

# OpenLDAP

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

V1.4 du 5 novembre 2004

1	Historique.....	3
2	Préambule.....	3
3	Présentation.....	3
4	Format de la base et définitions.....	4
4.1	Le Directory Information Tree.....	4
4.2	Les attributs.....	5
4.3	Les classes d'objet.....	5
4.4	Les schémas.....	5
5	Installation d'OpenLDAP.....	6
5.1	Présentation.....	6
5.2	Installation par tarball.....	6
5.3	Installation par RPM.....	7
6	Mettre en place son schéma d'annuaire.....	7
6.1	Mise en place des classes d'objet.....	7
6.2	Choix du suffixe.....	8
7	Configuration du serveur LDAP.....	9
8	Lancement du serveur.....	13
9	Utilisation sommaire.....	15
9.1	Ajouter un enregistrement.....	15
9.2	Rechercher un enregistrement.....	18
9.3	Modifier un enregistrement.....	19
9.3.1	Rajouter un attribut à un enregistrement.....	19
9.3.2	Modifier un attribut.....	19
9.3.3	Supprimer un attribut.....	20
9.4	Supprimer un enregistrement.....	20
10	Authentification des utilisateurs avec LDAP.....	21
10.1	Présentation.....	21
10.2	Installation.....	21
10.2.1	Installation avec packages Mandrake 9.0.....	21
10.2.2	Installation avec les tarballs.....	21
10.3	Configuration.....	22
10.3.1	Configuration d'un serveur.....	22
10.3.2	Configuration serveur et client.....	27
10.4	Test de fonctionnement.....	30
10.5	Gestion des utilisateurs et groupes.....	30
10.5.1	Créer un nouvel utilisateur.....	30
10.5.2	Rajouter un groupe.....	32
10.5.3	Supprimer un utilisateur.....	32

10.6Changer son mot de passe.....	32
10.7Suggestion de mise en place.....	33

# 1 Historique

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, ajout 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 peut être reproduit et distribué librement dès lors qu'il n'est pas modifié et qu'il soit toujours fait mention de son origine et de son auteur, si vous avez l'intention de le modifier ou d'y apporter des rajouts, contactez l'auteur pour en faire profiter tout le monde.

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 le 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](http://www.funix.org) 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 2.1.25 fournie avec la Mandrake 10.0 ou la dernière version stable 2.2.18 disponible à l'URL <http://www.openldap.org>. Je présente les deux installations sachant que ma préférence va vers la recompilation pour avoir une version optimisée et de plus récente.

L'installation par tarball crée des fichiers de configuration dans **/usr/local/etc/openldap**, alors que l'installation rpm les crée dans **/etc/openldap**, dans la suite des opérations on notera cela (**/usr/local/etc/openldap/**).

## 5.2 Installation par tarball

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 mdk 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-2.2.28.tgz** qu'on décompressera en tapant :

```
tar xvfz openldap-2.2.28.tgz
```

Cela va nous créer un répertoire **openldap-2.2.28**. Avant d'aller plus loin il faudra installer (commande **urpmi nom-package**) les packages suivants de la Mandrake 10.0

**libgdbm2**

**libgdbm2-devel**

En effet par défaut **openldap** utilise une base de donnée Berkeley dans sa version 4.2 qui est utilisée, comme la 10.0 ne propose que la 4.1, je me suis rabattu sur la base de donnée de type **ldbm**. Puis on tape successivement :

```
./configure --enable-crypt --disable-bdb --enable-ldbm
```

```
make depend
```

```
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/openldap** et les bases dans **/usr/local/var/openldap-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
```

## 5.3 Installation par RPM

Il suffit d'installer (commande **urpmi openldap**) les packages suivants (dans l'ordre) :

```
libldap2-2.1.25-6mdk
```

```
openldap-2.1.25-6mdk
```

```
perl-ldap-0.31-2mdk
```

```
openldap-servers-2.1.25-6mdk
```

```
openldap-clients-2.1.25-6mdk
```

Vous pouvez maintenant installer aussi le package **php-ldap** pour le support **LDAP** de **PHP** avec **Apache**.

# 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/openldap/schema/core.schema** (**/usr/share/openldap/schema/core.schema** pour une install avec package) 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 $ l ) )
```

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 conf **slapd.conf**

Le fichier dans sa version tarball

```
# $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

# 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

#####
# database definitions
#####

database    ldbm
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
```

```

# Droits d'accès que j'ai rajoutés en plus
access to attr=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

```

Le même fichier dans sa version RPM

```

# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.8.8.6 2001/04/20 23:32:43 kurt
Exp $
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
# Modified by Christian Zoffoli <czoffoli@linux-mandrake.com>
# Version 0.2
#

include /usr/share/openldap/schema/core.schema
include /usr/share/openldap/schema/cosine.schema
include /usr/share/openldap/schema/corba.schema
include /usr/share/openldap/schema/inetorgperson.schema
include /usr/share/openldap/schema/java.schema
include /usr/share/openldap/schema/krb5-kdc.schema
include /usr/share/openldap/schema/kerberosobject.schema
include /usr/share/openldap/schema/misc.schema
include /usr/share/openldap/schema/nis.schema
include /usr/share/openldap/schema/openldap.schema
include /usr/share/openldap/schema/autofs.schema
include /usr/share/openldap/schema/samba.schema
include /usr/share/openldap/schema/kolab.schema

#include /usr/share/openldap/schema/rfc822-MailMember.schema
#include /usr/share/openldap/schema/pilot.schema
#include /usr/share/openldap/schema/qmail.schema
#include /usr/share/openldap/schema/mull.schema
#include /usr/share/openldap/schema/netcape-profile.schema
#include /usr/share/openldap/schema/trust.schema
#include /usr/share/openldap/schema/dns.schema

```

```

#include /usr/share/openldap/schema/cron.schema

include /etc/openldap/schema/local.schema

# Define global ACLs to disable default read access.
include /etc/openldap/slapd.access.conf

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral ldap://root.openldap.org

pidfile /var/run/ldap/slapd.pid
argsfile /var/run/ldap/slapd.args

modulepath /usr/lib/openldap
#moduleload back_dnssrv.la
#moduleload back_ldap.la
#moduleload back_meta.la
#moduleload back_monitor.la
#moduleload back_passwd.la
#moduleload back_sql.la

# SASL config
#sasl-host ldap.example.com

# To allow TLS-enabled connections, create /etc/ssl/openldap/ldap.pem
# and uncomment the following lines.
#TLSEndFile /dev/random
#TLSCipherSuite HIGH:MEDIUM:+SSLv2
TLSCertificateFile /etc/ssl/openldap/ldap.pem
TLSCertificateKeyFile /etc/ssl/openldap/ldap.pem
#TLSCACertificatePath /etc/ssl/openldap/
#TLSCACertificateFile /etc/ssl/cacert.pem
TLSCACertificateFile /etc/ssl/openldap/ldap.pem
#TLSVerifyClient never # ([never]|allow|try|demand)

# logging
loglevel 256

#####
# database definitions
#####

```

```

database      bdb
suffix        "dc=kervao,dc=fr"
#suffix       "o=My Organization Name,c=US"
rootdn        "cn=Manager,dc=kervao,dc=fr"
#rootdn       "cn=Manager,o=My Organization Name,c=US"

# Cleartext passwords, especially for the rootdn, should
# be avoided. See slapd.conf(5) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw        secret
# rootpw      {crypt}ijFYncSNctBYg

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory     /var/lib/ldap

# Indices to maintain
#index objectClass                      eq
index objectClass,uid,uidNumber,gidNumber eq
index cn,mail,surname,givenname        eq,subinitial
# samba searches on sid
#index sambaSID                        eq

# Basic ACL (deprecated in favour of ACLs in /etc/openldap/slapd.access.conf)
access to attr=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

# ACL ensuring replicator has write access
#access to *
#    by group="cn=Replicator,ou=Group,dc=example,dc=com" write
#    by * read

# Replica configuration (if this server is a slave)
#updatedn      "cn=ldap-master.example.com,ou=Hosts,dc=example,dc=com"
#updateref     "ldap://ldap-master.example.com"

# Replication configuration (if this server is a master)

```

```
#replica host=ldap-slave1.example.com:389
#   binddn="cn=ldap-master.example.com,ou=Hosts,dc=example,dc=com"
#   bindmethod=simple credentials="mypassword"

# Uncomment to enable statistics gathering at basedn cn=monitor (load monitor
# module above too)
#database monitor
```

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.

Pour une install avec package il existe des lignes supplémentaires dans ce fichier, pensez à modifier les champs **dc=example,dc=com** en adaptant à votre domaine.

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 mettre crypté, il faudra taper (exemple avec **secret**) :

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

Voilà le résultat

```
{CRYPT}G.H5krNMMw0cc
```

A la place de

```
rootpw      secret
```

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

```
rootpw      {CRYPT}G.H5krNMMw0cc
```

## 8 Lancement du serveur

Pour une installation par RPM, vous allez retrouver un fichier de lancement **ldap** sous **/etc/rc.d/init.d**. Pour l'installation par tarball, voici un fichier **ldap** à placer sous **/etc/rc.d/init.d**,

```
#!/bin/sh
#
# ldap This shell script takes care of starting and stopping
#   ldap servers (slapd and slurpd).
#
# chkconfig: - 70 40
# description: LDAP stands for Lightweight Directory Access Protocol, used \
#   for implementing the industry standard directory services.
# processname: slapd
# config: /usr/local/etc/openldap/slapd.conf
```

```

# pidfile: /var/run/slapd.pid

# Source function library.
./etc/rc.d/init.d/functions

# Source networking configuration.
./etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/local/libexec/slapd ] || exit 0
[ -f /usr/local/libexec/slurpd ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
  start)
    # Start daemons.
    echo -n "Starting ldap: "
    daemon /usr/local/libexec/slapd -f /usr/local/etc/openldap/slapd.conf
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
    echo
    ;;
  stop)
    # Stop daemons.
    echo -n "Shutting down ldap: "
    killproc /usr/local/libexec/slapd
    RETVAL=$?
    echo
    if [ $RETVAL -eq 0 ]; then
      rm -f /var/lock/subsys/ldap
      rm -f /var/run/slapd.args
    fi
    ;;
  status)
    status slapd
    RETVAL=$?
    ;;
  restart)
    $0 stop
    $0 start

```

```

        RETVAL=$?
        ;;
    reload)
        killproc -HUP /usr/local/libexec/slapd
        RETVAL=$?
        ;;
    *)
        echo "Usage: $0 start|stop|restart|status"
        exit 1
esac

exit $RETVAL

```

attention ce fichier utilise les chemins par défaut, vous devez le modifier si nécessaire. Vous devez aussi modifier (non nécessaire pour installation par rpm) le fichier `/etc/rc.d/init.d/functions` et à la place de :

```
export PATH="/sbin:/usr/sbin:/bin:/usr/bin:/usr/X11R6/bin"
```

On mettra

```
export PATH="/sbin:/usr/sbin:/bin:/usr/bin:/usr/X11R6/bin:/usr/local/libexec"
```

pour que serveur **LDAP** soit lancé automatiquement à l'état de marche 3, 4 et 5 ( les deux types d'installation) on tapera :

```
chkconfig --level 345 ldap on
```

Pour l'arrêter à l'état de marche 0, 1, 2 et 6, on tapera:

```
chkconfig --level 0126 ldap off
```

Au prochain reboot le serveur sera lancé automatiquement, pour éviter un reboot pour lancer le serveur, il suffit de taper :

```
/etc/rc.d/init.d/ldap start
```

## 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**

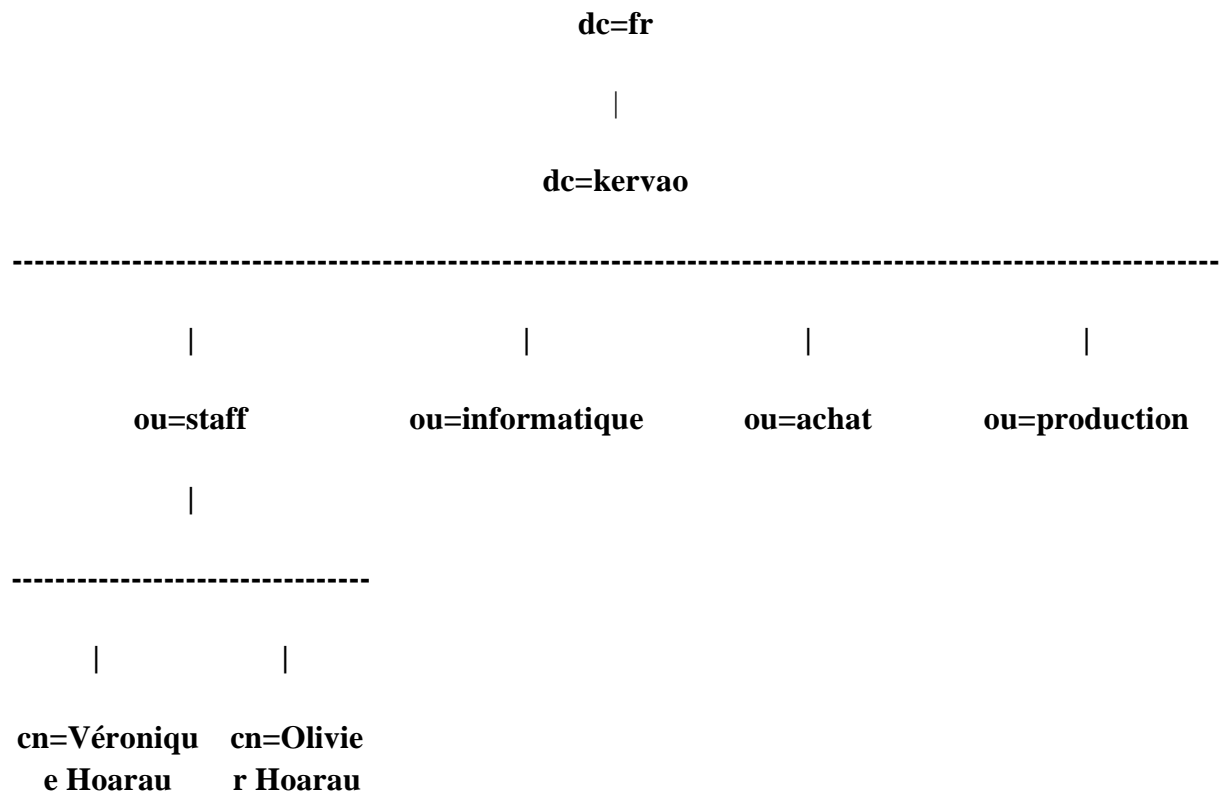
**sn: Hoarau**

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

```
dn: cn=Veronique Hoarau, ou=staff, dc=kervao, dc=fr
add: l
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**.

# 10Authentification des utilisateurs avec LDAP

## 10.1Présentation

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.

Je vous montre l'installation de l'authentification **LDAP** en utilisant les packages de la Mandrake 9.1 ainsi que les tarballs avec utilisation des "shadow passwords".

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.

## 10.2Installation

### 10.2.1 Installation avec packages Mandrake 9.0

Pour un serveur, on installera les packages suivants que l'on trouvera sur les CD d'install

**nss\_ldap**  
**openldap-migration**  
**pam\_ldap**

Pour un client, il ne sera pas nécessaire d'installer le package **openldap-migration**.

### 10.2.2 Installation avec les tarballs

On les récupèrera sur le site [www.padl.com](http://www.padl.com), cela concerne les tarballs suivants **pam\_ldap.tgz**, **nss\_ldap.tgz** et **MigrationTools.tgz**

On désarchive la première archive en tapant

**tar xvfz pam\_ldap.tgz**

Cela donne le répertoire **pam\_ldap-176** 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**

Pour la deuxième archive on tape

**tar xvfz nss\_ldap.tgz**

Cela donne le répertoire **nss\_ldap-226** on tape successivement

**./configure**

**Make**

Puis en tant que root

**make install**

Pour le dernier tarball on tape

**tar xvfz MigrationTools.tgz**

Cela donne le répertoire **MigrationTools-46**.

## 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 attr=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

## **/etc/rc.d/init.d/ldap restart**

A présent dans le répertoire **/usr/share/openldap/migration** (si installation par package, sinon dans le répertoire **MigrationTools-46**) , 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

```

**ATTENTION** Avec une installation par RPM la ligne suivante est absente au niveau de chaque utilisateur,

```
objectClass: account
```

Il faudra la rajouter manuellement sinon vous aurez des erreurs au rajout des utilisateurs (erreur **no structural object class provided**).

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: 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, dans la base, on commence par les **ou Group** et **People**

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

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::
e2NyeXB0fSQxJE95aWRVb0lVJHV3cFlOM7JXSkd6RjRBRkFlc3BTbS8=
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

Editer le fichier `/etc/ldap.conf`

```

# Your LDAP server. Must be resolvable without using LDAP.
# ici vous devez mettre l'adresse IP de votre serveur
host 192.168.13.11

# The distinguished name of the search base.
# votre nom de domaine
base dc=kervao,dc=fr

```

(...)

**# The distinguished name to bind to the server with.**

**# Optional: default is to bind anonymously.**

**# pour pouvoir se connecter à votre base**

**binddn cn=Manager,dc=kervao,dc=fr**

**# The credentials to bind with.**

**# Optional: default is no credential.**

**# le mot de passe qui va bien**

**bindpw secret**

(...)

**# Filter to AND with uid=%s**

**pam\_filter objectclass=account**

**# The user ID attribute (defaults to uid)**

**pam\_login\_attribute uid**

(...)

**# Hash password locally; required for University of**

**# Michigan LDAP server, and works with Netscape**

**# Directory Server if you're using the UNIX-Crypt**

**# hash mechanism and not using the NT Synchronization**

**# service.**

**# type de cryptage du mot de passe**

**pam\_password crypt**

**# RFC2307bis naming contexts**

**# Syntax:**

**# nss\_base\_XXX        base?scope?filter**

**# where scope is {base,one,sub}**

**# and filter is a filter to be &'d with the**

**# default filter.**

**nss\_base\_passwd ou=People,dc=kervao,dc=fr?one**

**nss\_base\_shadow ou=People,dc=kervao,dc=fr?one**

**nss\_base\_group        ou=Group,dc=kervao,dc=fr?one**

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/login** . 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 (login). Sans rentrer dans le détail, ce fichier et tous 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 **login** (rajoutez uniquement les lignes concernant **pam\_ldap** et laissez les autres), il doit ressembler à ça :

**##PAM-1.0**

```
auth    required    pam_securetty.so
auth    required    pam_stack.so service=system-auth
auth    required    pam_nologin.so
auth    sufficient  /lib/security/pam_ldap.so
account sufficient  /lib/security/pam_ldap.so
account required    pam_stack.so service=system-auth
password sufficient  /lib/security/pam_ldap.so
password required    pam_stack.so service=system-auth
session sufficient  /lib/security/pam_ldap.so
session required    pam_stack.so service=system-auth
session optional    pam_console.so
```

Le fichier **su** (rajoutez uniquement les lignes concernant **pam\_ldap** et laissez les autres) ressemble à ça

**##PAM-1.0**

```
auth    sufficient  pam_rootok.so
auth    required    pam_stack.so service=system-auth
auth    sufficient  /lib/security/pam_ldap.so
account sufficient  /lib/security/pam_ldap.so
account required    pam_stack.so service=system-auth
password sufficient  /lib/security/pam_ldap.so
password required    pam_stack.so service=system-auth
session sufficient  /lib/security/pam_ldap.so
session required    pam_stack.so service=system-auth
session optional    pam_xauth.so
```

Modifiez aussi le fichier **/etc/pam.d/smtp** (rajoutez uniquement les lignes concernant **pam\_ldap** et laissez les autres) car sinon vos mails vont se perdre dans la nature, le serveur **smtp** a besoin d'authentifier les utilisateurs.

**##PAM-1.0**

```
auth    required    /lib/security/pam_stack.so service=system-auth
auth    sufficient  /lib/security/pam_ldap.so
account sufficient  /lib/security/pam_ldap.so
account required    /lib/security/pam_stack.so service=system-auth
```

Modifiez les autres fichiers sous **/etc/pam.d** qui seraient susceptibles de faire appel à l'authentification **LDAP** sur le même modèle.

## 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    sufficient /lib/security/pam_ldap.so
auth    required  pam_stack.so service=system-auth
account  sufficient /lib/security/pam_ldap.so
account  required  pam_stack.so service=system-auth
password sufficient /lib/security/pam_ldap.so
password required  pam_stack.so service=system-auth
```

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
```

Ca marche !!!!

## 10.7 Suggestion de mise en place

Je vous suggère de créer une console d'administration en utilisant le module **LDAP** de **PHP** (voir document Apache sur le site <http://www.funix.org>) pour pouvoir faire une gestion des utilisateurs à partir d'un simple navigateur sur une machine (windows ou UNIX) quelconque de votre réseau.

L'idéal serait de disposer d'une console d'administration pour le Manager pour créer, supprimer, modifier des comptes et groupes. Une console utilisateur pour modifier son

propre mot de passe et éventuellement d'autres informations qu'on pourrait rajouter au niveau d'un compte (email, numéro de téléphone, bureau, ...).