Conteúdo

Lista de Figuras

Acrónimos

AAA Authentication, Authorization, Accounting

ACL Access Control Lists

API Application Program Interface

ASN.1 Abstract Syntax Notation One

BDB Berkeley´s Data Base

CCITT Consultative Committee for International Telegraphy and Telephony

DAP Directory Access Protocol

DIT Directory Information Tree

DN Distinguished Name

DSA Directory Service Agent

DUA Directory User Agent

IANA Internet Assigned Authority

IETF Internet Engineering Task Force

ISO International Organization Standardization

ITU International Telecommunications Union

LDAP Lightweight Directory Access Protocol

LDIF LDAP Data Interchange Format

OID Object Identifier

OSI Open Source Initiative

RDN Relative Distinguished Name

SASL Simple Authentication and Security Layer

SSL Secure Sockets Layer

TLS Transport Layer Security

Capítulo 1

LDAP

1.1 Enquadramento Histórico

Quando a International Organization Standardization (ISO) e o Consultative Committee for International Telegraphy and Telephony (CCITT) se juntaram no início da década de 80 para criar um serviço de mensagens (a série X.400), houve a necessidade de desenvolver um protocolo que organizasse *entradas* num serviço de nomes de forma hierárquica, capaz de suportar grandes quantidades de informação e com uma enorme capacidade de procura de informação. Esse serviço criado pelas duas instituições, foi apresentado em 1988, denominando-se X.500, juntamente com um conjunto de recomendações e das normas ISO 9594. O X.500 especificava que a comunicação entre o cliente e o servidor do Directório usava o Directory Access Protocol (DAP) que era executado sobre a pilha de protocolos do modelo Open Source Initiative (OSI). O facto de o X.500 ser muito complexo e de custo incompatível, levou os pesquisadores da Universidade de Michigan a criar um servidor Lightweight Directory Access Protocol (LDAP) standalone, o *slapd*, que actuava sobre o TCP/IP. Em 1993 o LDAP foi então apresentado como alternativa ao protocolo DAP para acesso a Directórios baseados no modelo X.500, sendo pela primeira vez implementado na própria universidade. Esse grupo de pesquisadores disponibilizou as fontes do *slapd* na Internet e criou listas de discussão para divulgar e aperfeiçoar o novo serviço, sendo a sua evolução acompanhada por pessoas do mundo inteiro. Com a divulgação do *slapd*, o LDAP deixou de ser uma mera alternativa ao DAP do X.500 e ganhou

estatuto de Serviço de Directório completo, passando a competir directamente com o X.500. Em Dezembro de 1997, o Internet Engineering Task Force (IETF) lançou a versão 3 do LDAP como proposta padrão Internet para Serviços de Directório. Actualmente varias empresas oferecem produtos LDAP, incluindo a Microsoft, Netscape e Novell. A OpenLDAP Foundation mantém e disponibiliza uma implementação Open Source do Serviço de Directório LDAP, baseada na Universidade de Michigan, que inclui os seguintes módulos: *slapd*, *slurpd*, bibliotecas que implementam o protocolo LDAP, utilitários, ferramentas e exemplos de clientes LDAP. A evolução do OpenLDAP prossegue, acompanhando a evolução dos padrões IETF [1].

1.2 Protocolos X.500 e DAP

1.2.1 X.500

O X.500 é um Serviço de Directório universal desenvolvido pela International Telecommunications Union (ITU), com o objectivo de definir a ligação entre Serviços de Directório locais para assim formar um directório global distribuído. Os serviços do X.500 foram implementados pelos seguintes agentes:

- Directory User Agent (DUA) - Aplicação Cliente através da qual os clientes (pessoas ou aplicações) efectuam as várias *entradas* no *Directório*.

- Directory Service Agent (DSA) - É a Aplicação Servidor que vai gerir a Directory Information Tree (DIT) e que disponibiliza os Serviços de Directório ao cliente.

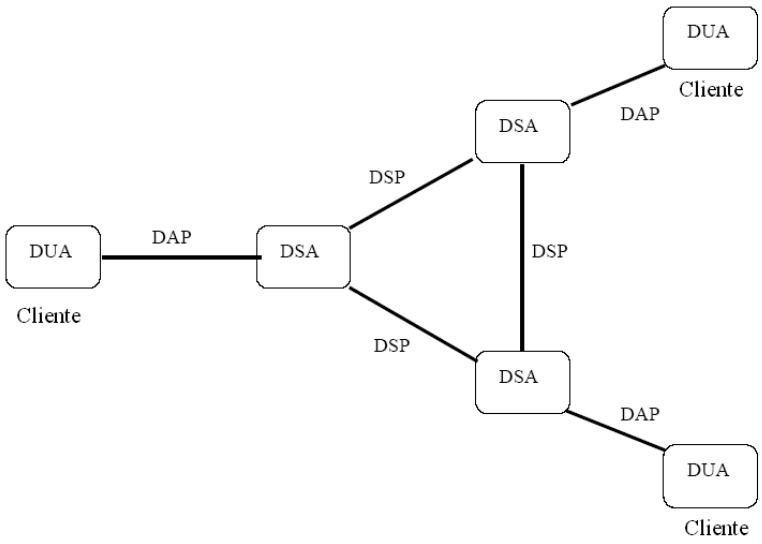

Figura 1.1: Serviço de Directório X.500 [1]

Estes dois agentes interagem através do DAP da camada de aplicação do modelo OSI, sendo os seus dados disponibilizados através de um servidor

DSA. O DSA local comunica com outros DSA espalhados pelo mundo, podendo assim um cliente, a partir de um servidor local, ter acesso a qualquer outro.
O X.500 contem funções para adicionar, modificar e apagar *entradas*. [2] Estas *entradas* estão organizadas num contexto de ordenação hierárquico, formando assim a DIT. A DIT actual está implementada segundo um critério geográfico. No topo da hierarquia estão normalmente representados países. Por baixo de cada país aparecem tipicamente organizações nacionais ou localidades. Por baixo de cada localidade estão representadas as organizações regionais. E por fim, por baixo de cada organização aparecem as *entradas* relativas aos departamentos que as constituem e debaixo destas, as pessoas que neles operam. As organizações internacionais são normalmente representadas imediatamente abaixo da "raiz".

A título de exemplo, existe uma infra-estrutura internacional baseada na tecnologia X.500, que contém actualmente cerca de um milhão e meio de *entradas* relativas a pessoas e 3000 relativas a organizações espalhadas por 35 países. Esta infra-estrutura foi elaborada num projecto da Comunidade Europeia, chamado PARADISE, concluído em Setembro de 1994.[3]

1.2.2 DAP

O Directory Access Protocol DAP é utilizado para definir o acesso a um Serviço de Directório (X.500) e estabelece o processo de pesquisa da DIT. No entanto, o facto de ser executado sobre a camada OSI exige uma quantidade significativa de recursos, tornando-o pesado e complexo, dificultando a sua execução em algumas máquinas clientes. Desta forma e a partir de 1993, os investigadores da Universidade do Michigan , trabalharam num protocolo com a maior parte das funcionalidades do DAP mas com muito menos complexidade para que corresse sobre TCP/IP e funcionasse numa máquina, independentemente do seu sistema operativo.[3]

1.3 LDAP vs X.500

O LDAP foi concebido efectuando-se as seguintes simplificações em relação ao X.500 [1]:

- Transporte - O LDAP é executado directamente sobre o TCP/IP, evitando o "overhead"das camadas superiores da pilha de protocolos OSI.
- Representação de dados - No LDAP a maioria dos elementos de dados são representados como cadeias de caracteres, processadas de modo mais fácil que os dados na representação estruturada Abstract Syntax Notation One (ASN.1) usada pelo X.500.
- Codificação de dados - O LDAP codifica dados para transporte em redes usando uma versão simplificada das mesmas regras de codificação usadas pelo X.500.
- Funcionalidade - o LDAP elimina características pouco usadas e também operações redundantes do X.500.

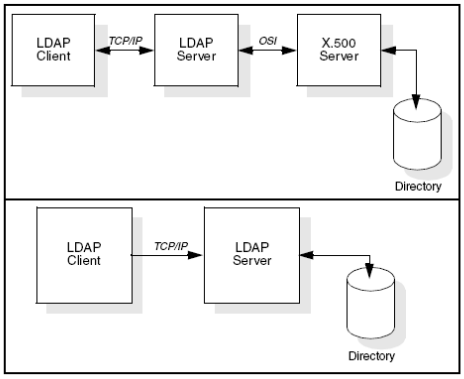

Figura 1.2: X.500 vs LDAP [4]

Como se pode observar na figura, o facto de evitar as camadas superiores dos protocolos OSI veio simplificar o modo de funcionamento do Serviço de Directório LDAP, tornando-o uma opção cada vez mais acessível a quem pretendia implementa-lo.

1.4 Noções Teóricas sobre o LDAP

Estas definições foram baseadas em: [1], [3], [5], [6], [7].

1.4.1 Directório

A palavra *Directório* que normalmente usamos para as pastas de um disco rígido, neste contexto, tem um significado diferente. Um *Directório* é uma base de dados especializada definida de forma hierárquica, optimizado para leitura suportando sofisticados métodos de pesquisa, com o objectivo de proporcionar uma resposta rápida a um enorme volume de consultas e onde são armazenadas informações estáticas de objectos. Não existe restrições quanto aos objectos que podem ser guardados num *Directório*. Esses objectos podem ser pessoas, organizações, endereços de email, impressoras, etc.

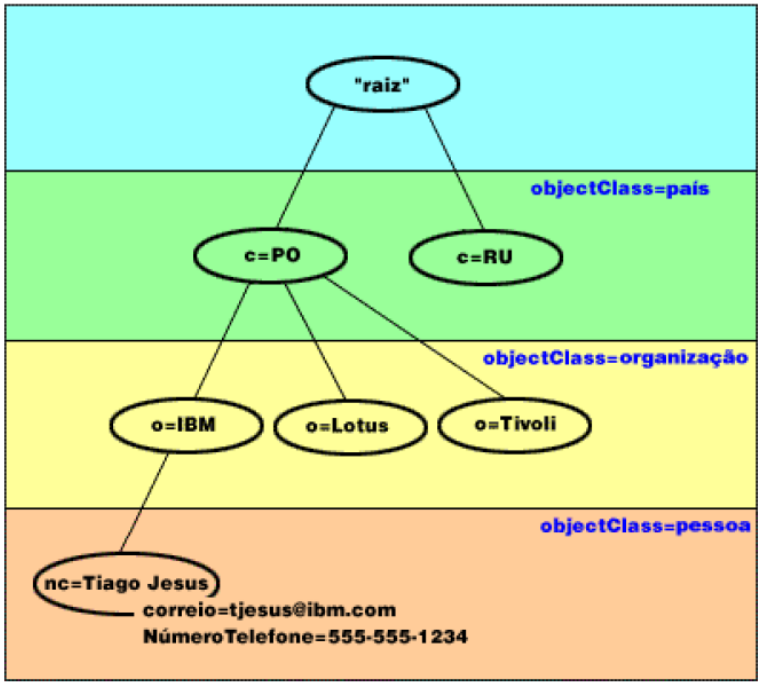

Figura 1.3: Exemplo da estrutura de um Directório [8]

1.4.2 Serviço de Directório

Um serviço de directório é uma aplicação projectada para gerir *entradas* e atributos num *Directório*, controlar o acesso aos clientes ou a qualquer outra aplicação que o requisite e possui mecanismos de pesquisa, remoção e actualização. Hoje em dia na maioria dos sistemas informáticos, existe a necessidade de centralizar informação e devido aos vários tipos de informações que possui, um Serviço de Directório deverá ter certas propriedades. Deverá ser:

- Flexível, na medida em que a informação pode ser de vários tipos;
- Seguro, possui mecanismos de autenticação tanto para o seu acesso interno como externo;
- Escalável e adaptável consoante a estrutura de implementação de rede de forma a corresponder às necessidades actuais mas também a necessidades futuras;
- Extensível, consoante as necessidades e alterações na rede

1.4.3 Schema

Os *schemas* em LDAP permitem manter a consistência dos dados do *Directório*. Uma importante característica é serem extensíveis e assim podemos adicionar mais atributos ou classes dependente das necessidades. Para usar um *schema* é necessário inclui-lo no ficheiro de configuração slapd.conf. Os *schemas* definem:

- quais as *object classes* que podem ser inseridas num *Directório*;
- quais os atributos de uma determinada *object class*;
- os valores possíveis para os atributos;

Se um objecto (*entrada*), não obedecer às regras do *schema* não pode ser inserido.

1.4.4 Distinguished Names

O *Distinguished Name (DN)* é usado para identificar uma *entrada* de forma não ambígua num Serviço de Directório. Os DN´s são compostos por uma sequência de Relative Distinguished Name (RDN)´s e cada RDN corresponde a um ramo na DIT, desde a raiz até a *entrada* a qual o DN faz referência. Um DN é formado por uma série de RDN´s separados por vírgulas. Por exemplo:

```
dn: uid=a9767,ou=exactas,dc=ubi,dc=pt
```

1.4.5 Atributos

Os atributos são identificados por um nome ou acrónimo, possuem um tipo e um ou mais valores. O tipo de atributo está associado a uma sintaxe. A sintaxe define que tipo de valor pode ser armazenado no atributo.

Alguns exemplos de atributos:

Abreviatura	Nome do atributo por extenso
dn	distinguishedName
cn	commonName
sn	surname
gn	givenName
o	organizationName
ou	organizationalUnitName
st	stateOrProvinceName
l	localityName
c	country
jpegPhoto	Fotografia em formato jpeg
telephoneNumber	Numero Telefone
postalCode	Código Posta
dc	domainComponent
uid	userID

1.4.6 Object Identifier

Cada *object class* ou tipo de atributo tem uma sintaxe que identifica o de tipo de objecto, isto é, um Object Identifier (OID) globalmente único. Os OID´s são representados como *strings* decimais separados por pontos representando uma árvore hierárquica. A Internet Assigned Authority (IANA) é a entidade responsável pelo registo de "sub-árvores"de OID´s. Exemplos:

OID	Utilização
1.1	Organizações OID
1.1.1	Elementos SNMP
1.1.2	Elementos LDAP
1.1.2.1	Tipos de Atributos
1.1.2.1.1	Meus Atributos
1.1.2.2	Object Classes
1.1.2.2.1	Minhas Object Classes

1.4.7 LDIF

LDAP Data Interchange Format (LDIF) é um ficheiro de texto usado para:

- importar dados para o *Directório*.
- alterar objectos existes;
- criar o *Backup* do *Directório*;
- a replicação;

Nota: Os dados devem obedecer ao *schema* do *Directório*.

1.4.8 Entrada

A unidade básica de informação armazenada num *Directório* é denominada por *entrada*. As *entradas* são compostas por um conjunto de atributos referentes a um objecto, sendo organizadas numa estrutura semelhante a uma árvore, isto é, organizada segundo uma DIT.

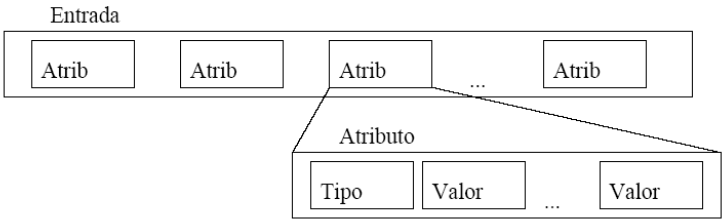

Figura 1.4: Entrada [1]

Um exemplo de uma *entrada* num ficheiro LDIF:

```
dn: cn=a9004,ou=saude,dc=ubi,dc=pt
objectClass: top objectClass:person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: a9004
sn: Abreu
userPassword: abreu
mail: a9004@hotmail.com
mobile: 969999995
street: Guimarães
disciplina: Anatomia I
disciplina: Biologia
disciplina: Bioquimica
```

Nota: Podem ser adicionadas várias *entradas* no mesmo ficheiro LDIF. O comando para adicionar uma ao mais *entradas* ao *Directório* é o *ldapadd* que será abordado com mais pormenor mais à frente na parte da implementação.

1.4.9 Object Class

Consiste num conjunto de atributos referentes a uma *entrada*. Quando uma *entrada* é definida, são atribuídas um ou mais *object classes*. Esses *object class* possuem atributos que podem ser opcionais ou obrigatórios. Existem dois tipos de *object classes*: *structural* e *auxiliary*. Toda a *entrada* deve ter um *object class* do tipo *structural* e pode ter uma ou mais *object class auxiliary*. Um exemplo retirado do core.schema da *object class* "*person*":

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

Como podemos ver no exemplo, é obrigatório o uso de sn (*surname*) ou cn (*common name*) e os atributos opcionais são: *userPassword; telephoneNumber; seeAlso;* e *description*.

1.5 Qual é a funcionalidade do slapd.conf?

O ficheiro slapd.conf é o ficheiro de configuração da daemon *slapd* do OpenLDAP. Normalmente está localizado em /etc/openldap/slapd.conf. Nele são especificados:[9]

- Quais são os *schemas* usados;
- Que *Backend* é usada: ldbm, bdb, etc;
- Qual a base do Serviço de Directório;
- Quem é o administrador e a sua *password*;
- A política de acesso;

1.6 Acesso aos dados do Directório

O acesso aos dados do *Directório* é controlado pelas Listas de Controlo de Acesso Access Control Lists (ACL)s. Estas definem:

- Quais as *entradas* e/ou atributos com acesso.
- Os clientes que podem ou não ter acesso.

Não existe um modelo padrão para o controlo de acesso no LDAP.

1.7 Serviços: slapd e slurpd

Slapd - Stand-alone LDAP Daemon

O *slapd* é um serviço LDAP autónomo (desenhado para correr como um servidor único), responsável por escutar ligações nas portas definidas, que podem ser uma ou mais (tipicamente é usada a porta 389). Toda a configuração do *slapd* é efectuada através do ficheiro slapd.conf.

Slurpd - Stand-alone LDAP Update Replication Daemon

O *slurpd* é também um serviço LDAP autónomo de actualização e replicação de dados entre as Bases de Dados dos vários servidores. Permite propagar as alterações de uma Base de Dados *slapd* para outra. Se o *slapd* estiver configurado para produzir um *log* de replicação com alterações, então o *slurpd* lê a partir desse *log* e envia as alterações para as outras instâncias *slapd*, via protocolo LDAP[10].

1.8 Características do LDAP

Algumas das características e potencialidades mais interessantes do *slapd* são: [11]

- Tem suporte para IPV6 - Desde a implementação LDAPv3, que o suporte LDAP sobre IPv6 tem vindo a ser uma realidade;
- Autenticação e Segurança - O *slapd* (serviço responsável pelo LDAP no servidor), sustenta serviços de forte autenticação através do uso do Simple Authentication and Security Layer (SASL). A implementação SASL do *slapd* utiliza o software *Cyrus SASL* o qual suporta um grande número de mecanismos: emphDIGEST-MD, EXTERNAL e GSSAPI;

  SASL - Permite ao cliente negociar um método de autenticação seguro;
- Segurança da Camada de Transporte - O *slapd* fornece protecção de privacidade, integridade e autenticação através do uso do *Transport Layer Security (TLS)* ou de *Secure Sockets Layer (SSL)*. A implementação TLS do *slapd* utiliza software OpenSSL;

  SSL - É um sistema de codificação para proporcionar a máxima confidencialidade dos dados pela Internet. Os dados são encriptados no ponto de envio e decifrados no ponto de destino.

  TLS - É semelhante ao SSL mas com uma tecnologia diferente.
- Escolha da Base de Dados de Backend - O *slapd* contém várias bases de dados *backend* disponíveis, que o permitem escolher a base de dados que mais se adapta à solução pretendida;
- Berkeley´s Data Base (BDB) - É uma base de dados *backend* de alta performance transaccional;
- Configuração - O *slapd* é altamente configurável através de um único ficheiro de configuração (slapd.conf), a partir do qual é possível efectuar as alterações pretendidas e adaptar o Serviço ao nosso sistema;

1.9 Bases Dados vs LDAP

A grande diferença entre o LDAP e as Bases de Dados Relacionais, é que no LDAP a informação está guardada segundo uma estrutura em árvore, raramente se efectuam actualizações, está optimizado para responder a um grande número de pesquisas e têm um alto nível de segurança.

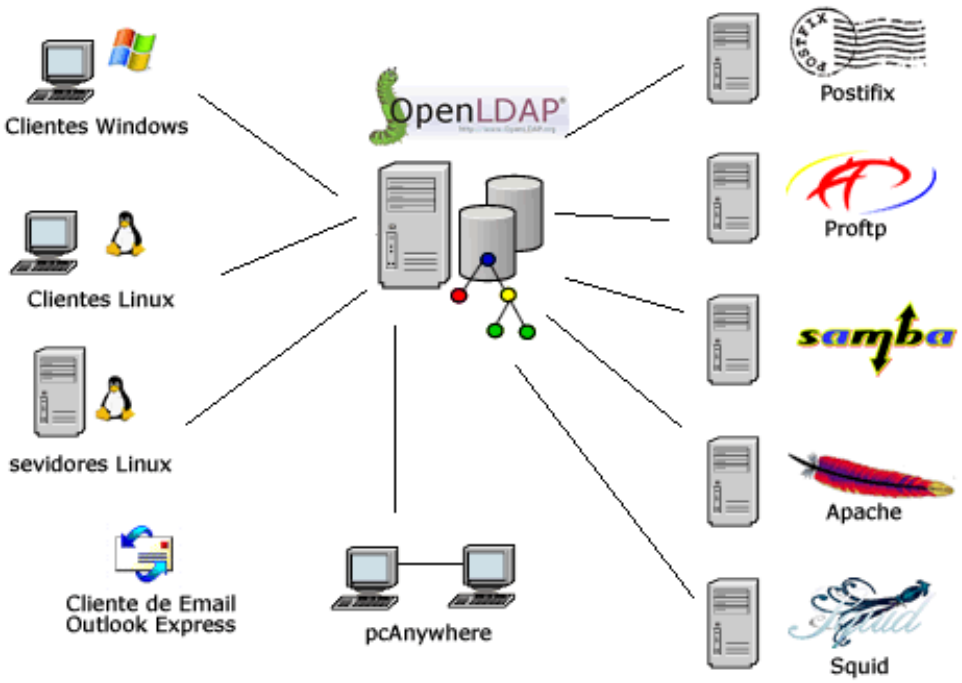

Figura 1.5: LDAP [12]

Actualmente, como podemos ver na figura, o LDAP é escolhido pela maioria dos administradores de rede em detrimento das Bases de Dados Tradicionais, porque para além das suas características já referidas, cada vez mais existem aplicações com suporte LDAP.

1.10 Modelos definidos pelo LDAP

O Serviço de Directório LDAP é composto pelos seguintes modelos: [1],[12],[3]

Modelo de Funcional

Define o que pode ser feito com a informação no *Directório* LDAP e como podemos altera-la e a forma de ter acesso. Funcionalmente as operações definidas pelo LDAP estão divididas em três categorias:

- Interrogação:

  - ldap_search - faz pesquisas das *entradas* no *Directório*;
  - ldap_compare - verifica se uma *entrada* contém um dado valor num atributo;

- Actualização:

  - ldap_modify - altera uma *entrada* existente;
  - ldap_add - adiciona uma nova *entrada*;
  - ldap_delete - apaga uma *entrada* existente;
  - ldap_modrdn - renomeia uma *entrada* existente;

- Autenticação:

  - ldap_bind - faz autenticação do cliente;
  - ldap_unbind - encerra uma sessão LDAP;

Modelo de Informação

Define o tipo de informação que pode ser armazenada num *Directório* LDAP. A unidade básica da informação armazenada no *Directório* é chamada de *entrada*. Esse modelo, herdado quase sem alterações do X.500, é extensível. Ao definir novas *object classes*, pode-se adicionar a um *Directório* qualquer tipo de informação.

Modelo de Nomes

Este modelo define a forma como a informação no *Directório* LDAP pode ser organizada e referenciada. As *entradas* são organizadas numa DIT e divididas segundo uma distribuição geográfica e/ou organizacional. Cada *entrada* tem um DN que especifica o caminho da raiz até à *entrada*.

Modelo de Segurança

Este importante modelo, define como os dados do *Directório* LDAP podem ser protegidos de acessos ou modificações não autorizadas. Para isso, existem três aspectos básicos na protecção da informação do *Directório*:

Acesso - Para o acesso seguro o LDAP suporta oTLS que criptografa toda a comunicação entre cliente e servidor. Desta forma garante a segurança das informações que são trocadas na rede.

Autenticação - é a forma de provar ao serviço que um cliente é válido. Para autenticação o LDAP suporta a SASL, que permite que o cliente e servidor negociem um método de autenticação (seguro).

Autorização - é o serviço que fornece ou nega direitos específicos ou funcionalidades ao cliente. A autorização é controlada pelas ACLs.

O LDAP irá controlar todos os três aspectos da Authentication, Authorization, Accounting (AAA) através de Listas de Controlo de Acesso, isto é, ACLs. As ACLs podem ser usadas para autorizar o acesso baseado em muitos factores diferentes. Elas podem ser usadas para forçar tipos específicos de autenticação. Uma vez que o cliente esteja autenticado como válido, as ACLs são usadas para autorizar o cliente. O cliente quando chama a operação "*bind*"fornece a sua identificação (*distinguished name*), credenciais de autenticação, *password*, chaves privadas, etc. Uma lista de controlo de acesso é usada para determinar que *entradas* do *Directório* o cliente pode ver e que alterações ele tem permissão para fazer. Há a possibilidade de um cliente não se identificar, ou seja, ter acesso ao *Directório* como anónimo. Nesse caso as regras de controlo de acesso também determinarão o que o cliente poderá ou não fazer.

1.11 Vantagens do LDAP

- É um standard aberto;
- Está optimizado para fazer pesquisas de informação;
- Centraliza toda a informação trazendo assim enormes benefícios, tais como: um único ponto de administração; menos dados duplicados;
- Tem um mecanismo de replicação incluído (*slurpd*);
- Tem mecanismos de segurança tanto para a autenticação (SASL) como para o troca de dados (SSL/TLS);
- Actualmente várias aplicações tem suporte para LDAP;

1.12 Desvantagens do LDAP

- O LDAP em alguns casos não substitui as Bases de Dados Relacionais;
- Raramente são efectuadas actualizações;
- Apenas convém ser guardados dados estáticos;
- Obviamente não é possível relacionar dois atributos, visto que não se trata de uma Base de Dados Relacional mas sim de uma base de dados estruturada hierarquicamente . Exemplo: Não é possível relacionar o código de uma disciplina com o nome da disciplina;
- Instalação torna-se difícil, pois cada vez tem mais pré-requisitos: OpenSSL, Kerberos, SASL (Cyrus), BerkeleyDB;

1.13 Conclusão

O LDAP tem vindo a ser usado cada vez mais por administradores de rede porque as suas características e as suas vantagens em muitos casos compensam as desvantagens. A prova disso é que cada vez mais aplicações e sistemas operativos possuem suporte para LDAP.

Capítulo 2

Implementação do LDAP

2.1 Introdução

Neste capítulo, serão descritos todos os passos para a implementação do LDAP. Será abordado o ficheiro de configuração slapd.conf, bem como alguns dos comandos mais importantes para adicionar, eliminar, alterar e pesquisar a informação alocada no servidor LDAP.

2.2 Implementação

2.2.1 Comandos do slapd

Para começar, é indispensável apresentar os seguintes comandos:

Iniciar o ldap	service ldap start
Reiniciar o ldap	service ldap restart
Parar o ldap	service ldap stop

Também se pode iniciar e parar o LDAP nos *services* do Linux.

2.2.2 Configuração do slapd.conf

Já foi vista a importância do ficheiro de configuração slap.conf e da sua funcionalidade. Esse ficheiro teve que ser alterado para estar em conformidade com as definições pretendidas. Sendo assim:

- Foi acrescentado um novo *schema*, o ubi.schema:

```
include /etc/openldap/schema/ubi.schema
```

- O domínio foi definido como sendo ubi.pt:

```
suffix "dc=ubi,dc=pt"
```

- Foi determinado o *login* do Administrador :

```
rootdn "cn=Manager,dc=ubi,dc=pt"
```

- Foram definidas as políticas de acesso:

```
access to attr=userPassword
by self write
by anonymous auth
by dn.base="cn=Manager,dc=ubi,dc=pt"write
by * none

access to *
by * read
```

Ficheiro slapd.conf na totalidade:

```
include /etc/openldap/schema/core.schema include
/etc/openldap/schema/cosine.schema include
/etc/openldap/schema/inetorgperson.schema include
/etc/openldap/schema/nis.schema include
/etc/openldap/schema/ubi.schema

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

database bdb

suffix "dc=ubi,dc=pt"
rootdn "cn=Manager,dc=ubi,dc=pt"
rootpw {crypt}ijFYNcSNctBYg

directory /var/lib/ldap

Indices to maintain for this database
index objectClass eq,pres
index ou,cn,mail,surname,givenname eq,pres,sub
index uidNumber,gidNumber,loginShell eq,pres
index uid,memberUid eq,pres,sub
index nisMapName,nisMapEntry eq,pres,sub

access to attr=userPassword
by self write
by anonymous auth
by dn.base="cn=Manager,dc=ubi,dc=pt"write
by * none

access to *
by * read
```

2.2.3 ACL´s

Como foram elaboradas as politicas de acesso?

Se o cliente autenticar com o atributo *userPassword*:

- Tem permissão para escrita nos atributos que o pertencem;
- se for anónimo precisa de autenticar (auth);
- O administrador tem permissão de escrita sobre todos os atributos;
- Todos as outras possíveis autenticações com o atributo *userPassword* não podem efectuar alteração nenhuma (none);

Para as outras possíveis autenticações :

- Apenas há permissão de leitura!

2.2.4 Ramificar a Árvore

Depois de configurado o ficheiro slapd.conf, seria altura de ramificar a árvore:

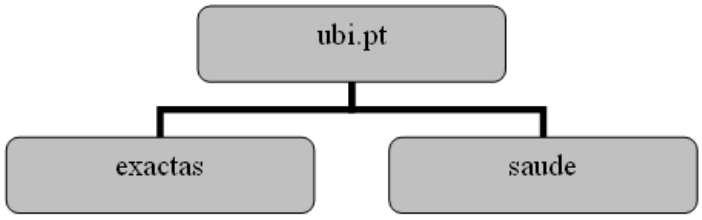

Figura 2.1: Árvore

Como se tratava de um teste apenas foram inseridos dois ramos.

Para isso, tive de criar um ficheiro ramos.ldif, que continha:

```
# Dominio
dn: dc=ubi,dc=pt dc: ubi
objectclass: top
objectclass: domain

# Ramos
dn: ou=exactas,dc=ubi,dc=pt
ou: exactas
objectClass: top
objectClass: organizationalUnit

dn: ou=saude,dc=ubi,dc=pt
ou: saude
objectClass: top
objectClass: organizationalUnit
```

Para o adicionar, usei o comando:

```
ldapadd -x -D "cn=Manager,dc=ubi,dc=pt-W -f /ramos.ldif
```

2.2.5 Adicionar novo schema

Agora, seria necessário juntar alunos a cada um dos ramos. Mas antes disso, e visto que, nos *schemas* pré-definidos pelo LDAP, não há nenhum atributo disciplina, decidi então criar um *schema* (ubi.schema), que contivesse o atributo disciplina e pertencia a *objectclass* cadeiras como se pode observar abaixo.

```
# ubi.schema

attributetype ( 1.1.2.1.1
NAME 'disciplina'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15128)

objectclass ( 1.1.2.2.1
NAME 'cadeiras'
SUP top AUXILIARY
MAY disciplina )
```

Nota: A escolha do atributo "disciplina"tem de ser bem ponderada, só fazendo sentido se não for constantemente alterado. O ubi.schema foi criado, observando e estudando a forma como outros atributos eram adicionados noutros schemas já predefinidos. Logicamente, depois tive que o colocar no ficheiro de configuração slapd.conf juntamente com os outros schemas:

```
include /etc/openldap/schema/ubi.schema
```

2.2.6 Ficheiros LDIF

Feito o novo schema, criei os ficheiros saúde.ldif e exactas.ldif, com alguns atributos. No exemplo, apenas é mostrado o primeiro e ultimo aluno tanto de saúde como de exactas. Podem ser inseridos infinitos alunos num único ficheiro LDIF e até se poderia juntar os alunos de exactas com os de saúde no mesmo ficheiro LDIF. Quanto ao atributo disciplina, a forma de introduzir várias disciplinas ao aluno é ir repetindo o nome do atributo seguido do valor, como se pode observar em baixo.

```
# saude.ldif

dn: uid=a9000,ou=saude,dc=ubi,dc=pt
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: cadeiras
uid:a9000
userPassword: abreu
cn: Jose Castro Abreu
sn: Abreu
mail: a9000@hotmail.com
mobile: 963199995
street:Braga
disciplina: Anatomia I
disciplina: Biologia
disciplina: Bioquimica

...

dn: uid=a9005,ou=saude,dc=ubi,dc=pt
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: cadeiras
uid:a9005
userPassword: pinto
cn: Mario Andre Pinto
sn: Pinto
mail: a9005@hotmail.com
mobile: 963199993
street: Coimbra
disciplina: Genetica
disciplina: Anatomia I
disciplina: Biologia
disciplina: Imunologia
```

```
# exactas.ldif

dn: uid=a9006,ou=exactas,dc=ubi,dc=pt
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: cadeiras
uid:a9006
userPassword: pinto
cn: Jose Pedro Pinto
sn: Pinto
mail: a9006@hotmail.com
mobile: 963196995
street: Braga
disciplina: Analise Matematica II
disciplina: Algebra
disciplina: ReDes

...

dn: uid=a9009,ou=exactas,dc=ubi,dc=pt
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: cadeiras
uid: a9009
userPassword: silva
cn: Joao Carlos Silva
sn: Silva
mail: a9009@hotmail.com
mobile: 963499995
street: Braga
disciplina: Analise Matematica II
disciplina: Algoritmos
disciplina: ReDes
disciplina: Lógica
```

A árvore neste momento, tem a seguinte forma:

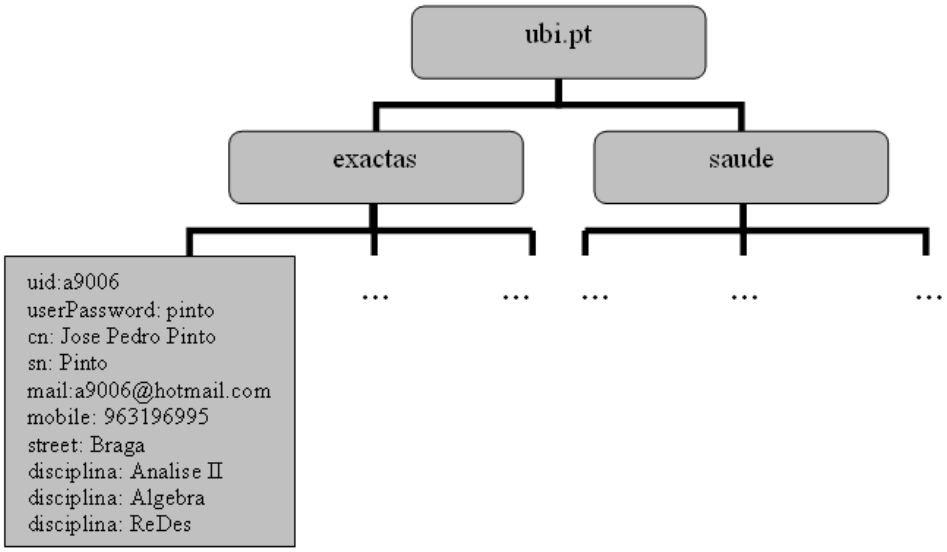

Figura 2.2: Árvore com entradas

Depois de ter a base de dados com alguns alunos fictícios inseridos, passei a testar os comandos: *ldapsearch, ldapadd, ldapdelete, ldapmodify*.

-x	Usa autenticação simples no lugar de SASL.
-D binddn	Usa o DN para se ligar ao LDAP
-W	É usada a *Prompt* para autenticação simples. É usado para não especificar a password na linha de comandos, omitindo assim a password.
-w passwd	passwd é a password para autenticação simples
-h ldaphost	Especifica qual o host no qual o servidor LDAP está a correr
-b searchbase	O uso de searchbase é para definir um ponto de partida especifico, senão seria usado o que esta por defeito.
-f file	São lidas as linhas do ficheiro file
-v	Retorna o diagnóstico da operação no output

2.2.7 ldapsearch

Mostra qual o domínio:

```
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

Exibe todas as *entradas* (DN mais atributos):

```
ldapsearch -x
```

Mostra todas as *entradas* (apenas os DN):

```
ldapsearch -x namingContexts
```

Nota: Acrescentando *namingContexs* a qualquer um dos comandos *ldapsearch*, apenas aparece o DN. Sem o *namingContexs* é devolvido o DN com todos os respectivos atributos e valores.

Todas as *entradas* de um domínio especifico serão mostradas:

```
ldapsearch -h 127.0.0.1 -x
ldapsearch -h localhost -x
ldapsearch -h ubi.pt -x
```

Mostra os dados de um determinando aluno:

```
ldapsearch -x "uid=a9002"
```

Mostra um atributo especifico (neste caso o *mail*) de um determinado aluno:

```
ldapsearch -x "uid=a9002"mail
```

Mostra todos os atributos do aluno que tem um determinado valor num dado atributo (neste caso foi usado o número de telemóvel):

```
ldapsearch -x mobile=969999995
```

Pesquisa por um atributo e mostra outro, isto é, segundo o exemplo abaixo, é mostrado todos os *mails* dos alunos de "Coimbra":

```
ldapsearch -x street=Coimbra mail
```

Mostra todos os alunos de apenas um ramo da árvore:

```
ldapsearch -x -b 'ou=saude,dc=ubi,dc=pt'
```

2.2.8 ldapadd

O comando para poder adicionar um ficheiro LDIF como administrador, é o seguinte:

```
ldapadd -x -D "cn=Manager,dc=ubi,dc=pt-W -f /ficheiro.ldif
```

2.2.9 ldapdelete

Para apagar um determinado aluno:

```
ldapdelete -v -x -W "uid=a9003,ou=exactas,dc=ubi,dc=pt"
-D "cn=Manager,dc=ubi,dc=pt"
```

Para apagar um ramo da árvore (neste caso, o ramo saúde):

```
ldapdelete -v -x -W "ou=saude,dc=ubi,dc=pt-D "cn=Manager,dc=ubi,dc=pt"
```

Podemos apagar vários alunos de uma só vez. Para isso é necessário criar um ficheiro LDIF com os DN dos alunos que queremos apagar.

O nome do ficheiro é apaga.ldif:

```
cn=membro1,ou=saude,dc=ubi,dc=pt
cn=membro2,ou=exactas,dc=ubi,dc=pt
cn=membro3,ou=saude,dc=ubi,dc=pt
```

Executando o comando seguinte, os alunos que estão no apaga.ldif serão apagados:

```
ldapdelete -x -D "cn=Manager,dc=ubi,dc=pt-W -f /apaga.ldif
```

2.2.10 ldapmodify

Desde que um atributo pertença a uma *objectclass*, e essa *objectclass* pertence a um determinado *schema* da nossa base de dados, podemos a qualquer altura adicionar um novo atributo a um aluno já inserido. Por exemplo, imaginemos que queremos adicionar o atributo *title* a um determinado aluno. É necessário construir um ficheiro mod.ldif (modifica) do seguinte modo:

```
dn: uid=a9004,ou=exactas,dc=ubi,dc=pt
changetype: modify
add: title
title: Teste
```

O comando para efectuar a modificação é:

```
ldapmodify -x -D "cn=Manager,dc=ubi,dc=pt-W -f /mod.ldif
```

Podemos também querer modificar um certo valor de um atributo (no exemplo foi modificado o valor do atributo *mail*). Então o ficheiro mod.ldif seria da seguinte forma:

```
dn: uid= 9003,ou=exactas,dc=ubi,dc=pt
changetype: modify
replace: mail
mail: teste@ubi.pt
```

Ao executar o mesmo comando que acima:

```
ldapmodify -x -D "cn=Manager,dc=ubi,dc=pt-W -f /mod.ldif
```

É possível, através do *ldapmodify* apagar um ou mais alunos. O ficheiro mod.ldif é então:

```
dn: cn=a9004,ou=exactas,dc=ubi,dc=pt
changetype: delete
```

Basta usar o mesmo comando que nos dois exemplos acima:

```
ldapmodify -x -D "cn=Manager,dc=ubi,dc=pt-W -f /mod.ldif
```

Também podemos apagar um atributo de um dado aluno. No exemplo, será mostrado como se pode apagar o atributo *title* de um determinado aluno.

Ficheiro modifica.ldif:

```
dn: cn=a9004,ou=exactas,dc=ubi,dc=pt
changetype: modify
delete: title
```

Usando o comando abaixo, o atributo title será apagado:

```
ldapmodify -x -D "cn=Manager,dc=ubi,dc=pt-W -f /mod.ldif
```

Uma pequena chamada de atenção, para o facto de nos exemplos acima apenas o administrador realizou estas modificações. Mas estas alterações, poderiam ser efectuadas também pelo próprio aluno, como aliais ficou definido nas ACLs. Considere-se o exemplo:

Com o seguinte ficheiro mod.ldif:

```
dn: uid=a9009,ou=exactas,dc=ubi,dc=pt
changetype: modify
replace: mail
mail: teste@ubi.pt
```

Tanto o administrador com o próprio aluno a9009 podem efectuar esta alteração:

```
ldapmodify -x -D "uid=a9009,ou=exactas,dc=ubi,dc=pt-W -f /mod.ldif
Enter LDAP Password:
modifying entry "uid=a9009,ou=exactas,dc=ubi,dc=pt"
```

```
ldapmodify -x -D "cn=Manager,dc=ubi,dc=pt-W -f /mod.ldif
Enter LDAP Password:
modifying entry "uid=a9009,ou=exactas,dc=ubi,dc=pt"
```

E agora, se por acaso um outro aluno por motivos desconhecidos, tentasse fazer essa alteração:

```
ldapmodify -x -D "uid=a9010,ou=exactas,dc=ubi,dc=pt-W -f /mod.ldif
Enter LDAP Password:
ldapbind: Invalid credentials (49)
```

Como esperado, essa operação não se poderia realizar!

2.3 Conclusão

Uma boa implementação do LDAP só se torna possível depois de perceber o funcionamento das *object class*, dos *schemas*, ACL´s e do ficheiro de configuração slapd.conf. É necessário uma análise bem atenta aos atributos que estamos a pensar introduzir, pois a finalidade do LDAP é guardar informação que raramente é alterada e que é constantemente pesquisada. Foram apresentados neste capítulo alguns comandos para adicionar, alterar, apagar, etc, mas para melhor interagir com o LDAP existem Application Program Interface (API)´s em linguagens como: C, Java, Pearl entre outras.

Bibliografia

[1] http://www.teses.usp.br/teses/disponiveis/45/45134/tde-28052003-100121/publico/gustavo.pdf
Documento Teórico sobre LDAP.

[2] http://penta2.ufrgs.br/rc952/trab2/x500.html
Documento Teórico sobre X500.

[3] http://www.deis.isec.pt/RSD/AULAS/0203/RD2/trabalhos.htm
Documento Teórico sobre X.500 e LDAP.

[4] http://borg.isc.ucsb.edu/aka/Ucdir/sg244986.pdf
Documento Teórico da IBM sobre LDAP.

[5] http://www.linuxchix.org.br/palestras/ldap.pdf
Documento Teórico sobre LDAP.

[6] http://geocities.yahoo.com.br/cesarakg/artigos.html
Página sobre LDAP.

[7] http://gsd.di.uminho.pt/teaching/5307Q3/2004/slides/LDAP
Documento sobre LDAP do professor Antonio Luís Sousa da Universidade do Minho.

[8] http://publib.boulder.ibm.com/html/as400/v5r1/ic2922/info/rzahy/rzahymst.pdf
Documento Teórico da IBM sobre LDAP.

[9] http://www.openldap.org/doc/admin21/slapdconfig.html
Página do site do openLDAP sobre o slapd.conf.

[10] http://paginas.fe.up.pt/ jruao/estagio/files/openLDAP.pdf
Documento sobre LDAP do professor Correia Araújo da Universidade do Porto.

[11] http://www.conectiva.com/doc/livros/online/10.0/servidor/ptBR/ch13s02.html
Página sobre LDAP.

[12] http://www.ldap.liceu.com.br/index.html
Site sobre LDAP.