Formation OpenLDAP

Ganaël Laplanche - **http://contribs.martymac.com**, 2005-2010

Ver.	***Auteur***	***Date***	***Description***
1.0	Ganaël LAPLANCHE	19/11/2005	Version initiale
1.1	Ganaël LAPLANCHE	05/11/2006	Relecture et modifications diverses
1.2	Ganaël LAPLANCHE	02/12/2006	Suppr. chaînage passdb backend (Samba v3.0.23)
1.3	Ganaël LAPLANCHE	02/01/2009	Mise à jour des RFCs - Cf. RFC 4010, juin 2006
1.4	Ganaël LAPLANCHE	14/01/2010	Mise à jour OpenLDAP v2.4 + corrections config

Table des matières

Avant-propos

Présentation et objectifs du cours

Ce cours a pour objectif de vous présenter les annuaires LDAP et une implémentation libre proposant à la fois un serveur LDAP mais un ensemble d'outils clients : le projet OpenLDAP.

Vous allez découvrir à travers ce cours les notions liées aux annuaires LDAP. Vous apprendrez également comment mettre en oeuvre un serveur OpenLDAP et utiliser les commandes clientes fournies par le projet. Un dernier chapitre vous proposera un exercice concret : la connexion d'un serveur Samba à l'annuaire mis en place.

La durée prévue du cours est de 8h. Cette durée peut varier d'une personne à une autre, progressez à votre rythme !

Organisation du travail

Le cours est divisé en trois parties majeures :

- une partie théorique concernant les annuaires LDAP (chapitres I à V)
- une partie pratique présentant le projet OpenLDAP (chapitre VI)
- une partie pratique présentant comment connecter Samba à OpenLDAP (chapitre VII)

La première partie est purement théorique. Elle présente les concepts liés à tous les annuaires LDAP. Il est indispensable de bien assimiler les notions qui y sont présentées car elles constituent la base de ce qui est présenté dans la seconde partie avec OpenLDAP.

Cette seconde partie présente comment mettre en place un annuaire. De nombreuses commandes seront utilisées, il est conseillé d'effectuer les manipulations en même temps sur l'ordinateur. Cette partie constitue en quelques sortes un exercice où il est expliqué pas à pas comment configurer et utiliser l'annuaire.

Enfin, la dernière partie présente comment il est possible de mettre en place un serveur autonome Samba/LDAP. Les comptes POSIX et Samba seront stockés sur l'annuaire. Il est indispensable d'avoir suivi la formation Samba avant d'aborder cette partie.

Le découpage horaire du cours est laissé à votre convenance, mais je vous conseille ceci :

- chapitres I à V : 2h
- chapitre VI : 3h
- chapitre VII : 3h

Pré-requis

Les pré-requis pour suivre cette formation sont les suivants :

- Maîtriser le shell et les commandes systèmes GNU/linux de base
- Maîtriser la gestion des droits Unix
- Maîtriser la gestion des utilisateurs
- Maîtriser Samba (exercice final)

Pré-requis matériels

Les exercices de ce document nécessitent une machine GNU/Linux, distribution de préférence basée sur Debian.

Voici la fiche signalétique de la machine utilisée dans ce document :
- OS : GNU/Linux Ubuntu (**http://www.ubuntulinux.org**)
- Utilisateur (standard) : martymac

Conventions utilisées dans ce document

Les conventions syntaxiques utilisées dans ce document sont les suivantes :

```
Ceci est le contenu d'un fichier
```

```
# Ceci est une commande exécutée en tant que root (#)
```

```
$ Ceci est une commande exécutée en tant qu'utilisateur standard ($)
```

Ceci est une note

Ceci est un texte standard

Introduction

L'informatique et la gestion de l'information prend une place de plus en plus importante dans notre société, particulièrement en entreprises. La multiplication des applications et des serveurs rend cette information difficile à maîtriser car très volative et éparse. Ceci entraîne bien souvent une obsolescence, voire une incohérence des données stockées.

Les annuaires LDAP offrent une réponse à ce problème en proposant de centraliser les informations et, par le biais d'un protocole standardisé, d'y connecter des applications clientes.

Définition d'un annuaire

Qu'est-ce qu'un annuaire ?

Vous utilisez fréquemment des annuaires : prenons l'exemple de l'annuaire téléphonique. Cet annuaire regroupe différentes entrées contenant chacune des informations particulières : nom, prénom, numéro de téléphone et adresse. Ces informations sont classées par département, puis par ville, puis enfin par nom.

Voici les caractéristiques communes aux annuaires :

- Un annuaire présente un **ensemble défini de données**
  (annuaire : nom, prénom, numéro de téléphone, adresse)
- Il **organise** ces données
  (annuaire : classées par département, villes, nom)
- Il offre un service de **consultation**
  (annuaire : diffusion au format papier)
- Il peut **protéger** les données
  (annuaire : liste rouge)
- Il est **plus consulté** que mis à jour
- Il est **disponible** de manière permanente

Le standard LDAP tient compte de ces différentes caractéristiques. Nous verrons comment elles sont mises en oeuvre au cours de la formation.

Différences avec un Système de Gestion de Bases de Données

Quelles sont les différences entre un annuaire et un Système de Gestion de Bases de Données (SBGD) ?

La principale a été évoquée ci-dessus : sur un annuaire, les écritures sont plus rares que les lectures, ce qui n'est pas forcément le cas pour un SGBD. Un annuaire n'est pas fait pour stocker des informations constamment en mouvement.

Une seconde différence est qu'un annuaire fournit une méthode de consultation standardisée, ce qui n'est pas le cas d'un SGBD. Le SQL est, certes, standardisé, mais la couche de connexion propre au SGBD utilisé ne l'est pas. Un client MySQL est différent d'un client PostgreSQL par exemple.

Un annuaire LDAP organisent les données de manière arborescente, tandis que les bases de données le font au sein de tableaux à deux dimensions

Enfin, un annuaire fournit des modèles de données standardisés. Alors que le modèle conceptuel de données (que stocker, où et comment ?) d'un SGBD peut varier d'une entreprise et d'une base à une autre, un annuaire fournit des modèles de données officialisés (par le biais des schémas, nous le verrons par la suite). Ceci procure une capacité d'interopérabilité sans commune mesure.

Avant de poursuivre, je vous propose de retracer un bref historique des annuaires informatiques...

Historique et aperçu des annuaires existants

Historique

En 1988, l'Union Internationale des Communications (UIT) met au point les annuaires X.500. Le but de cette opération est d'uniformiser l'accès aux services, de centraliser les ressources et de les protéger. Le protocole utilisé pour y accéder est le protocole DAP (Directory Access Protocol).

Malheureusement, le protocole DAP s'avère difficile à mettre en oeuvre et ne fonctionne pas sur les réseaux TCP/IP. En 1993, l'Université du Michigan réfléchit donc à un moyen de palier ces deux problèmes : elle met en place le protocole LDAP (Lightweight Directory Access Protocol), au départ simple "connecteur" TCP/IP avec des annuaires X.500.

En 1995, LDAP devient un protocole natif et utilisable indépendamment de X.500.

LDAP est donc une évolution de la norme X.500. Sa version actuelle est la version 3 (RFCs 2251, 4511, 4512, 4513), elle propose les évolutions suivantes par rapport à la version 2 :

- Le support des communications chiffrées via SSL/TLS
- L'authentification via SASL
- Le support des Referrals (une branche pointe vers un autre annuaire)
- Le support d'Unicode (internationalisation)
- La capacité d'étendre le protocole
- Le support des schémas dans l'annuaire

D'autres types d'annuaires...

D'autres types d'annuaires existent, vous les utilisez très certainement :

- DNS : Domain Name Services
- NIS : Network Information Services
- Whois : base d'information concernant les noms de domaines

Quelques annuaires LDAP

Voici une liste des principaux annuaires LDAP existant sur le marché :

- OpenLDAP : **http://www.openldap.org**
- Apache Directory Server : **http://directory.apache.org**
- Sun (One/Java) Directory Server : **http://www.sun.com**
- Active Directory : **http://www.microsoft.com**
- [...]

Les concepts du protocole LDAP

Quatre modèles

On a coutume de regrouper les caractéristiques et fonctionnalités de l'annuaire LDAP sous la forme de quatre modèles :

- Le modèle de nommage : définit comment l'information est stockée et organisée
- Le modèle fonctionnel : définit les services fournis par l'annuaire (recherche, ajout, ...)
- Le modèle d'information : définit le type d'informations stockées
- Le modèle de sécurité : définit les droits d'accès aux ressources

Ces différents points seront abordés au cours de la formation.

Un protocole

LDAP signifie "Lightweight Directory Access Protocol".

LDAP est un protocole, ce qu'il signifie que son rôle est de présenter des informations. Un serveur LDAP agit en tant qu'intermédiaire entre une source de données et un client.

Nous verrons qu'en tant qu'intermédiaire il définit quelques conventions, notamment l'organisation des données qu'il présente qui sera sous forme hiérarchique, mais aussi un format d'échange standard.

LDAP fonctionne sur le port TCP 389 (par défault).

Organisation des données (modèle de nommage)

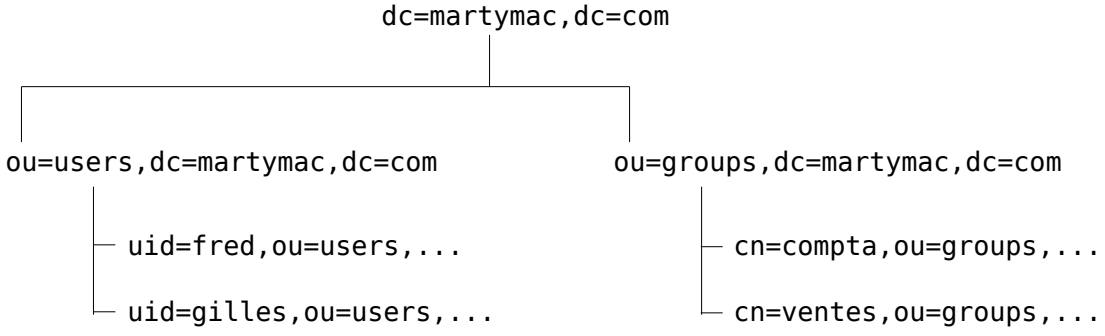
Introduction

Le modèle de nommage est la manière dont sont organisées les données dans l'annuaire. Etudions cette organisation plus en détails...

Une représentation hiérarchique des données

LDAP organise les données de manière hiérarchique dans l'annuaire. Ceci signifie que toutes les informations découlent d'une seule et même "racine".

Voici un exemple d'arborescence LDAP :



Cette arborescence est liée au nommage de chaque élément : un élément marque son appartenance à l'élément supérieur en en reprenant le nom, qu'il complète par le sien.

Ainsi, en étudiant simplement le nom de l'élément :

"cn=ventes,ou=groups,dc=martymac,dc=com"

il est possible de le situer dans la hiérarchie : il est situé sour l'élément "**ou=groups**" qui lui-même est situé sous l'élément "**dc=martymac,dc=com**".

Termes à connaître

Avant de poursuivre, voici quelques termes à connaître :

- Chaque élément est appelé une **entrée** (an entry). Une entrée peut être un branchement (un **noeud**, a node) ou un élément terminal (une **feuille**, a leaf).
- Chaque élément possède un **DN** (Distinguished Name). Le DN est le nom complet de l'élément qui permet de le positionner dans l'arborescence. Il est unique dans l'annuaire.
  Exemple : "cn=ventes,ou=groups,dc=martymac,dc=com"
- Chaque élément possède également un **RDN** (Relative Distinguished Name). Le RDN est la partie du **DN** de l'élément qui est relative au **DN** supérieur. Le RDN d'un élément ne permet pas de l'identifier de manière abolue dans l'annuaire.
  Exemple : "cn=ventes"
- La **racine** est l'élément supérieur de tous les autres, c'est la base de l'arborescence. On l'appelle **root** en anglais, parfois on parle de "**root DN**".
  Exemple : "dc=martymac,dc=com"

Les DN de chaque entrées sont composés au moins d'un attribut de l'élément (par exemple "cn" ou "uid") et de sa valeur. Un attribut est l'une des caractéristiques de cet élément.

Remarquez que la racine choisie ici est composée du nom du domaine où est hébergé notre serveur LDAP, martymac.com, décomposé en "dc" (Domain Components) pour obtenir dc=martymac,dc=com.

L'arbre se découpe ensuite en deux "ou" (Organisational Units) qui constituent deux branchements : "users" et "groups", dans lesquels nous trouvons ensuite les entrées feuilles de notre arbre : les utilisateurs et les groupes.

Chacune des entrées de notre arbre correspond à un type de donnée particulier, défini par une classe d'objet. Nous étudierons ces notions par la suite.

Règles de nommage

La RFC 2253 (rendue obsolète par la RFC 4514) normalise l'écriture des DN et conseille de ne pas ajouter d'espaces autour du signe "=", ni à la fin du DN. Les espaces sont autorisés par contre pour les valeurs des entrées.

Ainsi, le DN suivant est correct :

"cn=Ganael Laplanche,cn=ventes,ou=groups,dc=martymac,dc=com"

Alors que celui-ci ne l'est pas :

"cn = Ganael Laplanche, cn = ventes, ou = groups, dc = martymac, dc = com"

Les majuscules seront ou non prises en compte en fonction du type d'attribut utilisé et de ses particularités.

Accéder à l'annuaire (modèle fonctionnel)

Il existe plusieurs types d'opérations que l'on peut effectuer sur l'annuaire, voici les plus importantes :

- Rechercher une entrée suivant certains critères
- S'authentifier
- Ajouter une entrée
- Supprimer une entrée
- (Modifier une entrée)
- Renommer une entrée

Certaines de ces actions, notamment la recherche, nécessitent des outils particuliers pour nous faciliter l'accès à l'annuaire

La base

La base est le DN à partir duquel nous allons agir. Pour une recherche, il s'agir du noeud à partir duquel est effectuée la recherche. Il peut s'agir de la racine de l'arbre pour une recherche sur la totalité de l'arbre, par exemple "dc=martymac,dc=com".

La portée

La portée (scope) est le nombre de niveaux sur lesquels l'action va être effectuée. Il existe 3 niveaux différents :

- SUB : l'action est effectuée récursivement à partir de la base spécifiée sur la totalité de l'arborescence.
- ONE : l'action est effectuée sur un seul niveau inférieur par rapport à la base spécifiée (les fils directs). Si l'on effectuait une recherche avec la portée ONE à partir de "dc=martymac,dc=com", nous pourrions trouver `"ou=users,dc=martymac,dc=com"` et `"ou=groups,dc=martymac,dc=com"`.
- BASE : l'action est effectuée uniquement sur la base spécifiée. Une recherche sur "dc=martymac,dc=com" avec la portée BASE renverrait cette entrée uniquement.

Les filtres

Le troisième outil à notre disposition est le filtre. Un filtre va permettre d'effectuer des tests de correspondance lors d'une recherche. Il s'agit en quelques sortes du critère de la recherche.

Il existe 4 tests basiques, qui peuvent ensuite être combinés :

- Le test d'égalité : **X=Y**
- Le test d'infériorité : **X<=Y**
- Le test de supériorité : **X>=Y**
- Le test d'approximation : **X~=Y**

Les autres opérateurs (<, >) ou des tests plus complexes peuvent être mis en place par combinaison, il faut alors utiliser les parenthèses ( ) et l'un des opérateurs suivants :

- L'intersection (et) : **&**
- L'union (ou) : **|**
- La négation (non) : **!**

Un test d'inférioté stricte pourrait donner ceci : (&(X<=Y)(!(X=Y)))
On peut combiner plus de deux éléments : (&(X=Y)(Y=Z)(A=B)(B=C)(!(C=D)))

Ces filtres seront appliqués sur des attributs choisis pour sélectionner finement les données que nous voulons extraire de notre annuaire.

Les URLs LDAP

Récemment est apparue une méthode concise et simplifiée pour interroger un annuaire LDAP. Il s'agit d'un format d'URL combinant toutes les notions que nous avons étudiées. En une seule ligne, il est possible de spécifier tous les éléments de notre requête. Voici le format de cette URL (RFC 2255, rendue obsolète par la RFC 4516) :

ldap[s]://serveur[:port][/[base[?[attributs à afficher][?[portée][?[filtre][?extensions]]]]]]

L'exemple ci-dessous recherche tous les uid de notre arbre, à partir de la branche users :

ldap://localhost:389/ou=users,dc=martymac,dc=com?uid?sub

Les données contenues dans l'annuaire (modèle d'information)

Les attributs

Nous avons jusqu'ici évoqué la notion d'attribut sans trop l'expliquer. Un attribut est une valeur contenue dans une entrée. Une entrée peut bien entendu contenir plusieurs attributs. Prenons l'exemple de l'entrée LDAP complète d'un compte utilisateur POSIX :

```
dn: uid=martymac,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: martymac
uid: martymac
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/martymac
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: martymac
description: martymac
```

Ceci correspond à une entrée complète, extraite par une interrogation de l'annuaire. Le format affiché est le format **LDIF**, nous y reviendrons.

Ce paragraphe présente tous les attributs, un par ligne, que comprend notre entrée. Un attribut est séparé de sa valeur par ":". Suivant son type, un attribut peut avoir plusieurs valeurs : dans ce cas, il est dit "multi-valué" et apparaît sur plusieurs lignes avec des valeurs différentes.

Nous pouvons observer ici des attributs nommés "dn", "objectClass", "cn", "uid", ...

L'attribut "dn" qui est indiqué en première ligne est le nom unique de notre entrée dans l'arbre dont nous avons parlé précédemment. Il constitue un attribut à part entière dans notre entrée. Il est composé du dn de l'entrée supérieure, ainsi que du rdn.

> *Note : le rdn est défini par un ou plusieurs attributs de l'entrée (dans ce cas séparés par un +). Il est conseillé, pour une entrée de type posixAccount, d'utiliser les attributs uid ou cn (cf. RFC 2307). Nous avons choisi ici uid=martymac.*

Sur un annuaire LDAP la racine est toujours composée des attributs "dc" (Domain Component) associés à chacune des parties du nom de domaine où est hébergé le serveur ("dc=martymac,dc=com" pour le domaine martymac.com). Ceci est une convention. X500 préconisait les attributs "o", "l" et "c", mais LDAP a simplifié le procédé (cf. RFCs 2247, 4519, 4524).

L'attribut "ou" constitue une "Organisational Unit", c'est à dire une unité organisationnelle : en quelque sorte un regroupement. Nous avons choisi d'en créer deux dans notre exemple : "users", qui accueillera nos utilisateurs et "groups", nos groupes.

Nous n'allons pas étudier chacun des attributs présents ici, cependant, je souhaiterais porter votre attention sur l'un des attributs les plus importants, il s'agit de la classe d'objet, ou "objectClass"...

Les classes d'ojets

A première vue, l'entrée présentée ci-dessus constitue un amalgame de différentes informations qui ne semblent pas organisées. Eh bien détrompez-vous ! Toutes ces entrées sont induites par la présence des objectClass.

L'objectClass d'une entrée est un attribut qui permet de cataloguer cette entrée. Un objectClass définit un regroupement d'attributs **obligatoires** ou **autorisés** pour une entrée. Une entrée peut posséder un ou plusieurs objectClass. Ce sont ces objectClass qui définissent la présence de tous les autres attributs.

Ici, l'objectClass "posixAccount" rend ***obligatoire*** les attributs cn, uid, uidNumber, gidNumber et homeDirectory. Il rend ***possible*** l'utilisation des 4 autres attributs userPassword, loginShell, gecos et description.

Les schémas

Comment savoir quels sont les objectClass disponibles et quels attributs ils contiennent ? C'est très simple, la syntaxe et la liste des attributs connus de l'annuaire sont écrits dans ce que l'on appelle les "schémas".

Un annuaire LDAP a la capacité de charger en mémoire plusieurs schémas. A travers ces schémas, il est possible de définir de nouveaux attributs et de nouveaux objectClass. Cette souplesse permet de définir très finement ce qui sera stocké dans notre annuaire.

Concrètement, un schéma est un fichier qui décrit un à un les attributs disponibles (leur nom, leur type, etc...), ainsi que les objectClass qui y font appel. Au démarrage du serveur LDAP, le ou les fichiers de schéma spécifiés dans sa configuration seront chargés.

Dans notre exemple, l'objectClass posixAccount est défini dans le fichier **nis.schema**. Etudions une partie de ce fichier, livré avec OpenLDAP et situé dans **/etc/ldap/schema** :

```
# [...]
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
        DESC 'An integer uniquely identifying a user in a domain'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

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

```
# [...]
```

Le fichier est assez volumineux et a été tronqué.

Le premier paragraphe définit l'un des attributs utilisés par le posixAccount : uidNumber. Le second, l'objectClass posixAccount. Nous n'allons pas étudier en détail ces deux définitions, simplement, sachez que :

- A chaque définition correspond un OID (Object IDentifier), qui permet de rendre unique l'attribut spécifié. Ces OIDs sont déposés auprès de l'IANA (**http://www.iana.org**) et sont donc officiels.
- Un attribut définit un type d'égalité à mettre en oeuvre lors d'une recherche (ici, intergerMatch) ainsi que le type de données qu'il contient (l'OID spécifié après SYNTAX).
- Un objectClass définit les attributs que l'objet **doit** présenter (MUST) et ceux qu'il **peut** posséder (MAY).

Les schémas constituent donc une source d'information très importante. En cas de doute concernant le type ou le nom des attributs à spécifier dans une entrée, n'hésitez pas à vous y reporter !

Enfin, sachez qu'il est tout à fait possible de créer ses propres schémas, cependant, penser à réutiliser les schémas existants : ils offrent déjà de nombreuses possibilités et il y a fort à parier qu'un schéma existe déjà pour gérer les informations que vous souhaitez !

Le format LDIF

Les données contenues dans l'annuaire sont présentées dans un certain format : il s'agit du format LDIF (LDAP Data Interchange Format - RFC 2849). Nous en avons vu un exemple dans le paragraphe précédent.

Sachez que toute intéraction avec un annuaire se fait par le biais de ce format : l'ajout, la modification, la suppression d'entrées, l'interrogation de l'annuaire y compris.

Dans ce format, chaque entrée constitue un paragraphe, et, au sein de chaque paragraphe, chaque ligne constitue un attribut. Voici un exemple un peu plus complet, incluant le groupe de notre utilisateur :

```
# [...]
dn: cn=utilisateurs,ou=groups,dc=martymac,dc=com
objectClass: posixGroup
cn: utilisateurs
gidNumber: 10001

dn: uid=martymac,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: martymac
uid: martymac
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/martymac
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: martymac
description: martymac
# [...]
```

La sécurité (modèle de sécurité)

Lorsque vous mettez en place un annuaire d'entreprise, il convient de réfléchir au modèle de sécurité que vous souhaitez appliquer. LDAP fournit plusieurs mécanismes permettant de mener à bien votre projet.

L'authentification simple, le binding

L'annuaire met en place un mécanisme d'authentification : pour avoir accès aux données qu'il contient, il faut montrer pattes blanches !

L'une des opérations préalables à l'interrogation de l'annuaire est cette opération dite de "binding" (dans le cas d'une authentification simple). Le client envoie alors le DN d'un compte contenu dans l'annuaire lui-même, ainsi que le mot de passe associé. On pourra par la suite appliquer des droits particuliers sur ce compte en utilisant les ACLs.

Ceci correspond, si l'on fait le parallèle avec l'annuaire téléphonique, à la fonctionnalité de liste rouge, où certaines données ne sont pas accessibles à tout le monde.

Notez enfin qu'il est possible de se connecter de manière anonyme : le client envoie alors un DN vide au serveur LDAP.

Les ACLs

Les ACLs (Access Control Lists) interviennent après la notion de binding. Il sera possible de donner des droits de lecture, d'écriture (ou d'autres droits divers) sur des branches particulières de l'annuaire au compte connecté.

Ceci permet de gérer finement les droits d'accès aux données. Nous verrons plus tard des exemples concrets d'ACLs.

Le chiffrement des communications (SSL/TLS)

Le chiffrement des communications, via SSL (Secure Socket Layer, ou TLS - Transport Layer Security) est également une méthode de protection de l'information. Il est possible, avec la plupart des annuaires existants, de chiffrer le canal de communication entre l'application cliente et l'annuaire. Ceci permet de garantir (un minimum) la confidentialité des données et d'éviter qu'un tiers n'écoute les communications sur le réseau.

SASL

SASL (Simple Authentication and Security Layer) est un mécanisme qui permet d'ajouter des méthodes d'authentification à des protocoles orientés connexion tels que LDAP ou IMAP. Il est défini dans la RFC 2222 ; l'implémentation la plus couramment utilisée est Cyrus-Sasl (http://asg.web.cmu.edu/sasl).

SASL donne la possibilité au client et au serveur de sélectionner quelle sera la méthode d'authentification utilisée. Ces méthodes sont extensibles via des plugins. Il permet également de mettre en place une couche de connexion sécurisée telle que SSL/TLS (sans rapport direct avec le chiffrement indépendant des connexions que nous avons cité ci-dessus).

Concepts avancés

La réplication

Certains serveurs LDAP, dont OpenLDAP, permettent de manière native, de mettre en place un annuaire répliqué. Un annuaire dit "maître" envoie alors, par le biais du format LDIF, toutes les modifications effectuées sur un annuaire "esclave".

L'avantage d'une telle opération est double :

- permettre une meilleure montée en charge pour de gros annuaires : il est possible de rediriger le client vers l'un ou l'autre des annuaires répliqués
- disposer d'une copie conforme du premier annuaire, utile en cas de crash (attention, toute opération est reportée de l'annuaire maître vers l'esclave, donc ceci est non valable en cas de mauvaise manipulation).

Deux types de réplication existent :

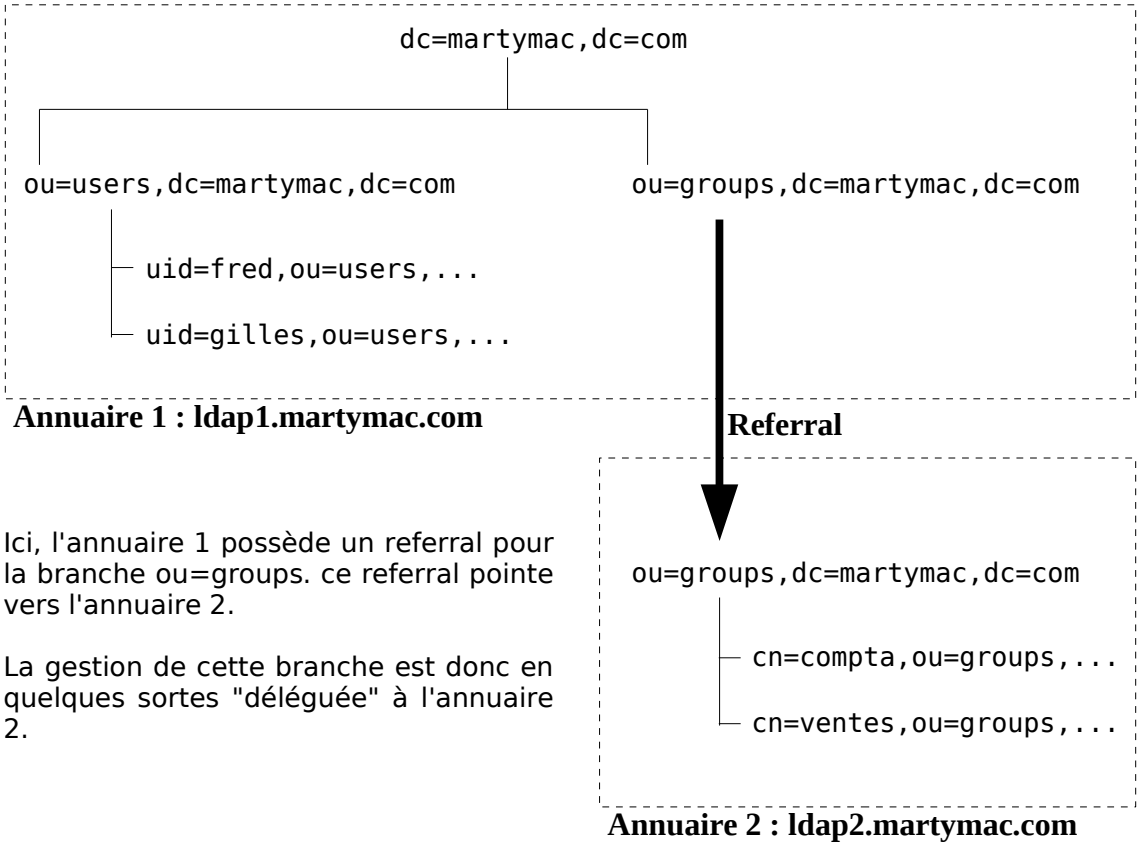
- le mode "maître-esclave", le plus courant : la réplication est unidirectionnelle, un annuaire maître envoie toutes les modifications à un annuaire esclave. Ceci n'autorise bien évidemment l'écriture que sur l'annuaire maître ; l'esclave est alors disponible uniquement en lecture.
- le mode "maître-maître" : la réplication est bidirectionnelle, chaque annuaire peut être maître de l'autre. Ceci permet d'écrire indifféremment sur l'un ou l'autre des annuaires.

Sachez enfin qu'il est possible de chaîner les réplications pour obtenir plusieurs réplicats.

La distribution (les referrals)

La distribution est un mécanisme qui va permettre de faire pointer un lien vers un autre annuaire pour une branche particulière. Ceci va permettre de déléguer la gestion de cette branche, un peu au sens DNS lorsqu'on délègue la gestion d'un domaine.

Ce mécanisme peut être représenté de la manière suivante, si l'on reprend l'exemple de notre domaine martymac.com :



Ici, l'annuaire 1 possède un referral pour la branche ou=groups. ce referral pointe vers l'annuaire 2.

La gestion de cette branche est donc en quelques sortes "déléguée" à l'annuaire 2.

Au niveau de l'annuaire 1, ceci se traduit par une entrée de la classe "referral", qui contient alors un attribut "ref" contenant l'adresse de la suite de l'arborescence :

```
dn: ou=groups,dc=martymac,dc=com
objectClass: referral
ref: ldap://ldap2.martymac.com/ou=groups,dc=martymac,dc=com
```

Note : Il existe également les "alias" qui sont des liens symboliques au sein du même annuaire. Cf. l'objectClass "alias".

OpenLDAP

Exercice : *Le chapitre suivant constitue un exercice à part entière et peut être effectué en même temps que la lecture du document, qui décrit toutes les étapes à suivre.*

Introduction et bref historique

Après avoir étudié les nombreux concepts liés aux annuaires LDAP, passons désormais à la pratique et étudions l'implémentation libre la plus utilisée : OpenLDAP.

OpenLDAP est un projet libre diffusé sous licence "OpenLDAP Public License" (**http://www.openldap.org/license.html**). Il est supporté par la fondation OpenLDAP, créée en 1998 par une société du nom de "Net Boolean", fournisseur de services professionnels liés à la messagerie.

La première version d'OpenLDAP (1.0) sort en août 1998. Il faudra attendre août 2000 pour que la version 2.0 ne voie le jour, offrant le support de LDAP v3. La version stable actuelle est la version 2.4, sortie dans le courant de l'année 2007.

Liens :

Site du projet : ***http://www.openldap.org***
Historique des versions : ***http://www.openldap.org/software/roadmap.html***
Historique du projet : ***http://www.openldap.org/conf/odd-sfo-2003/keynote.html***

Installation

Comme pour tout logiciel, il est possible d'installer OpenLDAP par le biais de paquets binaires fournis par votre distribution, ou bien en compilant les sources. L'installation par paquets est souvent conseillée car elle facilite la maintenance du logiciel par la suite.

Les outils clients sont souvent dissociés des outils serveur et fournis dans des paquets séparés. Nous allons donc installer deux paquets, puisque je vous propose d'installer les clients sur la même machine que la machine serveur. Ceci n'est pas forcément le cas dans un environnement de production où les clients agirons à distance depuis une autres machine.

Sur une distribution de type Debian (Debian, Ubuntu, ...), les paquets à installer sont les suivants : **slapd** et **ldap-utils**. Saisissez donc, en tant que root, la commande suivante :

```
# apt-get install slapd ldap-utils
```

Une fenêtre apparaît alors et vous demande le mot de passe associé à l'annuaire que vous mettez en place. Indiquez ce que vous souhaitez, nous le changerons par la suite.

Les outils fournis par OpenLDAP

Le projet OpenLDAP implémente un serveur LDAP, mais également les commandes clientes permettant de manipuler des informations contenues dans l'annuaire.

Les commandes liées au serveur

Le paquet slapd fournit les binaires suivants :

```
# dpkg -L slapd | grep bin
/usr/sbin
/usr/sbin/slapd
/usr/sbin/slurpd
/usr/sbin/slapadd
/usr/sbin/slapcat
/usr/sbin/slapdn
/usr/sbin/slapindex
/usr/sbin/slappasswd
/usr/sbin/slaptest
```

Chacune de ces commandes permet d'agir directement au niveau du serveur OpenLDAP, notamment au niveau de sa base de données. Il est donc impératif de les exécuter sur le serveur où fonctionne le serveur OpenLDAP.

Démons :
- **slapd** : le démon OpenLDAP !
- **slurpd** : le démon de réplication

Commandes de manipulation de la base (backend) gérée par OpenLDAP
- **slapindex** : crée les index au sein de la base
- **slapcat** : effectue un dump (une copie intégrale) de la base
- **slapadd** : ajoute des entrées LDIF dans la base
- **slappasswd** : utilitaire de conversion de mots de passe

Commandes de test/validation :
- **slaptest** : teste la validité du fichier de configuration slapd.conf
- **slapdn** : teste la conformité d'un DN donné en ligne de commande

Les commandes clientes

Le paquet ldap-utils fournit les commandes suivantes :

```
# dpkg -L ldap-utils | grep bin
/usr/bin
/usr/bin/ldapsearch
/usr/bin/ldapmodify
/usr/bin/ldapdelete
/usr/bin/ldapmodrdn
/usr/bin/ldappasswd
/usr/bin/ldapwhoami
/usr/bin/ldapcompare
/usr/bin/ldapadd
```

Chaque commande cliente utilise le protocole LDAP pour agir sur l'annuaire. Elles peuvent donc, cette fois-ci, être utilisées à distance. Elles agissent en tant que clients LDAP standard. Nous verrons qu'il existe d'autres clients, graphiques notamment.

- **ldapsearch** : effectue une recherche au sein de l'annuaire
- **ldapadd** : ajoute une entrée
- **ldapdelete** : supprime une entrée
- **ldapmodify** : modifie une entrée (ajoute/suppr. un attribut, ajoute/suppr. une entrée, ...)
- **ldapmodrdn** : modifie le rdn d'une entrée (renomme une entrée)
- **ldappasswd** : modifie le mot de passe d'une entrée LDAP
- **ldapwhoami** : affiche avec quel utilisateur le binding a eu lieu
- **ldapcompare** : permet de comparer l'attribut d'une entrée à une valeur spécifiée

Configuration du serveur

L'intégralité de la configuration du serveur OpenLDAP (le démon slapd) s'effectue en modifiant le fichier slapd.conf, situé dans le répertoire /etc/ldap.

Voici un exemple de configuration, ainsi que l'explication des directives :

```
#######################################################################
# Directives globales

# Inclusion des schemas
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema

# Ou sera stocke le PID du demon
pidfile         /var/run/slapd/slapd.pid

# Liste des arguments passes au demarrage du serveur
argsfile        /var/run/slapd.args

# Niveau de log
loglevel        0

# Emplacement des modules
# Chargement du module BDB (Berkeley DB)
modulepath /usr/lib/ldap
moduleload back_bdb

#######################################################################
# Declaration des options pour le premier type de backend utilise : bdb
# Toutes les options s'y appliquent jusqu'a la prochaine directive
# backend
#backend         bdb

#######################################################################
#backend   <autre>

#######################################################################
# Declaration des options de la premiere "base", c'est a dire de la
# premiere (et unique ici) arborescence geree par notre annuaire
# Toutes les options s'y appliquent jusqu'a la prochaine directive
# database
database        bdb
checkpoint      512 30
```

```
# La racine de notre arborescence
suffix          "dc=martymac,dc=com"

# Le compte administrateur de notre arborescence et son mot de passe
rootdn           "cn=Manager,dc=martymac,dc=com"
rootpw           "secret"

# Ou sont stockes les fichiers BDBs de notre arborescence
directory       "/var/lib/ldap"

# Options d'index
index           objectClass eq

# Sauvegarde de l'heure a laquelle est modifiee une entree
lastmod         on

# ACLs de notre premiere arborescence :
# Une personne non authentifiee peut s'authentifier
# Une personne authentifiee peut modifier son propre mot de passe
# Les autres n'ont pas accès à l'attribut mot de passe
access to attrs=userPassword
        by anonymous auth
        by self write
        by * none

# Tout le monde peut lire l'annuaire
access to *
        by * read

#######################################################################
# Autre arborescence
#database  <autre>
#suffix          "dc=debian,dc=org"
#[...]
```

Utilisez votre éditeur préféré, puis sauvegardez votre configuration. Testez-là ensuite pour voir si aucune erreur n'a été commise :

```
# slaptest -f /etc/ldap/slapd.conf
config file testing succeeded
```

Enfin, (re)-démarrez le serveur LDAP :

```
# /etc/init.d/slapd restart
```

Le fichier de configuration est subdivisé en trois sections importantes :

- la section globale (début du fichier)
- la section concernant les options de backends (débute par "backend")
- la section concernant les déclarations et les options des arborescences gérées (débute par "database")

Nous allons évoquer les directives les plus importantes ; n'hésitez pas à vous reporter au manuel du fichier de configuration pour plus d'informations :

```
# man 5 slapd.conf
```

L'inclusion des schémas

L'inclusion des schémas est effectuée par la directive "include". Comme nous l'avons étudié, les schémas LDAP permettent de définir les types de données contenus dans l'annuaire.

C'est grâce à ces inclusions au sein du fichier de configuration que l'on porte à la connaissance du serveur ces nouveaux types de données. Une fois les schémas chargés, il sera possible d'ajouter des entrées y faisant référence dans notre annuaire.

Plusieurs schémas sont fournis par défaut, je vous invite à regarder dans le répertoire /etc/ldap/schema pour les découvrir.

> *Note : la directive include inclut en fait un fichier de configuration (de manière générale). Il est donc possible de disposer de plusieurs fichiers de configuration spécifiques et de les regrouper de cette manière.*

Les niveaux de log

Il peut être important de savoir ce que fait exactement l'annuaire, ce, à des fins de débuggage par exemple. Pour ceci, nous avons à notre disposition la possibilité de changer le niveau de log dans le fichier slapd.conf.

Les niveaux de log disponibles sont les suivants (issus de "man slapd.conf") :

Valeur	Fonction correspondante
1	Appels de fonctions
2	Gestion des packets
4	Trace détaillée
8	Gestion des connexions
16	Affichage des packets envoyés et reçus
32	Gestion des filtres de recherche
64	Gestion du fichier de configuration
128	Gestion des ACLs
256	Affichage des connexions, opérations et résultats
512	Affichage des entrées retournées
1024	Affichage des communications avec les backends
2048	Parsing des entrées

Ces niveaux sont cumulables, c'est à dire qu'un niveau 48 équivaut aux niveaux 16 et 32. Un niveau 0 équivaut à un log désactivé.

Les logs sont gérés par syslog, ce qui signifie que vous pourrez consulter les informations logguées dans le fichier /var/log/syslog.

Les backends

Différents backends sont disponibles. Les plus couramment utilisés sont BDB (Berkeley DB, cf. : **http://www.sleepycat.com**) et LDBM. Ces deux backends sont des bases de données stockées dans des fichiers.

BDB est recommandé car réputé plus robuste que LDBM.

N'hésitez pas à consulter les pages de man de slapd-bdb et slapd-ldbm pour plus d'informations.

D'autres backends existent et permettent par exemple de stocker les informations dans de véritables SGBD, mais nous n'allons pas présenter ces fonctionnalités ici.

Les databases

Une section de database représente la déclaration d'une arborescence. Ceci implique plusieurs paramètres, dont une racine, un compte administrateur, ...

Voici les paramètres importants dans cette section :

La racine

Elle est spécifiée par la directive "suffix". La racine correspond souvent au **FQDN** (Fully Qualified Domain Name) de la machine associé aux attributs "dc".

```
suffix          "dc=martymac,dc=com"
```

L'accès administrateur

Il est possible de déclarer un compte qui ne sera sujet à aucune limitation. Il s'agit en quelques sortes d'un compte "root". Ce compte peut correspondre ou non à une entrée dans l'annuaire. Il sera purement virtuel si aucun DN n'est effectivement stocké dans l'annuaire.

Ce compte particulier n'est pas soumis aux restrictions imposées par les ACLs (voir ci-dessous). Il est déclaré par la directive "rootdn". Son mot de passe est spécifié par la directive "rootpw".

> *Note : attention, ne confondez pas "rootdn" et DN racine, qui correspond à la base de notre annuaire ! Ici le "rootdn" est bien le dn d'un utilisateur ayant les droits root...*

Le mot de passe du rootdn peut être soit en clair, comme dans notre exemple, soit un hash généré par la commande slappasswd. Nous pouvons générer ce hash de cette manière :

```
# slappasswd -s "secret" -h {CRYPT}
{CRYPT}hKCmXZ7Pqmm4Y
```

La valeur affichée est alors à copier-coller dans la valeur de la directive "rootpw" :

```
rootpw           "{CRYPT}hKCmXZ7Pqmm4Y"
```

Ceci permet de ne pas stocker en clair le mot de passe dans le fichier de configuration !

Les index

Les index sont un moyen d'accélérer les recherches au sein de l'annuaire. Dans le fichier de configuration, il convient de préciser quels attributs seront le plus fréquemment utilisés pour les recherches et doivent donc être indexés. Ceci se fait par la directive "index" :

```
index           objectClass eq
```

Ici, nous activons la gestion des index sur les objectClass, ce qui semble un minimum !

Chaque index est destiné à faciliter un type de recherche. Le type d'index à créer est ici "eq", ce qui signifie qu'il sera efficace pour une recherche faisant intervenir une égalité stricte de chaînes. Voici une liste des types d'index disponibles et leur type de recherche associé :

Index	**Type de recherche (filtre), exemple**
eq	'uid=martymac', égalité stricte, pas d'utilisation de "wildcard" * (cf. sub)
sub	'uid=marty*', utilisation d'un wildcard
subinitial	optimisation de sub pour 'uid=marty*', wildcard à la fin
subfinal	optimisation de sub pour 'uid=*mac', wildcard au début
subany	optimisation de sub pour 'uid=*rtym*', wilcard au début ou à la fin
approx	'uid~=martymac', recherche par approximation sonore
pres	'objectclass=posixAccount', recherche de présence

Le type de recherche effectué est déduit du filtre passé au client qui effectue cette recherche.

Un ou plusieurs types de recherches peuvent être spécifiés pour un ou plusieurs attributs à la fois. Dans ce cas, la virgule sépare les différents éléments. Exemple :

```
index           uid,gecos,description eq,subinitial
index           uidNumber,gidNumber eq
```

Les index doivent être générés par l'administrateur pour être fonctionnels (commande "slapindex"), nous aborderons ce point par la suite.

Les listes d'accès (ACLs)

Les ALCs permettent de définir finement les droits d'accès à l'annuaire. La syntaxe générale des ACLs est la suivante :

```
access to <quoi> [ by <qui> <acces> [ <contrôle> ] ]+
```

Nous avons créé dans notre exemple deux ACLs :

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

access to *
        by * read
```

La première concerne l'attribut userPassword :

- on autorise l'accès aux personnes non authentifiées uniquement pour une authentification (by anonymous auth)
- on autorise une personne authentifiée à modifier son propre mot de passe (by self write)
- enfin, on refuse l'accès à cet attribut aux autres personnes

La seconde ACL concerne toutes les informations contenues dans l'annuaire (*) :

- on autorise tout le monde à les lire

Note : *les ACls sont évaluées dans leur ordre d'apparition dans le fichier de configuration. OpenLDAP arrête leur évaluation lorsqu'il a trouvé une ACL faisant intervenir la cible recherchée. Les directives les plus générales doivent donc être situées* ***après*** *les directives s'appliquant à une cible particulière. C'est le cas ici avec l'ACL ciblant l'attribut userPassword, située avant celle ciblant toute information (*).*

Nous n'allons pas étudier ici plus en détail les ACLs, je vous invite à consulter la page de man de "slapd.access" pour plus de détails.

Administration du serveur

Introduction

Notre serveur est désromais configuré. Nous allons maintenant voir comment nous pouvons l'administrer.

Attention !!! *Les commandes que nous utilisons ici n'utilisent pas le protocole LDAP mais accèdent directement à la base de données sous-jacente (BDB dans notre cas). Il est donc* ***impératif*** *de toujours couper le serveur LDAP avant d'utiliser une commande slap(...), afin d'éviter un accès concurrent depuis le serveur lui-même, ce qui pourrait corrompre la base de données.*

Coupez donc le serveur LDAP avant de continuer :

```
# /etc/init.d/slapd stop
```

Slapindex

Nous avons configuré notre serveur pour qu'il utilise des index ; la première chose à effectuer avant d'utiliser notre serveur est donc de les générer. Il faut en effet initialiser les index pour qu'OpenLDAP puisse ensuite les utiliser et les maintenir.

L'opération de génération n'est à effectuer qu'une seule fois et ceci se fait par le biais de la commande slapindex.

```
# slapindex
```

Slapcat

Slapcat est une commande très utile au quotidien. Elle effectue un "dump" de la base LDAP au format LDIF. Il est conseillé de l'utiliser régulièrement pour effectuer des sauvegardes de notre annuaire.

Par défaut, slapcat affiche les informations sur la sortie standard, il faut donc la rediriger vers un fichier pour obtenir notre sauvegarde :

```
# slapcat > sauvegarde.ldif
```

Slapadd

Slapadd est l'inverse de slapcat. Cette commande permet de peupler notre annuaire en utilisant un fichier LDIF. Elle est typiquement utilisée pour restaurer une sauvegarde effectuée avec slapcat :

```
# slapadd < sauvegarde.ldif
```

Arrêt et démarrage du serveur

L'arrêt et le démarrage du serveur LDAP se font par le biais du script /etc/init.d/slapd :

```
/etc/init.d/slapd [start|stop|restart]
```

L'administration du serveur étant terminée, je vous propose de le démarrer et de découvrir les commandes clientes fournies par OpenLDAP :

```
# /etc/init.d/slapd start
```

Utilisation des outils clients

Introduction

Nous avons étudié les outils d'administration du serveur : les outils slap(...), nous allons maintenant étudier les outils clients. A la différence des outils slap(...), les outils ldap(...) utilisent le protocole LDAP, il peuvent donc être mis en oeuvre depuis n'importe quelle machine disposant d'un accès réseau au serveur LDAP. Ils utilisent bien évidemment le format LDIF pour échanger des informations avec le serveur.

Nous allons étudier les différentes possibilités offertes par ces outils à travers des exemples simples.

Ajouter une entrée : ldapadd

Pour ajouter une entrée dans l'annuaire, il faut utiliser la commande ldapadd. Sa syntaxe est la suivante :

```
ldapadd -W -D <binddn> -x -H ldap://<serveur> -f <fichier.ldif>
```

L'option "-W" active la demande de mot de passe pour s'authentifier en tant que <binddn>. L'option "-x" permet de ne pas utiliser SASL pour l'authentification. Enfin, le fichier LDIF source doit contenir une (ou plusieurs) entrée(s) à insérer et l'intégralité de ses (leurs) attributs.

Exemple :

Fichier LDIF à insérer (fichier.ldif) :

```
dn: uid=dupont,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: dupont
uid: dupont
uidNumber: 10001
gidNumber: 1024
homeDirectory: /home/dupont
userPassword:: e0NSWVBUfXZKblR0TjVSaXQ0Tmc=
loginShell: /bin/sh
gecos: dupont
description: dupont
```

Insertion de l'entrée :

```
# ldapadd -W -D "cn=Manager,dc=martymac,dc=com" -x -H ldap://localhost -f
fichier.ldif
Enter LDAP Password:
adding new entry "uid=dupont,ou=users,dc=martymac,dc=com"
```

Nous insérons cette entrée depuis le serveur lui-même (localhost) et nous utilisons ici le compte "Manager" déclaré dans le fichier de configuration. Notez que cette commande ne fonctionnera pas si vous n'avez pas initialisé l'annuaire avec les entrées de base que nous avons vues auparavant : ou=users et dc=martymac,dc=com. Voyons comment nous pouvons créer ces entrées basiques.

Initialiser l'annuaire

L'initialisation de l'annuaire n'est qu'un ajout massif de plusieurs entrées. Cet ajout massif peut se faire par le biais de slapadd si vous possédez déjà un dump de l'annuaire et si vous vous situez sur le serveur.

A distance, c'est l'outil ldapadd qui va nous permettre d'effectuer cette opération. Il suffit de fournir à ldapadd un fichier LDIF contenant plusieurs entrées qui seront ajoutées dans le même ordre avec lequel elles apparaissent dans le fichier.

Ce fichier va donc tout d'abord contenir l'entrée de la racine, qui est nécessaire, puis chacune des "ou" que nous avons vues en exemple. Enfin, les feuilles seront constituées d'utilisateurs et de groupes.

Fichier LDIF à insérer (fichier.ldif) :

```
dn: dc=martymac,dc=com
objectClass: dcObject
objectClass: organization
dc: martymac
o: martymac
description: martymac

dn: ou=users,dc=martymac,dc=com
objectClass: top
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=martymac,dc=com
objectClass: top
objectClass: organizationalUnit
ou: groups

dn: cn=utilisateurs,ou=groups,dc=martymac,dc=com
objectClass: posixGroup
cn: utilisateurs
gidNumber: 2000

dn: uid=garfield,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
userPassword:: e0NSWVBUfWRhSDJadHI4dElnZFE=
```

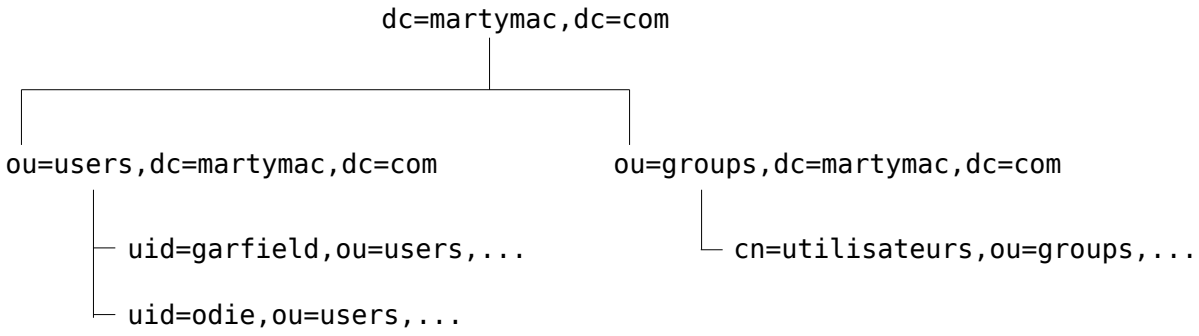
```
loginShell: /bin/sh
gecos: garfield
description: garfield

dn: uid=odie,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: odie
uid: odie
uidNumber: 10002
gidNumber: 2000
homeDirectory: /home/odie
userPassword:: e0NSWVBUfTQzZXltaHBSUzBqQVk=
loginShell: /bin/sh
gecos: odie
description: odie
```

Le fichier ci-dessus comprend :

- l'entrée de la racine (indispensable)
- l'entrée des deux ou : users et groups
- un groupe : utilisateurs
- deux utilisateurs : garfield et odie appartenant au groupe utilisateurs (attr. gidNumber)

Voici un schéma représentant cette arborescence :



Insertion de cette arborescence :

```
# ldapadd -W -D "cn=Manager,dc=martymac,dc=com" -x -H ldap://localhost -f
fichier.ldif
```

Rechercher une entrée : ldapsearch

La commande ldapsearch permet d'effectuer une recherche au sein de l'annuaire. Voici sa syntaxe :

```
ldapsearch -x -H ldap://<serveur> -b <base> [-s portée] [filtre] [attributs]
```

Elle reprend bien évidemment les concepts que nous avons abordés jusqu'ici :

- la base de la recherche
- la portée de la recherche (base, one ou sub) - sub est la portée par défaut
- le filtre
- le ou les attributs que l'on souhaite afficher - l'entrée entière est affichée par défaut

Il est possible de s'authentifier, si nécessaire, avec l'option -D (et l'option -W), mais notre annuaire est ici configuré pour permettre l'accès en lecture à tout le monde.

Exemples :

On recherche tous les uid commençant par "garf" à partir de la racine de l'annuaire :

```
ldapsearch -x -H ldap://localhost -b "dc=martymac,dc=com" "(uid=garf*)"
```

On recherche toutes les entrées ayant un gidNumber égal à 2000 :

```
ldapsearch -x -H ldap://localhost -b dc=martymac,dc=com "(gidNumber=2000)"
```

Cette commande nous retourne les 2 utilisateurs, mais aussi le groupe car il possède lui aussi l'attribut gidNumber. Améliorons notre requête pour ne retourner que les deux comptes utilisateurs :

```
ldapsearch -x -H ldap://localhost -b dc=martymac,dc=com "(&(gidNumber=2000)
(objectClass=posixAccount))"
```

On affiche enfin uniquement leur répertoire home (et pas la totalité de l'entrée comme c'est le cas par défaut) :

```
ldapsearch -x -H ldap://localhost -b dc=martymac,dc=com "(&(gidNumber=2000)
(objectClass=posixAccount))" homeDirectory
```

Le résultat de cette dernière requête est le suivant :

```
# [...]
# garfield, users, martymac.com
dn: uid=garfield,ou=users,dc=martymac,dc=com
homeDirectory: /home/garfield

# odie, users, martymac.com
dn: uid=odie,ou=users,dc=martymac,dc=com
homeDirectory: /home/odie
# [...]
```

Supprimer une entrée : ldapdelete

La suppression d'une entrée se fait par la commande ldapdelete. Voici sa syntaxe :

```
ldapdelete -W -D <binddn> -x -H ldap://<serveur> <dn>
```

Puisqu'un effacement correspond à une écriture, il faudra, la plupart du temps, s'authentifier (à la différence de ldapsearch).

Il est possible d'effacer récursivement une branche complète en utilisant l'option "-r" sur le noeud de la branche. Attention, commande potentiellement dangereuse !

Note : Il est possible de réinitialiser un annuaire par la méthode dite "sauvage et brutale, mais simple et rapide" ! En effet, il est possible de simplement supprimer les fichiers de la

base de données de l'annuaire et de le redémarrer. Ces fichiers sont souvent situés dans le répertoire /var/lib/ldap (cf. directive "directory" du fichier de configuration). Attention si vous possédez plusieurs bases à ne pas toutes les effacer...

Exemples :

Suppression de l'utilisateur odie :

```
ldapdelete -x -H ldap://localhost -W -D "cn=Manager,dc=martymac,dc=com"
"uid=odie,ou=users,dc=martymac,dc=com"
```

Suppression de la branche users :

```
ldapdelete -x -H ldap://localhost -W -D "cn=Manager,dc=martymac,dc=com" -r
"ou=users,dc=martymac,dc=com"
```

Je vous laisse re-créer cette branche ainsi que ses deux utilisateurs pour pouvoir poursuivre...

Modifier une entrée : ldapmodify

La commande ldapmodify est un peu le "couteau suisse" des annuaires LDAP ! Elle va permettre d'effectuer toutes sortes d'opérations, y compris l'ajout et la suppression d'entrées. Sa syntaxe est la suivante :

```
ldapmodify -W -D <binddn> -x -H ldap://<serveur> -f <fichier.ldif>
```

ldapmodify peut se substituer à ldapadd et ldapdelete, mais vous allez voir que son utilisation n'est pas des plus simples ! En effet, tout passe par le fichier ldif indiqué en entrée et qui va décrire l'opération à effectuer...

Voici une liste (non exhaustive) d'opérations possibles :

- ajouter d'une entrée
- supprimer d'une entrée
- ajouter un attribut
- supprimer un attribut
- modifier un attribut

Note : N'hésitez pas à consulter les pages de man de "slapd.replog" et "ldif" pour une liste exhaustive des opérations que l'on peut effectuer. La syntaxe est la même syntaxe que celle employée dans les mécanismes de réplications que nous avons évoqués plus haut.

Ajouter une entrée :

Ajoutons un utilisateur "john".

Fichier LDIF (fichier.ldif) :

```
dn: uid=john,ou=users,dc=martymac,dc=com
changetype: add
objectClass: account
objectClass: posixAccount
cn: john
uid: john
uidNumber: 10003
```

```
gidNumber: 2000
homeDirectory: /home/john
userPassword:: e0NSWVBUfTg0QmNhL1BhL2tIUC4=
loginShell: /bin/sh
gecos: john
description: john
```

On remarque la présence d'un attribut "changetype" en plus de chacun des attributs de l'entrée que nous souhaitons ajouter.

Application de la modification :

```
ldapmodify -W -D "cn=Manager,dc=martymac,dc=com" -x -H ldap://localhost -f
fichier.ldif
```

Supprimer une entrée :

Le principe est le même que pour l'ajout d'une entrée. Cette fois, la valeur de "changetype" n'est plus "add" mais "delete".

Fichier LDIF (fichier.ldif) :

```
dn: uid=john,ou=users,dc=martymac,dc=com
changetype: delete
```

Application de la modification : cf. point précédent, il s'agit de la même commande !

Ajouter un attribut :

L'ajout d'un attribut s'effectue par le changetype "modify" et par un nouvel attribut "add" qui précise quel attribut ajouter. Ici, nous allons ajouter une seconde description pour l'utilisateur "garfield". L'attribut "description" devient donc multi-valué.

```
dn: uid=garfield,ou=users,dc=martymac,dc=com
changetype: modify
add: description
description: gros chat paresseux
```

Après la modification, l'utilisateur "garfield" possède les attributs suivants :

```
ldapsearch -x -H ldap://localhost -b "ou=users,dc=martymac,dc=com"
"(uid=garfield)"

#[...]
# garfield, users, martymac.com
dn: uid=garfield,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
loginShell: /bin/sh
gecos: garfield
description: garfield
description: gros chat paresseux
```

```
#[...]
```

Supprimer un attribut :

La suppression d'attribut s'effectue via le changetype "modify" et l'utilisation d'un nouvel attribut : "delete".

Supprimons la description, fort négative pour Garfield, que nous venons d'ajouter :

```
dn: uid=garfield,ou=users,dc=martymac,dc=com
changetype: modify
delete: description
description: gros chat paresseux
```

Notez qu'il est possible de ne pas spécifier la valeur de la description à supprimer. Dans ce cas, toutes les descriptions seront supprimées.

Modifier un attribut :

Pour modifier un attribut, le "changetype" à employer est, ici encore, "modify". L'attribut supplémentaire à ajouter est l'attribut "replace" qui va préciser quel attribut remplacer. Enfin, nous spécifions la nouvelle valeur de l'attribut.

```
dn: uid=garfield,ou=users,dc=martymac,dc=com
changetype: modify
replace: description
description: ami fidele de john
```

Remarquez que la modification ci-dessus remplace toutes les descriptions par celle qui a été spécifiée. Il n'est pas possible de modifier uniquement l'une des valeurs d'un attribut multi-valué.

Il faudra ruser pour effectuer cette dernière opération et le faire en deux temps : d'abord supprimer l'attribut désiré, ensuite ajouter le nouvel attribut. Imaginons que nous soyons dans le cas où garfield ait deux attributs :

```
ldapsearch -x -H ldap://localhost -b "ou=users,dc=martymac,dc=com"
"(uid=garfield)"

#[...]
# garfield, users, martymac.com
dn: uid=garfield,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
loginShell: /bin/sh
gecos: garfield
description: ami fidele de john
description: chat gourmand
#[...]
```

Si nous souhaitons remplacer uniquement "chat gourmand" par "chat paresseux", nous pouvons utiliser le fichier ci-dessous :

```
dn: uid=garfield,ou=users,dc=martymac,dc=com
changetype: modify
delete: description
description: chat gourmand

dn: uid=garfield,ou=users,dc=martymac,dc=com
changetype: modify
add: description
description: chat paresseux
```

Ce qui peut s'écrire de manière plus concise par l'utilisation du tiret "-" qui permet de chaîner les actions pour un même DN :

```
dn: uid=garfield,ou=users,dc=martymac,dc=com
changetype: modify
delete: description
description: chat gourmand
-
add: description
description: chat paresseux
```

Renommer une entrée : ldapmodrdn

L'outil ldapmodrdn permet de modifier le RDN (uniquement) d'une entrée. Il s'utilise de cette manière :

```
ldapadd -W -D <binddn> -x -H ldap://<serveur> <dn> <nouveau_rdn>
```

Exemple :

Pour renommer "garfield" en "pookie", nous pourrions saisir cette commande :

```
ldapmodrdn -W -D "cn=Manager,dc=martymac,dc=com" -x -H ldap://localhost
"uid=garfield,ou=users,dc=martymac,dc=com" "uid=pookie"
```

L'attribut "uid: pookie" sera ajouté automatiquement à l'entrée car il compose le nouveau RDN. L'ancienne valeur de l'uid ("uid: garfield") sera conservée.

Configuration des outils clients

Voici une information que j'ai gardée secrète jusqu'ici et qui vous simplifiera la tâche par la suite : il existe un fichier de configuration pour les outils clients ! Ce fichier contient les options que les commandes clientes doivent utiliser par défaut : l'adresse du serveur cible, le binddn, etc... Jusqu'ici, ces options étaient passées à chaque fois en lignes de commandes ; renseigner ce fichier de configuration évitera cette tâche répétitive et fastidieuse.

Il existe deux fichiers de configuration : **/etc/ldap/ldap.conf** et **~/.ldaprc**. Le premier est disponible pour tous les utilisateurs ; le second peut être défini par l'utilisateur et permet de surcharger les options spécifiées par le premier.

Voici un exemple de fichier /etc/ldap/ldap.conf :

```
# Configuration des outils clients (voir "man ldap.conf")
```

```
# Racine
BASE dc=martymac,dc=com
# Nom et port de l'annuaire
URI  ldap://localhost:389
```

Ainsi, une interrogation de l'annuaire devient simplement :

```
# ldapsearch -x
```

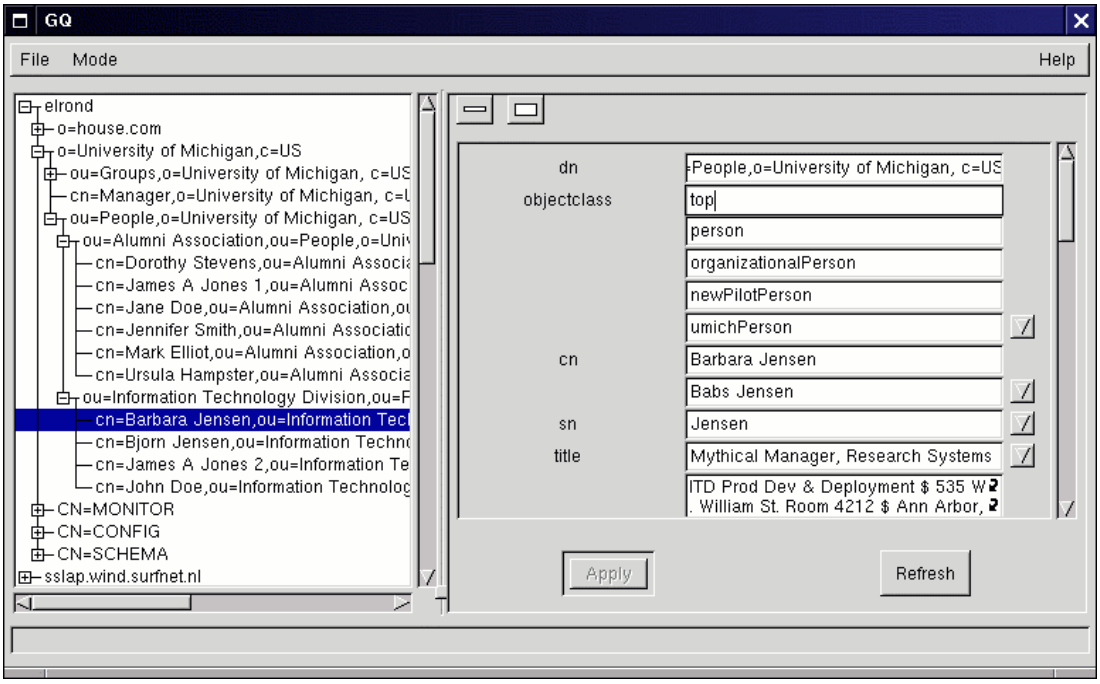
Les outils graphiques d'administration

Nous avons étudié comment intéragir avec notre annuaire par le biais des lignes de commandes. Rassuez-vous, il existe des outils graphiques pour nous simplifier la tâche au quotidien !

Trois sont particulièrement intéressants, il s'agit de gq, ldapbrowser et phpldapadmin.

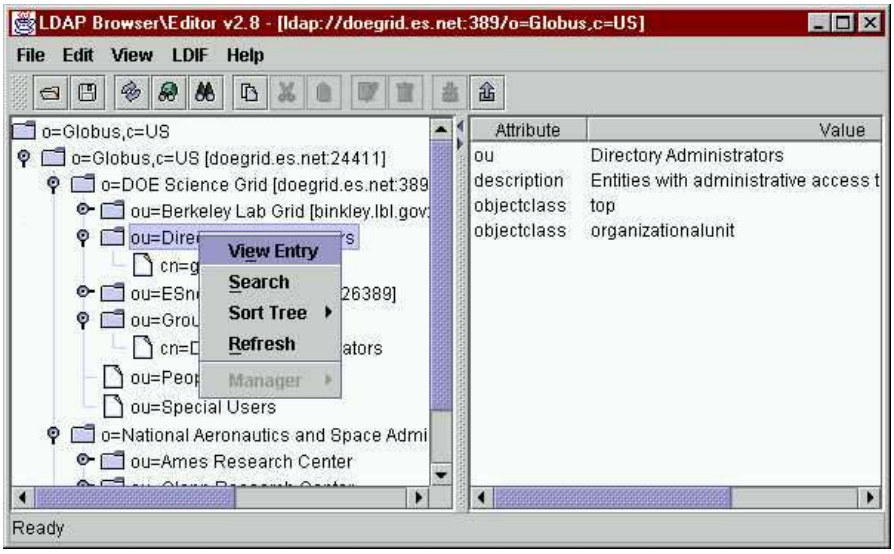
Gq

- Développé en C / GTK
- URL : **http://sourceforge.net/projects/gqclient**



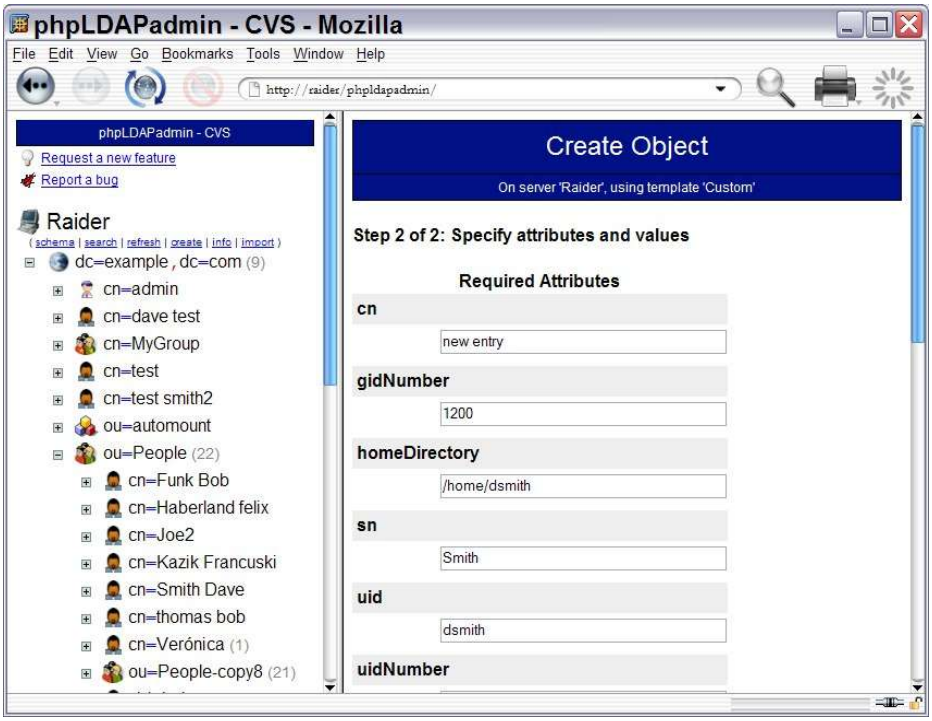
Ldapbrowser

- Développé en JAVA
- URL : **http://www-unix.mcs.anl.gov/~gawor/ldap/index.html**



PhpLDAPAdmin

- Développé en PHP
- URL : **http://phpldapadmin.sourceforge.net**



Connexion de Samba à notre annuaire

Exercice : Le chapitre suivant constitue un exercice à part entière et peut être effectué en même temps que la lecture du document, qui décrit toutes les étapes à suivre.

Introduction

Nous allons, dans ce dernier chapitre, présenter un exemple concret d'utilisation d'un annuaire OpenLDAP en y connectant Samba.

Vous savez que Samba nécessite deux types de comptes : un compte POSIX, mais aussi un compte Samba, qui complètera les informations du compte POSIX. Nous allons ici centraliser ces deux comptes sur le même annuaire, ce qui facilitera l'administration de notre serveur.

La configuration de Samba va rester très simple : nous allons mettre en place un serveur de fichiers autonome.

Pré-requis

Il est indispensable de bien maîtriser Samba et les notions qui s'y rapportent avant de continuer la lecture de ce chapitre. Je vous donc conseille de bien relire au préalable le cours sur Samba !

Préparation de l'annuaire

Samba gère 3 types de comptes : les comptes d'utilisateurs, de groupes et de machines. Je vous propose donc de créer 3 "ou" qui permettront de les stocker.

Notez que l'"ou" machines ne nous servira pas ici car Samba ne l'utilise que lorsqu'il est contrôleur de domaine. L'annuaire que nous mettons en place comporte quand même cette "ou", qui vous permettra de gérer le contrôle de domaine par la suite si vous le désirez.

Effaçons tout d'abord les anciennes données de notre annuaire :

```
# /etc/init.d/slapd stop
# rm -f /var/lib/ldap/*
# /etc/init.d/slapd start
```

Ajoutons ensuite l'arborescence de base que nous avons présentée (fichier base.ldif) :

```
dn: dc=martymac,dc=com
objectClass: dcObject
objectClass: organization
dc: martymac
o: martymac
description: martymac

dn: ou=users,dc=martymac,dc=com
objectClass: top
objectClass: organizationalUnit
ou: users
```

```
dn: ou=groups,dc=martymac,dc=com
objectClass: top
objectClass: organizationalUnit
ou: groups

dn: ou=machines,dc=martymac,dc=com
objectClass: top
objectClass: organizationalUnit
ou: machines
```

```
# ldapadd -W -D "cn=Manager,dc=martymac,dc=com" -x -H ldap://localhost -f base.ldif
```

Les comptes POSIX - Nsswitch

Notre annuaire est prêt à accueillir les différents comptes que nous allons mettre oeuvre avec Samba.

Comme je le rappelais en introduction, vous savez que Samba a besoin d'un compte POSIX pour pouvoir créer un compte Samba associé.

La première étape est donc de configurer notre serveur GNU/Linux pour aller chercher les comptes POSIX sur l'annuaire. Nous allons pour ceci utiliser le mécanisme "nsswitch".

Nsswitch (Name Service Switch) permet de rediriger les requêtes de noms vers des sources très diverses. Une requête de nom est le fait d'obtenir des informations concernant un nom particulier (résoudre un login en uid, un nom de machine en adresse IP, connaître le répertoire home d'un utilisateur, etc...).

La source de ces données est habituellement un fichier (/etc/passwd, /etc/shadow, /etc/hosts) mais elle peut être une base de données, un annuaire, etc... car Nsswitch propose un mécanisme de plugins qui permet d'étendre ses capacités d'interconnexion.

Installation de libnss-ldap

Nous allons commencer par installer le plugin (la librairie) qui permet à nsswitch d'utiliser un annuaire LDAP comme source de données. Cette librairie se nomme libnss-ldap et est développée par PADL (**http://www.padl.com**).

Nous allons l'installer très simplement en utilisant les paquets fournis par notre distribution :

```
# apt-get install libnss-ldap
```

Un assistant de configuration vous pose quelques questions. Validez rapidement, nous allons revoir la configuration manuellement.

Configuration de libnss-ldap

La librairie est installée, il faut maintenant la configurer pour lui indiquer notamment sur quel serveur se trouvent les comptes.

La configuration de la librairie se trouve dans le fichier /etc/libnss-ldap.conf :

```
# Emplacement du serveur LDAP a utiliser
```

```
host 127.0.0.1
port 389

# Racine de notre annuaire
base dc=martymac,dc=com

# Version du protocole a utiliser
ldap_version 3

# Portee de la recherche par defaut
scope sub

# Emplacement des comptes
nss_base_passwd  dc=martymac,dc=com?sub
nss_base_group   ou=groups,dc=martymac,dc=com?one
```

La partie importante du fichier concerne l'emplacement des comptes : il faut spécifier où trouver les informations fournies habituellement par /etc/passwd (les comptes utilisateurs) et celles fournies par /etc/group (les comptes de groupes).

Pour les comptes utilisateurs, nous utilisons une astuce : pour Samba, nsswitch doit connaître à la fois les comptes utilisateurs et les comptes machines. Puisque nous ne pouvons spécifier qu'une base de recherche et une portée, nous indiquons une recherche à partir du niveau supérieur (ici, la racine de l'annuaire) avec une portée sub. Cette méthode permettra à nsswitch de descendre à la fois dans l'"ou" "users", mais aussi dans l'"ou" "machines".

Les groupes ne posent pas de problème car ils sont tous stockés dans la même "ou". La portée choisie ici est "one".

Remarquez que dans notre exemple, aucun compte n'est utilisé pour nous connecter à l'annuaire. Ceci est possible car nous n'effectuons que des accès en lecture et notre annuaire permet à tout le monde de lire. Il est possible, si nécessaire, d'indiquer à la librairie un compte à utiliser avec les directives binddn (lecture), bindpw et rootbinddn (écriture). Je vous invite à consulter la page de "man libnss-ldap.conf" pour plus d'informations...

Note : ne soyez pas surpris en regardant le fichier fourni en exemple : il contient une quantité importante d'autres directives utilisées par une autre librairie : ***libpam-ldap****. Nous n'avons pas besoin de les spécifier ici.*

Utilisation de libnss-ldap dans nsswitch

Une fois notre librairie configurée, il faut activer la recherche dans LDAP au niveau de nsswitch. Ceci se fait très simplement en modifiant le fichier /etc/nsswitch.conf :

```
passwd:         compat ldap
group:          compat ldap
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Il suffit d'ajouter "ldap" à la fin des entrées passwd et group. Ceci signifie que pour chaque type de résolution, nsswitch utilisera tout d'abord le mode compat (recherche dans les fichiers appropriés) en priorité, puis le mode LDAP.

> *Note : Il est inutile de modifier shadow car nous travaillons avec des posixAccount et non des shadowAccount.*

Ajout d'un compte

Testons notre configuration en ajoutant deux comptes à l'annuaire : un groupe "utilisateurs" et un utilisateur "garfield" appartenant à ce groupe (fichier utilisateur.ldif) :

```
dn: cn=utilisateurs,ou=groups,dc=martymac,dc=com
objectClass: posixGroup
cn: utilisateurs
gidNumber: 2000

dn: uid=garfield,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
userPassword:: e0NSWVBUfUNBSzA2dFkzZG03Z0U=
loginShell: /bin/sh
gecos: garfield
description: garfield
```

```
# ldapadd -W -D "cn=Manager,dc=martymac,dc=com" -x -H ldap://localhost -f
utilisateur.ldif
```

Le mot de passe indiqué ici a peu d'importance car c'est celui configuré au niveau du compte Samba qui sera utilisé. Si toutefois vous souhaitez modifier le mot de passe de l'utilisateur ajouté, je vous invite à utiliser la commande "ldappasswd" (man ldappasswd).

> *Note : les "ldapscripts" permettent d'ajouter rapidement et simplement un compte POSIX à un annuaire LDAP en s'affranchissant des commandes présentées ci-dessus. Vous pouvez les trouver à cette adresse : **http://contribs.martymac.com**.*

Test de la reconnaissance du compte

Voyons si nos nouveaux comptes sont bien reconnus et pris en compte par le système. Les commandes "getent" et "id" vont nous donner la réponse :

```
# getent group
[...]
utilisateurs:x:2000:

# getent passwd
[...]
garfield:x:10001:2000:garfield:/home/garfield:/bin/sh

# id garfield
```

```
uid=10001(garfield) gid=2000(utilisateurs) groups=2000(utilisateurs)
```

Notre groupe et notre utilisateur apparaissent parmi la liste de ceux connus ! De plus, l'utilisateur appartient bien au groupe "utilisateurs" !

Connexion sur le système Unix avec le compte

Certains se demandent certainement s'il est possible de se connecter sur le système avec le compte "garfield". La réponse est négative. Pour que ceci soit possible, il faudrait configurer PAM (Pluggable Authentication Modules, le système modulaire d'authentification utilisé par GNU/Linux) pour qu'il utilise lui-aussi LDAP.

Nous avons travaillé avec nsswitch sur la "reconnaissance" des comptes par le système. PAM intervient en aval et permet l'authentification. Il faut donc configurer une librairie supplémentaire : libpam-ldap, disponible elle aussi sur **http://www.padl.com**. Ceci est une autre histoire car Samba ne nécessite pas cette configuration...

Connexion de Samba à l'annuaire

Le système est désormais "connecté" à l'annuaire... Il reste à configurer OpenLDAP pour qu'il soit capable de gérer les comptes Samba et Samba pour qu'il aille chercher ses comptes sur l'annuaire.

Copie du schema Samba

La première étape est de copier le schéma de Samba (fourni par le paquet samba-doc) dans le répertoire schema d'OpenLDAP et de l'inclure dans la configuration. De cette manière, OpenLDAP pourra gérer les comptes Samba :

```
# cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz /etc/ldap/schema
# cd /etc/ldap/schema ; gunzip samba.schema.gz
```

Ajoutons ensuite la ligne suivante à la fin des "include" dans le fichier /etc/ldap/slapd.conf :

```
include /etc/ldap/schema/samba.schema
```

Enfin, redémarrons le serveur LDAP :

```
# /etc/init.d/slapd restart
```

Configuration de Samba

La dernière étape est de modifier la configuration de Samba (/etc/samba/smb.conf) pour lui indiquer où stocker ses propres comptes. Il faut pour ceci modifier la directive "passdb backend". Je vous propose le fichier de configuration minimaliste suivant, incluant un partage simple de données :

```
[global]

# Identification Netbios
workgroup = Workgroup
netbios name = ubuntu
```

```
# Controle de domaine desactive
os level = 40
domain logons = no
domain master = no
local master = no

# Base de donnee de comptes
passdb backend = ldapsam:ldap://localhost
ldap admin dn = "cn=Manager,dc=martymac,dc=com"
ldap ssl = off
ldap delete dn = no
ldap user suffix = ou=users
ldap machine suffix = ou=machines
ldap group suffix = ou=groups
ldap suffix = dc=martymac,dc=com

# Authentification via la base de comptes locale
security = user

# Securite
encrypt passwords = yes

# Gestion des logs
log file = /var/log/samba/%m.log
log level = 2

# Partage accessible uniquement au groupe sambausers
[donnees]
path = /data/samba/donnees
comment = Partage Donnees
writeable = yes
browsable = yes
guest ok = no
valid users = @utilisateurs
```

Ce fichier indique que le "passdb backend" à utiliser est notre annuaire. Nous précisons également divers éléments tels que l'emplacement des comptes utilisateurs, groupes et machines.

Samba va avoir besoin d'écrire dans notre annuaire : nous spécifions donc quel est le compte administrateur LDAP à utiliser ("ldap admin dn"). Cependant, aucun mot de passe n'est précisé dans le fichier de configuration. Pour des raisons de sécurité, on va indiquer le mot de passe en lignes de commandes de cette manière :

```
# smbpasswd -w secret
```

Samba va stocker ce mot de passe dans un autre fichier : secrets.tdb.

Enfin, on prendra soin de créer le répertoire /data/samba/donnees et de donner les bons droits au groups "utilisateurs"...

```
# mkdir -p /data/samba/donnees
# chgrp utilisateurs /data/samba/donnees
# chmod 775 /data/samba/donnees
```

... Et de redémarrer Samba :

```
# /etc/init.d/samba restart
```

Ajout du compte Samba

Essayons désormais d'ajouter garfiled à nos utilisateurs Samba :

```
# smbpasswd -a garfield
New SMB password:
Retype new SMB password:
Added user garfield.
```

Notre utilisateur a été ajouté correctement ! Nous pouvons le vérifier en listant les utilisateurs connus de Samba :

```
# pdbedit -L
Searching for:[(&(objectClass=sambaDomain)(sambaDomainName=UBUNTU))]
smbldap_open_connection: connection opened
Searching for:[(&(objectClass=sambaDomain)(sambaDomainName=UBUNTU))]
smbldap_open_connection: connection opened
ldapsam_setsampwent: 1 entries in the base dc=martymac,dc=com
init_sam_from_ldap: Entry found for user: garfield
garfield:10001:garfield
```

Test de connexion au partage

Testons enfin la connexion à notre partage avec l'utilisateur garfield :

```
# smbclient -U garfield //ubuntu/donnees
```

Le test peut évidemment être réalisé depuis une machine Windows si vous préférez ! Dans tous les cas, l'utilisateur garfield a bien le droit de se connecter et peut créer un fichier dans le répertoire partagé !

Evolutions

Nous n'avons présenté ici qu'un serveur de fichiers autonome. Le principe pour un contrôleur de domaine est identique, excepté que le contrôleur de domaine doit, en plus, être capable d'ajouter lui-même les comptes POSIX à l'annuaire. Ici interviennent d'autres directives Samba (add user script, add machine script, ...) et la nécessité d'utiliser des scripts externes tels les ldapscripts (**http://contribs.martymac.com**).

Conclusion

LDAP, par le biais de sa standardisation, permet une interopérabilité simple, fiable et pérenne et offre ainsi cet avantage de pouvoir centraliser l'information au sein d'une entreprise : comptes POSIX, adresses de messagerie, et autres informations y trouvent leur place.

OpenLDAP offre une implémentation complète et robuste de ce standard en proposant un serveur et des outils clients.

Vous avez appris à travers ce cours à mettre en place votre premier annuaire. Félicitations, mais ça n'est qu'un début... ! La connexion de Samba à cet annuaire est un exemple parmi d'autres, qui, je l'espère, vous donnera envie d'aller plus loin et de mettre en place un véritable système d'informations centralisé grâce à LDAP.

Liens

Voici quelques liens utiles pour compléter ce cours :

- Le site d'OpenLDAP :
  http://www.openldap.org
- La documentation d'administration d'OpenLDAP 2.4 :
  http://www.openldap.org/doc/admin24
- Les pages de man : slapd.conf, slapd.access, slapd.replog, ldif, libnss-ldap.conf, [...]
- Les RFCs (Requests For Comments) :
  Note : Les RFCs évoluent régulièrement. Depuis juin 2006, la RFC 4510 propose des mises à jours aux RFCs originales citées dans ce document. Elles sont complétées ou remplacées par les RFC 4511 et suivantes.
  2247 (4519, 4524) : Nommage des domaines dans LDAP/X500
  2251 (4511, 4512, 4513) : LDAPv3
  2252 (4512, 4517, 4523) : LDAPv3 - Syntaxe des attributs
  2253 (4514) : LDAPv3 - Normalisation des DNs
  2254 (4515) : Les filtres de recherche LDAP
  2255 (4516) : L'URL LDAP
  2222 : SASL
  2307 : Utilisation de LDAP comme NIS (utilisation des comptes POSIX)
  2849 : Le format LDIF
- PADL qui implémente les librairies libnss-ldap et libpam-ldap :
  http://www.padl.com

- Dernière version de ce document et ldapscripts disponibles sur :
  http://contribs.martymac.com

Les listes de diffusion :

- Plusieurs listes existent. Vous pouvez vous y inscrire à cette adresse :
  http://www.openldap.org/lists

Quelques sites intéressants :

- L'essentiel du Labo-linux :
  http://www.labo-linux.org/index.php?page=essentiels&id=416
- Un livre complet sur OpenLDAP :
  http://www.zytrax.com/books/ldap

Licence : GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

*A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
*B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
*C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
*D. Preserve all the copyright notices of the Document.
*E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

*F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
*G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
*H. Include an unaltered copy of this License.
*I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
*J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
*K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
*L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
*M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
*N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
*O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this

License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
   Texts.  A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples

in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.