LDAP (*Lighweight Directory Access Protocol*)

Anne WEI
CNAM Paris

Bibliographie

- K. Zeilenga, «LDAP: Technical Specification Road Map», RFC 4510, IETF, juin 2006
- J. Sermersheim, «LDAP: The Protocol», RFC 4511, IETF, juin 2006
- K. Zeilenga, «LDAP: Directory Information Models », RFC 4512, IETF, juin 2006
- R. Harrison, «LDAP: Authentication Methods and Security Mechanisms», RFC 4513, IETF, juin 2006
- K. Zeilenga, «LDAP: String Representation of Distinguished Names», RFC 4514, IETF, juin 2006
- M. Smith and al, «LDAP: String Representation of Search Filters», RFC 4515, IETF, juin 2006
- M. Smith and al, «LDAP: Uniform Resource Locator», RFC 4516, IETF, juin 2006
- S. Legg, «LDAP: Syntaxes and Matching Rules », RFC 5617, IETF, juin 2006

Plan

1. Introduction
2. Structure de l'annuaire
3. Protocole et Opérations principales
4. Localisation URL
5. Modèle de l'information
6. LDAP en pratique
7. Conclusion

Introduction (1)

- **DAP** (*Directory Access Protocol*) est un protocole de *gestion répartie de l'annuaire X.500*.

- **LDAP** (*LightWeight Directory Access Protocol*) est un protocole *simplifié* dérivé de DAP permettant un accès à un annuaire en mode client/serveur à l'aide des protocoles TCP/IP.

- **X.500** est l'ensemble de technologies de services «annuaires» normalisées par CCITT (*Comité Consultatif International Téléphonique et Télégraphique)* en 1988 dans le but d'utiliser les annuaires téléphoniques via un réseau
  - X.500 repose sur l'architecture en couche ISO (*International Organization for Standardization).*
  - Maintenant les protocoles TCP/IP sont également utilisés par X.5000

Introduction (2)

- Créé par Tim Howes, Steves Kille and Wengyik Yeong, la version ***LDAPv1*** a été née en 1993. En 1997, la version 3 ***LDAPv3*** a été créée par Tim Howes et Steves Kille. La version 3 de LDAP devient le standard d'IETF (*Internet Engineering Task Force*).
- LDAPv3 permet d'accéder à un annuaire à distance par une connexion sécurisée
- A part du service annuaire (pages blanches, par exemple), LDAP peut servir comme une passerelle entre applications. C'est-à-dire, le protocole permet l'échange de données entre applications incompatibles, par exemples les carnets d'adresses Netscape Communicator et Microsoft Outlook

Concepts

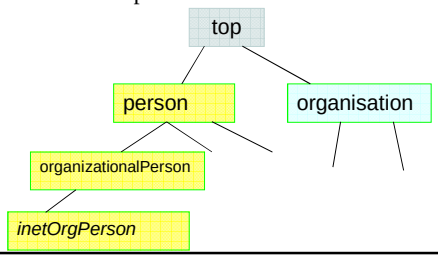
- Les concepts de LDAP consistent en
- Un protocole (RFC 4511) permettant d'accès à l'annuaire
- Un modèle d'information (RFC 4512) qui définit le type des informations
- Une authentification et des mécanismes de sécurité (RFC 4513)
- Un nommage de la structure de l'annuaire (RFC 4514 et RFC4519)
- Un format du filtre - chaîne de caractères (RFC 4515)
- Une localisation de l'annuaire – URL (RFC 4516)
- Un modèle fonctionnel – syntaxe et règles (RFC 4517)
- Un échange de données par LDIF - *LDAP Data Interchange Format* (RFC4525)
- Des API pour faciliter le développement

Plan

1. Introduction
2. Structure de l'annuaire
3. Protocole et Opérations principales
4. Localisation URL
5. Modèle de l'information
6. LDAP en pratique
7. Conclusion

Structure de l'annuaire

- La structure de l'annuaire est une architecture de l'*arborescence*. Elle s'appelle **DIT** (*Directory Information Tree).* En générale , elle se limite à une commuté *locale*.
  - chaque somme est une entrée qui représente une classe d'objet
  - une entrée contient une série d'affectations d'attributs ; un attribut associe à une (ou plusieurs) valeur(s)
  - chaque entrée hérite des attributs de son père
  - la classe d'objet s'appelle *objectclass.* Il faut obligatoirement indiquer la parenté de la classe d'objet à partir de l'entrée top.



Un exemple

La classe d'objet *inetOrgPerson* à la filiation suivante :
objectclass: *top*
objectclass: *person*
objectclass: *organizationalPerson*
objectclass: *inetOrgPerson* (*coordonnées d'internet)*

La classe *person* a les attributs suivants : *Nom*, *Prénom*, *Profil* et *motdepasse*
La classe *organizationalPerson* rajoute les attributs suivants : *établissement*, *garde*, *adresse* et *numéro-téléphone*
La classe *inetOrgPerson* rajoute des attributs comme : *mail*, *@IP et ID*

Par exemple : Wei Anne, depINFO, weianne, CNAM,PR,33-1-25,2524,anne.wei@cnam.fr, 192.168.123.123,456789

Nommage d'attributs (1)

- LDAP (RFC 4519) définit les types (noms) des attributs. Un attribut définit le chemin dans la structure arborescente de l'annuaire.
- L'attribut **DN** (*Distinguished Name*) est l'identifiant unique d'une entrée LDAP.
- L'attribut **RDN** (*Relative Distinguished Name*) est une partie d'une entrée DN.
- un exemple : DN = */user/bar/myfile.txt* (le chemin entier)
  RDN = *myfile.txt* (le chemin relative)
- La classe **objectclass** désigne les attributs d'une entrée

- Les objets et leurs attributs sont normalisés. Ils sont tous référencés par un *object identifier* (OID) unique tenu à jour par IANA (*nternet Assigned Numbers Authority*)

Nommage d'attributs (2)

- L'attribut **cn** (*commonName*) est le nom d'une entrée. Par exemple, « Marty Smith » . Il permet d'indiquer le chemin d'accès à celle-ci depuis le sommet de l'arbre
- L'attribut **gn** (*givenName*) est le prénom. «André » ou « Charles», Par exemple
- L'attribut **sn** *(surName)* est le nom de famille. Par exemple, « Smith »
- L'attribut **dc** *(domainComponent)* est une chaîne des caractères concernant un composant. Par exemple : «example» ou «com». *Attention* : « example.com » n'est pas validé, car il contient multi-domaines.
- L'attribut **ou** *(organizationalUnitName)* indique le nom de l'unité organisationnelle. « Finance» et « Ressources Humaines» par exemple
- L'attribut **description** est une note. Par exemple, « réunion à 14h »
- L'attribut **o** *(organizationName)*

Nommage d'attributs (3)

- le choix du suffixe est très important, car il permet de localiser (trouver) facilement un serveur LDAP (localement ou globalement)

- Dans la norme X500, le niveau *top* est le pays et vient ensuite le nom de l'organisation, ce qui donne par exemple comme suffixe : **o**=cnam, **c**=fr. Cependant, chaque établissement peut choisir son propre suffixe. Ceci peut créer un conflit entre deux annuaires.

- LDAPv3 a été normalisée en 2006, après le standard DNS qui se base sur un nommage «mondial». Par conséquent, le choix du suffixe LDAP devrait dépendre de la technique DNS
  - l'attribut *Domain Component* (dc) : dc=cnam, dc=fr
  - l'enregistrement du type SRV (*Service Record*) du DNS permet de localiser le serveur LDAP

    _ldap._tcp.cnam.fr. durée IN SRV 389 ldap.cnam.fr

Objets particuliers

- Deux objets sont abstraits particuliers : *aliase* et *referral*
- *aliase* permet à une entrée de l'annuaire de pointer vers une autre entrée (la même entrée)
- *Referral* permet à une entrée de l'annuaire de pointer vers un autre annuaire.
- L'attribut *aliasObjectName* de l'objet *alias* a pour le DN de l'entrée pointée.

Un exemple d'attributs

- Un exemple :

dn: **cn**=John Doe, **dc**=example, **dc**=com ← Le père de l'entrée
cn: John Doe ← Nom relatif
givenName: John
sn: Doe
description : professor
manager: cn=Barbara Doe, **dc**=example, **dc**=com
objectClass:inetOrgPerson
objectClass:organizationalPerson
objectClass:person
objectClass:top

Les attributs de l'entrée

- dans cet exemple, *example.com* est un sous-arbre
- un attribut peut avoir deux valeurs « équivalentes » ; **givenName**: John , JOHN

Création et Syntaxe d'attribut

- Si les attributs/classes définis par le standard LDAP ne suffisent pas, il nécessite de créer des nouveaux attributs/classes.
- La création d'attribut peut soit un nouveau attribut, soit un attribut qui hérite d'un attribut existant
- La création de classe peut soit une nouvelle classe, soit une classe qui hérite d'une classe existante

- La syntaxe d'attribut utilse l'ASN.1 (*Abstract Syntax Notation One*) – X.501. Quelques exemples :
- *bin* : les données binaires
- *ces* (case exact string) : les textes sont considérés en cas d'une comparaison ; *cis* (case ignore string) concernant le cas au contraire
- *tel* : un texte

Schéma

- Le schéma du LDAP décrit les classes d'objets (*objectclass*), leurs types d'attributs et leur syntaxe.

- La définition de schéma dépend de logiciels et des standards. Par exemple,
- Le logiciel OpenLDAP (Windows, OS MAX, Linux) utilise un fichier de configuration (slapd.conf) pour définir le schéma
- Le schéma utilisé par LDAPv3 est localisé par l'attribut opérationnel *subschemaSubentry* de l'entrée *rootDSE* (Directory Service Entry). La valeur de cet attribut est une liste de DNs qui pointent vers des entrées et la classe d'objet « subschema».

Plan

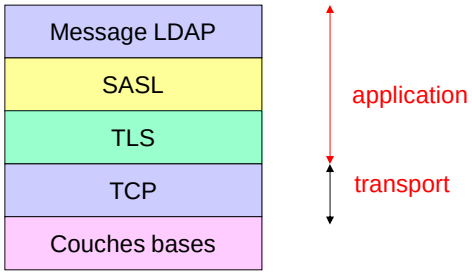
1. Introduction
2. Structure de l'annuaire
3. Protocole et Opérations principales
4. Localisation URL
5. Modèle de l'information
6. LDAP en pratique
7. Conclusion

Protocole LDAP

- Le protocole LDAP fonctionne en mode client-serveur. C'est-à-dire, les échanges entre un client et son serveur sont les requêtes/réponses.
- Un client est un agent de l'annuaire **DUA** (*Directory User Agent*)
- Un serveur est un agent du système de l'annuaire **DSA** (*Directory System Agents*)
- Les requêtes et réponses sont transmises par LDAP
- Une *session* LDAP consiste en une connexion de transport, une sécurité à la couche Transport (*Transport Layer Security*), une authentification SASL (*Simply Authentication and Security Layer*) et des messages LDAP
- TLS chiffre des messages LDAP venus de la couche *Application*
- SASL permet d'assurer l'authentification entre le client et le serveur
- TLS et SASL sont un couple pour beaucoup d'applications d'Internet telles que LDAP, SMTP, POP et IMAP.

Architecture de LDAP

- L'architecture du protocole LDAP (RFC 4511)



- Les échanges sont codés en BER (*Basic Encoding Rules*) de l'ASN.1
- TCP utilise le port n° 389 pour le protocole LDAP

Enveloppe de messages

- Les messages échangés (LDAP PDU) sont au format **LDAPMessage** décrit ci-dessous :

```
LDAPMessage ::= SEQUENCE {                          ← définition d'un type de données
messageID MessageID, protocolOp CHOICE {            ← choix des opérations dans la liste
bindRequest BindRequest,
bindResponse BindResponse,
unbindRequest UnbindRequest,
searchRequest SearchRequest,
searchResEntry SearchResultEntry,
searchResDone SearchResultDone,
searchResRef SearchResultReference,
modifyRequest ModifyRequest,
modifyResponse ModifyResponse,
addRequest AddRequest,
addResponse AddResponse,
delRequest DelRequest,
delResponse DelResponse,
modDNRequest ModifyDNRequest,
…},
controls [0] Controls OPTIONAL }                    ← contrôle sémantique
```

Opérations (1)

- ***Bind*** (request/response) permet d'établir une session *authentifiée*

Message est venu de la couche App

BindRequest ::= [APPLICATION 0] SEQUENCE {

version INTEGER (1 .. 127),

name LDAPDN,

authentication AuthenticationChoice }

AuthenticationChoice ::= CHOICE {

simple [0] OCTET STRING, -- 1 and 2 reserved

sasl [3] SaslCredentials, ... }

SaslCredentials ::= SEQUENCE { mechanism LDAPString, credentials OCTET STRING OPTIONAL }

- Si « simple » est choisi, le mot de passe pourrait être utilisé. Ensuite, TLS est employé pour la session.
- Si «sasl» est choisi, plusieurs mécanismes de sécurité pourront être employés, Kerberos par exemple.

Opérations (2)

- *Unbind* (= quit, request/response) permet la fermeture d'une session. Toutes les opérations qui ne sont pas encore exécutées sont abandonnées
- *Unsolicited Notification* (du serveur au client) indique que le serveur ferme la session LDAP au cours
- *Search* (request/response) permet l'accès à l'annuaire
- *Search result* (du serveur au client) indique le résultat ou la référence (location) du prochain serveur *si* le serveur actuel n'ai pas de données demandées. Si c'est le cas, *Continuation Reference* permet de rediriger le chemin à l'accès
- *Modify* (request/response) permet au client de modifier une entrée de l'annuaire
- *Add*, *Delete*, *Replace*, *Compare*, *Abandon*
- *Extended operation* permet au serveur d'implémenter des autres opération, un mécanisme de sécurité, par exemple.
- ….

Exemple – Search Result

SearchResultEntry for DC=example,DC=com

SearchResultEntry for CN=John Doe,DC=example,DC=com

SearchResultReference

{ ldap://hostb/OU=person,DC=example,DC=com??sub
ldap://hostc/OU=Person,DC=example,DC=com??sub }

SearchResultReference { ldap://hostd/OU=Roles,DC=example,DC=com??sub }

SearchResultDone (success)

trouver l'annuaire

- *hostb* est le serveur primaire et *hostc* est le serveur secondaire. Les deux possèdent le sous-arbre «example.com»

Exemples (suite)

- Ajouter un numéro de téléphone à l'entrée « John Doe» :

dn: **cn**=John Doe, **dc**=example, **dc**=com
changetype: **modify**
add: telephonenumber
telephonenumber: 33 (0)140272524

- Supprimer l'entrée « John Doe» :

dn: **cn**=John Doe, **dc**=example, **dc**=com
changetype: **delete**

Plan

1. Introduction
2. Structure de l'annuaire
3. Protocole et Opérations principales
4. Localisation URL
5. Modèle de l'information
6. LDAP en pratique
7. Conclusion

Introduction

- **URL** *(Uniform Resource Locator*) créé en 1990, permet de localiser des documents à l'aide du réseau Internet.
- Dans le cadre du LDAP, URL permet de réaliser la recherche de l'annuaire (l'opération *search*, par exemple). URL permet également de poursuivre la recherche (l'opération *Continuation Reference*, par exemple)
- Rappelons la syntaxe d'URL:

scheme://domain:port/path?query_string#fragment_id

nom de domaine — n° de port — la requête — les données à passer au serveur

- Le standard RFC 4516 définit le format d'URL afin que l'URL s'adapte au LDAP

Format URL

ldapurl = scheme COLON SLASH SLASH [host [COLON port]]
[SLASH dn [QUESTION [attributes]
[QUESTION [scope] [QUESTION [filter] [QUESTION extensions]]]]]
scheme = "ldap"

- *COLON* signifie « : »
- *SLASH* signifie « / »
- *host* est le nom ou l'adresse IP du serveur (@IPv4 ou @ IPv6)
- *port* est le n° 389
- *dn (*distinguishedName*)*
- *QUESTION* signifie « ? »
- *attributes* sont les attributs d'une (ou plusieurs) entrée(s)
- *scope* consiste en trois valeurs (base/one/sub). Il définit la zone de la recherche
- *filter* permet de définir des critères de recherche. Par défaut, «objectclass=*»
- *extensions* permet des futurs extensions (une extension «opération», par exemple)

Exemples

- URL indique explicitement (ou ne pas indiquer) le serveur de l'annuaire
- ldap://ldap1.example.net/o=University%20of%20Michigan,c=US -> le serveur = ldap1.exemple.net ; on cherche tous les entrées de l'Université de Michgigan
- ldap:///o=University%20of %20Michigan,c=US -> n'importe quel serveur

- URL indique la recherche de l'ensemble des attributs du sous-arbre (chemin) si le nom commun est « Babs Jensen ». Ici, 6666 est le n° port du serveur.
- ldap://ldap1.example.net:6666/o=University%20of%20Michigan, c=US??sub?(cn=Babs%20Jensen)

- URL peut spécifier une chaîne de caractères du LDAP (4 octets = 00/00/00/04)
- ldap://ldap3.example.com/o=Babsco,c=US ???(four-octet=%5c00%5c00%5c00%5c04)

Plan

1. Introduction
2. Structure de l'annuaire
3. Protocole et Opérations principales
4. Localisation URL
5. Modèle de l'information
6. LDAP en pratique
7. Conclusion

Modèle de l'information (1)

- **Syntaxes** de l'information LDAP définies par RFC 4517 consistent à structurer les entrées (classes d'objet) de l'annuaire et à représenter les attributs transférés par LDAP. Elles respectent aux règles de l'ASN.1
- Le principe :

  PrintableCharacter = ALPHA / DIGIT / SQUOTE / LPAREN / RPAREN / PLUS / COMMA / HYPHEN / DOT / EQUALS / SLASH / COLON / QUESTION / SPACE

  PrintableString = 1*PrintableCharacter

  IA5 String = *(%x00-7F) -> une chaîne de « 0 » à quelques caractères

  SLASH = %x2F ; forward slash ("/")

  COLON = %x3A ; colon (":")

  QUESTION = %x3F ; question mark ("?")

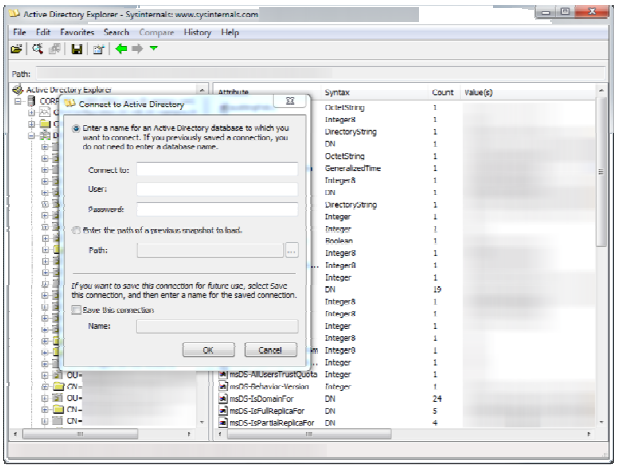
Modèle de l'information (2)

- **Règles de correspondance** (*matching rule*) concernant les opérations de la recherche et de la comparaison entre deux entrées.
- Le principe :
  numericStringMatch,
  numericStringSubstringsMatch,
  caseExactMatch,
  caseExactOrderingMatch,
  caseExactSubstringsMatch,
  caseExactIA5Match,
  caseIgnoreIA5Match,
  caseIgnoreIA5SubstringsMatch,
  caseIgnoreListMatch,
  caseIgnoreListSubstringsMatch,
  caseIgnoreMatch,
  caseIgnoreOrderingMatch….

Plan

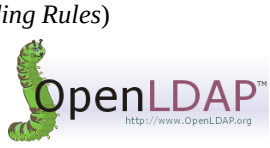
Logiciels - client

- Microsoft : Active Directory Explorer
- OS MAC : Address Book, Directory Utility
- Linux/UNIX : Evolution
- Plateforme JXplorer (JAVA), LDAP Account Manager (PHP)
- …



Logiciels - serveur

- Microsoft : Active Directory
- OS MAC : Appel Open Directory
- Linux/UNIX : 389 Directory Server (développé par Red Hat)
- Plateforme : OpenLDAP (Windows, OS MAC, UNIX, Android), OpenDS (Java)
- …
- OpenLDAP consiste principalement en quatre parties suivantes :
- Le démont *slapd* (*stand-alone LDAP*) gère les requêtes
- Les libraires du protocole LDAP (*libpamldap* et *libnssldap, par exemple*) et de la représentation de données BER (*Basic Encoding Rules*)
- La réplication de données (*slurpd*)
- Pages *man* : *ldapsearch*, *ldapadd*, *ldapdelete*



Configuration du serveur LDAP (1)

- Installation du logiciel *OpenLDAP* dans le répertoire /etc/ldap

ftp://ftp.openldap.org

- Configuration du fichier *slapd.conf* dans le répertoire /etc/ldap/slapd.conf
- Définition des schémas : include inetorgperson.schema
- Définition de la base de données : backend bdb
- Définition de la racine de l'arbre : suffix "dc=cnam, dc=fr"
- Définition de l'administrateur et son mot de passe : rootdn "cn=Manager, dc=cnam,dc=fr" ; rootpw XXXXXXXX
- Définition du répertoire où la base est stockée: directory "/var/lib/ldap"
- Définition du mode : chmod 600 /etc/ldap/slapd.conf

Configuration du serveur LDAP (2)

- Définition des listes de contrôle d'accès :

Par défaut : access to attribute=userPassword
by self write
by dn="cn=Manager,dc=cnam,dc=fr" write
by anonymous auth (lors d'une opération *bind*, l'utilisateur peut lire)
by * none

L'administrateur a un droit d'accès complet en écriture ; les utilisateurs ne peuvent que lire : access to *
by self write
by dn="cn=Manager,dc=exemple,dc=fr" write
by * read

Configuration du client LDAP (1)

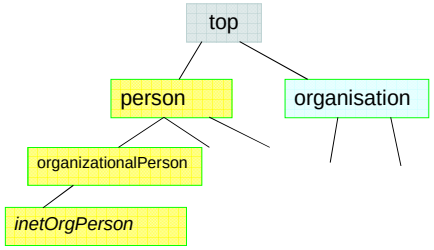
- Installation des paquets : libpam-ldap et libnss-ldap sous *Ubuntu*
- Configuration des paquets
- Indiquer l'adresse IP du serveur, le nom de l'arbre (dc=cnam,dc=fr), la version du LDAP (version 3), le compte *root* (cn=Manager, dc=cnam, dc=fr) ...
- Configuration du fichier *nsswitch.con (*NSS – *Name Service Switch* permet de configurer les noms d'user, les groupes et des autres informations dans le répertoire /etc/nsswitch.conf)
- passwd: files ldap
  group: files ldap
  shadow: files ldap

Configuration du client LDAP (2)

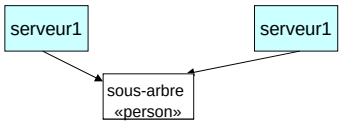
- Configuration de PAM (*Pluggable Authentication Modules*) pour tous les fichiers *common-**. Par exemple,
- Le fichier common-account dans le répertoire /etc/pam.d
  account sufficient pam_ldap.so
  account required pam_unix.so
- Le fichier common-auth dans le répertoire /etc/pam.d
  auth sufficient pam_ldap.so (l'authentification LDAP est suffisante)
  auth required pam_unix.so nullok_secure use_first_pass
- Le fichier common-password dans le répertoire /etc/pam.d
  password sufficient pam_ldap.so
  password required pam_unix.so nullok obscure min=4 max=8 md5

Conception d'un système

- La conception d'un système LDAP consiste à déterminer
- La structure de l'annuaire afin d'augmenter la performance du système



- Le nombre des serveurs afin d'assurer la tolérance aux pannées



Conclusion

- LDAP est un système d'annuaire *distribué local*. Mais l'URL permet une recherche de localisation « mondiale » à l'aide de DNS.
- Une structure de l'annuaire en arbre *standardisée* et efficace.
- L'ensemble des standards IETF a été redéfini en 2006. Donc, les logiciels LDAP sont souvent recents.
- Les problèmes de performances et de sûreté sont résolus par les serveurs primaires et secondaires mais aussi par l'utilisation de cache
- Les mesures de sécurité reposent sur le protocole sécurisé (TLS) et l'authentification (SASL).