Autenticação Integrada Baseada em Serviço de Diretório LDAP

Erich Soares Machado, Universidade de São Paulo <erichmachado@gmail.com>
Flavio da Silva Mori Junior, Universidade de São Paulo <flaviomori2001@gmail.com>

Autenticação Integrada Baseada em Serviço de Diretório LDAP

por Erich Soares Machado e Flavio da Silva Mori Junior

Resumo

Esse texto foi escrito como parte do projeto de formatura dos autores, desenvolvido para cumprir as atividades exigidas durante o curso de graduação em Bacharelado em Ciência da Computação do IME-USP.

O objetivo é oferecer uma apresentação ao serviço de diretório LDAP, através de uma abordagem teórica e técnica. Os conceitos teóricos são apresentados no primeiro capítulo, que serve como introdução para quem não estiver familiarizado com o assunto.

A parte técnica é apresentada na forma de um manual de instalação e configuração do serviço de diretório em um ambiente de rede, e de sua integração com sistemas de autenticação. Nessa parte também são apresentados conceitos importantes, porém de maneira aplicada.

Além disso, apresentamos informações que consideramos de fundamental importância para o tópico, mas que não se encaixam na estrutura do texto principal, na forma de apêndices.

Ao final do texto, existe uma seção que não está listada no índice, com conteúdo subjetivo a respeito do processo de elaboração do projeto, em conformidade com os requisitos da disciplina MAC0499 (Trabalho de Formatura Supervisionado). Essa seção chama-se Parte Subjetiva.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Exemplos

Capítulo 1. Introdução

Resumo

Este capítulo descreve o que é um serviço de diretório, o protocolo LDAP, e um cenário (redes heterogêneas) em que é interessante usar um serviço LDAP para organizar e gerenciar as informações. Também descreve formas de implantar esse serviço.

Introdução ao LDAP

Redes de computadores estão presentes na maioria das empresas atualmente, devido a grande necessidade de comunicação que as aplicações distribuídas exigem. Muitas dessas aplicações utilizam os mesmos dados para realizar as suas operações. Sendo assim, torna-se necessário buscar uma maneira de organizar essa informação de maneira clara e consistente, de forma a facilitar o acesso às mesmas, reduzir o custo de sua manutenção e por conseqüência aumentar a funcionalidade dos vários sistemas que a usam.

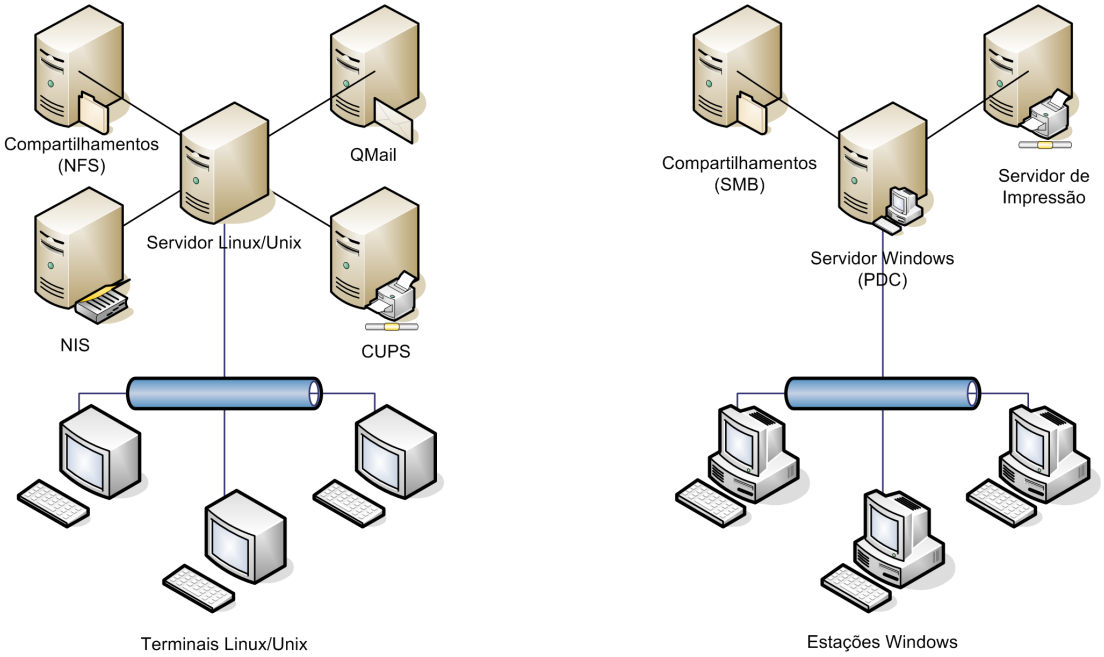
A necessidade de integração desse tipo de informação motivou o surgimento de um padrão aberto que possa atendê-la. Esse padrão chama-se LDAP (*Lightweight Directory Access Protocol*), e trata-se de um protocolo que define um método para o acesso e a atualização de informações em um diretório. Diretório é uma espécie de banco de dados, otimizado para leitura e busca.

O LDAP define um protocolo de comunicação, ou seja, o transporte e o formato das mensagens utilizadas pelo cliente para acessar os dados que estão armazenados em um diretório do tipo X.500. O padrão X.500 organiza as entradas do diretório em um espaço de nomes hierárquico (uma árvore) capaz de incorporar grandes volumes de informação. O LDAP também define métodos de busca poderosos o suficiente para tornar a recuperação dessa informação fácil e eficiente. Ele não define o serviço de diretório em si. Com o LDAP, o cliente não é dependente da implementação em particular do serviço de diretório que está no servidor.

Redes heterogêneas

Todo ambiente de rede precisa armazenar informações para possibilitar o seu gerenciamento (autenticação, grupos de usuários, permissões, cotas de armazenamento e impressão, compartilhamentos e etc.). Hoje em dia, a maioria das grandes organizações possui ambientes de rede heterogêneos, com várias plataformas presentes (Linux, Windows, Solaris…) e com redes virtuais fisicamente conectadas, muitas vezes distribuídas geograficamente. Um exemplo de organização desse tipo é a Universidade de São Paulo, que possui uma grande rede de dados interconectando todos os seus *campi*, espalhados pelo estado.

Figura 1.1. Exemplo de serviços em uma rede heterogênea



Esse exemplo ilustra uma rede heterogênea composta de um ambiente Windows e um ambiente Linux. Apesar de ambos os ambientes estarem fisicamente conectados (utilizando a mesma infra-estrutura), não existe comunicação entre os serviços e a informação usada para administrar os recursos não está sendo compartilhada.

Um problema decorrente desse tipo de implantação é que para cada plataforma ou para cada rede local virtual existente no ambiente de rede (rede física), é necessário suprir essas mesmas informações de gerenciamento. Se não for adotada uma boa solução de gerenciamento, podem surgir problemas decorrentes da replicação desses dados. Os principais são: redundância, falta de sincronia nas informações, dificuldade de organização, maior custo no suporte e falta de segurança.

Serviços de diretório

Um diretório é um repositório de informações sobre objetos, organizados segundo um critério que facilite a sua consulta. Dois exemplos práticos de diretórios que usamos no nosso cotidiano são: uma lista telefônica e um dicionário. Ambos armazenam informações para consulta, ordenadas para facilitar a busca por uma entrada: a lista telefônica organiza as entradas em ordem alfabética pelo nome da pessoa e o dicionário organiza as entradas também em ordem alfabética, por verbete.

O modo como o diretório é usado pode ser descrito como *white pages* ou *yellow pages*. Se o nome do objeto é conhecido, é possível recuperar suas características. Isso é semelhante a procurar um nome de uma pessoa em uma lista telefônica residencial (*white pages*). Se o nome de um objeto em particular não é conhecido, é possível fazer uma busca no diretório em busca de objetos que cumpram certos requerimentos. Isso é semelhante a procurar um dentista em uma lista telefônica comercial (*yellow pages*).

Serviço de diretório é um serviço de armazenamento de informações otimizado para busca e leitura. Eles tendem a conter informações descritivas baseadas em atributos, para assim suportarem algum mecanismo de filtragem. Por exemplo, buscar em um serviço de diretório da cidade de São Paulo, mulheres entre 18 e 25 anos e que sejam solteiras. Além disso, eles têm que ser extensíveis, para servirem a uma gama maior de aplicações e propósitos. Um serviço de diretório pode armazenar os telefones de bares e casas noturnas

de São Paulo. Se ele aceitar também endereços como atributos, o serviço pode ser estendido, acoplando-o a um guia de ruas.

Serviços de diretório e bancos de dados compartilham várias características importantes, como buscas rápidas e um esquema extensível. A diferença é que um serviço de diretório é projetado mais para leitura do que para escrita, enquanto que em um banco de dados assumimos que as operações de leitura e de escrita ocorrem mais ou menos com a mesma freqüência. Portanto, para os serviços de diretório não são essenciais certas características que são a bancos de dados, que permitem lidar com um grande volume de atualizações complexas. Dentre elas estão suporte a transações[1] (*transactions*) e travas de escrita (*write locks*).

A maioria dos bancos de dados suportam um método de acesso padrão e muito poderoso chamado SQL (*Structured Query Language*). Os serviços de diretório usam um protocolo de acesso simplificado. Já que eles não fornecem todas as funções que um banco de dados, eles podem ser otimizados para fornecer economicamente dados para leitura a um maior número de aplicações, em um ambiente distribuído. As atualizações dos diretórios são tipicamente simples.

Os serviços de diretórios são ajustados para dar resposta rápida a operações de busca em grande volume. Assim sendo, eles podem ter a habilidade de replicar informação com o objetivo de aumentar a disponibilidade e a confiabilidade, além de reduzir o tempo de resposta. Quando a informação do diretório é replicada, é aceitável que aconteçam inconsistências temporárias entre as réplicas, desde que elas se sincronizem eventualmente.

Alguns serviços de diretório são locais, fornecendo serviço a um contexto restrito como, por exemplo, o programa finger em uma máquina Linux/Unix. Outros serviços são globais, fornecendo serviço para um contexto mais abrangente, como por exemplo, o DNS (*Domain Name System*) da Internet. Os serviços globais são geralmente distribuídos, os dados que eles contêm estão espalhados em várias máquinas, cada uma cooperando para fornecer o serviço de diretório final. Tipicamente um serviço global define um espaço de nomes que dá a mesma visão dos dados, não importando onde você está. Por exemplo, uma pesquisa no DNS deve retornar o mesmo resultado, independente de em qual computador foi realizada a busca.

Protocolo LDAP

LDAP (*Lightweight Directory Access Protocol* ou Protocolo Leve de Acesso a Diretório) é um protocolo leve para acessar serviços de diretório baseados nos padrões X.500, que funciona sobre TCP/IP. O conjunto original das principais definições do LDAP (Versão 3) está nos RFC's 2251-2256. As especificações técnicas estão no [RFC3377 *"Lightweight Directory Access Protocol (v3): Technical Specification"* [ftp://ftp.rfc-editor.org/in-notes/rfc3377.txt]]. Além desses, existem outros RFC's que definem outras características do LDAP.

LDAP é baseado no modelo cliente/servidor e a comunicação é assíncrona. Ou seja, um cliente pode fazer múltiplas requisições e as respostas dadas pelo servidor podem chegar em qualquer ordem.

Modelo de informação do LDAP

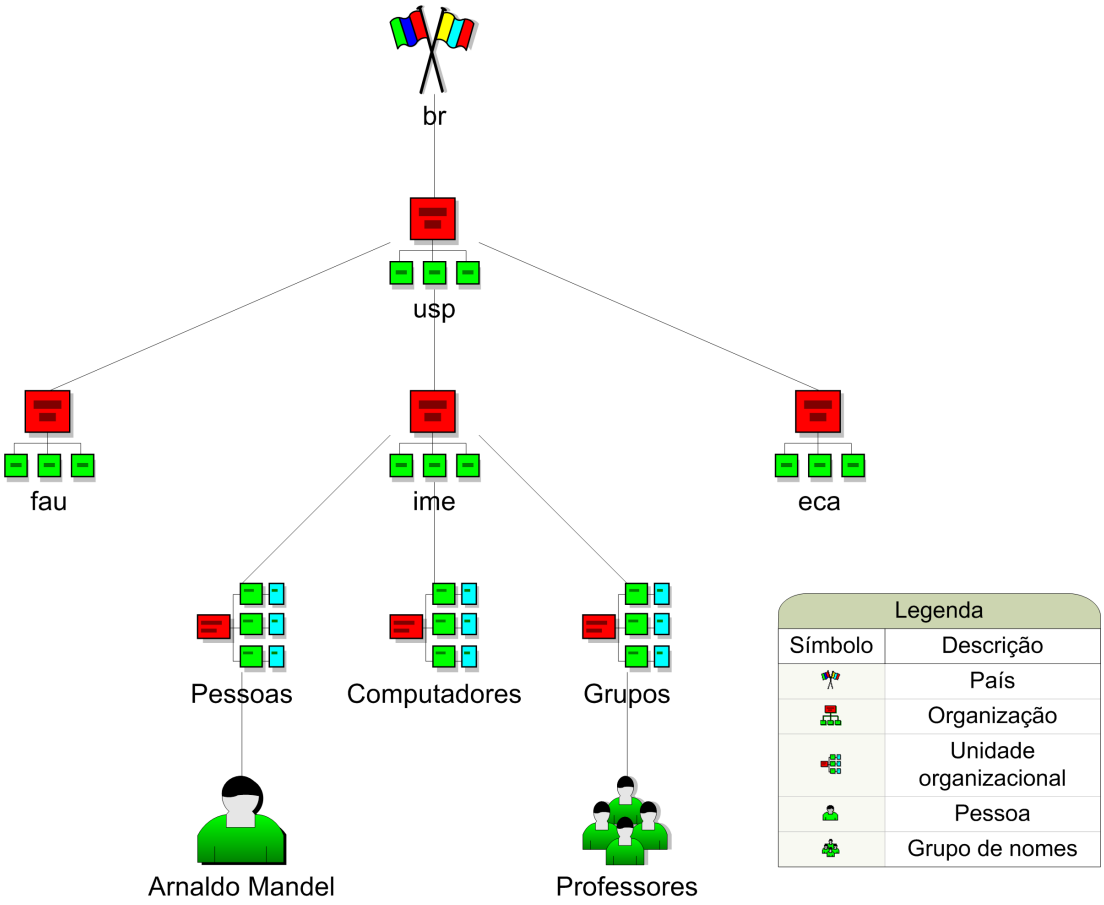
O modelo de informação do LDAP é baseado em entradas. Uma entrada é uma coleção de atributos que tem um Nome Distinto (*Distinguished Name* ou DN) globalmente único. O DN é usado para se referir à entrada sem ambigüidade. Cada atributo da entrada tem um tipo e um ou mais valores. Os tipos são normalmente *strings* mnemônicas, como "cn" para *commom name*, ou "*mail*" para endereço de *e-mail*. A sintaxe dos valores depende do tipo do atributo. Por exemplo, um atributo `cn` pode conter o valor "Arnaldo Mandel". Um atributo `mail` pode conter o valor "am@ime.usp.br". Um atributo `jpegPhoto` poderia conter uma fotografia no formato JPEG (binário).

[1] Transações são operações *all-or-nothing*, ou seja, que só devem ser realizadas totalmente, não podendo ser concluídas parcialmente.

No LDAP, as entradas do serviço de diretório são organizadas em uma estrutura de árvore hierárquica. Essa árvore é conhecida como DIT (*Directory Information Tree*). Tradicionalmente essa estrutura refletia os limites geográficos ou organizacionais. Entradas representando países aparecem no topo da árvore. Abaixo delas estão entradas representando estados (Unidades Federativas). Abaixo delas podem estar entradas representando unidades organizacionais, pessoas, impressoras, documentos ou qualquer outra coisa.

A árvore também pode ser organizada conforme os nomes de domínios da Internet. Essa forma de nomenclatura está se tornando cada vez mais popular, já que permite ao serviço de diretório ser localizado usando o DNS.

Figura 1.2. Exemplo de DIT (*Directory Information Tree*)



Além disso, é possível controlar quais atributos são requeridos e permitidos em uma entrada, através do uso de um atributo especial chamado `objectClass` (classe do objeto). Os valores do atributo `objectClass` determinam o que ela representa e quais regras a entrada deverá obedecer.

Uma entrada é referenciada pelo seu Nome Distinto (DN), o qual é construído pegando o nome da entrada, chamado RDN (*Relative Distinguished Name* ou Nome Distinto Relativo), e concatenando os nomes de suas entradas antecessoras. Por exemplo, a entrada "Arnaldo Mandel" no exemplo de nomenclatura da Internet acima, tem um RDN de `uid`=am e um DN de `uid`=am,`ou`=Pessoas,`dc`=ime,`dc`=usp,`dc`=br. O formato completo do DN é descrito no [RFC2253 *"Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names"* [ftp://ftp.rfc-editor.org/in-notes/rfc2253.txt]].

LDAP define operações para interrogar e atualizar o serviço de diretório. Operações são fornecidas para adicionar e apagar uma entrada do diretório, modificar uma entrada existente, e modificar o nome da entrada. A operação de busca do LDAP permite a certas partes do diretório serem pesquisadas em busca de entradas que obedeçam certos critérios especificados por um filtro de busca.

Por exemplo, você pode querer procurar na sub-árvore cuja raiz é `dc`=ime,`dc`=usp,`dc`=br por pessoas cujos nomes sejam "Arnaldo Mandel", recuperando o endereço de *e-mail* de cada entrada achada. O LDAP permite que você faça isso facilmente.

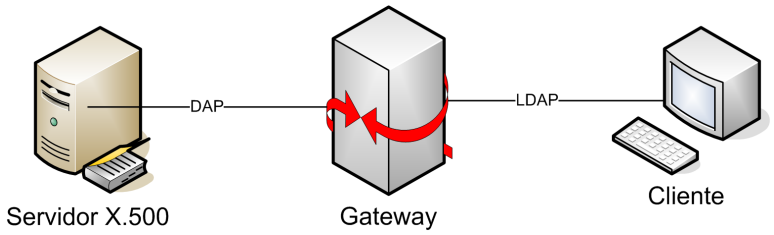
É fornecido um mecanismo para o cliente autenticar-se, ou comprovar sua identidade para um serviço de diretório. O LDAP também suporta serviços de segurança de dados (integridade e confidencialidade).

Origem do LDAP

LDAP é um protocolo de acesso a diretórios do tipo X.500, o serviço de diretório OSI (*Open Systems Interface*). Inicialmente, os clientes LDAP acessavam *gateways* para o serviço de diretório X.500. Esse *gateway* (também chamado de *proxy* ou *front-end*) rodava LDAP entre o cliente e o *gateway*; e rodava o DAP (*Directory Access Protocol* ou Protocolo de Acesso a Diretório) X.500 entre o *gateway* e o servidor X.500.

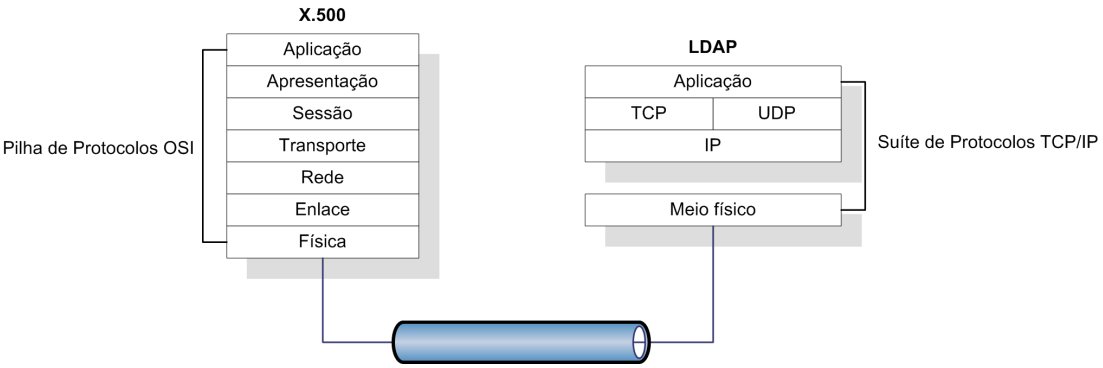
O X.500 é um protocolo pesado, que opera sobre a pilha completa de protocolos OSI e requer uma quantidade significante de recursos computacionais. O LDAP é projetado para operar sobre TCP/IP e fornece a maioria das funcionalidades do X.500 com um custo muito menor.

Figura 1.3. *Modelo gateway LDAP/DAP*



O LDAP é considerado leve, pois não precisa rodar na pilha de sete camadas OSI, como o protocolo da camada de aplicação X.500. Os pacotes X.500 carregam mais bagagem, pois precisam de cabeçalhos para cada uma das camadas da pilha de protocolos OSI. A suite de protocolos TCP/IP, na qual o LDAP roda, também necessita de cabeçalhos nos pacotes, mas tem um *overhead* menor.

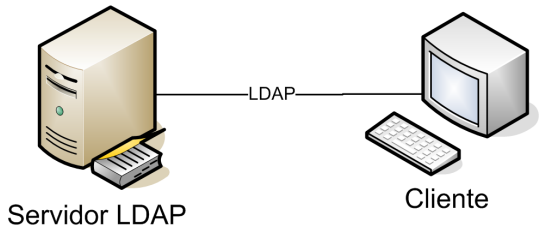
Figura 1.4. X.500 sobre OSI vs. LDAP sobre TCP/IP



O segundo motivo é que o LDAP omite várias operações do X.500 que são raramente usadas. LDAPv3 possui apenas nove operações principais e fornece um modelo mais simples para os programadores e administradores. Assim é possível que eles se foquem mais na semântica de seus programas, sem terem que se preocupar com características do protocolo raramente usadas.

Além do LDAP ainda ser usado para acessar o serviço de diretório X.500 através de *gateways*, LDAP é também agora implementado direto em servidores LDAP do tipo X.500. Note o uso de "do tipo X.500" em vez de simplesmente "X.500", pois um servidor X.500 não entende mensagens LDAP. O segundo uso é o mais comum atualmente, pois atende a praticamente todas necessidades.

Figura 1.5. *Modelo cliente/servidor*



Diretórios no contexto do LDAP

Resumo

Nesta seção vamos esclarecer alguns detalhes conceituais que às vezes não ficam muito claros em documentações do LDAP.

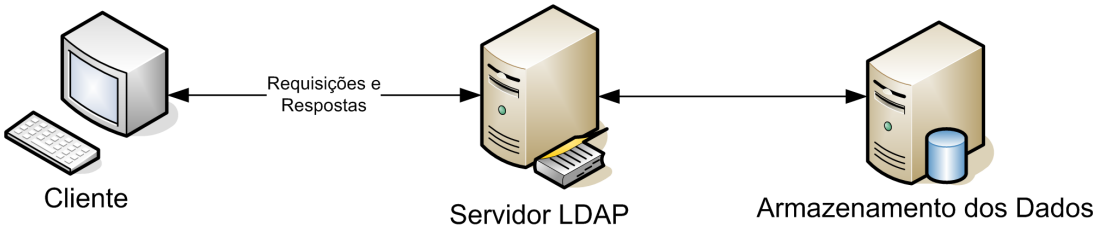
Não devemos esquecer que o *LDAP é apenas um protocolo de acesso*, ele define um conjunto de mensagens para acessar certos tipos de dados. O protocolo em si não diz nada a respeito de onde ou como os dados são armazenados.

Um *servidor LDAP é uma aplicação que implementa o lado servidor* conforme as especificações do protocolo. Existem várias implementações e a que descreveremos adiante é o OpenLDAP, uma solução livre. É esse *servidor que fornece o serviço de diretório* aos seus clientes. Um servidor LDAP está para o diretório assim como o SGBD (sistema gerenciador de banco de dados) está para o banco de dados.

O *diretório é representado através de um backend*, que é uma implementação de base de dados[2]. O servidor pode usar qualquer *backend* para armazenar as informações, desde arquivos de texto até bancos de dados relacionais. Dizemos que o LDAP (e conseqüentemente o servidor LDAP) não suporta algumas características dos bancos de dados, porque que o protocolo não possui as mensagens para usar essas características e, sendo assim, não requer que o *backend* que armazena os dados as tenha.

[2] A princípio, base de dados e banco de dados são a mesma coisa. Porém, normalmente quando usamos o termo banco de dados, estamos nos referindo a um banco de dados relacional. Portanto adotamos nesta monografia a convenção de usarmos base de dados quando nos referimos a qualquer tipo de banco de dados, o que inclui até mesmo um arquivo de texto, e usamos banco de dados para nos referirmos a um banco de dados relacional.

Figura 1.6. ***Relacionamento entre o cliente LDAP, servidor LDAP e backend***



O cliente não terá (ou pelo menos não deveria ter) informações a respeito do mecanismo de armazenamento que está sendo usado. Assim sendo, clientes e servidores LDAP podem se comunicar, independentemente de quais empresas os produziram.

Já se pensou em usar um servidor LDAP como *backend* para um servidor *Web*. Todos os HTML's e os arquivos gráficos ficariam armazenados no diretório para serem lidos por vários servidores *Web*. Afinal, um servidor *Web* normalmente somente lê arquivos e os manda para seus clientes; e esses arquivos não mudam com muita freqüência. Apesar de ser possível implementar um servidor *Web* que use LDAP para acessar seu *backend*, já existe um tipo especial de diretório para servir arquivos, chamado sistema de arquivo.

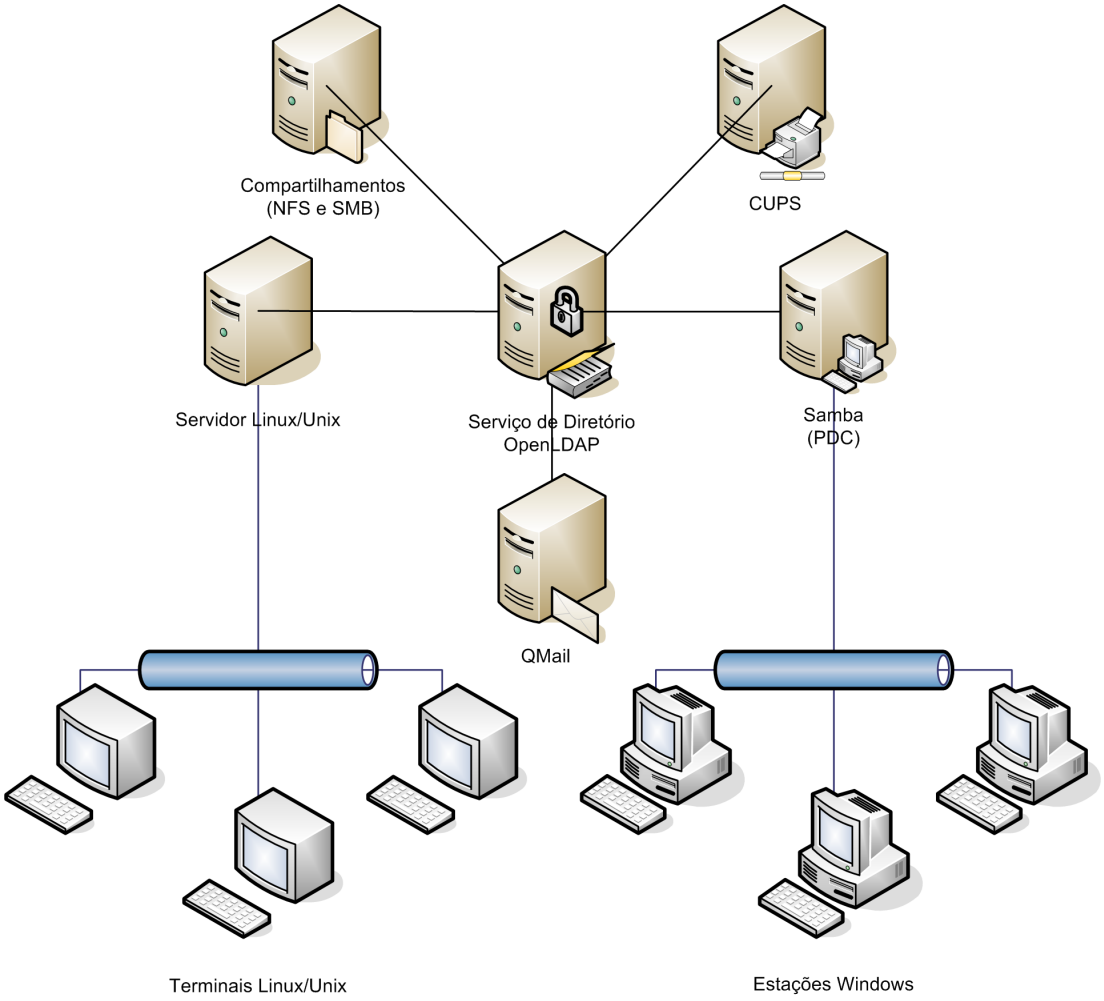
O LDAP foi criado para atender uma certa gama de problemas, não sendo destinado a substituir diretórios especializados, como sistemas de arquivo e o DNS.

Implantação

Uma solução cada vez mais empregada para este cenário é armazenar as informações do ambiente de rede em um diretório, através de um serviço de diretório LDAP. Isso torna possível acessar de forma padronizada, ágil e segura, todas essas informações. Portanto todos os serviços da rede (autenticação, compartilhamento, impressão, *e-mail*, etc.) buscarão as informações de que precisam nesse diretório, de forma integrada.

A seguir está o exemplo da mesma rede heterogênea mostrada anteriormente, utilizando um servidor LDAP para integração dos serviços.

Figura 1.7. Exemplo de serviços em uma rede heterogênea com integração LDAP



Esse exemplo ilustra uma rede heterogênea utilizando um servidor LDAP para integração dos serviços.

Uma maneira de disponibilizar um serviço de diretório LDAP é utilizando soluções livres disponíveis atualmente. Um exemplo de implantação desse tipo é instalar um servidor OpenLDAP, integrando-o ao PAM (*Pluggable Authentication Modules*) para realizar a autenticação dos clientes Linux/Unix, e integrando-o ao Samba para autenticar os clientes Windows. Toda a comunicação entre os serviços pode ser protegida através do suporte TLS (*Transport Layer Security*).

OpenLDAP

OpenLDAP é uma suíte de aplicativos LDAP *open-source*, que inclui todas as ferramentas necessárias para fornecer um serviço de diretório LDAP em um ambiente de rede (clientes, servidores, utilitários e ferramentas de desenvolvimento), disponível para várias plataformas (Linux, Solaris, MacOS). É uma solução considerada madura hoje em dia e possui amplo suporte, sendo largamente utilizada como alternativa às implementações comerciais existentes (Microsoft Active Directory, Novell eDirectory, Sun Java System Directory Server, etc.).

Ele implementa a versão 3 do LDAP, a versão mais recente do protocolo e que é o padrão atualmente, e suporta LDAP em IPv4, IPv6 e Unix IPC. O projeto OpenLDAP é uma continuação do servidor LDAP da Universidade de Michigan.

O OpenLDAP possui suporte a *threads* para aumentar a performance de seu servidor, reduzindo o *overhead* requerido para atender as múltiplas requisições que chegam dos clientes.

O *daemon* que implementa o servidor LDAP é o slapd. Além desse *daemon*, existe um outro, o slurpd, que é usado quando se deseja fornecer um serviço replicado de diretório. Ele é explicado em mais detalhes na próxima seção.

O OpenLDAP fornece várias opções para segurança, como suporte a TLS, SSL e SASL. Além disso, o acesso às informações pode ser restrito baseado na topologia da rede, endereços IP, nome de domínio e outros critérios.

O slapd pode ser configurado para servir a múltiplos bancos de dados ao mesmo tempo, ou seja, um único servidor slapd pode responder a requisições de várias porções diferentes da árvore LDAP, usando o mesmo ou vários *backends* de base de dados.

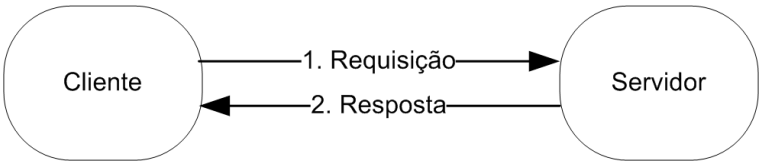
O OpenLDAP vem com várias opções de *backends* de armazenamento. Eles incluem BDB, um *backend* transacional de alta performance; HDB, um *backend* transacional hierárquico de alta performance; LDBM, um *backend* leve baseado no DBM; SHELL, uma interface de *backend* para *scripts shell* arbitrários; e PASSWD, uma interface simples de *backend* para o arquivo `/etc/passwd`. Os *backends* BDB e HDB utilizam o BD Sleepycat Berkeley. O LDBM utiliza o Berkeley ou o GDBM.

Modelos de serviços LDAP

Como já foi dito anteriormente, o serviço de diretório LDAP é baseado no modelo cliente/servidor. Um ou mais servidores LDAP contêm os dados, compondo a árvore de informação do diretório (DIT). O cliente se conecta a um servidor e faz requisições. O servidor responde com a informação requisitada ou com um apontador para um outro servidor LDAP. Não importa a qual servidor o cliente se conecte, ele tem a mesma visão dos dados.

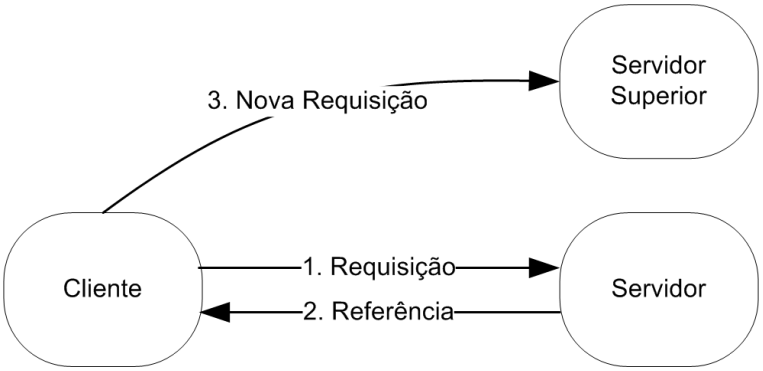
O serviço pode ser local, assim um servidor LDAP rodando em uma máquina fornece serviço de diretório apenas para o domínio local.

Figura 1.8. Modelo cliente/servidor simples



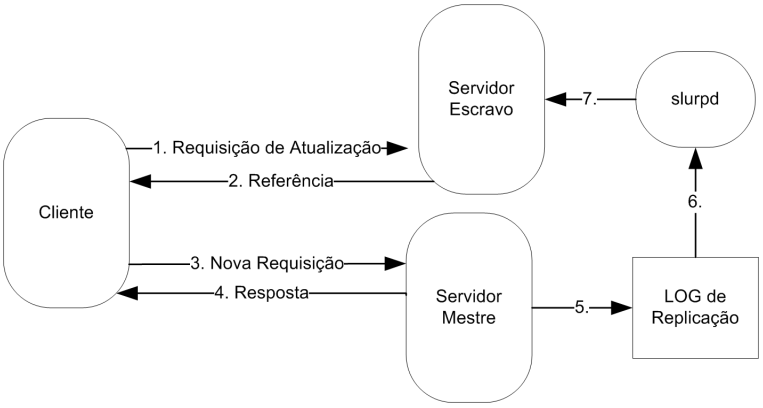
Ele também pode ser local com referências, assim ele fornece o serviço de diretório para o domínio local e retorna referências para um outro servidor capaz de lidar com requisições para fora do domínio. Essa configuração é usada caso se deseje que o serviço participe de um "diretório global".

Figura 1.9. Modelo cliente/servidor com referência



O serviço pode ser replicado. Nessa configuração, o slurpd é usado para sincronizar as alterações realizadas na base de dados do slapd *master* para as outras réplicas do slapd. O slapd e o slurpd comunicam-se através de um arquivo de texto que é usado como um *log* de alterações. Essa configuração pode ser usada em conjunto com qualquer uma das duas primeiras configurações, em situações em que um único slapd não fornece a disponibilidade ou a confiabilidade requerida. Existe um outro método de replicação, o LDAP *Sync*, o qual não entraremos em detalhes.

Figura 1.10. Modelo cliente/servidor com replicação



Uma outra configuração possível, é o serviço de diretório local ser particionado em vários serviços menores, onde cada um armazena as informações de uma sub-árvore. Então podemos juntá-los através de referências, para formar o serviço de diretótio final.

Capítulo 2. Configurando um serviço de diretório LDAP

Resumo

Esse capítulo descreve o processo de instalação e configuração dos pacotes necessários para implantar um serviço de diretório LDAP no seu ambiente de rede. O serviço de diretório escolhido foi o OpenLDAP, por ser livre, maduro e ter amplo suporte. Todo o procedimento de instalação e configuração descrito foi realizado utilizando a distribuição Linux Ubuntu 6.06 LTS (Dapper Drake), e pode facilmente ser aplicado a outras distribuicões fazendo-se as devidas adaptações.

Instalação

Execute os seguintes comandos para instalar o pacote necessário no servidor:

1. Atualize as listas dos repositórios

```
usuario@ldapserver:~$ sudo aptitude update
```

2. Instale o pacote slapd

```
usuario@ldapserver:~$ sudo aptitude install ldap-server
```

Este pacote não é um pacote real, ele aponta para o pacote slapd. Durante a configuração ele vai pedir uma senha para o administrador do LDAP, você pode deixá-la em branco se quiser. Vamos explicar como criar um novo arquivo de configuração a seguir.

Dica

Sempre mantenha uma cópia dos arquivos de configuração gerados automaticamente durante a instalação dos pacotes, eles podem servir como referência caso algum parâmetro de configuração sofra alterações em versões mais atualizadas.

Configuração

O arquivo de configuração[1] do serviço de diretório da suíte OpenLDAP (slapd) é o `/etc/ldap/slapd.conf`. Esse arquivo possui vários parâmetros que configuram desde a execução do serviço slapd até o *backend* de banco de dados que será utilizado, assim como os índices que devem ser gerados para agilizar as buscas e também a senha de administração para acessar o diretório. Ou seja, esse arquivo é a peça chave da implantação e sua configuração deve ser feita com bastante rigor. Devemos também nos certificar de que o acesso a esse arquivo estará restrito.

[1] Atualmente existe uma maneira alternativa de armazenar as configurações do serviço de diretório slapd, que é dentro de um diretório LDAP, na forma de um DIT. Ou seja, usando a própria estrutura que o serviço disponibiliza. A configuração nesse caso é carregada através de um arquivo LDIF e a vantagem é a de que existe a possibilidade de alterar os parâmetros em tempo de execução, isto é, sem a necessidade de reiniciar o servidor, através de comandos de acesso ao diretório. Porém, atualmente existem problemas de compatibilidade com alguns *backends* e com o sistema de replicação. Sendo assim preferimos adotar o modo "antigo".

As seguintes regras, comuns aos arquivos de configuração de sistemas Unix, são válidas para o `/etc/ldap/slapd.conf`:

- Linhas em branco e linhas começando com # são ignoradas.
- Parâmetros e valores associados são separados por caracteres em branco (espaço ou tabulação).
- Linhas com espaços em branco na primeira coluna serão consideradas como continuação da linha anterior.

Exemplo 2.1. Arquivo de configuração `/etc/ldap/slapd.conf`

```
######################## /etc/ldap/slapd.conf ######################
# Arquivo de configuração do serviço slapd.

# schema's
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema

schemacheck     on

# serviço
pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args

loglevel        296

modulepath      /usr/lib/ldap
moduleload      back_bdb

# segurança
allow           bind_v2

# bases de dados
backend         bdb
checkpoint 512 30

## base de dados no. 1
database        bdb
suffix          "dc=ime,dc=usp,dc=br"
directory       "/var/lib/ldap"
index           objectClass     eq
rootdn          "cn=admin,dc=ime,dc=usp,dc=br"
rootpw          {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==
lastmod         on
mode            0600
cachesize       2000

## ACL's para a base de dados no. 1
access to attrs=userPassword
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by anonymous auth
       by self write
       by * none
access to *
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by * read
###########################################################################
```

Atenção

O arquivo `/etc/ldap/slapd.conf` contém informações sigilosas. Não se esqueça de ajustar as permissões deste arquivo para 600 e certifique-se de que o seu proprietário seja o usuário *root*.

Os parâmetros de configuração do arquivo `/etc/ldap/slapd.conf` podem ser organizados em seções que ajustam cada aspecto do serviço. Se organizados em seções, podemos dividir os parâmetros como abaixo:

- *schema*'s
- serviço
- segurança
- bases de dados

Essa estrutura será adotada neste texto de forma a facilitar a explicação de cada parâmetro.

Importante

A ordem de alguns parâmetros no arquivo de configuração é significativa. Se resolver adotar outro tipo de organização, consulte a *manpage* do arquivo `/etc/ldap/slapd.conf` antes de mudar um parâmetro de lugar.

schema's

Os arquivos de *schema* definem que tipo de informação poderá ser armazenada no diretório, de acordo com as necessidades de cada aplicação que irá acessá-lo. Em outras palavras, eles definem quais atributos estarão disponíveis para cada entrada adicionada à árvore do diretório.

O pacote slapd contém alguns arquivos de *schema* por padrão. O principal deles é o `core.schema`, que provê a funcionalidade básica do serviço. A maioria dos *schema*'s inclusos fornecem a estrutura necessária para o armazenamento de informações de autenticação e suporte a aplicações distribuídas. O seguinte exemplo exibe a seção de *schema*'s que utilizaremos por enquanto:

```
...
# schema's
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema

schemacheck     on
...
```

Cada *schema* a ser utilizado pelo servidor deve ser declarado através do parâmetro *`include`*. Arquivos de *schema* possuem dependências entre si, ou seja, para utilizar um determinado *schema* em seu diretório deve-se declarar também todos os arquivos dos *schema*'s dos quais ele depende. Os *schema*'s normalmente encontram-se armazenados dentro do subdiretório `schema`, que reside no mesmo diretório onde as configurações do serviço estão armazenadas. Na distribuição Ubuntu, os *schema*'s estão localizados em `/etc/ldap/schema`.

Os *schema*'s adicionados no exemplo acima são os necessários para prover um serviço básico de autenticação em ambientes Linux/Unix. Mais arquivos de *schema*'s podem ser incluídos para aumentar a versatilidade do diretório ao torná-lo compatível com outros serviços da rede.

`core.schema`	Núcleo do OpenLDAP. Esse *schema* é obrigatório para que o serviço funcione.
`cosine.schema`	O *schema* `cosine.schema` é pré-requisito do `nis.schema`.
`nis.schema`	Contém a estrutura de atributos básica para armazenar informações de autenticação de usuários Linux/Unix.
`inetorgperson.schema`	Especifica os atributos utilizados para armazenar informações de catálogo de endereços (informações sobre os usuários).

O parâmetro *`schemacheck`* serve para configurar se as entradas adicionadas ou modificadas na base de dados serão verificadas para garantir que obedeçam às regras impostas por cada *schema* incluído.

serviço

Essa seção configura alguns aspectos gerais do serviço de diretório. Existem parâmetros que controlam a quantidade de informações que será gerada nos arquivos de *log*, assim como o local onde serão armazenadas as informações sobre o processo em execução no servidor. Também configuramos quais módulos deverão ser carregados.

Abaixo temos um exemplo dessa seção contendo os principais parâmetros:

```
...
# serviço
pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args

loglevel        296

modulepath      /usr/lib/ldap
moduleload      back_bdb
...
```

Esses parâmetros são explicados abaixo:

`pidfile` *`nomedoarquivo`*	Esse parâmetro especifica o local absoluto do arquivo que contém o PID (*Process ID*) do processo slapd em execução no servidor.
`argsfile` *`nomedoarquivo`*	Esse parâmetro especifica o local absoluto do arquivo que contém os parâmetros de linha de comando utilizados pelo processo slapd em execução no servidor.
`loglevel` *`níveldelog`*	Esse parâmetro é configurado através de um inteiro, que representa os tipos de informação que devem ser guardados nos arquivos de *log* do serviço. Esses níveis de informação estão listados na Tabela 2.1, “Níveis de *log* do OpenLDAP”.
`modulepath` *`caminho`*	Especifica o diretório onde estão os módulos que serão carregados dinâmicamente.

`moduleload módulo` Carrega um determinado módulo. No caso estamos carregando o módulo que dará suporte à *backend* BDB (Sleepycat Berkeley Database v4).

Tabela 2.1. Níveis de *log* do OpenLDAP

Nível	Informação gravada
-1	Todas as informações de *log*
0	Nenhuma informação de *log*
1	Chamadas de funções
2	Depuração do manuseamento dos pacotes
4	Depuração detalhada
8	Gerenciamento da conexão
16	Pacotes enviados e recebidos
32	Processamento do filtro de pesquisa
64	Processamento do arquivo de configuração
128	Processamento das listas de controle de acesso
256	Estatísticas para conexão, operações e resultados
512	Estatísticas para resultados devolvidos aos clientes
1024	Comunicação com *backends* de shell
2048	Depuração da análise sintática (*parsing*) das entradas

No exemplo, configuramos o nível de *log* para 296, que é igual a 8 + 32 + 256. Ou seja, estamos guardando informações relacionadas ao gerenciamento da conexão, processamento do filtro de pesquisa e estatísticas para conexão, operações e resultados. Recomenda-se utilizar pelo menos o nível de *log* 256.

Toda essa informação será armazenada através do sistema syslog `LOG_LEVEL4`. Para gravar os *log*'s do slapd em um arquivo diferente, configure o arquivo `/etc/syslog.conf` e reinicie o serviço syslogd. Um exemplo de configuração seria adicionar a seguinte linha ao `/etc/syslog.conf`:

```
local4.debug /var/log/slapd.log
```

Consulte a *manpage* do `syslog.conf` para maiores detalhes.

Atenção

Se nenhum dado estiver sendo guardado com essa configuração, tente criar um arquivo de *log* vazio com o comando **touch**. Algumas versões do syslog precisam que o arquivo de *log* exista antes que comecem a escrever as informações nele.

segurança

Aspectos mais específicos relacionados a segurança serão abordados posteriormente na seção “Aumentando a segurança”. Por enquanto vamos configurar apenas um parâmetro relacionado com a compatibilidade com versões anteriores do protocolo:

```
...
# segurança
```

```
allow           bind_v2
…
```

O parâmetro *allow* e seu complementar *disallow* permitem especificar algumas permissões do serviço. Nessa linha estamos permitindo que clientes LDAP conectem ao nosso diretório utilizando a versão 2 do protocolo. Atualmente a suíte OpenLDAP utiliza a versão 3, mas mantém o suporte à versão 2 pois muitos aplicativos, principalmente clientes de e-mail, ainda a utilizam.

bases de dados

Aqui concentramos todos os parâmetros relacionados às bases de dados que armazenarão as informações do diretório. Primeiro definimos as opções que são específicas de cada *backend* utilizado, depois criamos as instâncias de bases de dados.

```
…
# bases de dados
backend         bdb
checkpoint 512 30

## base de dados no. 1
database        bdb
suffix          "dc=ime,dc=usp,dc=br"
directory       "/var/lib/ldap"
index           objectClass     eq
rootdn          "cn=admin,dc=ime,dc=usp,dc=br"
rootpw          {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==
lastmod         on
mode            0600
cachesize       2000

## ACL's para a base de dados no. 1
access to attrs=userPassword
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by anonymous auth
       by self write
       by * none
access to *
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by * read
…
```

1. **Configurando opções que afetarão todas as bases de dados**

   A diretiva *backend* permite configurar as opções específicas de cada *backend*. No exemplo, estamos configurando opções que são específicas da *backend* BDB. Todas as opções listadas abaixo dessa diretiva serão aplicadas à essa *backend* até que exista outra diretiva *backend* no arquivo de configuração.

   As opções de uma determinada *backend* configuradas utilizando essa diretiva afetarão todas as instâncias de bases de dados que a usam. Por exemplo, se estivéssemos utilizando duas bases de dados, cada uma utilizando uma *backend* diferente, poderíamos incluir duas diretivas *backend* para configurar as opções específicas a cada uma das *backends*.

O parâmetro `checkpoint kbyte min` define quando os *buffers* da base de dados BDB devem ser gravados no disco, como medida de segurança para evitar possíveis perdas. Um limite em *kbytes* e um outro em minutos é fornecido, o que vencer primeiro dispara o *checkpoint* e grava uma entrada no arquivo de *log* para registrar o evento. Esse parâmetro é específico da backend BDB e é recomendado pela configuração padrão do pacote slapd na distribuição Ubuntu. Para maiores informações sobre as opções específicas do BDB, consulte a *manpage* slapd-dbd.

2. **Adicionando bases de dados ao diretório**

A diretiva `database` inicia uma nova instância de uma base de dados, utilizando a *backend* especificada. Todos os parâmetros colocados depois dessa diretiva serão aplicados a essa instância, até que exista outra diretiva `database`. Pode ser interessante, por exemplo, criar bases de dados diferentes para armazenar cada sub-árvore do diretório, ou distribuir essas sub-árvores entre vários servidores e configurá-los de forma que um referencie o outro quando for consultado sobre uma partição do diretório que não está armazenada localmente (semelhante à maneira como o serviço de DNS funciona).

Os seguintes parâmetros configuram a instância de base de dados:

`database` Define a *backend* utilizada pela instância de base de dados. BDB é a *backend* recomendada atualmente pela documentação do OpenLDAP e também a utilizada por padrão na distribuição Ubuntu.

`suffix` Determina o contexto do diretório que será servido por essa base de dados, especificando qual será a sua raiz. No nosso exemplo configuramos o contexto como sendo a raiz do nosso domínio, pois vamos colocar todo o nosso diretório em apenas uma instância de base de dados. O sufixo é escrito utilizando atributos no padrão X.500.

`directory` Configura o local onde serão armazenados os arquivos da base de dados. Se estiver utilizando mais de uma instância de base de dados, é uma boa prática armazenar os arquivos de cada uma em um subdiretório diferente.

`index` Especifica para quais atributos o slapd deve manter índices de forma a otimizar as operações de busca. Existem quatro tipos de índices, e o suporte de um tipo de índice por parte de um atributo é determinado pelo arquivo de *schema* ao qual o atributo pertence. Os quatro tipos de índices são:

`approx` (*approximate*) Indexa a informação de acordo com uma combinação aproximada ou fonética dos valores dos atributos.

`eq` (*equality*) Indexa a informação de acordo com a combinação exata dos valores dos atributos. Essa combinação pode ser sensível a maiúsculas/minúsculas ou a espaços em branco, de acordo com as regras definidas na sintaxe do atributo.

`pres` (*presence*) Indexa a informação de acordo com a presença de valores. Se um atributo não possui um valor, então ele não estará presente na entrada do diretório para efeito de busca.

`sub` (*substring*) Indexa a informação para realizar pesquisas por *substrings* dentro dos valores dos atributos.

Pode-se definir mais de um tipo de índice para o parâmetro *index*, desde que seja suportado pelo atributo e eles estejam separados por vírgula. Também podem existir várias definições *index* para uma mesma base de dados.

rootdn Um diretório LDAP pode ter um usuário *root*, semelhante ao super-usuário dos sistemas Linux/Unix. Quando autenticado, tem acesso irrestrito ao diretório e por essa razão muitos administradores preferem não configurá-lo. O nome do *rootdn* é arbitrário e ele não precisa ser um usuário do sistema, apenas do diretório. Normalmente utiliza-se nomes como *admin* ou *manager*. Esse parâmetro não é obrigatório.

rootpw Especifica a *hash* que contém a senha do *rootdn* e o algoritmo utilizado para gerá-la. Utilize o comando **slappasswd** para gerar a *hash* que será utilizada. A *hash* listada no exemplo foi gerada com o comando **slappasswd -h {MD5} -s *secret***. Se o administrador optar por não configurar um *rootdn* para o diretório esse parâmetro não é necessário.

lastmod Grava informações de tempo em cada modificação e criação das entradas no diretório. Necessário para o *caching* no lado do cliente, pois permite verificar quando a informação foi atualizada.

mode Modo em que os arquivos serão criados. É recomendável permitir acesso de gravação e leitura apenas ao proprietário do processo slapd (0600), que normalmente será o usuário *root* do sistema.

cachesize Parâmetro para melhorar a performance do banco de dados. Especifica quantas entradas devem ser armazenadas em *cache* na memória.

3. **ACL's**

As ACL's (*Access Control Lists*) definem quem tem acesso a qual informação no diretório. Sua sintaxe é muito flexível e não entraremos em detalhes nessa seção, se quiser saber mais a respeito de sua implementação consulte o Apêndice A, *ACL's* no final do texto.

A ordem das ACL's também é relevante, sendo que as que contém as regras mais restritivas devem aparecer antes das mais gerais para que tenham efeito. As configurações das ACL's serão aplicadas apenas à instância de base de dados que as contém.

A primeira configuração listada no exemplo garante direito de acesso ao atributo *userPassword* para escrita (e conseqüentemente leitura) ao usuário *admin*, para autenticação aos usuários anônimos e para escrita aos usuários autenticados (apenas para as suas próprias entradas), ou seja, permite que os usuários alterem a própria senha.

A segunda configuração garante direito de acesso à todo o diretório para escrita (e conseqüentemente leitura) ao usuário *admin* e para leitura aos demais usuários.

Verifique que como a primeira ACL é mais restritiva, ela vai impedir que a segunda ACL garanta direito de leitura das senhas aos usuários comuns. Se a ordem estivesse invertida, a primeira ACL listada perderia seu efeito.

Como dissemos anteriormente, muitos administradores preferem não especificar um *rootdn* para o diretório por questões de segurança. Um dos motivos é o de que será necessário armazenar a *hash* da senha no arquivo de configuração do slapd, o que é inconveniente, mesmo estando com os arquivos devidamente protegidos. O outro motivo é o de que existirá um usuário no diretório para o qual nenhuma ACL terá efeito, tornando-o um perigo em potencial. A alternativa existente é não definir

um *rootdn* e criar um usuário administrador, determinando suas permissões explicitamente através das ACL's. Definimos o *rootdn* nesse caso apenas para ilustrar uma possibilidade de configuração.

Dica

Ao finalizar a configuração do seu arquivo `/etc/ldap/slapd.conf`, execute o comando **slaptest** para verificar se está tudo certo:

```
usuario@ldapserver:~$ sudo slaptest
config file testing succeeded
```

Inicializando a base de dados

Antes de adicionar informações ao diretório, precisamos estabelecer a sua estrutura básica. No momento, vamos precisar pelo menos da raiz do diretório e do usuário que será utilizado para administrá-lo.

Para inicializar o diretório com essas informações, podemos utilizar arquivos LDIF contendo essas entradas. Para maiores informações sobre os arquivos LDIF consulte o Apêndice B, *Arquivo LDIF*.

Abaixo temos um exemplo de um arquivo LDIF que contém a raiz do nosso diretório:

Exemplo 2.2. Arquivo `base.ldif`

```
############################## base.ldif ##############################
# Arquivo LDIF contendo a entrada com as informações da raiz do
# diretório LDAP.
dn: dc=ime,dc=usp,dc=br
objectClass: domain
dc: ime
```

Este outro arquivo contém as informações do usuário administrador:

Exemplo 2.3. Arquivo `admin.ldif`

```
############################## admin.ldif ##############################
# Arquivo LDIF contendo a entrada com as informações do usuário
# administrador do diretório LDAP.
dn: cn=admin,dc=ime,dc=usp,dc=br
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: Administrador do LDAP
# Hash gerada para a senha "secret" utilizada no nosso exemplo
userPassword: {MD5}Xr4ilOzQ4PCOq3aQ0qbuaQ==
```

Para adicionar essas informações ao diretório, execute os seguintes comandos:

```
usuario@ldapserver:~$ sudo slapadd -l base.ldif
usuario@ldapserver:~$ sudo slapadd -l admin.ldif
```

Em seguida, verifique as entradas adicionadas:

```
usuario@ldapserver:~$ sudo slapcat
dn: dc=ime,dc=usp,dc=br
objectClass: domain
dc: ime
structuralObjectClass: domain
entryUUID: 88c92866-175a-102b-91c6-8d4ba6b5c9fa
creatorsName: cn=admin,dc=ime,dc=usp,dc=br
modifiersName: cn=admin,dc=ime,dc=usp,dc=br
createTimestamp: 20061203204228Z
modifyTimestamp: 20061203204228Z
entryCSN: 20061203204228Z#000001#00#000000

dn: cn=admin,dc=ime,dc=usp,dc=br
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: Administrador do LDAP
userPassword:: e01ENX1YcjRpbE96UTRQQ09xM2FRMHFidWFRPT0=
structuralObjectClass: organizationalRole
entryUUID: 8d3995ca-175a-102b-8057-e850202b49a7
creatorsName: cn=admin,dc=ime,dc=usp,dc=br
modifiersName: cn=admin,dc=ime,dc=usp,dc=br
createTimestamp: 20061203204235Z
modifyTimestamp: 20061203204235Z
entryCSN: 20061203204235Z#000001#00#000000
```

Aumentando a segurança

Antes de colocar o serviço de diretório em um ambiente de produção, é recomendável fazer alguns ajustes para garantir que as informações consultadas pelos clientes não sejam vistas por pessoas não-autorizadas.

Tendo em vista que na maioria das redes, os serviços que consultarão as informações contidas no diretório (como autenticação, por exemplo) não estarão necessariamente no mesmo servidor que contém o serviço de diretório, torna-se necessário lançar mão de recursos que permitam uma comunicação segura entre as máquinas.

Um desses recursos, disponibilizado pelo slapd, é o suporte a transações TLS (Transport Layer Security). O TLS é uma espécie de reimplementação do SSL, mais eficiente e segura, e por isso tem sido adotada na maioria dos serviços atualmente.

Para configurar o suporte a TLS no slapd, primeiro precisamos criar um certificado SSL, conforme descrito no Apêndice C, *Gerando um certificado SSL auto-assinado*. Com os arquivos do certificado devidamente gerados, vamos agora armazená-los no local correto segundo a distribuição Ubuntu.

1. Copie o arquivo do certificado para o diretório `/etc/ssl/certs`, com o nome seguindo o mesmo padrão da distribuição:

   ```
   usuario@ldapserver:~$ sudo cp newreq.pem /etc/ssl/certs/ssl-cert-
   ldapserver.pem
   ```

2. Copie o arquivo da chave privada *sem senha* para o diretório `/etc/ssl/private`, seguindo também esse padrão:

```
usuario@ldapserver:~$ sudo cp openkey.pem /etc/ssl/private/ssl-cert-ldapserver.key
```

Cuidado

O arquivo contendo a chave privada do certificado que será usado pelo slapd precisa ser armazenado sem a senha de proteção, caso contrário o serviço poderá travar ao tentar iniciar automaticamente junto com o sistema operacional. Apenas utilize um arquivo protegido por senha se você quiser inicializar o serviço de diretório manualmente.

Agora precisamos ajustar as permissões adequadas para esses arquivos.

1. Primeiro proteja o arquivo contendo a chave privada:

```
usuario@ldapserver:~$ sudo chown root:ssl-cert /etc/ssl/private/ssl-cert-ldapserver.key
usuario@ldapserver:~$ sudo chmod 640 /etc/ssl/private/ssl-cert-ldapserver.key
```

2. Em seguida ajuste o acesso ao arquivo do certificado:

```
usuario@ldapserver:~$ sudo chown root:root /etc/ssl/certs/ssl-cert-ldapserver.pem
usuario@ldapserver:~$ sudo chmod 644 /etc/ssl/certs/ssl-cert-ldapserver.pem
```

Agora precisamos alterar a seção de segurança do arquivo `/etc/ldap/slapd.conf`, para que ela fique dessa maneira:

```
…
# segurança
allow           bind_v2

TLSCACertificateFile    /etc/ssl/certs/ssl-cert-ldapserver.pem
TLSCertificateFile      /etc/ssl/certs/ssl-cert-ldapserver.pem
TLSCertificateKeyFile   /etc/ssl/private/ssl-cert-ldapserver.key
…
```

TLSCACertificateFile

O certificado que criamos é auto-assinado, por isso configuramos o parâmetro do certificado do CA (*`TLSCACertificateFile`*) com o mesmo arquivo que o parâmetro do certificado do servidor (*`TLSCertificateFile`*). Caso esteja usando um certificado assinado por uma CA ao invés de um certificado auto-assinado, ajuste o parâmetro *`TLSCACertificateFile`* para o certificado da CA responsável pela sua emissão, que deve estar contido no servidor.

Para que as alterações tenham efeito, é necessário reiniciar o serviço slapd:

```
usuario@ldapserver:~$ sudo /etc/init.d/slapd restart
Stopping OpenLDAP: slapd.
Starting OpenLDAP: running BDB recovery, slapd.
```

Se o serviço iniciar normalmente, é sinal de que tudo correu bem. Caso contrário, ocorreu algum erro na configuração ou nos arquivos do certificado. Verifique todos os passos novamente em caso de problemas.

Você pode agora testar se o certificado está funcionando corretamente. Execute o seguinte comando para realizar uma operação de busca com suporte a TLS no servidor LDAP:

```
usuario@ldapserver:~$ ldapsearch -x -b 'dc=ime,dc=usp,dc=br' -D
"cn=admin,dc=ime,dc=usp,dc=br" '(objectclass=*)' -H
ldap://ldapserver.ime.usp.br -W -ZZ
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base dc=ime,dc=usp,dc=br with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# ime.usp.br
dn: dc=ime,dc=usp,dc=br
objectClass: domain
dc: ime

# admin, ime.usp.br
dn: cn=admin,dc=ime,dc=usp,dc=br
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: Administrador do LDAP
userPassword:: e01ENX1YcjRpbE96UTRQQ09xM2FRMHFidWFRPT0=

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Capítulo 3. Integração

Resumo

Esse capítulo descreve o processo de integração do serviço de diretório com os serviços de autenticação para redes Linux/Unix e Windows.

Em redes Linux/Unix a integração consiste em instalar e configurar os módulos que permitam aos serviços NSS e PAM buscarem as informações de autenticação no diretório. Para integrar a rede Windows, vamos instalar e configurar o Samba como um PDC (*Primary Domain Controller*) que irá operar exclusivamente como servidor de autenticação ligado ao OpenLDAP, de forma a fornecer uma ponte transparente entre as estações Windows e o serviço de diretório.

O procedimento de integração, que consiste em instalar e configurar os pacotes necessários, é realizado no lado cliente em redes Linux/Unix (instalação dos módulos libnss-ldap e libpam-ldap) e é realizado no lado servidor em redes Windows (instalação do Samba).

NSS

O NSS *(Name Service Switch)* é o serviço responsável por realizar as pesquisas das bases de dados em ambientes Unix. Ele permite configurar várias fontes para a realização de pesquisas e é utilizado por vários outros serviços para realizar a recuperação da informação armazenada na rede.

O pacote libnss-ldap é o *plugin* do LDAP para o NSS, ou seja, ele é que permitirá ao NSS realizar buscas no diretório LDAP. Esse módulo será então utilizado pelo NSS como fonte para as informações de autenticação da rede.

Procedimento 3.1. Instalação

- Execute o seguinte comando para instalar os pacotes necessários no cliente:

```
usuario@cliente:~$ sudo aptitude install ldap-client libnss-ldap
```

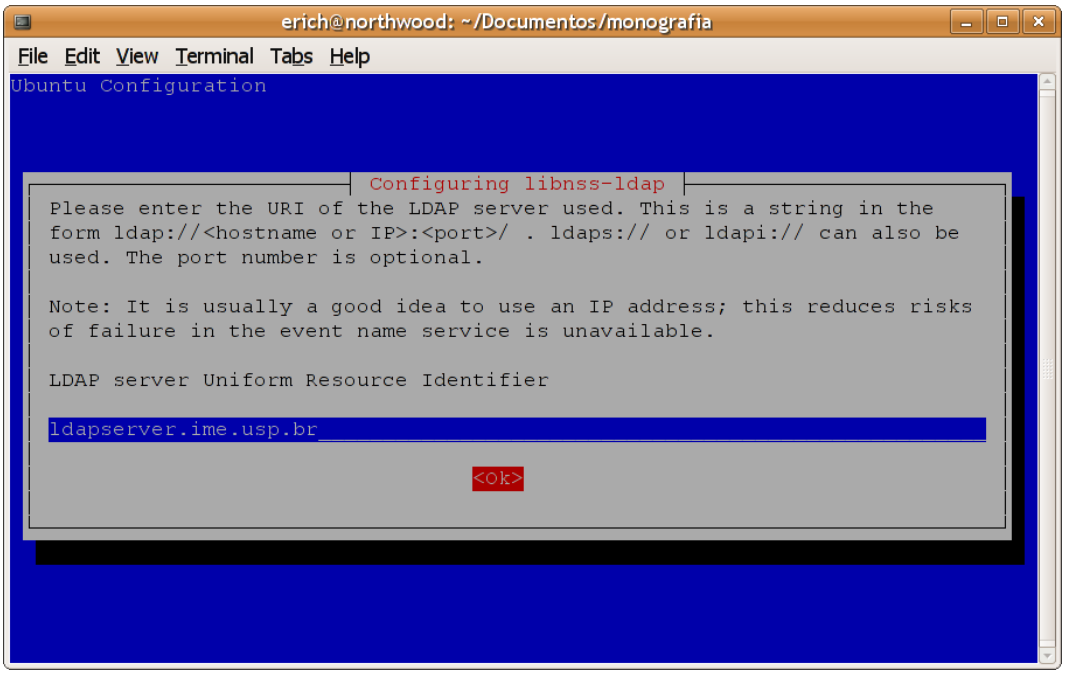
Cuidado

Esse pacote pertence ao repositório *Universe* da distribuição Ubuntu. Para saber como adicionar os repositórios extras à lista de repositórios do gerenciador de pacotes, consulte a documentação do Ubuntu: *Extra Repositories* [https://help.ubuntu.com/6.06/ubuntu/desktopguide/C/extra-repositories.html#id2580924].

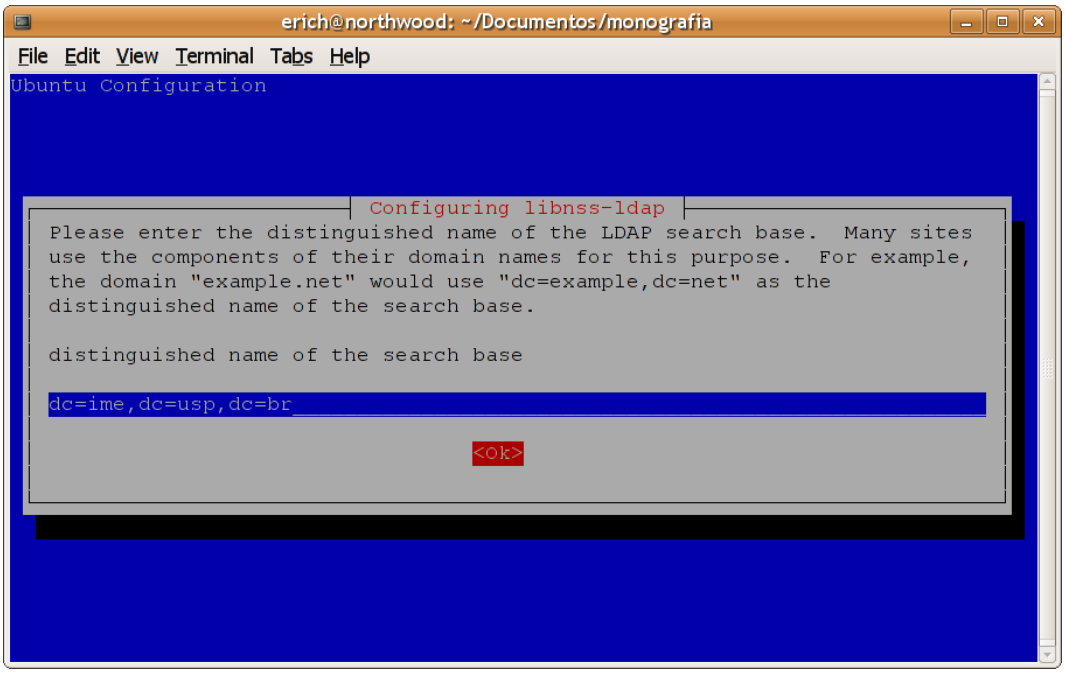
Procedimento 3.2. Configuração

O processo de configuração do pacote libnss-ldap deve iniciar automaticamente após a instalação. A seguir explicaremos esse processo para cada tela apresentada. O processo pode ser repetido a qualquer momento, utilizando o seguinte comando:

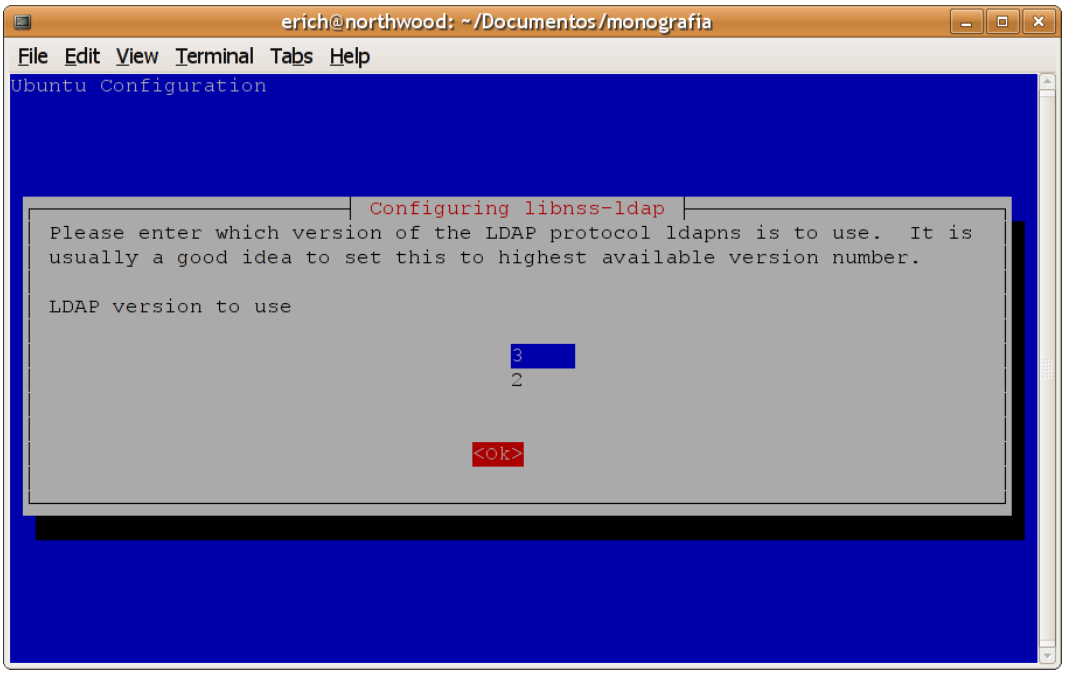
```
usuario@cliente:~$ sudo dpkg-reconfigure libnss-ldap
```

1. 

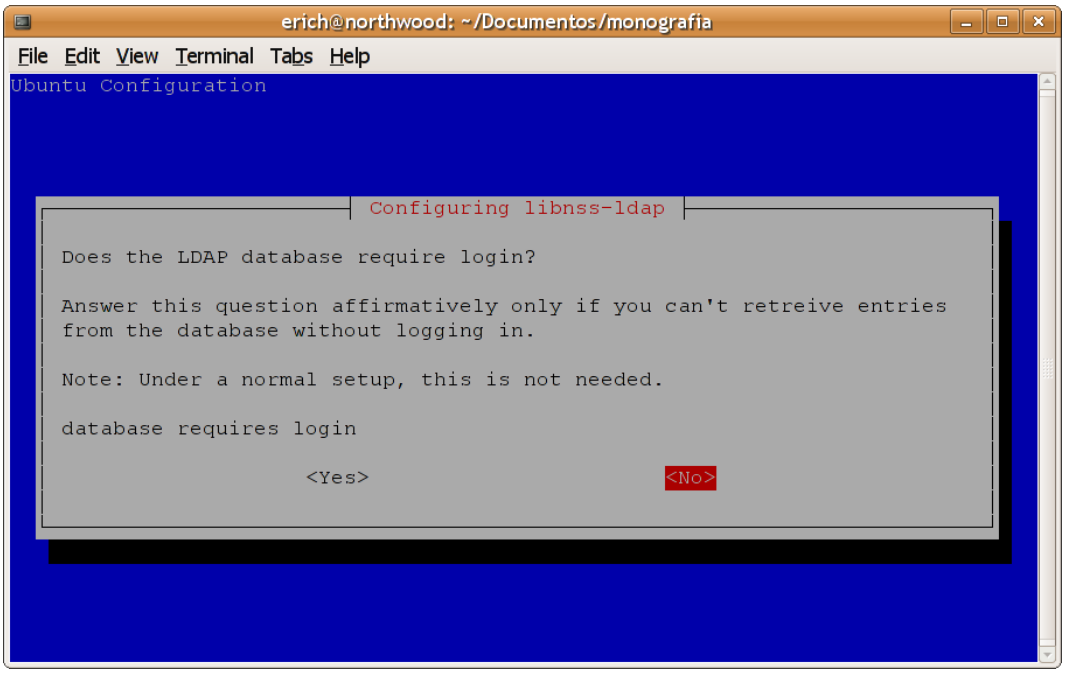
A primeira tela pede o endereço do servidor LDAP.

2. 

Aqui você precisa informar qual é a raiz (ou base) do diretório LDAP.

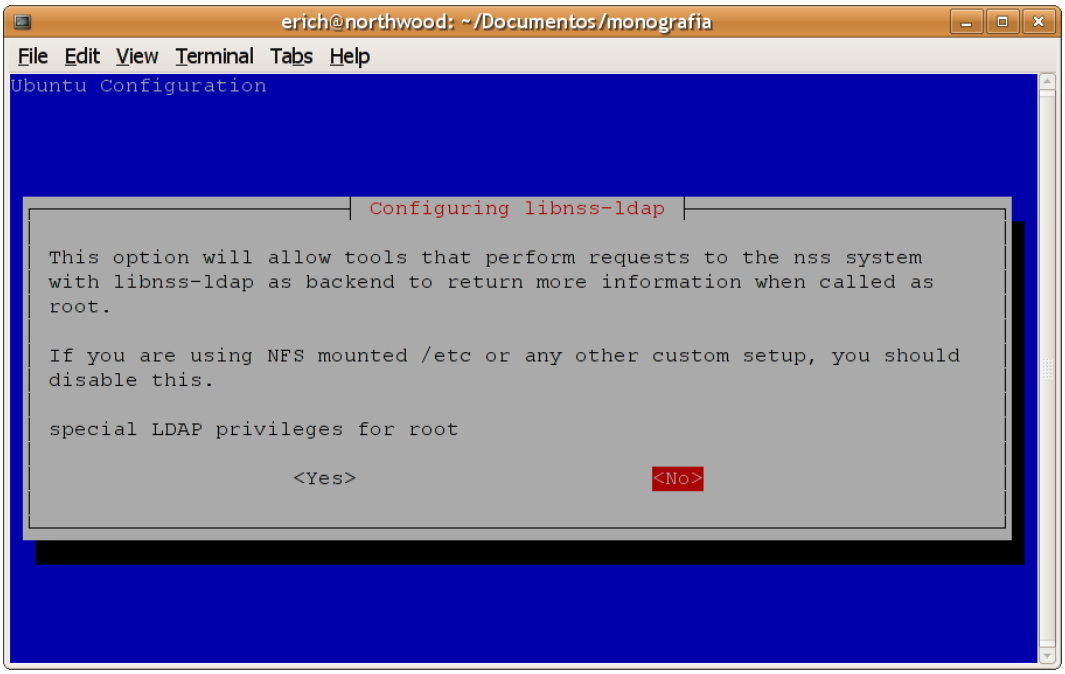
3. 

Configure para a versão 3 do protocolo LDAP.

4. 

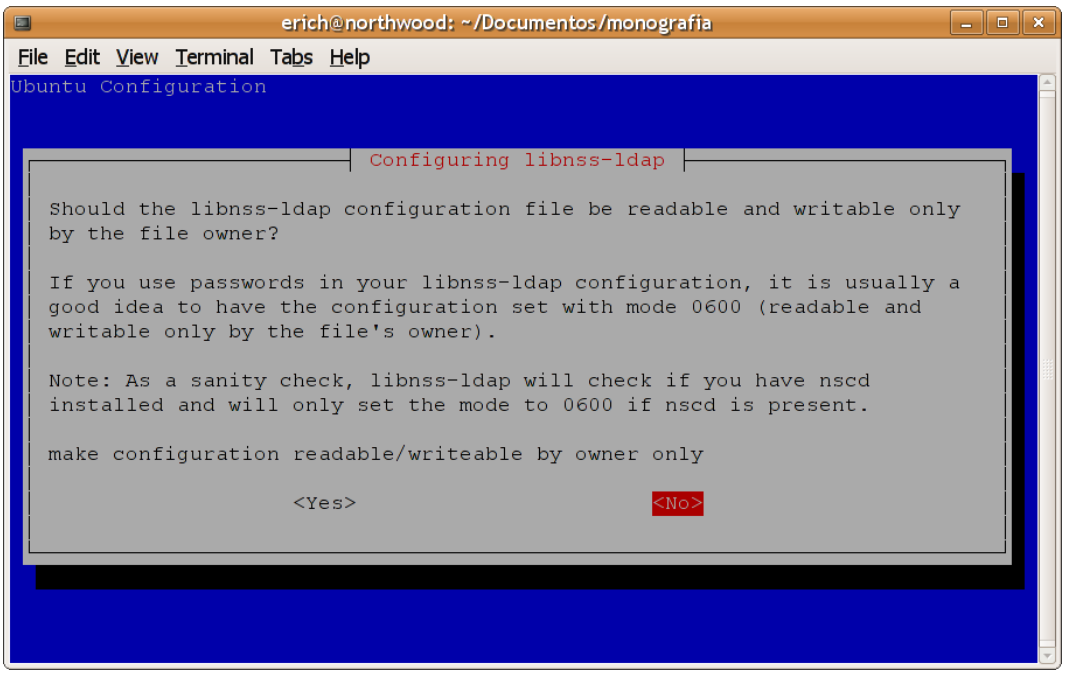
Nossa base de dados não precisará de autenticação para leitura. Uma alternativa é restringir a leitura apenas ao administrador do diretório ou outro usuário e configurá-lo nessa opção, ao custo de ter que deixar sua senha armazenada em texto puro no arquivo de configuração.

5.

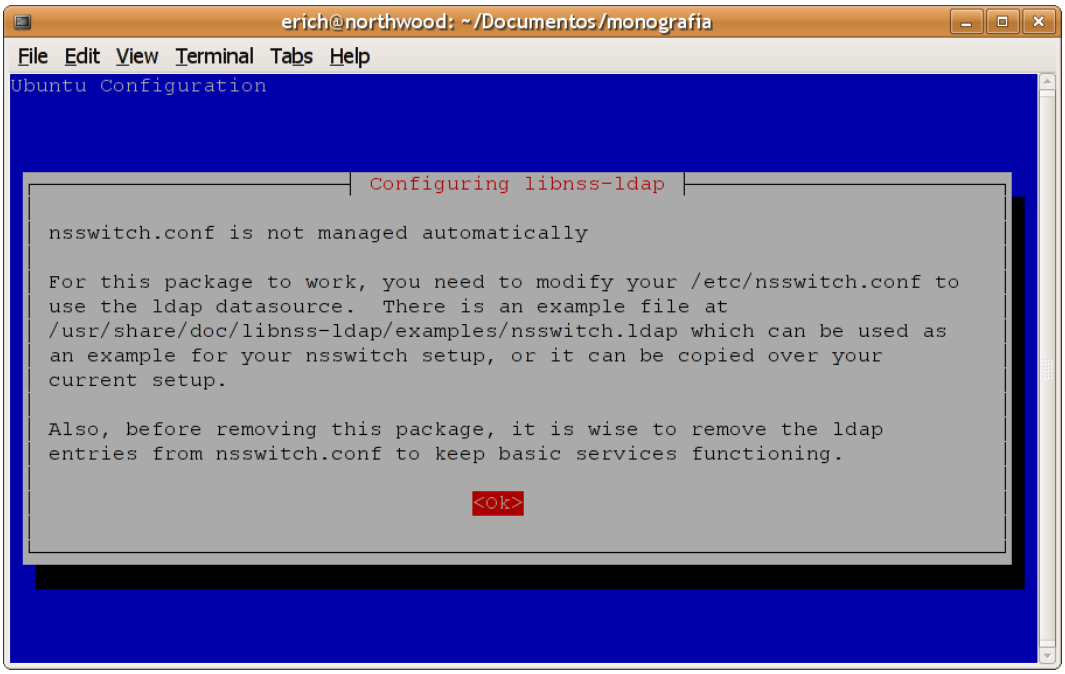


Também não vamos precisar de um usuário com privilégios no NSS. Se isto for necessário para algum outro serviço que não seja autenticação, pode-se configurar essa opção com o administrador do diretório, mas sua senha terá que ficar armazenada no arquivo `/etc/libnss-ldap.secret`.

6.



Não precisaremos proteger a leitura a esse arquivo, pois não estamos armazenando informações confidenciais nele. Isso permitirá aos usuários consultar o diretório através de ferramentas como **finger** e **id**.

7. 

   O gerenciamento do arquivo `/etc/nsswitch.conf` é feito manualmente. Vamos configurá-lo a seguir.

8. Para configurar o NSS para usar o módulo libnss-ldap, devemos alterar o arquivo `/etc/nsswitch.conf`. Basta adicionar o módulo *`ldap`* como fonte de pesquisa para as bases de dados *`passwd`*, *`group`* e *`shadow`*. Um exemplo desse arquivo com as alterações necessárias é apresentado abaixo:

Exemplo 3.1. Arquivo de configuração `/etc/nsswitch.conf`

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

# As fontes serão pesquisadas na ordem em que são listadas. Para
# evitar problemas de login no servidor caso ocorra algum problema
# com o serviço slapd, é uma boa opção deixar o ldap como segunda
# opção.

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

hosts:          files dns mdns
networks:       files

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

netgroup:       nis
```

Importante

Certifique-se de que as permissões do arquivo `/etc/nsswitch.conf` estão ajustadas para 644 e de que o seu proprietário seja o usuário *root*.

PAM

O PAM *(Pluggable Authentication Modules)* é o serviço responsável por realizar a autenticação de usuários nos ambientes Linux/Unix. Através do PAM e de suas bibliotecas, é possível configurar um esquema de autenticação segura para qualquer aplicação de forma transparente.

O pacote libpam-ldap é o *plugin* do LDAP para o PAM, ou seja, ele é que permitirá ao PAM autenticar usuários armazenados no diretório LDAP. O PAM apenas realiza a autenticação dos usuários que foram reconhecidos durante a pesquisa do NSS, ou seja, é necessário instalar e configurar o módulo libnss-ldap antes do libpam-ldap para que a autenticação dos usuários do diretório LDAP seja efetuada.

Procedimento 3.3. Instalação

- O pacote libpam-ldap deverá ser instalado junto com o pacote libnss-ldap normalmente, no entanto, caso isso não ocorra execute o seguinte comando:

```
usuario@cliente:~$ sudo aptitude install libnss-ldap
```

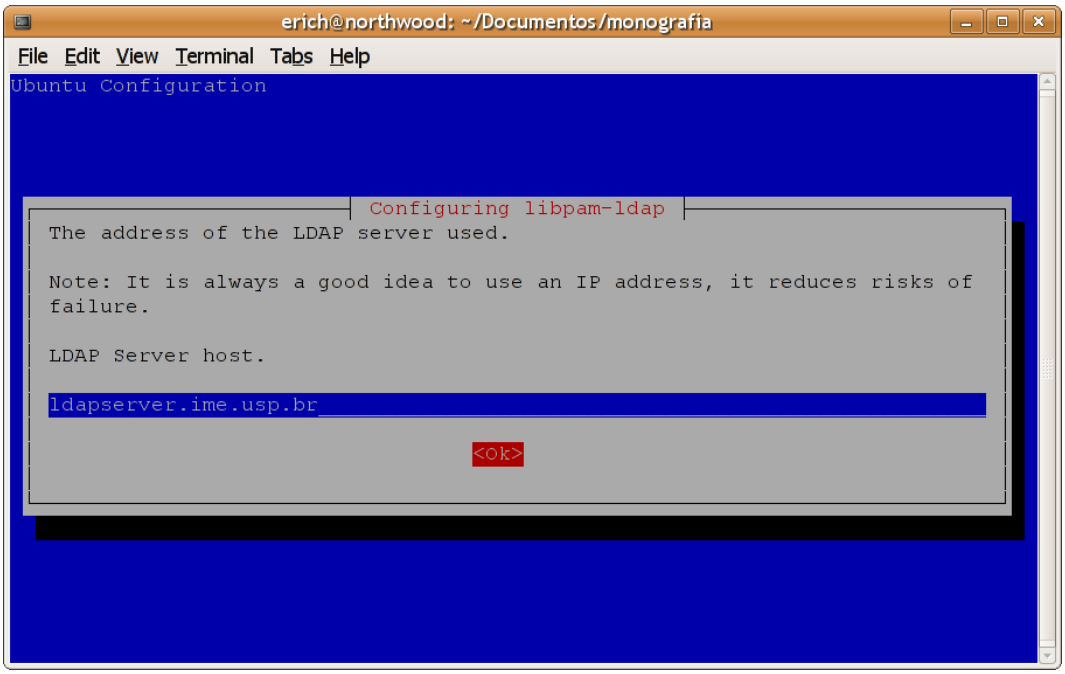
Cuidado

Esse pacote pertence ao repositório *Universe* da distribuição Ubuntu. Para saber como adicionar os repositórios extras à lista de repositórios do gerenciador de pacotes, consulte a documentação do Ubuntu: *Extra Repositories* [https://help.ubuntu.com/6.06/ubuntu/desktopguide/C/extra-repositories.html#id2580924].

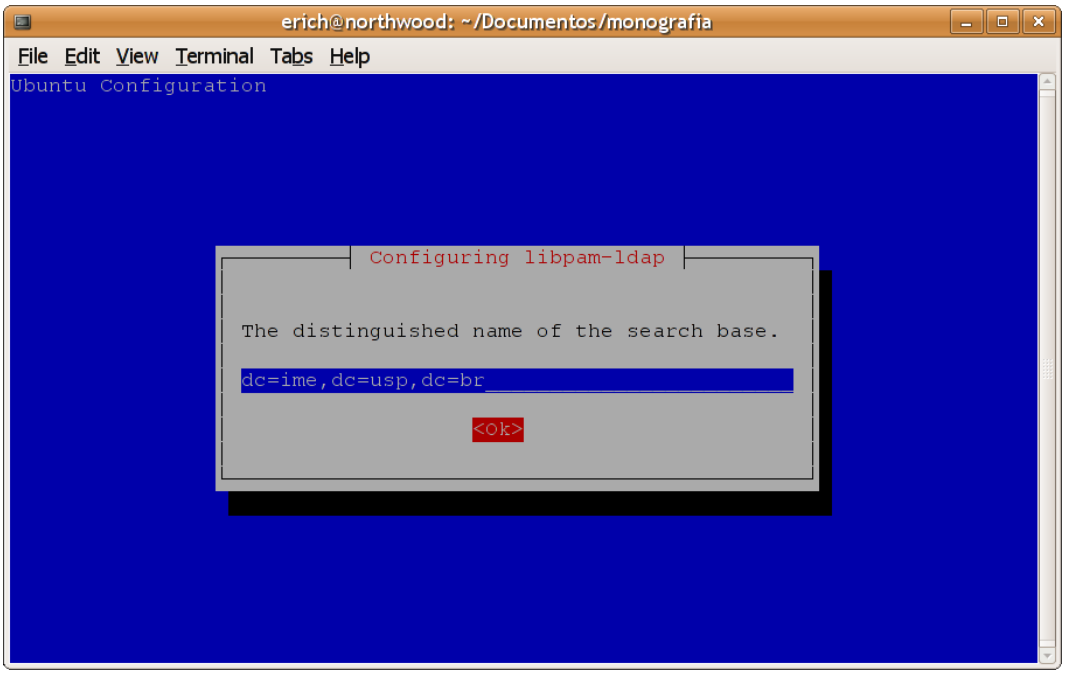
Procedimento 3.4. Configuração

O processo de configuração do pacote libpam-ldap deve iniciar automaticamente após a instalação. Caso ele tenha sido instalado automaticamente junto com o libnss-ldap, sua configuração será realizada após a primeira. A seguir explicaremos esse processo para cada tela apresentada. O processo pode ser repetido a qualquer momento, utilizando o seguinte comando:

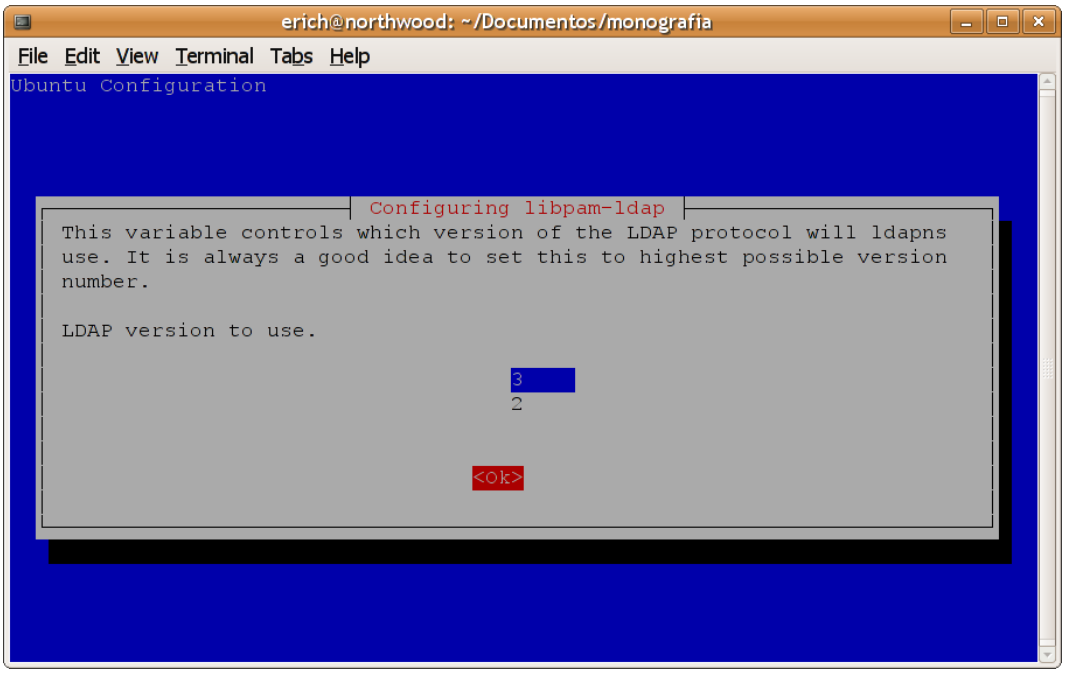
```
usuario@cliente:~$ sudo dpkg-reconfigure libpam-ldap
```

1. 

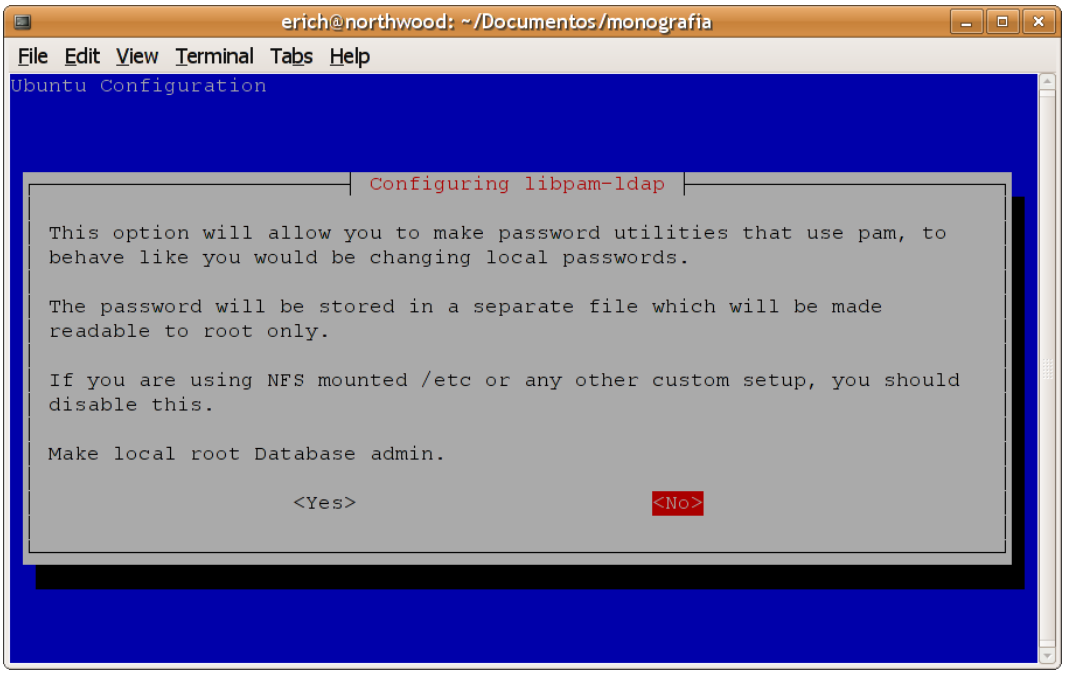
   Digite o endereço do servidor LDAP.

2. 

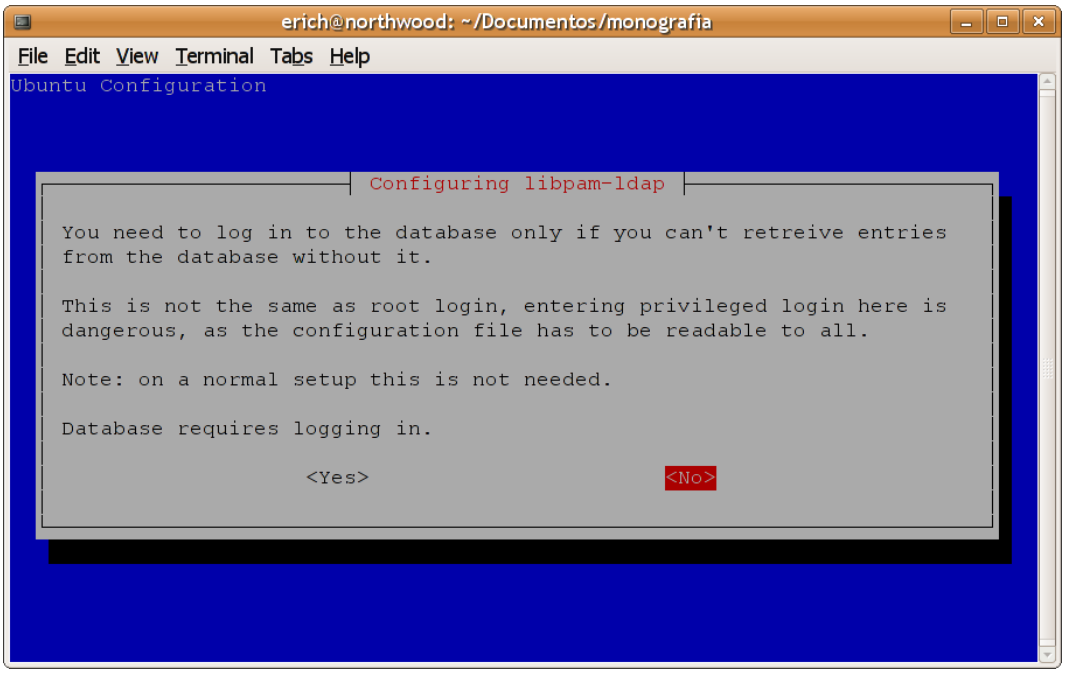
Digite a raiz (ou base) do diretório LDAP.

3. 

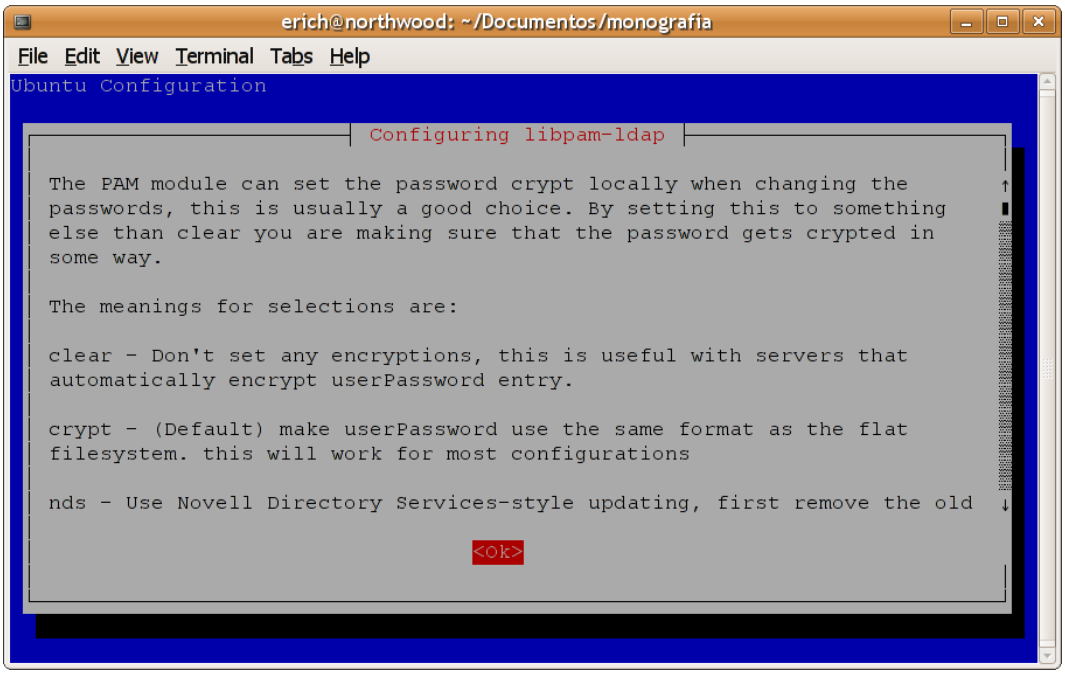
Configure para a versão 3 do protocolo LDAP.

4. 

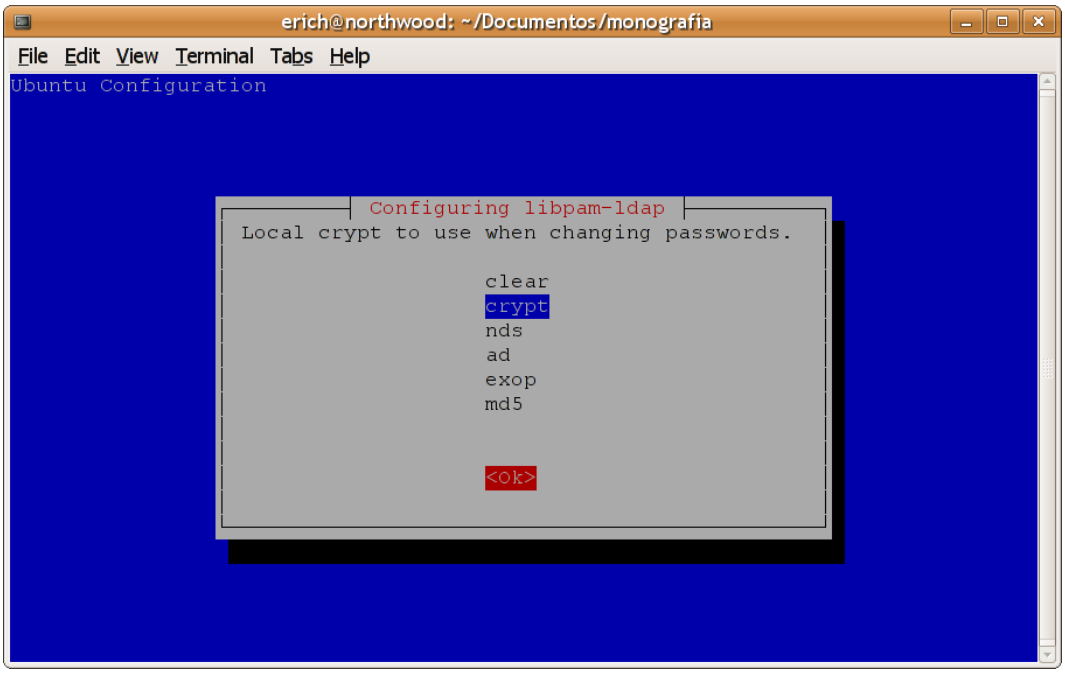
Selecione não para essa opção também.

5. 

Configure a base de dados para que ela não precise de autenticação.

6. 

Na próxima tela vamos configurar o algoritmo utilizado para criptografar as senhas nos casos em que um algoritmo não for explicitamente especificado.

7. 

Apenas mantenha a opção padrão (*crypt*) e confirme.

8. Precisamos agora alterar os arquivos `/etc/pam.d/common-account`, `/etc/pam.d/common-auth` e `/etc/pam.d/common-password`. Os exemplo abaixo ilustram esses arquivos com as alterações necessárias para que o PAM utilize os usuários do diretório para autenticação do sistema:

   Listagem do arquivo `/etc/pam.d/common-account`:

Exemplo 3.2. Arquivo `/etc/pam.d/common-account`

```
#
# /etc/pam.d/common-account - authorization settings common to all
# services
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
account sufficient      pam_ldap.so
account required        pam_unix.so
```

Listagem do arquivo `/etc/pam.d/common-account`:

Exemplo 3.3. Arquivo `/etc/pam.d/common-auth`

```
#
# /etc/pam.d/common-auth - authentication settings common to all
# services
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
auth    sufficient      pam_ldap.so
auth    required        pam_unix.so nullok_secure use_first_pass
```

Nota

A opção *`use_first_pass`* utilizada na última linha do arquivo `/etc/pam.d/common-auth` evita que o usuário digite a senha duas vezes durante a autenticação.

Listagem do arquivo `/etc/pam.d/common-account`:

Exemplo 3.4. Arquivo `/etc/pam.d/common-password`

```
#
# /etc/pam.d/common-password - password-related modules common to all
# services
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define  the services to be
# used to change user passwords.  The default is pam_unix

# The "nullok" option allows users to change an empty password, else
# empty passwords are treated as locked accounts.
#
# (Add `md5' after the module name to enable MD5 passwords)
#
# The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
# login.defs. Also the "min" and "max" options enforce the length of the
# new password.

password   sufficient pam_ldap.so
password   required   pam_unix.so nullok obscure min=4 max=8 md5

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#
# password required       pam_cracklib.so retry=3 minlen=6 difok=3
# password required       pam_unix.so use_authtok nullok md5
```

9. O PAM precisará se comunicar com o servidor LDAP através de TLS para garantir a segurança dos dados transmitidos, nos casos em que ele for instalado em uma máquina diferente da que está rodando o serviço slapd. Para isso, precisamos descomentar a seguinte linha no arquivo `/etc/pam_ldap.conf`:

```
...
ssl start_tls
...
```

Também será necessário ter um arquivo `/etc/ldap/ldap.conf` com o seguinte conteúdo:

Exemplo 3.5. Arquivo de configuração `/etc/ldap/ldap.conf`

```
######################## /etc/ldap/ldap.conf ######################
# Arquivo de configuração para os clientes do diretório LDAP.

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=ime,dc=usp,dc=br
URI     ldap://ldapserver.ime.usp.br

TLS_CACERT      /etc/ssl/certs/ssl-cert-ldapserver.pem
```

Cuidado

Este arquivo deve ter as permissões ajustadas para 644.

O certificado do servidor (caso ele seja do tipo auto-assinado) ou o da CA que o certificou deve ser colocado no diretório `/etc/ssl/certs` e sua permissão de acesso deve ser ajustada para 644 também. Altere o valor do parâmetro *`TLS_CACERT`* de acordo com o nome do arquivo do certificado.

Samba

O Samba é um servidor SMB *open-source* para ambientes Linux/Unix. Com o Samba é possível participar de um domínio Windows, tanto como membro quanto como PDC (*Primary Domain Controller*). Isso significa que o Samba pode ser usado para autenticar os usuários e computadores de um domínio Windows.

A seguir vamos explicar o procedimento de instalação e configuração do Samba em um servidor que irá atuar como PDC, com o propósito de realizar a autenticação dos usuários e máquinas do domínio Windows, de forma integrada com o nosso diretório LDAP. Dessa forma, os usuários que estão armazenados no diretório LDAP poderão utilizar tanto os terminais Linux/Unix da rede quanto as estações de trabalho Windows, e o gerenciamento desses dois ambientes ficará centralizado.

Será possível garantir permissão de acesso a um usuário apenas aos terminais Linux/Unix ou apenas às estações Windows também, se necessário. E também será possível manter as senhas de acesso sincronizadas, de tal forma que se um usuário mudar sua senha na linha de comando do Linux ela também será alterada para a autenticação no Windows e vice-versa.

Essa automação no gerenciamento dos registros que estão no diretório LDAP por parte do Samba é feita através de *scripts* auxiliares que vamos instalar e configurar.

Procedimento 3.5. Instalação

- Execute o seguinte comando para instalar os pacotes necessários no servidor que disponibilizará o Samba:

```
usuario@sambaserver:~$ sudo aptitude install samba smbldap-tools
```

Procedimento 3.6. Configuração

1. Vamos agora configurar o Samba para atuar como um PDC pronto para autenticar os usuários armazenados no diretório LDAP a partir das estações Windows. Não vamos entrar nos detalhes dos parâmetros de configuração do Samba pois além de serem muito numerosos isso foge do escopo deste documento. Para obter mais informações a respeito dos parâmetros de configuração do samba, consulte a *manpage* do `smb.conf` ou acesse o site *The Official Samba-3 HOWTO and Reference Guide* [http://us3.samba.org/samba/docs/man/Samba-HOWTO-Collection/]. Um exemplo do arquivo `/etc/samba/smb.conf` apropriado para realizar a função de autenticação dos usuários Windows é apresentado abaixo:

Exemplo 3.6. Arquivo de configuração `/etc/ldap/ldap.conf`

```
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentary and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.
#

#======================= Global Settings =======================

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will
# part of
# Não precisa ser igual á raiz do diretório LDAP
   workgroup = ldap.ime.usp.br

# server string is the equivalent of the NT Description field
   server string = Servidor SAMBA
   netbios name = sambaserver

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS
# Server
   wins support = yes

# This will prevent nmbd to search for NetBIOS names through DNS.
   dns proxy = no

#### Debugging/Accounting ####

# This tells Samba to use a separate log file for each machine
# that connects
   log level = 2
   log file = /var/log/samba/log.%m

# Put a capping on the size of the log files (in Kb).
```

```
   max log size = 1000

# We want Samba to log a minimum amount of information to syslog.
# Everything should go to /var/log/samba/log.{smbd,nmbd} instead.
# If you want to log through syslog you should set the following
# parameter to something higher.
   syslog = 0

# Do something sensible when Samba crashes: mail the admin a backtrace
   panic action = /usr/share/samba/panic-action %d

####### Authentication #######

# "security = user" is always a good idea. This will require a Unix
# account in this server for every user accessing the server. See
# /usr/share/doc/samba-doc/htmldocs/Samba-HOWTO-Collection/ServerType.
# html
# in the samba-doc package for details.
   security = user
   admin users = root

# You may wish to use password encryption.  See the section on
# 'encrypt passwords' in the smb.conf(5) manpage before enabling.
   encrypt passwords = true

# If you are using encrypted passwords, Samba will need to know what
# password database type you are using.
   passdb backend = ldapsam:ldap://ldapserver.ime.usp.br/
   passdb expand explicit = no

   obey pam restrictions = no

# Script para alterar e sincronizar as senhas dos usuarios
   ldap passwd sync = Yes
   passwd program = /usr/sbin/smbldap-passwd %u
   passwd chat = *New*password* %n\n *Retype*new*password* %n\n
                 *all*authentication*tokens*updated*

   ldap ssl = start_tls
   ldap admin dn = cn=admin,dc=ime,dc=usp,dc=br
   ldap suffix = dc=ime,dc=usp,dc=br
   ldap group suffix = ou=Groups
   ldap user suffix = ou=Users
   ldap machine suffix = ou=Computers
   ldap idmap suffix = ou=Idmap
   ldap delete dn = Yes

# Configurações dos scripts do pacote smbldap-tools
   add user script = /usr/sbin/smbldap-useradd -m "%u"
   add machine script = /usr/sbin/smbldap-useradd -t 0 -w "%u"
   add group script = /usr/sbin/smbldap-groupadd -p "%g"
   add user to group script = /usr/sbin/smbldap-groupmod -m "%u" "%g"
   delete user script = /usr/sbin/smbldap-userdel "%u"
   delete group script = /usr/sbin/smbldap-groupdel "%g"
```

```
   delete user from group script = /usr/sbin/smbldap-groupmod -x "%u" "%g"
   set primary group script = /usr/sbin/smbldap-usermod -g "%g" "%u"

########## Domains ###########

# Is this machine able to authenticate users. Both PDC and BDC
# must have this setting enabled. If you are the BDC you must
# change the 'domain master' setting to no
#
   domain logons = yes
   enable privileges = yes

############ Misc ############

# Most people will find that this option gives better performance.
# See smb.conf(5) and /usr/share/doc/samba-doc/htmldocs/speed.html
# for details
# You may want to add the following on a Linux system:
   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# Domain Master specifies Samba to be the Domain Master Browser. If this
# machine will be configured as a BDC (a secondary logon server), you
# must set this to 'no'; otherwise, the default behavior is recommended.
   domain master = auto

# Some defaults for winbind (make sure you're not using the ranges
# for something else.)
   idmap uid = 10000-20000
   idmap gid = 10000-20000
```

Cuidado

Não se esqueça de ajustar as permissões do arquivo `/etc/samba/smb.conf` para 644 e certifique-se de que o seu proprietário seja o usuário *root*.

Dica

Ao finalizar a configuração do seu arquivo `/etc/samba/smb.conf`, execute o comando **testparm** para verificar se está tudo certo:

```
usuario@sambaserver:~$ sudo testparm
Loaded services file OK.
Server role: ROLE_DOMAIN_PDC
Press enter to see a dump of your service definitions
…
```

2. Agora teremos que adicionar o *schema* do Samba no arquivo `/etc/ldap/slapd.conf` para que os atributos das entradas do domínio Windows sejam reconhecidos pelo serviço de diretório. Para isso, primeiro teremos que obter esse arquivo, que está contido no pacote samba-doc. Execute o seguinte comando no servidor que está rodando o serviço slapd para instalar o pacote necessário:

```
usuario@ldapserver:~$ sudo aptitude install samba-doc
```

Depois execute o comando abaixo para extrair o arquivo necessário para o local correto:

```
usuario@ldapserver:~$ sudo zcat /usr/share/doc/samba-doc/examples
/LDAP/samba.schema.gz > /etc/ldap/schema/samba.schema
```

A seção *schema*'s do arquivo `/etc/ldap/slapd.conf` ficará assim:

```
…
# schema's
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema
include         /etc/ldap/schema/samba.schema

schemacheck     on
…
```

Vamos alterar a seção base de dados também para que fique de acordo com as nossas alterações. Primeiro precisamos indexar a instância de base de dados que contém o diretório de maneira diferente para obter um desempenho adequado ao buscar registros do Samba:

```
…
## base de dados no. 1
…
index           objectClass                             eq
index           uid,uidNumber,gidNumber,memberUid       eq
index           cn,mail,surname,givenname               eq,subinitial
index           sambaSID                                eq
index           sambaPrimaryGroupSID                    eq
index           sambaDomainName                         eq
…
```

Também vamos precisar alterar as ACL's dessa instância para proteger os dados confidenciais desses registros:

```
…
## ACL's para a base de dados no. 1
access to attrs=userPassword,sambaNTPassword,sambaLMPassword
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by anonymous auth
       by self write
       by * none
…
```

Como alteramos as configurações dos índices da base de dados, teremos que reconstruí-los. Para isso, primeiro pare o serviço slapd:

```
usuario@ldapserver:~$ sudo /etc/init.d/slapd stop
```

Em seguida reconstrua os índices com o comando **slapindex**:

```
usuario@ldapserver:~$ sudo slapindex
```

Agora reinicie o serviço:

```
usuario@ldapserver:~$ sudo /etc/init.d/slapd start
Starting OpenLDAP: running BDB recovery, slapd.
```

3. Precisamos armazenar a senha do usuário que será usado para administrar o diretório LDAP no arquivo `/var/lib/samba/secrets.tdb`. Para isso, execute o seguinte comando no servidor em que o Samba foi instalado:

```
usuario@sambaserver:~$ sudo /usr/bin/smbpasswd -w secret
Setting stored password for "cn=admin,dc=ime,dc=usp,dc=br" in secrets.tdb
```

4. Finalmente, reinicie o servidor Samba:

```
usuario@sambaserver:~$ sudo /etc/init.d/samba restart
 * Stopping Samba daemons...                              [ ok ]
 * Starting Samba daemons...                              [ ok ]
```

5. Como dissemos anteriormente, a automação do gerenciamento dos registros do diretório LDAP por parte do Samba é feita com o auxílio de *scripts*. Esses *scripts* foram instalados pelo pacote smbldap-tools, e já os incluímos no `/etc/samba/smb.conf`, agora precisamos configurá-los.

   Os arquivos de configuração do smbldap-tools residem em `/etc/smbldap-tools`, mas eles não são instalados por padrão. O pacote vem apenas com exemplos de configuração no diretório `/usr/share/doc/smbldap-tools/examples`. Digite os seguintes comandos para fazer uma cópia dos arquivos necessários para o local correto:

```
usuario@sambaserver:~$ sudo zcat /usr/share/doc/smbldap-tools/exa
mples/smbldap.conf.gz > /etc/smbldap-tools/smbldap.conf
usuario@sambaserver:~$ sudo cp /usr/share/doc/smbldap-tools/examp
les/smbldap_bind.conf /etc/smbldap-tools
```

   Agora precisamos ajustá-los para as permissões adeqüadas:

```
usuario@sambaserver:~$ sudo chmod 644 /etc/smbldap-tools/smbldap.
conf
usuario@sambaserver:~$ sudo chmod 600 /etc/smbldap-tools/smbldap_
bind.conf
usuario@sambaserver:~$ sudo chown root:root /etc/smbldap-tools/sm
bldap.conf
usuario@sambaserver:~$ sudo chown root:root /etc/smbldap-tools/sm
bldap_bind.conf
```

O arquivo `/etc/smbldap-tools/smbldap_bind.conf` contém as informações de autenticação ao diretório, por isso suas permissões devem estar mais restritas. A seguir apresentamos um exemplo desse arquivo com os parâmetros ajustados para o acesso ao diretório:

Exemplo 3.7. Arquivo de configuração /etc/smbldap-tools/smbldap_bind.conf

```
############################
# Credential Configuration #
############################
# Notes: you can specify two differents configuration if you use a
# master ldap for writing access and a slave ldap server for reading
# access
# By default, we will use the same DN (so it will work for standard
# Samba release)
slaveDN="cn=admin,dc=ime,dc=usp,dc=br"
slavePw="secret"
masterDN="cn=admin,dc=ime,dc=usp,dc=br"
masterPw="secret"
```

O arquivo `/etc/smbldap-tools/smbldap.conf` contém as informações de configuração dos *scripts* que serão usados pelo Samba. O primeiro parâmetro que iremos configurar é o *Security Identifier* (SID) do domínio Samba. Esse número é uma *hash* utilizada pelos domínios Windows para identificar os recursos presentes na rede. Para obter esse número, digite o seguinte comando:

```
usuario@sambaserver:~$ sudo net getlocalsid
SID for domain SAMBASERVER is: S-1-5-21-2244078416-
1265281458-506834435
```

Copie essa *hash* para que possamos colocá-la no arquivo de configuração do smbldap-tools. Não se esqueça de que se o Samba tiver sido instalado em um servidor diferente do que está rodando o LDAP, precisaremos configurar o suporte a TLS para os *scripts* do smbldap-tools, e também vamos precisar de uma cópia do certificado do servidor LDAP em `/etc/ssl/certs`.

Um exemplo do arquivo `/etc/smbldap-tools/smbldap.conf` configurado com suporte a TLS é exibido a seguir:

Exemplo 3.8. Arquivo de configuração /etc/smbldap-tools/smbldap.conf

```
##############################################################################
#
# General Configuration
#
##############################################################################

# Put your own SID. To obtain this number do: "net getlocalsid".
# If not defined, parameter is taking from "net getlocalsid" return
SID="S-1-5-21-2244078416-1265281458-506834435"

# Domain name the Samba server is in charged.
# If not defined, parameter is taking from smb.conf configuration file
# Ex: sambaDomain="IDEALX-NT"
sambaDomain="LDAP.IME.USP.BR"

##############################################################################
#
# LDAP Configuration
#
##############################################################################

# Notes: to use to dual ldap servers backend for Samba, you must patch
# Samba with the dual-head patch from IDEALX. If not using this patch
# just use the same server for slaveLDAP and masterLDAP.
# Those two servers declarations can also be used when you have
# . one master LDAP server where all writing operations must be done
# . one slave LDAP server where all reading operations must be done
#   (typically a replication directory)

# Slave LDAP server
# Ex: slaveLDAP=127.0.0.1
# If not defined, parameter is set to "127.0.0.1"
slaveLDAP="ldapserver.ime.usp.br"

# Slave LDAP port
# If not defined, parameter is set to "389"
slavePort="389"

# Master LDAP server: needed for write operations
# Ex: masterLDAP=127.0.0.1
# If not defined, parameter is set to "127.0.0.1"
masterLDAP="ldapserver.ime.usp.br"

# Master LDAP port
# If not defined, parameter is set to "389"
masterPort="389"

# Use TLS for LDAP
# If set to 1, this option will use start_tls for connection
```

```
# (you should also used the port 389)
# If not defined, parameter is set to "1"
ldapTLS="1"

# How to verify the server's certificate (none, optional or require)
# see "man Net::LDAP" in start_tls section for more details
verify="require"

# CA certificate
# see "man Net::LDAP" in start_tls section for more details
cafile="/etc/ssl/certs/ssl-cert-ldapserver.pem"

# certificate to use to connect to the ldap server
# see "man Net::LDAP" in start_tls section for more details
# clientcert="/etc/opt/IDEALX/smbldap-tools/smbldap-tools.pem"

# key certificate to use to connect to the ldap server
# see "man Net::LDAP" in start_tls section for more details
# clientkey="/etc/opt/IDEALX/smbldap-tools/smbldap-tools.key"

# LDAP Suffix
# Ex: suffix=dc=IDEALX,dc=ORG
suffix="dc=ime,dc=usp,dc=br"

# Where are stored Users
# Ex: usersdn="ou=Users,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for
# usersdn
usersdn="ou=Users,${suffix}"

# Where are stored Computers
# Ex: computersdn="ou=Computers,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for
# computersdn
computersdn="ou=Computers,${suffix}"

# Where are stored Groups
# Ex: groupsdn="ou=Groups,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for
# groupsdn
groupsdn="ou=Groups,${suffix}"

# Where are stored Idmap entries (used if samba is a domain member
# server)
# Ex: groupsdn="ou=Idmap,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for
# idmapdn
idmapdn="ou=Idmap,${suffix}"

# Where to store next uidNumber and gidNumber available for new users
# and groups
# If not defined, entries are stored in sambaDomainName object.
# Ex: sambaUnixIdPooldn="sambaDomainName=${sambaDomain},${suffix}"
# Ex: sambaUnixIdPooldn="cn=NextFreeUnixId,${suffix}"
```

```
sambaUnixIdPooldn="sambaDomainName=LDAP.IME.USP.BR,${suffix}"

# Default scope Used
scope="sub"

# Unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA, CLEARTEXT)
hash_encrypt="SSHA"

# if hash_encrypt is set to CRYPT, you may set a salt format.
# default is "%s", but many systems will generate MD5 hashed
# passwords if you use "$1$%.8s". This parameter is optional!
crypt_salt_format="%s"

########################################################################
#
# Unix Accounts Configuration
#
########################################################################

# Login defs
# Default Login Shell
# Ex: userLoginShell="/bin/bash"
userLoginShell="/bin/bash"

# Home directory
# Ex: userHome="/home/%U"
userHome="/home/%U"

# Default mode used for user homeDirectory
userHomeDirectoryMode="700"

# Gecos
userGecos="System User"

# Default User (POSIX and Samba) GID
defaultUserGid="513"

# Default Computer (Samba) GID
defaultComputerGid="515"

# Skel dir
skeletonDir="/etc/skel"

# Default password validation time (time in days) Comment the next line
# if you don't want password to be enable for defaultMaxPasswordAge
# days (be careful to the sambaPwdMustChange attribute's value)
defaultMaxPasswordAge="45"

########################################################################
#
# SAMBA Configuration
#
########################################################################
```

```
# The UNC path to home drives location (%U username substitution)
# Just set it to a null string if you want to use the smb.conf 'logon
# home' directive and/or disable roaming profiles
# Ex: userSmbHome="\\PDC-SMB3\%U"
userSmbHome=""

# The UNC path to profiles locations (%U username substitution)
# Just set it to a null string if you want to use the smb.conf 'logon
# path' directive and/or disable roaming profiles
# Ex: userProfile="\\PDC-SMB3\profiles\%U"
userProfile=""

# The default Home Drive Letter mapping
# (will be automatically mapped at logon time if home directory exist)
# Ex: userHomeDrive="H:"
userHomeDrive="H:"

# The default user netlogon script name (%U username substitution)
# if not used, will be automatically username.cmd
# make sure script file is edited under dos
# Ex: userScript="startup.cmd" # make sure script file is edited under
# dos
userScript="logon.bat"

# Domain appended to the users "mail"-attribute
# when smbldap-useradd -M is used
# Ex: mailDomain="idealx.com"
mailDomain="ime.usp.br"

##############################################################################
#
# SMBLDAP-TOOLS Configuration (default are ok for a RedHat)
#
##############################################################################

# Allows not to use smbpasswd (if with_smbpasswd == 0 in smbldap_conf
# .pm) but prefer Crypt::SmbHash library
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"

# Allows not to use slappasswd (if with_slappasswd == 0 in smbldap_
# conf.pm) but prefer Crypt:: libraries
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"

# comment out the following line to get rid of the default banner
# no_banner="1"
```

Com os *scripts* devidamente configurados, podemos inserir as entradas necessárias ao funcionamento do domínio do Samba no diretório LDAP através do *script* **smbldap-populate**:

```
usuario@sambaserver:~$ sudo /usr/sbin/smbldap-populate
Populating LDAP directory for domain LDAP.IME.USP.BR (S-1-5-21-22440
```

```
78416-1265281458-506834435)
(using builtin directory structure)

entry dc=ime,dc=usp,dc=br already exist.
adding new entry: ou=Users,dc=ime,dc=usp,dc=br
adding new entry: ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: ou=Computers,dc=ime,dc=usp,dc=br
adding new entry: ou=Idmap,dc=ime,dc=usp,dc=br
adding new entry: uid=root,ou=Users,dc=ime,dc=usp,dc=br
adding new entry: uid=nobody,ou=Users,dc=ime,dc=usp,dc=br
adding new entry: cn=Domain Admins,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Domain Users,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Domain Guests,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Domain Computers,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Administrators,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Account Operators,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Print Operators,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Backup Operators,ou=Groups,dc=ime,dc=usp,dc=br
adding new entry: cn=Replicators,ou=Groups,dc=ime,dc=usp,dc=br
entry sambaDomainName=LDAP.IME.USP.BR,dc=ime,dc=usp,dc=br already
exist. Updating it...

Please provide a password for the domain root:
Changing UNIX and samba passwords for root
New password: secret
Retype new password: secret
```

Ele pedirá a senha do super-usuário do domínio Samba. Essa senha não precisa ser igual à senha do administrador LDAP. Essa senha é da conta *root* do domínio, que será usada para incluir as estações Windows ao mesmo.

Capítulo 4. Ferramentas de gerenciamento

Resumo

Esse capítulo apresenta algumas ferramentas de gerenciamento de linha de comando e ferramentas de gerenciamento gráficas.

Trabalhar com arquivos LDIF assim como com ferramentas de linha de comando é construtivo para o aprendizado e o entendimento das estruturas de dados do diretório. Porém não é muito conveniente na prática. Por isso existem várias ferramentas gráficas para isso.

O uso de ferramentas de linha de comando é interessante para testar se novos serviços estão funcionando corretamente, à medida em que eles são instalados e configurados. Além disso, existem certos casos em que é necessário utilizar uma ferramenta de linha de comando, pois uma ferramenta gráfica não atende a necessidade específica que se quer.

As ferramentas gráficas de gerenciamento são muito úteis para o uso no dia-a-dia, em que não é necessário resolver um problema muito específico. A vantagem é que ela fornece uma visualização das estruturas dos diretórios mais concreta do que um arquivo LDIF ou do que uma ferramenta de linha de comando.

Ferramentas de linha de comando

Slap Tools

Essa coleção de ferramentas vem junto com o OpenLDAP e fornece um mecanismo para importar e exportar dados diretamente da base de dados do servidor OpenLDAP (slapd). Essas ferramentas podem ser usadas para verificar se a base de dados está correta.

Atenção

Já que essas ferramentas mexem diretamente na base de dados, o servidor LDAP não pode estar rodando quando essas ferramentas forem ser usadas, pois a base de dados LDAP poderá ser corrompida.

slapadd Lê registros LDIF de um arquivo ou entrada padrão e escreve as novas entradas na base de dados slapd.

slapcat Lê entradas da base de dados slapd e as escreve em um arquivo ou saída padrão.

slaptest Verifica a sintaxe do arquivo `/etc/ldap/slapd.conf`.

slapindex Regenera os índices em uma base de dados slapd.

slappasswd Gera um *hash* de *password* apropriado para se usar no `/etc/ldap/slapd.conf`.

LDAP Tools

É um conjunto de ferramentas clientes LDAP do OpenLDAP para se comunicar com qualquer servidor LDAPv3. Essas ferramentas podem ser usadas para verificar se o servidor LDAP está funcionando corretamente.

Existe uma relação um-para-um entre essas ferramentas e as operações do protocolo LDAP.

ldapadd Adiciona entradas (no formato LDIF) em em diretório LDAP.

ldapmodify Alteras entradas LDAP existentes.

ldapcompare Verifica se uma entrada possui um dado valor de atributo.

ldapdelete Apaga entradas LDAP.

ldapmodrdn Renomeia (altera o RDN) uma entrada LDAP existente.

ldappasswd Altera o *password* de uma entrada.

ldapsearch Procura entradas LDAP.

smbldap-tools

Conjunto de ferramentas *scripts*, que vem junto com o Samba, para gerenciamento de contas Samba em um diretório LDAP.

Possui *scripts* para mostrar informações, adicionar, deletar e alterar grupos, alterar o *password*, mostrar informações, adicionar, deletar e alterar usuários e popular a base de dados LDAP.

Um comando útil é

```
usuario@sambaserver:~$ sudo smbldap-usermod -a login
```

que adiciona o `objectclass` sambaSAMAccount, permitindo a um usuário que antes era somente POSIX se tornar um usuário Samba.

MigrationTools

Conjunto de scripts Perl para migrar usuários, grupos, *aliases*, *hosts*, grupos de redes, redes, protocolos, RPC's e serviços de nomes existentes (arquivos, NIS, NetInfo) para servidores LDAP.

A seguir vamos citar apenas alguns *scripts*:

migrate_passwd.pl Migra usuários do arquivo `/etc/passwd`.

migrate_group.pl Migra grupos do arquivo `/etc/group`.

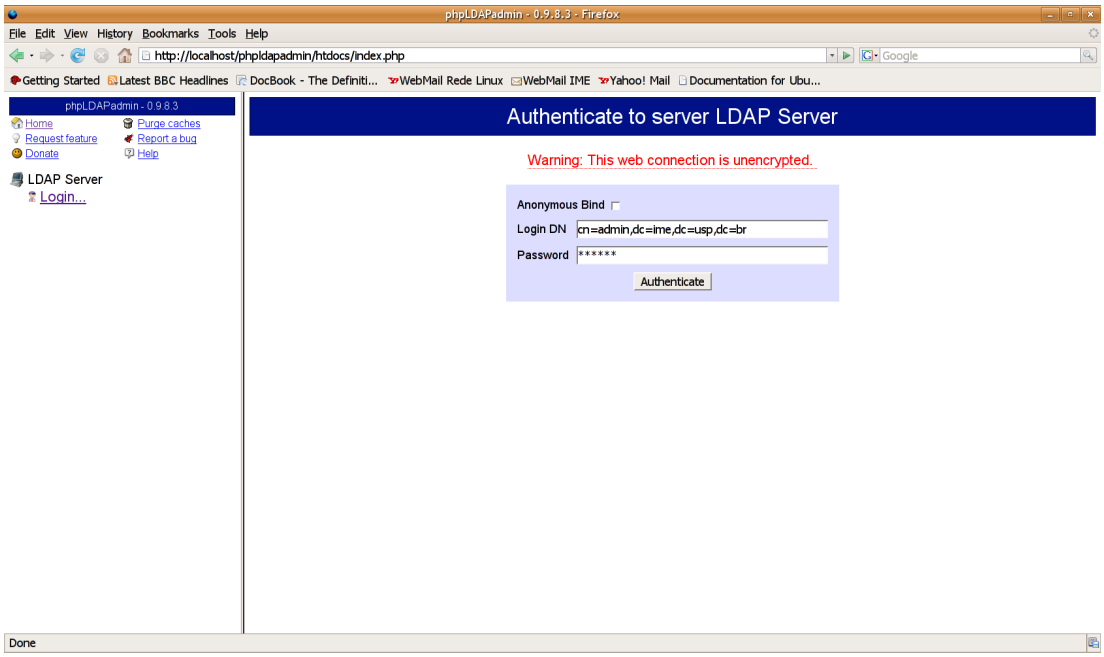
migrate_hosts.pl Migra *hosts* do arquivo `/etc/hosts`.

Ferramentas gráficas

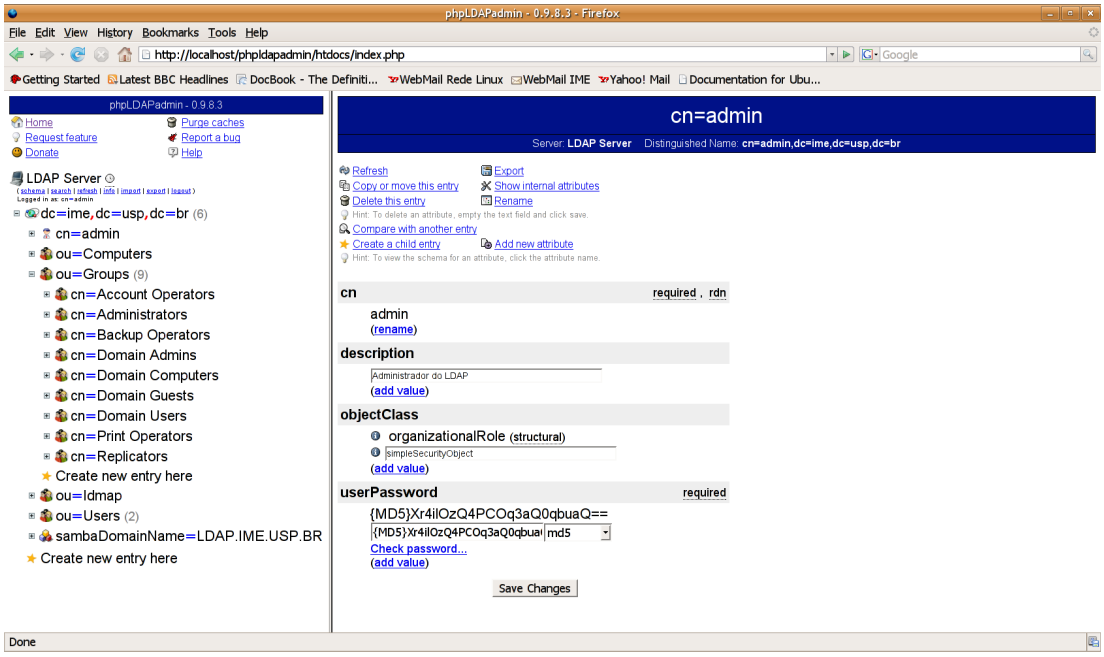
phpLDAPadmin

Ele é definido em seu site oficial [http://phpldapadmin.sourceforge.net/] como um "um navegador LDAP baseado na Web para gerenciar o seu servidor LDAP".

Essa ferramenta é um cliente LDAP implementado em PHP, que pode ser acessada por navegadores *Web*. A sua visualização da DIT e sua avançada funcionalidade de busca ajuda a tornar mais intuitiva a administração do diretório LDAP.



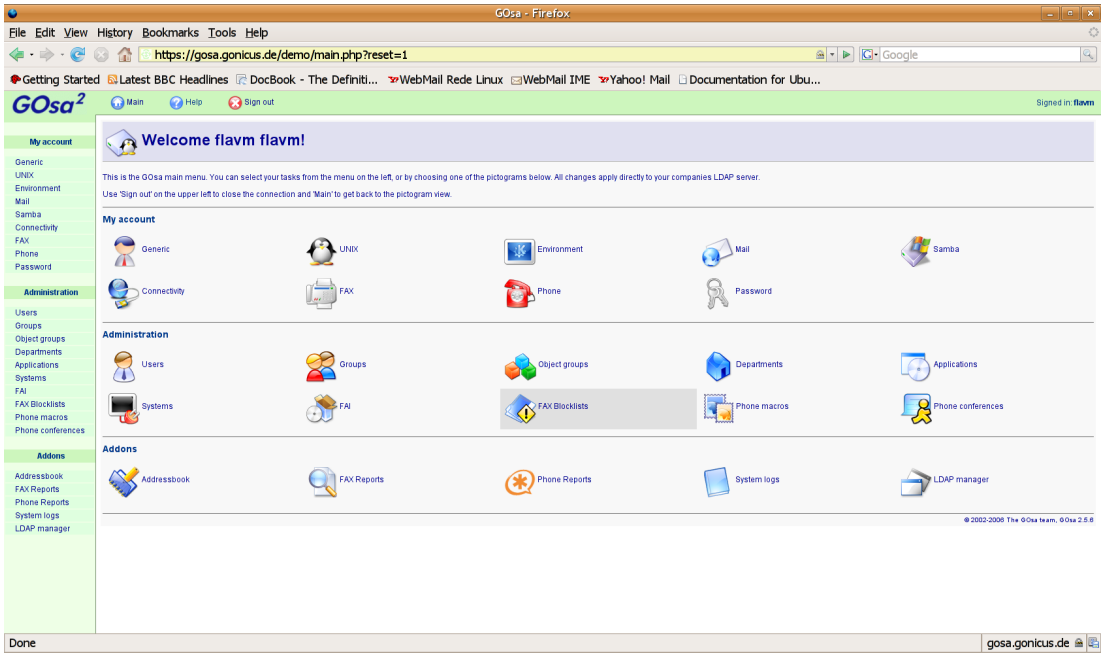
Tela de *login* do phpLDAPadmin



Tela de edição do usuário da interface do phpLDAPadmin

GOsa

O GOsa, assim como o phpLDAPadmin, é um cliente LDAP implementado em PHP que pode ser acessada por navegadores *Web*. A diferença principal deles é que o GOsa mostra os dados em um nível mais alto de abstração, deixando mais transparente a estrutura de árvore hierárquica do diretório.



Tela principal da interface do GOsa

LAT

Essa ferramenta é relativamente nova e significa LDAP *Administration Tool*. Ela permite navegar e modificar as entradas de diretórios LDAP. Ela é integrada com o GNOME, pois é escrita em C# usando Mono e Gtk#.

Capítulo 5. Ajuda

Resumo

Esse é um capítulo contendo informações que o ajudarão a resolver certos problemas e solucionar algumas dúvidas. Organizamos ele em duas seções: uma de perguntas freqüentes (também conhecida como FAQ e outra de solução de problemas (*Troubleshooting*).

Perguntas freqüentes (*Frequently Asked Questions*)

P: Existe outra maneira de realizar a integração com redes Windows?

R: Sim, existe. Uma alternativa é fazer com que o os clientes Linux/Unix busquem as informações de autenticação que estão armazenadas em um servidor Microsoft Active Directory®. Para maiores informações sobre como realizar essa implementação, consulte a documentação mantida pela comunidade Ubuntu: *Active Directory® How To* [https://help.ubuntu.com/community/ActiveDirectoryHowto].

P: Gostaria de manter o serviço NIS em minha rede, pois alguns terminais utilizam Solaris e não terei como atualizá-los a curto prazo. Existe uma maneira de integrar o NIS ao diretório LDAP?

R: O serviço de diretório LDAP tem como intuito na verdade substituir o NIS. No entanto, existe uma maneira de configurar um *gateway* NIS/LDAP utilizando uma solução comercial da PADL Software. Consulte o link NIS/LDAP Gateway [http://www.padl.com/Products/NISLDAPGateway.html] para maiores informações.

Solução de problemas (*Troubleshooting*)

1. Após alguns pequenos ajustes nos índices da base de dados, no arquivo `/etc/ldap/slapd.conf`, o NSS e o **finger** não conseguem mais encontrar os usuários que estão armazenados no diretório.

   Sempre que os índices forem alterados no arquivo `/etc/ldap/slapd.conf`, eles precisam ser reconstruídos. Para fazer isso, primeiro pare o serviço slapd:

   ```
   usuario@ldapserver:~$ sudo /etc/init.d/slapd stop
   Stopping OpenLDAP: slapd.
   ```

   depois reconstrua os índices utilizando o comando **slapindex**:

   ```
   usuario@ldapserver:~$ sudo slapindex
   ```

   e então reinicie o serviço:

   ```
   usuario@ldapserver:~$ sudo /etc/init.d/slapd start
   Starting OpenLDAP: running BDB recovery, slapd.
   ```

   Tente realizar novamente a busca.

2. Executar o **finger**, **id** ou o **getent** como *root* retorna os registros que estão armazenados no diretório. Porém, ao executar esses mesmos comandos como um usuário comum, não é possível obter esses resultados.

   Para que o NSS consiga realizar consultas no diretório LDAP ele precisa ter acesso ao arquivo `/etc/libnss-ldap.conf`. Certifique-se de que esse arquivo possui acesso de leitura para todos os usuários, caso contrário, apenas o usuário *root* conseguirá realizar essas buscas.

   O ajuste de permissão recomendado para o arquivo `/etc/libnss-ldap.conf` é 644, para permitir a pesquisa dos registros do diretório através do **finger**, **id** ou **getent** por todos os usuários do sistema.

   Não se esqueça de que esse ajuste só é recomendado caso esse arquivo não contenha nenhuma informação confidencial, como a senha do administrador do diretório (*rootdn*). Ou seja, o NSS terá que fazer as buscas utilizando *bind* anônimo (sem autenticação) e para isso o direito de leitura terá que ser liberado a todos os registros para todos os usuários, sem comprometer a segurança do sistema. Consulte o Apêndice A, *ACL's* para maiores informações.

Apêndice A. ACL's

As ACL's (*Access Control Lists*) fornecidas pelo OpenLDAP possuem uma sintaxe simples e são muito flexiveis e poderosas na sua implementação. A idéia básica é definir *Quem* têm qual *Nível de Acesso* a *O Quê*?

As principais formas de "Quem" são:

*	Qualquer usuário conectado, incluindo conexões anônimas.
self	O DN do usuário atualmente conectado.
anonymous	Conexões de usuários não autenticadas.
users	Conexões de usuários autenticadas.
`expressão regular`	Que case com um DN ou com uma identidade SASL.

A seguir está uma tabela que resume os vários níveis de acesso. Níveis maiores possuem todas as capacidades de níveis abaixo deles.

Tabela A.1. Níveis de acesso das ACL's

Nível de acesso	Permissão concedida
`write`	Acesso para atualizar valores de atributos
`read`	Acesso para ler resultados de buscas
`search`	Acesso para aplicar filtros de busca
`compare`	Acesso para comparar atributos
`auth`	Acesso para autenticar. Requer que o cliente mande o nome de usuário na forma de um DN e algum tipo de credencial para provar a sua identidade
`none`	Nenhum acesso

Finalmente, "O Quê" define as entradas ou atributos aos quais a ACL deve ser aplicada. Ele é composto de três partes, todas opcionais.

- Uma expressão regular definindo o DN. A sintaxe é dn.*`targetstyle`*=*`regex`*, onde *`targetstyle`* é *`base`*, *`one`* ou *`children`*, e *`regex`* é uma expressão regular representando um DN.
- Um filtro LDAP que obedeça o [RFC4515 *"Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters "* [ftp://ftp.rfc-editor.org/in-notes/rfc4515.txt]]. A sintaxe básica para especificar um filtro é filter=*`filtroLDAP`*.
- Uma lista de nomes atributos separados por vírgula, cuja forma é *`attrs`*=*`listaDeAtributos`*.
- `*` para incluir tudo.

A seguir está um exemplo de ACL.

Exemplo A.1. Uma ACL básica

```
access to attrs=userPassword,sambaNTPassword,sambaLMPassword
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by anonymous auth
       by self write
       by * none
access to *
       by dn.base="cn=admin,dc=ime,dc=usp,dc=br" write
       by * read
```

Uma ACL listada antes tem precedência sobre as ACL's listadas depois. Isso significa que as ACL's mais restritas devem ser listadas antes das mais gerais para que tenham efeito.

A primeira configuração listada no exemplo garante direito de acesso aos atributos `userPassword`, `sambaNTPassword` e `sambaLMPassword` para escrita ao `cn`=admin,`dc`=ime,`dc`=usp,`dc`=br, para autenticação aos usuários não autenticados (anônimos), de escrita aos usuários autenticados (apenas para as suas próprias entradas), ou seja, permite que os usuários alterem a própria senha, e nenhum acesso aos outros.

A segunda configuração garante direito de acesso à todo o diretório ao `cn`=admin,`dc`=ime,`dc`=usp,`dc`=br, e para leitura para todos os outros usuários.

Apêndice B. Arquivo LDIF

Introdução

LDIF (LDAP *Data Interchange Format* ou Formato de Intercâmbio de Dados do LDAP) é usado para representar entradas LDAP em um formato de texto simples. Sua especificação técnica está no [RFC2849 *"The LDAP Data Interchange Format (LDIF) - Technical Specification"* [ftp://ftp.rfc-editor.org/in-notes/rfc2849.txt]].

Esse formato de arquivo é apropriado para descrever informações de diretório ou modificações feitas nas informações de diretório. Ele é tipicamente usado para importar ou exportar informações de diretório entre servidores de diretório LDAP, ou para descrever um conjunto de modificações que é aplicado a um diretório.

Existem várias situações em que um formato de intercâmbio padrão é desejável. Por exemplo, alguém pode querer exportar uma cópia dos conteúdos de um servidor de diretório para um arquivo, mover o arquivo para uma outra máquina e importar os conteúdos em um segundo servidor de diretório.

Além disso, usando um formato bem definido, o desenvolvimento de ferramentas de importação de dados é facilitado.

O formato LDIF foi desenvolvido e usado originalmente na implementação LDAP da Universidade de Michigan. O primeiro uso do LDIF era para descrever entradas entradas de diretório. Depois o formato foi expandido para permitir representar também modificações nas entradas de diretório.

Definição do LDIF

Um arquivo LDIF é:

- Uma coleção de registros separados por linhas em branco.
- Um mapeamento de atributos a valores.
- Uma coleção de diretivas que dizem ao parser como processar a informação.

Um registro consiste de uma seqüência de linhas descrevendo uma entrada do diretório ou um conjunto de modificações em uma entrada do diretório. Ele especifica um conjunto de entradas do diretório ou um conjunto de mudanças a ser aplicado nas entradas do diretório, mas não ambos.

Existe uma relação um-para-um entre as operações LDAP que modificam o diretório (add, delete, modify e modrdn) e os tipos de registros. Essa correspondência é intencional, pois permite uma tradução direta do registro LDIF para as operações do protocolo. Mais informações sobre as operações no Apêndice D, *Operações do LDAP*.

A forma básica de um registro é:

```
# comentário
dn: distinguished name
attrdesc: attrvalue
attrdesc: attrvalue
```

A seguir está um exemplo de um arquivo LDIF.

Exemplo B.1. Arquivo LDIF

```
dn: dc=ime,dc=usp,dc=br
objectClass: domain
dc: ime

dn: cn=admin,dc=ime,dc=usp,dc=br
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: Administrador do LDAP
userPassword: {CRYPT}NHiC2OYs6eym2

dn: ou=Users,dc=ime,dc=usp,dc=br
objectClass: organizationalUnit
ou: Users
description: Usuários da organização

dn: ou=Groups,dc=ime,dc=usp,dc=br
objectClass: organizationalUnit
ou: Groups
description: Grupos do sistema
```

Cada registro consiste em um `DN` obrigatoriamente, e uma ou mais classes de objetos e múltiplas definições de atributos. O `objectClass` especifica a classe de objeto da entrada. A classe define quais atributos ou *schema*'s são permitidos ou obrigatórios para a entrada. Os dados em um arquivo LDIF devem obedecer as regras de *schema* do diretório LDAP.

Agora vamos analisar alguns registros:

```
dn: dc=ime,dc=usp,dc=br
objectClass: domain
dc: ime
```

Esse registro define o domínio, que é a raiz da árvore de diretório.

```
dn: cn=admin,dc=ime,dc=usp,dc=br
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: admin
description: Administrador do LDAP
userPassword: {CRYPT}NHiC2OYs6eym2
```

Esse registro define a entrada que será usada como administrador do diretório. A classe `organizationalRole` define entradas que representam cargos desempenhados pelas pessoas em uma organização. A classe `simpleSecurityObject` permite o uso de *passwords* por outras classes. `cn` é um atributo obrigatório da classe `organizationalRole`. `description` é um atributo da classe `organizationalRole`, que define a descrição do cargo. `userPassword` é um atributo obrigatório da classe `simpleSecurityObject`, que define o *password* da entrada.

`cn` (*Common Name*) é um atributo usado para definir o nome (RDN) de uma entrada no diretório. Ele é bem genérico, pois seu tipo não diz nada a respeito do que representa a entrada. O atributo `dc` (*Domain*

Component) é usado para definir uma parte do nome de um domínio. Ele também pode definir o RDN de uma entrada, no exemplo, seria `dc`=ime.

```
dn: ou=Users,dc=ime,dc=usp,dc=br
objectClass: organizationalUnit
ou: Users
description: Usuários da organização

dn: ou=Groups,dc=ime,dc=usp,dc=br
objectClass: organizationalUnit
ou: Groups
description: Grupos do sistema
```

Esses registros definem duas unidades organizacionais, `ou`=Users,`dc`=ime,`dc`=usp,`dc`=br, que contém os usuários, e `ou`=Groups,`dc`=ime,`dc`=usp,`dc`=br, que contém os grupos de usuários. `ou` (*Organizational Unit*) é um atributo da classe `organizationalUnit`. Ela permite definir uma entrada como uma unidade organizacional e também é usado para definir o RDN da entrada.

Outro exemplo:

Exemplo B.2. Outro arquivo LDIF

```
dn: uid=barbosa,ou=Users,dc=ime,dc=usp,dc=br
objectClass: account
objectClass: posixAccount
uid: barbosa
uidNumber: 2424
gidNumber: 20000
cn: Barbosa Pinto
userPassword: {CRYPT}KSC/iTKEfKm0A
loginShell: /bin/bash
homeDirectory: /home/barbosa/
gecos: Barbosa Pinto

dn: cn=SI,ou=Groups,dc=ime,dc=usp,dc=br
objectClass: posixGroup
cn: SI
gidNumber: 20000
description: Grupo dos usuários do SI
```

O primeiro registro, que é das classes `account` e `posixAccount`, representa um usuário Linux/Unix, e o segundo, que é da classe `posixGroup`, representa um grupo Linux/Unix. Na verdade essas classes servem para definir usuário e grupo de qualquer sistema operacional que siga o padrão POSIX.
`uid` (*User Identification*) é um atributo que é usado para definir o RDN da entrada que represente um usuário.

Esses exemplos de arquivos LDIF poderiam ser usados para importar informações de um diretório. A seguir está um exemplo que poderia ser usado para representar uma mudança a ser aplicada em uma entrada do diretório.

Exemplo B.3. Arquivo LDIF para remover usuário

```
dn: uid=barbosa,ou=Users,dc=ime,dc=usp,dc=br
changetype: delete
```

A palavra-chave `changetype` é usada para modificar registros existentes no diretório. `changetype` recebe o valor *delete*, que significa que o registro cujo `dn` é `uid`=barbosa,`ou`=Users,`dc`=ime,`dc`=usp,`dc`=br é para ser apagado.

Dica

Trabalhar com arquivos LDIF assim como com ferramentas de linha de comando é construtivo para o aprendizado e entendimento das estruturas de dados do diretório. Porém não é muito conveniente na prática. Por isso existem várias ferramentas gráficas para isso. Mais informações no Capítulo 4, *Ferramentas de gerenciamento*.

Apêndice C. Gerando um certificado SSL auto-assinado

Certificados SSL são necessários para que um determinado serviço opere com suporte a conexão segura por meio de criptografia.

Uma maneira de se obter o certificado, é através de uma *Autoridade Certificadora* (*Certificate Authority* ou CA). Outra maneira, é gerando ele por conta própria através de ferramentas adeqüadas.

Um certificado gerado através de ferramentas pode ser de dois tipos: auto-assinado (*self-signed*) ou assinado por uma CA. No caso de ser assinado por uma CA, você terá que gerar dois certificados: um para a sua própria CA, e o outro para o servidor, sendo que esse último será assinado pela CA que você criou. A maneira mais direta é gerar um certificado auto-assinado, pois nesse caso, não será necessário um certificado separado apenas para a CA.

Quando utilizamos conexão segura, os clientes precisam apenas dos certificados das CA's em quem eles confiam. Por isso que manter certificados auto-assinados pode tornar-se mais difícil. Para que o cliente se conecte a um serviço que usa certificado auto-assinado, precisará de uma cópia do próprio certificado do servidor já que não existe um certificado separado apenas para a CA que o emitiu. Ou seja, para cada serviço existirá um certificado que deverá ser instalado em cada cliente, enquanto que poderíamos instalar apenas um certificado em cada cliente: o da CA responsável pela emissão de todos os certificados da rede.

Repare que cada certificado aponta para uma CA, formando uma cadeia de confiança. A raiz dessa cadeia, que geralmente é uma CA de maior confiança, sempre vai possuir um certificado auto-assinado. Ou seja, certificados auto-assinados funcionam como se fossem a raiz da cadeia de confiança.

Vamos explicar abaixo o processo de criação de um certificado auto-assinado. Se você estiver pensando em disponibilizar suporte SSL/TLS a outros serviços de sua rede, talvez a melhor opção seja utilizar certificados assinado por uma CA. Nesse caso, consulte o site *Certificate Management and Generation with OpenSSL* [http://www.gagravarr.org/writing/openssl-certs/ca.shtml] para maiores informações sobre como criar uma CA para sua rede e assinar os seus próprios certificados, ou o site da CAcert [http://www.cacert.org/], que é uma organização que fornece certificados digitais assinados gratuitamente. Também existem várias empresas que vendem este serviço.

Para criar um certificado digital, é necessário que o pacote OpenSSL esteja instalado no servidor. Execute os seguintes comandos na distriuição Ubuntu para realizar essa instalação:

1. Atualize as listas dos repositórios

```
usuario@servidor:~$ sudo aptitude update
```

2. Instale o pacote OpenSSL

```
usuario@servidor:~$ sudo aptitude install openssl
```

Tendo o pacote OpenSSL instalado, execute o *script* **CA** com o parâmetro `newreq` para criar os arquivos do certificado auto-assinado e de sua respectiva chave privada:

```
usuario@servidor:~$ /usr/lib/ssl/misc/CA.sh -newreq
Generating a 1024 bit RSA private key
...........++++++
..++++++
writing new private key to 'newkey.pem'
Enter PEM pass phrase:secret
Verifying - Enter PEM pass phrase:secret
-----
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]:Sao Paulo
Locality Name (eg, city) []:Sao Paulo
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Universidade de
Sao Paulo
Organizational Unit Name (eg, section) []:Instituto de Matematica e Estatis
tica
Common Name (eg, YOUR name) []:servidor.ime.usp.br
Email Address []:.

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
Request is in newreq.pem, private key is in newkey.pem
```

Não é necessário preencher os três últimos campos, por isso colocamos um ponto ('.') em cada um deles. Você pode preenchê-los se achar conveniente. É importante que o campo *Common Name* seja preenchido com o endereço correto do servidor (nome e domínio).

Teremos como resultado da execução do *script* **CA** dois arquivos: `newreq.pem`, que contém o certificado auto-assinado do servidor; e `newkey.pem`, que contém a chave privada do certificado protegida por uma senha. Esses dois arquivos estão listados abaixo:

Exemplo C.1. Arquivo do certificado SSL auto-assinado (`newreq.pem`)

```
-----BEGIN CERTIFICATE REQUEST-----
MIIB6DCCAVECAQAwgacxCzAJBgNVBAYTAkJSMRIwEAYDVQQIEwlTYW8gUGF1bG8x
EjAQBgNVBAcTCVNhbyBQYXVsbzEiMCAGA1UEChMZVW5pdmVyc2lkYWRlIGRlIFNh
byBQYXVsbzEuMCwGA1UECxMlSW5zdGl0dXRvIGRlIE1hdGVtYXRpY2EgZSBFc3Rh
dGlzdGljYTEcMBoGA1UEAxMTc2Vydmlkb3IuaW1lLnVzcC5icjCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAzKogq6bdDdJeqwMeMf0tUV6k6D+x9cRNA4x3mWwJ
oFPmJAA39W8PoH8aEBNvbMuy4NSiM/Biy/1r6FmWiXQslsESnGTevfl4eZCDAsSK
XuoXWnP7k9AI7ViVZFP44dFnXjW7z58z4VtCJ8fmfhpiqhPdarHVYxf0CxabXdQU
hwUCAwEAAaAAMA0GCSqGSIb3DQEBBQUAA4GBALiDK8gb3omPp0Q+wk04QhfWavbQ
DTkOhCJuIsA3hDtGgCE+7JTHIBDLJPLdIQWs9qaVDjN9Uo3xRHZXQCM0pZ2dfhot
IGasL7xjfdnx8scacAQ/orv98xEe0jOQhZVuuel8Hh4n127S7tBlrggOwc+l25p5
1CAYNwmoTWIcAbmj
-----END CERTIFICATE REQUEST-----
```

Exemplo C.2. Arquivo da chave privada com senha (`newkey.pem`)

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,3F786C9DC54571D9

yQrotxI53BdVvY+q+uy/0G0TTBr1e9V3TE65J0qLSqgdnIQ8Z68c5iwUFA/FF0ay
j40YaESPSodFbvPt9fMzLaNhkoRj3QmCfNkuboiJpfHX+DOXmDakE17OWOjOoUR2
hIOK6aSysNZ34UJEhdvL33voAzFp3SXNqhgJY4L9ctEeR7fr5GqQ+ewCR4IYiuOI
nfqbDVTE7H2tsaHqzB+ND/Aqul+YDq3NQRH0zutX4k0TEEiCIimKFbz8LDbXLHzN
sdwdq8ODwjlVnFbOTXq7ZpdEdcWgFs+mdLu7Bp6maG9u9qXPxD/ToLOiH/B68lJQ
T7SdysxUrft0xo5LNMgakKCrPqXjfsGhT6IxHltFV6WEKiOnVLH3CaEYsm1qpyDR
G7Wtf+/gzwAMRVdYptO1MaL8paupxq4OXjBBWWvpymQ/qbnpVDi5CA9rPCTfv9OQ
sXgA9fo0tGGd5AD6mJZd94LxHG+54v3mWR+N6SvMppqtD8Ym2yVwDJi3zQk7uhwH
PEQ1I7/6TU3VHLaL6Zf90+CRKUrOe6+86jNj/p4Q2WRojfEabcyO4yqE7jPez1aB
05PIaXMh0LwBbqTNmWPEe+nStvvP9znc8fNGa9KMZ2SA/vVczGu9nwTBav3NZtYF
SRZ71P9pAxK1rfU5jGDNts61+56hGMMwIuqLg0l4Dr3M0w1Y+zFjT0JQqoQKxcJz
UEMLIlX+WO8uUol2PGchqBRnsk6wsp3mdab3FjHm79uiEpi2Oi79uPM6D2Acmxue
VGCACKw+hjttKn4VprOIXNA0UD9SL6EFuGAlsynJn2DRsJzLiNXTmg==
-----END RSA PRIVATE KEY-----
```

Se o certificado auto-assinado que foi criado for usado para proteger um serviço de rede, talvez seja necessário remover a senha de acesso à chave privada. Se a chave privada for mantida com a senha, toda vez que o serviço que a utilizar for inicializado, a senha terá que ser digitada pelo administrador. Para remover a senha de acesso à chave privada execute o seguinte comando:

```
usuario@servidor:~$ openssl rsa -in newkey.pem -out openkey.pem
Enter pass phrase for newkey.pem:secret
writing RSA key
```

Esse comando vai abrir o arquivo da chave privada, para isso ele pedirá a mesma senha que você digitou no início da geração do certificado. Ele criará então um novo arquivo, contendo a mesma chave privada, porém sem senha de acesso, que será guardado no `openkey.pem` ou outro arquivo especificado na linha de comando. A seguir temos um exemplo desse arquivo:

Exemplo C.3. Arquivo da chave privada sem senha (`openkey.pem`)

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDMqiCrpt0N0l6rAx4x/S1RXqToP7H1xE0DjHeZbAmgU+YkADf1
bw+gfxoQE29sy7Lg1KIz8GLL/WvoWZaJdCyWwRKcZN69+Xh5kIMCxIpe6hdac/uT
0AjtWJVkU/jh0WdeNbvPnzPhW0Inx+Z+GmKqE91qsdVjF/QLFptd1BSHBQIDAQAB
AoGBAJnzt5Ond0z4XAHj3IjZ24cD3KRflWw793qd5TFhVXRm6A1YpL9YhbbDJUXG
eOCr8aiyCjlrr8Wmoc8r5NMMsqRy4FIfJzVaAJozTM6r1JkdITHjW2DVN2zF3Tt6
PGfWygRwfMVqH5rNMpXV/3HwNqemoZJwRiWNhp9dP5vzzjhBAkEA9gWG9USF2mMx
hD0Jg0ZRm99JX6pwsJ+PXm+j1r1sILawNZcs4KFfcl2HAe9VWq/15eK9QJV8AmZ3
iD6QPXjYGQJBANT3L2yKxdxawEj5I7gf+iUm4h9Jxc2UlSEoFO27ZvegQ4Mcw4cd
+/MFvjbM+5KBnFu2RV3t0Gc8j/lCODOms80CQQDKUnFScy/BEaI/JhMUqlei4FXv
HrPeDNpSx2ztxImPva3b5J87fHqKCvBkXvbKxbpre7Q30LdTIgFbQlhMtXzBAkEA
t0mWk0BMUf35B9UEnO9IhrkUXAFOMET0pHiuqnxjfjN8Z1dWIO/5a4tBzkBVNB5A
x93zjYejmXZSIyCO5kVO8QJAJN54yvk62yolf0AKgDVB21A4uwGUFTyxZCdN/ZoO
m9W7YpQnowAe0t2RRm/J5IdDWSOAhdM5aXYDQTJVSN0qeg==
-----END RSA PRIVATE KEY-----
```

Repare a ausência do cabeçalho em relação ao anterior. Não se esqueça de restringir o acesso a esse arquivo agora, já que a chave do certificado não está mais protegida.

Apêndice D. Operações do LDAP

Essas são as principais operações do protocolo LDAP. Suas operações são um subconjunto das operações do protocolo X.500, do qual se originou.

Bind - autenticar
: Essa operação serve para autenticar o cliente no servidor. Ela envia o DN, o *password* do usuário e a versão do protocolo que está sendo usada. Por isso a conexão deve usar TLS ou algum outro mecanismo de segurança. O servidor checa o *password* olhando o atributo `userPassword` na entrada do usuário.

Unbind - encerrar sessão
: Encerra uma sessão LDAP.

Search - buscar
: O servidor busca e devolve as entradas do diretório que obedecem ao critério da busca.

Compare - comparar
: O servidor recebe um DN, um atributo e um valor, e responde se a entrada com aquele DN possui aquele atributo com aquele valor.

Add - adicionar
: Adiciona uma nova entrada no diretório.

Modify - modificar
: Modifica uma entrada existente. O servidor recebe o DN da entrada a ser modificada e as modificações a serem feitas.

Delete - apagar
: Apaga uma entrada existente. O servidor recebe o DN da entrada a ser apagada do diretório.

Modrdn - modificar RDN
: Renomeia uma entrada existente. O servidor recebe o DN original da entrada, o novo RDN, e se a entrada é movida para um local diferente na DIT, o DN do novo pai da entrada.

Abandon - abandonar
: O servidor recebe o ID da mensagem da operação a ser abandonada.

As operações de atualização são atômicas, ou seja, são transações. Elas são: Add, Delete, Modify e Modrdn. LDAP não tem suporte para múltiplas operações serem tratadas como uma única transação. Se essa característica for desejada, fica a cargo do servidor LDAP implementá-la.

Parte Subjetiva

A Epopéia

Quando estávamos no terceiro ano do Bacharelado em Ciência da Computação, trabalhávamos na SI (Seção de Informática) do IME-USP. Éramos encarregados da assistência técnica para todo tipo de problema que surgia nos equipamentos do Instituto. Nosso serviço ia desde instalar um sistema Linux em uma máquina e configurar alguns serviços, até trocar uma placa de rede queimada.

Isso não era nenhum trabalho acadêmico, mas foi muito construtivo, pois acreditamos que a pessoa tem que ter conhecimentos gerais em toda área da computação, incluindo até mesmo a parte técnica. Tudo ajuda para a formação do indivíduo.

Na época, quem ocupava o cargo de administrador da rede IME era o ex-BCC Alex Camargo. Um dia ele chegou com uma proposta para nós: estudar LDAP para uma futura implantação de um servidor no IME, visando integrar a autenticação dos clientes Windows e Linux/Unix atendidos pela nossa rede.

Na época, o nosso amigo Paulo Cheque trabalhava conosco no SI e fazia parte de nossa equipe no projeto, mas ele não está mais entre nós... Hoje ele é um fiel discípulo do professor Fabio Kon e faz o seu trabalho de formatura sob a sua orientação.

No fatídigo dia em que o Alex apresentou para nós a proposta, nunca havíamos ouvido falar em LDAP. Ele então explicou o que era e começamos a estudar sobre o assunto. LDAP não era tão divulgado na época como é atualmente. Ele estava numa época de trevas onde não existia muita documentação, algumas das que existiam eram desatualizadas e não seguiam nenhum padrão. Além disso, alguns programas que tinham que ser usados não existiam na forma de pacotes do Debian com suporte a TLS.

Na época, o Debian havia sido a distribuição Linux escolhida, pois era o padrão no Instituto e além disso nós já estávamos bem familiarizados com ela. Depois foi decidido que usaríamos a distribuição brasileira Kurumin, que tem compatibilidade com o Debian. Inclusive utiliza os mesmos repositórios de pacotes. Hoje, a que estamos usando é a distribuição Ubuntu, pois apesar de ser relativamente nova, já está bem madura hoje em dia. Além disso, ela tem pacotes mais novos e estáveis do que os da distribuição original Debian.

Onde está o Paul?

Depois que o Alex deixou o cargo de administração da rede IME para trabalhar com desenvolvimento em Java, tivemos outros administradores na rede (e inclusive nenhum por um tempo). Logo após o Alex, tivemos a breve presença do Paul na administração. O Paul ficou tão empolgado com o LDAP, que queria implantá-lo no dia seguinte na rede, e já migrar todos os usuários de uma só vez. É claro que isso não foi possível, pois a nossa configuração ainda não estava muito madura, ainda estávamos trabalhando para que a transição fosse feita minimizando a possibilidade de erros e da maneira mais transparente possível para os usuários.

Algo muito estranho que aconteceu é que um dia o Paul simplesmente parou de ir para o IME, sem antes avisar. Os outros funcionários do SI tentaram entrar em contato com ele, mas simplesmente não o encontraram. Até hoje o paradeiro dele é desconhecido. Se você viu esse homem ou tem qualquer informação que possa ajudar, ou sobre o paradeiro dele, entre em contado com o SI.

O sumiço de Paul deixou a todos no Instituto de "calças curtas", e como a abertura de um novo processo de seleção levaria mais algum tempo para ser efetuado, ficamos um tempo sem uma pessoa ocupando o cargo de administrador da Rede IME oficialmente. É claro que a rede não ficou largada à sua própria sorte nesse momento, muito pelo contrário. Tivemos a colaboração mais do que valiosa de nossos colegas Airton Vilela e Marcelo Modesto, sempre com o auxílio do prof. Arnaldo Mandel.

Mas a rede não está mais órfã: nosso administrador hoje em dia é o Marcelo Succi (SUUUUUUUUUUUUCCIIIIII!!!!).

Desafios e Frustrações

Para nós realmente foi um desafio começar a trabalhar com LDAP em uma época em que ele não estava tão popular e não existia tanta documentação a respeito dele. Os pacotes Debian não tinham suporte a TLS, então era preciso compilar alguns pacotes, em vez de simplesmente usar o gerenciador de pacotes APT. Hoje esses pacotes já vem com suporte TLS. Os pacotes do Ubuntu que usamos atualmente também possuem suporte a TLS. Isso é de vital importância para manter a segurança da rede.

Além disso, os desenvolvedores do OpenLDAP e do Samba adoram modificar os seus arquivos de configuração. Muitas vezes acontecia de nós apenas atualizarmos os pacotes instalados e o servidor parar de funcionar ou não funcionar corretamente. Esse é um problema que enfrentamos ainda hoje de vez em quando.

Os arquivos de exemplo de alguns pacotes que acompanham as distribuições (Debian, Ubuntu), os encontrados em documentações sobre o assunto e até mesmo nos sites oficiais são todos despadronizados. Às vezes exemplos encontrados nos sites oficiais são os mais bagunçados, não passando de um monte de linhas de configuração jogadas de qualquer jeito sem nenhum critério de organização.

O pior de tudo são as documentações que não explicam direito pontos importantes da configuração. Têm até mesmo configurações cujos parâmetros que vêm de padrão parecem verdadeiras "Pegadinhas do Mallandro". Se o usuário não for atento, e deixar a configuração padrão que vem com o programa, simplesmente ao reiniciar o computador, corre o risco de não conseguir *logar* mais.

A idéia inicial do projeto era implantarmos o serviço de diretório LDAP no IME inteiro, mas tivemos vários contra-tempos. Um deles era a mudança na administração da rede. Cada administrador tinha um enfoque diferente e idéias diferentes do que devia ser feito.

Então, no início desse semestre, recebemos uma proposta do professor Alexandre Roma, que é o responsável pelo laboratório de Matemática Aplicada. O problema que ele e os usuários do laboratório enfrentavam era um caso típico que requeria o uso de um serviço de diretório LDAP para solucionar os problemas de autenticação. Ele então deixou em nossas mãos o problema do laboratório, para que tentássemos resolvê-lo.

O plano havia, naquele momento, sido modificado. Nós iríamos fazer a implantação do LDAP no laboratório de Matemática Aplicada, o que serviria como uma implantação modelo para o Instituto. A implantação na Rede IME será feita pelos seus administradores, mas eles usarão a nossa instalação como base, além de darmos assessoria para eles.

Apesar da mudança de planos, gostamos bastante do trabalho realizado, pois a nossa implantação e a do IME farão parte de algo maior: o futuro serviço de diretório da USP.

Erich Soares Machado

Durante o estágio na Seção de Informática do IME com certeza não faltaram desafios. Além dos problemas que surgiam nos equipamentos, que muitas vezes só eram resolvidos com uma solução criativa, o tempo sempre foi um contra-peso em relação às atividades que tinham que ser desenvolvidas.

Simplesmente não havia tempo suficiente para tudo. Então o que se podia fazer era "escalonar" as atividades de acordo com a prioridade. Durante as provas, não dava para manter a lista de chamados técnicos em dia, em compensação

depois era necessário tirar o atraso. Em casos críticos, como na ocasião em que a rede do Instituto foi atacada por uma avalanche de vírus que exploravam uma falha de segurança dos sistemas Windows 2000 e XP, todo o tempo livre da semana era utilizado nesses atendimentos.

Após um tempo de estágio, os administradores da rede perceberam a nossa vontade em aprender mais do que o que seria possível apenas com os atendimentos. Na época, o administrador era o Alex Camargo, que é um ex-aluno do BCC. Ele sabia que nós teríamos que fazer algo de maior profundidade, para que ajudasse em nossa formação acadêmica. Ele estava, na época, procurando uma maneira de melhorar o gerenciamento da rede do Instituto.

Durante uma reunião com a administração da rede, o Alex resolveu passar para nós (eu, o Flavio e o Paulo), a tarefa de pesquisar a solução que ele estava procurando. Não tivemos muita informação a respeito do assunto, ele apenas disse que seria algo baseado em LDAP e que ele estava planejando implantar na rede do IME, para acabar com problemas de sincronismo entre as contas dos usuários dos ambientes Windows e Linux/Unix. Como material, recebemos um exemplar da revista *Linux Journal*, em que o assunto de capa era justamente o gerenciamento centralizado baseado em serviço de diretório LDAP.

No início não fazíamos a menor idéia do que era LDAP nem serviço de diretório. O conteúdo da revista que ele nos passou ajudou a entender o objetivo de forma mais clara, mas era uma matéria meramente ilustrativa. A maior parte do material que conseguíamos na época era resultado de pesquisas na Internet. Naquela época o material sobre LDAP era escasso, e o que encontrávamos não tinha uma qualidade muito boa. Ou era técnico demais, apenas com as especificações das RFC's e do protocolo, ou era superficial de forma que não resolvia nossas dúvidas sobre o assunto. Mesmo assim, reunimos informações suficientes para, depois de muito esforço, colocar o serviço pela primeira vez em funcionamento na nossa máquina de testes (um antigo servidor da rede, IBM NetFinity 5000).

Para nós, o sistema ainda estava funcionando como se fosse um "passe de mágica". Ainda não tínhamos idéia do que muitos parâmetros faziam. Somente após várias tentativas e instalações, reunimos informações suficientes sobre as principais configurações e começamos a escrever o nosso "manual de instalação".

Vimos que era necessário uma fonte de informação mais embasada para entender certos aspectos e conceitos relacionados ao serviço de diretório que estávamos implantando. Foi quando descobrimos o livro *LDAP System Administration* de Gerald Carter, publicado pela editora O'Reilly. A princípio iríamos comprar o livro com recursos próprios, mas após uma conversa com o pessoal do SI, vimos a possibilidade de utilizar uma verba que estava disponível para a nossa seção para realizar essa aquisição. Foi também mais ou menos nessa época que o Paulo decidiu deixar o projeto.

Mesmo depois que já tínhamos um conhecimento mais aprofundado, e demonstramos os testes que fizemos no laboratório com algumas poucas máquinas que nos foram disponibilizadas, enfrentamos problemas para implantar o serviço na rede do IME, já que esse era o objetivo do nosso projeto. Após a saída do Alex do cargo de administração, a prioridade do projeto de implantação passou a ter um caráter indefinido.

Em um dado momento, o administrador seguinte, o Paul, queria colocar todas as contas dos usuários do Instituto no nosso servidor de testes. Isso parecia perigoso demais, já que apesar dos testes que havíamos feito no laboratório, a dimensão da rede do IME era incomparável. De qualquer maneira, não tivemos tempo para realizar a loucura, pois o Paul desapareceu sem deixar vestígios...

Depois, a espera sempre tinha um motivo diferente: uma hora era pelo futuro administrador, já que ele teria que acompanhar a nossa instalação, outra hora por um novo servidor, já que as máquinas que tínhamos disponíveis não iriam suportar a carga da rede inteira a longo prazo.

Mesmo assim, as experiências que desenvolvemos não foram em vão. A documentação que criamos vai ajudar os administradores da Rede IME a implantar o serviço de forma definitiva em um futuro próximo, e também pudemos transformar o nosso ambiente de testes em uma implantação real dentro do próprio Instituto, através de uma oportunidade que surgiu no Departamento de Matemática Aplicada, graças ao Prof. Alexandre Roma.

De uma forma geral, gostei bastante do contato que tivemos com administração de redes, e espero continuar meus estudos nessa área também. Pretendo estudar para as provas de certificação, aproveitando o fato de que LDAP e

autenticação de redes são os assuntos da prova de nível mais avançado do LPI (*Linux Professional Institute*), para a certificação de administradores de rede Linux avançados. É claro que ainda terei que estudar muito para obter essa certificação, pois o conteúdo dos níveis I e II, que são pré-requisitos para o nível III, é muito mais abrangente do que esse assunto que estudamos. No entanto, o fato de ter surgido um exame de certificação sinaliza que esse conhecimento está sendo valorizado pelo mercado de trabalho atual.

Também desenvolvi interesses na área de desenvolvimento de sistemas, através de paradigmas orientados a objetos e métodos ágeis, e pretendo estudar estes tópicos no futuro.

Disciplinas cursadas no BCC mais relevantes para o trabalho

MAC0110 - Introdução à Computação	É a primeira matéria de programação do curso e os conceitos apresentados aqui são fundamentais para compreender a lógica do computador. Também tem uma participação significativa como incentivo para os alunos no início da graduação.
MAC0211 - Laboratório de Programação I	Aprendemos a trabalhar com as ferramentas de programação e de produtividade, como o *shell* do Linux. Também desenvolvemos o primeiro projeto em equipe.
MAC0242 - Laboratório de Programação II	Os primeiros passos em orientação a objetos (pelo menos para a nossa turma) e programação com *scripts*, o que é muito importante em administração de redes. Essa matéria foi uma das que mais ajudou a cultivar o conhecimento de programação que obtive na graduação.
MAC0323 - Estruturas de Dados	Fundamental para compreender as estruras de armazenamento e o seu processamento. Com certeza um pré-requisito para o desenvolvimento de nosso projeto de formatura.
MAC0422 - Sistemas Operacionais	O conhecimento adquirido nessa matéria ajudou a compreender como os sistemas evoluíram e funcionam nos dias atuais. Fundamental para entender o funcionamento dos serviços de rede.
MAC0426 - Sistemas de Bancos de Dados	Os modelos de dados, o seu armazenamento e recuperação estão intimamente relacionados com a proposta do nosso projeto, por isso essa disciplina teve importância chave durante o desenvolvimento do trabalho de formatura e da monografia.
MAC0438 - Programação Concorrente	Os modelos de concorrência têm presença constante atualmente, principalmente em sistemas de rede e distribuídos, já que existem vários usuários acessando a atualiando as mesmas informações simultaneamente. Uma das matérias mais interessantes do curso.
MAC0441 - Programação Orientada a Objetos	Orientação a objetos é um paradigma que ajuda a resolver muitos problemas da área da computação e, com criatividade, até de outras áreas. Essa matéria com certeza nos ajuda a sair com mais preparo para o mercado de trabalho e com a visão mais aberta a novos conceitos. É difícil acreditar que ela não é obrigatória para a graduação.
MAC0448 - Programação para Redes de Computadores	Essencial para quem quiser ter experiência com a programação em ambientes de rede. Essa matéria foi um ótimo exercício para os conceitos apresentados em PCS0210.

PCS0210 - Redes de Computadores	Apresenta os conceitos necessários a qualquer pessoa que queira atuar na área de redes de computadores. Seus conhecimentos são úteis não apenas na área de programação para redes, como na área de administração de redes.
MAC0433 - Administração de Sistemas Unix *	* Gostaria muito de ter feito essa matéria como optativa eletiva, mas infelizmente não consegui aproveitar o oferecimento dela em um semestre que tivesse horário compatível. Espero ter a oportunidade de freqüentá-la no futuro, mesmo que como ouvinte.

Flavio da Silva Mori Junior

Fazer esse trabalho de formatura foi uma verdadeira Epopéia, não só pelo tempo que levamos como pelas dificuldades que encontramos.

A proposta do Alex foi muito interessante para nós, pois além de ser uma área do nosso interesse, também poderíamos começar a desenvolver o nosso trabalho de formatura com antecedência. Essa antecedência foi boa, pois nos deu mais tempo para entrarmos em contato com os assuntos relacionados e pesquisarmos. Porém, isso também fez enfrentarmos mais desafios, como por exemplo, as documentações existentes no início do nosso trabalho, que não eram tão atualizadas e detalhadas como as existentes atualmente.

Além disso, alguns programas que usamos não tinham pacotes APT com suporte TLS, o que nos obrigava a compilá-los. Algo que achei legal foi que o período do desenvolvimento do nosso trabalho de formatura foi mais ou menos o mesmo do LDAP tornar-se popular.

O que me impressionou, foi ver como a maioria das equipes que desenvolvem *software* livre não se preocupa em fazer uma documentação decente. Além de não explicar pontos críticos, elas possuem exemplos despadronizados e sem nenhuma organização. Várias vezes, quando tínhamos implantado algum serviço, descobríamos que havia um outro modo melhor de fazê-lo.

No início, principalmente, foi uma verdadeira batalha o nosso trabalho no SI. Já que dividíamos o tempo de atender chamados com o de pesquisa do LDAP, muitas vezes ficávamos atolados em chamados. Computadores necessitando manutenção, instalação de programas e sempre um vírus novo ameaçando a segurança da Rede IME.

Depois que paramos de trabalhar no SI e continuamos somente com o projeto de formatura, tivemos mais tempo, mas mesmo assim nem tanto, afinal tínhamos as outras matérias para fazer.

Um dos nossos objetivos é que essa monografia sirva de base para pesquisa para pessoas que estão iniciando no LDAP.

Para escrevermos ela, utilizamos um formato chamado DocBook, que é o formato em que o TLDP (*The Linux Documentation Project*) aceita as documentações e em que os livros da O'Reilly são escritos. Portanto, também tivemos que aprender DocBook, mas com certeza valeu a pena, pois esse formato é muito bom para se escrever artigos, livros e documentações.

No início, fazíamos regularmente reuniões com o Arnaldo para explicarmos as características do LDAP e como ele poderia atender às necessidades da rede. Então o Arnaldo nos dizia os requisitos que deveriam ser atendidos e deixávamos combinado que iríamos pesquisar sobre algum assunto.

Além dessas reuniões com o Arnaldo, que é o supervisor da rede, também fazíamos reuniões o Alex, que na época era o administrador da rede.

Depois do Alex, tivemos mais dois administradores de rede: o Paul, que não sabemos o que aconteceu com ele após o seu surto de empolgação com o LDAP, e o Succi, o administrador atual.

Essas mudanças na administração também foram desafiadoras para nós. Tínhamos que chegar a um acordo entre as idéias do Arnaldo e a dos administradores, já que cada vez era um diferente. Passou-se até mesmo uma época entre o Paul e o Succi em que a rede ficou sem administrador.

O Marcelo e o Airton, que veio do CEC para o SI, também acompanharam o desenvolver do nosso trabalho.

De uma forma geral, o trabalho com o Erich e os outros membros do SI fluiu bem. Aprendemos com isso a ter jogo de cintura e a trabalhar em equipe.

Eu gosto da área de gerenciamento de redes, pois atualmente todo sistema grande de computação em qualquer empresa envolve uma rede de computadores. O serviço de diretório do LDAP vem de uma idéia muito boa de centralização de informação. Sabemos que no mundo de hoje, o armazenamento e gerenciamento de informação é algo vital, principalmente com a demanda crescente pelo seu acesso.

Eu provavelmente vou querer seguir a minha vida profissional na área de conhecimento relacionada com o trabalho de formatura. Além do LDAP propriamente dito, em áreas relacionadas a gerenciamento de rede, banco de dados e segurança.

Eu pretendo prestar as provas da LPI. Até mesmo porque a terceira prova cobra muito do conhecimento que usamos no nosso trabalho: autenticação, Samba e LDAP. Uma outra certificação muito boa para gerenciamento de redes, que eu viso fazer, é a da CISCO. Além disso, também quero outras certificações relacionadas a outras áreas, como a de Java.

Disciplinas do BCC Mais Relevantes

MAC0110 - Introdução à Computação	É através dessa matéria que temos o primeiro contato com a computação na faculdade. Com certeza essa matéria é pre-requisito para todas as outras, pois introduz conceitos importantes que usamos ao longo da faculdade.
MAC0211 - Laboratório de Programação I	Nessa matéria aprendemos ferramentas de linha de comando do Linux. Esse conhecimento é indispensável para a administração de uma rede de computadores.
MAC0242 - Laboratório de Programação II	Aprendemos programação com scripts. O uso de scripts são fundamentais para a implantação do serviço de diretório assim como de vários outros serviços de rede.
MAC0323 - Estruturas de Dados	A estrutura da DIT (*Directory Information Tree*) é uma estrutura de dados em que são representadas as informações do diretório. É uma estrutura de árvore hierárquica. Essa matéria ajudou a conhecermos a estrutura de dados de árvore e outras estruturas que são usadas no serviço de diretório LDAP.
MAC0332 - Engenharia de Software	Nessa matéria aprendemos como fazer a análise e especificação de requisitos e também testes, necessários para esse nosso projeto.
MAC0426 - Sistemas de Bancos de Dados	As informações do serviço de diretório são armazenados em uma base de dados que funciona como *backend*. Nessa matéria aprendemos os conceitos de bancos de dados necessários para o entendimento de como funciona o armazenamento das informações.
MAC0422 - Sistemas Operacionais	O objetivo do nosso trabalho é autenticação integrada de diferentes sistemas operacionais em uma rede de computadores. Com essa matéria, entendemos melhor a diferença entre o funcionamento dos diferentes sistemas operacionais.

MAC0448 - Programação para Redes de Computadores e PCS0210 - Redes de Computadores	Essas matérias ajudaram a entender melhor como funciona uma rede de computadores. Foram inúmeros tópicos fundamentais para desenvolvermos nosso trabalho de formatura, como TCP/IP, serviços de rede e segurança de dados.
MAC0441 - Programação Orientada a Objetos e MAC0413 - Tópicos de Programação Orientada a Objetos	Na implementação do servidor LDAP existem vários conceitos de programação orientada a objeto. Por exemplo: uma entrada do diretório pertence a uma ou mais classes, que determinam quais os atributos a entrada tem. Além disso, essas matérias ensinaram a gente como organizar um projeto.
FLC0474 - Língua Portuguesa	Essa matéria é importante, pois sem saber o uso correto da língua portuguesa, não seria possível escrever essa monografia!

Agradecimentos

Agradecemos primeiramente ao professor Arnaldo Mandel, o nosso orientador nesse trabalho de formatura. Nas reuniões que tínhamos com ele eram levantados requisitos importantes para serem analisados. Isso serviu como uma linha-guia para o nosso projeto.

Também agradecemos ao Alex Camargo, pois foi ele quem teve a iniciativa de implantar LDAP no IME e nos apresentou a proposta do trabalho. Foi através dele que tivemos o contato inicial com LDAP.

Também agradecemos aos funcionários do SI, que sempre estiveram conosco desde a época em que trabalhávamos como técnicos. Eles sempre estiveram acompanhando o desenvolver do nosso trabalho. O Marcelo Modesto até mesmo foi em um *workshop* sobre LDAP conosco e o Airton Vilela de Oliveira tem nos acompanhado muito ultimamente.

Ao professor Alexandre Megiorin Roma, que teve a confiança de deixar sob nossa administração o laboratório de Matemática Aplicada, o qual ele é o responsável.

Agradecemos a todos os amigos que conhecemos na faculdade e também aos bons professores que fazem parte do corpo docente do IME (Carlos Eduardo Ferreira, João Eduardo Ferreira, José Coelho de Pina Júnior, Siang Wun Song, entre outros).

As ilustrações presentes nesta monografia foram feitas por uma amiga nossa chamada Camila Torrano e por isso agradecemos a ela.

Também não podemos deixar de agradecer às nossas famílias e às nossas maravilhosas namoradas, Fabiana Vidoto e Karina Andrade, que sempre estiveram ao nosso lado, ajudando a liberar a tensão e agüentando a gente quando estávamos sob grande pressão na faculdade!!!

Glossário

A

ACL's - *Access Control Lists*	Definem quem tem qual nível de acesso a qual informação no diretório LDAP.
Active Directory®	Implementação do serviço de diretório da Microsoft, que possui suporte a LDAP.

B

backend	É uma base de dados. Esse termo está relacionado ao termo *front-end*, que é aplicação que acessa a base de dados. Ver Também Banco de dados.
Banco de dados	Conjunto de dados com uma estrutura definida para organizar informações. Normalmente é gerenciado por um SGBD. Ver Também SGBD - Sistema Gerenciador de Banco de Dados.

C

CA - *Certificate Authority*	CA's são entidades responsáveis por emitir certificados digitais para terceiros. Essas autoridades são típicas de esquemas de Infra-estrutura de Chaves Públicas (ICP). Ver Também ICP - Infra-estrutura de Chaves Públicas.
CUPS - *Common Unix Printing System*	Sistema que fornece uma camada de impressão portável para sistemas operacionais Unix.

D

daemon	Processo que roda em *background* e realiza uma função específica ou uma tarefa relacionada ao sistema.
DAP - *Directory Access Protocol*	Protocolo para acessar serviços de diretório X.500, que funciona sobre a pilha de protocolos OSI. Ver Também LDAP - *Lightweight Directory Access Protocol*.
Diretório	Repositório de informações sobre objetos, organizados segundo um critério que facilite a sua consulta. Ver Também Serviço de diretório.
DIT - *Directory Information Tree*	Estrutura de árvore hierárquica em que são organizadas as entradas do serviço de diretório LDAP.
DN - *Distinguished Name*	Atributo de uma entrada em um diretório LDAP usado para se referir a uma entrada sem ambigüidade. Ver Também Diretório.
DNS - *Domain Name System*	Um serviço de diretório distribuído que faz o mapeamento entre o os *hostnames* e endereços IP.

Ver Também Serviço de diretório.

I

ICP - Infra-estrutura de Chaves Públicas
: Uma Infra-Estrutura de Chaves Públicas é um órgão ou inciativa pública ou privada que tem como objetivo manter uma estrutura de emissão de chaves públicas, baseando-se no princípio da terceira parte confiável, oferecendo uma mediação de acreditação e confiança em transações entre partes que utilizam certificados digitais.

IP - *Internet Protocol*
: Protocolo que roteia os pacotes de dados de uma máquina a outra.

IPC - *Inter-Process Communication*
: Grupo de mecanismos que permite aos processos transferirem informação entre si.

L

LDAP - *Lightweight Directory Access Protocol*
: Protocolo leve para acessar serviços de diretório baseados nos padrões X.500, que funciona sobre a suite de protocolos TCP/IP. É considerado leve em comparação com o protocolo DAP, do qual se originou.
Ver Também DAP - *Directory Access Protocol*.

LDIF - LDAP *Interchange Format*
: Formato padrão de arquivo texto para armazenamento de informações de configuração e conteúdos de diretórios LDAP.
Ver Também LDAP - *Lightweight Directory Access Protocol*.

N

NFS - *Network File System*
: Sistema de arquivo projetado para compartilhar arquivos entre *hosts* Unix.

NSS - *Name Service Switch*
: *Framework* que fornece um serviço que permite aos administradores especificarem em quais arquivos ou serviços de diretório serão realizadas pesquisas de nomes, no ambiente Linux/Unix.

O

OpenLDAP
: Uma suíte de aplicativos LDAP *open-source*.
Ver Também LDAP - *Lightweight Directory Access Protocol*.

OSI - *Open Systems Interconnection*
: Modelo de referência desenvolvido pela ISO (*International Standards Organization*) para que os fabricantes pudessem criar protocolos a partir desse modelo.

OU - *Organizational Unit*
: Usado para representar uma unidade organizacional, por exemplo usuários, grupos, computadores, etc, em um diretório LDAP.
Ver Também Diretório.

P

PAM - *Pluggable Authentication Modules*
: *Framefork* que permite desenvolvedores e administradores personalizarem os serviços usados para autenticar usuários.

PDC - *Primary Domain Controller*	É um servidor Windows responsável por manipular todas as contas em um domínio. Ele é quem autentica os usuários.
POSIX - *Portable Operating System Interface X*	Ramo do IEEE (*Institute of Electrical and Electronics Engineers*) cujo objetivo é padronizar os comandos, chamadas de sistema e bibliotecas de interface.

S

SAM - *Security Account Manager*	Base de dados que armazena as informações de usuários do domínio Windows. Ver Também Samba.
Samba	Suíte de aplicativos *open-source* que fornece serviços de rede a clientes SMB/CIFS (incluindo várias versões do Microsoft Windows) em *hosts* Linux. Ver Também SMB - *Server Message Block*.
SASL	Mecanismo genérico de autenticação que pode ser integrado em uma variedade de protocolos. Ver Também TLS.
schema's	Arquivos que definem qual tipo de informação poderá ser armazenada no diretório.
Serviço de diretório	É um serviço de armazenamento de informações otimizado para busca e leitura.
SGBD - Sistema Gerenciador de Banco de Dados	Conjunto de programas responsáveis pelo gerenciamento de um banco de dados. O principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, manipulação e organização dos dados. Ver Também Banco de dados.
SID - *Security Identifier*	Identificador único atribuído a todos os elementos de um domínio Windows (usuário, grupo ou computador).
slapd	*daemon* da suíte de aplicativos OpenLDAP que implementa o servidor LDAP. Ver Também OpenLDAP.
slurpd	*daemon* da suíte de aplicativos OpenLDAP usado para fornecer um serviço replicado de diretório.
SMB - *Server Message Block*	Protocolo para o compartilhamento de arquivos, impressoras e portas seriais entre computadores. Funciona segundo os modelos cliente-servidor e requisição-resposta. Ver Também Samba.
SQL - *Structured Query Language*	Linguagem de consulta estruturada para bancos de dados relacionais. Ver Também Samba.
SSL - *Secure Sockets Layer*	Protocolo projetado para fornecer criptografia de dados e autenticação entre um cliente e um servidor sobre TCP/IP. É a base do protocolo TLS. Ver Também TLS.

T

TLS	Protocolo projetado para fornecer criptografia de dados e autenticação entre um cliente e um servidor sobre TCP/IP. Ver Também SSL - *Secure Sockets Layer*.

Transação	É uma operação *all-or-nothing*, ou seja, que só deve ser realizada totalmente, não podendo ser concluida parcialmente. Ver Também Samba.
Trava de escrita	É usada em sistemas, como sistemas gerenciadores de bancos de dados, para evitar que mais de uma pessoa edite o mesmo arquivo ao mesmo tempo. Ver Também SGBD - Sistema Gerenciador de Banco de Dados.

X

X.500	Um conjunto de padrões para serviços de diretório. Ver Também DAP - *Directory Access Protocol*.

Bibliografia

[LSA03] *LDAP System Administration [http://www.oreilly.com/catalog/ldapsa/index.html]*. Gerald Carter. O'Reilly Media, Inc.Março de 2003. Primeira Edição. ISBN 1-56592-491-6.

[OAG05] *OpenLDAP Software 2.3 Administrator's Guide [http://www.openldap.org/doc/admin23/]*. The OpenLDAP Project.9 de Agosto de 2005.

[ULD04] *Using LDAP for Directoty Integration*. Steven Tuttle, Kedar Godbole, e Grant McCarthy. IBM.Fevereiro de 2004. Segunda Edição. SG24-6163-01.

[PPG05] *Plug-in Programmer's Guide [http://www.redhat.com/docs/manuals/dir-server/plugin/7.1/titlepg.html]*. Red Hat Directory Server. Red Hat, Inc.26 de Maio de 2005. Versão 7.1.

[LAH02] *Linux Administration Handbook*. Evi Nemeth, Garth Snyder, e Trent R. Hein. Prentice Hall PTR.2002. Primeira Edição. ISBN 0-13-008466-2.

[FDL02] *Licença de Documentação Livre GNU [http://www.ic.unicamp.br/~norton/fdl.html]*. Norton T. Roman e João S. O. Bueno Calligaris. Free Software Foundation, Inc.Novembro de 2002. Versão 1.2.

[TLS03] *What is TLS/SSL? [http://technet2.microsoft.com/WindowsServer/en/library/ed5ae700-e05e-45ef-b536-45795dbb99a21033.mspx?mfr=true]*. Copyright © 2006 Microsoft Corporation. Microsoft TechNet.28 de Março de 2003.