

# OpenLDAP

Olivier Hoarau ([olivier.hoarau@funix.org](mailto:olivier.hoarau@funix.org))

V1.8 du 24 décembre 2016

1	Historique.....	2
2	Préambule.....	2
3	Présentation.....	3
4	Format de la base et définitions.....	4
4.1	Le Directory Information Tree.....	4
4.2	Les attributs.....	4
4.3	Les classes d'objet.....	4
4.4	Les schémas.....	5
5	Installation d'OpenLDAP.....	5
5.1	Présentation.....	5
5.2	Installation.....	5
6	Mettre en place son schéma d'annuaire.....	6
6.1	Mise en place des classes d'objet.....	6
6.2	Choix du suffixe.....	7
7	Configuration du serveur LDAP.....	7
8	Lancement du serveur.....	9
9	Utilisation sommaire.....	10
9.1	Ajouter un enregistrement.....	10
9.2	Rechercher un enregistrement.....	12
9.3	Modifier un enregistrement.....	13
9.3.1	Rajouter un attribut à un enregistrement.....	13
9.3.2	Modifier un attribut.....	13
9.3.3	Supprimer un attribut.....	14
9.4	Supprimer un enregistrement.....	14
10	Authentification des utilisateurs avec LDAP.....	14
10.1	Présentation.....	14
10.2	Installation.....	15
10.3	Configuration.....	16
10.3.1	Configuration d'un serveur.....	16
10.3.2	Configuration serveur et client.....	21
10.4	Test de fonctionnement.....	23
10.5	Gestion des utilisateurs et groupes.....	23
10.5.1	Créer un nouvel utilisateur.....	23
10.5.2	Rajouter un groupe.....	25
10.5.3	Supprimer un utilisateur.....	25
10.6	Changer son mot de passe.....	25
10.7	Suggestion de mise en place.....	26

# 1 Historique

V1.6 24.12.16 passage à OpenLDAP 2.4.44 et Admin4 2.2.4, modification pour supprimer des logs et warnings gênants

V1.7 15.10.15 passage à la version OpenLDAP 2.4.42, suppression de l'installation de pam\_ldap et nss\_ldap, présentation de nss\_pam\_ldapd 0.9.6 et de Admin4 modifications suite passage à Mageia5 et systemd

V1.6 24.12.13

V1.5 04.09.10 passage à la version 2.4.23, suppression de la version rpm

V1.4 05.11.04 passage à la version 2.2.18 et 2.1.25 (RPM mdk)

V1.3 04.05.03 Passage à OpenLDAP 2.1.17 et Mandrake 9.1 (OpenLDAP 2.0.27)

V1.2 24.12.02 Passage à OpenLDAP 2.1.8 et Mandrake 9.0 (OpenLDAP 2.0.25)

V1.1 07.07.02 Passage à OpenLDAP 2.1.2, rajout d'un paragraphe sur l'authentification des utilisateurs du système basé sur LDAP

V1.0 09.06.02 Création du document

## 2 Préambule

Ce document présente **OpenLDAP** avec une application pratique (authentification des utilisateurs).

La dernière version de ce document est téléchargeable à l'URL <http://www.funix.org>.

Ce document est sous licence Creative Commons Attribution-ShareAlike 3.0 Unported, le détail de la licence se trouve sur le site <http://creativecommons.org/licenses/by-sa/3.0/legalcode>. Pour résumer, vous êtes libres

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création

suivant les conditions suivantes:

- **Paternité** — Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'oeuvre).
- **Partage des Conditions Initiales à l'Identique** — Si vous transformez ou modifiez cette oeuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

Par ailleurs ce document ne peut pas être utilisé dans un but commercial sans le consentement de son auteur. Ce document vous est fourni "dans l'état" sans aucune garantie de toute sorte,

l'auteur ne saurait être tenu responsable des quelconques misères qui pourraient vous arriver lors des manipulations décrites dans ce document.

## 3 Présentation

**LDAP** est un protocole basé sur TCP/IP qui permet de partager des bases de données d'information sur un réseau interne (intranet) ou externe (internet). Ces bases de données sont appelées annuaire électronique (Directory en anglais), elles peuvent contenir tout type d'informations, des informations sur les personnes, à des données systèmes. Qui dit base de données, dit recherche, il est donc possible de faire des recherches dans la base en employant plusieurs critères, mais aussi bien sûr de la modifier, mais contrairement à un SGBD, un annuaire est très rapide en lecture, mais l'est beaucoup moins en écriture, en effet comme un annuaire est plutôt lu que modifier il a été optimisé pour la lecture et ne possède pas les mécanismes de transaction complexe que les SGBD possèdent pour traiter de gros volumes de données.

Le **LDAP** ou Lightweight Directory Access Protocol est la version TCP/IP du protocole **DAP**, ce dernier étant le protocole pour accéder au protocole OSI du service d'annuaire X500. Dans un premier temps **LDAP** s'est contenté d'être l'interface à des annuaires X500, mais maintenant **LDAP** peut gérer complètement les bases (standalone **LDAP**).

Si on rentre dans les détails, le protocole **LDAP** est du type client serveur, le serveur contient la base de données, et le client consulte la base de données, le protocole fournit les bases pour cette communication entre la client et le serveur (normalisée par l'IETF par la RFC2251), et les commandes nécessaires au client pour rechercher, créer, modifier ou effacer des données. **LDAP** est bien entendu sécurisé pour le transfert et l'accès aux des données, avec des outils de cryptage comme SSL et d'authentification.

Par ailleurs **LDAP** fournit des outils pour que les serveurs **LDAP** puissent communiquer entre eux, on a ainsi la possibilité de créer des serveurs miroirs qui pourront se synchroniser, ou de relier simplement les serveurs entre eux, les serveurs redirigeant automatiquement les requêtes qui ne les concernent pas.

Les exemples d'applications de **LDAP** sont nombreux:

- bases de données d'employés,
- bases de données de produits,
- bases de données pour certaines applications, exemple :
  - toutes les infos contenant les utilisateurs de votre réseau (mot de passe, shell, homedirectory, ...) peuvent être dans la base, on a ainsi beaucoup plus de possibilités qu'un simple fichier **/etc/passwd**, l'authentification peut donc utiliser **LDAP** plutôt que **passwd** ou **shadow** ou encore **NIS**. Vos utilisateurs pourront ainsi changer leur mot de passe et certains de leurs attributs à partir d'une interface web.
  - les préférences d'applications ou d'environnement (**netscape**, environnement graphique KDE, ...) sont sauvegardés dans la base, ainsi l'utilisateur peut passer d'une machine à une autre et retrouver ses préférences.

Cette page est une introduction à **LDAP** elle ne couvre pas certains aspects comme les liens avec d'autres bases (duplication, miroir, ...), la sécurité (access control, SSL, ...). Elle n'a

seulement pour but de mettre en place un serveur **LDAP** simplement configuré pour que vous puissiez faire vos "premières armes" dans le domaine.

Pour utiliser une base **LDAP** à partir de script **PHP**, voir mon document Apache téléchargeable sur [www.funix.org](http://www.funix.org).

## 4 Format de la base et définitions

### 4.1 Le Directory Information Tree

Les **LDAP** standalone utilisent le format de base de données **LDBM**, ce dernier utilise le modèle hiérarchique comme le système de fichiers UNIX, c'est à dire qu'il s'apparente à un arbre, qu'on appelle **DIT** (Directory Information Tree). Au sommet de cet arbre se trouve la racine ou suffixe et à chaque nœud de l'arborescence on a un **DSE** (Directory Service Entry) qui correspond à une entrée de l'annuaire. L'entrée située à la racine est appelé **rootDSE** (root Directory Specific Entry), qui décrit la structure de l'arborescence (le **DIT**) ainsi que son contenu.

Chaque entrée est connue de manière unique dans l'arborescence grâce à son **dn** (Distinguished Name). Le **dn** indique le chemin à parcourir pour en partant du sommet arriver à l'entrée correspondante. Par exemple pour identifier une personne, on part du pays (fr), puis le nom de domaine (kervao pour la suite des opérations), le groupe de travail et enfin le nom de la personne, l'ensemble de ces paramètres est le **dn** qui identifie de manière unique une personne.

### 4.2 Les attributs

Chaque entrée peut être considérée comme un objet (au sens C++) possédant donc certains attributs, par exemple si une personne est une entrée, les attributs peuvent être, le nom, le prénom, l'âge, .... On peut aussi définir des attributs obligatoires et d'autres optionnels, en d'autres termes, les attributs obligatoires devront être renseignés mais pas forcément les optionnels. Il existe par ailleurs pour chaque **DSE** des attributs d'administration qui ne servent qu'au serveur.

### 4.3 Les classes d'objet

On regroupe les objets qui sont du même domaine dans une classe d'objet, celle-ci est caractérisée par des attributs obligatoires ou optionnels et un type. Les types de classe d'objet sont:

- type structurel car elle contient des d'objets concrets de l'annuaire (personnes, groupes de personnes, ...),
- type auxiliaire, c'est des classes d'objets qu'on peut créer, pour rajouter des informations (attributs) supplémentaires à des classes d'objet de type structurel déjà existantes. En C++ on dira que la classe auxiliaire dérive d'une classe structurelle,
- type abstraite, c'est les classes d'objet qui existent par défaut et qui n'ont pas de signification concrète, par exemple la classe top est la classe d'objet générique, toutes les autres classes dérivent de cette classe.

Le principe est donc le même qu'en C++, on retrouve une structure arborescente, avec à la racine la classe **top**, toutes les autres classes d'objet dérivent de cette classe générique, chaque

classe hérite des propriétés d'une classe père et possède des attributs supplémentaires par rapport à ce dernier.

## 4.4 Les schémas

Un schéma décrit toutes les règles qu'utilisent le serveur **LDAP** pour décrire les classes d'objets (attributs, syntaxe, ...).

# 5 Installation d'OpenLDAP

## 5.1 Présentation

Il existe de nombreux serveurs **LDAP**, nous utiliserons **OpenLDAP** qui comme son nom l'indique est sous licence GPL. Vous avez le choix entre la version packagée par votre distribution ou la dernière version stable 2.4.44 à l'URL <http://www.openldap.org>. Ma préférence va vers la recompilation pour avoir une version optimisée et de plus plus récente.

## 5.2 Installation

On vérifie d'abord qu' **Openldap** n'est pas déjà installé sur votre système en tapant :

```
rpm -qa | grep -i ldap
```

On supprime du système les packages contenus dans la liste avec la commande **rpm -e nom-du-package** (sauf **libldap2** qui sert pour de nombreux packages).

Si pour des histoires de dépendance vous n'arrivez pas à tout supprimer ce n'est pas bien grave car par défaut le tarball et les packages ne placent pas les fichiers au même endroit. Lors du lancement du daemon et des exécutables il faut juste faire attention d'appeler le bon exécutable (servez vous de la commande **which nom-exe** ).

L'archive à récupérer est **openldap-2.4.44.tgz** qu'on décompressera en tapant :

```
tar xvzf openldap-2.4.44.tgz
```

Cela va nous créer un répertoire **openldap-2.4.44**. Avant d'aller plus loin il faudra installer (commande **urpmi nom-package** ) les packages suivants

```
lib64db5.3-devel
```

```
gnutls-devel
```

```
lib64sasl2-devel
```

```
lib64nss-devel
```

Puis on tape successivement :

```
./configure
```

```
make
```

On peut tester maintenant que tout marche bien en tapant :

```
cd tests
```

```
make
```

pour installer les binaires de **ldap** on tapera, en tant que **root** :

```
cd ..  
make install
```

Les binaires sont installés par défaut dans `/usr/local/sbin` et `/usr/local/libexec`, les fichiers de config dans `/usr/local/etc/openssl` et les bases dans `/usr/local/var/openssl-data`. Les biblio vont se trouver sous `/usr/local/lib`, si ce n'est pas fait, rajouter ce chemin à la fin du fichier `/etc/ld.so.conf` et tapez

```
ldconfig
```

pour changer l'emplacement de tous ces fichiers taper:

```
configure -help
```

## 6 Mettre en place son schéma d'annuaire

### 6.1 Mise en place des classes d'objet

Le fichier de conf `slapd.conf` fait appel à `/usr/local/etc/openssl/schema/core.schema` qui décrit les classes d'objet. Voilà un exemple avec la classe "person"

```
objectclass ( 2.5.6.6 NAME 'person'  
  DESC 'RFC2256: a person'  
  SUP top STRUCTURAL  
  MUST ( sn $ cn )  
  MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

**MUST** correspond aux attributs obligatoires et **MAY** à ceux facultatifs

**objectClass** est le nom de la classe qui descend elle-même de la classe **top**

**sn** correspond à surname (nom)

**cn** correspond à common name (prénom nom)

Je vous laisse deviner la signification des autres attributs.

On voit qu'il est nécessaire de fournir les attributs **sn** (surname) et **cn** (common name), sont facultatifs le mot de passe ( **userPassword**), le numéro de téléphone ( **telephoneNumber** ), les liens (**seeAlso**) et la description.

Les attributs sont définis dans le même fichier, la syntaxe est la suivante pour **telephoneNumber** par exemple :

```
attributetype ( 2.5.4.20 NAME 'telephoneNumber'  
  DESC 'RFC2256: Telephone Number'  
  EQUALITY telephoneNumberMatch  
  SUBSTR telephoneNumberSubstringsMatch  
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.50{32} )
```

Je vous présenterai la syntaxe plus tard, on peut dans un premier temps se limiter aux attributs disponibles. Pour créer une classe d'objet **breizhPerson** dérivant de **person**, disposant de l'attribut obligatoire **title** en plus et des arguments facultatifs **ou** (groupe de travail) et **l** (localisation). On tapera dans le fichier `core.schema` juste après la définition de la classe **person**

```
objectclass ( 2.5.6.6.2 NAME 'breizhPerson' SUP person STRUCTURAL
  MUST ( title )
  MAY ( ou $ 1 ) )
```

Vous noterez le nombre 2.5.6.6.2, ce nombre doit être unique dans le fichier, il dérive directement du numéro de la classe objet **person** qui a pour numéro 2.5.6.6. Il est évident que comme **breizhPerson** dérive de **person**, les attributs **sn** et **cn** sont aussi obligatoires.

A noter qu'avec une installation avec package les classes "locales" peuvent être créées dans le fichier `/etc/openldap/schema/local.schema`

## 6.2 Choix du suffixe

Le **rootDSE** ou suffixe correspond à l'entrée tout en haut de l'arbre (**DIT**) de l'annuaire, on utilise généralement le nom de domaine, avec la syntaxe suivante **dc=kervao, dc=fr** pour le domaine **kervao.fr** (**dc** correspond à Domain Component).

# 7 Configuration du serveur LDAP

On va créer un annuaire **LDAP** pour votre domaine privé **kervao.fr**. On doit modifier les fichiers **slapd.conf** et **ldap.conf** se trouvant sous `/usr/local/etc/openldap`. Voilà pour le fichier de configuration **slapd.conf**

```
# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.8.8.4 2000/08/26 17:06:18
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#

include /usr/local/etc/openldap/schema/core.schema
include /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema
include /usr/local/etc/openldap/schema/nis.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral ldap://root.openldap.org
# le chemin est différent avec une install avec package

pidfile /usr/local/var/slapd.pid
argsfile /usr/local/var/slapd.args

# Load dynamic backend modules:
# modulepath /usr/local/libexec/openldap
# moduleload back_ldap.la
# moduleload back_ldbm.la
# moduleload back_passwd.la
# moduleload back_shell.la

# slapd est bavard, pour le limiter aux messages critiques on mettre none
loglevel none
```

```

# Sample security restrictions
#   Require integrity protection (prevent hijacking)
#   Require 112-bit (3DES or better) encryption for updates
#   Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64
#####
# database definitions
#####
database    bdb
suffix      "dc=kervao,dc=fr"
rootdn      "cn=Manager,dc=kervao,dc=fr"

```

```

#mot de passe en clair, on verra plus loin comment le crypter
rootpw      secret

```

```

# là où va se trouver la base ldap /var/lib/ldap dans le cas d'une install par package
directory   /usr/local/var/openldap-data

```

```

# Indices to maintain
index       objectClass eq

```

Attention vous devez vous assurer que le répertoire où se trouvera la base **LDAP** a été créé, par défaut c'est **/usr/local/var/openldap-data** mais vous pouvez très bien mettre **/var/lib/ldap** si ça vous chante.

Le fichier **ldap.conf** peut être vide dans un premier temps voire inexistant.

Le mot de passe de l'administrateur est **secret** en clair, si ça ne vous convient pas et que vous voulez le crypter, il faudra taper

**slappasswd**

on saisit son mot de passe

**New password:**

**Re-enter new password:**

et voilà le résultat

```
{SSHA}vNCYRQthN6u3OqcTAJ4lt/9vIhjsFmI
```

A la place de

```
rootpw      secret
```

Dans **slapd.conf**, vous mettez donc:

```
rootpw      {SSHA}vNCYRQthN6u3OqcTAJ4lt/9vIhjsFmI
```

pour plus de sécurité on crée maintenant un utilisateur et un groupe système **ldap**

```
groupadd ldap
```

```
useradd -g ldap -c "slapd demon" -s /sbin/nologin -d /tmp ldap
```

on le rend propriétaire du fichier de configuration et de la base **ldap** en tapant

```
chown ldap:ldap /usr/local/etc/openldap/slapd.conf
```

```
chown -fR ldap:ldap /usr/local/var/openldap-data
```

```
chown -fR ldap:ldap /usr/local/var/run
```

maintenant sous **/usr/local/var/openldap-data** on tapera



```
cp DB_CONFIG.example DB_CONFIG
chown ldap:ldap DB_CONFIG
```

## 8 Lancement du serveur

La configuration du lancement sous **systemd** est la suivante. On crée un fichier **slapd.service** qu'on place sous **/usr/lib/systemd/system/** voilà son contenu

**[Unit]**

**Description=OpenLDAP server daemon**

**[Service]**

**Type=forking**

**ExecStart=/usr/local/libexec/slapd -u ldap -g ldap -f /usr/local/etc/openldap/slapd.conf**

**[Install]**

**WantedBy=multi-user.target**

maintenant pour que le service soit lancé à chaque boot de la machine il faudra taper

```
systemctl enable slapd.service
```

voilà le résultat

**Created symlink from /etc/systemd/system/multi-user.target.wants/slapd.service to /usr/lib/systemd/system/slapd.service.**

pour le lancer il suffit maintenant de taper

```
systemctl start slapd.service
```

et pour connaître son état

```
systemctl status slapd.service
```

voilà le résultat

- **slapd.service - OpenLDAP server daemon**

**Loaded: loaded (/usr/lib/systemd/system/slapd.service; enabled)**

**Active: active (running) since sam. 2016-12-24 11:14:52 CET; 27s ago**

**Process: 15515 ExecStart=/usr/local/libexec/slapd -u ldap -g ldap -f /usr/local/etc/openldap/slapd.conf (code=exited, status=0/SUCCESS)**

**Main PID: 15517 (slapd)**

**CGroup: /system.slice/slapd.service**

└─15517 /usr/local/libexec/slapd -u ldap -g ldap -f /usr/local/etc/openldap/slapd.conf

déc. 24 11:14:41 mana.kervao.fr slapd[15515]: @(#) \$OpenLDAP: slapd 2.4.44 (Dec 24 2016 11:12:03) \$

olivier@mana.kervao.fr:/usr/local/linux/systeme/openldap-2.4.44/servers/slapd

# 9 Utilisation sommaire

## 9.1 Ajouter un enregistrement

Vous avez différent moyen d'ajouter des données à l'annuaire, pour une meilleure compréhension on va d'abord aborder la méthode manuelle. Pour ajouter des données au serveur **LDAP** vous devez vous fournir un fichier au format **LDIF** (pour LDAP Directory Interchange Format), le format est un format texte facilement lisible au contraire du format interne de l'annuaire. Voici un exemple de fichier **LDIF**, à noter que:

- chaque enregistrement dans le fichier est séparé du précédent et du suivant par une ligne vierge,
- les espaces sont pris en compte. **ATTENTION**, il est très important qu'il n'y ait aucun espace en fin de ligne. Dans ce cas vous risqueriez d'obtenir une erreur du style

**ldap\_add: Invalid syntax (21)**

**additional info: objectClass: value #0 invalid per syntax**

La syntaxe du format **LDIF** est la suivante:

**dn:** description du distinguished name

**objectclass:** classe d'objet d'origine

...

**objectclass:** classe d'objet dérivée

type attribut: valeur

...

On va par exemple utiliser la classe **breizPerson** définie plus haut pour décrire une nouvelle personne **Veronique Hoarau** qu'on va rajouter dans l'annuaire. Elle appartient au service (**organizationalUnit**) **staff**, ce même service appartenant à l'organisation **kervao.fr**

Soit le fichier **entree.ldif**

**dn: dc=kervao, dc=fr**

**objectClass: dcObject**

**objectClass: organization**

**dc: kervao**

**o: kervao.fr**

**dn: ou=staff, dc=kervao, dc=fr**

**objectclass: organizationalUnit**

**ou: staff**

**dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr**

**objectclass: person**

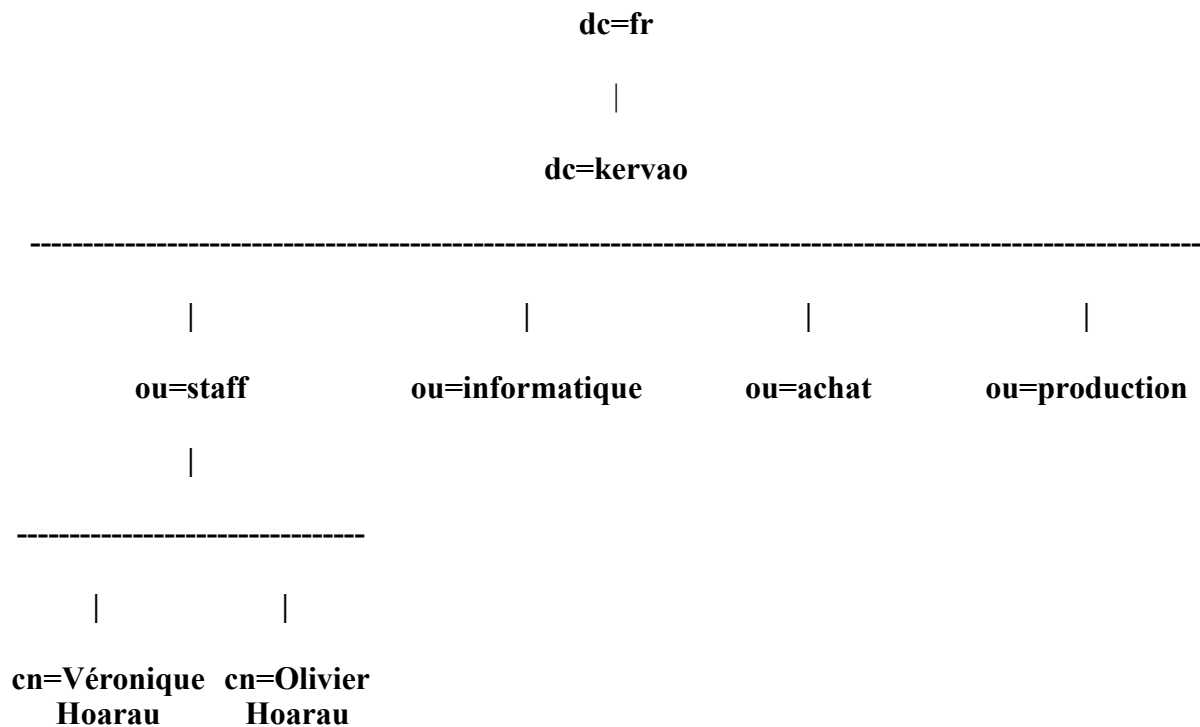
**objectclass: breizhPerson**

**cn: Veronique Hoarau**

**sn: Hoarau**

**title: madame**

Quelques commentaires, le premier groupe correspond à la définition de votre organisation, le deuxième à celui du groupe de travail (**organizationalUnit**) et le dernier à la personne. Celle-ci est définie par son **dn** (Distinguished Name), on part du sommet **bn** (suffixe du nom de domaine), puis le nom de domaine, le groupe de travail et enfin la personne. L'arbre (**DIT**) pourrait ressembler à ça:



Au niveau de la définition de la personne:

**objetclass: person** définit la classe père de la classe **breizPerson**,  
**objetclass: breizPerson** classe décrivant la personne,  
**cn** et **sn** sont des attributs à renseigner obligatoirement,  
**title** est un attribut obligatoire

On rajoutera l'enregistrement en utilisant la syntaxe suivante (en tant que simple utilisateur):

```
ldapadd -x -D "description du dn de l'administrateur" -W -f nom-du-fichier-ldif
```

Exemple concret:

```
ldapadd -x -D "cn=Manager, dc=kervao, dc=fr" -W -f entree.ldif  

Enter LDAP Password: secret  

adding new entry "dc=kervao, dc=fr"
```

```
adding new entry "ou=staff, dc=kervao, dc=fr"
```

```
adding new entry "cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr"
```

Pour rajouter par la suite un autre enregistrement dans le groupe **staff**, il sera plus nécessaire de rajouter la définition du groupe et de l'organisation. Soit le fichier **entree.ldif**

```
dn: cn=Olivier Hoarau, ou=staff, dc=kervao, dc=fr
objectclass: person
objectclass: breizhPerson
cn: Olivier Hoarau
sn: Hoarau
title: monsieur
```

On tape ensuite la commande:

```
ldapadd -x -D "cn=Manager, dc=kervao, dc=fr" -W -f entree.ldif
Enter LDAP Password:
adding new entry "cn=Olivier Hoarau, ou=staff, dc=kervao, dc=fr"
```

## 9.2 Rechercher un enregistrement

On utilisera la fonction **ldapsearch**. Pour visualiser tout l'annuaire on peut taper :

```
ldapsearch -x -b 'dc=kervao, dc=fr' '(objectclass=*)'
```

Voilà le résultat

```
# extended LDIF
#
# LDAPv3
# filter: (objectclass=*)
# requesting: ALL
#
# kervao, fr
dn: dc=kervao, dc=fr
objectClass: dcObject
objectClass: organization
dc: kervao.fr
o: kervao.fr

# staff, kervao, fr
dn: ou=staff, dc=kervao, dc=fr
objectClass: organizationalUnit
ou: staff

# Veronique Hoarau, staff, kervao, fr
dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
objectClass: person
objectClass: breizhPerson
cn: Veronique Hoarau
sn: Hoarau
title: madame
```

```
# Olivier Hoarau, staff, kervao, fr
dn: cn=Olivier Hoarau, ou=staff, dc=kervao, dc=fr
objectClass: person
objectClass: breizhPerson
cn: Olivier Hoarau
sn: Hoarau
title: monsieur
```

```
# search result
search: 2
result: 0 Success
```

```
# numResponses: 5
# numEntries: 4
```

## 9.3 Modifier un enregistrement

### 9.3.1 Rajouter un attribut à un enregistrement

On va rajouter l'attribut facultatif location (**I**) à l'enregistrement **Veronique Hoarau**. On va créer un fichier **modif.ldif** contenant:

```
dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
add: 1
title: bureau36
```

On tape ensuite

```
ldapmodify -x -D "cn=Manager, dc=kervao, dc=fr" -W -f modif.ldif
Enter LDAP Password:secret
modifying entry "cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr"
```

### 9.3.2 Modifier un attribut

On va modifier l'attribut titre (**title**) à l'enregistrement **Veronique Hoarau**. On va créer un fichier **modif.ldif** contenant:

```
dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
changetype: modify
replace: title
title: mademoiselle
```

On tape ensuite

```
ldapmodify -x -D "cn=Manager, dc=kervao, dc=fr" -W -f modif.ldif
Enter LDAP Password:secret
modifying entry "cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr"
```

### 9.3.3 Supprimer un attribut

On va supprimer l'attribut **location (l)** à l'enregistrement **Veronique Hoarau**. On va créer un fichier **modif.ldif** contenant:

```
dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
delete: l
```

On tape ensuite

```
ldapmodify -x -D "cn=Manager, dc=kervao, dc=fr" -W -f modif.ldif
Enter LDAP Password:secret
modifying entry cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
```

## 9.4 Supprimer un enregistrement

Pour supprimer l'enregistrement **Veronique hoarau**, on va créer un fichier **modif.ldif** contenant

```
dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
changetype: delete
```

On tape ensuite:

```
ldapmodify -x -D "cn=Manager, dc=kervao, dc=fr" -W -f modif.ldif
Enter LDAP Password:secret
deleting entry cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
```

**ATTENTION** Vous ne pouvez pas supprimer un attribut obligatoire comme **title** pour la classe **breizhPerson**.

# 10 Authentification des utilisateurs avec LDAP

## 10.1 Présentation

L'authentification des utilisateurs sur le système se fait par défaut au moyen des fichiers **/etc/passwd** (définition des utilisateurs), **/etc/group** (identification des groupes d'utilisateurs) et éventuellement **/etc/shadow** si vous utilisez les "shadow password". C'est satisfaisant quand l'on dispose d'une machine isolée, par contre avec un parc d'une centaine de machines, il est peut concevable d'avoir à modifier ces fichiers sur tous les postes pour rajouter un utilisateur. L'idée est de centraliser l'authentification, NIS fait cela très bien ainsi que **LDAP**, c'est ce que l'on va voir dans ce paragraphe.

Dans la suite des opérations, on appellera serveur, la machine qui centralise la définition de tous les utilisateurs et groupes, le client fait appel au serveur pour l'authentification des utilisateurs.

## 10.2 Installation

On récupèrera sur le site [www.padl.com](http://www.padl.com), le tarball **MigrationTools.tgz** on le décompresse en tapant

```
tar xvfz MigrationTools.tgz
```

Cela donne le répertoire **MigrationTools-47**

Maintenant à l'adresse <http://arthurdejong.org/nss-pam-ldapd/> on récupèrera **nss-pam-ldap** qui comme c'est marqué sur le site fournit un module Name Service Switch (NSS) qui accède aux informations de compte et tous autres informations qu'on trouve dans les fichiers du système (hosts, alias, netgroup, etc...). Il fournit également un module **PAM** (Pluggable Authentication Module) pour un serveur **LDAP**. On décompresse l'archive en tapant :

```
nss-pam-ldapd-0.9.7.tar.gz
```

cela donne le répertoire **nss-pam-ldapd-0.9.7** dans lequel on tape

```
./configure
```

Avant d'aller plus loin, installer le package **pam-devel** on tape ensuite

```
make
```

Puis en tant que root

```
make install
```

puis on crée les liens suivants

```
ln -s /lib/security/pam_ldap.so /lib64/security
```

```
ln -s /lib/security/pam_ldap.so /usr/lib64/security/
```

```
ln -s /lib/libnss_ldap.so.2 /lib64
```

Le daemon **nsld** (Name Service LDAP Connection Daemon) fournit dans le package **nss-pam-ldapd** permet de gérer les connexions vers serveur **LDAP** via **PAM**. Pour qu'il soit actif au démarrage on va créer un fichier **nsld.service** dans le répertoire **/usr/lib/systemd/system** voilà ce qu'il contient

```
[Unit]
```

```
Description=Naming services LDAP client daemon
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=forking
```

```
PIDFile=/run/nsld/nsld.pid
```

```
ExecStart=/usr/sbin/nsld
```

```
[Install]
```

```
WantedBy=multi-user.target
```

on crée maintenant un utilisateur et un groupe système **nsld**

```
groupadd nsld
```

```
useradd -g nsld -c "system user for nss-pam-ldapd" -s /bin/false -d / nsld
```

pour le rendre actif à chaque démarrage on tapera donc

```
systemctl enable nsld.service
```

voilà le résultat

**Created symlink from /etc/systemd/system/multi-user.target.wants/nslcd.service to /usr/lib/systemd/system/nslcd.service.**

on tachera de le lancer après avoir configuré le fichier `/etc/nslcd.conf`

## 10.3 Configuration

### 10.3.1 Configuration d'un serveur

On modifiera si nécessaire le fichier `/usr/local/etc/openldap/slapd.conf` pour rajouter les règles d'accès d'usage:

**# Basic ACL**

**access to attrs=userPassword**

**by self write**

**by anonymous auth**

**by dn="cn=Manager,dc=kervao,dc=fr" write**

**by \* none**

**access to \***

**by dn="cn=Manager,dc=kervao,dc=fr" write**

**by \* read**

Relancer le serveur LDAP

**systemctl restart slapd.service**

A présent dans le répertoire **MigrationTools-47**, on va modifier le fichier **migrate\_common.ph**, on doit y indiquer son nom de domaine, comme ceci :

**# Default DNS domain**

**\$DEFAULT\_MAIL\_DOMAIN = "kervao.fr";**

**# Default base**

**\$DEFAULT\_BASE = "dc=kervao,dc=fr";**

Eventuellement vous pouvez modifier la ligne suivante spécifiant le serveur de mail bien que ce ne soit pas absolument nécessaire.

**\$DEFAULT\_MAIL\_HOST = "mail.padl.com";**

A présent il faut rentrer les utilisateurs et groupes du système dans la base de données **LDAP**. Commençons d'abord par créer des fichiers temporaires au format **ldif**. On tape maintenant en tant que root (pour pouvoir lire `/etc/shadow`)

**ETC\_SHADOW=/etc/shadow**

**export ETC\_SHADOW**

**./migrate\_passwd.pl /etc/passwd /tmp/passwd.ldif**

**./migrate\_group.pl /etc/group /tmp/group.ldif**

Editez les deux fichiers **ldif** pour ne laisser que les utilisateurs, enlever tous les utilisateurs et groupes système (root, lp, sys, apache, ...). Voici le contenu de mon **group.ldif** avec mon groupe utilisateur **hoarau**



```
dn: cn=hoarau,ou=Group,dc=kervao,dc=fr
objectClass: posixGroup
objectClass: top
cn: hoarau
userPassword: {crypt}*
gidNumber: 5000
memberUid: olivier
memberUid: veronique
```

Voici maintenant le contenu de mon **passwd.ldif** avec mes deux utilisateurs

```
dn: uid=olivier,ou=People,dc=kervao,dc=fr
uid: olivier
cn: olivier
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}$1$76UeLH8Z$K8rdYPRmUoiONZQm6hV4q.
shadowLastChange: 11858
shadowMax: 99999
shadowWarning: 7
shadowInactive: -1
shadowExpire: -1
shadowFlag: 1081428222
loginShell: /bin/bash
uidNumber: 5001
gidNumber: 5000
homeDirectory: /home/olivier
gecos: olivier
```

```
dn: uid=veronique,ou=People,dc=kervao,dc=fr
uid: veronique
cn: veronique
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}$1$OyedUoIU$uwpYR0bWJGzF4AFAHspSm/
shadowLastChange: 11858
shadowMax: 99999
shadowWarning: 7
shadowInactive: -1
shadowExpire: -1
shadowFlag: 1081428222
loginShell: /bin/bash
uidNumber: 5002
gidNumber: 5000
homeDirectory: /home/veronique
gecos: veronique
```

A présent il faudra créer le fichier **temp.ldif** qui va contenir la définition des Organizational Unit (**ou**) **Group** (groupe d'utilisateur) et **People** (utilisateur). Voici son contenu :

```
dn: dc=kervao,dc=fr
objectClass: dcObject
objectClass: organization
dc: kervao
o: kervao.fr
```

```
dn: ou=Group,dc=kervao,dc=fr
ou: Group
objectClass: top
objectClass: organizationalUnit
description: groupe d utilisateurs
```

```
dn: ou=People,dc=kervao,dc=fr
ou: People
objectClass: top
objectClass: organizationalUnit
description: utilisateurs du systeme
```

On peut commencer à rajouter tout cela (en mode verbeux), dans la base, on commence par les **ou Group** et **People**

```
ldapadd -v -x -D "cn=Manager,dc=kervao,dc=fr" -W -f /tmp/temp.ldif
```

voilà le résultat

```
ldap_initialize( <DEFAULT> )
Enter LDAP Password:
add objectClass:
    dcObject
    organization
add dc:
    kervao
add o:
    kervao.fr
adding new entry "dc=kervao,dc=fr"
modify complete

add ou:
    Group
add objectClass:
    top
    organizationalUnit
add description:
    groupe d utilisateurs
adding new entry "ou=Group,dc=kervao,dc=fr"
modify complete

add ou:
    People
add objectClass:
    top
    organizationalUnit
add description:
    utilisateurs du systeme
```

```
adding new entry "ou=People,dc=kervao,dc=fr"  
modify complete
```

On continue avec le rajout des groupes et utilisateurs:

```
ldapadd -x -D "cn=Manager,dc=kervao,dc=fr" -f /tmp/passwd.ldif -W
```

```
Enter LDAP Password:
```

```
adding new entry "uid=olivier,ou=People,dc=kervao,dc=fr"
```

```
adding new entry "uid=veronique,ou=People,dc=kervao,dc=fr"
```

Puis

```
ldapadd -x -D "cn=Manager,dc=kervao,dc=fr" -f /tmp/group.ldif -W
```

```
Enter LDAP Password:
```

```
adding new entry "cn=hoarau,ou=Group,dc=kervao,dc=fr"
```

On visualise tout ça en tapant

```
ldapssearch -x -D "cn=Manager, dc=kervao, dc=fr" -W -b "dc=kervao,dc=fr"
```

```
Enter LDAP Password:
```

```
version: 2
```

```
#
```

```
# filter: (objectclass=*)
```

```
# requesting: ALL
```

```
#
```

```
# kervao, fr
```

```
dn: dc=kervao, dc=fr
```

```
objectClass: dcObject
```

```
objectClass: organization
```

```
dc: kervao.fr
```

```
o: kervao.fr
```

```
# Group, kervao, fr
```

```
dn: ou=Group,dc=kervao,dc=fr
```

```
ou: Group
```

```
objectClass: top
```

```
objectClass: organizationalUnit
```

```
description: groupe d utilisateurs
```

```
# People, kervao, fr
```

```
dn: ou=People,dc=kervao,dc=fr
```

```
ou: People
```

```
objectClass: top
```

```
objectClass: organizationalUnit
```

```
description: utilisateurs du systeme
```

```
# hoarau, Group, kervao, fr
```

```
dn: cn=hoarau,ou=Group,dc=kervao,dc=fr
```

```
objectClass: posixGroup
```

```
objectClass: top
```

```
cn: hoarau
```

```
gidNumber: 5000
```

```
memberUid: olivier
```

```
memberUid: veronique
```

```
# olivier, People, kervao, fr
dn: uid=olivier,ou=People,dc=kervao,dc=fr
uid: olivier
cn: olivier
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword::
e2NyeXB0fSQxJDc2VWvMSDhaJEs4cmRVUFJtVW5pT05aUW02aFY0cS4=
shadowLastChange: 11858
shadowMax: 99999
shadowWarning: 7
shadowInactive: -1
shadowExpire: -1
shadowFlag: 1081428222
loginShell: /bin/bash
uidNumber: 5001
gidNumber: 5000
homeDirectory: /home/olivier
gecos: olivier
```

```
# veronique, People, kervao, fr
dn: uid=veronique,ou=People,dc=kervao,dc=fr
uid: veronique
cn: veronique
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword::
e2NyeXB0fSQxJE95aWRVb0lVJHV3cFlOM7JXSkd6RjRBRkFIc3BTbS8=
shadowLastChange: 11858
shadowMax: 99999
shadowWarning: 7
shadowInactive: -1
shadowExpire: -1
shadowFlag: 1081428222
loginShell: /bin/bash
uidNumber: 5002
gidNumber: 5000
homeDirectory: /home/veronique
gecos: veronique
```

```
# search result
search: 2
result: 0 Success
```

```
# numResponses: 7
# numEntries: 6
```

### 10.3.2 Configuration serveur et client

Pour un client sous mageia il faut installer les packages suivants

```
openldap  
nss-pam-ldapd
```

Pour le dernier package vous avez également l'alternative d'installer la version tarball. Le client repose sur **nsld** (Name Service LDAP Connection Daemon), à noter que sur un serveur si vous voulez également vous connecter sur le daemon LDAP, il faudra le configurer comme un client. On édite le fichier **/etc/nsld.conf** et on apporte les modifications suivantes

```
# The user and group nsld should run as.  
uid nsld  
gid nsld  
  
# le serveur LDAP  
uri ldap://192.168.13.11/  
  
# The distinguished name of the search base.  
base dc=kervao,dc=fr  
  
# The distinguished name to bind to the server with.  
# Optional: default is to bind anonymously.  
binddn cn=Manager,dc=kervao,dc=fr  
  
# The credentials to bind with.  
# Optional: default is no credentials.  
# Note that if you set a bindpw you should check the permissions of this file.  
bindpw secret  
  
# Customize certain database lookups.  
base group ou=Group,dc=kervao,dc=fr  
base passwd ou=People,dc=kervao,dc=fr  
base shadow ou=People,dc=kervao,dc=fr  
map shadow userPassword userPassword
```

Dans le fichier **/etc/nsswitch.conf** on modifiera les lignes suivantes pour lire

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

A noter que le **nis** n'est pas nécessaire si vous n'avez pas mis en place de domaine **NIS**.

Bon maintenant on va éditer le fichier **/etc/pam.d/system-auth**. Pour information, c'est dans ce fichier qui va servir à appeler les bibliothèques qui vont bien pour l'authentification d'un utilisateur à la connexion sur une machine. Sans rentrer dans le détail, ce fichier et ceux qui se trouvent dans le même répertoire font parti du système **PAM** (Pluggable Authentication Modules), qui permet de gérer l'authentification sur le système que ce soit pour le login mais aussi pour d'autres services comme l'accès au serveur de mail, à **ftp**, etc.

Reprenons donc notre fichier **system-auth** il doit ressembler à ça (rajoutez uniquement les lignes concernant **pam\_ldap** et laissez les autres) :

## #%PAM-1.0

```
auth    required    pam_env.so
auth    sufficient  pam_tcb.so shadow nullok prefix=$2a$ count=8
auth    required    pam_deny.so
auth    sufficient  pam_ldap.so

account sufficient  pam_tcb.so shadow
account required   pam_deny.so
account sufficient  pam_ldap.so

password required    pam_cracklib.so try_first_pass retry=3 minlen=4 dcredit=0
ucredit=0
password sufficient  pam_tcb.so use_authtok shadow write_to=shadow nullok
prefix=$2a$ count=8
password required    pam_deny.so
password sufficient  pam_ldap.so

session optional    pam_keyinit.so revoke
session required    pam_limits.so
session [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
-session optional   pam_systemd.so
session required    pam_tcb.so
session sufficient  pam_ldap.so
```

Relancer LDAP sur le client (ou le serveur) maintenant

```
systemctl restart nslcd
```

ou lancer le s'il ne l'était pas à l'origine

```
systemctl start nslcd
```

A noter que le fichier `/etc/ldap.conf` ne sert plus à rien, vous pouvez l'ignorer superbement. Pour connaître l'état du daemon on tapera

```
systemctl status nslcd.service
```

voilà le résultat

```
● nslcd.service - Naming services LDAP client daemon
   Loaded: loaded (/usr/lib/systemd/system/nslcd.service; enabled)
   Active: active (running) since sam. 2016-12-24 11:22:34 CET; 4min 19s ago
   Process: 21038 ExecStart=/usr/sbin/nslcd (code=exited, status=0/SUCCESS)
  Main PID: 21041 (nslcd)
   CGroup: /system.slice/nslcd.service
           └─21041 /usr/sbin/nslcd
```

```
déc. 24 11:22:34 mana.kervao.fr nslcd[21041]: version 0.9.7 starting
```

```
déc. 24 11:22:34 mana.kervao.fr nslcd[21041]: accepting connections
```

si vous obtenez dans les logs du serveur ce genre de messages

```
déc. 23 19:25:06 mana.kervao.fr slapd[12893]: <= bdb_equality_candidates:
(uidNumber) not indexed
```

```
déc. 23 19:25:07 mana.kervao.fr slapd[12893]: <= bdb_equality_candidates:
(uidNumber) not indexed
```

déc. 23 19:25:07 mana.kervao.fr slapd[12893]: <= bdb\_equality\_candidates:  
(uidNumber) not indexed

sur le serveur il faut revenir au fichier de configuration de **slapd** à savoir **slapd.conf** et juste après

**# Indices to maintain**

```
index objectClass eq
```

on va déclarer d'autres index pour la base de données LDAP

```
index uid pres,sub,eq
```

```
index member eq
```

```
index uidNumber eq
```

```
index gidNumber eq
```

```
index memberUid eq
```

on stoppe **nsld** sur les clients (et le serveur)

```
systemctl stop nsld.service
```

on stoppe **slapd** sur le serveur

```
systemctl stop slapd.service
```

on reconstruit les index en tapant

```
slapdindex
```

et on relance le serveur

```
systemctl start slapd.service
```

ainsi que **nsld** sur tous les clients (et le serveur)

```
systemctl start nsld.service
```

## 10.4 Test de fonctionnement

C'est simple que ce soit sur le serveur ou le client, supprimer les lignes qui correspondent à vos utilisateur dans **/etc/passwd**, et **/etc/shadow** et faites de même pour vos groupes utilisateurs dans **/etc/group**. N'oubliez pas de faire une sauvegarde de ces fichiers au cas où ! Maintenant essayer de vous loguer en tant que simple utilisateur, et là normalement, vous devriez vous loguer sans problème.

## 10.5 Gestion des utilisateurs et groupes

### 10.5.1 Créer un nouvel utilisateur

Maintenant pour créer un nouvel utilisateur, **useradd** ne fonctionne pas, car il repose uniquement sur **/etc/passwd**. On doit d'abord définir un mot de passe. On se sert pour cela de la fonction **slappasswd**

```
slappasswd -v -s toto345 -h {CRYPT}
```

On obtient

```
{CRYPT}rURm18fYhMvew
```

Il faudra créer un fichier **new.ldif** qui aura cette tête là :

```
dn: uid=utilisateur,ou=People,dc=kervao,dc=fr  
uid: utilisateur  
cn: utilisateur  
objectClass: account  
objectClass: posixAccount  
objectClass: top  
objectClass: shadowAccount  
userPassword: {CRYPT}rURm18fYhMvew  
shadowLastChange: 11858  
shadowMax: 99999  
shadowWarning: 7  
shadowInactive: -1  
shadowExpire: -1  
shadowFlag: 1081428222  
loginShell: /bin/bash  
uidNumber: 5004  
gidNumber: 5000  
homeDirectory: /home/utilisateur  
gecos: utilisateur
```

Pour mémoire voici la signification de chacun des paramètres des shadow passwords

**shadowLastChange:** date de dernière modification (en jour depuis le 1.1.70),  
**shadowMax:** nombre de jours d'utilisation max du mot de passe (changement requis à l'issue), pas de période de validité si égal à 99999  
**shadowWarning:** nombre de jours avant l'expiration pour avertir l'utilisateur ,  
**shadowInactive:** nombre de jours après la date de l'expiration où on rend le compte inactif, fonctionnalité désactivé si égal à -1  
**shadowExpire:** nombre de jours après le 1.1.70 où le compte sera désactivé, fonctionnalité désactivée si égal à -1  
**shadowFlag:** ne sert à rien (dispo pour une utilisation future) .

**ATTENTION** Il ne doit pas y avoir de blanc ou de tabulation à la fin des lignes de votre fichier ldif

On tape **ldapadd** pour la saisie de l'utilisateur dans la base

```
ldapadd -x -D "cn=Manager,dc=kervao,dc=fr" -f /tmp/new.ldif -W  
Enter LDAP Password:  
adding new entry "uid=utilisateur,ou=People,dc=kervao,dc=fr"
```

C'est pas tout il faut indiquer maintenant que cet utilisateur appartient bien au groupe **hoarau**  
On crée ce fichier **groupe.ldif** contenant

```
dn: cn=hoarau,ou=Group,dc=kervao,dc=fr  
Add: memberUid  
memberUid: utilisateur
```

On modifie la base en tapant



```
ldapmodify -x -D "cn=Manager, dc=kervao, dc=fr" -W -f groupe.ldif
Enter LDAP Password:
modifying entry "cn=hoarau,ou=Group,dc=kervao,dc=fr"
```

Par contre il faudra créer manuellement la home directory en tant que root en tapant :

```
mkdir /home/utilisateur
chown -R utilisateur:hoarau /home/utilisateur
cp -R /etc/skel/* /home/utilisateur
```

### 10.5.2 Rajouter un groupe

Sans rentrer dans les détails des commandes, le fichier ldif à créer pour la saisie d'un nouveau groupe est le suivant

```
dn: cn=newgroupe,ou=Group,dc=kervao,dc=fr
objectClass: posixGroup
objectClass: top
cn: newgroupe
gidNumber: 5000
```

Voilà le fichier pour rajouter des utilisateurs au groupe

```
dn: cn=newgroupe,ou=Group,dc=kervao,dc=fr
Add: memberUid
memberUid: new-utilisateur
```

### 10.5.3 Supprimer un utilisateur

Pour supprimer l'utilisateur **utilisateur** on tapera

```
ldapdelete -x -D "cn=Manager, dc=kervao, dc=fr"
"uid=utilisateur,ou=People,dc=kervao,dc=fr" -W
```

## 10.6 Changer son mot de passe

Maintenant le problème consiste à changer le mot de passe sans avoir à passer par le **Manager**. Un simple utilisateur doit pouvoir changer son propre mot de passe, prenons le cas de l'utilisateur **olivier**, il doit d'abord en trouver un avec la commande **slappasswd**

```
slappasswd -v -s tutu728 -h {CRYPT}
```

Voilà le résultat

```
{CRYPT}WW6h470hoW4nI
```

Il crée maintenant le fichier **modif.ldif** contenant

```
dn: uid=olivier,ou=People,dc=breizland,dc=bz
changetype: modify
```

```
replace: userPassword
userPassword: {CRYPT}WW6h470hoW4nI
```

Et il modifie la base en tapant

```
[olivier@asterix olivier]$ ldapmodify -x -D "uid=olivier, ou=People, dc=kervao, dc=fr"
-f /tmp/modif.ldif -W
Enter LDAP Password: (mot de passe d'olivier)
modifying entry "uid=olivier, ou=People, dc=kervao, dc=fr"
```

Voilà son mot de passe a été changé. Vous me direz c'est bien compliqué, y a plus simple !

En tant que root modifiez le fichier `/etc/pam.d/passwd` (rajoutez uniquement les lignes concernant `pam_ldap` et laissez les autres) pour lire

```
##%PAM-1.0
auth    include    system-auth
auth    sufficient pam_ldap.so
account sufficient pam_ldap.so
account include    system-auth
password sufficient pam_ldap.so
password substack  system-auth
password optional  pam_gnome_keyring.so
```

Maintenant pour changer son mot de passe `olivier` tape le plus simplement du monde

```
passwd
```

Voilà le résultat

```
Enter login(LDAP) password:
New password:
Re-enter new password:
LDAP password information changed for olivier
passwd: all authentication tokens updated successfully
```

Ça marche !!!!

## 10.7 Suggestion de mise en place

Pour l'administration je vous suggère **admin4** qu'on peut récupérer par ici <http://www.admin4.org/> on le décompresse en tapant

```
unzip Admin4-2.2.2-Src.zip
```

cela donne le répertoire **Admin4-2.2.2-Src**. On installera préalablement les packages suivants

```
wxPython
dnspython
python-requests
python-ldap
```

**postgresql9.4-devel**  
**lib64python-devel**

puis on installe **psycopg2** qu'on trouvera par ici <https://pypi.python.org/pypi/psycopg2> on décompresse l'archive en tapant

**tar xvzf psycopg2-2.6.2.tar.gz**

dans le répertoire **psycopg2-2.6.2** on tape

**python setup.py build**

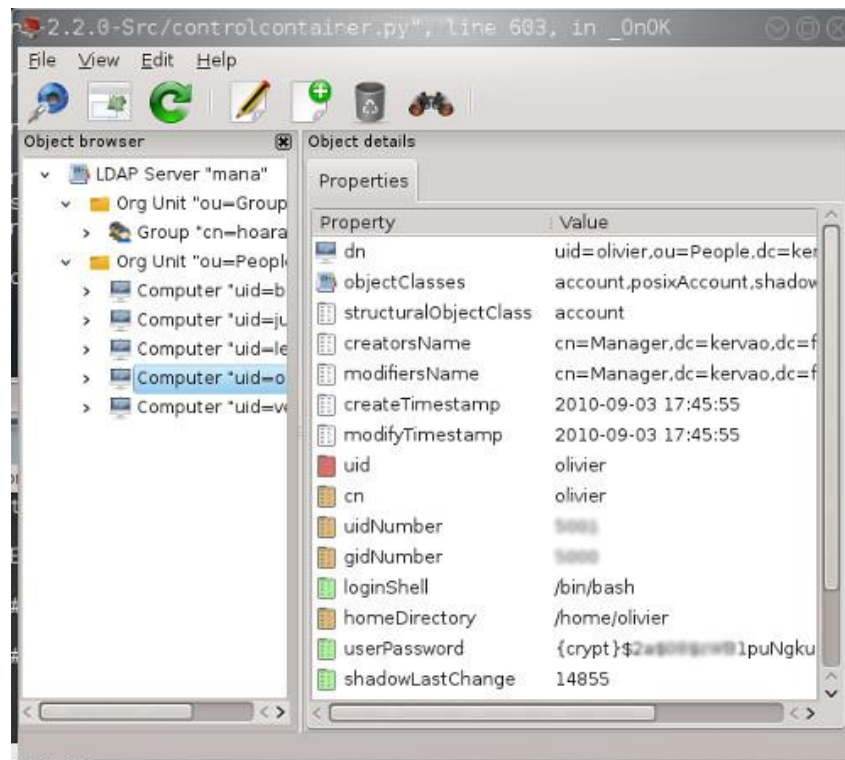
puis en tant que root

**python setup.py install**

on revient au répertoire **Admin4-2.2.2-Src** et on tape

**python admin4.py**

et voilà le résultat



Il existe également **luma** mais qui n'est visiblement plus en développement, le site officiel est <http://luma.sourceforge.net/>

