

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

**INTEGRAÇÃO DO SERVIÇO DE DIRETÓRIO LDAP COM O SERVIÇO DE
AUTENTICAÇÃO DO SQUID**

**INTEGRAÇÃO DO SERVIÇO DE DIRETÓRIO LDAP COM O SERVIÇO DE
AUTENTICAÇÃO DO SQUID**

Monografia apresentada ao Curso de Especialização
em Redes de Computadores da Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte
como requisito parcial para a obtenção do título de
Especialista em Redes de Computadores.

Orientador: Aluizio Ferreira da Rocha Neto, (Msc.)

MARIA JOSÉ DA SILVA

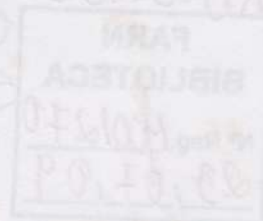
**NATAL/RN
2009**

LIGA DE V. MARIA JOSÉ DA SILVA DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

**INTEGRAÇÃO DO SERVIÇO DE DIRETÓRIO LDAP COM O SERVIÇO DE
AUTENTICAÇÃO DO SQUID**

Monografia apresentada ao curso de Especialização
em Redes de Computadores da Faculdade Natalense
para o Desenvolvimento do Rio Grande do Norte
como requisito parcial para a obtenção do título de
Especialista em Redes de Computadores.

Orientador: Aluizio Ferreira da Rocha Neto, (Msc.)



Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Silva, Maria José

Integração do serviço de diretório LDAP com o serviço de autenticação do SQUID / Maria José Silva. – Natal: FARN, 2009.
38f.

Orientador: Aluizio Ferreira de Rocha Neto (Msc.)

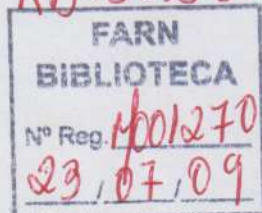
Monografia (Especialização em Redes de Computadores) – Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Segurança em Redes – Monografia. 2. Redes sem Fio – Wireless – Monografia. 3. Autenticação – Monografia. I. Título.

RN/FARN/BC

CDU 004.72

RB:32.533



004.72
5581 x
2009
Ex: 01

FARN

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RN
ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

**INTEGRAÇÃO DO SERVIÇO DE DIRETÓRIO LDAP COM O SERVIÇO DE
AUTENTICAÇÃO DO SQUID**

Maria José da Silva

Monografia apresentada e aprovada em ____ de ____ de ____, pela banca examinadora
composta pelos seguintes membros

Aluizio Ferreira da Rocha Neto Msc.
Orientador

Gleydson de Azevedo Ferreira Lima, Msc.
Membro examinador

Membro examinador

Natal/RN, ____ de ____ de ____.

AGRADECIMENTOS

A Deus, por estar sempre ao meu lado em minha vida, e tornar tudo possível.

Ao mestre Masahiro Taniguchi pelos seus nobres ensinamentos, dentre eles destaca-se "a honra e a capacidade humana".

Aos meus pais, Raimundo Abílio e Francisca Barbosa, que me educaram ensinando, incentivando-me na escalada do conhecimento e na obtenção de mais um título da Formação universitária.

Ao Prof. Aluísio Ferreira da Rocha Neto, pela orientação e incentivo que tornaram possível a conclusão desta monografia.

Um especial ao funcionário da FARN Lázaro Para, por todo o apoio e pela contribuição direta na construção deste trabalho.

A amiga Wilma Medeiros, pela credibilidade e pelo incentivo na realização deste projeto.

A amiga Helderline Câmara, pelo incentivo, pela contribuição com seus conhecimentos e apoio que tornaram possível a realização deste projeto.

A meus pais, familiares, amigos e todos aqueles que direta ou indiretamente, acreditaram e me incentivaram a correr atrás dos meus ideais.

AGRADECIMENTOS

A Deus, por estar sempre presente em minha vida, e tornar tudo possível.

Ao mestre Masaharu Taniguchi pelos seus sublimes ensinamentos, dentre eles destaca-se “o homem tem capacidade infinita”.

Aos meus pais, Raimundo Abílio e Francisca Barbosa, que me educaram enfatizando, incentivando-me na escalada do conhecimento e na obtenção de mais um título na formação universitária.

Ao Prof. Aluizio Ferreira da Rocha Neto, na orientação e incentivo que tornaram possível a conclusão desta monografia.

Em especial ao funcionário da FARN Lineu Paiva, por todo o suporte e pela contribuição direta na construção deste trabalho.

A amiga Wilma Medeiros, pela credibilidade e pelo incentivo na realização deste desejo.

A amiga Hilderline Câmara, pelo incentivo, pela contribuição com seus conhecimentos e apoio que tornaram possível a realização deste projeto.

RESUMO

Dado os problemas relacionados à segurança em redes sem fio - *wireless* abertas, sem chaves de segurança ou sistemas de autenticação, esta monografia apresenta o uso do protocolo LDAP (*Lightweight Directory Access Protocol*) e de outras ferramentas livres para resolver tais problemas, bem como demonstrar os passos utilizados na configuração de um servidor Proxy com base de dados LDAP e suas várias formas de interagir com outros serviços. É sugerido o LDAP devido a sua otimização para pesquisas de informação e possuir mecanismo de replicação, aumentando assim a sua confiabilidade. Além disso, o seu código é aberto, o que não implica em qualquer custo para a empresa. Para este estudo foi necessário seguir alguns passos metodológicos como: pesquisa e revisão bibliográfica, teste de laboratório em uma rede simulada e levantamentos de dados. Com a autenticação dos usuários é possível identificá-los, definir quais são os recursos que estão disponíveis a eles e inibir acessos indevidos e não autorizados, tornando assim o ambiente mais seguro e conseqüentemente com melhor desempenho.

Palavras-chave: Redes de Computadores. Autenticação. Redes sem fio.

LISTAS DE ABSTRACTAS, SIGLAS

ACLs Access Control Lists

Given the problems related to security in open wireless networks, without keys or security authentication systems, this paper presents the use the LDAP protocol (Lightweight Directory Access Protocol) and other free tools to resolve such problems and demonstrate the steps used in configure a proxy server with the database LDAP and several ways to interact with other services. LDAP is suggested due to optimization for search of information and it has the replication mechanism, increasing so their reliability. Also, it is open source, which does not involve any cost to the company. For this study it was necessary to follow some methodological steps such as: research and review literature, test laboratory in a simulated network and survey data. With the user authentication you can identify them, determine what resources are available to them and inhibit undue access, thus making the environment safer and consequently with better performance.

DNS Domain Name System

EAP Extensible Authentication Protocol

Key -words: Computer networks. Authentication. Wireless network.

HTTP HyperText Transfer Protocol

IEEE Institute of Electrical and Electronic Engineers

IETF Internet Engineering Task Force

ISO International Organization Standardization

ISQUE ISO Development Environment

LDAP Lightweight Directory Access Protocol

LDIF LDAP Data Interchange Format

LTS Longo Tempo de Suporte

MAC Media Access Control

MSNT Microsoft New Technology

NCSA National Center for Supercomputing Applications

NTLM New Technology Lan Manager

OSI Open Systems Interconnection

OSLDAP Standalone LDAP

PAM Pluggable Authentication Modules

RFC Request For Comments

SASL Simple Authentication and Security Layer

SMB Server Message Block

SSID Service Set Identifier

TCP/IP Transmission Control Protocol/ Internet Protocol

TLS Transport Layer Security

WEP Wired Equivalent Privacy

LISTAS DE ABREVIATURAS, SIGLAS

ACLs	Acces 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
DNS	Domain Name System
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
ISO	International Organization Standardization
ISODE	ISO Development Environment
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LTS	Longo Tempo de Suporte
MAC	Media Access Control
MSNT	Microsoft New Technology
NCSA	National Center for Supercomputing Applications
NTLM	New Technology Lan Manager
OSI	Open Systems Interconnection
OSLDAPD	Standalone LDAP
PAM	Pluggable Authentication Modules
RFC	Request For Comments
SASL	Simple Authentication and Security Layer
SMB	Server Message Block
SSID	Service Set Identifier
TCP/IP	Transmission Control Protocol/ Internet Protocol
TLS	Transport Layer Security
WEP	Wired Equivalent Privacy

LISTA DE FIGURAS

Figura 1: Estrutura da árvore LDAP	18
Figura 2: Estrutura da rede da FATERN.....	22
Figura 3: Comando de instalação do OpenLDAP	23
Figura 4: Conteúdo do arquivo slapd.conf.....	25
Figura 5: Definições de controle de acesso	25
Figura 6: Índices necessários para otimização das buscas.	26
Figura 7: Arquivo LDIF usado para população inicial do diretório	28
Figura 8: Comando de instalação do software PHPLDAPAdmin	28
Figura 9: Comando de instalação do Squid.....	30
Figura 10: Arquivo /etc/squid/squid.conf.....	30
Figura 11: Arquivo de autenticação com LDAP	31
Figura 12: Comando para recriar o cache do Squid.....	32
1.1 Surgimento e Evolução do LDAP	16
1.2 O Serviço de Diretório LDAP	17
1.3 Características do LDAP	19
2 DESENVOLVIMENTO DA SOLUÇÃO	21
2.1 AMBIENTE DE REDE	21
2.1.1 Sistema Operacional	21
2.1.2 Arquitetura	22
2.2 OPENLDAP	22
2.2.1 Instalação	23
2.2.2 Configuração	23
2.2.3 Índices	25
2.2.4 Banco de dados Berkeley DB - BDB	26
2.3 POPULAÇÃO DOS DADOS	27
2.4 PHPLDAPADMIN	28
2.5 AUTENTICAÇÃO	29
2.5.1 Proxy	29
2.5.2 Squid	30
2.6 INSTALAÇÃO E CONFIGURAÇÃO DO SQUID	31
2.7 AUTENTICAÇÃO EM LDAP	34
3 CONSIDERAÇÕES FINAIS	36
REFERÊNCIAS	37

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	PROBLEMA DE PESQUISA	11
1.2	OBJETIVOS.....	11
1.2.1	Objetivos Específicos	12
1.3	JUSTIFICATIVA	12
2	REFERENCIAL TEÓRICO	13
2.1	AUTENTICAÇÃO DE REDE SEM FIO.....	13
2.2	CONCEITO DE DIRETÓRIO	14
2.3	SERVIÇO DE DIRETÓRIO	15
2.4	LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP).....	15
2.4.1	Surgimento e Evolução do LDAP	16
2.4.2	O Serviço de Diretório LDAP	17
2.4.3	Características do LDAP	19
3	DESENVOLVIMENTO DA SOLUÇÃO	21
3.1	AMBIENTE DE REDE.....	21
3.1.1	Sistema Operacional	21
3.1.2	Arquitetura	22
3.2	OPENLDAP.....	22
3.2.1	Instalação	23
3.2.2	Configuração	23
3.2.3	Índices	25
3.2.4	Banco de dados Berkeley DB – BDB.....	26
3.3	POPULAÇÃO DOS DADOS.....	27
3.4	PHPLDAPADMIN	28
3.5	AUTENTICAÇÃO	29
3.5.1	Proxy	29
3.5.2	Squid	30
3.6	INSTALAÇÃO E CONFIGURAÇÃO DO SQUID	31
3.7	AUTENTICAÇÃO EM LDAP.....	34
4	CONSIDERAÇÕES FINAIS.....	36
	REFERÊNCIAS.....	37

1.2 OBJETIVOS

Diante dessa problemática surgiu a ideia de um projeto que teve como objetivo implementar um sistema de autenticação em uma rede com uma única base LDAP, possibilitando a unificação de diferentes serviços de autenticação em um único diretório. Assim, seria possível limitar o acesso à rede wireless da FATERN a apenas pessoas que

1 INTRODUÇÃO

Sabe-se que a identificação e a localização de pessoas e recursos são questões fundamentais num ambiente de rede. Um serviço de diretório fornece um modo universal de armazenar e localizar “objetos” que podem estar situados em qualquer parte da rede. Tais objetos podem ser pessoas ou recursos como discos, filas de impressão, endereços de e-mail, máquinas e diversos tipos de cadastros. Uma aplicação dessas funções é o serviço de ajuda de uma companhia telefônica, que nos informa o número do telefone de uma pessoa se souber o nome, ou endereço, ou algum outro dado dessa pessoa. Dado um conjunto de atributos de uma pessoa, servidor ou outro recurso, o serviço de diretório pode retornar qualquer informação desse recurso na rede. Outro exemplo de aplicação de um diretório atualmente é armazenar dados suficientes de usuários para realizar autenticação de algum outro serviço ou sistema.

1.1 PROBLEMA DE PESQUISA

Este trabalho trata como problema de pesquisa a segurança da rede sem fio (*wireless*) implantada na FATERN (Faculdade de Excelência Educacional do Rio Grande do Norte). Devido ao fácil acesso e a insegurança desta rede *wireless*, faz-se necessário que a FATERN possua algum mecanismo de autenticação na rede, onde apenas pessoas que façam parte da comunidade FATERN tenham acesso a essa rede, podendo assim, utilizá-la com mais segurança.

1.2 OBJETIVOS

Diante dessa problemática surgiu a idéia de um projeto que teve como objetivo implantar um sistema de autenticação em uma rede com uma única base LDAP, possibilitando a unificação de diferentes serviços de autenticação em um único diretório. Assim, seria possível limitar o acesso à rede *wireless* da FATERN a apenas pessoas que

fazem parte do grupo FATERN, assim aumentando a segurança e bom uso da rede. Neste estudo será apresentada uma arquitetura de rede local que dispõe de um serviço de diretório e um exemplo de autenticação de um serviço de *proxy*.

1.1 AUTENTICAÇÃO DE REDE SEM FIO

1.2.1 OBJETIVOS ESPECÍFICOS

Assim, o escopo do trabalho é sugerir à FATERN a implementação dessa arquitetura utilizando os softwares de diretório OpenLDAP e de *proxy* SQUID. Todos os passos do processo de implementação são documentados e explicados, acompanhados de um referencial teórico sobre os sistemas envolvidos. Os resultados dos testes realizados também são apresentados.

1.3 JUSTIFICATIVA

Essa proposta se justifica pela importância de se entender a necessidade da autenticação em uma rede *wireless* para obter a devida segurança dos dados que nela trafegam. Com a autenticação, apenas pessoas cadastradas e com as devidas permissões terão acesso a essa rede, evitando assim maiores problemas.

Visando atender os requisitos metodológicos do trabalho se fez necessário seguir algumas etapas como: estudo do protocolo LDAP; análise de configuração da rede *wireless* da FATERN; implementação de um servidor LDAP; implementação de um servidor *proxy-cache* Squid; e análise dos resultados do estudo de caso.

2 REFERENCIAL TEÓRICO

2.1 AUTENTICAÇÃO DE REDE SEM FIO

Atualmente com o aumento da mobilidade os indivíduos desejam se libertar dos escritórios e procuram alternativas que lhes garantam produtividade com liberdade. As redes sem fio possuem esse apelo natural, pois, sem fios, as estações de trabalho podem estar em qualquer lugar. Porém, é necessário garantir a segurança deste processo.

Percebe-se que cada vez mais há uma corrida para implementar novas tecnologias e rapidamente colocá-las à disposição dos consumidores. Com as redes sem fio não foi diferente, o trabalho de desenvolvimento aconteceu muito rapidamente e sem muita padronização, e como consequência foram feitas revisões de segurança pouco rigorosas. Como resultado, temos uma série de falhas de segurança nos padrões, e novas técnicas para explorá-las são publicadas regularmente (AHMAD et al, 2002).

Antes do estabelecimento do padrão IEEE 802.11 os fabricantes de equipamentos sem fio utilizavam tecnologia proprietária, o que tornava os usuários dependentes do par fabricante/tecnologia escolhido. O padrão do IEEE surgiu como uma solução para os consumidores, ávidos por mobilidade, mas os itens de segurança adotados não garantiram o crescimento esperado no número de usuários das redes locais sem fio. A adoção do padrão foi acompanhada pelo aparecimento de vulnerabilidades de segurança, que comprometem sua confiabilidade. A concepção dos mecanismos de segurança foi feita considerando um nível de administração não aplicado pelos administradores das redes e nem tão pouco por seus usuários.

O padrão IEEE 802.11 estabelece três mecanismos para prover segurança, sendo eles: filtragem por Identificador de Grupo de Serviço (*Service Set Identifier* - SSID), filtragem por endereço Controle de Acesso à Mídia (*Media Access Control* - MAC) e o Privacidade Equivalente à Rede Cabeada (*Wired Equivalent Privacy* - WEP).

O padrão IEEE 802.1X atende a dois requisitos básicos de segurança: privacidade e autenticação. Neste trabalho o foco será o tratamento da autenticação, mais especificamente utilizando um serviço de diretório dos usuários.

O padrão IEEE 802.1X atende a dois requisitos básicos de segurança: privacidade e autenticação. Neste trabalho o foco será o tratamento da autenticação, mais especificamente utilizando um serviço de diretório dos usuários.

2.2 CONCEITO DE DIRETÓRIO

Um diretório pode ser definido como uma base de dados global, que pode estar fisicamente distribuída, mas mantém uma centralização lógica, cujo objetivo é armazenar e fornecer informações sobre objetos. Esses objetos podem ser pessoas, equipamentos, listas de distribuição, endereços de rede, endereços de aplicações, organizações, endereços de correios eletrônicos, dentre outros diversos dados cadastrais (ISQUEIRDO, 2001).

Não há restrições quanto aos objetos que podem ser armazenados num diretório. Embora sejam muito usados por aplicações interessadas em armazenar e recuperar informações sobre recursos de redes, os serviços de diretório não são de modo algum limitados ou especificamente destinados a essa área de aplicação. O administrador de um serviço de diretório pode permitir que qualquer tipo de informação seja colocado no diretório.

Dois aspectos diferenciam os diretórios dos bancos convencionais: o valor mais alto da razão consultas/alterações e a inexistência de transações atômicas. Os serviços de diretório se distinguem dos serviços de nomes por oferecerem mais recursos de busca (pesquisa) e recuperação de informações.

No contexto de uma rede, um diretório é uma estrutura hierárquica que armazena informações a respeito de itens (recursos) de uma rede. Um diretório pode ser composto por diversos tipos de itens (objetos), como, por exemplo, servidores, discos, impressoras, contas de usuários, arquivos compartilhados; ou ainda domínios de *logon*, aplicativos, políticas de segurança e de acesso (BRITO, 2000).

Cada item é um objeto e possui uma série de informações (atributos) associadas a ele. Por exemplo, os atributos de um determinado objeto "usuário" podem ser nome, endereço, nome de usuário (*login name*), senha de acesso e grupos do qual faz parte. O serviço de diretório é um conjunto de funções para criação, armazenamento e recuperação de informações de um diretório.

Diversos aplicativos, ao funcionarem em redes de computadores, podem necessitar de um serviço de diretório. Podemos citar como exemplo: agendas de correio eletrônico, as quais precisam localizar o endereço eletrônico do usuário a partir do nome; serviços para cadastro de contas de usuário, que precisa localizar as informações de senha a partir do nome do usuário.

2.3 SERVIÇO DE DIRETÓRIO

Serviço de diretório é um software de rede com um banco de dados otimizado para ler, navegar e procurar objetos neste diretório. Os diretórios tendem a conter descritivos, atributos/características e um suporte sofisticado para filtros. Os serviços de diretórios geralmente não suportam complicadas transações de registros, grandes volumes de dados, encontradas atualmente em sistema de banco de dados.

As atualizações dos diretórios são simples e rápidas. São feitos para respostas rápidas de um alto volume de operações de buscas. É possível replicar informações largamente para aumentar a confiança e a disponibilidade do recurso.

2.4 LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)

O LDAP é um protocolo para atualizar e pesquisar diretórios rodando sobre TCP/IP (*Transmission Control Protocol/ Internet Protocol*). Um diretório LDAP segue o modelo X.500, que é uma árvore de nós, cada qual consistindo de um conjunto de atributos com seus respectivos valores. O LDAP foi criado como alternativa ao pesado protocolo conhecido como DAP (*Directory Access Protocol*) (SENA, 2005).

Um diretório LDAP tende a refletir vários limites políticos, geográficos e/ou organizacionais, dependendo do modelo adotado. A utilização do LDAP hoje em dia tende a se basear nos nomes já existentes do sistema DNS (*Domain Name System*), na estruturação

dos níveis mais básicos de hierarquia. Mais profundamente, podem aparecer estruturas representando pessoas, unidades organizacionais, impressoras, documentos, grupos de pessoas ou qualquer outra coisa que represente um nó. (Wikipedia- LDAP, 2009).

O LDAP é um serviço de diretório padronizado pela IETF (*Internet Engineering Task Force*). Ele mantém uma base de dados global com informações sobre objetos, fornecendo interfaces para busca, inserção, remoção e atualização dessas informações. O LDAP tem um conjunto de facilidades de busca bastante rico, que permite realizar buscas complexas em todo o diretório ou em partes dele. Possui mecanismos de autenticação, garantindo a segurança das informações no diretório através de controles de acesso, e oferece API (*Application Program Interface*) padronizadas e bem documentadas.

Como o LDAP é um padrão aberto mantido pela IETF, ele pode ser utilizado por qualquer desenvolvedor, companhia, ou administrador sem receio de ficar aprisionado a protocolos proprietários ou a vendedores específicos, e permite que a escolha da implementação seja baseada nos detalhes do projeto em vez de preocupações de interoperabilidade.

O autor desse protocolo foi Tim Howes da Universidade de Michigan, Steve Kille da ISODE (*ISO Development Environment*) e Wengyik Yeong da *Performance Systems Internacional*.

Um cliente começa uma sessão de LDAP ligando-se a um servidor LDAP, normalmente pela porta padrão: 389 TCP. Este envia requisições para o servidor, o qual devolve respostas. As operações básicas são:

- *Bind* – autentica e especifica a versão do protocolo LDAP;
- *Search* – procura por e / ou recupera entradas dos diretórios;
- *Compare* – teste se uma entrada tem determinado valor como atributo;
- *Add* – adiciona uma nova entrada;
- *Delete* – apaga uma entrada;
- *Modify* – modifica uma entrada;
- *Modify DN* – move ou renomeia uma entrada;
- *Start TLS* – protege a conexão com a Transport Layer Security (TLS);
- *Abandon* – aborta uma requisição prévia;
- *Extended Operation* – operação genérica para definir outras operações;
- *Unbind* – fecha a conexão, não o inverso de Bind.

2.4.1 Surgimento e Evolução do LDAP

Quando a ISO (*international Organization Standardization*) e o CCITT (*Consultative Committee for International Telegraphy and Telephony*) 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 cliente e o servidor de Diretório usava DAP (*Directory Access Protocol*) que era executado sobre a pilha de protocolos do modelo OSI (*Open Systems Interconnection*). O fato de o X.500 ser muito complexo e de custo incompatível, levou os pesquisadores da Universidade de Michigan a criar um servidor LDAP *standalone*, o SLAPD, que atuava sobre o TCP/IP.

Em 1993 o LDAP foi então apresentado como alternativa ao protocolo DAP para acesso a Diretórios 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 Diretório completo, passando a competir diretamente com o X.500. Em dezembro de 1997, o IETF (*Internet Engineering Task Force*) lançou a versão 3 do LDAP como proposta padrão Internet para serviços de Diretório. Atualmente várias empresas oferecem produtos LDAP, incluindo a Microsoft, Netscape e Novell. A OpenLDAP Foundation disponibiliza uma implementação *Open Source* do Serviço de Diretório 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.

2.4.2 O Serviço de Diretório LDAP

No LDAP, as entradas dos diretórios são organizadas de forma hierárquica, como uma estrutura de árvore. Essa estrutura possui um limite organizacional.

Um possível exemplo prático dessa estrutura em árvore seria a mostrada na Figura 1. As entradas são representadas pelos países que aparecem no topo da árvore. Abaixo aparecem as entradas dos estados, logo abaixo as entradas que representa o nome da Empresa. Abaixo as entradas que representam os setores da empresa. A seguir as entradas que representam as pessoas, documentos, impressoras ou qualquer coisa que se possa imaginar dentro dessa estrutura hierárquica.

O LDAP foi concebido efetuando-se as seguintes simplificações na especificação X.500 (*Howes e Smith*):

Transporte – o LDAP executa diretamente sobre TCP (Transmission Control Protocol), evitando o “overhead” das camadas superiores da pilha de protocolo OSI.

Representação de Dados – no LDAP a maioria dos elementos de dados são representados como cadeias de caracteres, processadas bem mais facilmente que dados na representação estruturada “ASN.1” (*Abstract Syntax Notation One*) 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 operações redundantes do X.500.

Interrogação:

LDAP_search – faz pesquisas das entradas no Diretório;

LDAP_compare – verifica se uma entrada contém um dado um atributo;

Atualização:

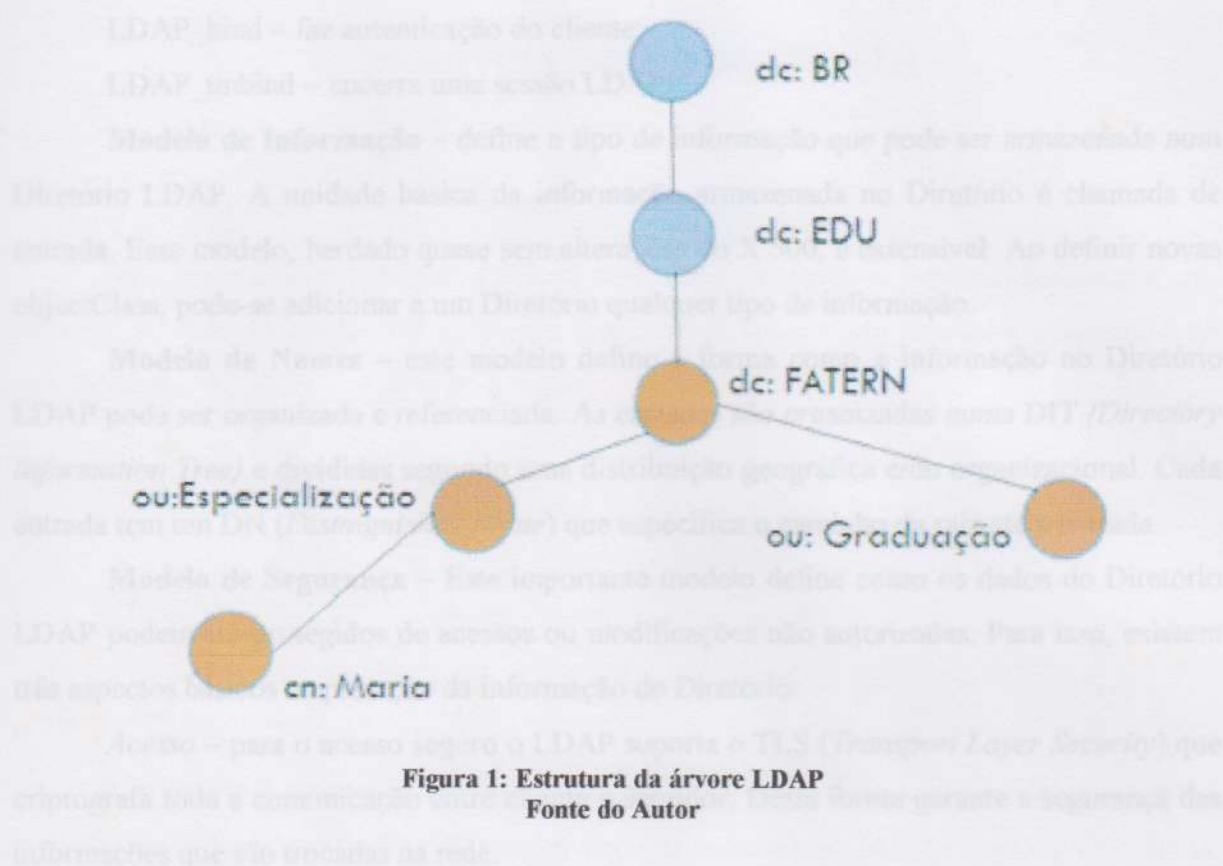
LDAP_modify – altera uma entrada existente;

LDAP_add – adiciona uma nova entrada;

LDAP_delete – apaga uma entrada existente;

LDAP_modify – renomeia uma entrada existente;

Autenticação:



2.4.3 Características do LDAP

Do ponto de vista conceitual, o serviço LDAP é composto pelos seguintes Modelos:

Modelo Funcional – define o que pode ser feito com a informação no Diretório LDAP e como podemos alterá-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 Diretório;

LDAP_compare – verifica se uma entrada contém um dado num atributo;

Atualização:

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 Diretório LDAP. A unidade básica da informação armazenada no Diretório é chamada de entrada. Esse modelo, herdado quase sem alterações do X.500, é extensível. Ao definir novas objectClass, pode-se adicionar a um Diretório qualquer tipo de informação.

Modelo de Nomes – este modelo define a forma como a informação no Diretório LDAP pode ser organizada e referenciada. As entradas são organizadas numa DIT (*Directory Information Tree*) e divididas segundo uma distribuição geográfica e/ou organizacional. Cada entrada tem um DN (*Distinguished Name*) que especifica o caminho da raiz até a entrada.

Modelo de Segurança – Este importante modelo define como os dados do Diretório LDAP podem ser protegidos de acessos ou modificações não autorizadas. Para isso, existem três aspectos básicos na proteção da informação do Diretório:

Acesso – para o acesso seguro o LDAP suporta o TLS (*Transport Layer Security*) 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 (*Simple Authentication and Security Layer*), que permite que o cliente e o 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 (*Access Control Lists*).

3 DESENVOLVIMENTO DA SOLUÇÃO

Para realizar este trabalho, foi feita uma série de testes e implementações em um ambiente de rede simulado. Tal ambiente, que tinha o objetivo de simular a arquitetura de rede mostrada na Figura 2, é descrito a seguir.

3.1 AMBIENTE DE REDE

Para efeito dos testes realizados, o hardware não possui importância. Já os sistemas operacionais e os demais softwares passam por um processo de escolha baseada em alguns fatores. A seguir são indicados os softwares utilizados para realizar os testes.

3.1.1 Sistema Operacional

Ubuntu é uma antiga palavra africana que significa algo como "Humanidade para os outros" ou ainda "Sou o que sou pelo que nós somos". A distribuição do Linux chamada Ubuntu traz o espírito desta palavra para o mundo do software livre.

O Ubuntu é uma distribuição de Sistema Operacional Linux, desenvolvida por uma comunidade de software livre, com uma forte base na distribuição Debian GNU/Linux e no GNOME Desktop. Seu objetivo é ser livre, gratuito, agradável e confiável para ser utilizado em laptops, desktops e servidores.

O seu desenvolvimento tem grande preocupação com a segurança. Uma nova versão é lançada a cada seis meses, incluindo atualizações de segurança gratuitas por pelo menos dezoito meses. Manter o sistema atualizado é muito importante, com isto correções de falhas ou *bugs* podem ser aplicadas ao sistema e mantê-lo seguro.

3.1.2 Arquitetura

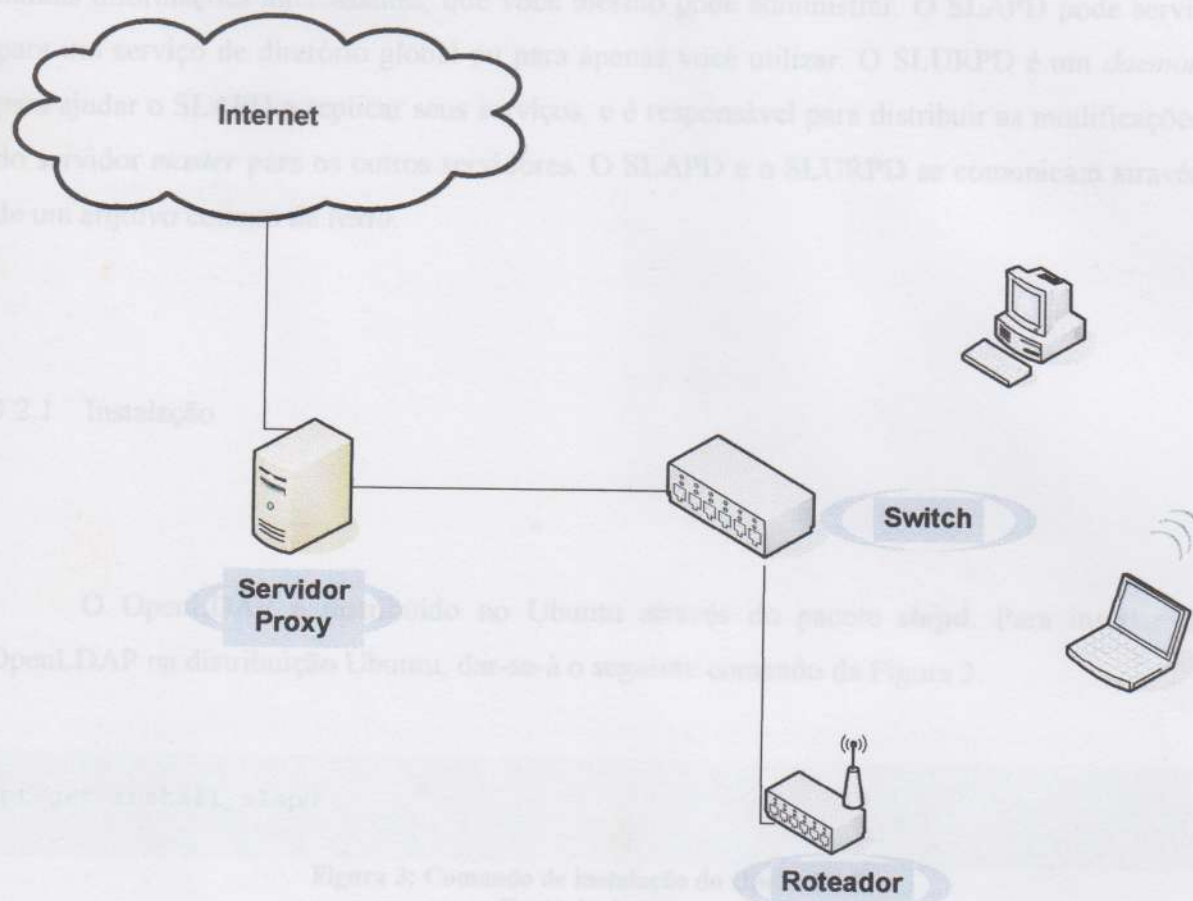


Figura 2: Estrutura da rede da Fatern
Fonte do Autor

3.2 OPENLDAP

É um software livre de código aberto que implementa o protocolo LDAP. Ele é um serviço de diretório baseado no padrão X.500.

A autenticação centralizada com OpenLDAP implementa o servidor OpenLDAP como meio de autenticação para vários serviços como: e-mail, web, *proxy* e FTP. Além da autenticação de usuários para redes heterogêneas (com *Linux* e *Windows*). Aborda ainda, a parte de *troubleshooting* e criptografia.

A suíte de servidores OpenLDAP é formada pelos softwares servidores SLAPD e SLURPD. O SLAPD é um servidor de diretórios LDAP que roda em vários tipos de plataformas. Utilizado para criar os serviços de diretório. Em cada diretório pode conter muitas informações interessantes, que você mesmo pode administrar. O SLAPD pode servir para um serviço de diretório global ou para apenas você utilizar. O SLURPD é um *daemon* para ajudar o SLAPD a replicar seus serviços, e é responsável para distribuir as modificações do servidor *master* para os outros servidores. O SLAPD e o SLURPD se comunicam através de um arquivo comum de texto.

3.2.1 Instalação

O OpenLDAP é distribuído no Ubuntu através do pacote **slapd**. Para instalar o OpenLDAP na distribuição Ubuntu, dar-se-á o seguinte comando da Figura 3.

```
apt-get install slapd
```

Figura 3: Comando de instalação do OpenLDAP
Fonte do Autor

3.2.2 Configuração

Para configurar o OpenLDAP no Ubuntu, é necessário configurar o arquivo `slapd.conf` localizado no diretório `/etc/ldap`. A Figura 4 mostra o conteúdo deste arquivo.

```
# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.

#####
# Global Directives:

# Features to permit
allow bind_v2
```

```
# Schema and objectClass definitions
include      /etc/ldap/schema/misc.schema
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile      /var/run/slapd/slapd.pid

# List of arguments that were passed to the server
argsfile     /var/run/slapd/slapd.args

# Read slapd.conf(5) for possible values
loglevel     256

# Where the dynamically loaded modules are stored
modulepath   /usr/lib/ldap
moduleload   back_hdb
moduleload   back_bdb

# The maximum number of entries that is returned for a search operation
sizelimit    500

# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1

#####
# Specific Backend Directives for hdb:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs

backend      bdb

#####

database     bdb
suffix       "dc=fatern,dc=edu,dc=br"

rootdn       "cn=admin,dc=fatern,dc=edu,dc=br"
rootpw       {SSHA}D/984gKir8UK9ZO3KV2LJuleQH2x16oX

directory    "/var/lib/ldap"

dbconfig set_cachesize 0 209715200 0
dbconfig set_lk_max_objects 1500
dbconfig set_lk_max_locks 1500
dbconfig set_lk_max_lockers 1500

index        objectClass eq
index        uid eq

lastmod      on
checkpoint   512 30

access to attrs=userPassword,shadowLastChange
    by dn="cn=admin,dc=fatern,dc=edu,dc=br" write
    by self write
```

```

        by * read

access to *
    by dn="cn=admin,dc=fatern,dc=edu,dc=br" write
    by * read

```

Figura 4: Conteúdo do arquivo slapd.conf
Fonte do Autor

O arquivos slapd.conf está dividido em três partes de configuração: *Global Directive* (diretiva global), *Backend* e *Database* (banco de dados). A base do diretório foi definida pelo modelo de *dcObjects* (dc=meu domínio, dc=edu, dc=br). Um ponto importante para mencionar é a gerência de controle de acesso, onde é dada a permissão de acesso usada na diretiva de banco de dados. Estávamos usando a estrutura abaixo.

```

# definicoes de controle de acesso
21. access to attr=senhaUsuario
22. by self write
23. by anonymous auth
24. by dn="cn=Admin,dc=example,dc=com"
    write
25. by * none
26. #definicao ldb

```

Figura 5: Definições de controle de acesso
Fonte do Autor

A linha `by anonymous auth`, foi um dos problemas encontrados nos testes, pois sempre que se tentava autenticar no servidor *proxy* era retornado uma mensagem de erro. Essa linha declara que qualquer pessoa pode se autenticar. Sendo assim substituímos para `by * read`, a qual indica que qualquer pessoa pode ler.

3.2.3 Índices

Os índices são usados para melhorar o desempenho das buscas. Uma busca é feita através de um filtro, que pode ser formado por um *objectClass* ou um atributo qualquer. Sem o uso de índices, a busca percorre cada entrada de todas as que fazem parte do escopo da pesquisa para checar e comparar seu valor. Com o uso de índices, a busca por um filtro indexado lerá o índice e fará a comparação apenas nas entradas apontadas pelo índice, ou seja,

as entradas que possuem tal filtro. Na implementação foi usado a configuração vista na Figura 6.

index	objectClass eq
index	uid eq

Figura 6: Índices necessários para otimização das buscas
Fonte do Autor

A primeira linha da Figura 6 indexa o atributo `objectClass`, para otimizar o desempenho das pesquisas por tipos de objeto dentro do diretório. A segunda linha indexa o atributo `uid`, para otimizar as pesquisas por nome de usuário, que serão realizadas pelo proxy para realizar autenticação. Ambas as linhas são indexadas através do método `eq`, que é utilizado para buscas por igualdade, isto é, utilizado para buscar valores exatamente iguais aos passados para o filtro.

3.2.4 Banco de Dados Berkeley DB – BDB

O Banco de Dados Berkeley DB foi originado na Universidade da Califórnia, em Berkeley. Em 1996, a *Netscape* pediu a autores de *Berkeley DB* para melhorar e ampliar a biblioteca. Então surgiu a versão 1.85 para atender aos requisitos do *Netscape* para um servidor LDAP para uso neste navegador. Este pedido levou a criação da *Sleepycat Software*, posteriormente adquirida pela *Oracle Corporation*, em fevereiro de 2006. Aqueles que não desejam acatar os termos da licença pública *Sleepycat* tem a opção de compra de outra licença proprietária para redistribuição da *Oracle Corporation*. Esta técnica é chamada duplo licenciamento.

Berkeley DB tem uma arquitetura nomeadamente mais simples do que a de outros bancos de dados como o *Microsoft SQL Server* e *Oracle*. Por exemplo, ela não fornece suporte para o acesso a rede – programas acessar o banco de dados usando no processo de chamadas API. Ela não suporta o SQL ou qualquer outro idioma consulta, nem o apoio de esquemas tabela ou tabelas colunas. Um programa acessando o banco de dados é livre de

decidir a forma como os dados devem ser armazenados em um registro. *Berkeley DB* não coloca restrições sobre o registro de dados (Wikipedia - *Berkeley DB*, 2009).

Apesar de ter uma arquitetura simples, *Berkeley DB* suporta muitas funcionalidades avançadas de dados, tais como ácidos operações, *fine-grained locking* e replicação.

3.3 POPULAÇÃO DOS DADOS

A população dos dados é criada através do arquivo LDIF (*LDAP Data Interchange Format*), que é um arquivo o qual possui um formato textual utilizado para adicionar, alterar e remover informações da base de dados. Devendo sempre conter DN's (*Distinguished Name*) referenciando entradas e nomes, seu padrão é definido na RFC (*Request For Comments*) 2849. Na implementação foi criada a Organização FATERN; as Unidades Organizacionais Usuários e Grupos; e dentro da Unidade Organizacional os usuários Maria e Lineu, mostrados na Figura 7.

Figura 7: Arquivo LDIF usado para população inicial da diretório
Fonte: do Autor

```
dn: dc=fatern,dc=edu,dc=br
objectClass: dcObject
objectClass: organization
dc: fatern
o: FATERN
description: Faculdade de Excelencia Educacional

dn: ou=Usuarios,dc=fatern,dc=edu,dc=br
objectClass: top
objectClass: organizationalUnit
ou: Usuarios

dn: ou=Grupos,dc=fatern,dc=edu,dc=br
objectClass: top
objectClass: organizationalUnit
ou: Grupos

dn: cn=usuarios,ou=Grupos,dc=fatern,dc=edu,dc=br
objectClass: posixGroup
objectClass: top
cn: usuarios
gidNumber: 1000

dn: cn=computadores,ou=Grupos,dc=fatern,dc=edu,dc=br
objectClass: posixGroup
objectClass: top
cn: computadores
gidNumber: 1001

dn: cn=Lineu Paiva,ou=Usuarios,dc=fatern,dc=edu,dc=br
```

```

objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
givenName: Lineu
sn: Paiva
cn: Lineu Paiva
uid: lineu.paiva
userPassword: {SSHA}OX5t8SSqenwAQYYHmiQxz7NABeNLLPFL
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/lineu.paiva
loginShell: /bin/sh

dn: cn=Maria Silva,ou=Usuarios,dc=fatern,dc=edu,dc=br
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
givenName: Maria
sn: Silva
cn: Maria Silva
uid: maria.silva
userPassword: {SSHA}OX5t8SSqenwAQYYHmiQxz7NABeNLLPFL
uidNumber: 1001
gidNumber: 1000
homeDirectory: /home/maria.silva
loginShell: /bin/sh

```

Figura 7: Arquivo LDIF usado para população inicial do diretório
Fonte do Autor

3.4 PHPLDAPADMIN

O PHPLDAPAdmin é um software com interface web onde é possível consultar, visualizar e administrar o banco do OpenLDAP, assim como sua estrutura de objetos. No entanto, nada tão poderoso se comparado aos comandos do modo texto. Ele usa a porta 389 do protocolo IP aberta pelo slapd. Para instalá-lo é usado o comando visto na Figura 8.

```
apt-get install phpldapadmin
```

Figura 8: Comando de instalação do software PHPLDAPAdmin
Fonte do Autor

3.5 AUTENTICAÇÃO

A autenticação é a operação de validar um usuário através de uma senha. Caso um diretório possua contas de usuário, ele pode ser utilizado para realizar autenticação dessas contas de usuário através de um atributo de senha. Diversos tipos de objectClass podem servir como base para criação de contas de usuário. No exemplo ilustrado na figura 7, o objectClass utilizado foi o PosixAccount, que possui um atributo uid para armazenar um identificador de usuário e um atributo userPassword para armazenar a senha.

As operações envolvidas em uma autenticação começam com uma busca por um valor de uid recebido. Caso uma entrada que contém esse uid seja encontrada, o valor do atributo userPassword é comparado com uma senha recebida. A resposta de validação é então enviada como resultado da operação.

Diversas implementações de autenticação podem existir sobre entradas de diretório: as ferramentas incluídas na suíte OpenLDAP; o programa PHPLDAPAdmin; um módulo PAM para autenticação de sistemas Unix; um módulo do sistema de proxy Squid para autenticar usuários de http; entre outros.

3.5.1 Proxy

Em sua grande maioria, os navegadores de páginas web, fazem conexões diretas com a Internet. Mas há outra forma bem mais interessante de conexão: eles podem ser configurados para se conectarem através de um servidor *proxy*.

O *proxy* é um serviço que está disponível em um ambiente servidor, que recebe requisições das estações de trabalho para conexões à Internet, onde seu papel fundamental é buscar a informação primeiramente no seu *cache* local e caso não encontre o documento requisitado, faz a busca no site solicitado pela estação de trabalho. Na segunda situação, o endereço Internet que fica registrado no servidor da página solicitada, é o do servidor *proxy*, pois o mesmo é o dispositivo que está entre a rede local e a Internet.

O servidor *proxy* surgiu da necessidade de ligar a rede local à grande rede de computadores, a Internet, através de um computador que provesse o compartilhamento de Internet com os demais computadores. Pode-se fazer a seguinte analogia: rede local é uma rede interna e a Internet é uma rede externa, sendo assim, *proxy* é o dispositivo que permite as máquinas da rede interna se conectar ao mundo externo. Como na maioria dos casos as máquinas da rede local não têm um endereço válido para a Internet, elas fazem a solicitação de um endereço externo para o servidor *proxy*, que encaminha a requisição a Internet. Caso não ache o documento solicitado em seu *cache* de Internet, o servidor está habilitado a fazer essa consulta, pois o mesmo tem um endereço válido na Internet. Sendo assim, pode-se dizer que é normal ter um servidor *proxy* diretamente ligado à Internet e com um endereço válido (CONNECTIVA, 2001).

Um dos elementos mais importante de um servidor *proxy* é o seu *cache*, além é claro dos seus filtros de bloqueio ou liberação de sites, as ACLs (*Access Control Lists*).

3.5 INSTALAÇÃO E CONFIGURAÇÃO DO SQUID

3.5.2 Squid

O Squid é o servidor *proxy* mais utilizado atualmente na Internet. Suporta os protocolos de comunicação HTTP (*HyperText Transfer Protocol*), FTP (*File Transfer Protocol*). Completamente livre e com excelente suporte para operação em servidores Linux. Com ele você pode instalar um servidor Linux com acesso a Internet, e fazer com que outras máquinas clientes (usando Linus, Windows ou outro Sistema Operacional) acessem páginas web e sites FTP através do servidor Linux, mesmo que estas máquinas clientes não tenham conexão direta com a Internet – tudo que elas precisam é o acesso ao próprio servidor onde está rodando o Squid. A única configuração necessária na máquina cliente é feita no próprio browser: você precisa definir qual o endereço do servidor *proxy*.

O Squid é um servidor *proxy* para os protocolos já mencionados anteriormente. Portanto o acesso a outros serviços como, por exemplo, o correio eletrônico, deve ser configurado com a ferramenta responsável pelo filtro de pacotes, que trabalha diretamente na camada de rede (CONNECTIVA, 2001).

O arquivo de configuração do Squid é chamado de *squid.conf*. O Squid é totalmente configurado nesse arquivo, ou seja, toda e qualquer alteração nas ACLs ou alguma

configuração específica deve ser feita nesse arquivo. Este arquivo é organizado em *tags* que tratam de todas as configurações, tais como porta de acesso ao servidor, programa utilizado para manipulação de senhas, tamanho e estruturação do *cache*, definição e manipulação das ACLs, que vão estipular quais são os grupos de usuários a serem utilizados, quais são os arquivos com os sites proibidos e/ou liberados, quais são as extensões dos downloads proibidos, entre outras *tags*.

O que normalmente sofre maiores alterações, e com mais frequência, são justamente as regras definidas na tag ACL, onde podem ser alterados os usuários, como também as listas de sites proibidos e/ou liberados e as extensões dos downloads proibidos.

O servidor Squid pode ser obtido no seu site oficial: <http://www.squid-cache.org.br>

3.6 INSTALAÇÃO E CONFIGURAÇÃO DO SQUID

Em um sistema Ubuntu, o Squid pode ser instalado através do utilitário de linha de comando apt-get, que realiza o download do pacote de software mais recente e faz a instalação, conforme Figura 9.

```
apt-get install squid
mkdir /etc/squid/regras
touch /etc/squid/regras/iplivres
touch /etc/squid/regras/ipbloqueados
touch /etc/squid/regras/liberadototal
touch /etc/squid/regras/proibidototal_n1
```

Figura 9: Comandos de instalação do Squid
Fonte do Autor

Após a instalação, é necessário criar algumas pastas e arquivos que serão úteis na configuração do Squid, como exibidos na Figura 9.

A Figura 10 mostra o arquivo de configuração do Squid, localizado em `/etc/squid/squid.conf`. No exemplo abaixo não foi utilizada a porta padrão do Squid, a 3128, e sim a porta 8080. Na configuração foi definido também o controle de acesso ao Proxy através de ACL (*Access Control Lists* – Listas de Controle de Acesso). Essas listas dão nomes a objetos definidos pelo administrador, e esses objetos podem ser domínios de origem, domínios de destino, endereços de IP, etc.

```

acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
acl dstlocal dst 10.0.0.0/8
cache deny dstlocal
http_access allow dominiolocal
visible_hostname squid.fatern.edu.br

# SOBRE O CACHE
cache_mem 256 MB
maximum_object_size 200 MB
cache_dir ufs /var/spool/squid 10000 16 256
cache_access_log /var/spool/squid/access.log

# PAGINAS DE ERROS E AVISOS
#error_directory /var/www/squiderrors

# ALCs BASICAS (NAO ALTERAR)
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563 2096 2083
acl Safe_ports port 80 # http
acl Safe_ports port 81 # biblioteca
acl Safe_ports port 82 # biblioteca
acl Safe_ports port 83 # biblioteca
acl Safe_ports port 8080 # http
acl Safe_ports port 2083 # cpanel
acl Safe_ports port 2096 # cpanel
acl Safe_ports port 1863 # msn
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

###
### ACLs ###
###

# ORIGENS
acl rede_lan src 10.0.0.0/255.0.0.0
acl ip_livres src "/etc/squid/regras/iplivres"
acl ip_bloqueados src "/etc/squid/regras/ipbloqueados"

# WHITELIST
acl liberado_total url_regex -i "/etc/squid/regras/liberadototal"

# BLACKLIST
acl proibido_total_n1 url_regex -i "/etc/squid/regras/proibidototal_n1"

###
### REGRAS ###
###

# Autenticacao com LDAP
auth_param basic program /usr/lib/squid/ldap_auth -b
"ou=Usuarios,dc=fatern,dc=edu,dc=br" -f "uid=%s" -v 3 -h 10.13.1.136

```

```

auth_param basic realm LOGIN - FATERN
auth_param basic children 5
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
acl autenticados proxy_auth REQUIRED

http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports

# Proibido ate na pesquisa do google
http_access deny proibido_total_n1

# Sites liberados nivel maximo
http_access allow liberado_total

# REDE LAN
http_access allow rede_lan autenticados all

# Bloqueia o que passou sem querer
http_access deny all
icp_access allow all

```

Figura 10: Arquivo /etc/squid/squid.conf
Fonte do Autor

A Figura 11 destaca o trecho do arquivo squid.conf que define a autenticação do proxy em um servidor LDAP. Primeiramente, foi definido qual programa será utilizado para realizar a autenticação. O programa `ldap_auth` recebe como parâmetro uma base de diretório onde irá procurar um objeto, e recebe um identificador do objeto para usar como filtro na pesquisa. Além disso é indicado o endereço IP do servidor LDAP. O objeto, neste caso, é um usuário identificado por um atributo "uid" e localizado na base "ou=Usuários, dc=fatern, dc=edu, dc=br". Diversos parâmetros são especificados para o processo de autenticação de http que será iniciado um navegador Web. Por último, a ACL "autenticados" indica que a autenticação de proxy é requerida e guardará nela todos os usuários que realizarem autenticação com sucesso.

```

# Autenticacao com LDAP
auth_param basic program /usr/lib/squid/ldap_auth -b
"ou=Usuarios,dc=fatern,dc=edu,dc=br" -f "uid=%s" -v 3 -h 10.13.1.136
auth_param basic realm LOGIN - FATERN
auth_param basic children 5
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off
acl autenticados proxy_auth REQUIRED

```

Figura 11: Autenticação com LDAP
Fonte do Autor

Caso seja necessário recriar o repositório de *cache*, é preciso executar o comando da Figura 12.

```
squid -z
/etc/init.d/squid restart
```

Figura 12: Comando para recriar o *cache* do Squid
Fonte do Autor

3.7 AUTENTICAÇÃO EM LDAP

A autenticação do Squid só pode ser habilitada se o mesmo for configurado em modo *proxy cache* (MARCELO, 2005). Caso seja configurado no modo transparente *proxy cache*, a autenticação não permitida com seus módulos padrões. Os módulos de autenticação padrões do Squid são (VESPERMAN, 2001):

Lightweight Directory Access Protocol (LDAP) – módulo que permite a autenticação baseada no banco de dados LDAP;

Microsoft New Technology (MSNT) – módulo que permite a autenticação baseada em um controlador de domínio Windows NT;

National Center for Supercomputing Applications (NCSA) – módulo que permite a autenticação baseado no tipo de arquivo *password* de muitos servidores web NCSA e segundo Marcelo (2005, p. 23) esse é o mais utilizado;

Pluggable Authentication Modules (PAM) – é um módulo de autenticação plugável e pode ser configurado para utilizar vários sistemas de autenticação;

Server Message Block (SMB) – módulo que permite a autenticação baseado em um servidor SMB tipo Microsoft NT ou Samba;

New Technology Lan Manager (NTLM) – módulo baseado em um protocolo de desafio / resposta, muito utilizado em ambiente Microsoft;

Getpwnam – módulo baseado nos arquivos de senhas do GNU/Linux: o *passwd* e o *shadow*.

O Squid utiliza processos auxiliares para processar as solicitações de autenticação para evitar que o mesmo seja parado ou bloqueado por causa de conexões lentas. Esses processos auxiliares são conectados por *pipes* Unix padrão e o Squid se comunica através de entradas e saídas padrão. Se o processo responder "OK", a autenticação foi feita; se responder "ERR", a autenticação falhou (VESPERMAN, 2001).

O utilitário `squid_ldap_auth` é uma ferramenta que conecta em um servidor LDAP e realiza autenticação de senha e identificador. Uma vez aberto, ele espera por uma entrada de texto simples, a qual é enviada pelos navegadores Web na forma de um par identificador e senha separados por um espaço simples. Como esperado pelo Squid, o utilitário responde com OK ou ERR, de acordo com a resposta obtida no servidor LDAP.

4 CONSIDERAÇÕES FINAIS

As tecnologias de redes de computadores sem fio hoje estão entre as mais utilizadas, devido a sua facilidade de uso, comodidade e flexibilidade. Porém, como esse meio de acesso é compartilhado, diversos métodos de segurança devem ser oferecidos para garantir que somente usuários legítimos tenham acesso aos recursos da rede.

Uma das primeiras e mais simples formas de segurança sugeridas para o ambiente de rede *wireless* da FATERN é a utilização de autenticação e autorização para acesso ao meio. Com o processo de autenticação, somente usuários legítimos e devidamente identificados terão acesso aos recursos disponíveis. Já o processo de autorização fornece flexibilidade para implementar uma hierarquia de acesso, tornando a rede mais segura e mantendo o foco principal para fins de pesquisa e crescimento intelectual dos que a utilizam. Os serviços descritos neste documento possibilitam uma implementação simples desses aspectos, com baixo custo e eficiência operacional.

5 REFERÊNCIAS

AHMAD, et all. **Hack Proofing your Wireless Network**. Syngress, 2002.

BERKELEY DB: WIKIPEDIA, a enciclopédia livre. Disponível em: <http://en.wikipedia.org/wiki/Berkeley_DB>. Acesso em 14 fev.2009.

BRITO, Ivana Campos. **Serviço de Diretório**. Salvador. 16. Trabalho Apresentado a Disciplina de Redes do curso de Bacharelado em Informática da Universidade Católica do Salvador. 2000. Disponível em:
<http://www.logicengenharia.com.br/mcamara/ALUNOS/diretorios.pdf>. Acesso em 10 fev.2009.

EQUIPE CONECTIVA. **Segurança de redes: firewall**. Curitiba. Conectiva, 2001. Disponível em: <<http://www.dimap.ufrn.br/~aguilar/Manuais/Servidor/index.html>>. Acesso em 11 fev. 2009.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social**. São Paulo: Atlas, 1987.

ISQUIERDO, Gustavo Scalco. **Integração do serviço de diretório LDAP com o serviço de nomes CORBA**. 2001. Dissertação (mestrado em Ciência da Computação) - Departamento de Matemática e Estatística da Universidade, Universidade de São Paulo, São Paulo, 2001. Disponível em: www.teses.usp.br/teses/disponiveis/45/45134/tde-28052003-100121/. Acesso em 10 fev. 2009.

LDAP. Disponível em: <<http://www.vivaolinux.com.br/artigo/OpenLDAP-a-chave-e-a-centralizacao?pagina=2>>. Acesso em 10 fev. 2009.

LDAP: WIKIPEDIA, a enciclopédia livre. Disponível em:
<<http://pt.wikipedia.org/wiki/LDAP>>. Acesso em 09 fev. 2009.

LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL. Disponível em:
<<http://www.ldap.org.br>>. Acesso em 09 fev. 2009.

MARCELO, A. **Squid: configurando o proxy para Linux**. 4. ed. Rio de Janeiro: Brasport, 2005.

OPENLDAP: WIKIPEDIA, a enciclopédia livre Disponível em: <<http://pt.wikipedia.org/wiki/OpenLDAP>>. Acesso em 10 fev. 2009.

OPENLDAP. Disponível em: <<http://www.openldap.org.br>>. Acesso em 10 fev. 2009.

PHPLDAPADMIN. Disponível em: <<http://under-linux.org/wiki/index.php/Tutoriais/LDAP/OpenLDAP-phpLDAPadmin>>. Acesso em 11 fev. 2009.

PROXY: WIKIPEDIA, a enciclopédia livre. Disponível em: <<http://pt.wikipedia.org/wiki/Proxy>>. Acesso em 01 mar. 2009.

SENA, Clóvis. **LDAP - Lightweight Directory Access Protocol – UM GUIA PRÁTICO**. Rio de Janeiro. Ciência Moderna, 2005

SERVIÇO DE DIRETÓRIO: WIKIPEDIA, a enciclopédia livre. Disponível em: <http://pt.wikipedia.org/wiki/Servi%C3%A7o_de_diret%C3%B3rio>. Acesso em 12 fev. 2009.

UBUNTU. Disponível em: <[http:// www.ubuntu-br.org/](http://www.ubuntu-br.org/)>. Acesso em 15 fev. 2009.

VESPERMAN, J. **Autenticação e o squid**. Disponível em: <<http://br.geocities.com/cesarakg/AuthenticationAndSquid.html>>. Acesso em 06 mar. 2009.