UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

CAMPUS LUIZ MENEGHEL

ALEX DE MELLO

AUTENTICAÇÃO CENTRALIZADA DE USUÁRIOS EM UMA REDE LOCAL UTILIZANDO FREE-RADIUS E OPENLDAP EM SERVIDORES LINUX

Bandeirantes

2009

UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

CAMPUS FACULDADES LUIZ MENEGHEL

ALEX DE MELLO

AUTENTICAÇÃO CENTRALIZADA DE USUÁRIOS EM UMA REDE LOCAL UTILIZANDO FREE-RADIUS E OPENLDAP EM SERVIDORES LINUX

Bandeirantes

2009

ALEX DE MELLO

AUTENTICAÇÃO CENTRALIZADA DE USUÁRIOS EM UMA REDE LOCAL UTILIZANDO FREE-RADIUS E OPENLDAP EM SERVIDORES LINUX

Trabalho de Conclusão de Curso submetido à Universidade Estadual do Norte do Paraná - Campus Luiz Meneghel, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Luiz Fernando Legore do Nascimento

Bandeirantes

2009

ALEX DE MELLO

AUTENTICAÇÃO CENTRALIZADA DE USUÁRIOS EM UMA REDE LOCAL UTILIZANDO FREE-RADIUS E OPENLDAP EM SERVIDORES LINUX

Trabalho de Conclusão de Curso submetido à Universidade Estadual do Norte do Paraná - Campus Luiz Meneghel, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

COMISSÃO EXAMINADORA

_____

Prof. Mestre Ailton Sergio Bonifacio

_____

Prof. Mestre Glauco Carlos Silva

_____

Prof. Luiz Fernando L. do Nascimento

Bandeirantes, 05 de Dezembro de 2009

Aos amigos de todas as horas
À família de todos os segundos
A Deus de todo o ser.

AGRADECIMENTO

A minha esposa Mônica que acreditou em mim durante todo o tempo.
A minha família que sempre me deu forças para prosseguir.
Ao Prof. Orientador, que sem ele o projeto não teria tido sucesso.
A todos os membros da banca e aos professores que me a apoiaram.
Aos meus companheiros de trabalho que me apoiaram.
Ao meu grande amigo Ederson que me forneceu os equipamentos necessários.

A adversidade desperta em nós capacidades que,
em circunstâncias favoráveis,
teriam ficado adormecidas.
(Horácio)

RESUMO

Em uma rede de computadores a informação e a segurança devem ser itens primordiais, assim faz-se necessário implantar soluções que consigam unir as duas funções em um único serviço. Com isso acontece o que se chama de centralização de informação, neste trabalho será descrito a forma de centralizar os serviços existentes em uma rede.

Com o intuito de prover maior segurança para os diversos serviços dispostos na rede, com a utilização de *softwares* de centralização pretende-se apresentar a melhor opção para uma implementação de um servidor de diretórios, uma vez que o ambiente e a arquitetura onde se encontra a rede necessitam de autenticação tanto dos usuários quanto dos professores para uma melhor distribuição dos serviços.

Palavra chave → LDAP, FreeRadius, autenticação.

ABSTRACT

In a computer network and information security items should be paramount, so it is necessary to deploy solutions that can match both functions into a single service. With which this happens is called the centralization of information, this paper will describe how centralize services within a network.

In order to provide greater security for the various services on the network ready, with the use of centralized software is intended to present the best option for an implementation of a directory server, since the environment and architecture where the network require authentication of both users and teachers for a better distribution of services.

LISTA DE ABREVIATURAS

AP	Access Point
ASIC	Application Specific Integrated Circuits
BD	Banco de Dados
CSMA	Carrier Sense Multiple Access with Collision Avoidance
DAP	Directory Acceess Protocol
DSSS	Direct Sequence Spread Spectrum
FDDI	Fiber Distributed Data Interface
FHSS	Frequency Hopping Spread Spectrum
FTP	File Transfer Protocol
GHz	Gigahertz
IEEE	Institute of Electrical and Eletronics Engineers
ISL	Inter-Switch Link
LAN	Local Area Network
LDAP	Lightweight Directory Acceess Protocol
MAC	Media Access Control
Mbps	Megabit por Segundo
OSI	Open Systems Interconnection
RADIUS	Remote Authentication Dial In User Service
RNP	Rede Nacional de Pesquisa
SQL	Structured Query Language
SSL	Secure Sockets Layer
TI	Tecnologia da informação
TCP/IP	Transmission Control Protocol/Internet Protocol
UTP	Unshielded Twisted Pair
VLAN	Virtual local area network
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A grande demanda por informação e a infinidade de recursos existentes para buscá-las atualmente faz com que a mesma seja centralizada de forma que fique cada vez mais acessível para todos. Para garantir que a informação se torne acessível para quem as procure esta tende a ficar centralizada e necessita se distribuída pela rede, o objetivo central de uma rede de comunicação de dados é permitir uma partilha eficiente, de forma interativa de recursos de um modo mais transparente possível ao utilizador, independentemente da localização física dos recursos [Gouveia 2005].

Assim garante que a informação se torne acessível para todos que necessitem, possibilitam grande demanda de serviços de acordo com as necessidades de seus utilizadores. Devido à grande quantidade de informação de serviços que estes servidores podem fornecer, é preciso implantar em seus sistemas maneiras de gerenciar e facilitar todo o processo de manipulação de dados e gerenciamento dos usuários a fim de proporcionar melhor segurança para ambos. O trabalho a seguir apresenta um estudo de duas ferramentas de autenticação de usuários (*free-radius* e *OpenLdap)* para redes locais, utilizando servidores *Linux* em ambiente acadêmico.

Um servidor de dados fornece serviços de arquivamento físico e distribuição de Dados pela rede, via FTP ou mídia removível, agregam diversos serviços como, por exemplo, o compartilhamento de arquivos, serviços de *e-mail*, servidores *Proxy* [Barros 2003].

Disponibilizar serviços em uma rede de computadores pode gerar inúmeros problemas quando não se dá ênfase na segurança, para garantir que pessoas não autorizadas tenham acesso a áreas reservadas, é necessário que exista pelo menos registros de acesso a essa informação, uma autenticação.

> Autenticação é o ato de estabelecer ou confirmar algo como autêntico, isto é, a reivindicação de ser ou ter feito algo ser verdadeiro. Autenticar um objeto pode significar confirmar a sua procedência ou autenticar uma pessoa consiste em verificar sua identidade dependendo de um ou mais fatores que a caracterizam [Loi 2007].

Mas como administrar o acesso dos usuários aos diversos serviços da rede, gerando autenticação de acesso, seria necessário para cada usuário uma senha de acesso para utilizar cada serviço, ou seja, um usuário disponibilizará de inúmeros cadastros diferentes ou na pior das hipóteses, para todos os serviços ele terá um mesmo nome e senha, ainda assim terá que fazer autenticação em cada serviço que necessitar usar, por outro lado os administradores de sistema teriam que gerar em cada serviço o cadastro destes usuários um a um, isto não só é difícil como impossível de gerenciar.

A centralização do serviço de autenticação dos usuários é necessário tanto para garantir uma melhor manipulação dos dados como proporcionar serviços de segurança de acesso à diretórios e usuários.

Este trabalho apresentada um estudo dos *softwares* utilizados para a centralização do serviço de diretório OpenLDAP e Free-radius implementados em servidores *Linux* de forma a permitir uma melhor administração dos acesso a informação. Analisar, demonstrar e implementar o uso desses *softwares* para autenticação dos serviços de compartilhamento de arquivos (Samba), serviços de *Proxy/Cache* (*Squid*) e serviços de transferência remota de arquivos FTP (*File Transfer Protoco*l) é o objetivo deste trabalho.

2 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em da seguinte forma, a começar pelo capitulo 1º uma introdução onde é apresentada uma previa do que será descrito no decorrer do trabalho, no capitulo 3º é abordado os objetivos do trabalho, onde é apresentado o foco geral de todo o trabalho, no capitulo 4º é apresentado os objetivos específicos, é nesta parte que será designado o caminho que será seguido dentro do trabalho.

No 5º capitulo apresento as justificativas de se realizar ou desenvolver o trabalho neste ponto é expresso o porquê de se desenvolver o trabalho, apresentando as necessidades à que o trabalho se designa, no capitulo 6º apresento os materiais e métodos utilizados para a confecção de to o trabalho, abordando os equipamentos *softwares* e a formas utilizadas para o desenvolvimento.

No 7º capitulo, é onde se encontra a revisão bibliográfica, neste ponto é abordado todos os aspectos, assuntos e informações que o trabalho aborda, no capitulo 8º é apresentado o estudo de caso, mostrando todo o processo de implantação das ferramentas necessárias e suas devidas configurações para um bom funcionamento do servidor de diretórios, no 9º capitulo é onde esta descrito a conclusão geral do trabalho.

No 10º apresenta-se a revisão bibliográfica e por fim o 11º capitulo traz as anexos onde estão contidos os relatórios sobre a métrica obtida no trabalho sobre o funcionamento das ferramentas OpenLDAP e FreeRadius.

3 OBJETIVO

Estudar as ferramentas de autenticação centralizada para serviços de FTP (*File Transfer Protocol*), Samba (Compartilhamento de dados) e *Proxy* (*OpenLdap* e Free-Radius, para serem utilizadas em servidores *Linux* para atender as necessidades de uma instituição pública de nível superior.

4 OBJETIVO ESPECÍFICO

Implantar os pacotes de instalação dos serviços propostos para o estudo, utilizando plataforma *Linux*, de forma a proporcionar possibilidade de estudo das ferramentas OpenLdap e Free-Radius, gerando assim um relatório sobre a métrica entre as duas ferramentas.

Para gerar os resultados será analisado o nível de dificuldade para a implantação dos pacotes e a quantidade dos mesmos e também qual das ferramentas é mais flexível para suportar outros aplicativos.

5 JUSTIFICATIVA

Em uma rede, seja ela acadêmica ou empresarial, a segurança e o controle sobre os dados e usuários que a utilizam é algo primordial, para garantir que este controle se torne possível, é necessário centralizar os serviços de autenticação existentes nos diferentes serviços presentes na rede, assim o controle fica centralizado e a manipulação e gerencia dos dados se torna mais consistente.

A rede em foco apresenta um sistema de autenticação dividido entre os serviços da rede, os processos ficam dispersos entre os servidores, quando surge a necessidade de alterações nos dados de acesso dos usuários a quantidade de informação a ser manipulada é muito grande, outro ponto importante que pode ser observado em uma rede sem autenticação é que o administrador desta não consegue ter o controle sobre quem acessa a rede em serviços como internet nos servidores *Squid* (*Proxy*) onde o acesso é aberto a falta de autenticação se torna um problema.

Para solucionar estes problemas de autenticação de serviços é necessário implantar um processo de autenticação, para isso, é estudado as ferramentas de autenticação OpenLdap e RADIUS afim de suprir as necessidade da rede da instituição garantindo assim uma melhor administração dos serviços.

6 MATERIAIS E MÉTODOS

Para a construção do servidor de dados que foi utilizada para a realização do trabalho foi necessário montar um computador pessoal para a instalação do sistema *Linux* versão *Slackware* 12.2.0 com *kernel* 2.6.27.7-smp.

Este procedimento foi necessário devido à falta de experiência em manipulação de uma rede de grande porte como exemplo a rede acadêmica da instituição de ensino UENP, e a não disponibilidade da mesma para a realização de testes de implementação, uma vez que poderia ocasionar problemas nas instalações com uma manipulação inadequada.

A máquina onde foi instalado o sistema é composta de uma placa mãe ASUS modelo p4v800 soquete 478, processador Pentiun 2.4, memória de 512MB DDRI, um HD de 40gb para a instalação do sistema e outro HD de 40gb para servir de armazenamento dos arquivos do servidor.

Para o procedimento de acesso de um terminal ao servidor de diretórios foi utilizado um notebook Acer, processador celeron 1.6, HD de 120gb e 512Mb de memória, com sistema operacional *Linux Ubuntu* v9.1, outra máquina modelo PC com um processador modelo Pentium III com 256MB de RAM e um HD de 40gb, com sistema operacional *Linux Ubuntu* v9.1.

Pra a apresentação da métrica entre as ferramentas será considerado o grau de dificuldade encontrado para a implantação e configuração das ferramentas.

7 REVISÃO BIBLIOGRÁFICA

7.1. SEGURANÇA DA INFORMAÇÃO

Quando houve o surgimento das redes de computadores as máquinas que as compunham e seus utilizadores não utilizavam muitos recursos e nem poderiam contar com tantos a disposição, então os computadores da rede eram usados somente para a troca de arquivos fazendo compartilhamentos com outros usuários e também, utilizados por pesquisadores, principalmente para a troca de mensagens eletrônicas, assim tendo uma utilização não tão expressiva a necessidade e a preocupação com a segurança não era levada tanto em consideração.

Com o passar do tempo e o desenvolvimento de inúmeros recursos que necessitam da utilização de redes de computadores, também houve o surgimento de milhões de usuários que necessitam ou utilizam a rede para simples diversão ou passatempo, seja fazendo transações bancárias, comprando produtos via internet ou fazendo declaração de imposto de renda.

Devido essa grande transição de arquivos pela rede, a necessidade de se implantar um sistema de segurança torna-se indispensável, tanto para prover a

segurança dos dados como para inibir a ação de pessoas mal intencionadas que se utilizam das redes para promover verdadeiras “anarquias”, na maior parte elas querem obter algum benefício chamar a atenção ou prejudicar alguém [Tanenbaum 2003].

Os problemas das redes podem ser divididos em quatro áreas interligadas: sigilo, autenticação, não-repúdio e controle de integridade. Em se tratando do sigilo é ele que trata de manter informações confidenciais longe de pessoas que não são autorizadas, quando se pensa em segurança em redes, confidencialidade de dados é a primeira coisa que vem em mente.

Já a autenticação trata de revelar com quem você está se comunicando antes de revelar dados sigilosos. Quando se trata da comprovação de que a pessoa que pediu determinado serviço ou informação é realmente a autora do pedido, trata-se do não-repúdio, ou seja, se alguém pede uma mercadoria por um preço “x” e no ato do recebimento que pagar um preço “y” então que entra o não-repúdio onde ele não poderá negar que é o autor do pedido da mercadoria.

O controle de integridade requer que o acesso à informação possa ser controlado pela rede que contenha a informação. Há vezes em que se quer dar acesso somente de leitura a um determinado arquivo, assim é necessário garantir que o leitor não possa, de modo algum, alterar o conteúdo do que está sendo exibido [Veríssimo 2002].

7.2. EQUIPAMENTOS DE CONTROLE DE SEGURANÇA

7.2.1. SWITCHES

Nas redes ligadas por cabos é comum a utilização de *hubs* ou *switches* para prover a comunicação entre os computadores existentes na rede, entre estes equipamentos existem diferenças no gerenciamento da informação que trafegam por

eles, os *hubs* são repetidores de sinal servem para interligar redes sem nenhum gerenciamento, seu funcionamento é simples, todos os sinais que chegam a eles são enviados a todas as portas do equipamento e todos os computadores ligados a ele recebem a mesma informação, geralmente são utilizados em redes caseiras para ligar vários computadores à internet [Figueiredo 2007].

Nos *switches* existem sistemas que gerenciam o tráfego de informação, conceitualmente são considerados *bridges* multi-portas. Tecnicamente *bridging* é uma função da camada 2 do modelo OSI, assim todos os padrões atuais de rede podem ser conectados através de *switches*, exemplo *Ethernet*, *Token Ring* e FDDI.

Devido o sistema existente nos *switches* eles têm a capacidade de aprender as informações de endereço das estações que estão conectadas em cada um dos segmentos de suas portas, com isso eles examinam o tráfego de entrada, deduz o endereço MAC de todas as estações conectadas a cada porta e usa estas informações para construir uma tabela de endereçamento local. Muitos *switches* usam uma arquitetura baseada em ASIC (*Application Specific Integrated Circuits*), ao invés dos microprocessadores tradicionais, permitindo com isto uma maior velocidade na comutação, e um barateamento do custo [Figueiredo 2007].

Basicamente os *switches* são semelhantes a pontes, pois ambos baseiam-se no roteamento de endereços de quadros, a principal diferença é que *switches* são usados com mais freqüência para conectar computadores individuais, já as pontes servem para ligar LANs distintas, assim os *switches* devem encaminhar ativamente os dados, já que em cada porta é ligado um único equipamento.

Devido o sistema de envio de quadros dos *switches* eles necessitam de muito mais espaço para placas de extensão do que as pontes, cada placa de extensão fornece espaço de *buffer* pra os quadros que chegam a suas portas, como cada porta é seu próprio domínio de colisão, os *switches* nunca perdem quadros devido a colisões, por isso a necessidade de espaço, pois os quadros chegam com velocidade maior que aquelas que podem ser retransmitidos e se houver falta de espaço os quadros poderão ser descartados [Tanenbaum 2003].

7.2.2. FIREWALLS

> “Os firewalls são apenas uma adaptação moderna de uma antiga forma de segurança medieval: cavar um fosso profundo em torno do castelo. Esse recurso forçava todos aqueles que quisessem entrar ou sair do castelo a passar por uma única ponte levadiça, onde poderiam ser revistados por guardas” [Tanenbaum 2003].

Firewall é um dispositivo *hardware* ou *software* utilizado para tentar impedir uma invasão a uma rede ou computador, limitar os danos que podem ser causados por esta e isolar uma porção de dados sensíveis que devem ser protegidos do resto do mundo. Ele não deve ser considerado o único meio de proteger a rede, outros mecanismos existentes devem ser utilizados em conjunto com o *firewall*. Em geral, o *firewall* é um meio de reforçar as políticas de segurança existentes em uma rede, geralmente pertence a uma política maior, onde é uma das partes responsáveis pela segurança. O *firewall* baseia-se nas regras à que foi submetido para efetuar o controle de acesso à rede.

Ele deve ser instalado na fronteira das redes para limitar o acesso entre elas, é normalmente instalado com o objetivo de proteger a rede interna dos problemas originados pelos males vindos da Internet, pelo fato da mesma ser uma rede pública que não é confiável. Também pode ser utilizado para separar redes internas que devem ter seu acesso limitado, por exemplo, a rede do departamento financeiro de uma empresa que deve ser separado dos demais computadores de outros departamentos [Pontes 2007].

A recomendação é que ele seja instalado em um único equipamento, desta maneira os serviços serão integrados e o mesmo será menos vulnerável a ataques. Em geral os *firewalls* compartilham algumas características: Gerenciar e controlar o tráfego de rede, autenticar o acesso, agir como um intermediário, proteger recursos, gravar e relatar eventos.

Deve se levar em conta que um *firewall* sozinho, ou seja, um *software* de *firewall* instalado em um computador, porém se não configurado adequadamente e sem um banco de regras bem definido, somente ira consumir recursos de *hardware* e *software* da maquina e não proporcionará defesa alguma [Ribeiro 2004].

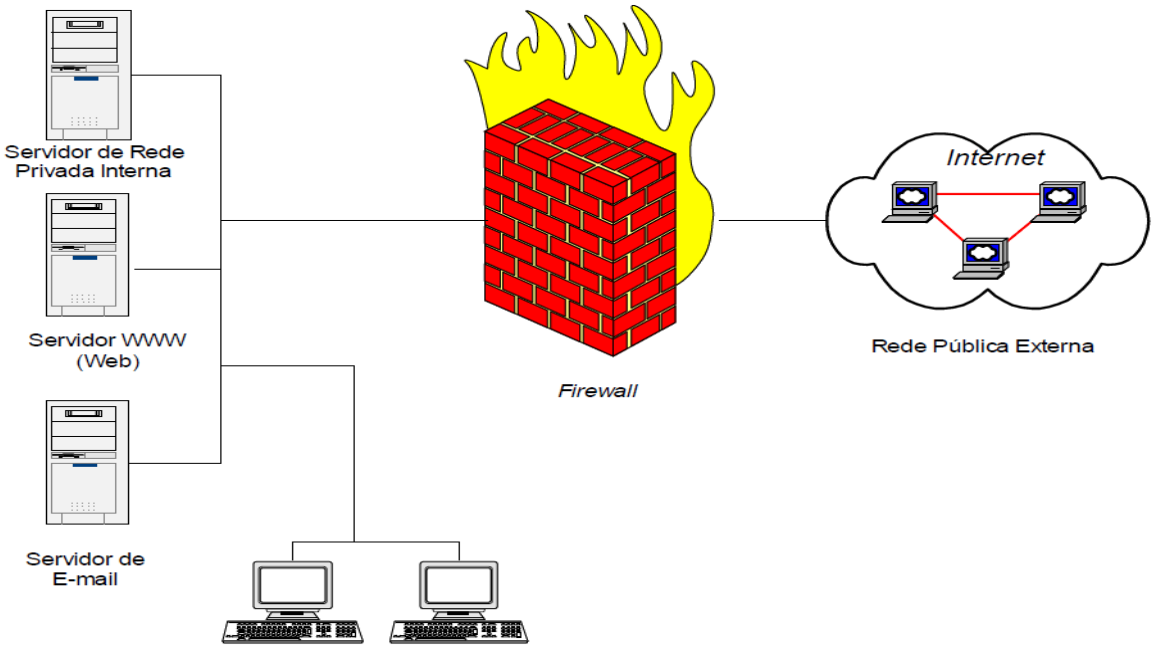

Ilustração 1 - Exemplo de um firewall [Ribeiro 2004].

7.2.3. VLANs

Quando foram criadas as primeiras redes de computadores os edifícios e casas não estavam preparados para suportar um novo sistema de cabos em seus conduítes, o projeto original não contava com novos cabos de transmissão e por isso não foram implantados, assim grossos cabos amarelos se espremiam em seus conduítes. Por onde eles pudessem ser passados todos os computadores eram ligados. Todos os cabos eram conectados ao mesmo *backbone* central ou um *hub*, ninguém parava para pensar quais computadores pertenciam a cada LAN, não importava qual setor ou departamento este ou aquele pertencia, todos estavam conectados entre si [Tanenbaum 2003].

Com a evolução, novas tecnologias e novos padrões foram lançados, os grossos cabos foram substituídos por cabos UTP, novos equipamentos possibilitaram a conexão de uma enorme quantidade de computadores em *switches* e *hubs* com cada vez mais tecnologia empregada. Atualmente é comum encontrar redes com vários *switches* e roteadores, os *switches* atuais possuem implementações que possibilitam a criação de redes virtuais dentro de um mesmo equipamento, assim podem ser criadas diversas redes sem a necessidade de aquisição de novos equipamentos.

Uma VLAN é, basicamente, uma coleção de nós que são agrupados em um único domínio broadcast, baseado em outra coisa que não a localização física. Em uma rede comum, tudo que estiver de um mesmo lado do roteador faz parte do mesmo domínio *broadcast*. Um switch com uma VLAN implementada tem múltiplos domínios *broadcast* e funciona de maneira semelhante a um roteador [Figueiredo 2007].

A tecnologia *multicast* é a base de um serviço de rede no qual um único fluxo de dados, proveniente de uma determinada fonte, pode ser enviado simultaneamente para diversos receptores interessados. Ao longo do trajeto, a própria infra-estrutura de rede replica o fluxo de dados, quando necessário, para todos os receptores que registraram interesse em receber estes dados. Uma mensagem de broadcast é uma mensagem destinada a todos os nós da rede de uma só vez.

A utilização de VLANs dentro de uma rede [Figueiredo 2007] pode trazer algumas vantagens como, por exemplo:

Segurança. Separa os sistemas que contêm dados sigilosos do resto da rede, funcionando como um *firewall* onde protege os terminais de uma rede reduzindo a possibilidade de acesso não autorizado.

Projetos/aplicativos especiais. As tarefas de gerenciar um projeto ou trabalhar com um aplicativo compartilhado com vários outros usuários da rede

podem ser simplificados pelo uso de uma VLAN que agrupa todos os nós necessários.

Desempenho/Largura de banda. Um monitoramento cuidadoso da utilização da rede permite que o administrador crie VLANs que reduzam o número de saltos entre os roteadores, esta redução acaba gerando maior velocidade de transmissão e aumentem a largura de banda aparente para os usuários da rede.

Broadcasts/Fluxo de tráfego. A característica principal de uma VLAN é que ela não permite que o tráfego *broadcast* chegue aos nós que não fazem parte da VLAN. Isso ajuda a reduzir o tráfego de *broadcast*. As listas de acesso permitem que o administrador da rede controle quem esteja utilizando-se do tráfego da rede.

Departamentos/Tipos específicos de cargos. Pode-se configurar VLANs para os departamentos que utilizam muito a Internet ou para usuários que compartilham dados específicos e ainda VLANs que conectam categorias específicas de empregados que se encontrem em departamentos diferentes.

7.2.4. PROTOCOLO 802.1q

O protocolo 802.1q foi criadas para reduzir as redes que utilizam uma grande largura de banda que conseqüentemente se tornam lentas. É conveniente acabar com estas enormes LANs e criar unidades menores, mais gerenciáveis para as redes, chamadas VLANs.

Para resolver este problema, a IEEE lançou o padrão 802.1q, que foi desenvolvido como uma parte do IEEE 802.1, este padrão possibilita que os administradores de redes dividam estas LANs em fragmentos menores. Os fragmentos podem acabar com os broadcasts e *multicast* e assim aumentar muito a largura de banda. Fragmentar uma LAN que utilizam este protocolo prevê também um ambiente muito mais seguro entre os segmentos de rede interna [Souza 2006].

O padrão 802.1q introduz a técnica conhecida como *tagging* e o conceito de VLAN nativa. Permite expandir as VLANs através de vários comutadores. As tramas são etiquetadas com uma etiqueta que contém entre outra informação um VLAN Id. Desta forma, as tramas associadas a uma VLAN podem passar de uns comutadores para outros, sem que a informação relativa às VLANs se perca [Lima 2006].

7.2.5. WLANs

As redes locais sem fios são normalmente conhecidas como WLAN, são baseadas e construídas de acordo com o padrão 802.11. Este padrão surgido quase na mesma época que os *notebooks,* veio para garantir uma maior mobilidade para os usuários de computadores móveis [Tanenbaum 2003].

Atualmente as redes sem fio estão sendo adotadas por empresas e pessoas que necessitam de mais mobilidade, é comum ver pequenas e grandes empresas utilizando redes sem fio, também é comum a existência de redes sem fio em domicílios e lugares públicos como *shoppings* e praças.

Como se trata de um meio de comunicação sem fio onde existe um sinal sendo transmitido pelo ar na forma de ondas de radio, esta rede acaba sendo mais vulnerável que outras redes cabeadas, assim os padrões de segurança para este tipo de rede devem ser mais sofisticados.

O medo relacionado com insegurança das redes sem fios tem limitado sua adesão pelas empresas, pois em uma WLAN não existe o conceito de rede privada, qualquer individuo com um computador ou telefone equipado com um adaptador de rede sem fio pode de conectar a rede, desde que tenha sinal e este não tenha um forte sistema de segurança, o que torna possível o acesso as informações de uma empresa por pessoas desautorizadas, uma preocupação real e mesmo assim, ainda é possível encontrar diversas redes se fio funcionando sem proteção alguma [Oliveira 2002]

O 802.11 é um padrão projetado pelo IEEE (*Institute of Electrical and Eletronics Engineers*). E é um dos mais recentes padrões de redes sem fio. Ele tem as seguintes características [Veríssimo 2002].

- Teoricamente, pode alcançar até 11 Mbps de velocidade, mas estudos dizem que ele aceita até 8Mbps.

- Tem conexão *peer-to-peer* ou baseada num ponto fixo de acesso.

- Opera na freqüência de 2,4GHz.

- Funciona com os dois métodos de espalhamento na freqüência: O FHSS (*Frequency Hopping Spread Spectrum*) e o DSSS (*Direct Sequence Spread Spectrum*). Além disso, funciona com infravermelho, que não utiliza nenhum desses dois métodos.

- Utiliza o protocolo de acesso CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*).

- Utiliza o WEP para autenticação e privacidade

7.2.6. VPN

A necessidade de se utilizar redes pública não confiáveis para o envio e recebimento de dados sigilosos foi o grande trampolim para o surgimento do conceito de VPN (*Virtual Private Network*), que transforma redes públicas em redes privadas virtuais.

As redes públicas são consideradas inconfiáveis, devido o fato de que, os dados que trafegam por ela correm o risco de intercepção e captura, por outro lado estas normalmente tendem a ter um custo inferior em relação as redes proprietárias, pois não se necessita de uma nova infra-estrutura com cabeamento e equipamentos.

Mas não é só para transformar a rede de internet em uma rede segura que o VPN pode ser utilizado, existem muitos aplicativos que são desenvolvidos para operar em redes privadas e habitualmente não fornecem segurança quando utilizada em redes públicas, para modificar estes aplicativos demanda-se tempo, mas aplicando o conceito do VPN sobre eles pressupõe que não haja necessidade de modificações nos sistemas [Silva 2002].

A idéia da VPN é criar um túnel seguro entre os gateways para proteger os dados privados enquanto eles estão navegando pela internet, ou seja, enquanto eles estão trafegando por redes não confiáveis. Para que seu funcionamento seja efetivo a VPN dever prover um conjunto de funções que garantam confidencialidade, integridade e autenticidade [Veríssimo 2002].

Confidencialidade: Tendo em vista que os dados que trafeguem em uma rede pública podem ser facilmente capturados, torna-se necessário que os dados sejam absolutamente privados garantindo que mesmo que os dados sejam capturados não possam ser entendidos.

Integridade: Considerando que algum dado seja capturado na rede, a integridade deve garantir que este não seja adulterado e re-encaminhado, de tal forma que qualquer tentativa neste sentido não tenha sucesso, assim somente dados validos possam ser recebidos por aplicações suportados pelo VPN.

Autenticidade: Garante que somente equipamentos que tenham sido cadastrados em ema mesma VPN possam trocar dados entre si, ou seja, um elemento da VPN somente reconhecera dados originários de uma mesma VPN ou de VPN autorizadas.

7.3. SERVIÇOS DE AUTENTICAÇÃO

7.3.1. OPENLDAP

O LDAP (*Lightweight Directory Acceess Protocol*) trata se de um protocolo utilizado para pesquisar diretórios de uma rede TCP/IP, é composto de um conjunto de regras que controla a comunicação entre serviços de diretórios e seus clientes, devido a facilidade e a eficiência dos seus recursos, diversos programas já vem implementados para o suporte ao LDAP [Silva 2002].

Inicialmente foi desenvolvido para servir como um cliente para o protocolo X-500, este protocolo é quem define o acesso ao diretório para os clientes utilizarem quando conectados aos servidores de diretório (DAP) Protocolo de Acesso a Diretório. O grande problema é que o DAP roda sobre o protocolo OSI completo e precisa de muita capacidade computacional para ser executado [Steim 2007].

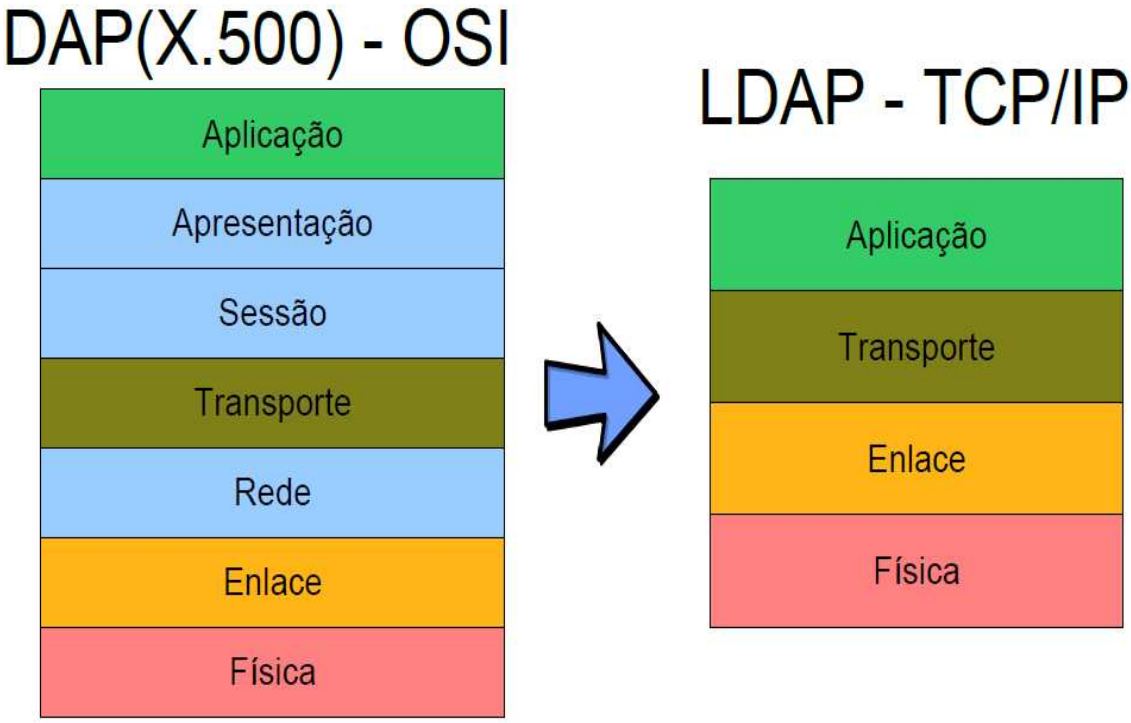

Ilustração 2 - DAP e LDAP na camada OSI e TCP/IP [Steim 2007].

> “O modelo OSI é chamado de modelo de referencia ISO OSI (*Open Systems Interconnection*), pois ele trata de interconexão de sistemas abertos – ou seja, sistemas que estão abertos à comunicação com outros sistemas, o modelo OSI tem sete camadas, Física, Enlace de dados, Rede, Transporte, Sessão, Apresentação, Aplicação, apresentadas na respectiva ordem”. [Tanenbaum 2003].

O LDAP é uma forma mais simplificada do DAP, consegue fornecer a maioria de suas funções com uma exigência muito menor, pois trabalha sobre a camada TCP. O protocolo de comunicação TCP-IP surgiu quando houve a criação das redes de radio [Tanenbaum 2003], o que forçou a criação de uma nova arquitetura de referência devido a necessidade de conectar varias redes de maneira uniforme.

A razão da flexibilidade e dinamismo do LDAP é o fato de sua estrutura ser em forma de hierarquia. A árvore hierárquica contém um elemento que é a raiz por onde as pesquisas serão iniciadas, a partir daí a pesquisa continua pelos nós-filhos até encontrar o elemento pesquisado. Tanto a raiz quanto os nós-filhos da arvore são diretórios. Em cada diretório pode haver outros diretórios ou elementos que são chamados de entradas, por sua vez cada entrada pode conter um ou mais atributos, e cada atributo pode conter um ou mais valor associado, todos mantendo uma forma de dados predefinida [Trigo 2003].

Uma característica herdada do padrão X.500 foi o uso de nemônicos para definir nomes de atributos de diretórios e entradas. Para diretórios, por exemplo, você encontrará os seguintes atributos mostrados na tabela a seguir:

Atributo	Descrição
c	Para diretórios que representam países (do inglês *country*).
o	Para o nome da empresa (do inglês *organization*).
ou	Para departamento (do inglês *organizational unit*).
No caso das entradas você encontrará, por exemplo:	
Atributo	Descrição
cn	Como atributo de nome (do inglês *common name*).
uid	Para identidade de usuário (do inglês *user ID*).
gn	Para o nome próprio de uma pessoa (do inglês *given name*).
sn	Para o sobrenome de uma pessoa (do inglês *surname*).

Ilustração 3 - Atributos de diretórios [Trigo 2003].

O protocolo OpenLDAP consegue se interagir com diversos serviços, sejam eles de dados e-mail ou outros, a imagem a seguir mostra a integração dos diversos serviços pelo OpenLDAP.

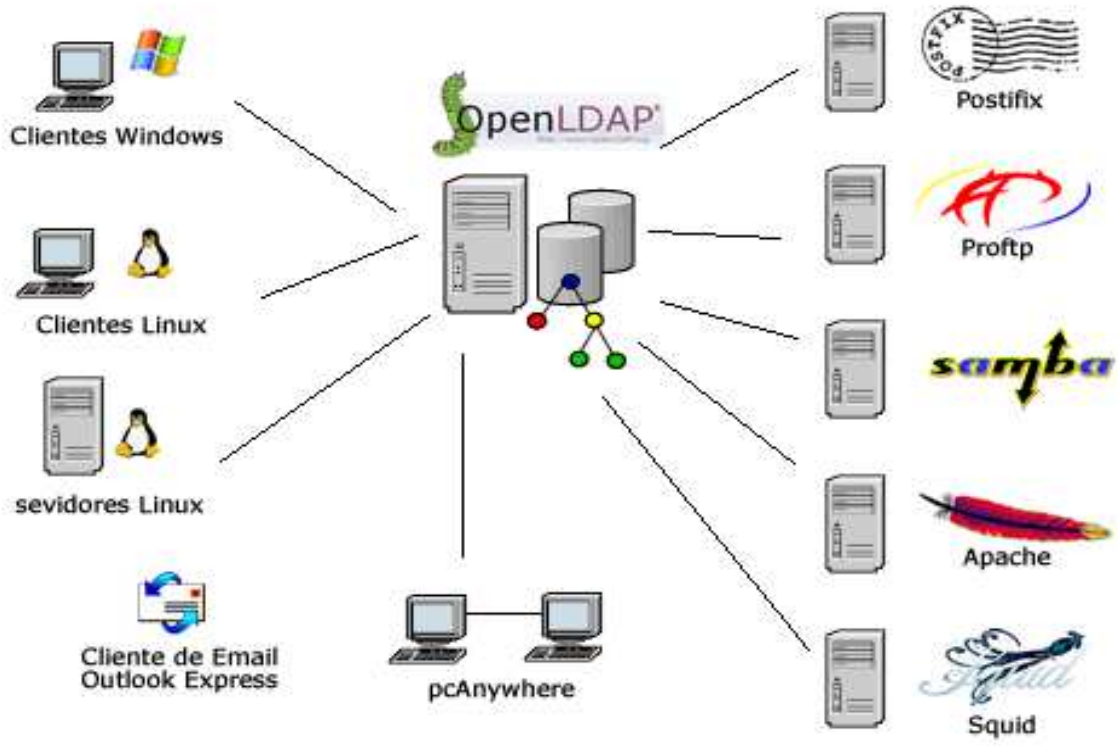

Ilustração 4 - Integração de serviços no OpenLDAP [Valcy 2006].

7.3.1.1. BANCO DE DADOS BERKELEY

O OpenLdap poder fazer integração com diversos banco de dados como por exemplo, MySQL, PostgreSQL entre outros, por padrão o OpenLdap já vem implementado para rodar com o Berkeley DB, um banco de dados produzido pela Oracle, de código aberto, sua principal diferença em relação aos outros tipos de banco de dados é o fato dele não ser relacional, usas apenas uma tabela de dados, as bases de dados BERKELEY contêm registros, e cada um deles representa uma única entrada neste BD Base de Dados. Cada registro é composto por um par de informações (chave, dados), e estes podem ser compostos por qualquer tipo de dados, isto o torna mais rápido para as buscas do OpenLdap [Oracle 2009].

7.3.1.2. BANCO DE DADOS MySQL

MySQL é um sistema gerenciador de banco de dados relacional que utiliza a linguagem padrão SQL (*Structured Query Language*), e é largamente utilizado em aplicações, principalmente para a Internet. É o mais popular entre os bancos de dados com código-fonte aberto. Há mais de cinco milhões de instalações do MySQL no mundo todo, inclusive em sites com alto volume de dados e de tráfego, como *Associated Press*, Google, NASA, *Sabre Holdings e Suzuki*.

O MySQL á uma alternativa atrativa porque, mesmo possuindo uma tecnologia complexa de banco de dados, seu custo é bastante baixo. Tem como destaque suas características de velocidade, escalabilidade e confiabilidade, o que vem fazendo com que ele seja adotado por departamentos de TI (Tecnologia da Informação), desenvolvedores Web e vendedores de pacotes de *softwares* [Pontes 2007].

7.3.2. FREE RADIUS

O RADIUS (*Remote Authentication Dial In User Service*) é um protocolo empregado em redes que necessitam disponibilizar acesso a rede de computadores e internet com Autenticação, Autorização e Contabilização dos seus utilizadores.

Com uma grande simplicidade na utilização, eficiência no serviço a que foi implementado e facilidade de implementação de seus recursos, o RADIUS foi inicialmente desenvolvido para uso em serviços de acesso a internet discado é atualmente é também implementado em pontos de acesso sem fio e outros tipos de dispositivos que permitem acesso autenticado a redes de computadores [Barros 2003].

Devido à existência de um grande número de sistemas independentes que acabam inviabilizando a administração descentralizada, o RADIUS foi idealizado para centralizar as atividades de Autenticação, Autorização e Contabilização.

7.3.3. DIRETÓRIOS

O termo diretório na literatura especializada tem vários significados, dependendo do contexto. No contexto de sistemas de arquivos ele possui um significado, no contexto de redes e ambientes distribuídos outro e no contexto de banco de dados um terceiro significado.

Considerando inicialmente o significado elementar de diretório que é lista. Lista nada mais é do que um depósito de informações. Desta forma fica mais fácil entender porque diretório é usado nesses contextos.

No contexto sistema de arquivos nada mais é do que um arquivo especial que direciona o sistema para um local que contem a lista dos arquivos pertencentes a esse diretório.

No contexto de redes e ambientes distribuídos, diretório é uma lista que contem informações, referentes aos acessos realizados por seus utilizadores, quase todos os serviços de rede exigem algum tipo de autenticação de acesso, obrigando desta forma que os serviços mantêm um diretório de usuários (uma lista de usuários).

Já no contexto de banco de dados é muito natural já que lista é na verdade um depósito de informações, ou seja, onde os dados são armazenados.

O termo diretório é aplicado a tudo aquilo que indique uma direção, é como um banco de dados, mas seu conteúdo é composto de mais informações descritivas, e sua organização é baseada em forma de arvore mantendo uma hierarquia dos dados para prover uma facilidade maior à pesquisa e ou recuperação das informações.

A diferença mais clara entre um banco de dados e um diretório é que um diretório tende a ser otimizado apenas para a leitura podendo ocasionalmente ser escrito, portanto é mais lido do que escrito, o banco de dados mantêm a mesma eficiência entre a leitura e escrita, com isso os diretórios normalmente não são usados em transações mais complexas [Silva 2002].

Os diretórios podem ser comparados às listas telefônicas ou um mapa de um shopping Center, a comparação é dada pelo fato das listas serem impressas anualmente, mas se pararmos para pensar quase todos os dias é preciso utilizar a lista para encontrar um telefone, então ele se torna, mas lida do que escrita isto faz com que seja comparada a um diretório, o mesmo se da com o mapa do *shopping* Center onde ele só é reestruturado quando são instaladas novas lojas.

Exemplos de diretórios:

- Lista Telefônica
- Guia de TV
- Livro de endereços
- DNS (*Domain Name System*)

7.3.3.1. SERVIÇOS DE DIRETÓRIOS

Um serviço de diretórios é uma base de dados especializada, com o propósito de prover o acesso rápido aos dados de maneira padronizada. Essa base tem o objetivo de organizar e facilitar eventuais alterações e inclusões de dados de configuração nas contas individuais, ou em grupos, dos usuários de uma determinada rede [Kanies 2001].

7.4. SERVIÇOS DE SEGURANÇA NA AUTENTICAÇÃO

7.4.1. SSL

O SSL (*Secure Sockets Layer*) é um protocolo que funciona em uma camada acima do TCP e abaixo de protocolos de alto nível em camada de aplicação, como o HTTP ou o OpenLdap. O objetivo do SSL é autenticar cliente e servidor com a finalidade de abrir um canal de comunicação segura entre ambos [Alan 2002].

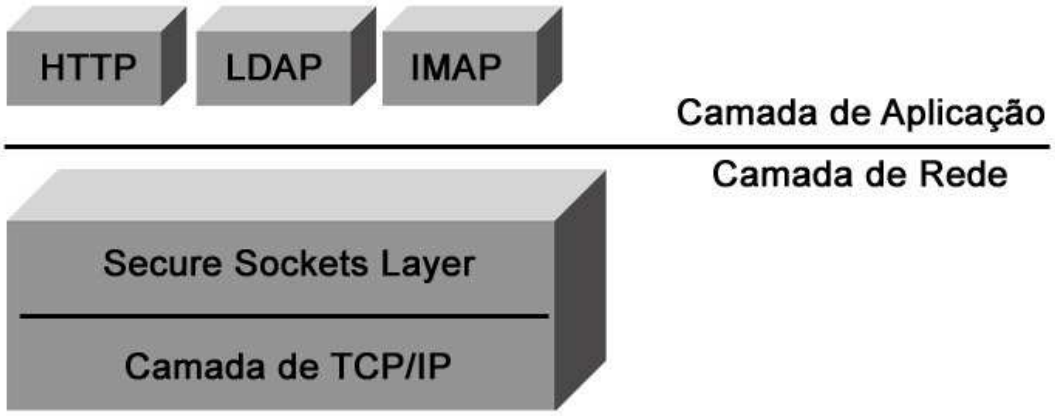

Ilustração 5 - SSL na pilha de protocolos [Pontes 2007].

Quando se utiliza o protocolo SSL em um navegador de internet é necessário que tanto o navegador quanto o servidor de internet estejam habilitados para a utilização do protocolo.

O SSL é composto de dois sub protocolos: o SSL *Record protocol* e o SSL *handshake protocol*. O SSL *Record protocol* define formatos para a transferência dos dados. O SSL *handshake protocol* utiliza o SSL *record protocol* para trocar uma série de mensagens entre um servidor habilitado com SSL e um cliente habilitado com SSL [Alshamsi 2008]. Esta troca de mensagens é para:

- Autenticar o servidor junto ao cliente;
- Permitir ao cliente e ao servidor selecionar algoritmos criptográficos que ambos suportem;
- Autenticar o cliente junto ao servidor;
- Utilizar técnicas de criptografia de chave pública para gerar chaves secretas;
- Estabelecer uma conexão SSL segura;

Decisões sobre qual algoritmo aplicar nas configurações do SSL devem levar em consideração o grau de segurança necessário aos dados envolvidos. O protocolo SSL utiliza uma combinação de técnicas de criptografia simétrica e de criptografia de chave pública. Algoritmos de criptografia simétrica executam muito mais rápidos, ao passo que criptografia de chave pública provê melhores técnicas de autenticação.

O *HandShake Protocol* permite ao cliente e ao servidor se autenticarem simultaneamente utilizando criptografia de chave pública. Permite ainda que criem uma chave simétrica de sessão que será utilizada para cifrar e decifrar as mensagens da sessão subseqüente mais rapidamente [Pontes 2007].

7.4.2. CYRUS

O cyrus é basicamente uma implementação do SASL (*Simple Authentication and Security Layer*), trata-se de um quadro de autenticação que age juntamente com outros aplicativos proporcionando segurança na autenticação de acesso.

O cyrus não age sozinho, ele deve ser incorporado a algum aplicativo orientado a conexão, ou seja, aplicativos que utilizam os protocolos orientados a conexão, por exemplo, SMTP, FTP, POP3, IMAP, LDAP. O cyrus contém um protocolo que executa um comando que serve para identificar e autenticar usuários dentro de um serviço, se a comunicação for estabelecida e o usuário autenticado uma camada de segurança é implantada entre o protocolo e a conexão [Koetter 2006].

O TCP/IP é um protocolo orientado à conexão, é confiável e uma das suas funções é o controle de fluxo. O UDP é um protocolo sem conexão, não confiável. Utilizado em aplicações onde a entrega imediata é mais importante que uma entrega precisa [Tanenbaum 2003].

Algumas vantagens e desvantagens podem ser apresentadas sobre o cyrus.

Vantagens:

- O desenvolvimento e a aplicação em *softwares* são de extrema simplicidade
- Fornece a funcionalidade estável e confiável aumentando assim a interoperabilidade com outros *softwares*.

Desvantagens:

- A documentação existente incide sobre os desenvolvedores não pode exceder alguns limites, porque muitas coisas não são documentadas.
- É difícil de memorizar, porque tudo é manipulado diferentemente.

7.5. SISTEMAS DE AUTENTICAÇÃO

Os sistemas de autenticação foram concebidos para permitir, além da verificação da autenticidade do usuário, realizar processos de autorização, e principalmente de auditoria. Estes tipos de soluções são conhecidos como soluções AAA (“*Authentication*”, “*Authorization*” and “*Accounting*”).

As soluções AAA definem um sistema capaz de ordenar políticas e gerenciar a configuração e acesso de vários dispositivos de rede e de segurança. Basicamente uma solução AAA é composta de uma base de dados, que contém arquivos com os dados dos usuários, seus perfis, suas configurações, que se comunica com roteadores, serviços de acesso remoto, servidores, etc. [Simom 2005].

- *Authentication:* consiste em validar a identidade dos usuários, ou seja, autenticar, permitindo que os mesmos tenham acesso aos sistemas;

- *Authorization:* está relacionada a autorização, que envolve definir quais os sistemas que o usuário tem direitos para acessar, incluindo endereços de rede, aplicações, sistemas, recursos;

- *Accounting:* está relacionada com a bilhetagem ou consumo dos recursos pelo usuário e, principalmente, a auditoria. Este elemento é essencial na área de segurança, de forma a possibilitar descobrir usuários que tenham cometido ações não permitidas na rede e nos sistemas;

Cada componente tem uma permissão que deve ser dada pelo administrador de rede, determinando o que o usuário comum pode ou não usar o componente ou mesmo se ele é gerente do componente. Sendo assim a arquitetura deve prever que a partir do momento em que o usuário efetua o *login* no sistema, somente os componente ao qual ele tem permissão de utilizar, assim como aqueles que ele tem o direito de gerenciar devem ser mostrados a ele, evitando dessa forma a visualização ou alteração de componentes que ele não possui permissão [Moraes 2009].

8 ESTUDO DE CASO

8.1. ROTEIRO SEGUIDO PARA A IMPLEMENTAÇÃO DO OPENLDAP

A implementação do trabalho foi feito em ambiente *Linux*, e a distribuição escolhida foi a mesma utilizada nos servidores da Universidade Estadual do Norte do Paraná, campus Luiz Meneghel. A distribuição escolhida foi *Slackware* 12.2.0 com *kernel* 2.6.27.7-smp.

Foi utilizado o ldap-account-manager como *front-end* por considerá-lo de ser mais amigável.

Os pacotes obtidos para instalação estão abaixo relacionados:

cyrus-sasl-2.1.22.tar.gz
http://sunsite.rediris.es/pub/mirror/cyrus-mail/cyrus-sasl-2.1.22.tar.gz

openldap-2.3.24.tgz
ftp://ftp.openldap.org/pub/OpenLDAP/openldap-release/openldap-2.3.24.tgz

nss_ldap-251.tgz
http://www.padl.com/download/nss_ldap.tgz

MigrationTools-57.tgz
http://www.padl.com/download/MigrationTools.tgz

Authen-SASL-2.10.tar.gz
http://www.cpan.org/authors/.../Authen-SASL-2.10.tar.gz

Convert-ASN1-0.20.tar.gz
http://search.cpan.org/CPAN/.../Convert-ASN1-0.18.tar.gz

Crypt-SmbHash-0.12.tar.gz
ftp://ftp.inria.fr/pub/CPAN/.../Crypt-SmbHash-0.12.tar.gz

Digest-SHA1-2.11.tar.gz
http://search.cpan.org/CPAN/.../Digest-SHA1-2.11.tar.gz

IO-Socket-SSL-0.97.tar.gz
http://mirrors.ibiblio.org/pub/.../IO-Socket-SSL-0.97.tar.gz

Jcode-2.05.tar.gz
http://search.cpan.org/CPAN/.../Jcode-2.05.tar.gz

Net_SSLeay.pm-1.25.tar.gz
http://mirror.uta.edu/CPAN/.../Net_SSLeay.pm-1.25.tar.gz

URI-1.33.tar.gz
http://www.volity.org/frivolity/perl/URI-1.33.tar.gz

Unicode-Map-0.112.tar.gz
http://search.cpan.org/CPAN/.../Unicode-Map-0.112.tar.gz

Unicode-Map8-0.12.tar.gz
http://search.cpan.org/CPAN/.../Unicode-Map8-0.12.tar.gz

Unicode-MapUTF8-1.11.tar.gz
http://search.cpan.org/CPAN/.../Unicode-MapUTF8-1.11.tar.gz

Unicode-String-2.09.tar.gz

http://search.cpan.org/CPAN/.../Unicode-String-2.09.tar.gz

XML-SAX-Base-1.04.tar.gz
http://www.volity.org/frivolity/perl/XML-SAX-Base-1.04.tar.gz

perl-ldap-0.33.tar.gz
http://search.cpan.org/CPAN/.../perl-ldap-0.33.tar.gz

httpd-2.0.58.tar.gz
http://ftp.unicamp.br/pub/apache/httpd/httpd-2.0.58.tar.gz

php-5.1.4.tar.gz
http://br2.php.net/get/php-5.1.4.tar.gz/from/us2.php.net/mirror

ldap-account-manager-1.0.2.tar.gz
http://prdownloads.sourceforge.net/.../ldap-account-manager-1.0.2.tar.gz

db4-4.2.52-i486-2.tgz
ftp://ftp.slackware-brasil.com.br/slackware-12.2/pasture/db4-4.2.52-i486-2.tgz

Para a instalação dos pacotes, foi utilizado o sistema de descompactação utilizando o tar –xzvf {nome_do_pacote} e os comandos /configure;make e make install para compilar o pacote junto ao sistema operacional.

Após a instalação do *Cyrus-Sasl* que é um pacote de autenticação segura, foi criado um link simbólico para que o LDAP consiga acessar o DB (Banco *Berkeley*).

```
ln -s /usr/local/lib/sasl2 /usr/lib/sasl2
```

Após a instalação do LDAP, foi editado o arquivo *slapd.conf, presente* no diretório /usr/local/etc/openldap/

```
# slapd.conf
host localhost
include /usr/local/etc/openldap/schema/core.schema
include /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema
include /usr/local/etc/openldap/schema/nis.schema
include /usr/local/etc/openldap/schema/samba.schema
include /usr/local/etc/openldap/schema/qmail.schema
```

```
pidfile /usr/local/var/run/slapd.pid
argsfile /usr/local/var/run/slapd.args

database bdb
suffix "dc= labinfo,dc= com "
rootdn "cn=administrador,dc= labinfo,dc= com "
rootpw {SSHA}KwwbIdAjWcAOlxLjgq0O4iRnl7C05NhZ
directory /usr/local/var/openldap-data

password-hash {CRYPT}
password-crypt-salt-format "$1$.8s"

index objectClass,uidNumber,gidNumber eq
index cn,sn,uid,displayName pres,sub,eq
index
memberUid,mail,mailAlternateAddress,givenname,accountStatus,mailHost,deliveryM
ode eq
index sambaSID,sambaPrimaryGroupSID,sambaDomainName eq
index default sub

access to attrs=userPassword,sambaLMPassword,sambaNTPassword
        by self write
        by anonymous auth
        by * none

access to *
        by * read
```

Com o servidor OpenLDAP devidamente configurado, é chegada a hora de “popular” sua base, procedimento este que envolve a criação de grupos e usuários que utilizarão o serviço da rede.

Editando o arquivo */root/base.ldif* :

```
dn: dc= labinfo,dc= com
dc: labinfo
objectClass: top
objectClass: domain

dn: ou=Usuarios,dc= labinfo,dc= com
ou: Usuarios
objectClass: top
objectClass: organizationalUnit

dn: ou=Grupos,dc= labinfo,dc= com
ou: Grupos
objectClass: top
```

```
objectClass: organizationalUnit

dn: ou=Computadores,dc= labinfo,dc= com
ou: Computadores
objectClass: top
objectClass: organizationalUnit
```

Para as entradas no *LDAP* foi utilizado o comando abaixo:

```
ldapadd -x -D cn=administrador,dc= labinfo,dc= com -W -f /root/base.ldif
```

```
Enter LDAP Password: 321
adding new entry "dc= labinfo,dc= com "
#grupo principal labinfo.com
adding new entry "ou=usuarios,dc= labinfo,dc= com "
# sub diretorio para as entradas de usuarios
adding new entry "ou=grupos,dc= labinfo,dc= com "
# sub diretório para as entradas dos grupos
adding new entry "ou=conputdores, dc= labinfo,dc= com "
#sub diretorios para as entradas dos computadores
```

No caso de haver grupos já existentes no sistema, foi utilizado as seguintes linhas de código no arquivo migrate_common.ph para migração dos grupos ao LDAP.

```
$NAMINGCONTEXT{'passwd'} = "ou=Usuarios";
$NAMINGCONTEXT{'group'} = "ou=Grupos";
$DEFAULT_MAIL_DOMAIN = " labinfo. com ";
$DEFAULT_BASE = "dc= labinfo,dc= com ";
$DEFAULT_MAIL_HOST = "mail. labinfo. com ";
```

Outro arquivo que foi modificado para que LDAP pudesse buscar as informações de *login* foi o /etc/nsswitch.conf, onde foi direcionado para os arquivo de senhas e grupos.

```
[global]
workgroup = LABINFO
netbios name = PDC-SRV
server string = Slackware
security = user
encrypt passwords = yes
guest account = nobody
log file = /var/log/%m.log
max log size = 50
os level = 100
local master = yes
domain master = yes
preferred master = yes
domain logons = yes
admin users = administrador root
logon script = %U.bat
logon path = %Lprofiles%U
hosts allow = 10.0.0. 10.
wins support = no
dns proxy = no

ldap passwd sync = yes
ldap delete dn = Yes
passdb backend = ldapsam:ldap://10.1.1.1/
ldap admin dn = cn=administrador,dc= labinfo,dc= com
ldap suffix = dc= labinfo,dc= com
ldap group suffix = ou=Grupos
ldap user suffix = ou=Usuarios
ldap machine suffix = ou=Computadores
ldap idmap suffix = ou=Idmap
idmap backend = ldap:ldap://10.1.1.1
idmap uid = 10000-15000
idmap gid = 10000-15000

template shell = /bin/false
winbind use default domain = no
;smb passwd file=/etc/samba/smbpasswd
passwd program=/usr/bin/passwd %u
passwd chat = *New*password* %n *Retype*new*password* %n
;#*passwd:*all*authentication*tokens*updated*successfully*
socket options = TCP_NODELAY IPTOS_LOWDELAY SO_RCVBUF=8192
SO_SNDBUF=8192

add machine script = /usr/local/sbin/smbldap-useradd -w "%u"
add user script = /usr/local/sbin/smbldap-useradd -m "%u"
delete user script = /usr/local/sbin/smbldap-userdel "%u"
add machine script = /usr/local/sbin/smbldap-useradd -w "%u"
add group script = /usr/local/sbin/smbldap-groupadd -p "%g"
delete group script = /usr/local/sbin/smbldap-groupdel "%g"
add user to group script = /usr/local/sbin/smbldap-groupmod -m "%u" "%g"
```

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

dos charset = UTF-8
unix charset = UTF-8
```

8.1.1. CONFIGURANDO O SQUID COM O LDAP

A versão do sistema *Linux* usada nesta implementação foi o Slackware 12.2.0 com kernel 2.6.27.7-smp, já traz juntamente com o pacote de instalação os pacotes do squid, portanto não a necessidade de instalação destes pacotes e nem a necessidade de compilação.

Para o Squid funcionar perfeitamente foi necessário alterar uns item do arquivo de configuração do squid.conf dentro do diretório /etc/squid para que fique compatível com o servidor.

```
http_port 10.1.1.1:3128

visible_hostname servidor.labinfo.com
```

Para a autenticação dos usuários da internet que estão usando o Servidor *Proxy* é necessário compilar o pacote squid-2.5.STABLE8/helpers /basic_auth/LDAP, daemon squid_ldap_group e daemon squid_ldap_auth, após a compilação, deve-se copiar os arquivos para o diretório /usr/local/squid/libexec.

Depois da compilação tive que fazer algumas alterações no arquivo de configuração do squid o squid.conf que esta dentro do diretório /etc/squid/.

```
auth_param basic program /usr/lib/squid/squid_ldap_auth -b "dc=labinfo,dc=com" -f
"uid=%s" -h ldap.labinfo.com
#codigo para prover comunicação com o ldap
authenticate_program /urs/bin/ncsa_auth /etc/
#linha onde serão armazenada os informações de autenticação
squid/passwd
#habilita o squid para utilizar senhas na utenticação
```

Por padrão estas linhas vem comentadas dentro do arquivo squid.conf, a mudança é necessária para habilitar o squid para usar autenticação.

8.1.1.1. **ADICIONANDO USUÁRIOS NO SQUID.**

Adicionando usuários no squid.

```
htpasswd /etc/squid/passwd João
password 123
confirm password 123
```

```
htpasswd /etc/squid/passwd maria
password 123
confirm password 123
```

Configurando o squid para trabalhar com autenticação ldap.

```
/usr/lib/squid/squid_ldap_auth -b "dc=labinfo,dc=com " -f "uid=%s" ldap.labinfo.com
```

8.1.2. CONFIGURANDO O SAMBA COM O LDAP

Assim como o squid o samba também se encontra no pacote de instalação do lackware 12.2.0 com kernel 2.6.27.7-smp, não havendo a necessidade de instalação posterior.

Alteração na configuração do samba arquivo smb.conf.

```
netbios name = SERVIDOR_SAMBA
workgroup = labinfo
server string = Samba PDC Server
security = user
domain master = yes
domain logons = yes
time server = yes
```

```
logon drive = S:
hosts allow = 10.1.1.2
# IP com acesso ao servidor
interfaces = 10.1.1.1/255..0.0.0
#IP do servidor que o samba vai escutar
log file = /var/log/samba/\%m.log
unix charset = iso8859-1
display charset = cp850
veto files =
```

Modificando o samba para autenticar juntamente com o ldap, neste caso tive que adicionar algumas linhas no smb.conf.

```
# linhas adicionadas no smb.conf
passdb backend = ldapsam:ldaps://10.1.1.1/
#identifica o ip do servidor onde reside o samba
ldap admin dn = cn=administrador,dc=labinfo,dc=com
#identificando o workgroup do ldap
ldap suffix = dc=labinfo,dc=com
ldap group suffix = ou=Grupos
ldap user suffix = ou=Usuários
ldap machine suffix = ou=computadores
```

8.1.2.1. ADICIONANDO USUÁRIOS NO SAMBA

```
smbpasswd -c /etc/samba/smbpasswd –a João
password 123
confirm password 123
```

```
smbpasswd -c /etc/samba/smbpasswd –a Maria
password 123
confirm password 123
```

Migrando os usuários do samba para o ldap

```
slapadd -l usuarios.ldif
```

8.1.3. INSTALANDO O SERVIÇO DE FTP COM OPENLDAP

Para o serviço de FTP foi utilizado o *software* ProFTPd que é um *software* que fornece serviços de FTP, sua manipulação é bastante simples e suas vantagens são segurança e a flexibilidade uma vez que poder ser implantado em grande parte das plataformas operacionais. Houve a necessidade de instalar também o MySQL para fazer a autenticação dos usuários, este é necessário devido o fato do ProFTPd não ter um sistema de autenticação implementado, por padrão ele busca os usuários do próprio sistema o que não é muito pratico.

O ProFTPd e o MySQL foram instalados utilizando os pacotes.

proftpd-1.3.2.tar.bz2
ftp://ftp1.be.proftpd.org/proftpd/distrib/source/proftpd-1.3.2.tar.bz2

mysql-4.0.15a-i486-1.tgz
http://www.slackware.at/data/slackware-9.1/slackware/ap/mysql-4.0.15a-i486-1.tgz

Apos realizar a instalação do pacote do ProFTPd, foi realizada a edição do arquivo proftpd.conf que esta localizado dentro do diretório /etc/.

```
ServerName                  "Servidor_FTP"
ServerType                  standalone
DeferWelcome                  off
MultilineRFC2228               on
DefaultServer                 on
ShowSymlinks                   on
TimeoutNoTransfer               600
TimeoutStalled                600
TimeoutIdle                  1200
DisplayLogin                  Ben-vindo.msg
DisplayFirstChdir              .message
ListOptions                  "-l"
DenyFilter                  \*.*/
Port                      21
MaxInstances                   3
#numero de conexões simultaneas
DefaultRoot
```

```
# serve para definir o diretório que terá acesso
User                    nobody
Group                   nogroup
Umask                   022     022
# o umask serve para definir os privilégios de leitura e escrita
PersistentPasswd        on
# esta linha permite que seja utilizado a autenticação Ldap
TLSEngine               on
Quotas                  on
Ratios                  on

#linhas adicionadas para a cominicação com MySQL
SQLAuthTypes Plaintext Crypt
SQLAuthenticate users
SQLConnectInfo ftp alex 123
SQLUserInfo usuarios_ftp login senha
```

Após escrever as configurações acima no arquivo /etc/proftpd.conf , inicie o serviço com o seguinte comando:

```
/etc/init.d/proftpd start
```

Configurando o MySQL, criando as base de dados e tabelas.

```
mysqld_safe &
#comando para chamar o MySQL
mysqladmin create ftp -u root -p
#criando a base da dados do MySQL
mysql -u root –p
#logando

#inicio da criação das tabelas basicas
mysql> CREATE TABLE usuarios_ftp (
-> nome varchar(255) NOT NULL DEFAULT,
-> login varchar(20) NOT NULL UNIQUE,
-> senha varchar(20) NOT NULL,
Query OK, 0 rows affected (0.00 sec)
#fim da criação das tabelas

mysql> GRANT ALL PRIVILEGES ON ftp.* to ‘Alex’ IDENTIFIED BY '321'; Query
OK, 0 rows affected (0.03 sec)
#criei um usuário para acesso ao MySQL

mysql> INSERT INTO usuarios_ftp VALUES ('joão silva', 'joão', '123');
```

```
mysql> INSERT INTO usuarios_ftp  VALUES ('maria silva', 'maria', '123');
#criei os susarios joão e maria no MySQL
mysql> quit
```

Com estes passos o serviço de FTP está configurado e rodando no servidor.

Para prover a comunicação do ProFTPd com o Ldap foi inserido as seguintes linhas no arquivo proftp.conf.

```
LDAPServer localhost
#host do serviço do Ldap
LDAPDNInfo cn=administrador,dc=labinfo,dc=com
LDAPDoAuth on "ou=Usuários,dc=labinfo,dc=com" &(uid=%v)(ftpaccess=1))
```

8.2. ROTEIRO SEGUIDO PARA A IMPLEMENTAÇÃO DO FREE RADIUS

Nesta instalação foi usado o FreeRadius habilitado para a base MySQL, que servirá como banco. Tanto o FreeRadius quanto o LDAP possuem suporte de diversos SGBD, neste caso foi utilizado o banco MySQL para não ter que fazer a integração do Radius com o LDAP.

A instalação do FreeRadius foi feita utilizando o pacote.

freeradius-server-2.1.7.tar.gz
ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-2.1.7.tar.gz

A configuração do arquivo radiusd.conf esta reduzida apenas a parte mais importante devido sua extensão o arquivo original esta em anexo, a configuração ficou da seguinte forma.

```
user = Alex
group = labinfo
bind_address = 10.1.1.1
port = 1812
$INCLUDE  ${confdir}/mssql.conf
sql
 radutmp
```

Configuração do arquivo clients.conf, para informações do cliente de acesso ao servidor.

```
# clients.conf

    client 10.1.1.1
    secret = 321
    # senha do servidor

    shortname = servidor
    # nome do cliente

    nastype = other

# fim clients.conf
```

Criando a base de dados no MySQL para a utilização do Radius.

```
mysqladmin create radius -u root -p
mysql -u root –p
mysql> CREATE TABLE usuarios_radius (
-> nome varchar(255) NOT NULL DEFAULT,
-> login varchar(20) NOT NULL UNIQUE,
-> senha varchar(20) NOT NULL,
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO usuarios_ftp VALUES ('joão silva', 'joão', '123');
mysql> INSERT INTO usuarios_ftp  VALUES ('maria silva', 'maria', '123');
mysql> quit
```

Foi utilizado o arquivo db_mysql.sql presente no diretório /usr/local/src/freeradius-1.0.1/src/modules/rlm_sql/drivers/rlm_sql_mysql o qual foi usado para criar as tabelas.

8.3. AUTENTICAÇÃO NA BASE DE DADOS MySQL

Para que o Radius faça a autenticação utilizando o MySQL coloque as seguintes linha no arquivo radius.conf.

```
sql {
    driver = "rlm_sql_mysql"
    # informa ao freeradius qual modulo de banco de dados usar, neste caso, mysql
    server = "servidor"
    # diz ao freeradius em qual host está o servidor mysql
    login = "Alex"
    # define o nome de usuário registrado no mysql
    password = "321"
    # senha do usuário definido no parâmetro "login"
    radius_db = "radius"
    # nome do banco de dados que contem as tabelas
}
```

.

8.4. CONFIGURANDO O SQUID PARA AUTENTICAR COM RADIUS

Para que o squid se comunique de forma correta com o Radius foi acrescentado algumas linhas no arquivo squid.conf.

```
auth_param basic program /usr/local/libexec/squid/squid_radius_auth -f \
/usr/local/etc/squid/squid_radius_auth.conf
auth_param basic children 5
auth_param basic realm 10.1.1.1, 321
auth_param basic credentialsttl 2 hours
authenticate_cache_garbage_interval 1 hour
authenticate_ttl 1 hour
authenticate_ip_ttl 0 seconds
```

Houve também a necessidade de editar o arquivo squid_radius_auth.conf para prover comunicação e autenticação dos serviços.

```
squid_rad_auth configuration file
#MvS: 28-10-2009
server 10.1.1.1
secret teste 321
```

8.5. CONFIGURANDO O SAMBA PARA AUTENTICAR COM RADIUS

Para configurar o samba para rodar junto com o Radius foi modificado algumas linhas do smb.conf. que ficou assim.

```
radius server admin dn =labinfo cn=administrador,dn=com
radiusbind_address = 10.1.1.1
radius port = 1812
passwd sync = Yes
suffix
ssl = start tls
passdb expand explicit = No
```

8.6. CONFIGURANDO O ProFTPd PARA AUTENTICAR COM RADIUS

Configuração feita no proftpd.conf para prover autenticação através do Radius.

```
RadiusServer localhost
DNInfo cn=administrador,dc=falm,dc=com
Radius_user = Alex
port = 1812
bind_address = 10.1.1.1
ServerName "ProFTPD"
ServerIdent on "FTP server ready."
RequireValidShell off
ServerType standalone
DefaultServer on
ScoreboardFile /var/run/proftpd/proftpd
DeferWelcome on
ServerAdmin root@tom.com.br
SyslogFacility AUTH
AllowOverwrite yes
Port 802
```

9 MÉTRICA

Esta métrica foi gerada através do estudo das ferramentas OpenLDAP e FrreRadius. Para a geração da métrica foi utilizado um valor que ficou entre 5 e 10 para a avaliação das ferramentas, sendo que 5 vale para a menor complexidade e 10 para a maior, estas avaliações se voltaram para complexidade de instalação e complexidade de configuração para o funcionamento.

Ferramenta	Numero de pacotes	Complexidade de instalação	Complexidade de configuração
OpenLDAP	22	9	5
FreeRadius	2	5	10

O OpenLDAP necessitou de uma grande quantidade de ferramentas pra a instalação e bom funcionamento apesar de que nos meus teste não utilizei a grande maioria das ferramentas principalmente as de segurança e criptografia como por exemplo o cyrus, nss, Authen, Crypt, este fato ocorreu por ter realizado teste básicos onde foi dado mais ênfase no processo de comunicação entre as ferramentas e os serviços da rede que foi utilizado, *Squid*, Samba e FTP.

A razão pela qual o LDAP necessita de uma grande quantidade de pacotes é que seu desenvolvimento foi realizado visando a integração com diversos serviços, enquanto que o Radius mantém uma integração limitada.

10 CONCLUSÃO

Percebe-se que o LDAP possui inúmeras áreas de aplicabilidade, pois se trata de um protocolo eficiente que pode ser utilizado desde pequenas empresas até grandes corporações para integração de seus inúmeros serviços. Além de possuir uma grande escalabilidade, isto é, podem ser adicionadas várias expansões tanto na linha de operações funcionais quanto em comandos de controle, o LDAP ainda possui várias opções para a segurança de dados, pois adota, atualmente, um dos *frameworks* mais utilizados e flexíveis da Internet (SASL). Porém, o LDAP deve ser projetado com muito cuidado, pois ele não se trata de uma substituição definitiva a bancos de dados ou outros serviços, como servidores FTP, servidores WEB ou sistemas de arquivos. Uma analise completa de quais são os requisitos do serviço onde se pretende empregar o LDAP deve ser realizada, pois as necessidades da rede podem não ser supridas com a adesão da ferramenta LDAP.

As duas ferramentas estudadas neste trabalho se mostraram muito estáveis no processo de autenticação, embora para redes que concentrem uma gama muito grande de serviço a melhor opção é o OpeLDAP devido a sua integração com uma enorme quantidade de serviços diferente do Radius que não fornece tanta integração assim.

Em trabalhos futuros pretende-se estudar a fundo a rede da instituição acadêmica para verificar quais os serviços mais necessários para prover autenticação e verificar se uma das duas ferramentas se enquadra nas necessidades da rede da instituição.

11 REVISÃO BIBLIOGRÁFICA

[1] ALAN O. F, Philip K, **The SSL Protocol version 3.0**, Netscape Corporation, disponivel em http://wp.netscape.com/eng/ssl3/draft302.txt, acessado em agosto de 209.

[2] ALSHAMSI Abdel Nasir; SAITO Takamichi, **A TECHNICAL COMPARISON OF IPSEC AND SSL,** disponível em http://eprint.iacr.org/2004/314.pdf, acessado em novembro de 2009.

[3] BARROS Luiz Gustavo, **AUTENTICAÇÃO IEEE 802.1X EM REDES DE COMPUTADORES UTILIZANDO TLS E EAP**, disponivel em http://www.4eetcg.uepg.br/oral/62_1.pdf, acessado em agosto de 2009.

[4] CASANOVA Marco Antonio, **INTEGRAÇÃO E INTEROPERABILIDADE ENTRE FONTES DE DADOS GEOGRÁFICOS,** disponível em http://www.dpi.inpe.br/gilberto/livro/bdados/cap9.pdf, acessado em agosto de 2009.

[5] FIGUEIREDO João Filho Matos, **PROJETO DE SEGURANÇA E MODERNIZAÇÃO DA REDE DO DEPARTAMENTO DE INFORMATICA DA UFPB**, disponível em http://www.joaomatosf.com/blog/files/projeto_seguranca.pdf, acessado em agosto de 2009.

[6] GOUVEIA Luiz Manuel Borges, **SISTEMAS DE INFORMAÇÃO,** disponível em http://homepage.ufp.pt/~lmbg/textos/si_texto.pdf, acessado em outubro de 2009.

[7] KANIES A. Luke, **UMA INTRODUÇÃO AO LDAP**, disponível em http://www.oreillynet.com/pub/a/onlamp/2001/08/16/ldap.html, acessado em julho de 2009.

[8] KOETTER Patrick, HILDEBRANDT Ralf, **SURVIVING CYRUS SASL,** disponivel em http://www.arschkrebs.de/slides/surviving_cyrus_sasl-handout.pdf, acessado em novembro de 2009.

[9] LIMA Marcelo Barboza, **ROGUE ACCESS POINT, UM GRANDE RISCO PARA WLAN**, disponível em http://www.las.ic.unicamp.br/srac/imagens/SSI2003-RogueAP.pdf, acessado em agosto de 2009.

[10] LOI Leandro Nascimento, DELUCCA José Eduardo, **UM ESTUDO DAS METODOLOGIAS OPEN SOURCE IDENTITY MANAGEMENT E INDICAÇÃO DA MELHOR A SER IMPLANTADA NO PROJETO VIA DIGITAL,** disponível em http://projetos.inf.ufsc.br/arquivos_projetos/projeto_793/ArtigoTCC.pdf, acessado em agosto de 2009.

[11] MORAES Alexandre Fernandes de, **MÉTODO PARA AVALIAÇÃO DA TECNOLOGIA BIOMÉTRICA NA SEGURANÇA DE AEROPORTOS,** disponível em http://sabia.pcs.usp.br/gas/files/publications/dissertacao_alexandremoraes.pdf, acessado em novembro de 2009.

[12] OLIVEIRA Álvaro Mendes de, **REDES LOCAIS SEM FIOS SEGURAS EM AMBIENTES CORPORATIVOS**, disponível em http://alvaro.mendes.oliveira.nom.br/mba/monografia-mba-alvaro.pdf, acessado em agosto de 2009.

[13] ORACLE Technical White Paper, **A COMPARISON OF ORACLE BERKELEY DB AND RELATIONAL DATABASE MANAGEMENT SYSTEMS**, disponível em http://www.oracle.com/database/docs/Berkeley-DB-v-Relational.pdf, acessado em setembro de 2009.

[14] PONTES Krishnen Lage, **PROPOSTA DE UM MODELO DE QUALIDADE DE SERVIÇO E SEGURANÇA PARA A TECNOLOGIA DE *WEB SERVICES*** disponível em http://www.tede.ufsc.br/teses/PGCC0521.pdf, acessado em novembro de 2009.

[15] RIBEIRO Sildemir Alves, **FIREWALL EM LINUX**, disponível em https://www.ginux.ufla.br/files/mono-SildenirRibeiro.pdf, acessado em agosto de 2009.

[16] SALDANHA Leônidas Klein, **FIREWALLS DE BAIXO CUSTO**, disponível em http://nead.feevale.br/tc/files/995.pdf, acessado em agosto de 2009.

[17] SILVA Lino Sardo da, **REDES PRIVADAS VIRTUAIS EM PLATAFORMAS LINUX E Windows**, Novatec Editora Ltda 2002.

[18] SIMON D., **EXTENSIBLE AUTHENTICATION PROTOCOL (EAP) KEY MANAGEMENT FRAMEWORK,** disponível em http://www.ceset.unicamp.br/~nalon/ST664/rfc5247.pdf, acessado em novembro de 2009.

[19] SOUZA João Nunes, **UMA ANÁLISE DOS MECANISMOS DE SEGURANÇA DE REDES LOCAIS SEM FIO E UMA PROPOSTA DE MELHORIA**, disponível em http://svn.assembla.com/svn/odinIDS/Egio/artigos/SegurancaMovel/seguranca.pdf ,acessado em agosto de 2009.

[20] STEIN Gabriel, **ENTENDENDO O LDAP**, disponível em http://www.ldap.org.br/modules/ldap/files/files///entendendo_openldap.pdf, acessado em setembro de 2009.

[21] TANENBAUM Andrew S., **REDES DE COMPUTADORES** Tradução de quarta edição Editora Campus 2003.

[22] TRIGO Clodonil Honório, **OPENLDAP UMA ABORDAGEM INTEGRA**, disponível m http://www.vivaolinux.com.br/artigos/impressora.php?codigo=270&P, acessado em agosto de 2009.

[23] VALCY Ítalo, **ADMINISTRAÇÃO DE REDES COM GNU/LINUX**, disponível em http://www.linuxit.com.br/modules.php?name=News&file=print&sid=38, acessado em julho de 2009.

[24] VERISSIMO Fernando, **SEGURANÇAS EM REDES SEM FIO**, disponível em http://www.ravel.ufrj.br/arquivosPublicacoes/wnsmono.pdf, acessado em agosto de 2009.

ANEXO

Arquivo de configuração do Radius, radius.conf.

```
prefix = /usr/local/src
exec_prefix = ${prefix}
sysconfdir = ${prefix}/etc
localstatedir = ${prefix}/var
sbindir = ${exec_prefix}/sbin
logdir = ${localstatedir}/log/radius
raddbdir = ${sysconfdir}/raddb
radacctdir = ${logdir}/radacct
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/radiusd
log_file = ${logdir}/radius.log
libdir = ${exec_prefix}/lib
pidfile = ${run_dir}/radiusd.pid

user = Alex
group = labinfo
# definição do usuário e grupo

max_request_time = 30
# tempo Maximo em segundos para processar o pedido de autenticação

delete_blocked_requests = no
cleanup_delay = 5
max_requests = 256
# número máximo de pedidos que o freeradius pode atender simultaneamente

bind_address = 10.1.1.1
# IP que o freeradius irá escutar e responder

port = 1812
# porta de escuta

hostname_lookups = no
allow_core_dumps = no
regular_expressions = yes
extended_expressions = yes
log_stripped_names = no
log_auth = no
log_auth_badpass = no
log_auth_goodpass = no
usercollide = no
lower_user = no
lower_pass = no
nospace_user = no
nospace_pass = no
checkrad = ${sbindir}/checkrad
security {
```

```
    max_attributes = 200
    reject_delay = 1
    status_server = no
}

proxy_requests = yes
$INCLUDE ${confdir}/proxy.conf
$INCLUDE ${confdir}/clients.conf

snmp = no
$INCLUDE ${confdir}/snmp.conf
thread pool {
    start_servers = 5
    max_servers = 32
    min_spare_servers = 3
    max_spare_servers = 10
    max_requests_per_server = 0
}
modules {
    # formato:
    #    name [ instance ] {
    #        config_item = value

    pap {
        encryption_scheme = crypt
    }

    chap {
        authtype = CHAP
    }

    pam {
        pam_auth = radiusd
    }

    # /etc/passwd e /etc/shadow
    unix {
        cache = no
        cache_reload = 600
        radwtmp = ${logdir}/radwtmp
    }

    #  Extensible Authentication Protocol
    $INCLUDE ${confdir}/eap.conf

    # Micro$oft CHAP authentication
    mschap {
        authtype = MS-CHAP
        # protocolo M$ usado
        #use_mppe = no
```

```
        #require_encryption = yes
        #require_strong = yes
        #with_ntdomain_hack = no
        #ntlm_auth = "/path/to/ntlm_auth --request-nt-key --username=%{Stripped-User-
Name:-%{User-Name:-None}} --challenge=%{mschap:Challenge:-00} --nt-
response=%{mschap:NT-Response:-00}"
    }


    # Lightweight Directory Access Protocol (LDAP)
    # permite usa autenticação LDAP
    ldap {
        server = "ldap.your.domain"
        # identity = "cn=admin,o=My Org,c=UA"
        # password = senhadnaqui
        basedn = "o=My Org,c=UA"
        filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
        # base_filter = "(objectclass=radiusprofile)"

        start_tls = no
         # default_profile = "cn=radprofile,ou=dialup,o=My Org,c=UA"
        # profile_attribute = "radiusProfileDn"
        access_attr = "dialupAccess"

        dictionary_mapping = ${raddbdir}/ldap.attrmap
        ldap_connections_number = 5

        # password_header = "{clear}"
        # password_attribute = userPassword
        # groupname_attribute = cn
        # groupmembership_filter = "(|(&(objectClass=GroupOfNames)
(member=%{Ldap-UserDn})) (&(objectClass=GroupOfUniqueNames)
(uniquemember=%{Ldap-UserDn})))"
        # groupmembership_attribute = radiusGroupName
        timeout = 4
        timelimit = 3
        net_timeout = 1
        # compare_check_items = yes
        # do_xlat = yes
        # access_attr_used_for_allow = yes
    }

#
    # modulo Realm, para proxy
    #  'realm/username'
    realm IPASS {
        format = prefix
        delimiter = "/"
        ignore_default = no
        ignore_null = no
```

```
}
#  'username@realm'
realm suffix {
    format = suffix
    delimiter = "@"
    ignore_default = no
    ignore_null = no
}
#  'username%realm'
realm realmpercent {
    format = suffix
    delimiter = "%"
    ignore_default = no
    ignore_null = no
}
#  'domain\user'
realm ntdomain {
    format = prefix
    delimiter = "\\"
    ignore_default = no
    ignore_null = no
}

checkval {
    item-name = Calling-Station-Id
    check-name = Calling-Station-Id
    data-type = string
    #notfound-reject = no
}


preprocess {
    huntgroups = ${confdir}/huntgroups
    hints = ${confdir}/hints
    with_ascend_hack = no
    ascend_channels_per_line = 23
    with_ntdomain_hack = no
    with_specialix_jetstream_hack = no
    with_cisco_vsa_hack = no
}

files {
    usersfile = ${confdir}/users
    acctusersfile = ${confdir}/acct_users

    compat = no
}
```

```
   #  Para MySQL:        ${confdir}/mssql.conf

   $INCLUDE  ${confdir}/mssql.conf
   # inclusão do arquivo de configuração do banco de dados MySQL
authorise {
       preprocess
       cha
       mschap
       sulfix
       SQL
}
authenticate {
   Auth-Type PAP {
       pap
   }
   Auth-Type CHAP {
       chap
   }
   Auth-Type MS-CHAP {
       mschap
   }
#    digest
#    pam
#    unix
#    Auth-Type LDAP {
#        ldap
#    }
#    eap
}


preacct {
 preprocess
#    acct_unique
   #  home server as authentication requests.
#    IPASS
  suffix
#    ntdomain

   #
   #  Read the 'acct_users' file
#    files
}

accounting {
acct_unique
detail
#    daily
 unix
 radutmp
```

```
#    sradutmp
#    main_pool
  sql
#    pgsql-voip
}

session {
  radutmp
  sql
}

# fim radiusd.conf
```