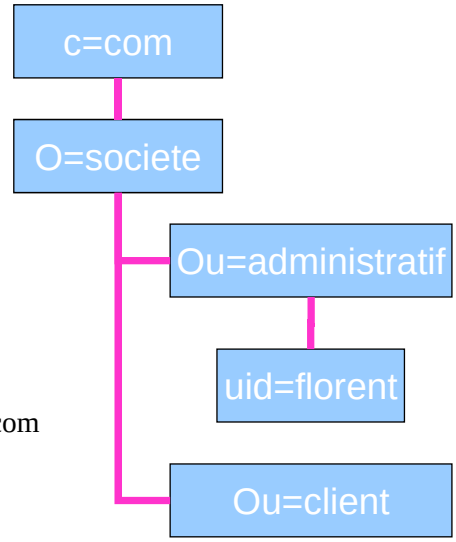
  RDN: uid=florent

  DN: uid=florent, ou=administratif, o=societe.com

  RDN: ou=client

  DN: ou=client, o=societe.com

« Directory Information Tree » (DIT) graphique

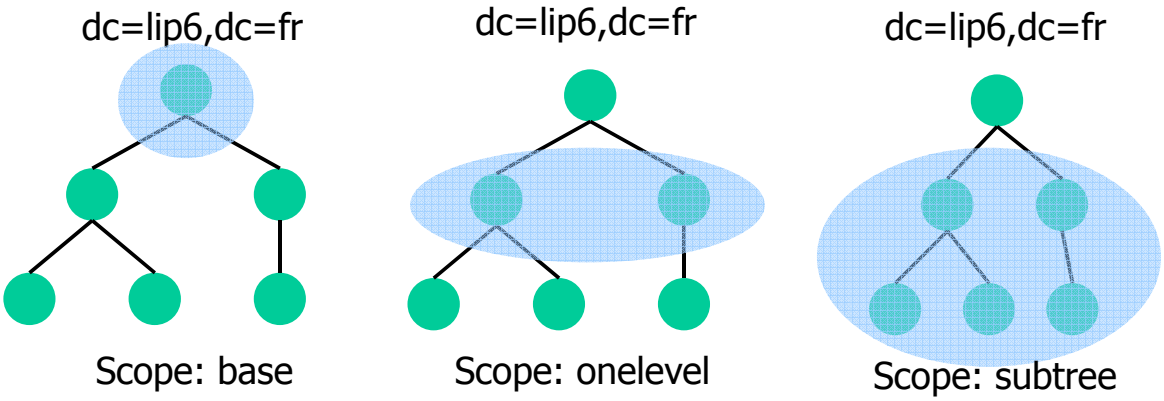


LDAP: Acronymes & définitions

- ***Classe*** : Décrit les entrées (qui sont composées d'attributs).
  - Abstraite: Définis pour les autres classes (grâce à l'héritage). Sans instance.
  - Structurelle: Définis des objets instanciés, se sont les plus classiques
  - Auxiliaire: Complète les classes structurelles (afin de ne pas modifier directement ces dernières). Pour ajouter des attributs à une classe déjà définie par exemple. Elles héritent de top.
  - L'ordre d'utilisation des classes n'est pas obligatoire avec LDAP.
- ***Attributs*** :
  - les instances de classe (les objets) ont des attributs
  - Une attribut peut être requis ou optionnel au sein d'une classe
  - Chaque attribut a un type et des règles de comparaison
- Règles comparaison d'attributs identifiés par un oid.
  - caseignorematch, CaseExactMatch, TelephoneNumberMatch
  - IntegerMatch, BouleanMatch, DNMatch, OctetStringMatch ... cf RFC 2252.

LDAP: Acronymes & définitions

- ***Scope*** : Sélectionne la profondeur de recherche dans l'arbre (DIT).
  - ***BASE***: Recherche uniquement sur la base sélectionnée (recherche sur une seule entrée)
  - ***ONELEVEL***: toutes entrées se trouvant au niveau juste inférieur à la base sélectionnée
  - ***SUBTREE***: recherche à partir de la base sélectionnée, toutes les branches en dessous.
- Exemple : « **dc=lip6, dc=fr** »



Le format LDIF

- Utilisé pour exprimer une modification du LDAP
  - Extension des fichiers .ldif
- Format :
  # commentaire
  dn: <distinguished name>
  <attrdesc>: <attrvalue>
  <attrdesc>: <attrvalue> …
- Une ligne peut être coupée en démarrant la ligne suivante par un UNIQUE caractère ESPACE ou TAB.
  dn: cn=Legond,dc=LIP6,dc=
  FR
- En cas de caractères spéciaux, on utilise un « : » après le nom d'attribue et il faut coder la chaîne en mime64
  cn:: IGJlZ2lucyB3aXRoIGEgc3BhY2U=
- Pour ajouter une URL dans un fichier
  jpegPhoto:< file://c:/chemin/monfichier.jpg
- Un fichier peut contenir plusieurs entrées LDIF séparées par une ligne vide

Structure des classes LDAP

ObjectClassDescription = "(" whitespace

numericoid whitespace *; this ObjectClass identifier*

["NAME" qdescrs]

["DESC" qdstring]

["OBSOLETE" whitespace]

["SUP" oids] *; Superior [parent] ObjectClasses*

[("ABSTRACT" / "STRUCTURAL" / "AUXILIARY") whitespace]

; default structural

["MUST" oids] *; Required AttributeTypes*

["MAY" oids] *; Optional AttributeTypes*

whitespace ")"

Créer la structure de la base

- Structure des objets de type ‘organization’:

  (2.5.6.4 NAME 'organization' DESC 'RFC2256: an organization' SUP top STRUCTURAL MUST o MAY (userPassword $ searchGuide $ seeAlso $ businessCategory $ x121Address $ registeredAddress $ destinationIndicator $ preferre dDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $ telephoneNumber $ internationaliSDNNumber $ facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $ postalAddress $ physicalDeliveryOfficeName $ st $ l $ description))

- Exemple au format LDIF:

```
# Création du domaine lip6.fr
# organization hérite de top
# o est obligatoire pour objectclass: organization
# description est optionnel
dn: dc=lip6,dc=fr
objectclass: dcObject
objectclass: organization
o: LIP6
description: Laboratoire Informatique de Paris 6
dc: lip6
```

Entité d'une organisation

- Structure des objets de type 'organizationUnit':

(2.5.6.5 NAME 'organizationalUnit' DESC 'RFC2256: an organizational unit' SUP top STRUCTURAL MUST ou MAY (userPassword $ searchGuide $ seeAlso $ businessCategory $ x121Address $ registeredAddress $ destinationIndicator $ preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $ telephoneNumber $ internationaliSDNNumber $ facsimileTelephoneNumber $ street $ postOfficeBox $ postalCode $ postalAddress $ physicalDeliveryOfficeName $ st $ l $ description)

- Exemple au format LDIF:

```
dn: ou=People,dc=src,dc=lip6,dc=fr
objectclass: organizationalUnit
description: Permanents SRC
ou: People
```


Remplacer NIS: Ajouter des utilisateurs

- Structure des objets de type 'posixAccount':

(1.3.6.1.1.1.2.0 NAME 'posixAccount' DESC 'Abstraction of an account with POSIX attributes' SUP top AUXILIARY MUST (cn $ uid $ uidNumber $ gidNumber $ homeDirectory) MAY (userPassword $ loginShell $ gecos $ description))

- Exemple au format LDIF:

```
dn: cn=Fabrice Legond,dc=src,dc=lip6,dc=fr
cn: Fabrice Legond
# LDAP 2.1 impose une unique classe ONE STRUCTURAL par élément
# impossible d'utiliser la classe account car inetOrgPerson est
# STRUCTURAL
# inetOrgPerson importe top et person
objectClass: organizationalPerson
objectclass: inetOrgPerson
objectClass: posixAccount
sn: Legond
uid: flegond
mail: flegond@system.lip6.fr
loginShell: /bin/bash
uidNumber: 1107
gidNumber: 1001
gecos: Doctorant Fabrice Legond
homeDirectory: /home/flegond
```

Remplacer NIS: Ajouter des groupes

- Structure des objets de type ‘posixGroup’:

  (1.3.6.1.1.1.2.2 NAME 'posixGroup' DESC 'Abstraction of a group of accounts' SUP top STRUCTURAL MUST (cn $ gidNumber) MAY (userPassword $ memberUid $ description))

- Exemple au format LDIF:

```
# Création de l’unité organisationnelle « groups »
dn: ou=Groups,dc=src,dc=lip6,dc=fr
objectclass: organizationalUnit
objectclass: top
description: Groupes du theme SRC
ou: Groups
# Création des groupes avec leurs membres
dn: cn=src,ou=Groups,dc=src,dc=lip6,dc=fr
cn: src
objectclass: top
objectClass: posixGroup
description: Utilisateur du groupe SRC
gidNumber: 1001
memberUid: 1100
memberUid: 1107
```


Section : « Les services réseaux »

LDAP: Manipuler les bases

Manipulation de la base LDAP : Interroger LDAP

- On peut interroger un annuaire LDAP avec un navigateur Internet. L'URL utilise la syntaxe suivante:

  protocol://server:port/base?attributs à extraire?type de scope?critères

  ➔ cf RFC 2255 (The LDAP URL Format)

- ***ldapsearch -x -v -d 4 -H ldap://eos.lip6.fr -b "" -s base filtres attributs***
  - « -x » authentification simple
  - « -v » affiche les échanges C/S
  - « -d » fixe le niveau de détail à afficher
  - « -H » (≠-h) URI du serveur
  - « -b » base dn
  - « -s » scope : base, one, sub
- Exemple:
  - ***ldapsearch –x -H ldap://eos.lip6.fr -b "" '(objectclass=*)' 'NamingContexts'***
  - ***ldapsearch -x -H ldaps://localhost -x -b "dc=src,dc=lip6,dc=fr" -s sub "uidNumber=1107"***

LDAP: Manipuler les bases

Manipulation de la base LDAP : Interroger LDAP

- Expressions de recherche
  - « = » : égalité (texte ou nombre)
  - Sous chaîne: utilisation d'* (ex: *john*)
  - >, <, >=, <=
  - Présence de l'attribut (ex: cn=*)
  - « ~= » : égalité approximative
- Opérateurs logiques
  - (&(filtre1)(filtre2)(filtre3)….) : et logique entre expressions
  - (|(filtre1)(filtre2)(filtre3)….) : ou logique entre expressions
  - (! (filtre1)) : non logique
- Exemples
  - (&(ou=SRC)(!(description=*permanent*)))
  - (!(objectClass=person))

Ajout des entrées LDIF dans la base LDAP

- 1er possibilité pour ajouter les entrées
  - Ajout via une connexion LDAP
    ldapadd -v -d 200 -x -D "cn=Manager,dc=lip6,dc=fr" -W -f myldap.ldif
- 2e possibilité pour ajouter les entrées
  - Ajout en utilisant les utilitaires d'accès directs à la base LDAP
    slapadd -v -d 200 -l myldap.ldif
  - Ne pas oublier de changer les droits/propriétaires du fichier
    chown ldap:ldap /var/lib/ldap/*
- Vérification de l'ajout: commande « ***slapcat*** »
  - Attaque directement la BD openldap
- Vérification de l'ajout: commande « ***ldapsearch*** »
  - Utilise l'interface LDAP

Manipulation de la base LDAP: Modifier un nœud

- Fichier de mise à jour d'un nœud (***updaleLegond.ldif***) :

```
dn: cn=Fabrice Legond,ou=People,dc=src,dc=lip6,dc=fr
changetype: modify

add: telephonenumber
telephonenumber: 09-12-33-11-17

-
add: secretary
secretary: faut pas rever

-
replace: mail
mail: legonda@src.lip6.fr

-
delete: secretary
# utile en cas d'attribut  multi-valué
secretary: faut pas rever
```

- Commande:
  ldapmodify -W -v -d 200 -x -H ldap://eos.lip6.fr -f updateLegond.ldif

Manipulation de la base LDAP: Effacer un nœud

- Effacer / Ajouter un nœud (***deleteAdd.ldif***):

```
# Effacer le noeud Fabrice Legond
dn: cn=Fabrice Legond,ou=People,dc=src,dc=lip6,dc=fr
changetype: delete
```

```
#Ajouter le noeud Robert Dupont
dn: cn=Robert Dupont,ou=People,dc=src,dc=lip6,dc=fr
changetype: add
objectclass: inetorgperson
cn: Robert Dupont
...
```

- Commande:

ldapmodify -W -v -d 200 -x -H ldap://eos.lip6.fr -f deleteAdd.ldif

LDAP: Sauvegarde et Restauration

- Sauvegarde / export partiel

slapcat –l /tmp/dump_db.ldif –f /etc/openldap/slapd.conf –b "dc=src,dc=lip6,dc=fr"

- Restauration
  - Ajout en utilisant les utilitaires d'accès directs à la base LDAP
    slapadd -v -d 200 -l dump_db.ldif
  - Ne pas oublier de changer les droits/propriétaires du fichier
    chown ldap:ldap /var/lib/ldap/*

Identification / Authentification

Network Information Service

LDAP: Principes

LDAP: Architecture

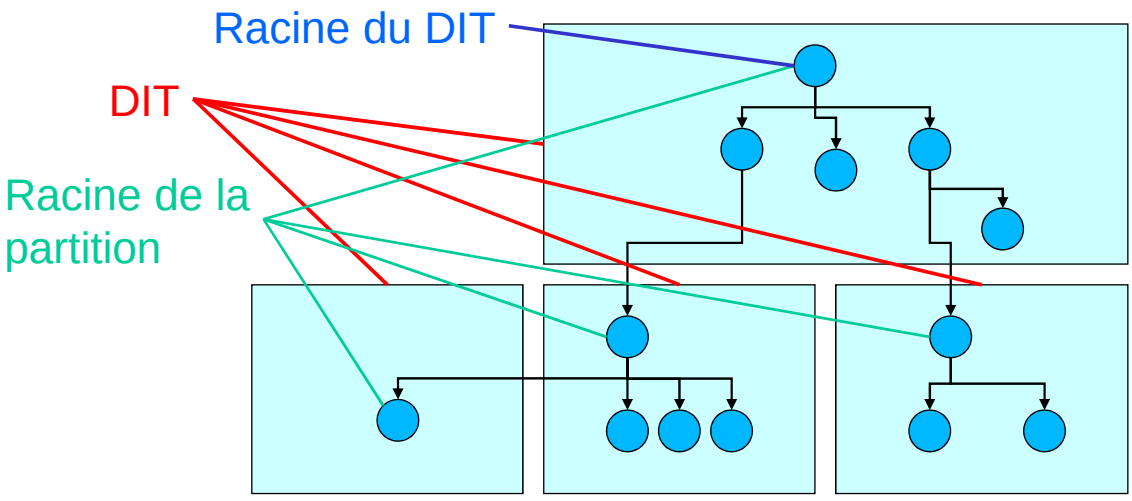
LDAP: Manipuler les bases

LDAP: Distribution et redondance

Un exemple: OpenLDAP

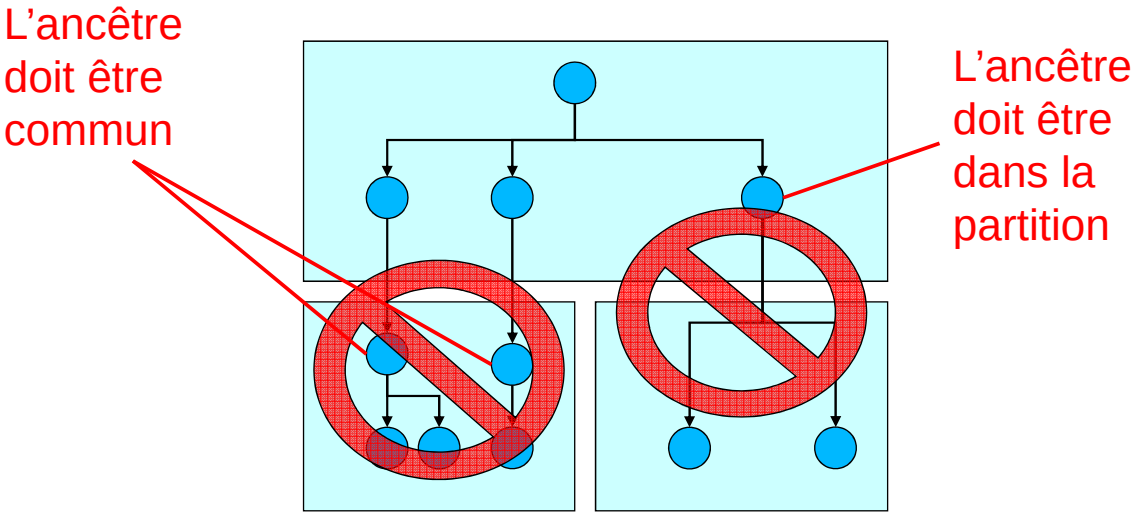
Répartition: Partitions

- Le DIT peut être découpé en plusieurs morceaux
- Les morceaux peuvent être répartis sur plusieurs machines
- Les morceaux sont appelés des « partitions »



Répartition: Partitions

- Toutes les partitions ne doivent avoir qu'un seul ancêtre
- Cet ancêtre doit être dans la partition



Répartition: Liens entre les partitions

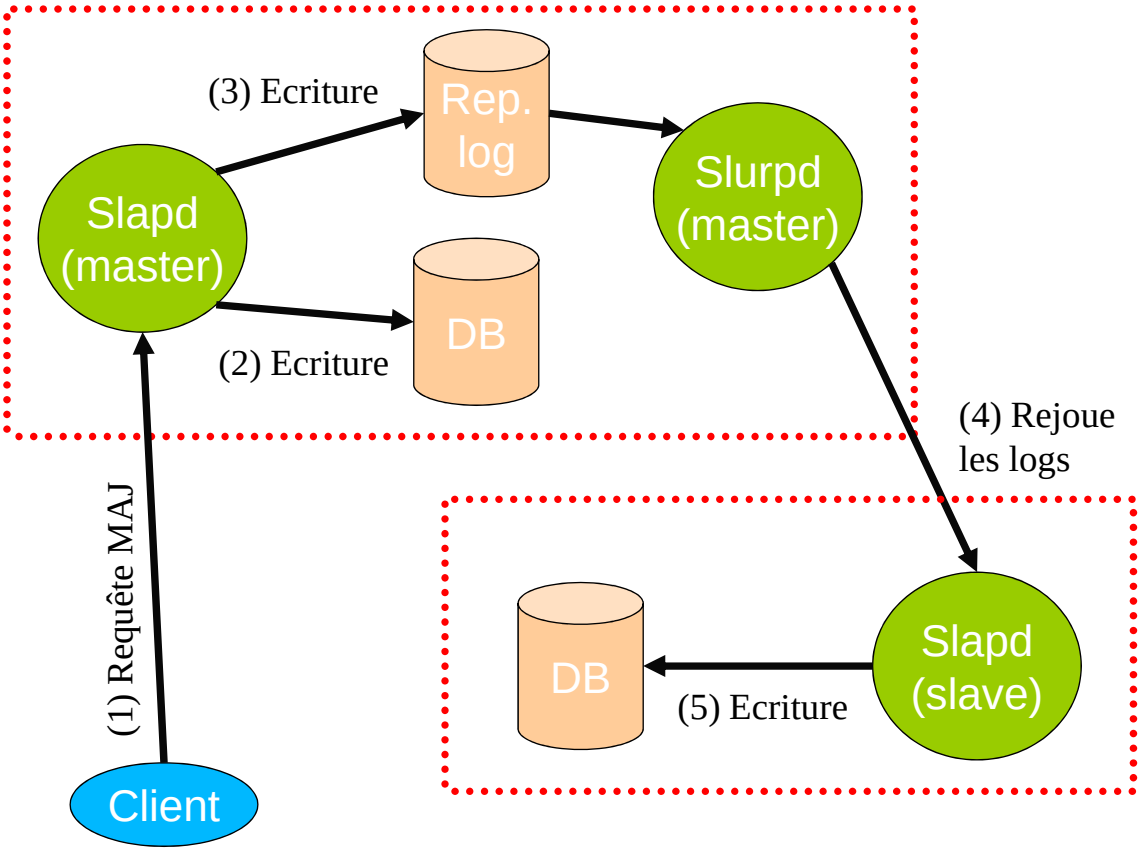
- Lien montant
  - entre une partition LDAP enfant et une partition parente
  - Option de configuration dans le serveur enfant
- Lien descendant
  - Utilisation d'une entrée « référé »

```
dn: ou=netgroups,dc=src,dc=lip6,dc=fr
objectClass: referral
objectClass: extensibleObject
dc: subtree
#transfert vers la partition enfant
ref: ldap://src.lip6.fr/ou=netgroups,dc=src,dc=lip6,dc=fr/
```

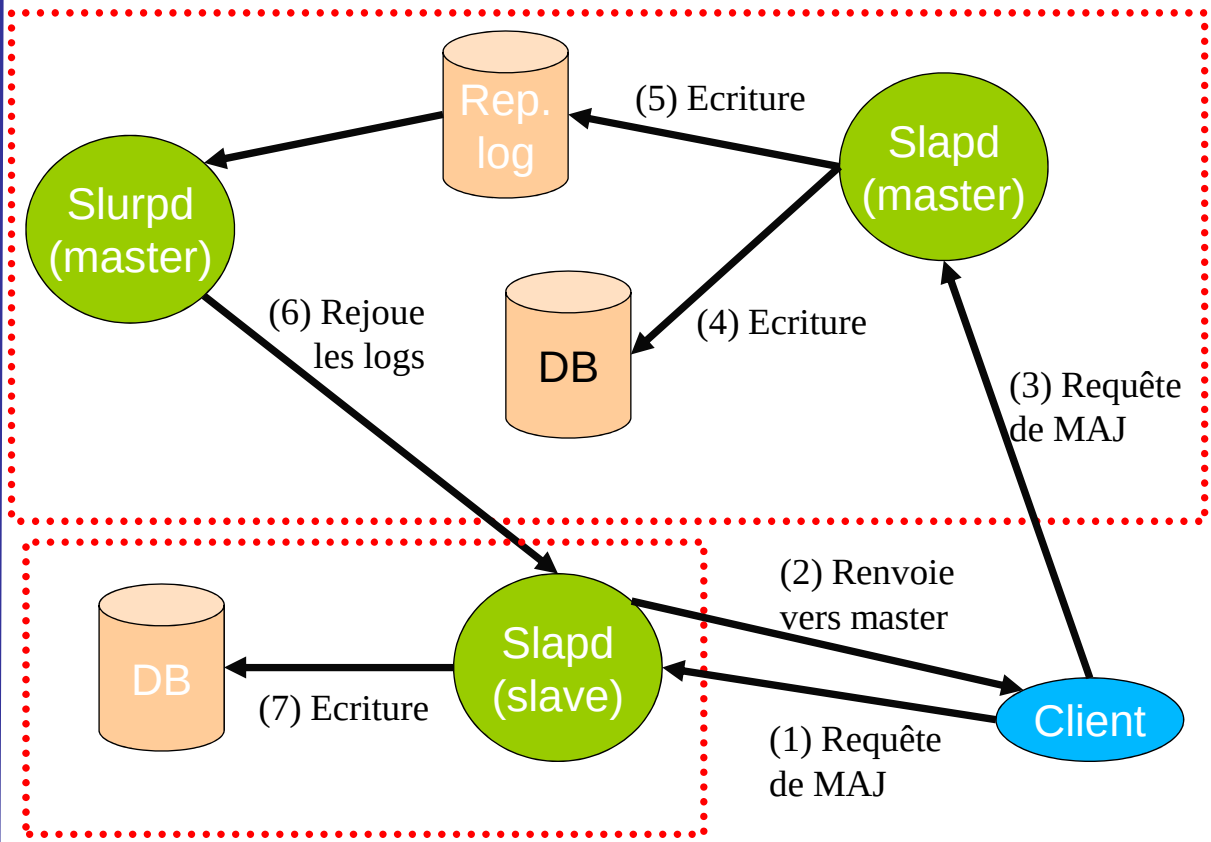
Redondance

- Pour la redondance un serveur maître ldap peut répliquer des changements sur un ou plusieurs serveurs esclaves
- La redondance utilise un démon nommé slurpd
  - ***slurpd*** = « Standalone LDAP Update Replication Daemon »
  - Les changement sont enregistrés dans des logs
  - Les logs sont rejoués sur les serveurs esclaves
  - Les logs ayant provoqués des erreurs sont conservés
  - Une réplication peut se propager sur plusieurs serveurs

Redondance: scénario 1



Redondance: scénario 2



Redondance: compléments

- En cas d'échec de la synchronisation
  - Les opérations non réplicables sont stockées dans un fichier de log de rejet
  - slurpd peut être exécuter en mode oneshot (-o) pour rejouer ce fichier de log
  - En cas de plantage de l'esclave, la synchro se fait au redémarrage
- Comportement ***slurpd***
  1. Fichier de log vide ou inexistant ➔ sommeil
  2. Check périodique du fichier. Vide ou inexistant ➔ retour en 1
  3. En cas de présence, slurpd verrouille, copie les données
  4. fork ou création de thread
  5. Bind des threads ou processus
  6. Envoie des données. Erreur ➔ écriture dans le log de rejet
  7. Fin des threads/processus, retour en 1

Section : « Les services réseaux »

Identification / Authentification

Network Information Service

LDAP: Principes

LDAP: Architecture

LDAP: Manipuler les bases

LDAP: Distribution et redondance

Un exemple: OpenLDAP

Configuration d'un serveur OpenLDAP

- Fichiers de configuration dans « ***/etc/openldap*** » ou « ***/etc/ldap*** »
- « ***etc/openldap/sldap.conf*** »
  - Configuration du serveur LDAP
- « ***/etc/openldap/slapd.access.conf*** »
  - Configuration du contrôle d'accès au server
- « ***/var/lib/openldap*** »
  - Contient les bases (en général au format berkeley bd)
  - PERFORMANCE: fichier « DB_CONFIG » , utilitaires dbm (db_stat)
- Utilitaires de manipulation d'OpenLDAP (pour les admins du serveur)
  - « ***slapcat*** » ➔ afficher le contenu de la base
  - « ***slapadd*** » ➔ ajouter des enregistrement à la base
  - « ***slaptest*** » ➔ tester les fichier de configuration
  - « ***slappasswd*** » ➔ générer le mot de passe de la base
  - « ***slapindex*** » ➔ régénérer les index LDAP
  - « ***slapdn -f ./slapd.conf -v DN dc=lip6,dc=fr*** » ➔ vérifier la structure du dn lip6.fr

Un exemple: OpenLDAP

Configuration d'un serveur OpenLDAP

- Le fichier « ***/etc/ldap.conf*** » est un fichier de configuration système (timeout, l'adresse de base)
- Les schémas sont
  - dans « ***/etc/openldap/schemas*** » pour openldap 2.x
  - dans « ***slapd.at.conf*** » et « ***slapd.oc.conf*** » pour openldap 1.x
- Extrait de « ***slapd.conf*** »:

```
# impose le contrôle de la base
schemacheck on
# importation de la description des schémas
include   /usr/share/openldap/schema/core.schema
include ...
include   /etc/openldap/slapd.access.conf
pidfile   /var/run/ldap/slapd.pid
# arguments par défaut pour le démarrage du serveur ldap
argsfile     /var/run/ldap/slapd.args
# Serveur utilisé en cas de requête au domaine supérieur
referral ldap://root.openldap.org
defaultsearchbase=''dc=src,dc=lip6,dc=fr''
threads 64
... ...
```


Un exemple: OpenLDAP

Configuration: directives Require / Disallow

- Interdiction via l'option « ***disallow*** »
- Obligation via l'option « ***require*** »
- Options pour require/disallow:
  - « ***disallow bind_v2*** » ➔ requête LDAPv2
  - « ***disallow bind_anon*** » ➔ requêtes anonymes
  - « ***disallow bind_simple*** » ➔ authentification simple
  - « ***require bind*** » ➔ bind requis
  - « ***require LDAPv3*** » ➔ protocole v3 obligatoire
  - « ***require SASL*** » ➔ SASL requis
  - « ***require none*** » ➔ rien n'est requis

Configuration du niveau de log

- L'option « ***loglevel*** » permet de choisir la quantité de log
- Les log sont envoyés à syslog sur la facility « ***LOCAL4*** »
- Pour changer de facility ➔ paramètre « -l » de l'app. « ***slapd*** »
- Valeur de l'option:

-1	tous les messages
1	trace function calls
2	debug packet handling
4	heavy trace debugging
8	connection management
16	print out packets sent and received
32	search filter processing
64	configuration file processing
128	access control list processing
256	stats log connections/operations/results
512	stats log entries sent
1024	print communication with shell backends
2048	entry parsing

Configuration de la redondance

- Au départ, maître et esclave doivent démarrer avec une base IDENTIQUE !
- Sur le serveur maître, pour chaque esclave,
  - il faut une entrée « ***replica*** »
  - Il faut une entrée « ***replogfile filename*** »
  - Exemple d'ajout dans slapd.conf:

```
replica    host=ldap-slave1.lip6.fr:389
           binddn=''cn=root,dc=src,dc=lip6,dc=fr''
           bindmethod=simple credentials=monmotdepasse

replogfile      /var/spool/ldap/replog
```

Un exemple: OpenLDAP

Contrôle d'accès (ACL)

- ACL = « Access Control Lists »
- Les ACL permet de définir qui à les droits de consultation, modification, de chaque partie du DIT.
- Attention :
  - Les ACL gèrent le contrôle d'accès (gestion des droits)
  - Les ACL ne gèrent pas la confidentialité !! (pas de cryptage)
  - Il faut activer le TLS sur les sockets pour la confidentialité
- Il n'y a pas de standardisation des ACL
  - Un draft rfc est en cours (**draft-ietf-ldapext-acl-model)**
  - Une standardisation LDIF aussi
  - Un peu abandonné

Un exemple: OpenLDAP

ACL OpenLDAP : Contrôle d'accès

- La structure des ACL
  - element_dit ➔ Elément du DIT sur lequel s'applique la règle
  - liste_attributs ➔ Attributs de l'élément du DIT sur lesquels s'appliquent la règle
  - identite_entite ➔ identifiant de l'entité autorisée à accéder
  - type_acces ➔ droits octroyés à l'entité
  - controle ➔ modification du comportement dans la lecture des règles
- Il peut y avoir de multiples lignes « by … » pour chaque règle ACL

```
access to <element_dit> <liste_attributs>
        [ by <identite_entite> <type_acces> [ <controle> ] ]+
```

Un exemple: OpenLDAP

Format des ACL : Eléments et attributs ciblés

```
access to <element_dit> [ <liste_attributs> ]
        [ by <identite_entite> <type_acces> [ <controle> ] ]+
```

- **element_dit**
  - Elément du DIT sur lequel s'applique la règle
  - ***** : désigne tous les éléments de l'arbre
  - **dn.dn_style="patron"** : permet de désigner une partie du DIT
    - ✔dn_style : regexp, base, exact, one, subtree, children
    - ✔L'expression "patron" désigne effectivement l'élément
    - ✔*dn="ou=People,dc=somedomain,dc=com"*
- **liste_attributs**
  - Attributs de l'élément du DIT sur lesquels s'appliquent la règle
  - Une liste d'attributs séparés par des virgules
    - ✔*attrs=userPassword*

Un exemple: OpenLDAP

Format des ACL : Identifiant de l'ayant droit

```
access to <element_dit> [ <liste_attributs> ]
        [ by <identite_entite> <type_acces> [ <controle> ] ]+
```

- **identite_entite**
  - ***** : désigne tout le monde
  - ***anonymous*** : toute personne non authentifiée.
    Note: toute personne est "anonymous" tant qu'elle ne s'est pas authentifiée (bind)
  - ***users*** : l'accès est autorisé à tout utilisateur authentifié
  - ***self*** : l'identité de l'entité est celle désigné par <element_dit> lui-même
  - ***dn[.<dnstyle>]=<patron>*** : identifiant de l'utilisateur de structure identique à celui trouvé dans <element_dit>
  - ***group[.<style>]=<patron>*** : un groupe d'entité. Le "patron" doit désigné un élement de classe "groupofNames".
  - ***peername[.<style>]=<patron>*** : l'identifiant est une adresse IP ou un nom de machine

Format des ACL : type d'accès

```
access to <element_dit> [ <liste_attributs> ]
        [ by <identite_entite> <type_acces> [ <controle> ] ]+
```

- **type_acces**
  - ***none*** : aucun accès (par ex. pour faire des exceptions dans un groupe)
  - ***auth*** : accès à l'attribut pour permettre l'authentification (bind)
  - ***compare*** : Autorise la comparaison d'attributs avec l'attribut sur lequel porte l'ACL
  - ***search*** : Autorisation de faire une recherche sur un attribut (sans pouvoir le lire)
  - ***read*** : Permission de lire le contenu de l'attribut
  - ***write*** : Permission de modifier l'attribut
- les autorisations sont cumulatives :
  - Chaque niveau inclus tous les précédents

Format des ACL : type d'accès

```
access to <element_dit> [ <liste_attributs> ]
        [ by <identite_entite> <type_acces> [ <controle> ] ]+
```

- Permet le contrôle de l'évaluation des règles
  - Par défaut les règles sont lues dans l'ordre jusqu'à trouver une règle qui concorde (match)
  - **IMPORTANCE DE L'ORDRE DES ACLS !!!**
- **« controle » :**
  - **stop :** action par défaut. Arrêt dès concordance avec une des règles « by… »
  - **continue :** force l'évaluation des autres lignes « by … » de l'ACL
  - **break :** force l'arrêt de l'évaluation des autres lignes « by … » de l'ACL

Un exemple: OpenLDAP

Format des ACL : exemples

- Accès sur l'attribut "userPassword"
  - Autorise la modification de mdp par l'entité elle-même
  - Autorise l'utilisation du mdp pour les anonymes pour l'authentification
  - Interdit tous les autres accès

```
access to attr=userPassword
     by self write
     by anonymous auth
     by * none
```

- Accès sur tous les autres éléments :
  - Autorise tous les droits par les entités elles-mêmes
  - Les utilisateurs authentifiés peuvent lire les données
  - Interdit tous les autres accès

```
access to *
     by self  write
     by users read
     by * none
```

- **Erreur en cas de changement de l'ordre !!!**

Un exemple: OpenLDAP

Format des ACL : exemples

- Accès sur l'attribut "userPassword"
  - A vous ….

```
Access to *
     by self                  write
     by users                 read
     by peername=127.0.0.1    read
     by peername=10.0.11.*    read
     by *                     none
```

- Pour faire un ET LOGIQUE (AND) sur des ACLs
  - **il faut mettre toutes les conditions sur la même ligne !!!**

```
access to *
     by self peername=132.227.64.* write
     by users read
     by * none
```

Format des ACL : exemples

- Contrôle d'accès utilisant les regexp :

```
Access to dn="^.*,uid=([^,]+),ou=People,(.*)$"
     by dn="uid=$1,ou=People,$2"              write
     by group="cn=LDAPaccess,ou=LDAPauth,$2" read
     by *
```

- Utilisation des () et des $x
  - Dans notre exemple le dn de <element_dit> contient deux ()
  - Regexp « [^,]+ » ➔ tout caractère sauf « , »

Configuration de la redondance

- Sur chaque esclave ajouter une ligne updatedn
  - Dans le fichier ***sladp.conf***
  - Dont la valeur est identique au binddn à l'option replica

    updatedn ''cn=root,dc=src,dc=lip6,dc=fr''
  - Indiquer qui est le serveur maître

    updateref ''ldap://eos.lip6.fr''
- Synchronisation initiale Maître/Esclave
  - Vérifier les droits d'accès des comptes utilisés
  - Autoriser des requêtes illimitées pour ces comptes
  - Fixer les default base, updatedn, updateref, replica, …
  - Vérifier la cohérence des schémas
  - Copier les bases

    ldapsearch –LLL –D ''binddn'' –w ''bind pwd'' ''objectclass=*'' | slapadd –n 1
  - Démarrer le slapd esclave, redémarrer le serveur maître

Un exemple: OpenLDAP

Intégration avec l'authentification linux

- Migration des base nis ou ***/etc/passwd***
  - Il existe des utilitaires. Exemple: script perl ***migrate_nis*.pl***
- Intégration à nss
  - Installation du module nss_ldap
  - Modification du fichier ***/etc/nsswitch.conf***

```
passwd:    files nis ldap
shadow:    files nis ldap
group:     files nis ldap
automount: files nis ldap
```

  - Création d'un fichier « ***/etc/ldap.conf*** »

```
host eos.lip6.fr ldap-slave1.lip6.fr
base dc=src,dc=lip6,dc=fr
ldap_version 3
port 389
binddn cn=binduser,ou=system,dc=src,dc=lip6,dc=fr
 bindpw secret
pam_password crypt
#filtre spécifique pour l'authentification
pam_filter
```


Un exemple: OpenLDAP

Intégration avec l'authentification linux

- Intégration au modèle PAM
  - Installation du module ***pam.d***
  - Utilisation du fichier ***/etc/ldap.conf***
  - Configuration de la pile d'authentification PAM
  - Copier les fichiers ***pam.d/**** dans le répertoire « ***/etc/pam.d*** »
  - Exemple (***/etc/pam.d/login***):

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_nologin.so
auth       sufficient   /lib/security/pam_ldap.so
auth       required     /lib/security/pam_unix_auth.so try_first_pass
account    sufficient   /lib/security/pam_ldap.so
account    required     /lib/security/pam_unix_acct.so
password   required     /lib/security/pam_cracklib.so
password   required     /lib/security/pam_ldap.so
password   required     /lib/security/pam_pwdb.so use_first_pass
session    required     /lib/security/pam_unix_session.so
#session    optional     /lib/security/pam_console.so
```

LDAP

R T F M

LDAP

- « cat /etc/nsswitch »

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
protocols:  files nis ldap
rpc:        files
services:   files nis ldap
netgroup:   files nis ldap
automount:  files nis ldap
```

- Comptes locaux
  - Démon de cache nscd
- Pas de NIS défini : « /usr/sbin/yptest »

```
Test 1: domainname
ERROR: domainname is not set!
```

LDAP

5 minutes pour découvrir Jussieu …

- « cat /etc/pam.d/system-auth »

```
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        sufficient    pam_ldap.so use_first_pass
auth        required      pam_deny.so

account     required      pam_unix.so broken_shadow
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     [default=bad success=ok user_unknown=ignore] pam_ldap.so
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so md5 nullok try_first_pass use_authtok
password    sufficient    pam_ldap.so use_authtok
password    required      pam_deny.so

session     required      pam_limits.so
session     required      pam_unix.so
session     optional      pam_ldap.so
```


LDAP

5 minutes pour découvrir Jussieu …

- « cat /etc/ldap.conf »

```
host ldap.ufr-info-p6.jussieu.fr
base ou=nis-infop6,ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr
scope sub
pam_filter  objectClass=posixAccount
pam_login_attribute uid
pam_password md5
pam_password_prohibit_message Please visit
http://www.annaire.upmc.fr/upmc/submit_password.upmc to change
your password.
nss_base_passwd ou=Users,ou=nis-infop6,ou=ARI,dc=ufr-info-
p6,dc=jussieu,dc=fr?one
nss_base_shadow ou=Users,ou=nis-infop6,ou=ARI,dc=ufr-info-
p6,dc=jussieu,dc=fr?one
nss_base_group ou=Groups,ou=nis-infop6,ou=ARI,dc=ufr-info-
p6,dc=jussieu,dc=fr?one
ssl no
```

LDAP

5 minutes pour découvrir Jussieu …

- « ldapsearch -x -H ldap://ldap.ufr-info-p6.jussieu.fr -x -b 'cn=Subschema' -s base -LLL objectclass=subschema objectClasses | less »

```
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1003 NAME 'identiteANNUAIRE'
SUP top STRUCTURAL MUST ( cn $ uid $ userPassword $ idANNUAIRE ) )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1002 NAME 'identiteDBUFR' SUP
top AUXILIARY MAY idDBUFR )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1008 NAME 'identiteARI' SUP top
AUXILIARY MAY idARI )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1001 NAME 'identiteUPMC' SUP top
AUXILIARY MAY uidInterne )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1010 NAME 'comptesMachine' SUP
top STRUCTURAL MUST name MAY ( uid $ userPassword $ rattachement $
localAcc ount ) )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1009 NAME 'personneARI' SUP top
STRUCTURAL MUST cn MAY idARI )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.10.1007 NAME 'personneDBUFR' SUP
top STRUCTURAL MUST cn MAY ( sn $ givenName $ email $ uid $ userPassword $
idDBUFR $ categorieDBUFR ) )
objectClasses: ( 1.3.6.1.4.1.7135.1.3.201.1.4.1.53 NAME 'exterieurCompte' SUP
exterieur STRUCTURAL MUST ( employeeType $ gereParHarpege ) MAY
identifiantHarpege )
….
```


LDAP

5 minutes pour découvrir Jussieu …

- « ldapsearch -x -H ldap://ldap.ufr-info-p6.jussieu.fr '(objectclass=*)' 'NamingContexts' | less »

```
# ufr-info-p6.jussieu.fr
dn: dc=ufr-info-p6,dc=jussieu,dc=fr
# Personnes, ufr-info-p6.jussieu.fr
dn: ou=Personnes,dc=ufr-info-p6,dc=jussieu,dc=fr
# ARI, ufr-info-p6.jussieu.fr
dn: ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr
# copieManager, ufr-info-p6.jussieu.fr
dn: cn=copieManager,dc=ufr-info-p6,dc=jussieu,dc=fr
# dbufr, ARI, ufr-info-p6.jussieu.fr
dn: ou=dbufr,ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr
# Groupes, dbufr, ARI, ufr-info-p6.jussieu.fr
dn: ou=Groupes,ou=dbufr,ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr
# Users, dbufr, ARI, ufr-info-p6.jussieu.fr
dn: ou=Users,ou=dbufr,ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr
…
# nis-tique, XX_artieres, Users, Comptes, ARI, ufr-info-p6.jussieu.fr
dn: name=nis-tique,idARI=XX_artieres,ou=Users,ou=Comptes,ou=ARI,dc=ufr-info-p6
 ,dc=jussieu,dc=fr
```

LDAP

5 minutes pour découvrir Jussieu …

- « **ldapsearch -x -H ldaps://ldap.ufr-info-p6.jussieu.fr -b "ou=nis-infop6,ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr" -s sub** »

  …
  # 2101351, Users, nis-infop6, ARI, ufr-info-p6.jussieu.fr
  dn: idARI=2101351,ou=Users,ou=nis-infop6,ou=ARI,dc=ufr-info-p6,dc=jussieu,dc=fr
  objectClass: top
  objectClass: person
  objectClass: posixAccount
  objectClass: shadowAccount
  objectClass: identiteARI
  objectClass: identiteDBUFR
  objectClass: identiteUPMC
  sn: NOM
  cn: NOM PRENOM
  description: Etudiant ufr info
  uidNumber: 2101351
  gidNumber: 2101351
  homeDirectory: /users/Etu1/2101351
  loginShell: /bin/bash
  gecos: NOM PRENOM
  idARI: 2101351
  idDBUFR: 2101351
  uidInterne: 2101351
  uid: 2101351
  shadowExpire: 13107
  …

- « **getent passwd 2101351** »
  2101351:x:2101351:2101351:DEJOUX MICHEL:/users/Etu1/2101351:/bin/bash
- Vous tenterez vous-même les écritures …

LDAP

Pour terminer les outils graphiques

- LDAP Browser/Editor est un butineur graphique LDAP en Java
- JXplorer est un autre butineur graphique LDAP en Java
- GQ est un butineur graphique s'appuyant sur GTK
- Luma est un gestionnaire graphique, basé sur python-ldap, extensible via des plugins
- Frood est un butineur Gtk-Perl/PerLDAP
- phpLDAPadmin est un client web d'administration/exploration LDAP
- KLDAP est un client ldap pour KDE
- ldapvi ou comment modifier des entrées LDAP avec un éditeur de texte
- web500gw est une passerelle HTTP/LDAP autonome
- web2ldap est une passerelle HTTP/LDAP en Python multi-plateformes (Unix/Windows)
- Netscape Communicator (avec en particulier le carnet d'adresses) est un bon client LDAP (en attendant Mozilla)
- LAST est un outil d'administration d'annuaire en http à base de scripts Perl et de CGI
- Calendra Directory Manager® est un gestionnaire de contenu d'annuaires orienté métiers