

# **OpenLDAP**

## Ultimate

© Anahuac de Paula Gil 2015

Editor: Rafael Martins Trombetta

Capa: Buqui Editora

Editoração: Cristiano Marques

[www.buqui.com.br](http://www.buqui.com.br)

[www.editorabuqui.com.br](http://www.editorabuqui.com.br)

[www.autopubli.com.br](http://www.autopubli.com.br)



Licença de Cultura Livre - CC BY 4.0

<http://creativecommons.org/licenses/by/4.0/deed.pt>

CIP-Brasil, Catalogação na fonte  
Sindicato Nacional dos Editores de Livros, RJ

---

G392o

Gil, Anahuac de Paula

Openldap : ultimate / Anahuac de Paula Gil.

1. ed. | Porto Alegre/RS | Buqui, 2015.

200p. | 21 cm

ISBN 978-85-8338-184-6

1. Software livre. 2. Software. 3. Informática - Estudo e ensino. I. Título.

15-21487 | CDD: 005.1 | CDU: 004.4

---

31/03/15 | 06/04/15



Anahuac de Paula Gil

# OpenLDAP

## Ultimate

buqi



## PREFÁCIO

Gestão centralizada de recursos é algo fundamental na vida de um administrador de sistemas que deseja ser um bom profissional. Por recursos, não pense apenas em identidades. Em sistemas mais complexos de TI, um administrador precisa lidar com dezenas de recursos diferentes - tais como pessoas, grupos, computadores, impressoras, dispositivos de rede, e muitos outros - e se não houver cuidado na organização dessas informações, naturalmente vão haver muitas coisas duplicadas, gerando confusão e aumentando significativamente o esforço para manter as coisas em ordem.

Organizar informação em forma de diretório (ou árvore) é algo bastante antigo. Companhias telefônicas fazem isso há muitas décadas, e daí nasceu a especificação X.500 nos anos 80, de onde o protocolo LDAP é derivado. LDAP é basicamente uma simplificação do X.500 para era TCP/IP. Embora seja uma tecnologia relativamente antiga, e de certa forma até um pouco confusa, LDAP teve uma grande adesão por toda indústria da tecnologia, tornando-se uma peça de conhecimento extremamente importante para qualquer administrador de sistemas.

Não se deixe enganar que LDAP é algo do mundo Unix. Muito pelo contrário. Existem grandes soluções implementando LDAP fora do mundo Unix, como o antigo Novell Directory Services - NDS (hoje NetIQ eDirectory), Microsoft Active Directory, entre outros. Do lado Unix, nenhuma solução se tornou tão popular quanto o OpenLDAP, que é uma excelente implementação, utilizado por infraestruturas de vários portes mundo afora.

Leia este livro com dedicação e pratique com os exemplos apresentados, pois o conhecimento que você irá adquirir aqui é muito valioso. Lembre também que muito do que você vai aprender aqui se aplica a outras soluções, incluindo proprietárias. Arquiteturas modernas de TI são normalmente baseadas em múltiplas plataformas. Mentalidade aberta e foco em interoperabilidade é uma das coisas que você mais precisa para ser um profissional de sucesso.

Anahuac foi além de tratar apenas de LDAP e OpenLDAP neste livro, diferenciando-se de praticamente tudo o que já foi escrito sobre o assunto. Preste atenção nos capítulos dedicados a outros softwares, como Apache, Squid, Postfix, etc. Para muitas pessoas, a maneira mais rápida de aprender algo é com exemplos práticos, o que não falta nas páginas seguintes.

Anahuac, além de um grande amigo, é um profissional que eu tenho muita admiração e respeito. Muito conversamos sobre LDAP há vários anos e fico muito contente que ele tenha absorvido tanto conhecimento no assunto ao ponto de escrever esta bela obra.

A você leitor, desejo uma boa leitura, e tenho absoluta certeza que você vai enriquecer seu conhecimento nos próximos dias.

Marlon Dutra

# SUMÁRIO

## CAPÍTULO 1

### CONCEITOS, ARQUITETURA E DESIGN

1.1 O que são os serviços de diretórios? .....	13
1.2 Qual sua utilidade? .....	13
1.3 O que é LDAP? .....	13
1.4 Qual é a estrutura de uma base LDAP? .....	15
1.5 Entendendo os diretórios .....	15
1.6 Entendendo os schemas, ObjectClasses e atributos	17
1.7 Registros em uma base LDAP .....	19
1.8 Resumindo .....	20

---

## CAPÍTULO 2

### INSTALAÇÃO

2.1 Instalando a partir dos pacotes pré-compilados.....	25
2.2 Instalando a partir do código fonte.....	26
2.3 Configuração do OpenLDAP.....	27
2.4 Dissecando o slapd.d .....	28
2.4.1 cn=config.ldif.....	28
2.4.2 cn=config .....	29
2.5 Opções do DB_CONFIG .....	32

---

## CAPÍTULO 3

### USO E GERENCIAMENTO

3.1 Série de comandos “slap” .....	37
3.1.1 slaptest.....	37
3.1.2 slapcat .....	37
3.1.3 slapadd .....	38
3.1.4 slappasswd.....	39
3.1.5 slapindex.....	40
3.2 Série de comandos “ldap” .....	40
3.2.1 ldapsearch .....	40
3.2.2 Aprendendo a sintaxe do “filter” .....	41

3.2.3 ldapadd.....	43
3.2.4 ldapmodify.....	44
3.2.5 ldapdelete.....	46
3.2.6 ldapmodrdn.....	46

---

## CAPÍTULO 4

### CONHECENDO ALGUNS CLIENTES LDAP

4.1 PhpLdapAdmin.....	49
4.2 Luma.....	52
4.3 ldapvi.....	55

---

## CAPÍTULO 5

### GERENCIANDO LOGS

5.1 Loglevel.....	59
5.2 Syslog.....	60

---

## CAPÍTULO 6

### TRABALHANDO COM ÍNDICES

6.1 Indexando.....	66
6.2 Cuidando do permissionamento.....	66

---

## CAPÍTULO 7

### ACL – ACCESS CONTROL LIST

7.1 Sintaxe.....	72
7.2 Tabela de Entidades.....	72
7.3 Tabela dos níveis de acesso.....	72
7.4 Explicando o formato cn=config.....	73
7.5 Exemplos.....	73

---

## CAPÍTULO 8

### BACKUPS E RESTAURAÇÃO

8.1 Usando ldapsearch.....	79
8.2 Usando slapcat.....	79
8.3 Salvando o cn=config.....	79
8.4 Exemplo de script.....	80



8.5 Restaurando o backup.....	81
8.5.1 Restaurando o cn=config.....	81
8.5.2 Restaurando a base .....	82
8.6 Explicando .....	82
8.7 Usando db_recovery.....	83
8.8 Recuperando a senha do usuário Admin .....	84
8.8.1 Restaurando a senha original.....	85

---

## CAPÍTULO 9

### SUORTE A CRIPTOGRAFIA

9.1 Ativando TLS.....	89
9.2 Alterando o cn=config para ativar o TLS .....	97
9.3 Ativando o suporte aos clientes.....	98
9.4 O teste final não funcionou. E agora?.....	99

---

## CAPÍTULO 10

### REPLICAÇÃO COM SYNCREPL

10.1 Entendendo o Syncrepl.....	103
10.2 Configurando o “Servidor Mestre” para syncrepl.....	105
10.2.1 Explicando o “ldif” .....	106
10.3 Configurando o “Servidor Secundário” ou “Slave” para Syncrepl .....	108
10.3.1 Explicando o “ldif” .....	110
10.8 Ativando a replicação “Syncrepl” .....	111
10.9 Ativando encriptação na replicação.....	111
10.10 Populando a base de testes .....	112

---

## CAPÍTULO 11

### INTEGRANDO SERVIDOR APACHE

11.1 Instalando o Apache .....	117
11.2 Ativando módulo de suporte ao LDAP .....	117
11.3 Criando “virtualhost” com acesso autenticado... ..	117
11.3 Criando acessos condicionais .....	119
11.4 Testando autenticação.....	120

---

## CAPÍTULO 12

### INTEGRANDO SERVIDOR PROXY

12.1 Instalando o Squid .....	125
12.2 Testando o autenticador LDAP .....	125
12.3 Lidando com usuários e grupos.....	125
12.4 Testando o autenticador de grupo e usuário .....	126
12.5 Alterando o squid.conf.....	126
12.6 Testando autenticação .....	128
12.7 Monitoramento de logs com SARG .....	129

---

## CAPÍTULO 13

### SERVIDOR POSTFIX

13.1 Instalando o Postfix .....	133
13.2 Criando schema e/ou utilizando os já existentes	140
13.3 Instalando o schema .....	141
13.4 Configurando o Postfix padrão.....	142
13.5 Testes de funcionamento.....	143

---

## CAPÍTULO 14

### SERVIDOR POP/IMAP COM CYRUS

14.1 Instalando e configurando suporte a SASL .....	149
14.2 Instalando o Cyrus .....	151
14.3 Configurando o Cyrus .....	151
14.4 Configurando o Postfix para usar LMTP .....	151
14.5 Criando contas no Cyrus .....	152
14.6 Testes de funcionamento.....	153

---

## CAPÍTULO 15

### SERVIDOR SAMBA

15.1 Conceituação sobre o Samba e seu modo de autenticação.....	157
15.2 Instalando o Samba .....	159
15.3 Samba como PDC .....	160
15.4 Instalando o schema do Samba no OpenLDAP..	160
15.5 Configurando o Samba para usar LDAP .....	161

15.6 Configurando o smbldap-tools .....	163
15.7 Testes de funcionamento.....	165

---

## CAPÍTULO 16

### NSS E PAM – AUTENTICANDO O SISTEMA NO LDAP

16.1 Instalando e configurando o NSS - Name Server Switches .....	172
16.2 Instalando e configurando o PAM - Pluggable Authentication Modules.....	174
16.3 Teste de funcionamento .....	177

---

## CAPÍTULO 17

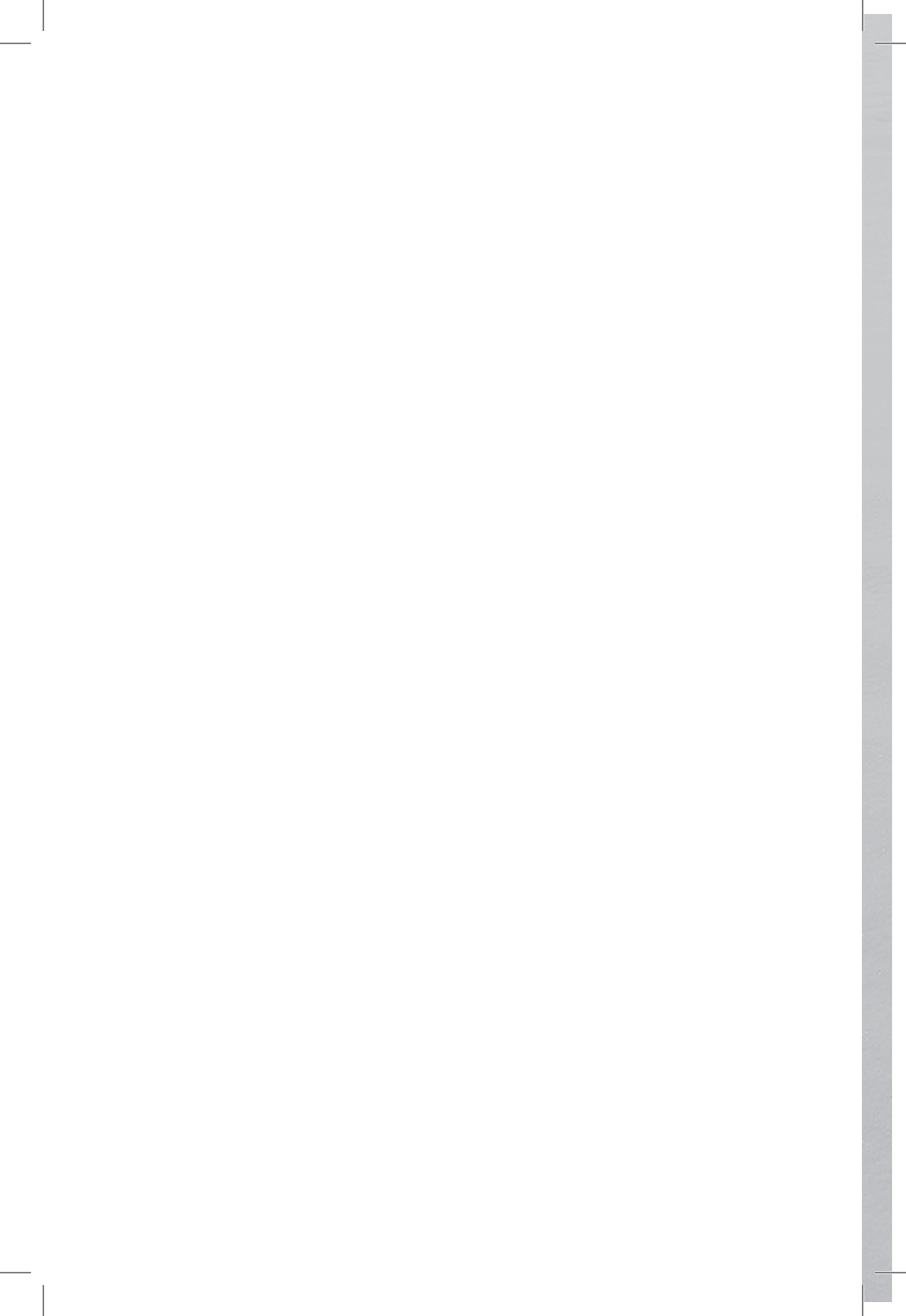
### SERVIDOR FTP

17.1 Instalando o “vsftp” .....	181
17.2 Configurando vsftp .....	181
17.3 Teste de autenticação.....	182
17.4 Instalando o “proftpd” .....	183
17.5 Configurando o “proftpd” .....	183
17.6 Teste de autenticação.....	185

---

## APÊNDICE

Apêndice I – Schemas: Tipo de valores .....	188
Apêndice II – qmailuser.schema .....	190





CAPÍTULO 1

CONCEITOS,  
ARQUITETURA E DESIGN



## 1.1 O que são os serviços de diretórios?

Segundo a Wikipédia, serviço de diretórios é:

*“Um serviço de diretório é um software que armazena e organiza informações sobre os recursos e os usuários de uma rede de computadores, e que permite os administradores de rede gerenciar o acesso de usuários e sistemas a esses recursos. Além disso, os serviços de diretório atuam como uma camada de abstração entre os usuários e esses recursos.*

*A palavra directory em inglês poderia ser melhor traduzida por catálogo, no sentido de uma lista ordenada com descrição curta dos itens, não necessariamente de arquivos.”*

A sua primordial diferença está na forma como os recursos são tratados. Entenda por recursos todos os componentes de uma rede de computadores, como os compartilhamentos, configurações, usuários, senhas, permissões e mais.

## 1.2 Qual sua utilidade?

Permitir a centralização de gestão dos recursos da rede, visando simplificar a administração, backup e replicação.

Centralização é a palavra chave de todas as facilidades encontradas no uso de um serviço de diretórios.

## 1.3 O que é LDAP?

LDAP é um protocolo. Como todos os protocolos sua função é definir a forma de funcionamento de um serviço de diretórios especificando critérios, mecanismos e métodos para armazenar e fornecer informações.

LDAP – “Lightweight Directory Access Protocol” (LDAP) ou Protocolo Leve de Acesso à Diretórios é um conjunto de protocolos desenhados para acessar informação centralizada em uma rede.

Esse conjunto de protocolos serve para interagir com o Serviço de Diretório. Assim as regras de acesso ao diretório estão definidas no LDAP. O LDAP é definido para uso em sistemas

Cliente-Servidor permitindo a um cliente LDAP consultar ou alterar o diretório comunicando-se com o servidor LDAP. É multi-plataforma.

O servidor LDAP tem a função de verificar as credenciais do cliente, verificar se as informações solicitadas estão armazenadas neste servidor e permitir ou não que o cliente realize consultas e modificações. As formas de armazenamento dos dados, tipo de gerenciador de bancos de dados usado e sistema operacional de base não fazem parte do protocolo. Fazem parte da implementação específica do LDAP.

A manutenção de um cadastro centralizado de usuários é um grande desafio para o administrador de TI. A rotatividade de colaboradores e a mobilidade na hierarquia tornam difícil a manutenção de informações coerentes sobre determinado recurso de uma organização.

O grande número de sistemas e a heterogeneidade de plataformas faz com que a manutenção de senhas e identidades seja um grande desafio. É repetidamente motivo para escolha de senhas frágeis e vazamento de informações.

O uso do LDAP permite que colaboradores, aplicativos e recursos de rede possam usar informações armazenadas em um repositório central. Isso unifica os esforços de criação, manutenção da base de informações. O LDAP permite a consulta a informações cadastrais, o que permite sua utilização como agenda de contatos central da organização, um dos primeiros usos para o protocolo.

Como vantagens do LDAP podemos citar:

- Por ser um padrão aberto, a interoperabilidade entre os diversos fornecedores é facilitada. Um cliente LDAP baseado em OpenLDAP pode perfeitamente realizar consultas e atualizações em um servidor de outro fornecedor que siga os padrões LDAP. O LDAP é um protocolo. As implementações podem trazer novas interfaces e ferramentas de administração e consulta, mas os métodos básicos são definidos no protocolo;



- API (“Application Programming Interface”, Interface para Programação) bem definida e com suporte para diversas linguagens de programação;
- Muito mais rápido que Sistemas de Bancos de Dados (considerando que atualizações são menos constantes que consultas);
- Esquemas (regras para o armazenamento de dados) padronizados existem para diferentes funções;
- Permite a consolidação de informações de várias fontes;
- Facilmente replicável e distribuível.

### ***1.4 Qual é a estrutura de uma base LDAP?***

Uma base LDAP busca organizar as informações em forma de diretório, ou seja, em forma de “árvore”. As partes que permitem essa formação são as especificações do protocolo.

Baseando-se em campos, chamados de atributos e em seus conjuntos, chamados de schemas, é possível armazenar praticamente qualquer tipo de informação de forma estruturada o que facilita sua administração.

Cada ramificação da “árvore” pode ser um departamento da organização, permitindo ter um efeito visual organizacional da base.

### ***1.5 Entendendo os diretórios***

O primeiro passo para entender como uma base LDAP está estruturada é realizar um exercício mental para eliminar quaisquer outros conceitos pré-existentes. Um erro muito comum é tentar estabelecer elos com bases de dados SQL ou DB.

Bases LDAP tem uma estrutura muito singular e qualquer comparação servirá apenas para criar confusão, especialmente para os iniciantes. Portanto evite as comparações e abra a mente para entender algo completamente novo.

Visando simplificar o entendimento vamos utilizar a palavra diretório com o seu mais conhecido conceito: o de pasta de armazenamento de arquivos e outros diretórios.

Sabemos que um diretório no sistema de arquivos nada mais é do que uma divisão lógica que visa organizar os arquivos existentes no disco rígido. Pense no LDAP nos mesmos termos. Assim começa a ficar claro que a estrutura de uma base LDAP é completamente diferente de qualquer outra base de dados comum.

Em um sistema de arquivos o diretório principal chama-se raiz e isso deve-se a sua estrutura em forma de “árvore”. O LDAP mantém essa mesma idéia, ou seja, a partir da “raiz da árvore”, estão suas ramificações, que permitem a organização lógica dos arquivos.

Assim como em uma estrutura de diretórios o LDAP permite a existência de outros diretórios dentro de um diretório já existente. Portanto cada novo diretório é raiz em si mesmo para todo o seu conteúdo.

Em um diretório não é permitido ter arquivos com o mesmo nome e isso faz sentido, afinal, é fundamental ter apenas um nome que identifique cada arquivo.

Uma vez compreendido que a base LDAP funciona como um diretórios de um sistema de arquivos basta fazer a relação entre arquivos e recursos. Em uma base LDAP não se armazenam arquivos mas sim recursos de rede, como usuários e senhas, por exemplo.

Assim como em alguns sistemas de arquivos existem regras para a criação de arquivos, a base LDAP também possui regras para o armazenamento de recursos. Estas regras são muito mais complexas do que simplesmente limitar o número de caracteres no nome de um arquivo, afinal trata-se do armazenamento de dados.

Estas regras ou limites são impostos pela definição do protocolo e elas tem que ser respeitadas. Caso contrário a base simplesmente negará a adição de informações.

Qual informação e qual conteúdo podem ser adicionados, são definidos pelo conjunto de atributos e seus conjuntos, as ObjectClasses. Portanto quanto mais schemas sua base LDAP tiver, mais flexível ela será, permitindo o armazenamento de mais recursos.

O conjunto de ObjectClasses e atributos são chamados de “schemas”

## ***1.6 Entendendo os schemas, ObjectClasses e atributos***

Os schemas nada mais são do que arquivos texto contendo uma série de ObjectClasses. Os ObjectClasses nada mais são do que um conjunto de atributos, e os atributos nada mais são do que a definição lógica dos campos que podem ser utilizados em uma base LDAP.

Cada schema habilita um determinado tipo informação a ser armazenada na base. Os schemas são lidos pela base LDAP e assim permite o uso dos campos nele definido.

Para ilustrar essa idéia vamos usar um exemplo de criação de um usuário válido para sistemas Gnu/Linux.

Sabemos que quando um usuário comum é criado, alguns atributos são obrigatórios, como seu UID e GID, além do diretório home, senha e shell de login. Esses atributos podem ser facilmente vistos no arquivo `/etc/passwd`.

De forma similar, quando se quer adicionar um usuário “posix” em uma base LDAP esses atributos também são obrigatórios. Mas como saber disso? Quem define quais são obrigatórios? Alguns desses atributos são apenas opcionais?

O arquivo de schema que contém esses atributos chama-se “`nis.schema`” e nele vamos encontrar o objectClass `posixAccount`. Veja abaixo:

1. objectclass (1.3.6.1.1.1.2.0 NAME ‘posixAccount’
2. DESC ‘Abstraction of an account with POSIX attributes’
3. SUP top AUXILIARY

4. MUST (cn \$ uid \$ uidNumber \$ gidNumber \$ homeDirectory)

5. MAY (userPassword \$ loginShell \$ gecos \$ description))

Entendendo essa descrição:

Linha um: Definição do objectClass, contendo o seu número único e nome;

Linha dois: Descrição livre do objectClass;

Linha três: Herança do objeto. Neste caso ele é um objeto auxiliar do objectClass “top”;

Linha quatro: Atributos obrigatórios para o uso deste objeto;

Linha cinco: Atributos opcionais para o uso deste objeto.

Agora vamos ver um exemplo de um dos atributos obrigatórios:

1. attributetype (1.3.6.1.1.1.3 NAME ‘homeDirectory’

2. DESC ‘The absolute path to the home directory’

3. EQUALITY caseExactIA5Match

4. SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)

Linha um: Definição do atributo, contendo seu número único e nome;

Linha dois: Descrição do atributo;

Linha três: Definição do tipo de valor que pode ser armazenado nesse atributo.

Linha quatro: Sintaxe válida

Todos esses valores fazem parte da especificação do protocolo LDAP.

Cada arquivo de schema contém uma série de objetos e atributos para atender necessidades específicas de recursos específicos, ou seja, os atributos necessários para se adicionar um usuário do Samba são bem diferentes dos atributos necessários para se adicionar um usuário “posix”. Portanto, para cada recurso que se deseje armazenar em uma base LDAP deve-se ter um schema próprio.

É fato que para a maioria dos serviços existentes já existe um schema, bastando incluí-lo na base LDAP para que possa ser utilizado.

Para ver mais detalhes sobre às regras dos tipos de informação que podem ser utilizados em um atributo e suas sintaxes verifique às tabelas no Apêndice I.

## *1.7 Registros em uma base LDAP*

Como foi visto acima, as regras para adicionar um registro em uma base LDAP, são muito mais complexas do que para criar um arquivo em um diretório comum em um sistema de arquivos.

Entretanto a similaridade está no fato de que não pode haver dois registros iguais em um mesmo diretório. Para garantir essa unicidade para todos os registros utiliza-se o identificador DN (Distinguished Name).

Um registro único pode conter diversos objetos e atributos, de acordo com sua finalidade. Aqui vemos um exemplo de registro único:

`cn="Anahuac de Paula Gil",ou=Usuarios,dc=anahuac,dc=org`

É importante entender o que está descrito acima. Esta é uma descrição do formato padrão que deve ser utilizado em nome da compatibilidade com os diversos serviços que oferecem suporte à bases LDAP. Portanto não é obrigatório, as extremamente desejável.

Quando uma base LDAP é criada precisa-se definir a raiz da “árvore” do diretório. Esta raiz normalmente é definida pelo atributo “domainComponent”, também conhecido como “dc”, com o nome do domínio da internet da organização.

Neste exemplo: `dc=anahuac,dc=org`

Dentro da raiz foi criado um outro “galho” da “árvore” chamado Usuarios cujo atributo é ou (Organizational Unit).

Neste exemplo: `ou=Usuarios`

Finalmente, dentro do “galho” Usuarios foi criado um usuário, cuja identificação única está sendo feita pelo atributo cn (Common Name).

Neste exemplo: cn="Anahuac de Paula Gil"

Pode-se também referenciar um usuário utilizando o atributo uid em vez de cn, entretanto o padrão é usar o cn.

O objeto final, que é o usuário, poderá e deverá conter diversos atributos para atender às exigências de diversos servidores. Por exemplo:

```
dn: cn="Anahuac de Paula Gil",ou=Usuarios,dc=anahuac,dc=org
uid: anahuac
sn: anahuac
objectClass: top
objectClass: person
objectClass: qmailUser
homeDirectory: /home/anahuac
userPassword:: e1NTSEF9MGNjcXJKTEVOQU9nTSswR2l-
4TVRtelhBWERObng5cFU=
cn: Anahuac de Paula Gil
mail: anahuac@anahuac.org
```

Perceba que o primeiro registro contém o identificador único: "dn:"

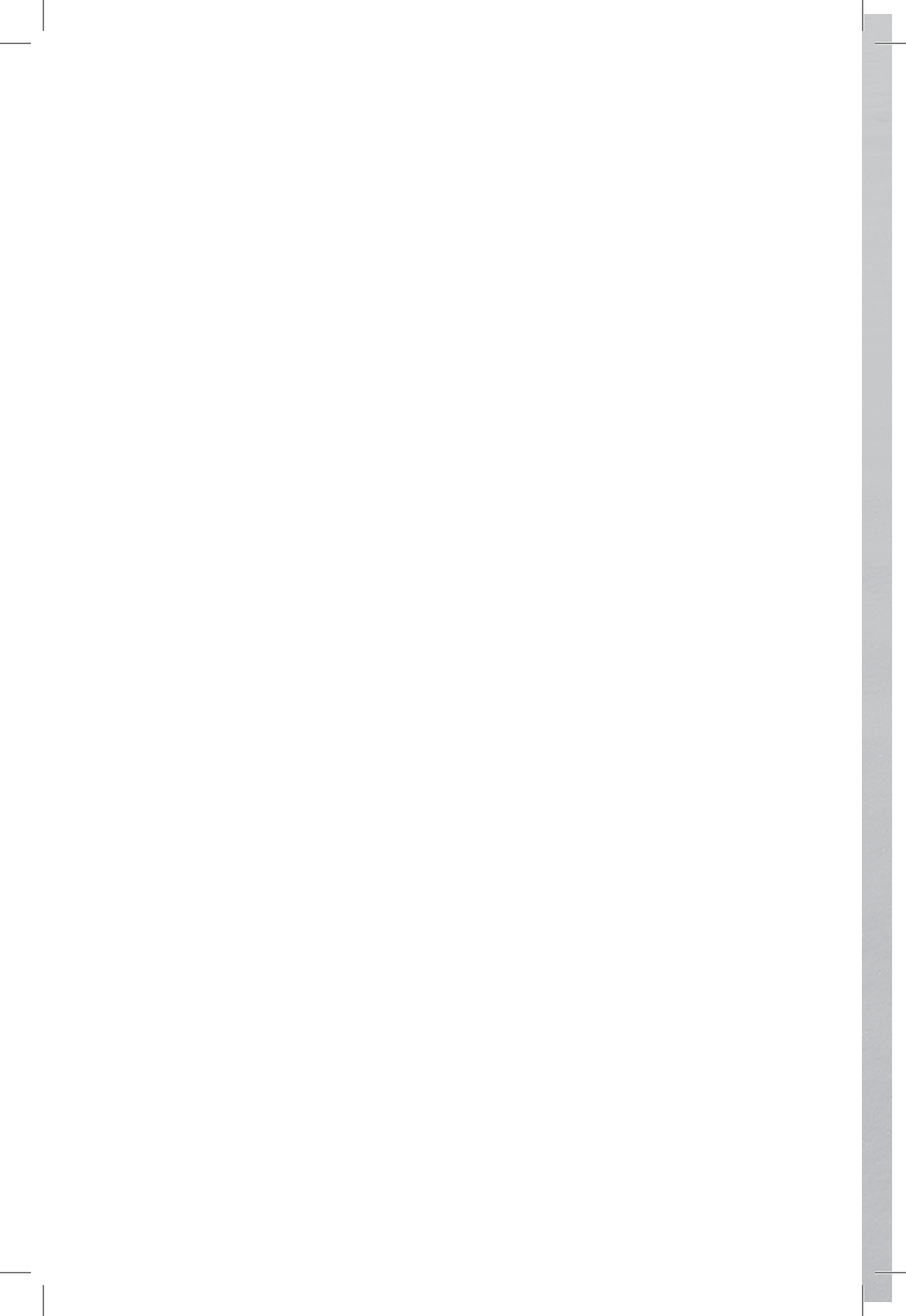
Para efeito de comparação teríamos o usuário anahuac dentro do diretório Usuarios que está dentro da raiz anahuac.org. Visualizando:

```
dc=anahuac,dc=org
|
-----> ou=Usuarios
      |
            -----> cn="Anahuac de Paula Gil"
```

O objetivo de manter as informações assim é simplesmente para facilitar a localização e conseqüentemente, facilitar a administração dos recursos.

## *1.8 Resumindo*

1. A base LDAP tem um funcionamento similar aos diretórios dos sistemas de arquivos. Portanto não há nenhuma relação direta com bases de dados SQL;
2. A base LDAP armazena recursos e não arquivos;
3. A base LDAP utiliza um conjunto de objetos e atributos para fazer esse armazenamento;
4. O conjunto de objetos e atributos é chamado de schema e não passa de um arquivo texto que deve ser lido pela base LDAP para que possam ser usados;
5. Um objeto é chamado de ObjectClass e ele é composto de atributos obrigatórios e atributos opcionais;
6. Um atributo é uma definição de campo, contendo a definição de que tipo de valor ele pode armazenar.
7. Dentro de uma base LDAP cada registro é único, ou seja, não podem haver dois registros iguais em um mesmo diretório.







CAPÍTULO 2

INSTALAÇÃO



O primeiro passo na instalação do do OpenLDAP é configurar corretamente o nome do servidor. Em servidores “posix” o formato do nome segue a seguinte sintaxe:

```
<nome>.<domínio>
```

Ex: server.anahuac.org

Para configurar corretamente o nome do servidor será necessário alterar dois arquivos:

1) /etc/hosts

Certifique-se que existe uma linha contendo o endereço IP do servidor, nome.domínio e apelido. Algo como:

```
192.168.0.222 server.anahuac.org server
```

2) /etc/hostname

Nesse arquivo deve estar o nome.domínio. Assim:

```
server.anahuac.org
```

3) Para que o novo hostname passe a ser utilizado pelo sistema será necessário reiniciar a máquina ou executar os comandos abaixo:

```
hostname server.anahuac.org
```

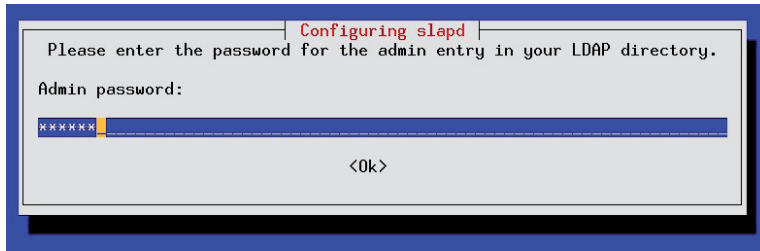
```
sysctl kernel.hostname=server.anahuac.org
```

## ***2.1 Instalando a partir dos pacotes pré-compilados***

No Gnu/Linux Debian será necessário instalar o pacote “slapd”, com o seguinte comando:

```
aptitude install slapd
```

Este comando irá fazer o download dos pacotes necessários e em seguida solicitará que a senha do usuário “admin” seja fornecida e confirmada. Como na figura abaixo:



Importante deixar claro que no Gnu/Linux Debian a base inicial é criada automaticamente pelo instalador. Por isso ele pede a senha do usuário “admin”.

Em outras distribuições, como RedHat ou Ubuntu, o instalador não cria a base inicial, ou seja, é necessário criá-la manualmente. Outra diferença entre ambos os sistemas é quanto à localização dos arquivos de configuração, que no Debian fica em `/etc/ldap` e no RedHat fica em `/etc/openldap`.

Pode-se usar o comando “`nmap localhost`” e verificar se a porta 389, que é a porta padrão do servidor OpenLDAP, está ativa.

## *2.2 Instalando a partir do código fonte*

Para fazer a instalação a partir do código fonte, será necessário instalar algumas bibliotecas exigidas. São elas:

- OpenSSL: Pacote no Debian: `libssl-dev`
- SASL: Pacote no Debian: `libsasl2`
- Berkeley DB: Pacote no Debian: `libdbXXX-dev` – onde XXX é a versão do db que estiver sendo usado no servidor.

Portanto comece instalando esses pacotes com o comando:  
`aptitude install libssl-dev libsasl2 libdb-dev time`

Baixe o código fonte do OpenLDAP no site oficial do projeto em:  
<http://www.openldap.org/software/download/>

E siga os seguintes passos:

1. Faça sua descompressão em `/usr/local/src`

- ```
# tar xzf openldap-stable-<versão>.tgz -C /usr/local/src
```
2. Vá até o diretório criado e, após ler o README, verifique quais as opções de compilação você quer ativar e quais quer desativar

```
# cd /usr/local/src/openldap-<versão>
# ./configure --help
```
  3. Utilize o comando “configure” com as opções desejadas para efetuar a verificação de que todos os requisitos de compilação estão presentes em seu sistema

```
# ./configure --with-tls --enable-memberof=mod
```
  4. Verifique as dependências

```
# make depend
```
  5. Compile o código

```
# make
```
  6. Uma vez compilado o código fonte, podemos testar se tudo ocorreu devidamente

```
# make test
```
  7. Se tudo estiver certo, podemos instalar o software

```
# make install
```

### ***2.3 Configuração do OpenLDAP***

Versões antigas do OpenLDAP utilizavam um arquivo de configuração como a maioria dos serviços “Posix”: alterava-se o arquivo com os parâmetros desejados e reiniciava-se o serviço.

Entretanto esse método deixou de ser utilizado. A partir da versão 2.4 as configurações são armazenadas dentro de uma base exclusiva para esse fim. Essa base é chamada de “cn=config” e o objetivo é permitir que as configurações tenham efeito imediato, sem a necessidade de reiniciar o serviço.

O cn=config usa arquivos textuais como base para armazenamento dos seus dados. Isso implica que pode-se alterar esses

arquivos e reiniciar o serviço como se fazia anteriormente ou pode-se criar um arquivo “ldif” com a configuração desejada e adicioná-lo de forma a que ela tenha efeito imediato.

Essa estrutura é mantida, por padrão, no diretório “/etc/ldap/slapd.d”.

O arquivo ldap.conf é destinado às configurações de ferramentas cliente para acesso a uma base LDAP.

## *2.4 Dissecando o slapd.d*

### *2.4.1 cn=config.ldif*

No diretório slapd.d contém um arquivo chamado “cn=config.ldif” e ele contém as configurações genéricas do OpenLDAP. Essas opções afetam todo o serviço.

Edite o arquivo com o editor de sua preferência. Vamos comentar todas as opções:

- dn: cn=config

Perceba que esse é um arquivo “ldif” que é lido pelo slapd sempre que ele é iniciado. Portanto ele tem que começar com o DN – Distinguished Name. Neste caso é o nome da base em si: cn=config.

- objectClass: olcGlobal

Este é o “objectClass” que disponibiliza todos os atributos necessários para configurar o OpenLDAP. Você perceberá, mais adiante, que muitos dos atributos possuem o prefixo “olc” que significa “Open LDAP Configuration”.

- cn: config

Nome do galho, ou “rama”. CN significa Common Name.

- `olcArgsFile: /var/run/slapd/slapd.args`

Esse atributo recebe, como valor, o caminho completo e nome do arquivo de argumentos extras do OpenLDAP. Em geral é nele onde se definem as opções de inicialização do serviço, como usuário que será utilizado e o diretório de configuração.

- `olcLogLevel: none`

Nível de registro de atividades no OpenLDAP. O nível 0 determina que não se faça registro de atividades. Os níveis de atividade (log), serão detalhados em um capítulo próprio.

- `olcPidFile: /var/run/slapd/slapd.pid`

Onde será salvo o arquivo contendo o PID do processo de execução do OpenLDAP. Perceba que se você mudar esta opção o “script” de inicialização em “/etc/init.d” não conseguirá parar o serviço.

- `olcToolThreads: 1`

Esta opção define o número de CPU’s que serão utilizadas para a indexação da base de dados. Se o servidor tiver mais de um processador, aumente este número.

Todos os demais atributos desse arquivo são gerados de forma automática e não há necessidade alguma de alterá-los.

## 2.4.2 *cn=config*

Entrando no diretório “cn=config” existem seis arquivos “ldif” e um diretório chamado “cn=schema”. Cada um dos arquivos configura uma parte da base LDAP e o diretório mantém os arquivos com os “schemas”, ou seja, todos os atributos disponíveis pela base.

Perceba que na composição do nome dos arquivos há um número entre chaves. Essa numeração indica a ordem pela qual esses arquivos serão lidos pelo OpenLDAP.

Vamos descrever esses arquivos e seus principais conteúdos:

- `cn=module{0}.ldif`

Esse é o arquivo onde os módulos são descritos e explicitados. O OpenLDAP é modular e para cada novo recurso será neces-

sário carregar um módulo. Diversos recursos como o “ppolicy” que permite definições elaboradas de políticas de acesso e o “syncprov” que permite a ativação de uma das forma de replicação, são exemplos de módulos.

O atributo “olcModuleLoad”, pelo seu valor “{0}back\_hdb” indica que existirá ao menos uma base de dados do tipo “hdb” que será utilizada pelo OpenLDAP.

- cn=schema.ldif

Esse arquivo cria o galho ou rama chamado “schema”

- olcBackend={0}hdb.ldif

Este arquivo indica a existência de uma base de dados “hdb” que será utilizada pelo OpenLDAP para armazenar os dados.

- olcDatabase={0}config.ldif

Este arquivo mantém as configurações da base de configuração “cn=config”, indicando inclusive, o usuário “admin” e as permissões de acesso.

Pode-se adicionar a este arquivo a senha do usuário “admin” da base de configuração. Isso permitirá fazer alterações de configuração no servidor OpenLDAP de forma remota. Por outro lado isso pode representar um risco indesejado.

A senha do usuário admin é definida pela adição do atributo “olcRootPW” a esse arquivo.

Por outro lado, pode-se utilizar o comando abaixo, para fazer alterações na base de configurações sem a necessidade de uso de senha.

- ldapadd -Y EXTERNAL -H ldapi:/// -f arquivo.ldif

Esse comando estará disponível, somente para o usuário “root” localmente no mesmo servidor.

- olcDatabase={-1}frontend.ldif

Este arquivo mantém configurações genéricas para todas as bases de dados do OpenLDAP. Eventualmente uma configuração específica de uma base de dados terá precedência sobre as configurações feitas neste arquivo. Especial atenção para o atributo “olcSizeLimit” e seu valor padrão de “500”. Esta opção de-



fine o retorno máximo a uma consulta feita na base OpenLDAP. Portanto se alguma pesquisa tiver que retornar mais do que 500 resultados será necessário aumentar este valor. Se o valor de `size-limit` for definido como -1 então o retorno será ilimitado.

- `olcDatabase={1}hdb.ldif`

Este arquivo traz as configurações da base de dados “hdb” que será utilizada pelo OpenLDAP para armazenar os dados. Esse é, de fato, o arquivo onde estão definidas as configurações da base propriamente dita. Nome da raiz, permissionamentos e outros. Por isso vamos dar-lhe mais atenção:

- `olcDbDirectory: /var/lib/ldap`

Onde os arquivos BDB serão armazenados no disco.

- `olcSuffix: dc=anahuac,dc=org`

Nome da raiz da árvore do serviço de diretórios

- `olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous auth by dn="cn=admin,dc=anahuac,dc=org" write by * none`
- `olcAccess: {1}to dn.base="" by * read`
- `olcAccess: {2}to * by self write by dn="cn=admin,dc=anahuac,dc=org" write by * read`

ACL's de acesso ou permissionamento. Quem pode acessar o que? Esses acessos serão estudados de forma detalhada mais adiante.

- `olcLastMod: TRUE`

Ativa o registro do horário de alteração das entradas da base de dados. Útil para replicações e backups

- `olcRootDN: cn=admin,dc=anahuac,dc=org`
- `olcRootPW:: e1NTSEF9di91RmZGbvFhME=`

Usuário e senha do administrador, definidas no momento da instalação.

- `olcDbCheckpoint: 512 30`

Este é um ajuste fino à operação da base de dados BDB. Esta diretiva especifica quando o BDB deve adicionar um ponto de

checagem em seus logs de transação. Neste caso depois de 512 Kbytes ou após 30 minutos desde a última checagem, o que acontecer primeiro.

**OBSERVAÇÃO:** quanto maior for o intervalo entre checkpoints, maior será a probabilidade de que alterações à base de dados sejam irrecuperáveis caso haja uma eventual falha de sistema. Entretanto, se a base de dados não sofrer alterações constantes esse parâmetro pode reduzir a atividade do disco.

- `olcDbConfig: {0}set_cachesize 0 2097152 0`

Esta é uma definição da base de dados DBD. Por padrão o OpenLDAP define 2Mb de memória RAM como limite para o armazenamento de cache. Se o servidor possuir bastante memória RAM pode-se aumentar este valor.

A sintaxe desta linha é: <fator multiplicador> <tamanho do cache em bytes> <quantos caches separados>

Por exemplo: “2 524288000 3” criará um cache de 2.0 Gb, dividido em três caches físicos separados.

- `olcDbConfig: {1}set_lk_max_objects 1500`

Número máximo de objetos que podem ser “trancados”

- `olcDbConfig: {2}set_lk_max_locks 1500`

Número máximo de “travas” tanto para as solicitadas quanto para as realizadas

- `olcDbConfig: {3}set_lk_max_lockers 1500`

Número máximo de “travadores”

- `olcDbIndex: objectClass eq`

Opções de indexação da base LDAP. Todas as opções serão vistas em detalhes mais adiante, neste manual.

## ***2.5 Opções do DB\_CONFIG***

O arquivo DB\_CONFIG é utilizado para passar parâmetros de otimização para o BDB, ou seja, o backend padrão do OpenLDAP.

Este arquivo deve ser escrito no mesmo diretório onde a base de dados BDB será armazenada.

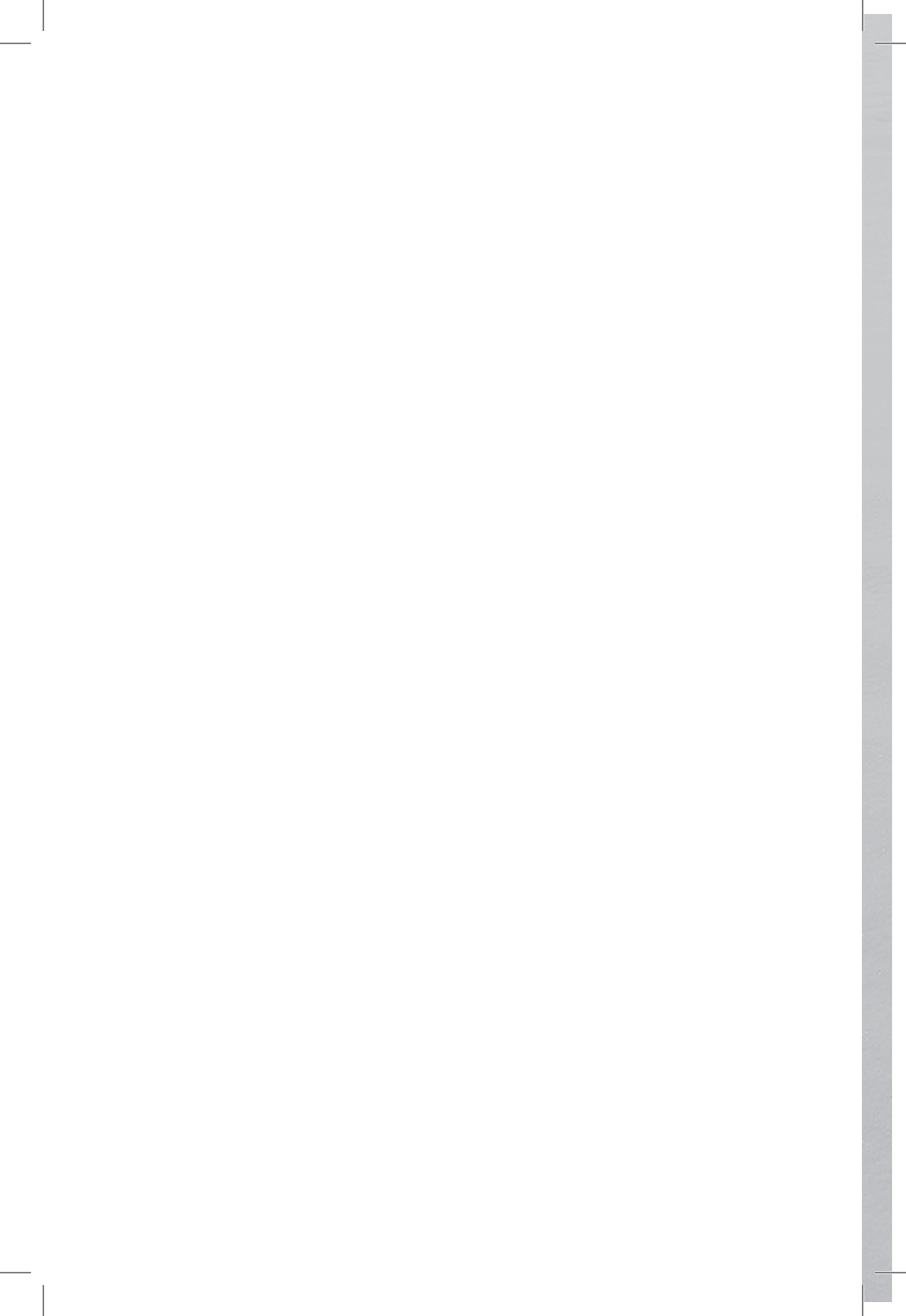
Suas opções são:

- `dbconfig set_cachesize`
- `dbconfig set_lk_max_objects`
- `dbconfig set_lk_max_locks`
- `dbconfig set_lk_max_lockers`

Entretanto estas opções já foram descritas acima. O importante a ser dito é que essas opções podem ser definidas no arquivo “`olcDatabase={1}hdb.ldif`” ou diretamente em `/var/lib/ldap`.

Se o arquivo `DB_CONFIG` não existir o OpenLDAP escreverá essas opções criando o arquivo. Entretanto, se o arquivo já existir o OpenLDAP apenas às ignorará.

O objetivo é otimizar o processo evitando que configurações preexistentes sejam sobrescritas.





CAPÍTULO 3

USO E GERENCIAMENTO



A forma padrão de uso e gerenciamento de uma base OpenLDAP é feita através de uma série de comandos que fazem parte do servidor. Além dos comandos que fazem parte do servidor o OpenLDAP oferece um conjunto extra de comandos.

Os comandos que já vem com o servidor iniciam com **slap** e os comandos extras começam com **ldap**. A diferença crucial é que os comandos **slap** somente devem ser executados “a frio”, ou seja, com o servidor OpenLDAP parado, enquanto que os comando **ldap** somente devem ser executados “quentes”, ou seja, com o servidor OpenLDAP sendo executado.

Para instalar os comandos extras utilize o seguinte comando:

- aptitude install ldap-utils

### ***3.1 Série de comandos “slap”***

Importante lembrar que estes comandos só devem ser executados com a base OpenLDAP parada.

Assim o primeiro passo é parar o OpenLDAP com o comando:

- /etc/init.d/slaped stop

#### ***3.1.1 slaptest***

Este comando testa a integridade do antigo arquivo de configuração “slaped.conf” e do novo sistema baseado na base de dados, e suas opções. Excelente para saber se as opções estão corretamente definidas.

#### ***3.1.2 slapcat***

Este comando permite que a base LDAP seja exportada para um arquivo ldif que é a extensão de arquivos utilizada para adicionar e/ou remover informações de uma base LDAP.

A sintaxe mais comum é: *slapcat -l base\_ldap.ldif*

Não se preocupe com as linhas de saída que por ventura apareçam.

### 3.1.3 *slapadd*

Este comando utiliza um arquivo ldif para adicionar objetos à base LDAP. A seguir vamos ver alguns exemplos de arquivos ldif para adicionar objetos:

#### Adicionando uma “ou” – Organizational Unit

1. Crie o arquivo ou.ldif e copie nele este conteúdo:
  - *dn: ou=Usuarios,dc=anahuac,dc=org*
  - *ou: Usuarios*
  - *objectClass: organizationalUnit*
  - *objectClass: top*
2. Salve o arquivo;
3. Pare o OpenLDAP;
4. Execute o comando:
  - `slapadd -l ou.ldif`

#### Adicionando um usuário ao “ou”

1. Crie o arquivo user.ldif e copie nele o seguinte conteúdo:
  - *dn: cn=”Marcos Lima”,ou=Usuarios,dc=anahuac,dc=org*
  - *uid: marcosl*
  - *cn: Marcos Lima*
  - *sn: MarcosL*
  - *objectClass: inetOrgPerson*
  - *objectClass: posixAccount*
  - *objectClass: shadowAccount*
  - *homeDirectory: /home/marcosl*
  - *loginShell: /bin/bash*
  - *uidNumber: 1000*
  - *gidNumber: 1000*
  - *userPassword: 123mudar*
2. Salve o arquivo;
3. Pare o OpenLDAP;



4. Execute o comando:

- `slapadd -l user.ldif`

*Testando a criação de objetos*

- Execute o comando “`slapcat`” para ver a saída. Os dois objetos devem existir.

### 3.1.4 *slappasswd*

Este comando é um utilitário para geração de senhas em “hash”. O seu uso é muito simples:

- `slappasswd -s sua_senha_aqui`

Sua utilidade é transformar senhas literais em senhas seguras em “hash”. Pode-se definir qual o “hash” desejado usando a opção “-h” e indicando o tipo. Os tipos de “hash” suportados pelo comando são: {CRYPT}, {MD5}, {SMD5}, {SSHA} e {SHA}.

Sim. É necessário escrever os {}. veja o exemplo:

- `slappaswd -h {MD5} -s sua_senha_aqui`

Se a opção -h não for utilizada o “hash” padrão é o SSHA que atualmente é um dos melhores.

No exemplo anterior tínhamos o atributo “userPassword” contendo uma senha em texto plano “123mudar”. Dessa forma funciona, mas a senha será armazenada em texto plano na base LDAP e isso é indesejável.

Podemos usar o comando, “`slappasswd -s 123mudar`”, para gerar um “hash” como este:

- {SSHA}B+3Ppz4uc/D722PwOFqrQIDF8v4ISYU5

Este “hash” deve ser colocado integralmente no atributo “userPassword”, assim a senha não mais será armazenada em texto plano. Devemos incluir, inclusive, a definição do “hash”: {SSHA}

### 3.1.5 *slapindex*

Este comando é utilizado para indexar a base LDAP seguindo às instruções definidas nas opções “olcDbIndex” do arquivo de configuração “olcDatabase={1}hdb.ldif”.

Maiores detalhes sobre a indexação da base LDAP serão vistos mais adiante.

## 3.2 *Série de comandos “ldap”*

Importante lembrar que estes comandos devem ser executados com a base OpenLDAP ativada.

Portanto o primeiro passo é ativar o OpenLDAP com o comando:

- `/etc/init.d/slaped start`

Por se tratarem de comandos que são executados em uma base LDAP ativa, algumas opções adicionais se farão sempre necessárias, como o “host” no qual se conectar e as credenciais de acesso.

As opções mais comuns são:

- `-h` – Host ao qual se vai conectar
- `-p` – Porta na qual se vai conectar
- `-x` – Autenticação não segura, ou seja, sem SASL
- `-ZZ` – Autenticação segura usando TLS - Transport Layer Security
- `-D` – BINDDN ou simplesmente o DN do usuário que será usado para autenticar
- `-w` – Senha do BINDDN

É certo que quase todos os comandos da série “ldap” irão usar, praticamente, todas essas opções.

### 3.2.1 *ldapsearch*

Este comando é utilizado para realizar buscas na base LDAP. Além das opções acima, este comando possui algumas opções próprias:

- *-b* – Define a partir de qual “galho” a pesquisa será feita
- *-s* – Define o escopo da pesquisa. As opções são: *base*, *one* e *sub*
  - ***base*** – Busca apenas na base DN
  - ***one*** – busca no mesmo nível do “galho” definido em *-b* e em mais um sub-nível;
  - ***sub*** – opção padrão, busca recursivamente a partir do “galho”
- *-LLL* – Mostra o resultado da pesquisa desabilitando comentários e informações extras

Exemplo:

```
ldapsearch -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -b ou=Usuarios,dc=anahuac,dc=org -LLL
```

### 3.2.2 *Aprendendo a sintaxe do “filter”*

O comando “ldapsearch” oferece, ainda, a possibilidade de refinamento da pesquisa. Pode-se definir um ou vários filtros para obter resultados mais precisos.

Para usá-los basta acrescentar ao final do comando, o filtro desejado e o(s) atributo(s) que deve(m) ser mostrados no resultado.

Vamos aos exemplos, para ilustrar:

1. ldapsearch -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -b ou=Usuarios, dc=anahuac, dc=org -LLL uid=marcosl

Perceba que nesta pesquisa não aparece o objeto “ou=Usuarios”

```
2. ldapsearch -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -b ou=Usuarios,dc=anahuac,dc=org -LLL uid=marcosl cn
```

Nesta pesquisa a busca foi feita pelo atributo “uid”, mas no final pedimos que fosse retornado o atributo “cn” e seu conteúdo.

Os filtros podem ser ainda mais elaborados utilizando algumas conjunções como o caracter \* para definir todos, ou inicia-se por “mar”, além dos operadores lógicos:

- “&” para condicional e;
- “|” para condicional ou;
- “!” para condicional não;

E operadores comparativos:

- “=” para igual;
- “~=” para igualdades aproximadas;
- “<=” para menor que;
- “>=” para maior que.

Vejamos alguns exemplos mais elaborados:

```
3. ldapsearch -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -b ou=Usuarios,dc=anahuac,dc=org -LLL ‘(&(uid=marcosl)(uidNumber=2000))’ cn
```

Esta pesquisa não deve retornar nada, afinal de contas o objeto que contém “uid=marcosl” tem “uidNumber” igual a 1000 e não a 2000.

```
4. ldapsearch -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -b ou=Usuarios,dc=anahuac,dc=org -LLL ‘(|(uid=marcosl)(uid=joao))’ cn
```

Esta pesquisa irá retornar todos os objetos que tenham “uid” igual a marcosl ou joao. Perceba o uso do operador lógico | na expressão do filtro.

```
5. ldapsearch -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -b ou=Usuarios,dc=anahuac,dc=org -LLL '(&(uid=marcosl)!(uidNumber=2000))' cn
```

Esta pesquisa retornará o CN do “uid=marcosl”, afinal de contas o usuário marcosl tem uidNumber igual a 1000 e o filtro pede que o uidNumber seja diferente de 2000.

Resumindo

Pode-se refinar muito a pesquisa utilizando os operadores do filtro.

Para simplificar a sintaxe assuma:

- OL – Operador lógico
- OC – Operador comparativo

Sintaxe básica:

```
'(OL(atributo1<OC>valor)(atributo2<OC>valor))'
```

Sintaxe complexa:

```
'(OL(atributo1<OC>valor)(OL(atributo2<OC>valor)))'
```

### 3.2.3 *ldapadd*

Este comando é similar ao *slapadd*, entretanto permite que objetos sejam adicionados à base LDAP com o serviço “no ar”.

O procedimento é o mesmo:

1. Crie o arquivo *group.ldif* e copie nele este conteúdo:

```
dn: ou=Grupos,dc=anahuac,dc=org
```

```
ou: Grupos
```

```
objectClass: organizationalUnit
```

```
objectClass: top
```

2. Salve o arquivo;
3. Pare o OpenLDAP;
4. Execute o comando:

```
ldapadd -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f group.ldif
```

*O sucesso ou fracasso na adição de um objeto dependerá, sempre, do permissionamento definido para o BINDDN. As ACL, que definem o permissionamento, serão estudadas, com detalhes, mais adiante.*

### 3.2.4 *ldapmodify*

Este comando permite fazer alterações nos atributos de um objeto já existente. Seu funcionamento é similar aos demais comandos, ou seja, ele utiliza um arquivo “ldif” para fazer as alterações desejadas.

Apesar de usar um arquivo “ldif”, o seu formato é levemente diferente, permitindo que os parâmetros de alteração sejam definidos.

A opção extra para indicar qual é o arquivo “ldif” a ser usado é: -f

Estão disponíveis três formas de alteração:

- add
- replace
- delete

Vejam como esses arquivos “ldif” de modificação são escritos:

- *dn: completo*
- *ação: atributo*
- *atributo: valor*

*Vamos fazer alterações em nosso usuário Marcos Lima, adicionando um novo atributo, chamado “title”, alterando o valor do atributo “loginShell” e finalmente, removendo o atributo “title”.*

#### 1. Adicionando um atributo

a) Crie o arquivo alter\_user.ldif com o seguinte conteúdo:

```
dn: cn="Marcos Lima",ou=Usuarios,dc=anahuac,dc=org
add: title
title: Professor
```

b) Execute o comando:

```
ldapmodify -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f alter_user.ldif
```

2. Alterando um atributo

a) Crie o arquivo alter\_user2.ldif com o seguinte conteúdo:

```
dn: cn="Marcos Lima",ou=Usuarios,dc=anahuac,dc=org
```

```
replace: loginShell
```

```
loginShell: /bin/false
```

b) Execute o comando:

```
ldapmodify -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f alter_user2.ldif
```

3. Removendo um atributo

a) Crie o arquivo alter\_user3.ldif com o seguinte conteúdo:

```
dn: cn="Marcos Lima",ou=Usuarios,dc=anahuac,dc=org
```

```
delete: title
```

b) Execute o comando:

```
ldapmodify -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f alter_user3.ldif
```

Visando facilitar múltiplas alterações, pode-se escrever um “ldif” de modificação contendo todas elas e executando-as de uma única vez. Para isso basta separar as ações por um hífen. Veja abaixo o exemplo:

a) Crie o arquivo alter\_user4.ldif com o seguinte conteúdo:

```
dn: cn="Marcos Lima",ou=Usuarios,dc=anahuac,dc=org
```

```
add: title
```

```
title: Professor
```

```
-replace: loginShell
```

```
loginShell: /bin/bash
```

```
-delete: title
```

b) Execute o comando:

```
ldapmodify -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f alter_user4.ldif
```

### 3.2.5 *ldapdelete*

Este comando permite remover um objeto especificando o seu DN completo. Veja o exemplo:

```
ldapdelete -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha cn="Marcos Lima",ou=Usuarios,dc=anahuac,dc=org
```

O resultado será a completa remoção do usuário “Marcos Lima”.

### 3.2.6 *ldapmodrdn*

Este comando permite que um DN seja renomeado. Até o momento vimos diversas opções que permitem alterações em atributos de um objeto, mas não do próprio objeto, ou seja, do seu DN.

Este comando permite essa alteração.

A opção extra e que deve ser sempre usada é “-r”. Ela evita que o novo DN modificado tenha dois registros CN.

Vejamos como ele funciona:

1. Crie o arquivo alter\_user5.ldif com o seguinte conteúdo:  
cn="Marcos Lima",ou=Usuarios,dc=anahuac,dc=org  
cn="Marcos Lima da Silva"

Perceba que a segunda linha contém apenas o valor para o qual se quer modificar o DN e não um DN completo.

2. Execute o seguinte comando:

```
ldapmodrdn -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -r -f alter_user5.ldif
```





CAPÍTULO 4

CONHECENDO  
ALGUNS CLIENTES LDAP



Existem diversos sistemas que auxiliam na administração de bases LDAP. Mostraremos aqui alguns deles: o “PhpLdapAdmin”, aplicativo Web, o “Luma”, uma interface local e o “ldapvi” que é um cliente via terminal para manipulação de bases OpenLDAP.

## 4.1 *PhpLdapAdmin*

Instalação do phpLDAPAdmin

1. Instale o phpLDAPAdmin com o comando:

```
aptitude install phpldapadmin
```

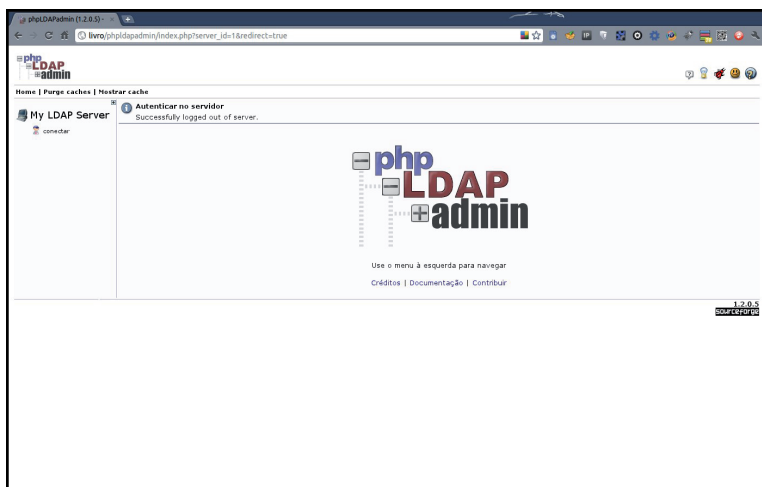
2. Após a instalação do PhpLdapAdmin, será necessário fazer algumas alterações no seu arquivo de configuração para que o seu uso seja mais simples. Para tanto edite o arquivo “/etc/phpldapadmin/config.php”, e altere as opções abaixo, assim:

```
$servers->setValue('server',base,array('dc=anahuac,dc=org'));  
$servers->setValue('login',bind_id,'cn=admin,  
dc=anahuac,dc=org');;w
```

3. Abra seu browser favorito e acesse o endereço IP da sua máquina seguido do nome PhpLdapAdmin:

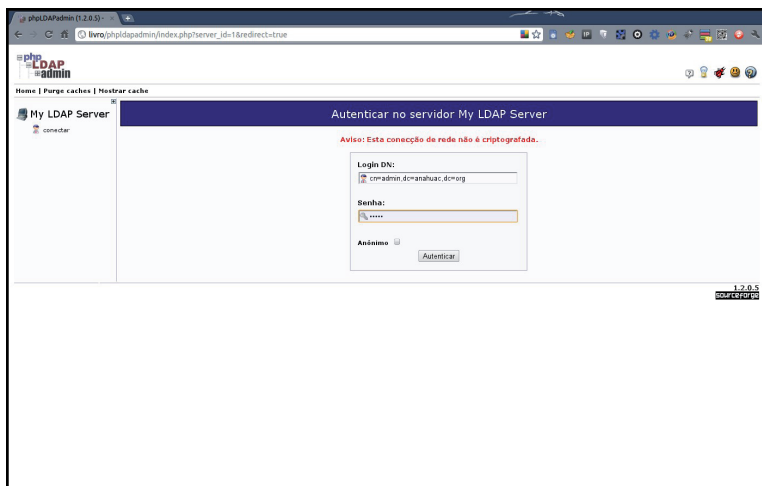
```
http://ldap__.anahuac.org/phpldapadmin
```

A tela inicial do PhpLdapAdmin deverá aparecer, como mostrada na figura abaixo

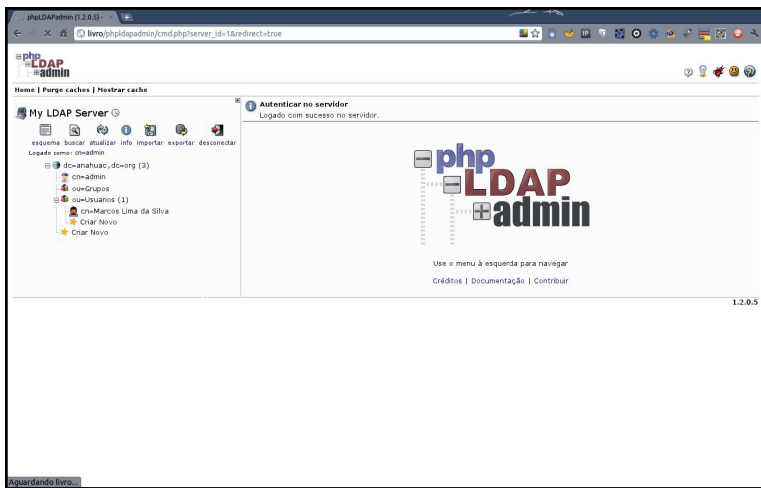


Ao clicar com o mouse sobre o link “login”, aparecerá a tela para efetuar o logon.

Em seguida, entre com a senha do “admin” e faça o logon, como mostrado na figura abaixo



Uma vez autenticado, você verá, do lado esquerdo da tela a base de dados apresentada em forma de árvore, como mostrado na figura abaixo.



## 4.2 Luma

### Instalação do Luma

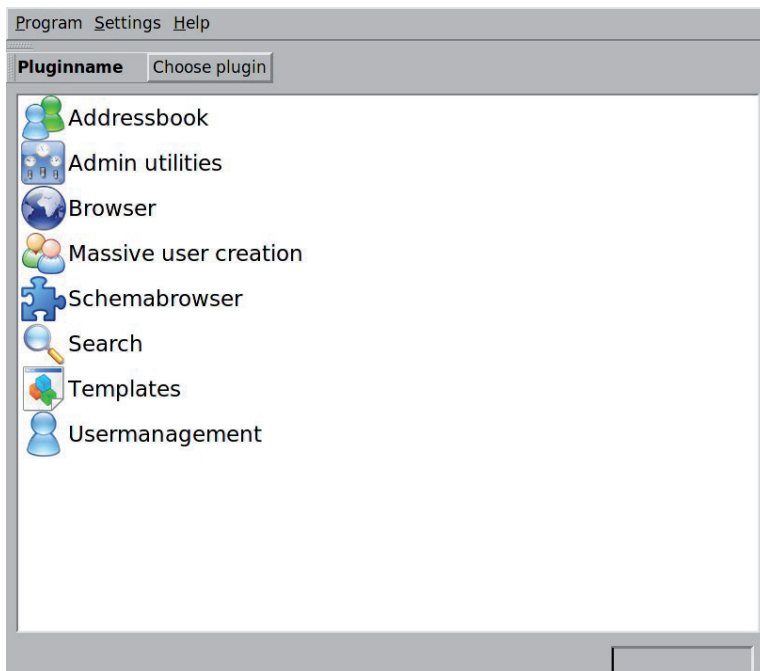
1. Instale o Luma com o comando:

```
aptitude install luma
```

2. Após a instalação do Luma, abra um terminal e execute o comando:

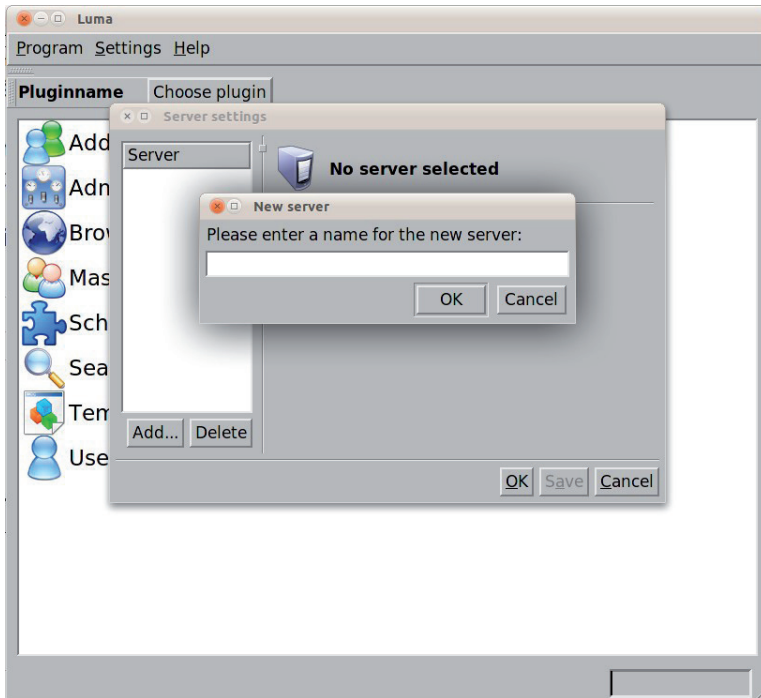
```
luma
```

A tela inicial do Luma deverá aparecer, como mostrada na figura abaixo



No menu de texto, na parte de cima da tela, selecione:  
Settings -> Edit Server List...

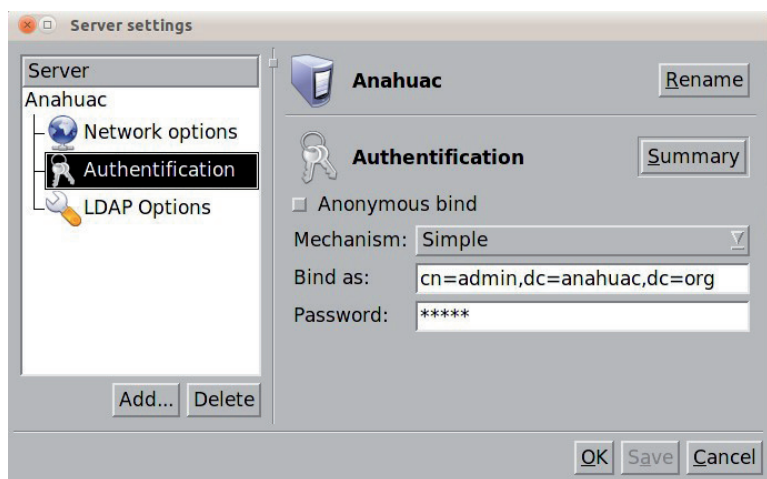
Surgirá, então a seguinte tela:



Um recurso interessante do Luma é que ele permite a conexão a diversos servidores diferentes de forma simultânea, bastando registrar cada um dos servidores através desta interface.

No lado inferior esquerdo clique no botão “Add...” para adicionar um novo servidor no qual o Luma irá se conectar. Defina um nome para o servidor. Esse nome é livre e fica a seu critério.

Depois de clicar em OK, clique sobre o nome do seu servidor e surgirá a a seguinte tela:



Clique o na opção “Network options” e preencha a opção “hostname” com o endereço IP do servidor LDAP ao qual o Luma deve se conectar.

Em seguida clique na opção “Authentication” e desmarque a opção “Anonymous bind”. Feito isso surgirão os campos para definir o usuário e senha de conexão.

No campo “Bind ad” coloque o DN completo do usuário “admin”, ou seja: cn=admin,dc=anahuac,dc=org

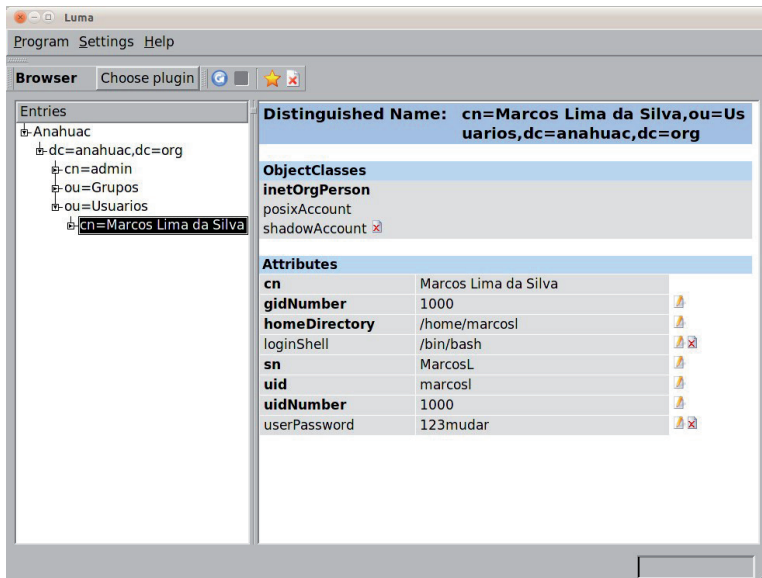
E no campo “Passowrd” defina a senha.

Para finalizar clique no botão “OK”. Assim você será levado à tela inicial do Luma.

Para visualizar a sua base clique no ícone do “Planeta Terra”, chamado Browser.



Veja a imagem abaixo:



O Luma oferece uma série de opções de gestão utilizando o botão direito do mouse sobre cada objeto. Vale a pena testar e brincar com as opções.

### 4.3 *ldapvi*

O “ldapvi” é um programa que faz pesquisas em sua base LDAP e envia o resultado para o seu editor favorito. Por princípio ele se parece muito com o comando “ldapsearch” mas como o resultado é redirecionado para um editor, pode-se fazer as alterações desejadas e no fim basta salvá-las normalmente para que elas tenham efeito.

O editor a ser utilizado é definida pela variável de ambiente “EDITOR”. Portanto para definir seu editor preferido basta exportar a variável com o caminho completo dele. Algo como: “export EDITOR=/usr/bin/nano” fará com que o “ldapvi” use o “nano” como editor.

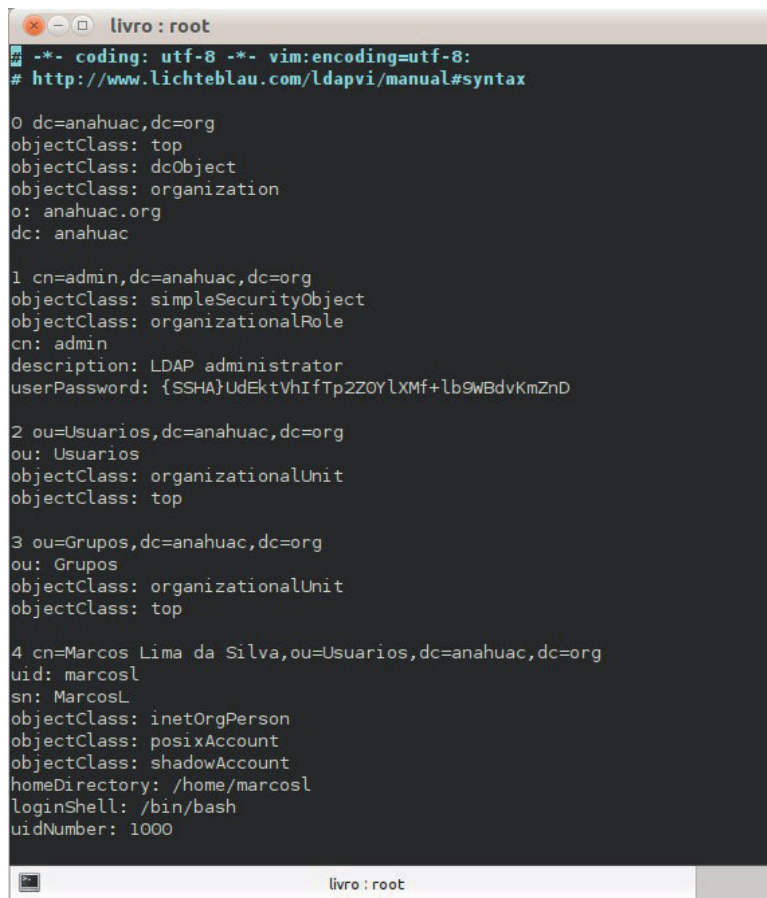
Este lindo cliente LDAP está disponível nos repositórios do Debian e pode ser instalado utilizando o comando abaixo:

```
aptitude install ldapvi
```

Prático, leve e simples de usar. Vejamos um exemplo de conexão usando o “ldapvi”

1. Abra um terminal
2. Execute o seguinte comando:  
ldapvi -h localhost -D cn=admin,dc=anahuac,dc=org -w senha -b dc=anahuac,dc=org

Perceba que a sintaxe do comando é similar à sintaxe usada pelos comandos “ldap”.



```
livro : root
# -*- coding: utf-8 -*- vim:encoding=utf-8:
# http://www.lichteblau.com/ldapvi/manual#syntax

0 dc=anahuac,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
o: anahuac.org
dc: anahuac

1 cn=admin,dc=anahuac,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword: {SSHA}UdEktVhIfTp2Z0YlXMf+lb9WBdvKmZnD

2 ou=Usuarios,dc=anahuac,dc=org
ou: Usuarios
objectClass: organizationalUnit
objectClass: top

3 ou=Grupos,dc=anahuac,dc=org
ou: Grupos
objectClass: organizationalUnit
objectClass: top

4 cn=Marcos Lima da Silva,ou=Usuarios,dc=anahuac,dc=org
uid: marcosl
sn: MarcosL
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
homeDirectory: /home/marcosl
loginShell: /bin/bash
uidNumber: 1000

livro : root
```

Uma vez conectado, basta fazer suas alterações da mesma forma que se edita um texto comum. Uma vez salvo as alterações serão realizadas na base LDAP.



CAPÍTULO 5

GERENCIANDO LOGS



O OpenLDAP é um programa extremamente flexível no que diz respeito ao seu registro de atividades. Ele possui diversos níveis de log, inclusive organizados por tipo de análise.

Por padrão o OpenLDAP vem com o registro de atividades desativado. O motivo é que o modo genérico gera informações demais e poderia causar problemas de espaço em disco.

A seguir veremos como lidar com os logs em um servidor OpenLDAP.

### 5.1 Loglevel

A opção “`olcLogLevel`” determina o nível de registros de atividades desejado. O OpenLDAP tem 16 níveis diferentes que são definidos por uma identificação numérica. Veja a tabela abaixo:

| Número | Nome   | Descrição                                               |
|--------|--------|---------------------------------------------------------|
| -1     | all    | all                                                     |
| 1      | trace  | trace function calls                                    |
| 2      | packet | debug packet handling                                   |
| 4      | args   | heavy trace debugging (function args)                   |
| 8      | conns  | connection management                                   |
| 16     | BER    | print out packets sent and received                     |
| 32     | filter | search filter processing                                |
| 64     | config | configuration file processing                           |
| 128    | ACL    | access control list processing                          |
| 256    | stats  | stats log connections/operations/results                |
| 512    | stats2 | stats log entries sent                                  |
| 1024   | shell  | print communication with shell backends                 |
| 2048   | parse  | entry parsing                                           |
| 4096   | cache  | caching (unused)                                        |
| 8192   | index  | data indexing (unused)                                  |
| 16384  | sync   | LDAPSync replication                                    |
| 32768  | none   | only messages that get logged whatever log level is set |

A separação de níveis de log por números auxilia muito no momento de tentar identificar um problema ou simplesmente monitorar uma determinada atividade.

Vejam alguns exemplos de registro de atividades:

1. Abra um terminal;
2. Pare seu servidor OpenLDAP - `/etc/init.d/slapd stop`;
3. Execute o seguinte comando:

```
slapd -d 256
```

Perceba que o shell fica “preso” imprimindo na tela os registros de atividades

4. Abra um segundo terminal e efetue uma pesquisa qualquer;
5. Verifique no primeiro terminal o resultado;
6. Para “liberar” o terminal, pressione as teclas “Ctrl+c”;
7. Inicie o seu servidor OpenLDAP - `/etc/init.d/slapd start`

Executar o comando “`slapd -d 256`” pode alterar as permissões dos arquivos da base OpenLDAP, causando problemas em sua inicialização posterior. Para evitar esse erro pode-se executar o comando indicando o usuário e grupo com as opções “-u” e “-g”. Assim:

```
slapd -d 256 -u openldap -g openldap
```

Em um primeiro momento o resultado impresso pode parecer confuso, mas é uma questão de hábito para conseguir ler e identificar possíveis problemas.

## 5.2 Syslog

Em alguns casos pode ser desejável armazenar os registros de atividades do OpenLDAP em um arquivo. É importante ter claro que o OpenLDAP é capaz de gerar quantidades imensas de informação, especialmente no nível -1.

Para ativar o armazenamento dos registros de log basta alterar a opção “`olcLogLevel`” na base de configuração `cn=config`.

O registro de atividades será feito no arquivo de log padrão do sistema. No caso do Debian `/var/log/syslog`.

Se for necessário indicar um arquivo exclusivo para armazenar os registros de atividades do OpenLDAP pode-se configurar o serviço “syslog” para fazer isso.

1. Edite o arquivo `/etc/rsyslog.conf` e insira o seguinte conteúdo:  
`local4.* /var/log/slapd.log`

2. Reinicie o “syslog” com o comando:  
`/etc/init.d/syslogd restart`

3. Agora é hora de alterar o OpenLDAP para iniciar o registro de logs. Para isso vamos criar um arquivo chamado “loglevel.ldif” com o seguinte conteúdo:

```
dn: cn=config
replace: olcLogLevel
olcLogLevel: 256
```

4. Execute o comando abaixo:

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f loglevel.ldif
```

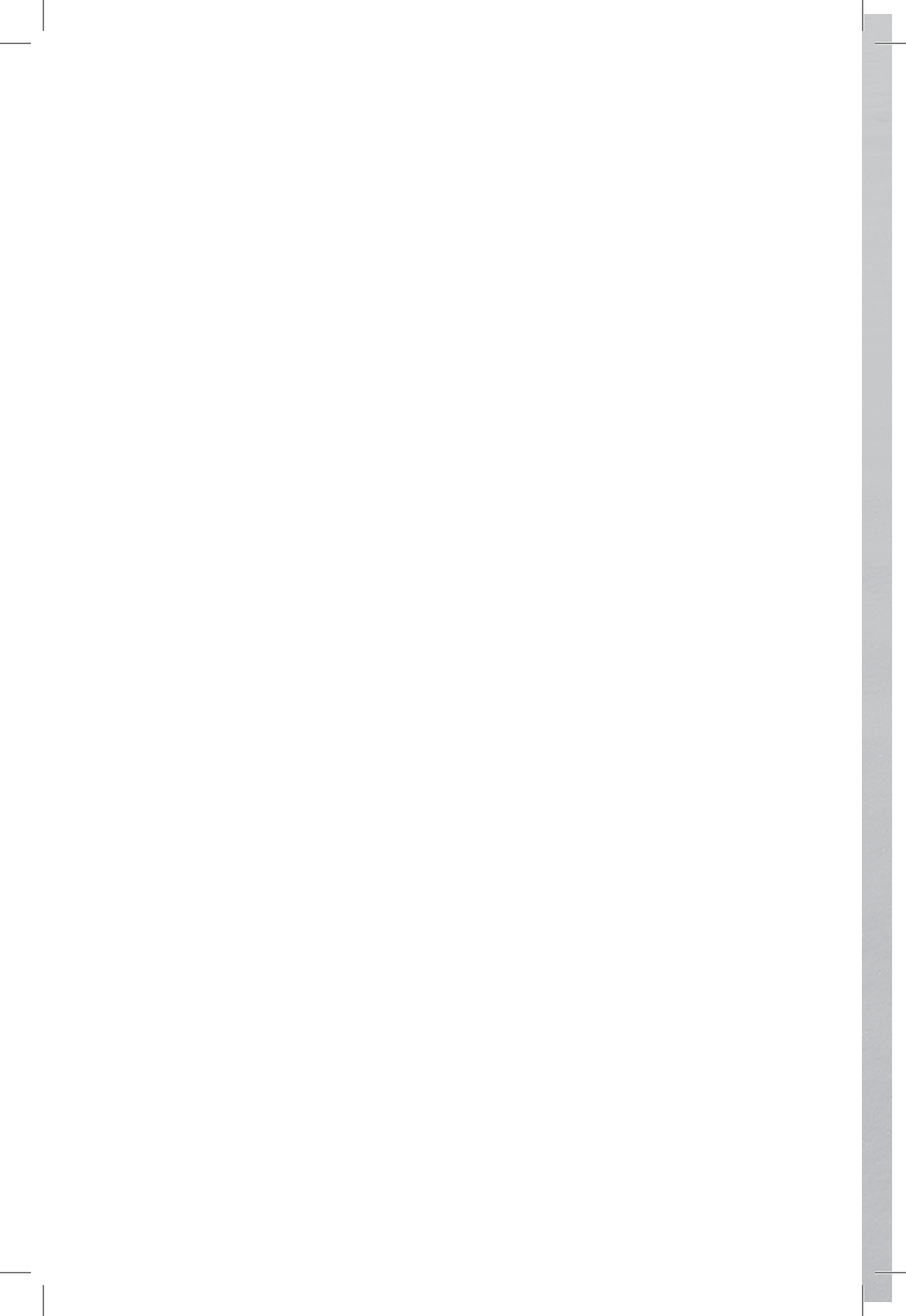
5. Faça uma pesquisa qualquer na base e veja a saída no arquivo de logs do sistema:

```
tail -f /var/log/syslog
```

Perceba que não foi necessário reiniciar o OpenLDAP para que a alteração no nível de registro de atividades entrasse em vigor. Essa é a grande vantagem do novo método de configuração.

6. Para retornar à situação original, altere o arquivo “loglevel.ldif”, definindo o valor do atributo “olcLogLevel” para “none” e depois execute o comando abaixo:

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f loglevel.ldif
```







CAPÍTULO 6

TRABALHANDO COM ÍNDICES



Os índices tem por objetivo melhorar o desempenho das consultas feitas à base LDAP. Quando uma base não está devidamente indexada a busca será sempre feita em toda a base, pois não há índice de consulta. Esse é um recurso comum, amplamente usado em bases de dados relacionais tipo SQL.

Imagine que quanto maior a base de dados maior será o tempo de uma busca. Então, podemos afirmar que gerar índices melhora a performance do nosso servidor? Em geral sim. Mas o que não podemos esquecer é que cada índice requer tempo para ser mantido e usa memória adicional.

Portanto é recomendável criar índices apenas para atributos que são usados frequentemente em consultas. Indexar atributos desnecessários pode causar perda de performance.

A indexação em uma base LDAP é feita indicando quais atributos devem ser indexados e qual tipo de indexação deve ser usada. Abaixo explicamos os tipos disponíveis.

#### Tipos de indexação

- **pres** – de “*present*”. Deve ser utilizado para consultas do tipo ‘*objectclass=person*’ ou ‘*attribute=mail*’. Indicado para buscas de existência.
- **eq** – de “*equal*”. Deve ser utilizado para consultas do tipo ‘*uid=marcosl*’, ou seja, para buscas por valores completos.
- **sub** – de “*substring*”. Deve ser para consultas do tipo ‘*uid=marcos\**’, ou seja, para buscas por partes
- Em termos gerais os tipos “pres” e “eq” se equivalem na maioria dos casos, isso porque o tipo “eq” também pode confirmar a existência ou não de um determinado valor. Assim sendo o tipo “eq” é mais amplo em seu escopo de atuação e portanto mais utilizado.

Os tipos de indexação podem ser combinados para atender situações onde as pesquisas mais realizadas são de dois ou mais tipos diferentes.

Perceba que a indexação é uma ação feita nos dados contidos na base de dados que contém informações, ou seja, na ár-

vore de dados propriamente dita, e não na base de configuração “cn=config”.

## 6.1 Indexando

Para fazer a indexação de sua base LDAP edite o arquivo de configuração vamos criar um arquivo “ldif” para esse fim, adicionando o atributo “olcDbIndex” com seu tipo correspondente, para cada atributo que será indexado.

Para indexar siga os passos:

1. Crie um arquivo chamado “index.ldif” com os seguinte conteúdo:

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uid,userPassword,cn pres,eq
```

Neste caso não só colocamos a lista de atributos, mas também a lista de tipos de indexação.

2. Salve o arquivo;
3. Execute o comando:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f index.ldif
```

4. Perceba que os índices foram criados em “/var/lib/ldap/”

Recomendamos que a indexação seja feita de tempos em tempos, à medida que sua base de dados for crescendo.

## 6.2 Cuidando do permissionamento

No Gnu/Linux Debian o OpenLDAP é executado pelo usuário “openldap”. Entretanto a indexação é executada com o usuário “root”. Isso faz com que os índices sejam feitos com permissionamento para o usuário “root”.

Essa alteração de permissionamento nos arquivos de índices pode provocar erro ao se iniciar o OpenLDAP. Por isso o comando “slapindex”, ao terminar a indexação, emite esta mensagem:

WARNING!

Runnig as root!

There's a fair chance slapd will fail to start.

Check file permissions!

Ela serve para alertar sobre o problema de permissionamento. O que deve ser feito para resolver o problema é corrigir o permissionamento com o comando abaixo:

```
chown openldap: /var/lib/ldap/*
```





CAPÍTULO 7

**ACL – ACCESS CONTROL LIST**





As diretivas de controle de acesso, como seu próprio nome sugere, definem o nível de acesso que cada usuário terá para cada tipo de informação disponível na base de dados. No OpenLDAP elas representam um ferramenta muito poderosa e versátil. Utilizando ACLs é possível controlar o acesso à base de dados, granularmente, baseando-se nas informações de autenticação, endereço IP, nome de domínio, dentre outras.

É importante entender que as ACLs são cheçadas na ordem em que são descritas. Portanto a ordem faz muita diferença. E essa ordem deve ser estabelecida indo do critério mais rigoroso para o mais complacente. Isso porque se uma regra complacente vier primeiro, ela poderá garantir um acesso indevido. Esse mesmo raciocínio se aplica a todos os softwares que usam ACLs, entre eles o “firewall” iptables e o servidor proxy “Squid”.

A primeira noção é que ACL são poderosas para controlar com mãos de ferro o que os usuários podem ou não fazer dentro de uma infraestrutura OpenLDAP. Apesar desta ser uma das funções mais importantes, o aspecto mais interessante do uso de ACL em servidores LDAP é outro: a possibilidade de delegar tarefas.

Imagine o seguinte caso, válido tanto para sistemas que guardam dados em arquivos texto quanto para sistemas gerenciadores de bancos de dados: como permitir que um usuário altere a sua e apenas a sua senha?

Na maioria das vezes uma instrução como essa exige malabarismos enormes do administrador de sistemas, principalmente quando o grau de integração entre os serviços aumenta. Entretanto o OpenLDAP torna essa tarefa bem simples.

## 7.1 Sintaxe

A sintaxe básica das ACL's é a seguinte:

```
{X}to access to <o que> [ by <quem> [ <acesso> ]  
[ <controle> ] ]+
```

Onde:

- **o que** – atributo(s) para o qual o controle será imposto;
- **quem** – entidade ou DN completo do usuário (veja tabela de entidades a baixo);
- **acesso** – define o nível de acesso à entidade ou ao usuário;
- **controle** – define a forma de processamento das ACL's. Por padrão quando uma entrada de ACL é satisfeita o processamento das ACL's é interrompido. Isto se dá, porque o valor de controle padrão é "stop". Outros valores possíveis são: "continue", que faz com que o processamento continue e, "break".

## 7.2 Tabela de Entidades

| Entidade  | Descrição                                       |
|-----------|-------------------------------------------------|
| *         | Todos                                           |
| anonymous | usuários não autenticados. Anônimos.            |
| users     | usuário autenticado                             |
| self      | usuário dono do objeto e/ou atributo controlado |

## 7.3 Tabela dos níveis de acesso

| Nível   | Privilégio | Descrição                                    |
|---------|------------|----------------------------------------------|
| none    | =0         | sem acesso                                   |
| auth    | =x         | necessário para autenticação                 |
| compare | =cx        | necessário para efetuar comparações          |
| search  | =scx       | necessário para utilizar filtros de pesquisa |
| read    | =rscx      | necessário para ler resultados de pesquisas  |
| write   | =wrscx     | necessário para modificar ou renomear        |

## 7.4 Explicando o formato *cn=config*

No formato “cn=config” de configuração do OpenLDAP, as ACL’s são armazenadas no atributo “olcAccess”. Podem haver quantos atributos “olcAccess” sejam necessárias para atender a necessidade de permissionamento a base.

Antes de descrever as restrições propriamente ditas, queremos chamamos a atenção para o prefixo “{X}to”, que estará presente em todas as “linhas” de ACL. Esse prefixo estabelece a ordem na qual as ACL’s serão lidas e implementadas pelo OpenLDAP. A variável “X” será substituída por um número, iniciando por “0”.

É importante perceber, também, que as ACL’s são aplicadas sobre a base em si, portanto o arquivo no qual essas diretrizes de encontram é “olcDatabase={1}hdb.ldif”.

Porque é importante entender em qual arquivo estão os parâmetros que queremos usar? Para poder escrever os arquivos “ldif” de forma correta, ou usar nosso cliente OpenLDAP preferido, da forma correta.

## 7.5 Exemplos

### Exemplo 1

```
olcAccess: {0}to attrs=userPassword,shadowLastChange by
self write by anonymous auth by dn="cn=admin,
dc=anahuac,dc=org" write by * none
```

Obedecendo à sintaxe podemos “quebrar”, a linha acima, assim:

1. Atributo: “olcAccess”;
2. Prefixo: {0}to – indica que esta será a primeira ACL a ser utilizada;
3. attrs – Define os atributos que serão controlados. A lista é separada por vírgulas;
4. by self write – Define que os usuários somente terão acesso de escrita aos atributos “userPassword” e “shadowLastChange” deles mesmos, ou seja, dos quais sejam donos;

5. `by anonymous auth` – Define que o acesso só será aceito, mediante autenticação;
6. `by dn="cn=admin,dc=anahuac,dc=org" write` – Define o usuário que tem acesso e em seguida que tipo de acesso;
7. `by * none` – Define que todos (\*) não tem acesso. Perceba que está última diretiva visa impedir quaisquer outros acesso que não estiverem dentro das categorias descritas acima dela.

### Exemplo 2

```
olcAccess: {1}to dn.base="" by * read
```

Obedecendo à sintaxe podemos “quebrar”, a linha acima, assim:

1. Atributo: “olcAccess”;
2. Prefixo: {1}to – indica que esta será a segunda ACL a ser utilizada;
3. `dn.base="" by * read` – Indica que qualquer usuário, inclusive anônimos, podem ler a raiz da base LDAP. Essa permissão é crucial para diversos clientes.

Perceba que no “dn.base”, “base” significa somente a raiz, assim como na opção “-s” do comando `ldapsearch`, explicada antes neste manual.

### Exemplo 3

```
olcAccess: {2}to * by self write by dn="cn=admin,  
dc=anahuac,dc=org" write by * read
```

Obedecendo à sintaxe podemos “quebrar”, a linha acima, assim:

1. Atributo: “olcAccess”;
2. Prefixo: {2}to – indica que esta será a terceira ACL a ser utilizada;

3. \* – Permite acesso a todos os usuários, inclusive anônimos;
4. by self write – Define que o usuário pode escrever em qualquer um dos atributos pertencentes ao seu objeto;
5. by dn="cn=admin,dc=anahuac,dc=org" write – Define que o usuário (através de seu DN completo) “admin” pode escrever na base LDAP, ou seja, tem plenos poderes;
6. by \* read – Define que todos os demais usuários podem, apenas, ler o conteúdo da base;

Estes três exemplos são os valores definidos por padrão na instalação do OpenLDAP em um servidor Gnu/Linux Debian.

É importante perceber que essas três ACLs se comportam como uma, ou seja, apesar da última permitir que qualquer usuário possa ler qualquer atributo a primeira não. Por isso é fundamental dar toda a atenção possível à ordem das ACLs.





CAPÍTULO 8

BACKUPS E RESTAURAÇÃO





Tão importante como manter o servidor no ar e bem configurado, é definir uma política de backups adequada. Sem falar na verificação do procedimento de restore, algumas vezes não verificado.

Vamos apresentar os dois métodos de backup mais seguros:

### ***8.1 Usando ldapsearch***

A melhor forma de fazer o backup de uma base OpenLDAP é utilizando o comando “ldapsearch”. Isso mesmo, utiliza-se o comando de pesquisa para gerar um arquivo “ldif” que, posteriormente, poderá ser utilizado para restaurar a base.

Vamos também usar a opção -LLL para reduzir o número de comentários.

```
Para fazer um backup completo utilize o seguinte comando:  
ldapsearch -x -D cn=admin,dc=anahuac,  
dc=org -w senha -b dc=anahuac,dc=org -LLL > backup.ldif
```

O resultado do comando será escrito no arquivo backup.ldif

### ***8.2 Usando slapcat***

Apesar de não recomendado pode-se usar o comando “slapcat” para fazer o backup. A única vantagem de usar este comando é a não necessidade de fornecer a senha do administrador.

```
Para fazer um backup completo utilize o seguinte comando:  
slapcat > backup.ldif
```

O resultado do comando será escrito no arquivo backup.ldif

### ***8.3 Salvando o cn=config***

Agora que as configurações do OpenLDAP são armazenadas em uma base pode-se optar por fazer o backup de duas maneiras: copiando o diretório “slapd.d” ou fazendo um “dump” da base “cn=config” para um arquivo “ldif” que poderá ser usado, no futuro, para consultar as configurações.

A cópia do diretório “slapd.d” é tão simples quanto o uso do comando “cp”, portanto vamos nos concentrar no método mais trabalhoso. Para fazer o backup da base “cn=config” use o comando abaixo:

```
ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config > cn_config.ldif
```

O conteúdo completo da base de configuração, incluindo os schemas, será inserido nesse arquivo. Mais adiante explicaremos como fazer o “restore” da base “cn=config”.

Nossa sugestão é copiar o diretório “slapd.d” pois há problemas na restauração do “ldif” gerado acima. Mas vamos deixar para discutir esses problemas na parte do restore.

## ***8.4 Exemplo de script***

Para automatizar o processo de backup aqui mostramos um exemplo de “script” para esse fim.

1. Instale o pacote “nail” com o comando:

```
aptitude install nail
```

2. Crie um arquivo chamado “ldap\_backup.sh”, com o seguinte conteúdo:

```
#!/bin/bash
```

```
# Este script tem por objetivo fazer o backup de uma base LDAP e enviá-la por e-mail para o administrador da rede.
```

```
# Copie este script para o diretório /etc/cron.daily para que seja executado todo dia às 04:00 da madrugada.
```

```
# Defina a variável abaixo:
```

```
ADMIN_MAIL=""
```

```
TODAY=`date +%Y%m%d`
```

```
# Este comando coleta toda a base LDAP e a escreve em um arquivo com a data de hoje
```

```
ldapsearch -x -D cn=admin,dc=anahuac,dc=org -w nuka -b dc=anahuac,dc=org -LLL > /tmp/$TODAY-backup.ldif
```

```

# Este comando copia o diretório “slapd.d” e a escreve em um
diretório com a data de hoje
cp -a /etc/ldap/slapd.d/ /tmp/$TODAY-slapd.d
# Mudando de diretório
cd /tmp
# Compactando o arquivo com o comando tar
tar -zcvf $TODAY-backup.tar.gz $TODAY-backup.ldif
$TODAY-slapd.d
# Enviando por e-mail
echo “Backup da base LDAP” | nail -s “Backup de $TODAY”
-a /tmp/$TODAY-backup.tar.gz $ADMIN_MAIL

```

## 8.5 Restaurando o backup

Restaurar o backup, apesar de simples, pode ser um procedimento penoso para usuários inexperientes. Aqui vamos descrever o passo a passo para recuperar a base de configuração “cn=config” e a base LDAP, propriamente dita.

### 8.5.1 Restaurando o cn=config

Se for necessário restaurar a base de configuração “cn=config”, sugerimos que seja feita a cópia do diretório “slapd.d” a partir do backup. Essa é a forma mais segura. Siga os passos abaixo:

1. Pare o OpenLDAP

```

/etc/init.d/slapd stop

```
2. Faça uma cópia do diretório original do “slapd.d”:

```

cd /etc/ldap
cp -a slapd.d slapd.d.orig

```
3. Remova o diretório original “slapd.d”:

```

rm slapd.d -Rf

```
4. Restaure o “slapd.d” do backup:

```

cp /backup/slapd.d /etc/ldap/ -Rf

```

5. Corrija os permissionamentos:  
`chown openldap: /etc/ldap/slapd.d -R`

### **8.5.2 Restaurando a base**

A restauração da base em si, requer que a corrompida ou indesejada seja removida, deixar o OpenLDAP criar uma nova base vazia, para então restaurar nossos dados. Depois disso será necessário corrigir os permissionamentos e finalmente recolocar o serviço no ar.

Siga os passos abaixo:

1. Pare o OpenLDAP:  
`/etc/init.d/slapd stop`
2. Remova o conteúdo do diretório `/var/lib/ldap` com o comando:  
`rm /var/lib/ldap/*`
3. Ative o OpenLDAP;  
`/etc/init.d/slapd start`
4. Pare o OpenLDAP;  
`/etc/init.d/slapd stop`
5. Restaure a base LDAP com o comando:  
`slapadd -l /caminho/completo/arquivo_de_backup.ldif`
6. Restaure as permissões com o comando:  
`chown openldap: /var/lib/ldap/*`
7. Ative o OpenLDAP  
`/etc/init.d/slapd start`

### **8.6 Explicando**

1. Se sua base está corrompida ou não confiável, então será necessário removê-la para restaurar o backup. A forma mais

eficiente de remover a base LDAP é apagando os arquivos físicos da base, que no Gnu/Linux Debian estão localizados em `/var/lib/ldap`.

2. Para que seja possível restaurar a base o OpenLDAP precisa ser reiniciado sem nenhum arquivo, assim ele gera os arquivos DBD iniciais, mas vazios.
3. Para poder restaurar a base LDAP com o comando “`slapd`” o OpenLDAP precisa estar parado.

### *8.7 Usando db\_recovery*

“NUNCA PRECISAREI DE UMA FERRAMENTA DE RECONSTRUÇÃO”

A frase, se seguida de todas as boas práticas de backup e administração, pode ter sentido. Se as rotinas de cópia de segurança e de verificação de erros estão sendo realizadas de forma periódica e correta, geralmente é muito mais simples recuperar os dados a partir de uma cópia de segurança do que recuperar os dados corrompidos.

Como a maioria dos mortais está sujeita a cometer erros, dentre os quais esquecer uma vez ou outra de criar as cópias de segurança, existe a necessidade eventual de recuperar a base de dados corrompida.

Para as bases baseadas em Berkeley DB, deve-se usar a ferramenta `db_recover`. O nome exato desta ferramenta variará de acordo com a versão do BDB.

O utilitário `db_recover` deve ser executado após uma falha de aplicação, de sistema ou para restaurar a base de dados para um estado consistente.

A linha de comando abaixo executa o utilitário `db_recover` em modo de recuperação catastrófica e com saída prolixa. O diretório onde ele será executado é aquele que consta no arquivo de configuração do servidor.

Para executar a recuperação de dados utilize o comando abaixo:

```
db_recover -c -v -h /var/lib/ldap
```

## 8.8 Recuperando a senha do usuário Admin

A senha do administrador é armazenada em dois lugares: na base de configuração “cn=config” e na base de dados propriamente dita. Quem faz isso é o configurador do pacote do Debian, quando ele popula a base OpenLDAP, após a instalação.

Caso você esqueça a senha do administrador de uma de suas bases de dados, é possível alterá-la diretamente na base de dados de configuração do “slapd”. Entretanto seja necessário alterar a senha na base também, se não pode-se terminar com duas senhas em funcionamento, o que seria uma situação indesejada e insegura.

O procedimento forçará primeiro a troca da senha na base de configuração “cn=config” para, em seguida, poder fazer a troca da senha do usuário “admin” na base de dados propriamente dita.

Os passos abaixo indicam o procedimento que deve ser seguido para alterar a senha do usuário “admin”. Siga-os:

1. Execute o comando abaixo para gerar o “hash” da senha:

```
slappasswd -s novasenha
```

2. Crie um arquivo chamado “admin\_pwd\_cn\_config.ldif” com o seguinte conteúdo abaixo, trocando o valor do atributo “olcRootPW” pelo resultado do comando acima:

```
dn: olcDatabase={1}hdb,cn=config
```

```
changetype: modify
```

```
replace: olcRootW
```

```
olcRootPW:{SSHA}QSFxMdlk4cTRr/45k9+q14SW9hn7mMnG
```

3. Execute o comando “ldapadd”, com a sintaxe abaixo, para aplicar a mudança descrita no arquivo “ldif”:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f admin_pwd_cn_config.ldif
```

Neste exato momento, seu OpenLDAP está com duas senhas funcionais para o usuário “admin”. Tanto a senha “perdida”,

quanto a nova valem. Se desejar, faça o teste, executando o comando “ldapsearch” com autenticação e utilize as duas senhas.

4. Crie um segundo arquivo chamado “admin\_pwd\_base.ldif” com o conteúdo abaixo:

```
dn: cn=admin,dc=anahuac,dc=org
```

```
changetype: modify
```

```
replace: userPassword
```

```
userPassword: {SSHA}QSFxMdIk4cTRr/45k9+q14SW9hn-7mMnG
```

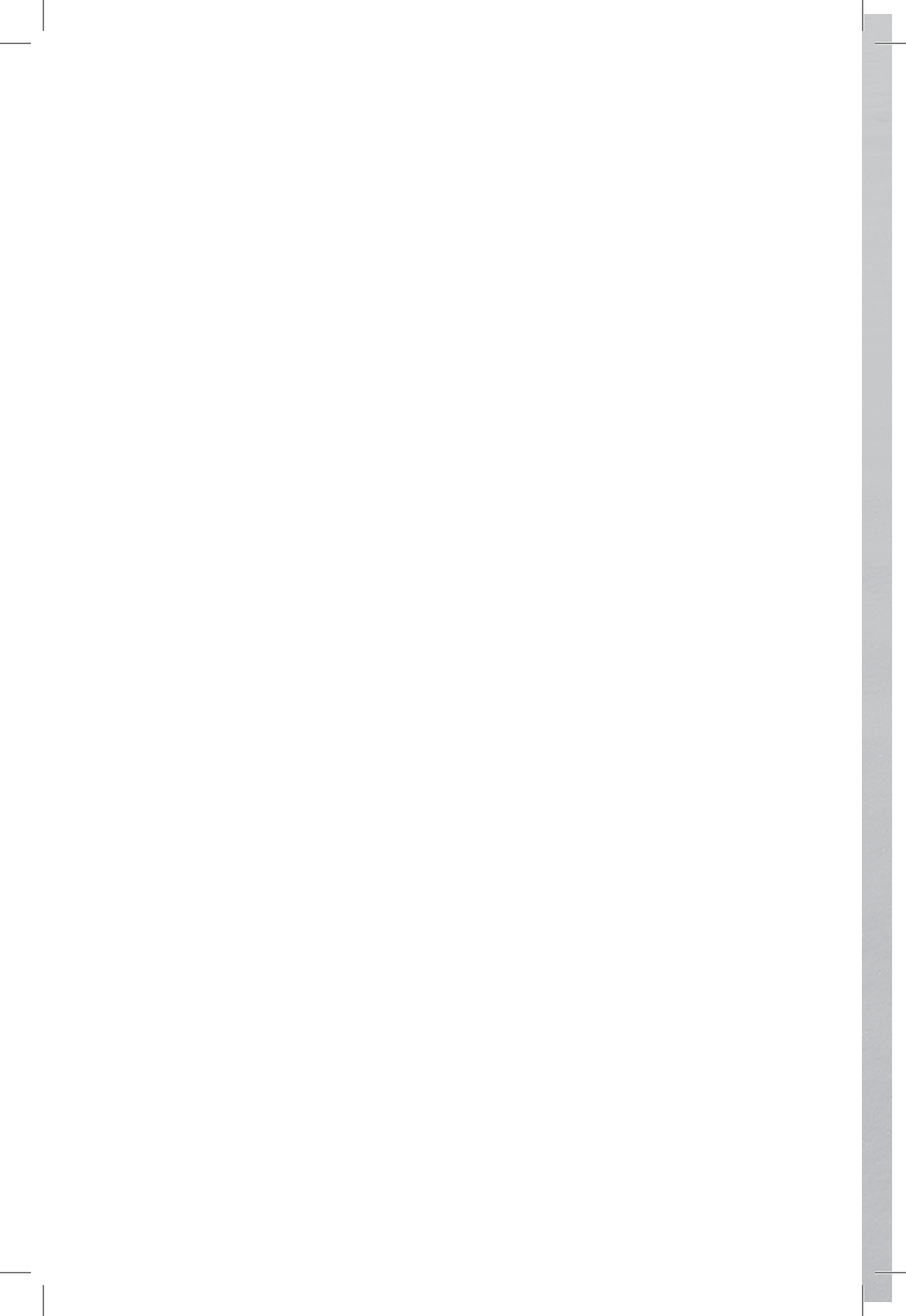
5. Execute o comando “ldapadd”, com a sintaxe abaixo, para aplicar a mudança descrita no arquivo “ldif”:

```
ldapadd -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w novasenha -f admin_pwd_base.ldif
```

Pronto! Agora a senha do usuário “admin” foi trocada para “novasenha”.

### ***8.8.1 Restaurando a senha original***

Refaça os procedimentos acima, devolvendo a senha do usuário “admin” para “senha”. Isso é importante porque a senha que será utilizada em todo o decorrer deste material é: senha.







CAPÍTULO 9

SUORTE A CRIPTOGRAFIA



O servidor OpenLDAP suporta trabalhar com dois esquemas de criptografia: SSL -- Secure Socket Layer -- ou TLS -- Transport Layer Security. A diferença básica entre ambos é que o SSL, por ser uma camada de segurança, a porta na qual o serviço trabalha deve ser diferente da padrão. TLS é criptografia na camada de transporte, como o próprio nome já diz. Sendo assim, TLS pode ser ativado sem a necessidade de alteração da porta do serviço.

A “força” da criptografia tanto do SSL quanto do TLS é basicamente a mesma pois ambos são criadas da mesma forma pelo **OpenSSL**.

A vantagem de se utilizar esquemas de criptografia é bastante óbvio: provar a identidade do servidor e proteger os dados em trânsito, que de outra forma trafegariam na rede em texto puro.

## 9.1 Ativando TLS

A ativação do suporte ao TLS exigirá a geração de um par de chaves criptográficas e a sua assinatura digital. O passo a passo abaixo descreve as atividades necessárias para isso.

Atenção para a pergunta “Common Name (eg, YOUR name) []:”! Este campo tem que ser preenchido, sempre, com o FQDN do seu host. Se você tiver dúvida sobre qual é o FQDN do seu servidor, execute o comando “hostname” em um terminal.

### 1. Instale o OpenSSL

```
aptitude install openssl
```

### 2. Crie um diretório para armazenamento das chaves

```
mkdir /etc/ldap/tls  
cd /etc/ldap/tls
```

### 3. Crie uma agência certificadora

```
/usr/lib/ssl/misc/CA.sh -newca
```

4. O seguinte questionário para criação da senha deve ser preenchido

CA certificate filename (or enter to create)

Making CA certificate ...

Generating a 1024 bit RSA private key

.....++++++

.....++++++

writing new private key to './demoCA/private/./cakey.pem'

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter ';', the field will be left blank.

-----

Country Name (2 letter code) [AU]:BR

State or Province Name (full name) [Some-State]:SP

Locality Name (eg, city) []:SaoPaulo

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Anahuac

Organizational Unit Name (eg, section) []:TI

Common Name (eg, YOUR name) []:Signatures Co.

Email Address []:webmaster@anahuac.org

Please enter the following 'extra' attributes to be sent with your certificate request

A challenge password []:123456

An optional company name []:Signatures Co.

5. Após preencher o formulário será gerada a agência certificadora e a chave para podermos assinar o certificado do nosso servidor

Using configuration from /usr/lib/ssl/openssl.cnf

Enter pass phrase for ./demoCA/private/./cakey.pem:

Check that the request matches the signature

Signature ok

Certificate Details:

Serial Number: 0 (0x0)

Validity

Not Before: May 31 13:04:53 2007 GMT

Not After : May 30 13:04:53 2010 GMT

Subject:

countryName = BR

stateOrProvinceName = SP

organizationName = Anahuac

organizationalUnitName = TI

commonName = Signatures Co.

emailAddress = webmaster@anahuac.org

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

OpenSSL Generated Certificate

X509v3 Subject Key Identifier:

B0:21:74:8E:AC:AC:9D:1A:E8:1F:2C:AB:D7:A4:B9:11:-  
D4:4B:3F:84

X509v3 Authority Key Identifier:

keyid:B0:21:74:8E:AC:AC:9D:1A:E8:1F:2C:AB:-  
D7:A4:B9:11:D4:4B:3F:84

Certificate is to be certified until May 30 13:04:53 2010 GMT  
(1095 days)

Write out database with 1 new entries

Data Base Updated

6. Agora podemos criar o certificado para o nosso servidor  
openssl req -new -nodes -keyout newreq.pem -out newreq.  
pem

7. Um novo questionário será apresentado, mas agora para  
a chave do nosso servidor. A única pergunta que deve ser  
respondida de forma precisa é o nome do servidor -- Com-  
mon Name -- que deve ser o FQDN da máquina servidora.

Generating a 1024 bit RSA private key

.....++++++

.....++++++

writing new private key to 'newreq.pem'

-----

You are about to be asked to enter information that will be  
incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished  
Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter ';', the field will be left blank.

-----

Country Name (2 letter code) [AU]:BR

State or Province Name (full name) [Some-State]:SP

Locality Name (eg, city) []:SaoPaulo

Organization Name (eg, company) [Internet Widgits Pty  
Ltd]:Anahuac

Organizational Unit Name (eg, section) []:TI

Common Name (eg, YOUR name) []:ldap\_\_.anahuac.org

Email Address []:webmaster@anahuac.org

Please enter the following 'extra' attributes  
to be sent with your certificate request

A challenge password []:123456

An optional company name []:Anahuac

8. Agora que o certificado do servidor e a agência certificadora foram criados, resta assinar o certificado do servidor usando a agência certificadora.

```
/usr/lib/ssl/misc/CA.sh -sign
```

9. Após esse comando será apresentado o certificado do servidor e surgirá a pergunta confirmando se é realmente esse certificado que deve ser assinado. Confirme.

```
Using configuration from /usr/lib/ssl/openssl.cnf
```

```
Enter pass phrase for ./demoCA/private/cakey.pem:
```

```
Check that the request matches the signature
```

```
Signature ok
```

```
Certificate Details:
```

```
Serial Number: 1 (0x1)
```

```
Validity
```

```
Not Before: May 31 13:15:00 2007 GMT
```

```
Not After : May 30 13:15:00 2008 GMT
```

```
Subject:
```

```
countryName          = BR
```

```
stateOrProvinceName  = SP
```

```
localityName         = SaoPaulo
```

```
organizationName     = Anahuac
```

```
organizationalUnitName = TI
```

```
commonName           = ldap__.anahuac.org
```

```
emailAddress          = webmaster@anahuac.org
```

```
X509v3 extensions:
```

```
X509v3 Basic Constraints:
```

```
CA:FALSE
```

```
Netscape Comment:
```

```
OpenSSL Generated Certificate
```

```
X509v3 Subject Key Identifier:
```

57:E9:23:6E:F4:D6:3D:11:BD:37:81:02:04:1E:-  
D4:02:04:55:C3:EB

X509v3 Authority Key Identifier:

keyid:B0:21:74:8E:AC:AC:9D:1A:E8:1F:2C:AB:-  
D7:A4:B9:11:D4:4B:3F:84

Certificate is to be certified until May 30 13:15:00 2008 GMT  
(365 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

10. Após as confirmações, será impresso na tela o conteúdo do  
certificado e a chave criptográfica, que agora localizam-se  
em “newcert.pem” e “newreq.pem”

Write out database with 1 new entries

Data Base Updated

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=BR, ST=SP, O=Anahuac, OU=TI, CN=Signatures  
Co./

emailAddress=webmaster@anahuac.org

Validity

Not Before: May 31 13:15:00 2007 GMT

Not After : May 30 13:15:00 2008 GMT

Subject: C=BR, ST=SP, L=SaoPaulo, O=Anahuac, OU=TI,  
CN=ldap\_\_.anahuac.org/emailAddress=webmaster@anahuac.  
org

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:bc:c5:ad:4e:89:c9:b1:d0:45:7e:07:5c:50:f5:



55:b4:7f:04:aa:06:5c:d4:33:fb:3f:d6:72:1c:8a:  
f3:1e:2b:b6:c7:d3:82:56:3e:0f:10:4a:79:73:c4:  
e9:8b:3b:37:b5:bf:24:28:f8:1d:ad:66:b2:6c:d4:  
c5:13:b3:42:2b:c7:bf:b3:5f:64:55:4f:c5:56:e2:  
d1:63:4d:eb:19:10:e5:cb:40:e5:02:43:e8:de:28:  
8c:51:db:98:77:10:4d:a7:19:0b:c6:fe:4d:53:3e:  
ea:6d:6a:6e:87:27:ba:fc:44:36:f9:ac:33:16:1d:  
a2:70:c6:0f:ab:f9:19:1b:2d

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

OpenSSL Generated Certificate

X509v3 Subject Key Identifier:

57:E9:23:6E:F4:D6:3D:11:BD:37:81:02:04:1E:-  
D4:02:04:55:C3:EB

X509v3 Authority Key Identifier:

keyid:B0:21:74:8E:AC:AC:9D:1A:E8:1F:2C:AB:-  
D7:A4:B9:11:D4:4B:3F:84

Signature Algorithm: sha1WithRSAEncryption

76:ff:d3:b3:c2:4d:62:82:77:d9:f9:09:98:10:10:16:79:46:  
63:18:dc:6e:18:8e:dc:cc:70:45:e9:b1:1a:85:f7:2f:36:76:  
c6:48:b2:eb:38:1f:9b:85:34:fb:5a:81:ba:a4:7f:3f:74:19:  
18:2a:c3:42:78:be:05:5f:75:46:78:69:f2:e7:ca:53:39:43:  
7a:9c:6f:b8:54:b0:66:93:31:eb:4b:dd:5e:01:e9:dc:52:05:  
cf:a1:d2:d5:26:4c:32:16:bd:51:fa:3c:50:43:09:00:27:75:  
cb:90:b8:2d:48:b2:96:8f:28:cf:bf:ac:09:1f:df:81:f5:67:  
e4:93

-----BEGIN CERTIFICATE-----

MIIC/DCCAmWgAwIBAgIBATANBgkqhkiG9w0BAQUFADB-  
5MQswCQYDVQQGEWJCUjEL

MAkGA1UECBMCU1AxDzANBgNVBAoTBjRMaW51eDEL-  
MAkGA1UECxmCkVekxZzAVBgNV  
BAMTDlNpZ25hdHVyZXMgQ28uMSYwJAYJKoZIhvcNA-  
QkBFhd3ZWJtYXN0ZXJANGxp  
bnV4LmNvbS5icjAeFw0wNzA1MzExMzE1MDBaFw0wO-  
DA1MzAxMzE1MDBaMIGRMQsw  
CQYDVQQGEWJCUjELMAkGA1UECBMCU1AxETAPBgN-  
VBAcTCFNhb1BhdWxvMQ8wDQYD  
VQQKEWY0TGludXgxZCZAJBgNVBAsTAlRJRmwwGgYDVQQ-  
DExNsZGFwMS40bGludXgu  
Y29tLmJyMSYwJAYJKoZIhvcNAQkBFhd3ZWJtYXN0ZXJAN-  
GxpbnV4LmNvbS5icjCB  
nzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwMwTtonJ-  
s-dBFfgdcUPVVtH8EqZc  
1 D P 7 P 9 Z y H I r z H i u 2 x 9 O C V j 4 P E E p 5 c -  
8 T p i z s 3 t b 8 k K P g d r W a y b N T F E 7 N C K 8 e /  
s19k  
VU/FVuLRY03rGRDly0DlAkPo3iiMUduYdxBNpxkLxv5NUz-  
7qbWpuhye6/EQ2+awz  
Fh2icMYPq/kZGy0CAwEAAaN7MHkwCQYDVR0TBAlwA-  
DAsBglghkgBhvhCAQ0EHxYd  
T3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwH-  
QYDVR0OBBYEFFfpI2701j0R  
vTeBAgQe1AIEVcPrMB8GA1UdIwQYMBaAFLAhdI6sr-  
J0a6B8sq9ekuRHUSz+EMA0G  
CSqGS1b3DQEBBQUAA4GBAHb/07PCTWKCd9n5CZgQEB-  
Z5RmMY3G4YjtzMcEXpsRqF  
9y82dsZIsus4H5uFNptagbqkfz90GRgqw0J4vgVfdUZ4afLnyl-  
M5Q3qcb7hUsGaT  
MetL3V4B6dxSBc+h0tUmTDIWvVH6PFBDCQAndcuQu-  
C1IspaPKM+/rAkf34H1Z+ST  
-----END CERTIFICATE-----  
Signed certificate is in newcert.pem

11. Para facilitar a identificação dos arquivos, vamos alterar seus nomes

```
mv newcert.pem srvcert.pem
mv newreq.pem srvkey.pem
```

12. Vamos posicionar o certificado da agência certificadora no mesmo diretório para facilitar nossa vida

```
cp demoCA/cacert.pem .
```

## 9.2 Alterando o `cn=config` para ativar o TLS

O OpenLDAP precisa ser configurado de forma a saber onde ler os arquivos contendo o certificado, e a assinatura digital. Para isso será necessário criar um arquivo “ldif” contendo essas configurações e, em seguida, adicionar seu conteúdo à base de configuração “cn=config”.

Abaixo seguem os passos para fazê-lo:

1. Para ativar o suporte a TLS no servidor devemos criar um arquivo “ldif”, chamado “tls\_ssl.ldif” com o conteúdo abaixo:

```
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/tls/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/tls/srvcert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/tls/srvkey.pem
```

2. Adicione esse conteúdo ao “cn=config” com o comando abaixo:

```
ldapadd -Y EXTERNAL -H ldapi:/// -f tls_ssl.ldif
```

### 9.3 Ativando o suporte aos clientes

Para podermos usar uma ferramenta cliente para verificar se o suporte ao “TLS” está funcionando, precisamos configurá-la para usar “TLS”.

1. Adicione as seguintes linhas ao `/etc/ldap/ldap.conf`  
`TLS_CACERT /etc/ldap/tls/cacert.pem`  
`TLS_REQCERT allow`

A primeira linha indica o caminho completo para o arquivo contendo o certificado da autoridade certificadora. A segunda indica que o certificado do servidor deve ser aceito mesmo que a assinatura não seja “confiável”. Isso porque o status de um certificado auto-assinado é “não confiável”.

2. Altere o campo “URI” para referenciar o nome (FQDN) do servidor, não o endereço IP. Isso pois o certificado foi emitido usando o nome FQDN do servidor e se essas duas informações não corresponderem o acesso via “TLS” ficará impossibilitado.

```
URI ldap://server.anahuac.org
```

3. Para testar se o “TLS” está funcionando, basta adicionar o parâmetro `-ZZ` ao `ldapsearch`:

```
ldapsearch -x -ZZ -b dc=anahuac,dc=org
```

Se o comando com o parâmetro “-ZZ”, que pede conexão segura, funcionou então o suporte a “TLS” está configurado. O que resta é ativar o suporte a “TLS” para todos os serviços que forem acessar nossa base OpenLDAP.

## ***9.4 O teste final não funcionou. E agora?***

Pode acontecer do comando final de teste, usando o comando “ldapsearch” com a opção “-ZZ” não resulte como esperado. O erro está, com toda certeza, na geração do certificado ou na criação da agência certificadora.

Neste exercício o que fizemos foi autoassinar o certificado. Essa é uma técnica pouco aceita na Internet, mas muito útil para realizar comunicações encriptadas entre servidores próprios. Isso porque não há necessidade alguma de pagar a uma agência certificadora externa para assinar os certificados que serão utilizados internamente.

Portanto se algo não está dando certo, relaxe, respire fundo e comece de novo. Apague todo o conteúdo do diretório “/etc/ldap/tls” e reinicie todos os procedimentos do capítulo. O processo é muito melindroso, basta uma senha errada, um caracter errado ou o esquecimento de usar o FQDN na opção “Common Name” e o certificado não será gerado como deveria.

O segredo aqui é ter calma e fazer tudo bem devagar, prestando muita atenção.





CAPÍTULO 10

REPLICAÇÃO COM SYNCREPL





A necessidade de replicar bases de dados OpenLDAP surge a partir do momento que apenas uma máquina não está dando conta da carga ou simplesmente quando necessitamos de redundância no serviço.

Se considerarmos que o serviço de diretórios está centralizando toda a informação em um único lugar, concluímos, logo, que se esse único ponto falhar, todos os serviços que dele dependem também sairão do ar.

Junto com a replicação da informação surge o problema de manter todas as bases de dados sincronizadas, o que pode ser uma tarefa bastante complicada.

O OpenLDAP opera em um ambiente com um “Servidor Mestre” e vários outros secundários (tantos quantos forem necessários). Nesse ambiente, as consultas são efetuadas nas máquinas “Secundárias” e as atualizações na máquina “Mestre”. Quando uma atualização é efetuada na máquina “Mestre” a replicação é feita automaticamente, propagando essa informação para as máquinas “Secundárias”.

O OpenLDAP já pode, também, trabalhar no modo “Master” x “Master”. Nesse cenário as alterações podem ser feitas em qualquer um dos servidores envolvidos no ambiente e a informação será replicada para os demais servidores. Entretanto essa configuração somente será abordada na edição avançada deste material.

## *10.1 Entendendo o Syncrepl*

Este é o método utilizado a partir da série 2.4 do OpenLDAP. Trata-se de um módulo do próprio “slapd” e não há necessidade de ativar mais um serviço secundário.

O Syncrepl é um sistema de replicação é ativado e controlado pelo “Servidor Secundário”, ou seja, a responsabilidade de manter a replicação fica no cliente e não no servidor. A vantagem imediata deste método é não ter que reiniciar o “Servidor Mestre” sempre que se adiciona uma réplica ao ambiente.

Outro aspecto importante é o fato de que se a conexão entre o “Servidor Master” e o “Servidor Secundário” for interrompida por qualquer intervalo de tempo e depois for restaurada, a sincronização será feita automaticamente, sem a necessidade de intervenção manual.

Dois métodos de replicação estão disponíveis no “syncrepl”:

- “**push-based**” - Neste método o “Servidor Secundário”, após a primeira sincronização, monitora o “Servidor Mestre” de tempos em tempos, comparando as duas bases e atualizando às informações que foram modificadas. O intervalo de tempo utilizado para fazer às verificações pode variar de dias a segundos.

Este método é ativado definindo a opção “type” da configuração como “refreshOnly”.

- “**pull-based**” - Neste método o “Servidor Secundário”, após a primeira sincronização, mantém uma conexão persistente com o “Servidor Mestre”. Assim qualquer alteração é sincronizada imediatamente.

Este método é ativado definindo a opção “type” da configuração como “refreshAndPersist”.

Os dois métodos são extremamente eficientes porque utilizam um sistema de controle que informa apenas os campos que foram alterados, não sendo necessário trafegar a base completa do “Servidor Mestre” para o “Servidor Secundário”.

Todas as alterações feitas, sejam no “Servidor Master”, seja no “Servidor Secundário” são armazenados, originalmente em memória, usando para isso um “log”. Este log é volátil e não pode ser acessado diretamente. É esse log que permitirá que, depois de uma falha de comunicação, os registros sejam atualizados no sentido certo.

## 10.2 Configurando o “Servidor Mestre” para *syncrepl*

Neste servidor as configurações são mínimas e perenes, ou seja, só precisam ser feitas uma única vez, independente de quantos “Servidores Secundários” forem ser utilizados.

A configuração consiste em alguns passos:

- O primeiro consiste em incluir o módulo que ativa o recurso da sincronização. Este módulo chama-se “syncprov”;
- Criar um galho na base de configuração “cn=config” para armazenar as configurações específicas da sincronização. Este galho chama-se “olcOverlay={0}syncprov” e estará um nível abaixo do galho que armazena as configurações da base, o “olcDatabase={1}hdb”
- Adicionar os atributos “entryUUID” e “entryCSN” na lista de atributos a serem indexados. Isso é importante porque esses atributos são consultados o tempo todos para efeito de comparação das bases do servidor Master e o Slave.
- Criar um usuário específico para ser utilizado no processo de replicação que tenha poderes suficientes para, apenas, permitir a cópia integral de todos os atributos desejados.

A seguir temos o passo a passo, prático para fazer a configuração do servidor Master:

1. Será necessário criar um arquivo “ldif”, chamado “syncrepl\_master.ldif” contendo o seguinte conteúdo:

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
```

```
dn: olcOverlay={0}syncprov,olcDatabase={1}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcConfig
objectClass: top
```

```
objectClass: olcSyncProvConfig
olcOverlay: {0}syncprov
olcSpCheckpoint: 100 10
olcSpSessionlog: 100

dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID,entryCSN eq
```

### 10.2.1 Explicando o “ldif”

- **Trecho 1** – ativa o módulo de sincronização
- **Trecho 2** – configura os “pontos de checagem”. O atributo “olcSpCheckpoint” recebe dois argumentos: o primeiro é o número de transações e o segundo é o tempo em minutos. Quando qualquer um dos dois for atingido o “Syncrepl” fará mais uma checagem de integridade.

Já o atributo “olcSpSessionlog” define o número máximo de registros que serão mantidos no “log” de alterações que serão replicadas.

- **Trecho 3** – Indica que os atributos “entryUUID” e “entryCSN” serão indexados pelo tipo “eq”.

A Replicação dos atributos “userPassword” e “shadowLast-Change” somente será possível, se o o usuário que será utilizado para esse fim tiver o devido permissionamento. Por padrão esses dois atributos só podem ser lidos/modificados pelo usuário “admin” e/ou por usuários autenticados. Neste exercício prático criaremos um usuário chamado “Replicator”. E para permitir que esse usuário leia esses atributos e assim replicá-los, alteraremos a ACL que controla os mesmos.

Serão necessários dois arquivos: um para criar o usuário e outro para dar-lhe as devidas permissões.

1. Criar o usuário exigirá um arquivo “ldif”, chamado “replicator.ldif”, com o seguinte conteúdo:

```
dn: cn=Replicator,dc=anahuac,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: Replicator
description: LDAP Replicator
userPassword: {SSHA}gwG/6jSd1S5TQ8vcKyC9Acik4kKfy-
Q+J
```

OBS: O hash acima, é referente a senha “123456”, sem as aspas.

2. Adicione o usuário com o comando abaixo:

```
ldapadd -h localhost -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f replicator.ldif
```

3. Adicionar o permissionamento correto para o usuário “Replicator” exigirá um arquivo “ldif”, chamado “replicator\_acl.ldif”, com o seguinte conteúdo:

```
dn: olcDatabase={1}hdb,cn=config
replace: olcAccess
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous auth by dn="cn=admin,dc=anahuac,dc=org" write by dn="cn=replicator,dc=anahuac,dc=org" read by * none
-
add: olcAccess
olcAccess: {1}to dn.base="" by * read
-
add: olcAccess
olcAccess: {2}to * by self write by dn="cn=admin,dc=anahuac,dc=org" write by * read
```

Perceba que foi adicionado permissionamento - dn="cn=replicator,dc=anahuac,dc=org" read – para garantir o acesso de somente leitura dos atributos que armazenam senhas.

Os dois trechos seguintes recolocam os permissionamentos originais de volta.

4. Adicione o permissionamento com o comando abaixo:  
`ldapmodify -Y EXTERNAL -H ldapi:/// -f replicator_acl.ldif`

### ***10.3 Configurando o “Servidor Secundário” ou “Slave” para Syncrepl***

Estas configurações devem ser feitas em todas as replicas que fizerem parte do ambiente de replicação. Perceba que o controle de acesso para replicações é a senha do usuário “replicator”.

O procedimento deve seguir uma sequência coerente. Será necessário parar o OpenLDAP para remover a base física e depois iniciá-lo para que uma base completamente limpa seja refeita automaticamente. Somente depois é que a primeira replicação deve ser feita. Isso garantirá que as bases sejam idênticas.

Parece óbvio, mas não custa relembrar que o “sufixo”, ou seja, a raiz das bases precisam ser iguais nos dois servidores. E os “schemas” incluídos em cada um deles também. Um atributo inexistente em um dos servidores, certamente, impedirá a replicação de ser feita.

O procedimento, passo a passo, está descrito abaixo:

1. Esteja seguro que o host ao qual vai se conectar seja “resolúvel” por DNS ou pelo “hosts”. Neste exemplo usaremos o nome do servidor “server.anahuac.org”. Para torná-lo resolúvel, altere o arquivo “/etc/hosts” do servidor Slave, adicionando uma linha assim:

```
endereço_ip server.anahuac.biz
```

Neste caso o nosso endereço IP será 192.168.0.222, mas você poderá utilizar qualquer outro endereço do seu ambiente.

2. Pare o OpenLDAP:  
`/etc/init.d/slaped stop`

3. Remova os arquivos físicos da base do OpenLDAP:  
`rm /var/lib/ldap/*`
4. Inicie o OpenLDAP em modo debug, apenas para que os arquivos da base sejam recriados:  
`slapd -d 256`
5. Interrompa o OpenLDAP pressionando as teclas “Ctrl + c”;
6. Corrija o permissionamento:  
`chown openldap: /var/lib/ldap/*`
7. Inicie o OpenLDAP da forma comum:  
`/etc/init.d/slapd start`
8. Agora é hora de criar um arquivo “ldif”, chamado “syncrepl\_slave.ldif”, com o seguinte conteúdo:

ATENÇÃO: altere o endereço IP ou host, para se adequar a sua realidade!

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
```

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
```

-

```
add: olcSyncrepl
```

```
olcSyncrepl: {0} rid=000 provider=ldap://server.anahuac.biz
type=refreshAndPersist interval=00:00:00:10 searchbase=dc=a-
nahuac,dc=org filter="(objectClass=*)" scope=sub attrs="*"
schemachecking=off bindmethod=simple binddn=cn=replica-
tor,dc=anahuac,dc=org credentials=123456 retry="10 +"
```

OBS: lembre-se de trocar o endereço IP da opção “provider” por um endereço IP ou host válido no seu ambiente.

9. Adicione o conteúdo do arquivo “ldif” com o seguinte comando:

```
ldapadd -c -Y EXTERNAL -H ldapi:/// -f syncrepl_slave.ldif
```

10. Aguarde o tempo definido pela opção “retry”, no atributo “olcSyncrepl” acima e, então, verifique a base para ver se a replicação já está sendo feita.

### *10.3.1 Explicando o “ldif”*

- **Trecho 1** – ativa o módulo de sincronização
- **Trecho 2** – Indica que os atributos “entryUUID” e “entryCSN” serão indexados pelo tipo “eq”.
- **Trecho 3** – Define todas as condições da replicação. Pode-se dividir o conteúdo do atributo “olcSyncrepl” pelos espaços. Isso será útil para entender cada um deles:

**Arg 1** – Indexador do atributo. Assim como ocorre com outros atributos, este prefixo é utilizado para estabelecer a ordem na qual eles deverão ser lidos, no caso de haver mais de um;

**Arg 2** – “rid” é o “replicator id” e é um número que não deve passar de três dígitos

**Arg 3** – “provider” define o host e a porta de conexão do “Servidor Master”

**Arg 4** – “type” define a técnica de replicação: “refreshOnly” ou “refreshAndPersist”

**Arg 5** – “interval” define o intervalo de verificação para a técnica de replicação “refreshOnly”, no formato “dias:horas:minutos:segundos”

**Arg 6** – “searchbase” define a parte da árvore a partir da qual a pesquisa para replicação será feita. Aconselhamos deixar a raiz da base definida nesta opção.



**Arg 7** – “filter” define o filtro de pesquisa que será feita em busca de alterações. Isso significa que pode-se definir o que se deseja replicar. Se não declarado, seu valor padrão é “(objectClass=\*)”

**Arg 8** – “scope” define se a pesquisa será recursiva ou não. Se não declarada, seu valor padrão é “sub”, ou seja, recursiva

**Arg 9** – “attrs” define quais atributos serão pesquisados. Pode-se definir uma lista de atributos separados por vírgula. Se não declarada, seu valor padrão é “\*”, ou seja, todos.

**Arg10** – “schemachecking” ativa a checagem dos schemas, ou seja, será feita uma verificação para saber se os schemas existem, antes de fazer a sincronização. Se não declarada, seu valor padrão é “off”, ou seja, desativada.

**Arg 11** – “bindmethod” é o método pelo qual a autenticação será realizada. As opções são “simple” ou “sasl”. A opção “simple” somente deve ser usada se houver um ambiente seguro de comunicação ou se a comunicação estiver sob “TLS”. A opção “sasl” exigirá a adição da opção “saslmech” para definir o mecanismo “sasl” que deve ser usado.

**Arg 12** – “binddn” define o usuário que será usado para a replicação.

**Arg 13** – “credentials” - define a senha do usuário que será usado para a replicação.

**Arg 14** – “retry” - define quanto tempo após a primeira falha, a conexão deve ser tentada outra vez.

## ***10.8 Ativando a replicação “Syncrepl”***

A replicação é ativada automaticamente no momento em que o “ldif” com as configurações é adicionado ao “cn=config”.

## ***10.9 Ativando encriptação na replicação***

O exemplo visto acima faz a replicação de forma plana, ou seja, sem encriptação. A ativação da encriptação se dá pela adição de mais um argumento nas configurações definidas no atributo “olcSyncrepl”.

O argumento a ser adicionado é “starttls=yes”.

Portanto, para ativar a encriptação à replicação siga os passos abaixo:

1. Crie um arquivo “ldif”, chamado “syncrepl\_slave\_tls.ldif”, com o seguinte conteúdo:

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
replace: olcSyncrepl
olcSyncrepl: {0} rid=000 provider=ldap://server.anahuac.biz
type=refreshAndPersist interval=00:00:00:10 searchbase=-dc=anahuac,dc=org filter="(objectClass=*)" scope=sub
attrs="*" schemachecking=off bindmethod=sasl binddn=cn=replicator,dc=anahuac,dc=org credentials=123456 retry="10 +"
starttls=yes
```

OBS: lembre-se de trocar o nome do host da opção “provider” por um host válido no seu ambiente.

2. Adicione o conteúdo do arquivo “ldif” com o seguinte comando:

```
ldapadd -c -Y EXTERNAL -H ldapi:/// -f syncrepl_slave_tls.ldif
```

3. Aguarde o tempo definido pela opção “retry”, no atributo “olcSyncrepl” acima e, então, verifique a base para ver se a replicação já está sendo feita.

## ***10.10 Populando a base de testes***

Antes de iniciarmos as configurações e integrações com outros serviços será importante popular a nossa base LDAP com alguns usuários e grupos. Assim os testes podem ser realizados de forma mais verídica.

Para popular a base LDAP utilizaremos o script abaixo.

1. Crie um arquivo chamado “popula.sh” e insira o conteúdo abaixo:

```

#!/bin/bash

LDAP_HOST="localhost"
LDAP_PORT="389"
LDAP_DN="cn=admin,dc=anahuac,dc=org"
LDAP_PWD="senha"

USERS_FILE="/tmp/users.ldif"
UID_NUMBER="2000"

>$USERS_FILE

I=1
while [ "$I" -le "30" ] ; do

# Criando arquivo ldif
echo "dn: cn="Fulano$I da Silva",ou=Usuarios,dc=anahuac,-
dc=org" >> $USERS_FILE
echo "uid: fulano$I" >> $USERS_FILE
echo "cn: Fulano$I da Silva" >> $USERS_FILE
echo "sn: fulano$I" >> $USERS_FILE
echo "givenName: fulano$I" >> $USERS_FILE
echo "objectclass: inetOrgPerson" >> $USERS_FILE
echo "objectclass: posixAccount" >> $USERS_FILE
echo "objectclass: shadowAccount" >> $USERS_FILE
echo "homeDirectory: /home/fulano$I" >> $USERS_FILE
echo "loginShell: /bin/bash" >> $USERS_FILE
echo "uidNumber: $UID_NUMBER" >> $USERS_FILE
echo "gidNumber: $UID_NUMBER" >> $USERS_FILE
USER_PWD=`slappasswd -s fulano$I`
echo "userPassword: $USER_PWD" >> $USERS_FILE
echo "gecos: Super $UID_NUMBER" >> $USERS_FILE
echo "" >> $USERS_FILE
echo "dn: cn="fulano$I",ou=Grupos,dc=anahuac,dc=org"
>> $USERS_FILE
echo "cn: fulano$I" >> $USERS_FILE

```

```
echo "objectclass: top" >> $USERS_FILE
echo "objectclass: posixGroup" >> $USERS_FILE
echo "gidNumber: $UID_NUMBER" >> $USERS_FILE
echo "memberUid: fulano$I" >> $USERS_FILE
echo "" >> $USERS_FILE

I=`expr $I + 1`
UID_NUMBER=`expr $UID_NUMBER + 1`
done

ldapadd -h $LDAP_HOST -p $LDAP_PORT -x -D $LDAP_
DN -w $LDAP_PWD -f $USERS_FILE

rm $USERS_FILE
```

2. Execute o script popula.sh com o comando:

```
sh popula.sh
```

3. Verifique se os usuários foram criados usando o comando slapcat ou algum cliente LDAP de sua preferência.

Serão criados trinta usuários com estas informações:

- cn="FulanoX da Silva"
- uid=fulanoX
- userPassword=fulanoX

Onde X é um número variando de 1 a 30.



CAPÍTULO 11

INTEGRANDO SERVIDOR APACHE



Criar uma área protegida em um site é uma necessidade cada vez mais comum. Seja para disponibilizar conteúdo pago ou simplesmente para restringir o acesso a pessoas autorizadas.

É possível proteger uma área de seu site usando os métodos próprios de autenticação do servidor Web Apache. Na essência, basta configurar o diretório para que apenas os usuários autenticados possam acessá-la.

Aqui usaremos o módulo de autenticação `libapache2-authnz-ldap`. A intenção é fornecer uma janela de login de acesso a uma área do site usando um par usuário/senha armazenado no servidor OpenLDAP.

### *11.1 Instalando o Apache*

Para instalar o Apache versão 2, execute o seguinte comando:  
`aptitude install apache2`

### *11.2 Ativando módulo de suporte ao LDAP*

O módulo de suporte ao LDAP será ativado no Apache ao executar o seguinte comando:

```
a2enmod authnz_ldap
```

Será necessário reiniciar o Apache:  
`/etc/init.d/apache2 restart`

### *11.3 Criando “virtualhost” com acesso autenticado*

Para efeito de testes vamos criar um “virtualhost” em nosso servidor Apache. Assim poderemos visualizar a autenticação acontecendo.

1. Crie um arquivo chamado “`anahuac.org`” e insira o seguinte conteúdo:

```
<VirtualHost *>  
    ServerAdmin webmaster@server.anahuac.org  
    ServerName server.anahuac.org
```

```
DocumentRoot /var/www/ldap
LDAPTrustedClientCert CERT_BASE64 /etc/ldap/tls/ca-
cert.pem
LDAPTrustedMode TLS

<Directory /var/www/ldap>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    AuthType Basic
    AuthName "Curso de OpenLDAP"
    AuthBasicProvider ldap
    AuthzLDAPAuthoritative off
        AuthLDAPURL "ldap://server.anahuac.org:389/
ou=Usuarios,dc=anahuac,dc=org?uid"
        require valid-user

    Order allow,deny
    allow from all
</Directory>

ErrorLog /var/log/apache2/ldap-error.log

# Possible values include: debug, info, notice, warn, error,
crit,
# alert, emerg.
LogLevel warn

CustomLog /var/log/apache2/ldap-access.log combined
ServerSignature Off
</VirtualHost>
```



2. Copie o arquivo para o diretório `/etc/apache2/conf.d`  
`cp anahuac.org /etc/apache2/conf.d`

3. Crie o diretório do nosso “virtualhost” e um arquivo “index.html”, com os comandos abaixo:

```
mkdir /var/www/ldap/  
echo “Autenticação por LDAP Funciona” > /var/www/ldap/  
index.html
```

4. Reinicie o Apache  
`apache2ctl restart`

5. Utilize seu navegador preferido para acessar a URL:  
`http://server.anahuac.org/ldap`

Usuário: fulano1

Senha: fulano1

### ***11.3 Criando acessos condicionais***

A configuração anterior permitirá que qualquer usuário válido existente na base LDAP possa acessar a página. Entretanto, em algumas situações, será desejável que o acesso seja permitido apenas para alguns usuários, ou seja, a simples restrição por senha não será suficiente.

Existem diversos métodos que permitem selecionar quais usuários podem ou não acessar uma determinada página web sob autenticação em bases LDAP. O método mais utilizado é a adição de um atributo aos usuários que poderão acessar a página, se a validação for realizada com sucesso.

Basta escolher um atributo e adicioná-lo aos usuários desejados e alterar a linha “require valid-user” nas configurações do Apache, por:

```
require ldap-attribute <atributo>=<valor>
```

Exemplo:

```
require ldap-attribute title=acessoweb
```

Neste exemplo apenas os usuários que possuírem o atributo “title” definido como “acessoweb” conseguirão acessar a página.

### ***11.4 Testando autenticação***

Para testar a autenticação basta utilizar o seu navegador preferido e acessar o “virtualhost”. Uma tela será exibida solicitando autenticação. Basta fornecer os dados.



CAPÍTULO 12

**INTEGRANDO SERVIDOR PROXY**



Servidor Proxy é um serviço que faz algo que os clientes deste servidor não podem, ou não são capazes de fazer. No caso do servidor Squid, o servidor é “proxy” para navegação.

Os clientes solicitam páginas ao servidor Squid e é ele quem se conecta aos sites externos, traz as páginas e os entrega aos clientes. Essa função de intermediação é fundamental para conseguir as vantagens adicionais do uso do Servidor Proxy Squid:

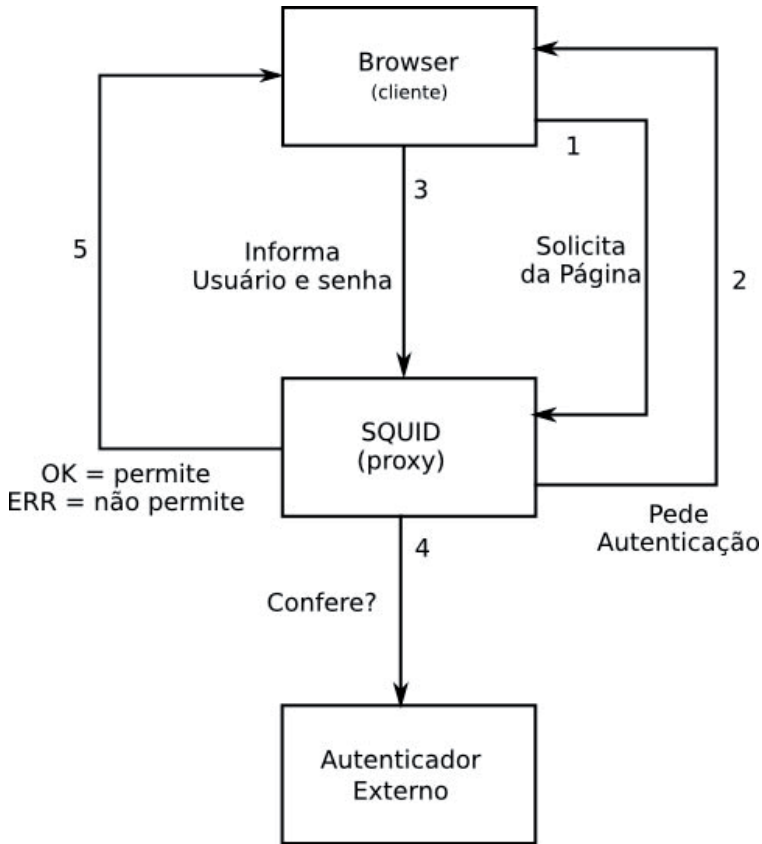
- Monitoramento do uso dos recursos de Internet
- Implementação de políticas de uso
- Aceleração da navegação com o uso de pré-armazenamento (cache).

O monitoramento de um log `<b>Squid</b>` pode ficar bem mais fácil se pudermos identificar quem acessou cada página (ou tentou acessar).

Na maneira padrão, sem autenticação, dispomos apenas do endereço IP da máquina para gerar as políticas de filtragem. Com a autenticação, podemos aplicar restrições baseadas em nomes de usuários.

A grande vantagem desta metodologia é a política de uso do Proxy ser válida para determinado usuário independentemente da máquina usada para a navegação (é óbvio que emprestar senhas continuará sendo um problema).

A autenticação no Squid funciona de uma forma muito simples. Veja a imagem abaixo:



1. O Squid recebe uma solicitação de navegação por parte do cliente;
2. O Squid envia um pedido de autenticação;
3. O cliente informa um usuário e senha;
4. O Squid usa um programa externo que executa as operações necessárias para verificar se aqueles parâmetros informados são válidos no contexto e devolve para o Squid a sua conclusão a respeito das informações enviadas pelo Squid;
5. Caso a resposta seja um simples OK, o usuário está autenticado e a página solicitada segue os caminhos normais de filtragem de conteúdo e restrições. Caso a resposta seja ERR, então o Squid devolve uma página de Acesso Negado.

## 12.1 Instalando o Squid

Para instalar o Squid execute o seguinte comando:

```
aptitude install squid
```

O Squid pode utilizar diversos autenticadores externos, que podem ser vistos em `/usr/lib/squid`.

## 12.2 Testando o autenticador LDAP

No nosso caso queremos que os usuários do nosso servidor Squid sejam autenticados contra a base de dados LDAP. Sendo assim, o autenticador a ser usado é o `/usr/lib/squid/ldap_auth`.

Para testá-lo basta executá-lo em um terminal e fornecer o par de usuário e senha. O comando é este:

```
/usr/lib/squid/ldap_auth -b ou=Usuarios,dc=anahuac,dc=org -v3 -f'(uid=fulano1)' -h ldap__.anahuac.org -ZZ
```

Explicando

- **-b** – Informa o “galho” onde a busca será feita
- **-v3** – Indica o uso do protocolo versão 3
- **-f** – Informa o filtro para busca
- **-h** – Especifica o host no qual se encontra o servidor OpenLDAP. Caso esteja usando “TLS” deve ser informado o “FQDN” do servidor
- **-ZZ** – Força o uso de “TLS” na conexão

## 12.3 Lidando com usuários e grupos

Em ambientes de maior porte, depender de políticas de controle de acesso baseadas apenas em usuários isolados é muito trabalhoso e sujeito a erros.

Para viabilizar e simplificar a administração, é interessante trabalhar com grupos e controlar o acesso através deles. Para isso é necessário outra metodologia disponível no pacote Squid: o uso de programas auxiliares externos para verificação de grupos.

Para testar se determinado usuário faz parte de determinado grupo, definido na base OpenLDAP, será necessário usar o programa `/usr/lib/squid/squid_ldap_group`. Que é parte integrante do Squid.

Para os testes em linha de comando, além de fornecer a informação de qual é a base de usuários da busca, é necessário também, informar qual é a base de busca dos grupos. Sendo assim, usamos a opção “-b” para informar a base de grupos e a opção “-B” para informar a base de usuários.

## *12.4 Testando o autenticador de grupo e usuário*

1. Para testar se um usuário faz parte de um grupo, utilize o comando abaixo:

```
/usr/lib/squid/squid_ldap_group -d -b ou=Grupos,dc=anahuac,dc=org -B ou=Usuarios,dc=anahuac,dc=org -f '(&(memberid=fulano1)(cn=fulano1))' -h ldap__.anahuac.org -ZZ
```

## *12.5 Alterando o squid.conf*

Para colocar em prática as configurações necessárias em uma instalação “limpa” como a que foi feita será necessário acrescentar diversas opções em diferentes lugares.

Para facilitar vamos contar com a ajuda do sistema de busca de texto do editor “vi”. Para fazer uma pesquisa no editor “vi” basta usar a tecla “/” e digitar o texto que se deseja encontrar.

Perceba que o arquivo de configuração do Squid, “`/etc/squid/squid.conf`” possui centenas de linhas de comentários, que visam explicar o funcionamento de todas as suas opções. A ordem é: primeiro comentários e explicações e depois a opção em si. Fique atento ao fato de que vamos colocar as nossas configurações, sempre ao final dos comentários de cada opção.

Não faz parte deste escopo explicar todas as funcionalidades do Squid, entretanto estaremos comentando todas as configurações necessárias para efetuar a validação de usuários.



1. Procure pela opção “auth\_param”. Após as linhas comentadas desta opção, adicione o seguinte conteúdo:

```
auth_param basic program /usr/lib/squid/ldap_auth -b ou=
=Usuarios,dc=anahuac,dc=org -v3 -f (uid=%s) -h ldap__.anahuac.org -ZZ
```

```
auth_param basic children 5
```

```
auth_param basic realm Squid Curso de LDAP
```

```
auth_param basic credentialsttl 3 hours
```

```
auth_param basic casesensitive off
```

```
external_acl_type grupo_no_LDAP %LOGIN /usr/lib/squid/
squid_ldap_group -d -b ou=Grupos,dc=anahuac,dc=org -B
ou=Usuarios,dc=anahuac,dc=org -f “(&(memberuid=%u)
(cn=%g))” -h ldap__.anahuac.org -ZZ
```

#### Explicando

Estas linhas ativam a solicitação de autenticação para permitir a navegação. Atenção para as opções extras:

- “auth\_param basic credentialsttl 3 hours” que determina que a validação somente é válida por 3 horas
- “auth\_param basic casesensitive off” que elimina a sensibilidade para maiúsculas e minúsculas

2. Procure pela opção “acl CONNECT method CONNECT”.

Após essa opção, adicione o seguinte conteúdo:

```
acl Gdiretoria external grupo_no_LDAP diretores
```

```
acl pedeSenha proxy_auth REQUIRED
```

```
acl minhaRede src 192.168.200.0/255.255.255.0
```

#### Explicando

Estas opções criam as ACL's (Listas de Controle de Acesso). Sua sintaxe é bem simples:

```
acl <nome> <ação> <argumento>
```

Atenção para a linha “acl pedeSenha proxy\_auth REQUIRED”. Esta é a linha que cria a ACL para que a autenticação seja solicitada.

Recomendamos a leitura da documentação do Squid para obter maiores detalhes sobre como criar ACL's.

3. Procure pela opção “http\_access deny CONNECT !SSL\_ports”. Após essa opção, adicione o seguinte conteúdo:

```
http_access allow Gdiretoria
http_access allow pedeSenha
http_access allow minhaRede
```

#### Explicando

Estas duas linhas ativam as ACL's criadas antes, ou seja, é aqui onde se determina o que fazer com as ACL's, se elas são permitidas (allow) ou negadas (deny).

Neste exemplo estamos permitindo as três ACL's (pedeSenha, Gdiretoria e minhaRede).

A ordem na qual as opções “http\_access” são escritas no arquivo de configuração fazem toda a diferença. Sugerimos uma leitura mais detalhada da documentação do Squid para entender como a ordem afeta o comportamento do Squid.

4. Salve as alterações e reinicie o Squid com o comando:

```
/etc/init.d/squid reload
```

## ***12.6 Testando autenticação***

Para testar a autenticação configurar o seu navegador preferido para utilizar o Proxy, apontando para este servidor na porta 3128, que é a porta padrão do Squid, e tentar navegar.

A tela de autenticação surgirá e será necessário informar usuário e senha.

## 12.7 Monitoramento de logs com SARG

Este é um excelente software desenvolvido pelo brasileiro Pedro Orso. Sua função é retornar um relatório HTML consolidado com as atividades dos usuários Squid.

O SARG é supor simples de instalar e configurar.

1. Instale o SARG com o comando:

```
aptitude install sarg
```

A configuração do SARG é bastante simples, os dois únicos parâmetros de configuração relevantes são “access\_log” e “output\_dir” que especificam a localização do arquivo do log de acesso do Squid e o diretório onde o relatório será gerado, respectivamente.

2. Edite o arquivo “/etc/squid/sarg.conf” e altere estas opções:

```
language Portuguese
access_log /var/log/squid/access.log
title “Relatorio de Acesso pelo Squid”
output_dir /var/www/ldap/squid-reports
```

3. Para atualizar a base de dados do SARG e poder visualizar os acessos, execute o seguinte comando:

```
sarg
```

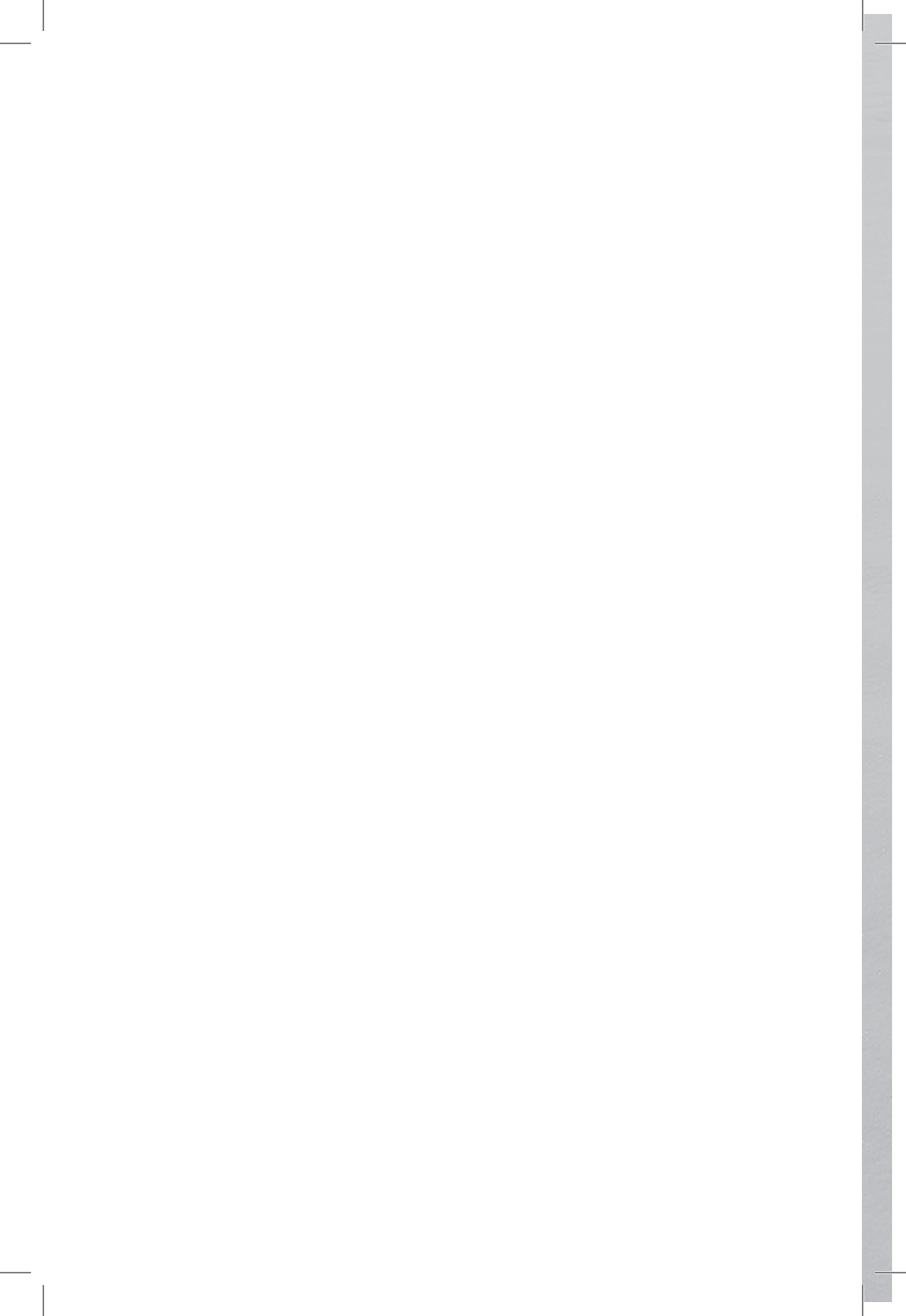
Este comando é apenas para poder visualizar o relatório imediatamente. o SARG instala automaticamente, “scripts” de geração de relatórios no “Cron” para serem executados todos os dias.

4. Finalmente, para poder ver os relatórios, usando o seu navegador preferido, acesse o link no servidor apache:

```
http://ldap__.anahuac.org/squid-reports
```

**ATENÇÃO:** para que este acesso funcione corretamente, na sequência deste manual, será necessário efetuar dois passos:

- a) `rm /etc/apache2/conf.d/anahuac.org`
- b) `apache2ctl restart`





CAPÍTULO 13

SERVIDOR POSTFIX



O Postfix é um MTA – Mail Transport Agent, ou mais popularmente um “Servidor SMTP”. Extremamente versátil, de fácil configuração e muito robusto ele tem se tornado o MTA padrão de quase todas as distribuições nos últimos anos.

Neste capítulo veremos como integrar o Postfix com o OpenLDAP.

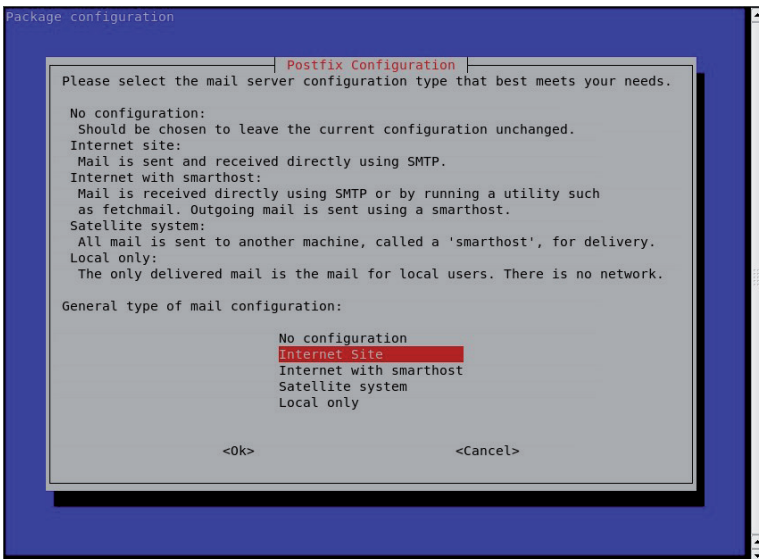
### 13.1 Instalando o Postfix

1. Para instalar o Postfix execute o seguinte comando:  
aptitude install postfix postfix-ldap

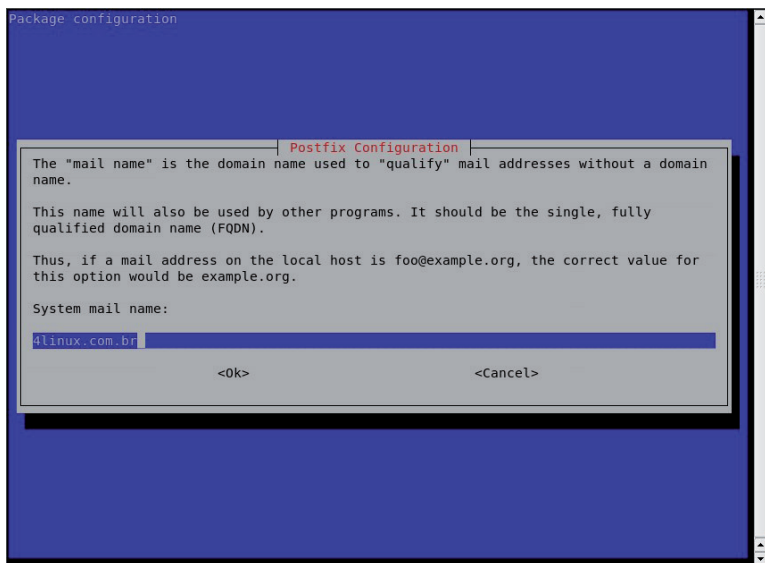
Perceba que no Debian o suporte a LDAP é fornecido por um pacote em separado. O Postfix oferece suporte a uma grande quantidade de bases diferentes.

Durante a instalação do servidor algumas telas de configuração irão aparecer. A primeira dela é informativa e explica os tipos de instalação que podemos utilizar, veja a sequência de figuras abaixo:

1. Selecione a opção “Internet Site”

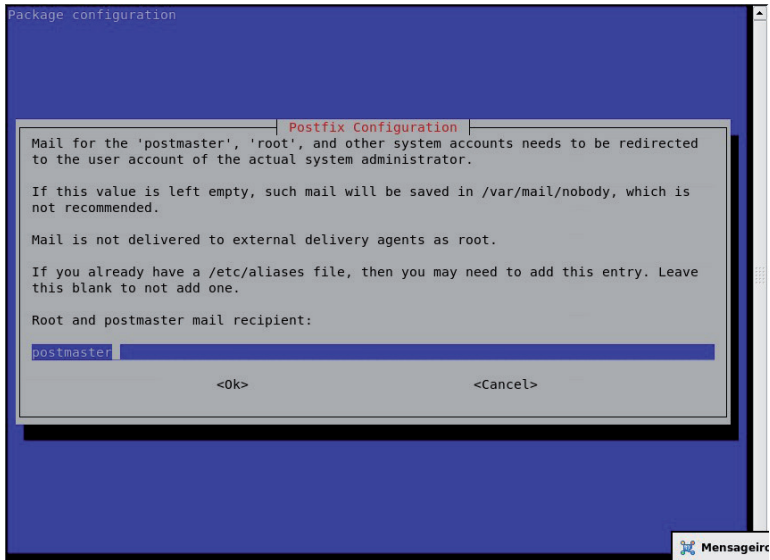


2. Nesta tela será solicitado o domínio padrão do sistema, ou seja, qual o domínio padrão que será acrescentado aos e-mails depois do @. Perceba que o instalador é inteligente o suficiente para já vir com o campo preenchido com o domínio do nome do servidor.

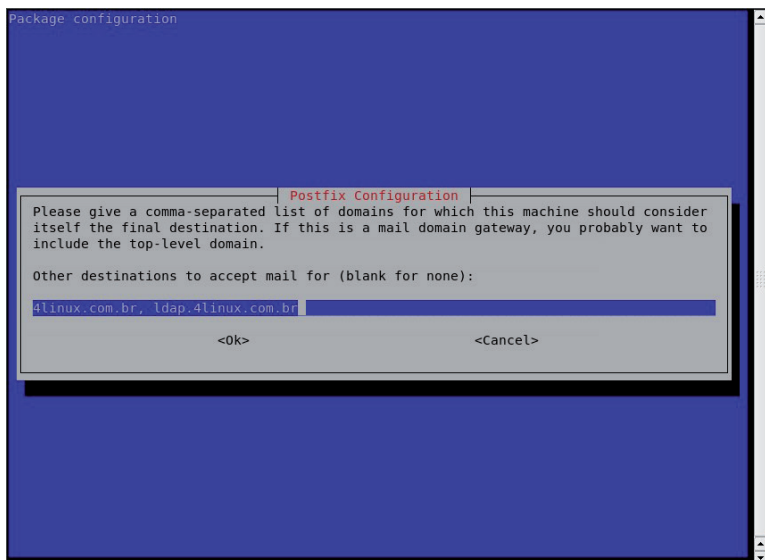


3. Algumas mensagens de alerta do sistema e/ou de outros servidores são enviados de forma automática para uma conta definida na configuração do Postfix. Recomendamos preencher esta tela com o usuário "postmaster". Mesmo que ele não exista agora, será importante criá-lo depois.

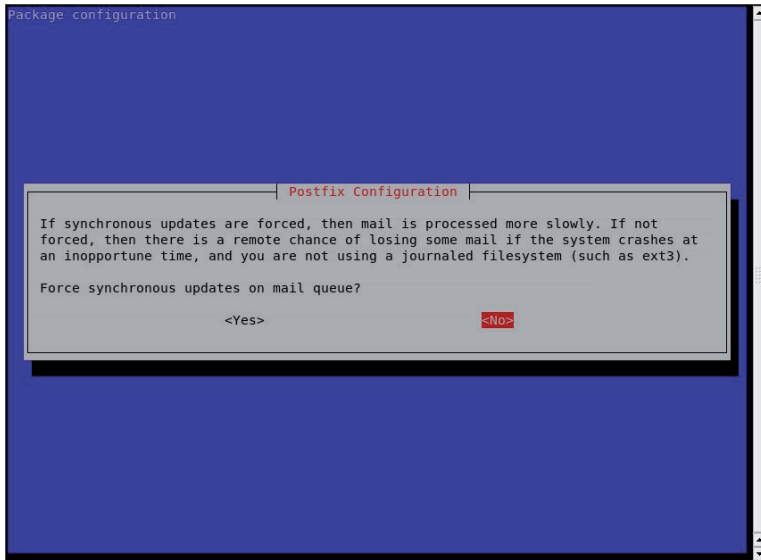




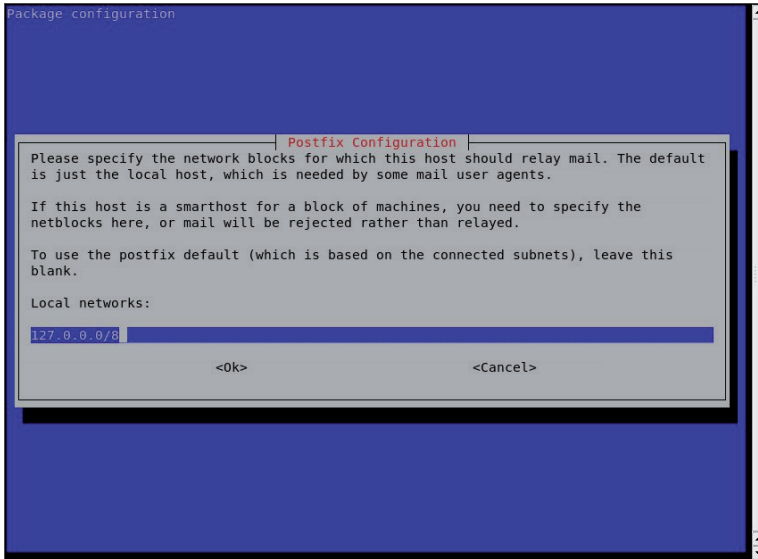
4. A tela seguinte pede que sejam informados quais são os domínios aceitos pelo Postfix. Inicialmente só temos um domínio e o FQDN do servidor. Assim basta deixar este campo preenchido com o domínio padrão e o nome do servidor. Como pode ser visto na figura abaixo:



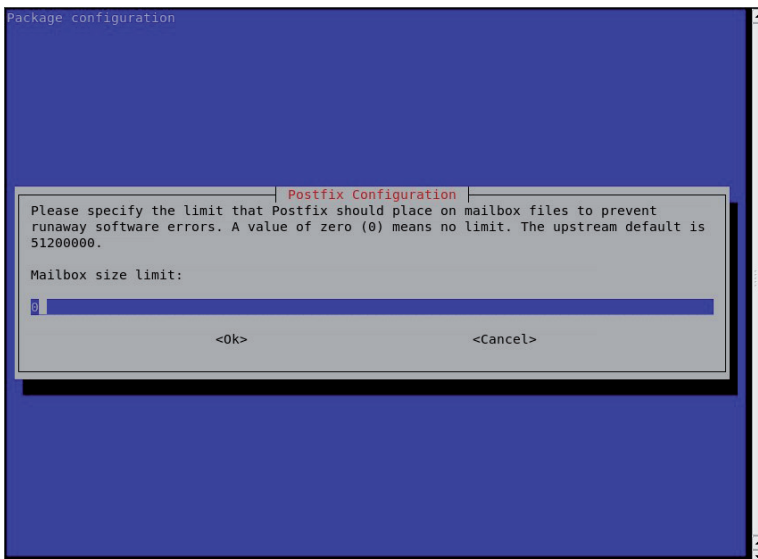
5. Esta tela configura a opção “synchronous updates”, que permite desativar as otimizações no envio das mensagens, fazendo com que os e-mails sejam enviados conforme são recebidos e em ordem. Esta opção aumenta um pouco a confiabilidade do servidor, pois reduz a possibilidade de perda de mensagens ainda não enviadas, em casos de travamentos ou quedas de energia. Por outro lado, ela reduz substancialmente o desempenho do servidor, por isso nunca deve ser ativada em servidores de grande volume.



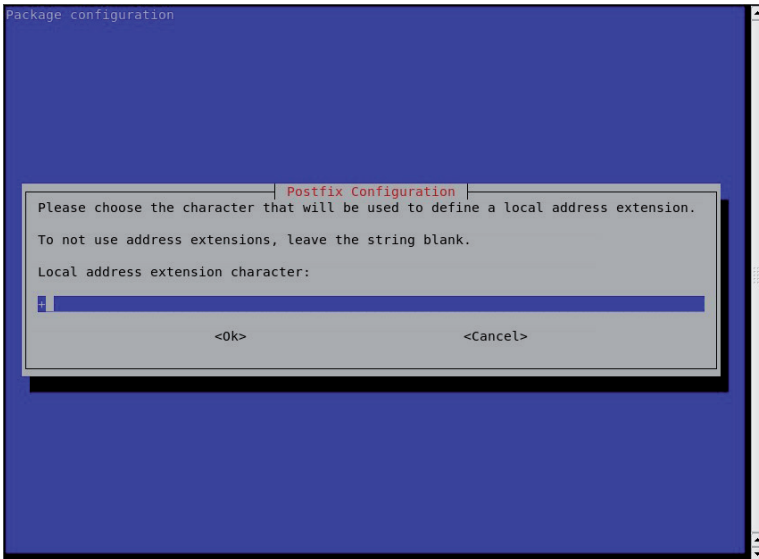
6. A tela seguinte pede que sejam informadas as redes que estão liberadas para envio de mensagens. Liberar redes por endereço IP é sempre uma má idéia. Deve-se, sempre que possível, utilizar mecanismos de autenticação de envio. Em nosso exercício o valor padrão é suficiente.



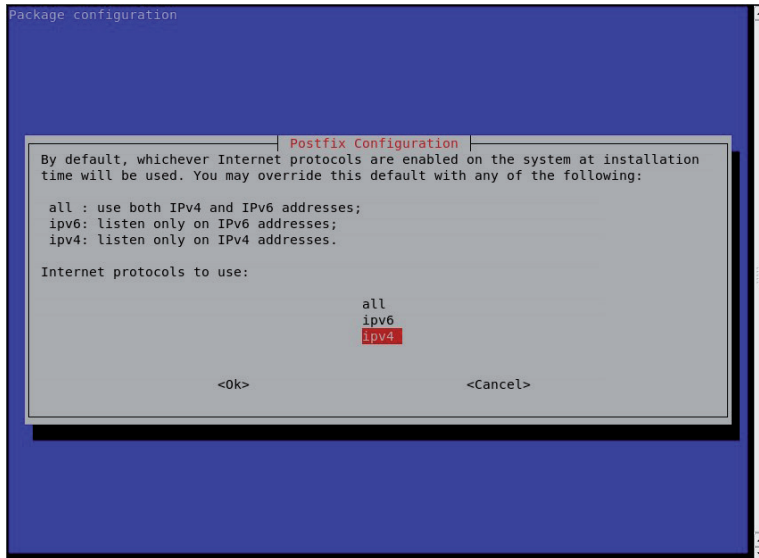
7. Nesta tela para que seja informado o tamanho máximo do “mailbox” das mensagens, ou seja, o tamanho máximo de armazenamento de mensagens.



- Defina nesta tela o caractere de extensão de endereço local. Um nome complicado para dizer: quando um e-mail chegar e for necessário verificar se é um usuário local posso tentar separar o nome do domínio usando um determinado caractere? Altere esta opção se for necessário utilizar separadores para alguma situação específica. Caso contrário mantenha o valor padrão:



- Qual o protocolo TCP que será utilizado? Se o ambiente não pedir IPV6, defina esta opção apenas como IPV4. Caso contrário seu registro de log pode ficar cheio de mensagens de aviso sobre IPV6.



### *13.2 Criando schema e/ou utilizando os já existentes*

O postfix é extremamente flexível quando se trata do suporte a LDAP. Tanto que ele não usa uma classe de objetos própria, fica critério do administrador configurar quais atributos serão utilizados. Entretanto outras partes integrantes de um servidor de e-mail completo, como o Maildrop podem exigir atributos específicos.

Pode-se criar um schema próprio para atender aos requisitos mínimos do Postfix ou pode-se utilizar algum outro schema já existente.

Aqui mostramos um exemplo de schema simples:

```
attributetype (1.3.6.1.4.1.14203.666.1.200
```

```
    NAME 'mailacceptinggeneralid'
```

```
    EQUALITY caseIgnoreMatch
```

```
    SUBSTR caseIgnoreSubstringsMatch
```

```
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024})
```

```

attributetype (1.3.6.1.4.1.4203.666.1.201
  NAME 'maildrop'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1024})

objectClass (1.3.6.1.4.1.4203.666.1.100
  NAME 'postfixUser'
  DESC 'Postfix Mail User'
  AUXILIARY
  MAY (mailacceptinggeneralid $ maildrop)
)

```

Entretanto, para atender a todos os requisitos sugerimos utilizar o schema de um outro MTA famoso: o Qmail.

Assim basta procurar na Internet, ou copiar do Apêndice 2 deste manual o qmailuser.schema e instalá-lo no OpenLDAP.

### *13.3 Instalando o schema*

Para instalar um novo schema no OpenLDAP basta copiá-lo para o diretório certo e fazer o “include” dele no arquivo de configuração do OpenLDAP.

1. Copie o qmailuser.schema para o diretório certo usando o comando abaixo:

```
cp qmailuser.schema /etc/ldap/schema
```

2. Edite o arquivo de configuração “slapd.conf” e procure pelas linhas de “include”. Logo abaixo da última “include” adicione isto:

```
include /etc/ldap/schema/ qmailuser.schema
```

3. Reinicie o OpenLDAP
4. É necessário adicionar novos atributos aos nossos usuários, para que estejam aptos a enviar e receber mensagens. Para isso vamos alterar todos os nossos usuários com um arqui-

```
vo "ldif" como este:
dn: cn="Fulano1 da Silva",ou=Usuarios,dc=anahuac,dc=org
add: objectClass
objectClass: qmailUser
-
add: mail
mail: fulano1@anahuac.org
-
add: accountStatus
accountStatus: 0
```

5. Faça um script para alterar todos os usuários

### *13.4 Configurando o Postfix padrão*

Para fazer com que o Postfix pesquise pelos usuários na base LDAP será necessário fazer algumas configurações no arquivo "main.cf", que é o seu arquivo de configuração.

1. Edite o arquivo /etc/postfix/main.cf
2. Procure pela opção "alias\_maps" e adicione o seguinte conteúdo a ela:

```
alias_maps = hash:/etc/aliases, ldap:accounts
```

#### Explicando

A linha original dessa opção é: `alias_maps = hash:/etc/aliases`. O que estamos fazendo é adicionando mais um "local" onde as contas podem ser localizadas. Neste caso estamos dizendo que se trata de uma pesquisa do tipo "ldap" em uma configuração chamada "accounts".

O nome da configuração é livre, ou seja, pode-se definir qualquer nome que melhor se adequa a realidade do seu ambiente.

3. No fim do arquivo adicione o seguinte conteúdo:

```
accounts_server_host = ldap://ldap__.anahuac.org
accounts_search_base = ou=usuarios,dc=anahuac,dc=org
```



```
accounts_query_filter = (&(uid=%u)(accountStatus=0))
accounts_result_attribute = uid
```

Explicando

Perceba que todas as linhas são iniciadas pelo nome da configuração. Se a escolha do nome da configuração for, por exemplo, “usuarios”, então todas as linhas iniciariam pelo nome “usuarios”.

- **linha 1** – Aqui é definido o “host” do servidor no qual a consulta será feita
- **linha 2** – Aqui é definido o “galho” onde a pesquisa deve ser feita. Pode-se definir a raiz da base LDAP, se necessário, mas dependendo da base LDAP pode ser mais conveniente e eficiente pesquisar em apenas um “galho”.
- **linha 3** – Filtro da pesquisa. Perceba que estamos usando um atributo do novo schema qmailuser.schema, o “accountStatus”. Se este atributo for diferente de “0” o usuário não receberá mensagens.
- **linha 4** – Atributo que deve ser retornado no caso da pesquisa ser efetuada com sucesso.

4. Reinicie o Postfix com o comando:

```
/etc/init.d/postfix restart
```

### *13.5 Testes de funcionamento*

Neste ambiente de configurações e testes, o envio da mensagem de testes irá falhar. Não por um problema de comunicação com o servidor OpenLDAP, mas porque os usuários da base LDAP não existem no sistema operacional.

Por padrão o Postfix só entrega mensagens na caixa postal dos usuários reais do sistema operacional. Apesar de termos feito as configurações para que a consulta de usuários fosse feita na base LDAP, esses usuários não existem localmente. Portanto a entrega de mensagens não pode acontecer, provocando uma mensagem de erro no Postfix.

Vejamos o erro:

1. Abra um terminal e monitore o arquivo de “log” do Postfix com o seguinte comando:

```
tail -f /var/log/mail.log
```

2. Abra um segundo terminal e envie uma mensagem com o seguinte comando:

```
echo “Teste de envio” | mail -s “Teste de envio” fulano1@anahuac.org
```

3. Verifique o erro encontrado no “log” do Postfix. Algo similar a isto:

```
Nov 22 16:00:54 server postfix/local[8493]: 9BD8410C3B: to=<fulano1@anahuac.org>, relay=local, delay=0.24, delays=0.13/0.02/0/0.09, dsn=5.1.1, status=bounced (unknown user: “fulano1”)
```

A mensagem de erro é clara: “unknown user” ou seja, usuário desconhecido. Pode-se pensar que esse erro deve-se a alguma falha na configuração do suporte a LDAP do Postfix, mas não é isso. O problema está na parcela de escrita da mensagem na caixa postal do usuário. Afinal de contas o usuário “fulano1” não existe no sistema, ele existe somente na base LDAP.

Para poder realizar nosso teste precisaremos adicionar um usuário local. Isso não será necessário em um ambiente de produção quando existir um DMA – Delivery Mail Agent para poder escrever a mensagem na caixa postal do usuário.

1. Execute o comando abaixo, para adicionar um usuário local:

```
useradd fulano1
```

2. Envie uma segunda mensagem para o usuário “fulano1”:

```
echo “Teste de envio” | mail -s “Teste de envio” fulano1@anahuac.org
```

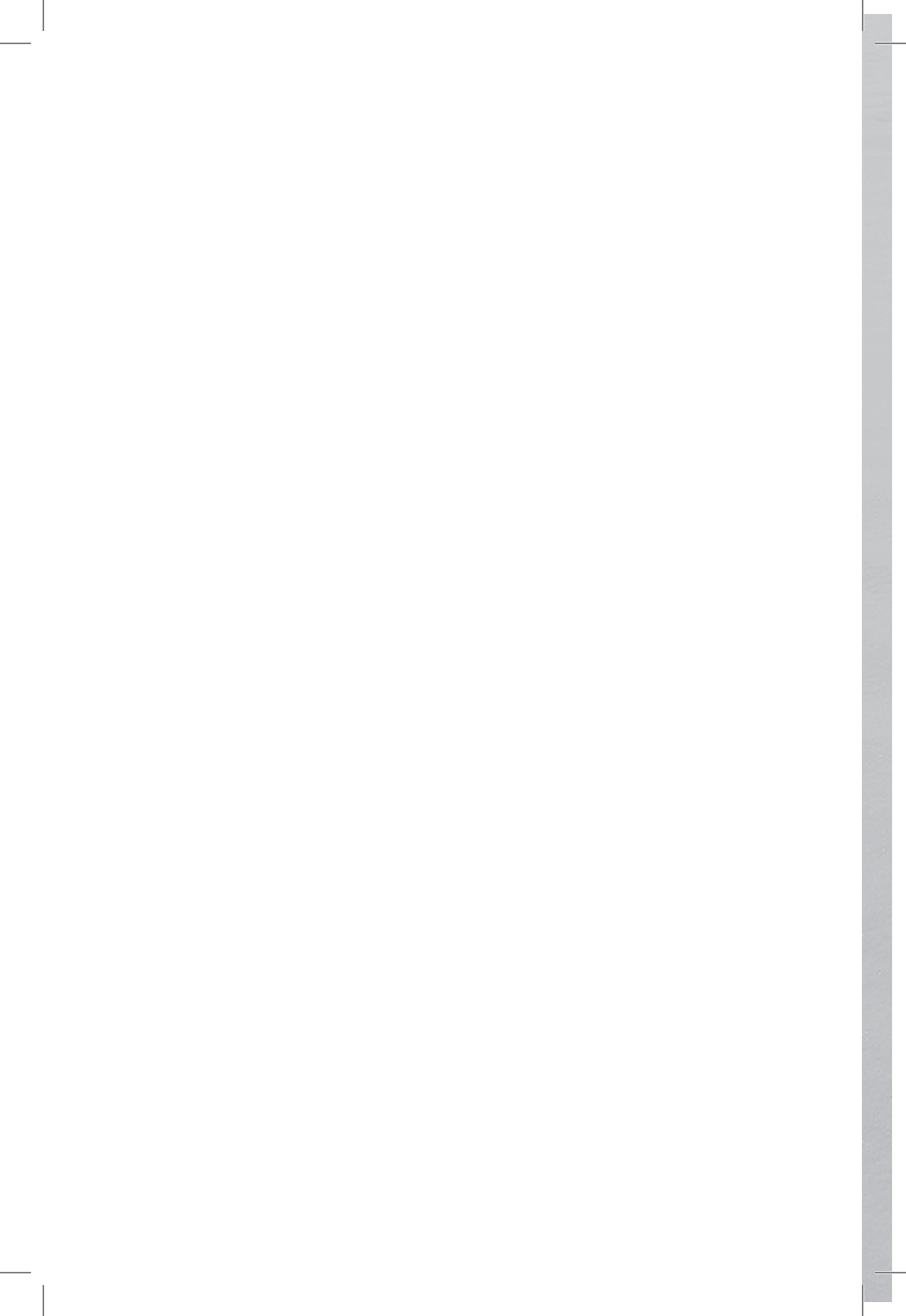
3. Verifique o “log” do Postfix. Deverá surgir uma mensagem de sucesso. Algo como isto:

Nov 22 16:12:34 server postfix/local[8506]: 15DA310C3B: to=<fulano1@anahuac.org>, relay=local, delay=0.08, delays=0.04/0.02/0/0.02, dsn=2.0.0, status=sent (delivered to mailbox)

Perceba que o “status” é: “sent (delivered to mailbox)”, ou seja, enviado e entregue com sucesso.

Mais adiante veremos a integração do Postfix com um servidor IMAP/POP que é também um DMA – Delivery Mail Agent. Assim não será necessário ter usuários reais no sistema.

4. Para finalizar remova o usuário local e siga para o próximo capítulo:  
userdel fulano1





CAPÍTULO 14

**SERVIDOR POP/IMAP**  
**COM CYRUS**



A implementação padrão, de referência, do protocolo IMAP é baseada no servidor Cyrus. O servidor Cyrus usa um método próprio de armazenamento de mensagens. Cada mensagem é armazenada em um arquivo. Os cabeçalhos das mensagens e o estado das mensagens (lido, por exemplo) são armazenados em uma base de dados.

Todas as mensagens são indexadas para melhor performance. O servidor Cyrus-IMAP é tido como umas das implementações mais rápidas e é apropriado para manuseio de grande volume de usuários e mensagens.

O Cyrus-IMAP usa o autenticador externo Cyrus-SASL, baseado na SASL (Camadas de Autenticação e Segurança Simples). O serviço “saslauthd” é responsável por atender aos pedidos de autenticação.

A tarefa de configurar a camada de autenticação para uso do OpenLDAP possibilita usos futuros da infraestrutura, como a implementação de SMTP autenticado.

### *14.1 Instalando e configurando suporte a SASL*

Uma vez que vamos utilizar o Cyrus com autenticação SASL é fundamental instalar e configurar esse serviço antes.

1. Instale o “saslauthd” com o seguinte comando:

```
aptitude install sasl2-bin
```

2. Edite o arquivo `/etc/default/saslauthd`

3. Altere a opção START para “yes”

4. Altere a opção MECHANISMS para “ldap”

5. Reinicie o “saslauthd” com o seguinte comando:

```
/etc/init.d/saslauthd restart
```

6. Crie o arquivo “`/etc/saslauthd.conf`” com o seguinte conteúdo:

```
ldap_servers: ldap://ldap__.anahuac.org
```

```
ldap_port: 389
```

```
ldap_version: 3
```

```
ldap_referrals: no
```

```
ldap_auth_method: bind
ldap_search_base: dc=anahuac,dc=org
ldap_filter: uid=%u
```

### Explicando

- **linha 1** – Define o “host” do servidor LDAP onde será feita a pesquisa
- **linha 2** – Define a porta de conexão com o servidor LDAP
- **linha 3** – Define a versão do protocolo que será utilizada
- **linha 4** – Define se o cliente poderá ou não seguir referências lógicas
- **linha 5** – Define o método de autenticação (bind | custom | fastbind)\*
- **linha 6** – Define o “galho” onde a pesquisa será realizada
- **linha 7** – Define o filtro de pesquisa
- A opção desconhecida na lista acima é a “ldap\_auth\_method”. Vamos entender as três opções:
- **bind** – Esta é a opção padrão e ela é utilizada para perguntar diretamente à base LDAP se o par usuário + senha coincidem;
- **custom** – Esta opção usa o atributo “userPassowrd” para validar o par usuário + senha. Ele suporta os “hashes”: crypt, md5, smd5, sha e ssh. Além de texto plano.
- **fastbind** – É igual à opção “bind” mas utiliza o DN completo para validação.

7. Utilize o comando “testsaslauthd” para testar o SASL. Abaixo um exemplo:

```
testsaslauthd -u fulano1 -p fulano1
```

Perceba que a opção “-u” define o usuário e a opção “-p” define a senha do usuário. O resultado do comando é auto-explicativa e indicará se a conexão entre o SASL e o LDAP foi ou não realizada com sucesso.



## 14.2 Instalando o Cyrus

1. Para instala o Cyrus execute o seguinte comando:

```
aptitude install cyrus21-pop3d cyrus21-imapd
cyrus21-admin
```

## 14.3 Configurando o Cyrus

1. Edite o arquivo de configuração do Cyrus, `/etc/imapd.conf`
2. Descomente a opção “admins”
3. Descomente a opção “sasl\_mech\_list”
4. Descomente e defina a opção “sasl\_pwcheck\_method” para “saslauthd”
5. Reinicie o Cyrus utilizando o seguinte comando:

```
/etc/init.d/cyrus21 restart
```

## 14.4 Configurando o Postfix para usar LMTP

Para que o Postfix possa entregar as mensagens ao Cyrus é necessário configurá-lo.

1. Edite o arquivo de configuração `/etc/postfix/main.cf` e adicione o conteúdo abaixo:

```
mailbox_transport = lmtp:unix:/var/run/cyrus/socket/lmtp
```

2. É necessário “desenjaular” o serviço “lmtp”, que por padrão vem enjaulado no Postfix. Para fazer isso edite o arquivo `/etc/postfix/master.cf` e procure pela opção “lmtp”. Altere a configuração para que fique assim:

```
lmtp  unix - - n - - lmtp
```

Perceba que a alteração está na troca do caracter “-” pelo caracter “n”.

3. Será necessário criar o grupo “lmtp”, adicioná-lo ao grupo postfix e depois ajustar às permissões do arquivo de “SOCKET” do Cyrus. Para isso execute:

```
dpkg-statoverride --remove /var/run/cyrus/socket
dpkg-statoverride --add cyrus postfix 750 /var/run/cyrus/
socket
```

4. Reinicie o Cyrus e o Postfix com o comando:

```
/etc/init.d/cyrus21 restart
/etc/init.d/postfix restart
```

## ***14.5 Criando contas no Cyrus***

Antes de poder criar as contas no Cyrus será necessário criar o usuário “cyrus” na base LDAP. Esse é o usuário administrativo do Cyrus.

1. Crie um arquivo chamado “cyrus.ldif” com o seguinte conteúdo:

```
dn: cn=cyrus,dc=anahuac,dc=org
cn: cyrus
objectClass: organizationalRole
objectClass: posixAccount
objectClass: simpleSecurityObject
uid: cyrus
uidNumber: 1010
gidNumber: 1010
homeDirectory: /home/cyrus
userPassword: 123456
description: Cyrus Admin user
```

2. Adicione o conteúdo do arquivo “ldif” com o comando:

```
ldapadd -h ldap__.anahuac.org -p 389 -x -D cn=admin,dc=anahuac,dc=org -w senha -f cyrus.ldif
```

3. Para adicionar as contas no Cyrus utilize os comandos abaixo:

```
cyradm --user cyrus ldap__.anahuac.org
IMAP Password:
```

```
ldap__.anahuac.org> cm user.fulano1
ldap__.anahuac.org> cm user.fulano1.SENT
ldap__.anahuac.org> cm user.fulano1.TRASH
ldap__.anahuac.org> cm user.fulano1.SPAM
ldap__.anahuac.org> quit
```

## *14.6 Testes de funcionamento*

Para saber se tudo está funcionando como deveria vamos enviar uma mensagem ao usuário `fulano1@anahuac.org` e ver se a mensagem está acessível pelo protocolo POP.

1. Abra um terminal e monitore o arquivo de “log” do Postfix:  
`tail -f /var/log/mail.log`

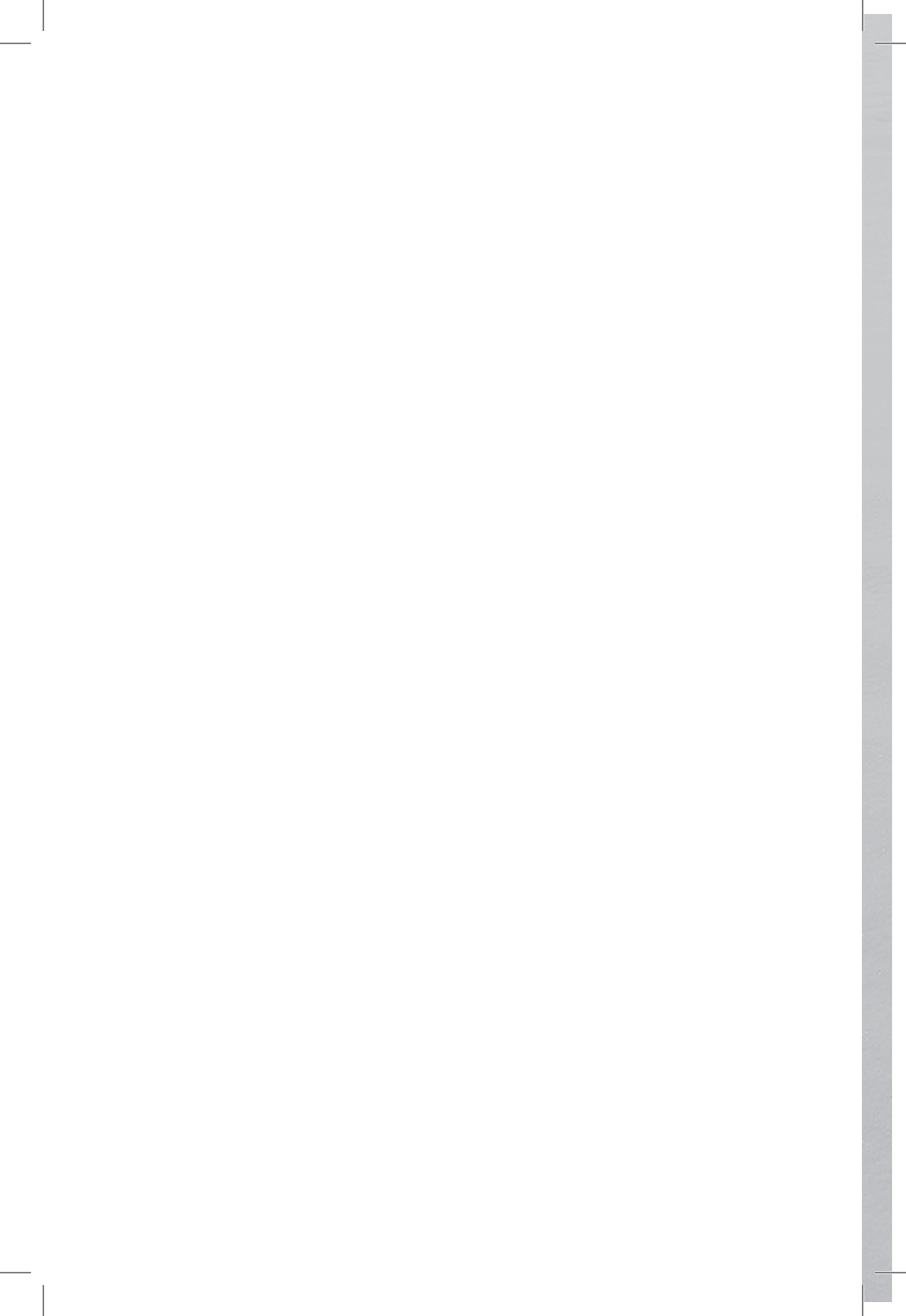
2. Abra um segundo terminal e envie uma mensagem para o usuário `fulano1@anahuac.org`:

```
echo “Teste final” | mail -s “Teste final” fulano1@anahuac.org
Procure pela mensagem de sucesso, ou seja, “status=sent”
```

3. No mesmo terminal utilize o comando “telnet” para se conectar na porta 110 que é a porta utilizada pelo protocolo POP3:

```
telnet ldap__.anahuac.org 110

user fulano1
pass fulano1
list
retr 1
quit
```





CAPÍTULO 15

SERVIDOR SAMBA



Segundo a Wikipédia “Samba é um programa de computador, utilizado em sistemas operacionais do tipo Unix, que simula um servidor Windows, permitindo que seja feito gerenciamento e compartilhamento de arquivos em uma rede Microsoft.”

### ***15.1 Conceituação sobre o Samba e seu modo de autenticação***

Uma das características mais controversas do Samba é sua exigência de ter dois usuários: um no sistema operacional e outro do próprio “Samba”. Mas por que são necessários os dois usuários?

O primeiro motivo se refere ao fato do usuário Windows possuir mais atributos que um usuário Unix/Gnu/Linux tradicional.

O segundo motivo é que a base de senhas do SAMBA usa técnicas distintas de criptografia e armazenamento de senhas.

Outro grande motivo é que nem sempre queremos que exista a possibilidade de usuários SAMBA terem permissão de login em máquinas Gnu/Linux e vice-versa. (por exemplo, os usuários criados apenas para executar serviços como www, cramd).

Resumindo: O usuário SAMBA é sempre usado pelo servidor para autenticar alguém. O usuário Gnu/Linux correspondente é usado pelo servidor para fornecer credenciais de acesso a um recurso ou autorizar o usuário a acessar.

O SAMBA poderia optar por usar apenas um usuário especial no Gnu/Linux e controlar totalmente o acesso a arquivos e pastas usando recursos internos. Mas isso não é feito dessa forma. Uma das desvantagens claras dessa abordagem é que a segurança das pastas pessoais em uma rede mista não poderia ser facilmente administrada. Com apenas um dono, existiriam grandes problemas.

Além disso a abordagem de uso dos usuários registrados no Sistema Operacional em uso tem uma grande vantagem. Torna o SAMBA mais adaptável para uso em múltiplas plataformas.

Em ambientes onde não se deseja ter nenhum usuário local para validação no SAMBA, pode-se configurar o Gnu/Linux

para espelhar todos os usuários locais em uma base LDAP. Isso é feito através da configuração dos módulos de autenticação PAM. Dessa forma os usuários do SAMBA e do sistema operacional passam a ser um só, ou seja, não há a necessidade de ter dois usuários, um do SAMBA e um do sistema operacional.

Apesar de ter um único usuário, nesse modelo, o SAMBA continua entendendo que são dois diferentes. A configuração do PAM será vista em um capítulo mais adiante.

Outro aspecto importante é a autenticação. O cliente envia um par usuário/senha. O servidor SAMBA usa este par para verificar se o usuário existe na base de usuários e se a senha está correta. Caso ocorra sucesso, o cliente é considerado como autenticado e suas credenciais serão usadas para as rotinas de Autorização.

Essa etapa corresponde ao “logon” em uma estação Windows ou o “logon” em um cliente SAMBA em Gnu/Linux.

Nessa etapa é possível ao cliente saber quais são os recursos que teoricamente estariam disponíveis a ele. Teoricamente por que o fato de um recurso ser visível e até mesmo reservado para uso para determinado usuário não significa que o usuário está autorizado a acessá-lo, as autorizações serão testadas no momento de acesso ao recurso. Assim, no ambiente de rede pode ser possível ver um compartilhamento e ainda assim a tentativa de acesso pode resultar em Acesso Negado.

O cliente autenticado tem seu status armazenado e quando necessita de um recurso da Rede, o servidor SAMBA verifica se esse cliente tem os direitos correspondentes ao recurso.

O controle de acesso ao recurso pode ser feito em dois locais:

No servidor SAMBA: Usando cláusulas de permissão e negação explícitas aos recursos como “valid users” e “invalid users” ou liberando acesso público e anônimo com “guest ok”;

No Sistema Operacional onde o SAMBA foi instalado: Esse é o método preferencial. O Samba respeita e usa os recursos de permissão e direitos do sistema operacional em uso. Assim no



caso do Gnu/Linux, as permissões de arquivos serão respeitadas pelo SAMBA.

Dessa forma, como regras de ouro: O primeiro lugar onde deve-se procurar por problemas nas permissões de recursos é no próprio sistema de arquivos usando comandos como “ls -la” e “getfacl”. Mantenha o smb.conf limpo e trabalhe as permissões no sistema de arquivos. NÃO use “valid users” ou “invalid users”.

Com o recurso de ACL corretamente configurado, use o “Windows Explorer” para o trabalho de manutenção de permissões.

Caso não seja possível deixar de usar “invalid users” e “valid users”, nunca ofereça compartilhamentos totalmente abertos no sistema de arquivos e controlados por cláusulas SAMBA sem garantir que não há possibilidade de uso desses recursos por outro caminho (um login remoto diretamente no servidor).

A necessidade das etapas de Autenticação e de Autorização, junto com os mecanismos de troca e forma de senhas do protocolo CIFS/SMB, explicam a necessidade da adição de usuários ser feita tanto no Samba como no Gnu/Linux.

## ***15.2 Instalando o Samba***

O comando acima instala os pacotes SAMBA necessários:

- Pacote de administração SAMBA +OpenLDAP (smbldap-tools);
- Pacote de autenticação Gnu/Linux na base OpenLDAP(libnss-ldap);
- Documentação SAMBA (samba-doc);
- Utilitários de configuração de listas de controle de acesso no sistema de arquivos(acl).

Não é necessário se preocupar em responder corretamente as questões que serão apresentadas nesta etapa. Os servidores serão configurados manualmente, desta forma, os princípios de configuração podem ser melhor adaptados a outras distribuições.

Para instalar o Samba execute o comando abaixo:

```
aptitude install samba smbclient smbldap-tools samba-doc acl
```

### ***15.3 Samba como PDC***

O Samba pode armazenar sua base de usuários em um servidor de serviços de diretórios OpenLDAP. A grande vantagem é que com a solução integrada SAMBA + OpenLDAP a tarefa de administração de usuários é bastante facilitada, pois o mapeamento de grupos e usuários é automático e a adição de usuários ao Gnu/Linux e à base de usuários SAMBA é feita simultaneamente, com sincronização de senhas.

Além disso, atualmente, a integração Samba + OpenLDAP é a única maneira prática para implementação de uma solução com vários servidores entre matriz e filiais que compartilham a mesma base de usuários.

Nesta solução, tanto os usuários do Gnu/Linux quanto os usuários SAMBA serão armazenados em uma base OpenLDAP.

A distribuição de exemplo é Debian Etch, contudo os princípios são os mesmos para todas as distribuições, com eventuais mudanças nos nomes de arquivos e/ou locais de armazenamento.

### ***15.4 Instalando o schema do Samba no OpenLDAP***

Será necessário ativar o “schema” do samba para que o OpenLDAP seja capaz de utilizar seus atributos e portanto armazenar usuários que sirvam para validação em uma rede Microsoft.

1. Copie o arquivo samba.schema, disponível no pacote samba-doc, com o comando abaixo:

```
zcat /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz > /etc/ldap/schema/samba.schema
```

2. Edite o arquivo de configuração do OpenLDAP, “slapd.conf” e insira a linha abaixo logo depois das opções “include” já existentes:

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

3. Procure pela opção ”index” e adicione logo abaixo dela:

```
index memberUID,mail,givenname eq
```

```
index sambaSID,sambaPrimaryGroupSID,sambaDomain-  
Name eq
```

4. Procure pela opção “access to attrs=userPassword” e na mesma linha, adicione à lista de atributos o seguinte:  
`,sambaLMPassword,sambaNTPassword`

O resultado final dessa linha deve ser:  
`access to attrs=userPassword,shadowLastChange,sambaLM-  
Password,sambaNTPassword`

5. Pare o OpenLDAP com o comando:

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

6. Indexe a base com o comando:

```
slapindex -v
```

7. Corrija possíveis erros de permissionamento com o comando:

```
chown openldap: /var/lib/ldap/*
```

8. Inicie o OpenLDAP com o comando abaixo:

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

## ***15.5 Configurando o Samba para usar LDAP***

As linhas de automação com uso do `smbldap-tools` são essenciais para o processo de adição de usuários por ferramentas Windows como o User Manager for Domains. Além disso, o processo de adição de máquinas ao domínio é automatizado.

Para facilitar vamos oferecer aqui um `smb.conf` pronto para o funcionamento. Estes parâmetros devem ser copiados no `smb.conf` original que estiver em produção respeitando as possíveis diferenças.

1. Crie uma cópia de segurança do arquivo de configuração original do Samba com o comando:

```
mv /etc/samba/smb.conf /etc/samba/smb.conf.original
```

2. Crie um arquivo de configuração novo, chamado `/etc/samba/smb.conf` e adicione o seguinte conteúdo:

```
[global]
workgroup = labopenldap
netbios name = LABSRV
username map = /etc/samba/smbusers
add user script = /usr/sbin/smbldap-useradd -m -a "%u"
delete user script = /usr/sbin/smbldap-userdel "%u"
add group script = /usr/sbin/smbldap-groupadd -p "%g"
delete group script = /usr/sbin/smbldap-groupdel "%g"
add user to group script = /usr/sbin/smbldap-groupmod -m
"%u" "%g"
delete user from group script = /usr/sbin/smbldap-group-
mod -x "%u" "%g"
set primary group script = /usr/sbin/smbldap-usermod -g
"%g" "%u"
add machine script = /usr/sbin/smbldap-useradd -a -w "%u"
domain logons = Yes
preferred master = Yes
wins support = Yes
passdb backend = ldapsam:ldap://ldap__.anahuac.org
ldap suffix = dc=anahuac,dc=org
ldap machine suffix = ou=Usuarios
ldap user suffix = ou=Usuarios
ldap group suffix = ou=Grupos
ldap admin dn = cn=admin,dc=anahuac,dc=org
```

3. Informe ao Samba a senha do usuário “admin” do OpenL-DAP com o comando:

```
smbpasswd -w senha
```

4. Crie um arquivo de usuários do samba com o comando abaixo:

```
> /etc/samba/smbusers
```

5. Reinicie o Samba com o comando:

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

6. Anote a identificação do domínio. Ela será necessária para configurar o “smbldap-tools”. Para ver a identificação do domínio execute o seguinte comando:

```
net getlocalsid
```

## 15.6 Configurando o *smbldap-tools*

Para que as ferramentas do “smbldap-tools” funcionem como o esperado, se faz necessário configurá-las. Veja a seguir:

1. Copie os arquivos `smbldap_bind.conf` e `smbldap.conf`

```
zcat /usr/share/doc/smbldap-tools/examples/smbldap.conf.gz
```

```
>/etc/smbldap-tools/smbldap.conf
```

```
cp /usr/share/doc/smbldap-tools/examples/smbldap_bind.conf
```

```
/etc/smbldap-tools/smbldap_bind.conf
```

2. Edite o arquivo `/etc/smbldap-tools/smbldap_bind.conf` e altere-o para que fique com as opções abaixo:

```
slaveDN="cn=admin,dc=anahuac,dc=org"
```

```
slavePw="senha"
```

```
masterDN="cn=admin,dc=anahuac,dc=org"
```

```
masterPw="senha"
```

3. Edite o arquivo `/etc/smbldap-tools/smbldap.conf`, modificando apenas as linhas citadas abaixo:

```
# A credencial do domínio
```

```
SID="S-1-5-21-3420362730-2273518020-356174992"
```

```
# Nome do domínio do SAMBA
```

```
sambaDomain="LABOPENLDAP"
```

```
#A raiz do servidor OpenLDAP
```

```
suffix="dc=anahuac,dc=org"
```

```
#Onde armazenar
usersdn="ou=Usuarios,${suffix}"
computersdn="ou=Usuarios,${suffix}"
groupsdn="ou=Grupos,${suffix}"
idmapdn="ou=Idmap,${suffix}"
sambaUnixIdPooldn="sambaDomainName=labopenl-
dap,${suffix}"
```

```
#Configuração padrão para pastas, mapeamento e script de
logon
```

```
userSmbHome="\\%L\homes\%U"
userProfile="\\%L\profiles\%U"
userHomeDrive="P:"
userScript="%U.bat"
mailDomain="anahuac.org"
```

```
#Configurando o suporte a TLS
cafile="/etc/ldap/tls/cacert.pem"
clientcert="/etc/ldap/tls/srvcert.pem"
clientkey="/etc/ldap/tls/srvkey.pem"
```

4. O `smbldap-tools` se encarrega de criar a base LDAP necessária para execução da solução. Para tanto, execute o comando:

```
smbldap-populate -u 2000 -g 2000
```

**OBSERVAÇÃO:** Se o SID (o identificador do domínio) não estiver devidamente configurado, os recursos adicionados não serão válidos para o domínio em uso. Os parâmetros `-u` e `-g` servem para indicar o uid e gid mínimos usados pelas ferramentas `smbldap-tools`.

O `smbldap-adduser` faz parte do conjunto de ferramentas `smbldap-tools`. Sem essas ferramentas, as vantagens no processo de administração de uma Solução SAMBA+OpenLDAP seriam muito menores. Por exemplo, a criação prévia de um usuário

Gnu/Linux antes da adição de um usuário ou estação seria necessária, usando ferramentas distintas para cada etapa.

### *15.7 Testes de funcionamento*

Usar o comando “smbldap-populate” é eficiente, somente se o objetivo for migrar os usuários existentes no Sistema Operacional para a base LDAP. É claro que ele também ajuda na criação dos grupos. Entretanto se o objetivo for validar os usuários já existentes na base LDAP ele não será efetivo.

Isso se deve a falta dos atributos necessários nos usuários existentes para poder realizar essa validação. O que deve ser feito neste caso é adicionar os atributos exigidos pelo Samba aos usuários existentes na base LDAP.

Para adicionar a habilidade de validar no Samba aos usuários “posix” existentes teremos que realizar algumas etapas:

1. Adicionar o usuário e o seu grupo ao Sistema Operacional:

```
useradd -s /bin/false fulano1  
groupadd fulano1
```

2. Adicionar os atributos necessários ao usuário e grupo “fulano1” para que ele possa se autenticar no Samba:

```
smbldap-usermod -a fulano1  
smbldap-groupmod -a fulano1
```

3. Também será necessário redefinir sua senha. Como estamos usando a senha igual ao uid, então:

```
(echo fulano1 ; echo fulano1) | smbldap-passwd fulano1
```

Um pequeno truque para não ter que digitar a senha duas vezes :-)

4. Agora sim podemos testar nossa validação. Para isso basta usar o comando abaixo:

```
smbclient -L localhost -U fulano1%fulano1
```

Este comando faz uma busca pelos compartilhamentos no servidor indicado pela opção “-L”. Já a opção “-U” define usuário%senha

Se a validação acontecer com sucesso então lhe será mostrado algo parecido com isto:

```
Domain=[LABOPENLDAP] OS=[Unix] Server=[Samba
3.0.24]
```

| Sharename | Type | Comment                    |
|-----------|------|----------------------------|
| -----     | ---- | -----                      |
| IPC\$     | IPC  | IPC Service (Samba 3.0.24) |

```
Domain=[LABOPENLDAP] OS=[Unix] Server=[Samba
3.0.24]
```

| Server      | Comment      |
|-------------|--------------|
| -----       | -----        |
| LABSRV      | Samba 3.0.24 |
| Workgroup   | Master       |
| -----       | -----        |
| LABOPENLDAP | LABSRV       |

Caso contrário a mensagem de retorno será:

```
session setup failed: NT_STATUS_LOGON_FAILURE
```

Visando facilitar o processo de alteração de diversos usuários, que tal fazer um script?

1. Crie um arquivo chamado “para\_smb.sh” e insira o seguinte conteúdo:

```
#!/bin/bash
```

```
ROOTDN=”cn=admin,dc=anahuac,dc=org”
```

```
ROOTPW=”senha”
```

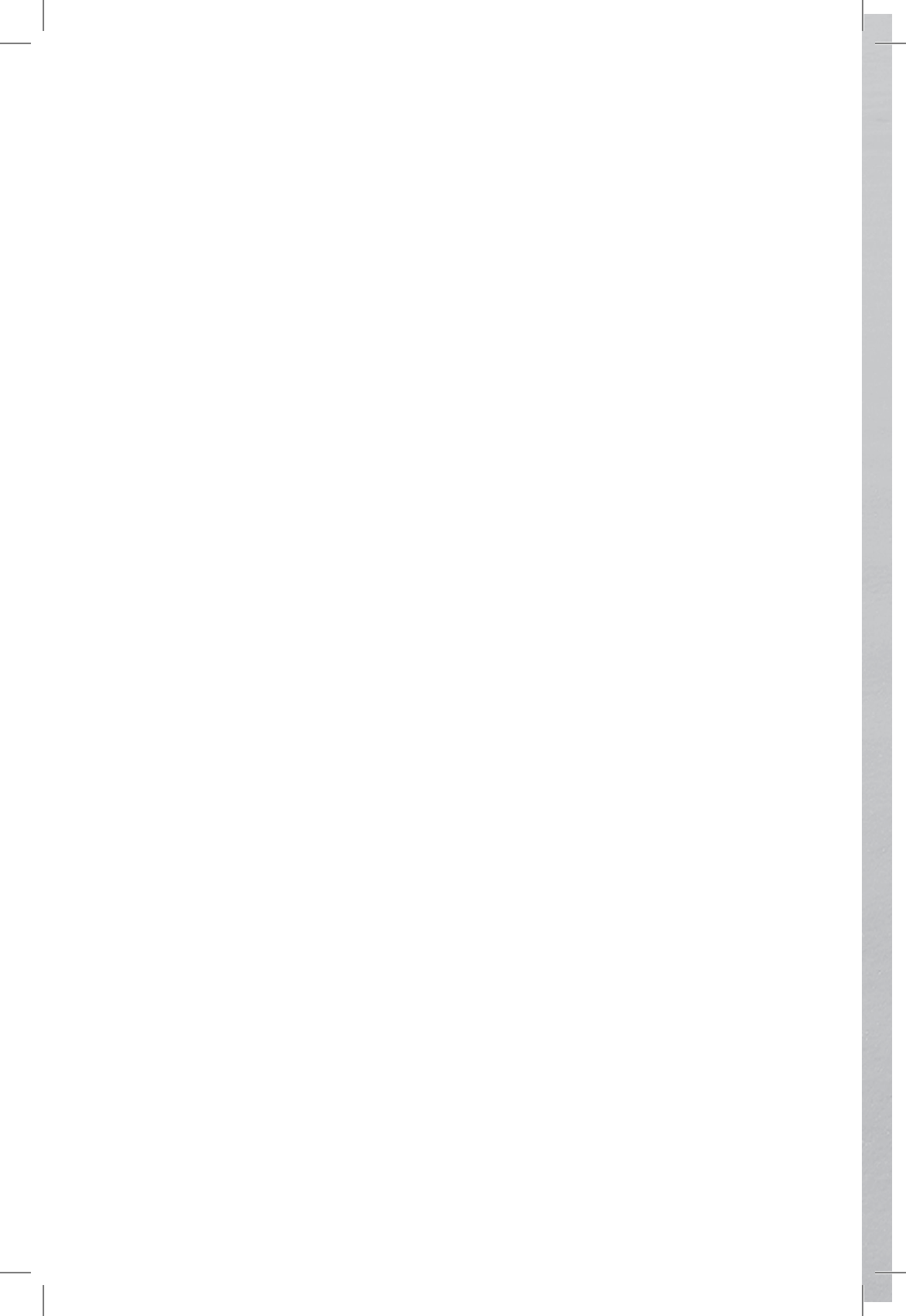
```
SUFFIX=”dc=anahuac,dc=org”
```

```
ALL_USERS=`ldapsearch -x -b “ou=Usuarios,$SUFFIX”
-LLL uid | grep uid: | cut -d” “ -f2`
```



```
for EACH in $ALL_USERS ; do
    SMB_PASS="$EACH"
    # Adiciona usuários locais
    useradd -s /bin/false $EACH
    groupadd $EACH
    # Adiciona os atributos necessários nas contas unix para
que elas sejam, também
    contas do Samba
    smbldap-usermod -a $EACH
    smbldap-groupmod -a $EACH
    # Altera a senha nos atributos do samba e do unix
    (echo $SMB_PASS ; echo $SMB_PASS) | smbldap-pas-
swd $EACH
done
```

2. Agora basta executar o script com o comando:  
sh para\_smb.sh





CAPÍTULO 16

NSS E PAM –  
AUTENTICANDO O SISTEMA  
NO LDAP



Em algumas situações não é o bastante autenticar apenas usuários de algum serviço instalado. Pode ser necessário autenticar todos os usuários do sistema em uma base LDAP. Para situações assim pode-se configurar o sistema de autenticação do Sistema Operacional para validar contas em uma base LDAP.

A autenticação de um usuário necessita de duas etapas:

- Busca pelo nome de usuário e seu ID (uid);
- Autenticação propriamente dita, confrontando-se o par usuário/senha fornecido com o par armazenado.

A primeira etapa envolve o uso de bibliotecas NSS -- Name Server Switches -- que se encarregam de fazer a busca nas bases de usuários especificadas. O módulo “nss\_ldap” permite que sistemas GNU/Linux façam a busca de usuários do sistema em bases OpenLDAP.

A segunda etapa é usada para aplicação de políticas relacionadas a autenticação (verificar se o diretório pessoal existe, por exemplo). Em sistemas GNU/Linux os módulos do PAM -- Pluggable

Authentication Modules -- são responsáveis por essa etapa. O módulo “libpam\_ldap” é o responsável pela integração do PAM com o OpenLDAP.

Ambas as etapas são realizadas por módulos permitindo que sistemas Gnu/Linux utilizem uma grande gama de serviços de armazenamento de usuários. São bastante usados sistemas baseados em Gerenciadores de Bancos de Dados (MySQL, por exemplo) e sistemas baseados em serviços de diretórios (OpenLDAP, por exemplo).

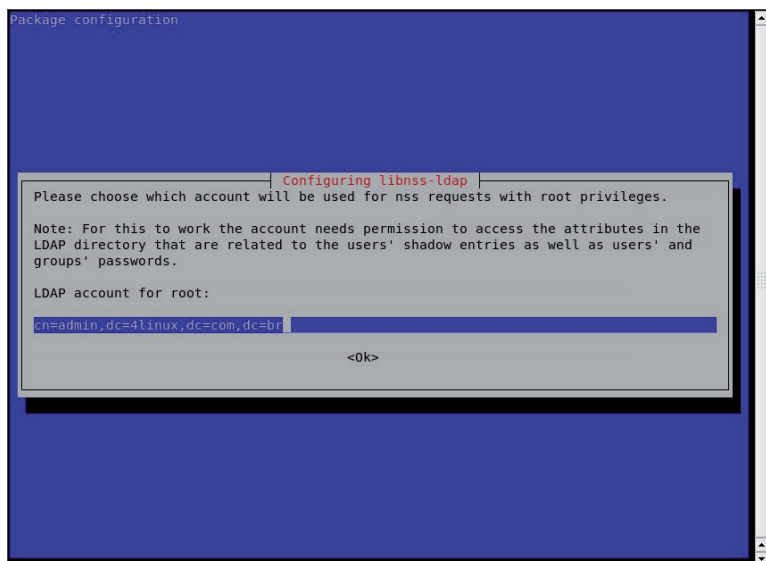
Na realidade o módulo “libpam\_ldap” não é necessário para login, mas é fundamental para que ferramentas funcionem corretamente. É o caso do utilitário de troca de senhas. Outro bom motivo é que sem o uso do “libpam\_ldap”, o método de criptografia a ser usado é obrigatoriamente o {CRYPT}, considerado por muitos como ultrapassado e passível de quebra em processadores contemporâneos.

Usando “libpam\_ldap” pode-se usar os mecanismos de criptografia entendidos pelo sistema OpenLDAP. Veja em: <http://www.debianplanet.org/node.php?id=1048>

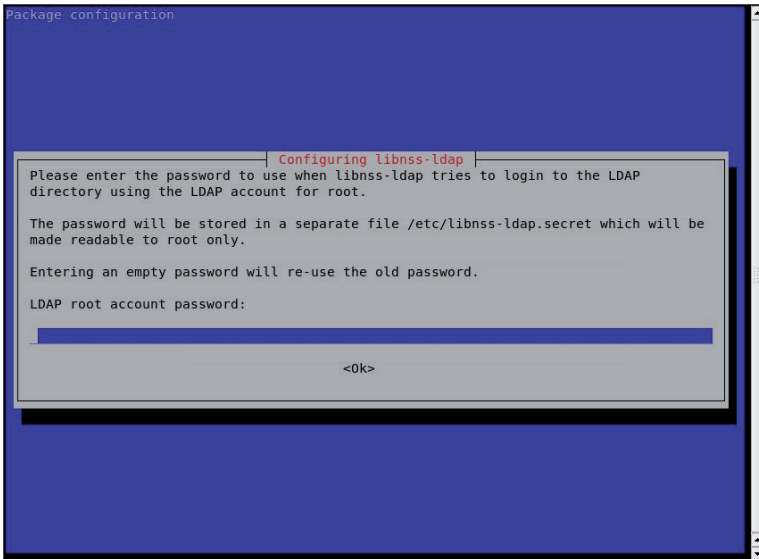
Praticamente todos os serviços que utilizem o sistema de autenticação interna do GNU/Linux podem se beneficiar do uso do OpenLDAP. A aplicação de políticas de restrição de usuários (uma regra baseada em horários, por exemplo) é geralmente realizada por módulos PAM.

## *16.1 Instalando e configurando o NSS - Name Server Switches*

1. Para instalar o NSS execute o seguinte comando:  
aptitude install libnss-ldap
2. Na tela de configuração preencha com o DN completo do seu usuário “admin”:



3. Na tela seguinte coloque a senha do usuário “admin”:



4. Altere o arquivo “/etc/nsswitch.conf” para que ele busque as informações de usuários e senhas na base OpenLDAP:

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

OBSERVAÇÃO: Quando se instala o “libnss-ldap” usando o aptitude também é instalado o programa “nscd”. Este aplicativo funciona como um cache para autenticação, mantendo informações na memória. Apesar de sua função ser agilizar o processo de busca sua presença traz mais instabilidade do que algum possível benefício. Portanto sugerimos que ele seja removido. Sendo assim, “aptitude purge nscd”

5. Confira o arquivo “/etc/libnss-ldap.conf”, que já deve ter sido configurado pelo “debconf”, e confira se as seguintes diretivas estão corretas:

```
uri ldap://ldap__.anahuac.org/  
base dc=anahuac,dc=org
```

ldap\_version 3

Se tudo estiver correto já deve ser possível listar o conteúdo do arquivo “passwd”, por exemplo e, ver as entradas da base OpenLDAP.

6. Liste o conteúdo do arquivo “/etc/passwd” com o comando abaixo:

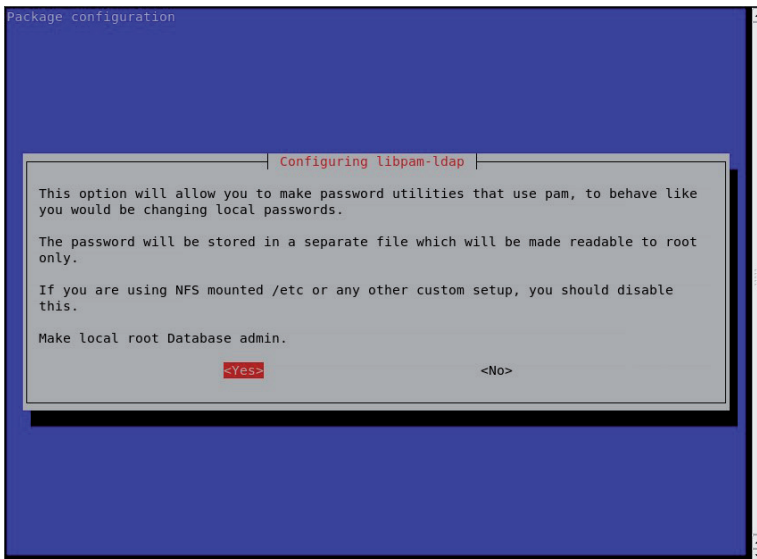
```
getent passwd
```

## *16.2 Instalando e configurando o PAM - Pluggable Authentication Modules*

1. Instale o libpam-ldap com o comando abaixo:

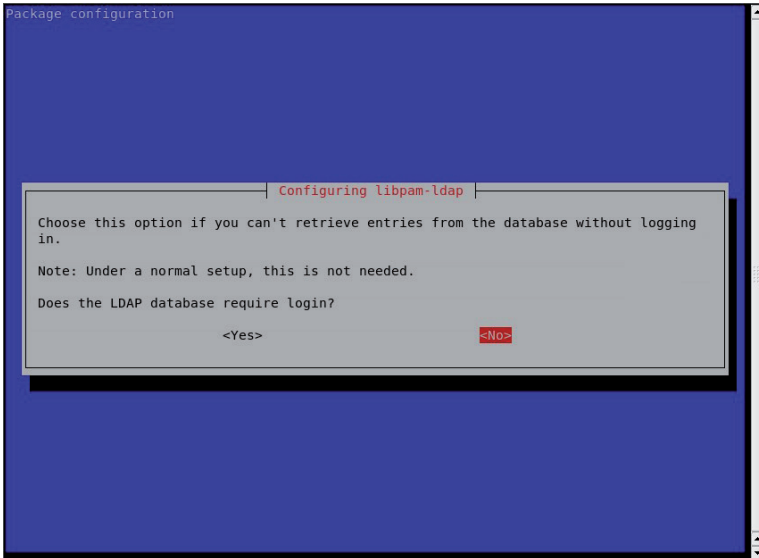
```
aptitude install libpam-ldap
```

2. A primeira tela de configuração pergunta se desejamos fazer o usuário “root local o administrados base. A resposta é sim:

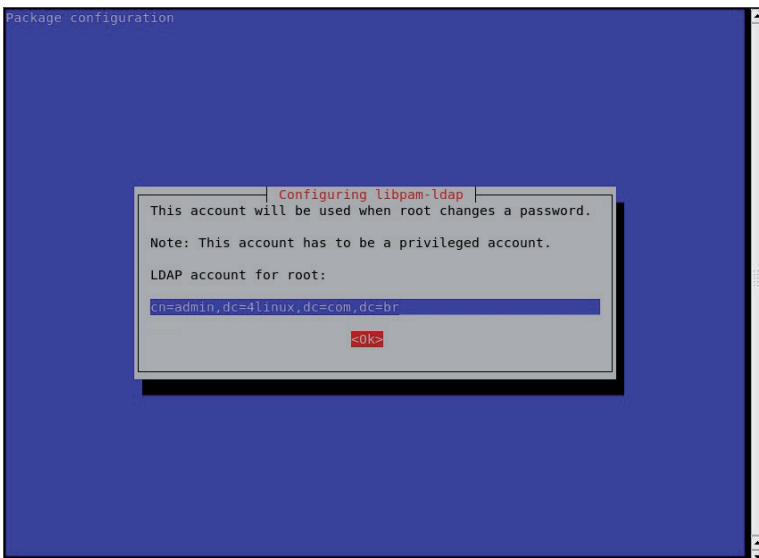




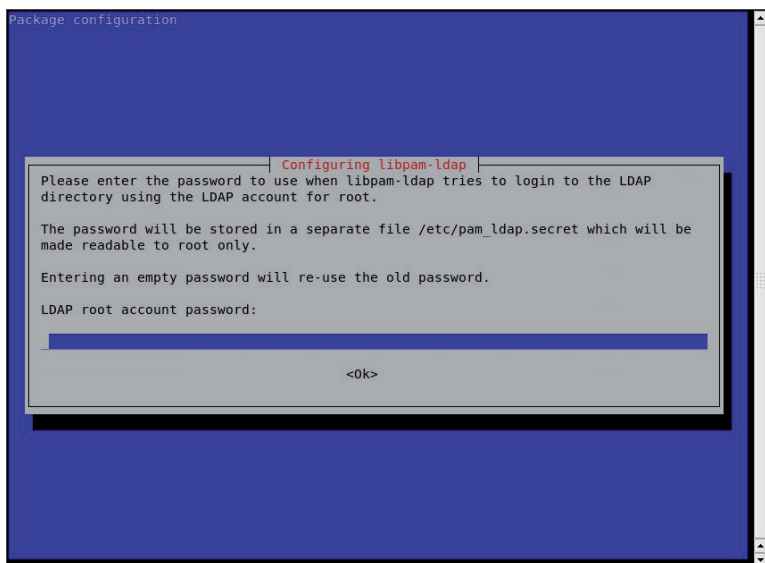
3. A segunda tela pergunta se é necessária autenticação na base de dados para uma simples pesquisa? A resposta padrão é não:



4. Na terceira tela informe a conta do usuário administrador da base para eventuais trocas de senhas.



5. Na quarta e última tela devemos fornecer a senha do usuário “admin”



O PAM atua nos quatro estágios da autenticação: session, account, authentication e password. Como queremos que nossas alterações tenham efeito nos quatro estágios e para todos os programas que suportam PAM, precisamos fazer as alterações nos arquivos comuns a todos os aplicativos, ou seja: common-auth, common-account, common-session e common-password.

6. Edite o arquivo “/etc/pam.d/common-account” e adicione o suporte ao OpenLDAP. Deixe ele assim:

```
account sufficient    pam_ldap.so
account required     pam_unix.so
```

7. edite o arquivo “/etc/pam.d/common-auth” para que seja suficiente a autenticação via base OpenLDAP. Deixe ele assim:

```
auth sufficient     pam_ldap.so
auth required       pam_unix.so nullok_secure
```

A opção “try\_first\_pass” tenta usar a mesma senha que o usuário já digitou em um módulo anterior, caso essa senha não tenha sido validada ele pede a senha do usuário novamente. Para obter maiores informações sugere-se a leitura da RFC86.0.

8. Edite o arquivo “/etc/pam.d/common-password” e adicione o suporte ao OpenLDAP. Deixe ele assim:

```
session required pam_unix.so nullok obscure min=4 max=8 md5
password required pam_ldap.so try_first_pass
```

9. Edite o arquivo “/etc/pam.d/common-session” e adicione o suporte ao OpenLDAP. Deixe ele assim:

```
session sufficient pam_ldap.so
session required pam_unix.so
```

10. Verifique as configurações do arquivo “/etc/pam\_ldap.conf” para ter certeza de que o “debconf” as fez corretamente:

```
base dc=anahuac,dc=org
uri ldap://ldap__.anahuac.org/
ldap_version 3
rootbinddn cn=admin,dc=anahuac,dc=org
pam_password crypt
```

### ***16.3 Teste de funcionamento***

Para testar a autenticação no sistema operacional basta abrir uma nova sessão ou executar uma conexão via “ssh” na própria máquina.

Entretanto é provável que, apesar de validar corretamente, o usuário não consiga ter acesso ao sistema. E se tiver será apresentado um erro. Algo como:

```
“Could not chdir to home directory /home/fulano1: No such
file or directory”
```

Isso acontece porque o diretório “HOME” do usuário não existe. Há duas forma de se lidar com esse problema:

1. Criar o diretório “home” do usuário manualmente. Algo como:

```
mkdir /home/fulano1
```

```
chown fulano1: /home/fulano1 -R
```

2. Pode-se instalar um programa que monitora as tentativas de autenticação e no caso dela ser bem sucedida e do diretório “home” não existir ele o cria. O nome deste programa é “Autodir”. para instalar o “Autodir” execute o seguinte comando:

```
aptitude install autodir
```

3. Edite o arquivo “/etc/default/autodir” e altere a opção “RUN\_AUTOHOME” de “no” para “yes”

4. Inicie o Autodir com o comando:

```
/etc/init.d/autodir start
```

5. Finalmente faça uma tentativa de “login” no seu servidor utilizando o usuário “fulano1”.



CAPÍTULO 17

**SERVIDOR FTP**



Os servidores de FTP mais populares no mundo “posix” são o VSFTP – Very Secure FTP e o ProFTP. Neste capítulo vamos ver como configurar os dois para utilizar a base OpenLDAP para autenticação.

Os dois utilizam usuários reais do sistema operacional para funcionar. Entretanto, como foi visto no capítulo anterior, podemos fazer todo o sistema operacional utilizar o OpenLDAP para validar usuários. Assim a validação será feita pelo PAM.

### *17.1 Instalando o “vsftp”*

1. Para instalar o “vsftp” execute o comando abaixo:

```
aptitude install vsftpd
```

### *17.2 Configurando vsftp*

1. Edite o arquivo “/etc/vsftpd.conf” e verifique as opções abaixo:

```
listen=YES
```

```
anonymous_enable=YES
```

```
local_enable=YES
```

```
dirmessage_enable=YES
```

```
xferlog_enable=YES
```

```
connect_from_port_20=YES
```

```
ftpd_banner=Somente Pessoas Autorizadas
```

```
secure_chroot_dir=/var/run/vsftpd
```

```
pam_service_name=vsftpd
```

```
rsa_cert_file=/etc/ssl/certs/vsftpd.pem
```

Essas linhas especificam o seguinte:

- **linha 1** – Faz com que o servidor rode como standalone;
- **linha 2** – Ativa ftp anônimo;
- **linha 3** – Permite que usuário local do sistema faça logon via ftp;

- **linha 4** – Ativa a possibilidade de haver mensagens customizadas por diretório;
- **linha 5** – Ativa o log de downloads e uploads;
- **linha 6** – Faz com que o servidor atue de modo ativo;
- **linha 7** – Define o banner que será mostrado durante o “logon” via ftp;
- **linha 8** – Diretório chroot a partir do qual o servidor irá rodar;
- **linha 9** – É o nome pelo qual o vsftp será chamado no PAM;
- **linha 10** – Especifica o local onde ficará o certificado de SSL.

2. Reinicie o “vsftp” com o comando:

```
/etc/init.d/vsftpd restart
```

### *17.3 Teste de autenticação*

Em algumas distribuições Gnu/Linux pode surgir uma mensagem de erro, como a mostrada abaixo, no momento de logar:

```
500 OOPS: cap_set_proc
```

```
Login failed.
```

```
Remote system type is Login.
```

Caso esse erro aconteça basta carregar o módulo “capability” como seguinte comando:

```
modprobe capability
```

Se este for o seu caso, não esqueça de fazer com que esse módulo seja carregado automaticamente na inicialização do sistema.

Para testar a conexão basta estabelecer uma conexão ftp com o comando:

```
ftp ldap__.anahuac.org
```

**OBSERVAÇÃO:** o usuário somente será capaz de logar no servidor FTP se o diretório “home” dele existir. Você pode criar esse diretório manualmente ou instalar o programa “Autodir” para fazer isso automaticamente.



## 17.4 Instalando o “proftpd”

O “Proftpd” é um servidor de FTP mais robusto e com mais opções de configuração do que o “vsftpd”. Tanto que ele tem suas próprias configurações para suportar LDAP.

1. Para instalar o “proftpd” e sua documentação execute o comando abaixo:

```
aptitude install proftpd-ldap proftpd-doc
```

2. Na primeira tela selecione “standalone”

## 17.5 Configurando o “proftpd”

A configuração do “proftpd” deve ser feita em dois lugares: no PAM e no próprio arquivo de configuração do servidor.

1. Edite o arquivo “/etc/pam.d/proftpd” e descomente a linha 9:

```
##PAM-1.0
```

```
auth required pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed
```

```
@include common-auth
```

```
# This is disabled because anonymous logins will fail otherwise,
```

```
# unless you give the ‘ftp’ user a valid shell, or /bin/false and add
```

```
# /bin/false to /etc/shells.
```

```
auth required pam_shells.so
```

```
@include common-account
```

```
@include common-session
```

Esse arquivo de configuração do “proftpd” via PAM é bastante similar ao do vsftpd. A diferença é que nesse arquivo a linha 9 vem comentada. Essa linha está especificando que para um usuário conseguir se logar no ftp ele deve possuir uma shell válida e, que se ativarmos essa configuração o login anônimo será desativado.

Para ativá-lo, devemos incluir a shell `/bin/false` em `/etc/shells`, para que ela seja considerada como sendo uma shell válida.

Diferentemente do “vsftp”, o “proftpd” precisa de configurações específicas para operar com LDAP.

2. Edite o arquivo de configuração do “proftpd”: `/etc/proftpd/proftpd.conf`. Abaixo relacionamos algumas sugestões de configuração:

```
UseIPv6                off
ServerName              "FTP do curso de OpenLDAP"
ServerType              standalone
DeferWelcome           off
MultilineRFC2228       on
DefaultServer           on
ShowSymlinks           on
TimeoutNoTransfer       600
TimeoutStalled         600
TimeoutIdle            1200
DisplayLogin            welcome.msg
DisplayFirstChdir      .message
ListOptions             "-l"
DenyFilter             \*.*
DefaultRoot             ~
# Port 21 is the standard FTP port.
Port                   21
```

As linhas alteradas foram:

- **linha 1** – Desativa o uso de Ipv6;
- **linha 3** – Mudança no banner apresentado durante o login via ftp;
- **linha 21** – Define que o diretório padrão, quando um usuário se logar, é o seu diretório “home”.

3. Para que o “proftpd” possa utilizar a base LDAP é necessário adicionar as linhas abaixo no arquivo de configuração do

servidor, ou seja, no “/etc/proftpd/proftpd.conf”:

```
LDAPServer ldap__anahuac.org
LDAPDNInfo      cn=admin,dc=anahuac,dc=org senha
LDAPDoAuth      on “ou=Usuarios,dc=anahuac,dc=org”
LDAPUseTLS      on
```

Explicando

- **linha 1** – Define o “host” do servidor LDAP que será utilizado
- **linha 2** – Define o DN do usuário “admin” e sua senha
- **linha 3** – Define o “galho” onde a pesquisa será feita
- **linha 4** – Ativa o uso de TLS

4. Reinicie o “proftpd” com o seguinte comando:

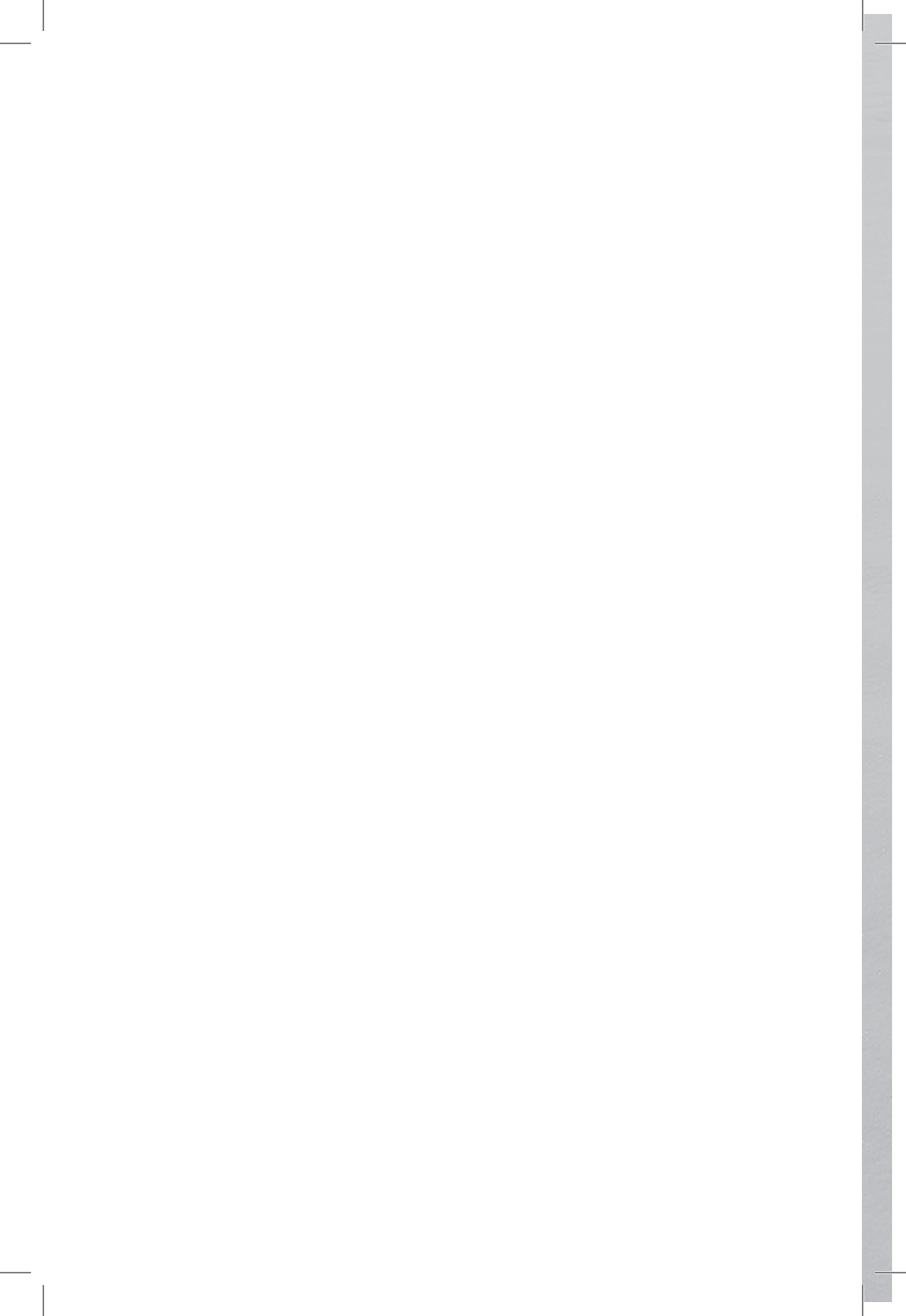
```
/etc/init.d/proftpd restart
```

## ***17.6 Teste de autenticação***

Para testar a conexão basta estabelecer uma conexão ftp com o comando:

```
ftp ldap__anahuac.org
```

**OBSERVAÇÃO:** o usuário somente será capaz de logar no servidor FTP se o diretório “home” dele existir. Você pode criar esse diretório manualmente ou instalar o programa “Autodir” para fazer isso automaticamente.





APÊNDICE

## Apêndice I – Schemas: Tipo de valores

A definição do tipo de valor que pode ser armazenado nesse atributo é organizado em três tipos:

- Equality

| Equality matching rules             |                            |
|-------------------------------------|----------------------------|
| Matching Rule                       | OID                        |
| caseExactIA5Match                   | 1.3.6.1.4.1.1466.109.114.1 |
| caseExactMatch                      | 2.5.13.5 IA5               |
| caseIgnoreIA5Match                  | 1.3.6.1.4.1.1466.109.114.2 |
| caseIgnoreMatch                     | 2.5.13.2                   |
| distinguishedNameMatch              | 2.5.13.1                   |
| generalizedTimeMatch                | 2.5.13.27                  |
| ibm-entryUuidMatch                  | 1.3.18.0.2.22.2            |
| integerFirstComponentMatch          | 2.5.13.29                  |
| integerMatch                        | 2.5.13.14                  |
| objectIdentifierFirstComponentMatch | 2.5.13.30                  |
| objectIdentifierMatch               | 2.5.13.0                   |
| octetStringMatch                    | 2.5.13.17                  |
| telephoneNumberMatch                | 2.5.13.20                  |
| uTCTimeMatch                        | 2.5.13.25                  |

- Ordering

| Ordering matching rules        |                  |
|--------------------------------|------------------|
| Matching rule                  | OID              |
| caseExactOrderingMatch         | 2.5.13.6         |
| caseIgnoreOrderingMatch        | 2.5.13.3         |
| distinguishedNameOrderingMatch | 1.3.18.0.2.4.405 |
| generalizedTimeOrderingMatch   | 2.5.13.28        |

- Substring

| Substring matching rules       |           |
|--------------------------------|-----------|
| Matching rule                  | OID       |
| caseExactSubstringsMatch       | 2.5.13.7  |
| caseIgnoreSubstringsMatch      | 2.5.13.4  |
| telephoneNumberSubstringsMatch | 2.5.13.21 |

---

**Syntax**

Directory String syntax

String syntax

IA5 String syntax

Directory String syntax

DN - distinguished name

Generalized Time syntax

Directory String syntax

Integer syntax - integral number

Integer syntax - integral number

String for containing OIDs. The OID is a string containing digits (0-9) and decimal points (.).

String for containing OIDs. The OID is a string containing digits (0-9) and decimal points (.).

Directory String syntax

Telephone Number syntax

UTC Time syntax

---

**Syntax**

Directory String syntax

Directory String syntax

DN - distinguished name

Generalized Time syntax

---

**Syntax**

Directory String syntax

Directory String syntax

Telephone Number syntax

## Apêndice II – qmailuser.schema

qmailuser.schema

```
#
# qmail-ldap (20030901) ldapv3 directory schema
#
# The official qmail-ldap OID assigned by IANA is 7914
#
# Created by: David E. Storey <dave[2297_files/at.gif]tamos.
net>
# Modified and included into qmail-ldap by Andre Oppermann
<opi[2297_files at.gif]nrg4u.com>
# Schema fixes by Mike Jackson <mjj[2297_files/at.gif]pp.fi>
#
#
# This schema depends on:
# - core.schema
# - cosine.schema
# - nis.schema
#
# Attribute Type Definitions
attributetype (1.3.6.1.4.1.7914.1.2.1.1 NAME 'qmailUID'
    DESC 'UID of the user on the mailsystem'
    EQUALITY integerMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.2.1.2 NAME 'qmailGID'
    DESC 'GID of the user on the mailsystem'
    EQUALITY integerMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.2.1.3 NAME 'mailMessageStore'
    DESC 'Path to the maildir/mbox on the mail system'
    EQUALITY caseExactIA5Match
    SUBSTR caseIgnoreIA5SubstringsMatch
```



```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.2.1.4 NAME
'mailAlternateAddress'
DESC 'Secondary (alias) mailaddresses for the same user'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})
#
# mailQuota format is no longer supported from qmail-ldap
20030901 on,
# user mailQuotaSize and mailQuotaCount instead.
#
#attributetype (1.3.6.1.4.1.7914.1.2.1.5 NAME 'mailQuota'
# DESC 'The amount of space the user can use until all further
messages get bounced.'
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.44 SINGLE-VALUE)
#
#attributetype (1.3.6.1.4.1.7914.1.2.1.6 NAME 'mailHost'
# DESC 'On which qmail server the messagestore of this user
is located.'
# EQUALITY caseIgnoreIA5Match
# SUBSTR caseIgnoreIA5SubstringsMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256}
SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.2.1.7 NAME
'mailForwardingAddress'
DESC 'Address(es) to forward all incoming messages to.'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})
attributetype (1.3.6.1.4.1.7914.1.2.1.8 NAME
'deliveryProgramPath'
DESC 'Program to execute for all incoming mails.'

```

EQUALITY caseExactIA5Match  
 SUBSTR caseIgnoreIA5SubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})  
 attributetype (1.3.6.1.4.1.7914.1.2.1.9 NAME 'qmailDotMode'  
     DESC 'Interpretation of .qmail files: both, dotonly, ldaponly,  
 ldapwithprog'  
     EQUALITY caseIgnoreIA5Match  
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32} SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.2.1.10 NAME 'deliveryMode'  
     DESC 'multi field entries of: nocal, noforward, noprogram,  
 reply'  
     EQUALITY caseIgnoreIA5Match  
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32})  
 attributetype (1.3.6.1.4.1.7914.1.2.1.11 NAME 'mailReplyText'  
     DESC 'A reply text for every incoming message'  
     EQUALITY caseIgnoreMatch  
     SUBSTR caseIgnoreSubstringsMatch  
         SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4096}  
 SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.2.1.12 NAME 'accountStatus'  
     DESC 'The status of a user account: active, noaccess, disa-  
 bled, deleted'  
     EQUALITY caseIgnoreIA5Match  
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.2.1.15 NAME 'mailQuotaSize'  
     DESC 'The size of space the user can have until further mes-  
 sages get bounced.'  
     EQUALITY caseIgnoreIA5Match  
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.2.1.14 NAME  
 'qmailAccountPurge'  
     DESC 'The earliest date when a mailMessageStore will  
 be purged'

```

    EQUALITY numericStringMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.36 SINGLE-VALUE)
attributetype(1.3.6.1.4.1.7914.1.2.1.16 NAME 'mailQuotaCount'
    DESC 'The number of messages the user can have until further
    messages get bounced.'
    EQUALITY integerMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.2.1.17 NAME 'mailSizeMax'
    DESC 'The maximum size of a single messages the user
    accepts.'
    EQUALITY integerMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)
#
# qmailGroup attributes
#
attributetype (1.3.6.1.4.1.7914.1.3.1.1 NAME 'dnmember'
    DESC 'Group member specified as distinguished name.'
    EQUALITY distinguishedNameMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)
attributetype (1.3.6.1.4.1.7914.1.3.1.2 NAME 'rfc822member'
    DESC 'Group member specified as normal rf822 email
    address.'
    EQUALITY caseIgnoreIA5Match
    SUBSTR caseIgnoreIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})
attributetype (1.3.6.1.4.1.7914.1.3.1.3 NAME 'filtermember'
    DESC 'Group member specified as ldap search filter.'
    EQUALITY caseIgnoreIA5Match
    SUBSTR caseIgnoreIA5SubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{512})
attributetype (1.3.6.1.4.1.7914.1.3.1.4 NAME 'senderconfirm'
    DESC 'Sender to Group has to answer confirmation email.'
    EQUALITY booleanMatch

```

SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.3.1.5 NAME 'membersonly'  
 DESC 'Sender to Group must be group member itself.'  
 EQUALITY booleanMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.3.1.6 NAME 'confirmtext'  
 DESC 'Text that will be sent with sender confirmation email.'  
 EQUALITY caseIgnoreMatch  
 SUBSTR caseIgnoreSubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4096}  
 SINGLE-VALUE)  
 attributetype (1.3.6.1.4.1.7914.1.3.1.7 NAME 'dnmoderator'  
 DESC 'Group moderator specified as Distinguished name.'  
 EQUALITY distinguishedNameMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)  
 attributetype (1.3.6.1.4.1.7914.1.3.1.8 NAME 'rfc822moderator'  
 DESC 'Group moderator specified as normal rfc822 email  
 address.'  
 EQUALITY caseIgnoreIA5Match  
 SUBSTR caseIgnoreIA5SubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})  
 attributetype (1.3.6.1.4.1.7914.1.3.1.9 NAME 'moderatortext'  
 DESC 'Text that will be sent with request for moderation  
 email.'  
 EQUALITY caseIgnoreMatch  
 SUBSTR caseIgnoreSubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4096}  
 SINGLE-VALUE)  
 #  
 # qldapAdmin Attributes  
 #  
 attributetype (1.3.6.1.4.1.7914.1.4.1.1 NAME 'qladnmanager'  
 DESC ''

EQUALITY distinguishedNameMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.12)

attributetype (1.3.6.1.4.1.7914.1.4.1.2 NAME 'qlaDomainList'  
 DESC "  
 EQUALITY caseIgnoreIA5Match  
 SUBSTR caseIgnoreIA5SubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})

attributetype (1.3.6.1.4.1.7914.1.4.1.3 NAME 'qlaUidPrefix'  
 DESC "  
 EQUALITY caseIgnoreIA5Match  
 SUBSTR caseIgnoreIA5SubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} SINGLE-VALUE)

attributetype (1.3.6.1.4.1.7914.1.4.1.4 NAME 'qlaQmailUid'  
 DESC "  
 EQUALITY integerMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

attributetype (1.3.6.1.4.1.7914.1.4.1.5 NAME 'qlaQmailGid'  
 DESC "  
 EQUALITY integerMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

attributetype (1.3.6.1.4.1.7914.1.4.1.6 NAME  
 'qlaMailMStorePrefix'  
 DESC "  
 EQUALITY caseIgnoreIA5Match  
 SUBSTR caseIgnoreIA5SubstringsMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256} SINGLE-VALUE)

attributetype (1.3.6.1.4.1.7914.1.4.1.7 NAME 'qlaMailQuotaSize'  
 DESC "  
 EQUALITY integerMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

attributetype (1.3.6.1.4.1.7914.1.4.1.8 NAME  
 'qlaMailQuotaCount'  
 DESC "  
 EQUALITY integerMatch  
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

```

EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.4.1.9 NAME 'qlaMailSizeMax'
DESC "
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)
attributetype (1.3.6.1.4.1.7914.1.4.1.10 NAME 'qlaMailHostList'
DESC "
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreIA5SubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256})
# Object Class Definitions
objectclass (1.3.6.1.4.1.7914.1.2.2.1 NAME 'qmailUser'
DESC 'QMail-LDAP User'
SUP top
AUXILIARY
MUST (mail)
MAY (uid $ mailMessageStore $ homeDirectory $ userPas-
sword $
mailAlternateAddress $ qmailUID $ qmailGID $
mailForwardingAddress $ deliveryProgramPath $
qmailDotMode $ deliveryMode $ mailReplyText $
accountStatus $ qmailAccountPurge $
mailQuotaSize $ mailQuotaCount $ mailSizeMax))
objectclass (1.3.6.1.4.1.7914.1.3.2.1 NAME 'qmailGroup'
DESC 'QMail-LDAP Group'
SUP top
AUXILIARY
MUST (mail $ mailAlternateAddress $ mailMessageStore)
MAY (dnmember $ rfc822member $ filtermember $ sen-
derconfirm $
membersonly $ confirmtext $ dnmoderator $
rfc822moderator $ moderatortext))

```

```
objectclass (1.3.6.1.4.1.7914.1.4.2.1 NAME 'qldapAdmin'  
  DESC 'QMail-LDAP Subtree Admin'  
  SUP top  
  AUXILIARY  
  MUST (qldapDnManager $ qldapDomainList $ qldapMailMStore-  
Prefix $  
  qldapMailHostList)  
  MAY (qldapUidPrefix $ qldapQmailUid $ qldapQmailGid $  
qldapMailQuotaSize $ qldapMailQuotaCount $ qldapMailSizeMax))
```

buqui

[www.buqui.com.br](http://www.buqui.com.br)  
[www.editorabuqui.com.br](http://www.editorabuqui.com.br)  
[www.autopubli.com.br](http://www.autopubli.com.br)