UFLA
UNIVERSIDADE FEDERAL DE LAVRAS

VPN'S DE BAIXO CUSTO COM DNS DINÂMICO E ADSL

DAVIS ANDERSON FIGUEIREDO

2006

Davis Anderson Figueiredo

VPN's de baixo custo com DNS dinâmico e ADSL

Monografia de Pós-Graduação apresentada ao Departamento de Ciência da computação da Universidade Federal de Lavras como parte das exigências do Curso ARL- Administração em Redes Linux.

Orientador
Prof. Joaquim Quinteiro Uchôa

Lavras
Minas Gerais – Brasil
2006

Davis Anderson Figueiredo

VPN's de baixo custo com DNS dinâmico e ADSL

Monografia de Pós-Graduação apresentada ao Departamento de Ciência da computação da Universidade Federal de Lavras como parte das exigências do Curso ARL- Administração em Redes Linux.

Aprovada em *30 de Setembro de 2006*

_____
Profa. Dsc. Simone Markenson Pech

_____
Prof. Msc. Denilson Vedoveto Martins

_____
Prof. Dsc. Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil

Agradecimentos

À minha querida esposa Nívea pelo carinho, dedicação e apoio, e por ter me incentivado nos momentos difíceis não deixando que eu abandonasse "o barco".
Aos meus pais pela minha criação e formação do meu caráter.
Ao Professor Joaquim pela oportunidade de crescimento e orientação neste trabalho.

Resumo

A VPN cria "túneis virtuais" de comunicação entre essas redes ou *hosts*, fazendo com que os dados trafeguem de forma segura usando métodos criptográficos, aumentando a segurança na comunicação. O objetivo desse trabalho visa a integração dos protocolos para VPN`s usando a tecnologia ADSL e o serviço de DNS Dinâmico. Esse estudo pode mostrar que o Linux juntamente com o software livre, podem proporcionar uma redução de custo e possibilidade de melhoria na qualidade de serviço usados e prestados por pequenas empresas.

Sumário

Lista de Figuras

Capítulo 1

INTRODUÇÃO

Com o surgimento das redes de computadores, esses equipamentos passaram a se comunicar através de protocolos, padrões definidos pela industria e seus fabricantes. O desenvolvimento dos protocolos tinha como objetivos principais, a comunicação bem definida e padronizada e o compartilhamento de recursos entre os *hosts* (computadores em rede). Com estruturas pequenas e implementadas em ambiente local, a segurança dessa comunicação não era levada em consideração nesse momento.

A intercomunicação dos equipamentos em rede, crescia mais a cada dia. E nesse período, meados dos anos 70, surgiu um modelo de protocolos que se destacou sobre os demais, tornado em pouco tempo um padrão de *facto* nas redes de computadores, seu nome era TCP/IP. O TCP/IP tem como característica principal ter um padrão de desenvolvimento aberto, dando assim liberdade a todos de usá-lo sem restrições de patentes ou licenças. E essas virtudes foram determinantes na sua disseminação.

Mas realmente o ponto crucial para o domínio maciço do uso do TCP/IP nas redes de computadores foi o surgimento da grande rede mundial de computadores, a Internet, que o adotou como arquitetura de comunicação. O problema era que como seus antepassados o TCP/IP também era um modelo de protocolo que não se preocupava coma segurança na troca de dados na rede.

Em uma rede mundial como a Internet isso se tornou um grande problema. Principalmente quando se queriam trafegar dados sigilosos entre empresas, filias e matriz, ou acesso remoto as sua redes internas passando pela Internet.

Para resolver essas questões, companhias de telecomunicações aproveitaram dessa necessidade e começaram a oferecer as empresas ligações de comunicação exclu-

sivas (*links* dedicados) para o tráfego das informações importantes e sigilosas. Mas a grande questão desse *links* eram os seus altos custos, que, na maioria das vezes, só conseguiam ser pagos por empresa de grande e médio porte.

Várias soluções foram criadas com o objetivo de reduzir o custo da comunicação segura, entre elas uma se destaca, a VPN *(Virtual Private Network)* ou Rede Privada Virtual. A VPN cria "túneis virtuais" de comunicação entre essas redes ou *hosts*, fazendo com que os dados trafeguem de forma segura usando métodos criptográficos, aumentando a segurança na comunicação. E para a construção dessas VPN`s tem se usado uma diversidade de protocolos, como: PPTP, L2TP, Ipsec e SSL.

Um outro fator muito favorável ao uso de VPN`s foi o surgimento e crescimentos das conexões dedicadas e de custo mais baixo chamadas de Banda Larga. Esses *links* de Banda Larga utilizam a tecnologia ADSL (*Asymetric Digital Subscriber Line*) desenvolvida para prover acesso com maior e melhor qualidade para pequenas e micro empresas e também para assinantes residenciais, através de uma infra-estrutura já disponível pela rede de telefonia (fios telefônicos).

Ao estabelecer conectividade via ADSL, é recebido dinamicamente do provedor de acesso um endereço IP válido para ligar se a Internet. Por essa características e pela escassez de endereços IP válidos, a cada reinicialização ou inatividade do equipamento, pode ser cedido um outro endereço IP.

Como a conexão de uma VPN está ligada diretamente a 2 endereços IP, um de origem e outro de destino, a troca repentina de um dos IP's interromperia a conexão, inviabilizando a comunicação na VPN. A resolução para esse problema se dá através de um serviço de DDNS (Dinâmico DNS) no qual se define um nome para cada *host* que liga a VPN, e a esse nome é associado seu respectivo endereço IP. Um programa que monitora a troca do endereço IP também é instalado em cada *host.* Então, caso ocorra qualquer mudança de endereço IP, o programa de monitoramento envia uma mensagem com o novo IP para o DDNS, associando ao nome, que sempre será único para cada máquina. O objetivo desse trabalho visa a integração dos protocolos para

VPN`s usando a tecnologia ADSL e o serviço de DNS Dinâmico. Para isso, no Capítulo 2 serão abordados os conceitos fundamentais e as tecnologias utilizadas para a construção de VPN's de baixo custo. No Capítulo 3, serão apresentados os processos de instalação e configuração dos aplicativos utilizados, apresentando também a implementação dessa proposta em um ambiente de produção. Por fim, no Capítulo 4 são apresentadas os principais resultados alcançados com este trabalho.

Capítulo 2

CONCEITOS BÁSICOS

2.1 Comentários Iniciais

Como as redes de computadores foram projetadas inicialmente com finalidade de viabilizar a conectividade e interoperabilidade entre os equipamentos, a segurança, não foi enfatizada nesse momento. Mas com o crescimento e com o uso comercial maciço dessa tecnologia, a segurança tornou-se um ponto crucial no processo de comunicação entre computadores, *software* e *hardware* de rede.

A discussão sobre formas de comunicação segura é muito abrangente, relatando-se processos, padrões, protocolos e infra-estrutura ideal. Para esclarecer melhor essa discussão serão abordados nesse capitulo diversas tecnologias para melhoria da segurança como: mecanismos de criptografia, algoritmos e protocolos criptográficos. Serão explanados também protocolos para redes privadas virtuais – VPN e infra-estrutura de baixo custo para acesso a Internet e intercomunicação entre pontos distintos com uma matriz e filial de uma empresa, utilizando a tecnologia ADSL.

E para finalizar esse capítulo serão apresentados conceitos sobre o processo de resolução de nomes de computadores em endereço IP, através do protocolo DNS, além de um provedore de DNS dinâmicos gratuitos como o No-IP (www.No-IP.com) para registro de nomes de *hosts*.

2.2 Criptografia

De acordo com (Scheneier 1996), a palavras criptografia vem do grego (Kryptos=escondido, oculto e Grafia = Escrita). A criptografia é a ciência que usa a matemática para encriptar e decriptar dados, permitindo que informações sigilosas ou mensagens

transmita por redes, trafeguem livremente sem serem lidas por indivíduos ou instituições que não sejam o destinatário final da mensagem. Dessa forma ela garante confidencialidade, autenticidade, integridade e não-repúdio.

O processo de criptografia pode ser descrito da seguinte forma: um emissor gera a mensagem original chamada de texto plano e utilizando uma chave e um algoritmo de cifragem gera um texto cifrado, ou seja, incompreensível para quem não tem autorização de lê-lo. Ao chegar ao receptor, este texto passa pelo processo inverso, chamado decifragem, resultando no texto plano original, observado na Figura 1

Figura 1: Processo de Cifragem e Decifragem de Arquivo – Fonte: (NAI, 1999)

2.2.1 Criptografia Simétrica

A Criptografia simétrica tem como base a igualdade (simetria) das chaves que participam do processo de criptografia, sendo assim, a chave usada para criptografar também será usada para descriptografar as informações, como mostra a Figura 2

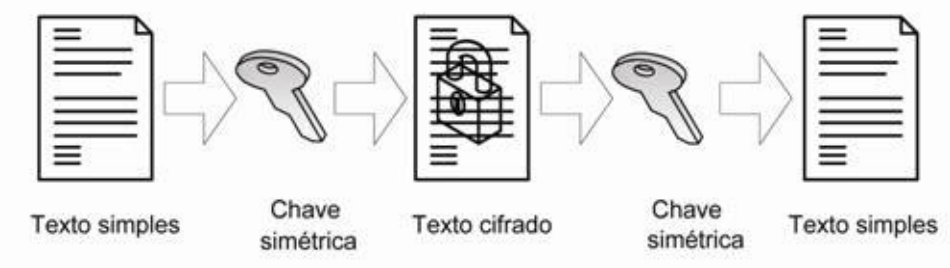

Figura 2: Processo de Criptografia Simétrica – Fonte: (MSDN, 2005)

Como exemplo: se Davis deseja enviar uma informação criptografada (secreta) para Nívea, ele deve fazer o seguinte:

Mensagem(texto plano) + ChaveSimétrica = Mensagem Criptografada

Então MensagemCriptografada é enviada para Nívea. Para ler a mensagem Nívea terá que fazer o seguinte:

MensagemCriptografada + ChaveSimétrica = Mensagem(texto plano)

O que é apresentado na Figura 2 ou no exemplo da soma da mensagem com a chave simétrica, é na verdade o algoritmo de cifragem que criptografa e descriptografa a mensagem.

A Chave Simétrica deve ser sempre secreta e bem guardada, pois é ela que define o nível de segurança da comunicação. Todas as pessoas que querem se comunicar de forma segura devem ter uma copia idêntica da chave Simétrica.

O algoritmo DES *(Data Encription Standard)* criado pela IBM foi um dos primeiros padrões criados. Ele gera uma chave simétrica de 56 bits que atualmente já pode ser quebrado (decifrada). Apesar disso, ele ainda é usado até hoje, mais algoritmos mais seguros e modernos já surgiram para substituí-lo, como 3DES, AES, IDEA, etc.

O maior problema da criptografia de chave simétrica é como o remetente envia a chave secreta ao destinatário Se um intruso descobri-la, poderá ler todas as mensagens trocadas. Mais ainda, pode comprometer a comunicação entre todo o conjunto de usuários que confiavam nessa chave.

2.2.2 Criptografia Assimétrica

Os problemas com Criptografia Simétrica foram eliminados ainda na década de 70, quando Whitfield Diffie e Martin Hellman publicaram os conceitos da chave assimétrica, também conhecida por criptografia por par de chaves ou de criptografia de chave pública. Trata-se de uma revolução no campo das comunicações. Eles descobriram fórmulas matemáticas que permitem que cada usuário tenha um par de chaves de criptografia matematicamente relacionadas, uma privada e outra pública, sendo a última, como o próprio nome diz, publicamente disponível para qualquer pessoa (Melo;Trigo,2004).

Essas fórmulas têm a seguinte característica: o que for criptografado com uma chave, só pode ser descriptografado com seu par, o que pode ser observado na Figura 3.

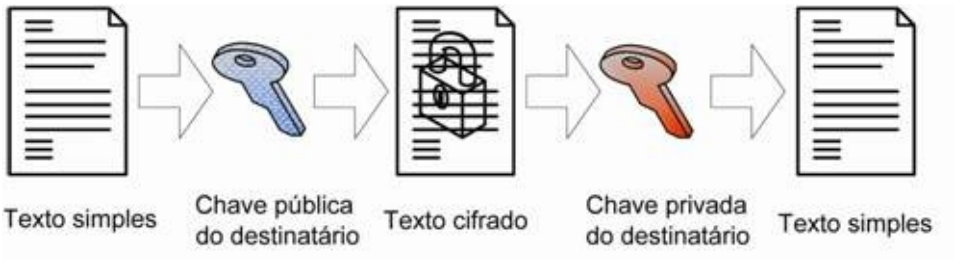

Figura 3: Processo de Criptografia Assimétrica – Fonte: (MSDN, 2005)

Então, no nosso exemplo, Nívea agora enviaria uma informação para Davis da seguinte maneira:

Mensagem(texto plano) + ChavePública(Davis) = MensagemCriptografada

E Davis leria a mensagem assim:

MensagemCriptografada + ChavePrivada(Davis) = Mensagem(texto plano)

E Davis responderia para Nívea da mesma forma:

Resposta(texto plano) + ChavePública(Nívea) = RespostaCriptografada

Sendo assim, uma mensagem criptografada com a chave pública de uma pessoa, só pode ser descriptografada com a chave privada da mesma pessoa, então a primeira chave pública pode ser livremente disponibilizada na Internet. E se a chave privada da Nívea for roubada, somente as mensagens para a Nívea estariam comprometidas.

Criptografia assimétrica pode permitir também a garantia de autenticidade: se Davis quer publicar um documento e, e que essa garantia ele pode fazer:

Documento (texto claro) + ChavePrivada(Davis) = DocumentoCriptografado

Se um leitor conseguir descriptografar este documento com a chave pública da Davis significa que ele foi criptografado com a chave privada da Davis, que somente ela tem a posse, o que significa que somente o Davis poderia tê-lo publicado.

Uma aplicação muito comum e que poderia ser usada para implementar os exemplos citados anteriormente é o GPG (*GnuPG*), que utiliza o algoritmo para assinatura digital e criptografia DSA(*Digital Signature Algorithm*).

Um outro algoritmos muito utilizado para criptografia de chaves publica e o RSA (*Ron **R**ivest, Adi **S**hamir e Len **A**dleman*) que leva o nome de seus criadores e é considerada a implementação mais conhecida para esse tipo de criptografia (Uchôa, 2005) .

2.3 O Protocolo SSL

Um grande problema na Internet e no comércio eletrônico era o protocolo HTTP (*Hypertext Transfer Protocol*), que não tinha preocupação com a privacidade, a integridade e a autenticidade dos dados. Na busca de uma solução para uma comunicação mais segura, a *Netscape* apresentou uma solução para esse problema: o protocolo SSL (*Secure Sockets Layer*).

Esse protocolo (SSL) foi incorporado nos servidores *Web* e *browsers,* tornando-se um padrão *de fato.* Atualmente, encontra-se padronizado pela IETF (Internet Engineering Task Force), através da RFC-2246 (Dierks; Allen, 1999) referenciado como TLS (Transport Layer Security).

Basicamente, o SSL fornece um método de criptografia no nível de soquete, com uso de tecnologia simétrica e assimétrica, esse está entre a camada de aplicação e a camada de transporte. O SSL é formado por três protocolo situados, dois deles, a nível de aplicação e, o terceiro, entre o protocolo de aplicação e o TCP, como apresentado na Figura 4. Seu objetivo é prover um canal seguro entre os pares envolvidos na comunicação, garantindo a integridade da mensagem.

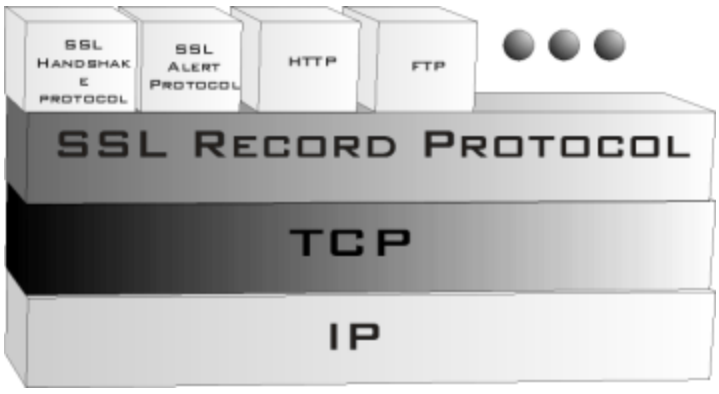

Figura 4: Modelo de camadas SSL – Fonte: (Araújo, 2004)

Ao estabelecer a conexão, o *SSL Handshake Protocol* estabelece um identificador de sessão, um conjunto criptográfico (*cypher suite*) a ser adotado e um método de compressão a ser utilizado. O conjunto criptográfico do SSL constitui-se de três algoritmos:

1. Algoritmo para troca de chaves
2. Algoritmo para cifragem de dados
3. Algoritmo para inserção de redundância nas mensagens.

O algoritmo para troca de chaves é um algoritmo de criptografia de chave pública que será utilizado para enviar uma chave privada do algoritmo de cifragem de dados. Assim, o SSL utiliza-se de um algoritmo assimétrico apenas para criar um canal seguro para enviar uma chave secreta, a ser criada de forma aleatória e que é utilizada para cifrar os dados utilizando-se de um algoritmo simétrico. O algoritmo simétrico é utilizado para efetivamente cifrar os dados da camada de aplicação. Sua escolha é adequada por ser, em geral, mais rápida do que os assimétricos. Por fim, o algoritmo de inserção de redundância é utilizado para garantir a integridade da mensagem. Note-se aqui que o SSL não utiliza nenhum algoritmo específico, mas estabelece, em função dos algoritmos implementados no cliente e no servidor, qual o conjunto comum aos dois para implementar os três papéis necessários para criar-se um canal seguro.

Feito o *handshake* inicial, tem-se um canal que faz uso de um algoritmo simétrico de criptografia e um algoritmo de inserção de redundância na mensagem (chamada de *MAC, Message Autentication Code*). As mensagens do protocolo de aplicação são então comprimidas, inseridas as MACs e então cifradas antes de serem envidadas ao TCP. No destino, após a mensagem ser decifrada, a autenticidade da mensagem é verificada, comparando-a com a MAC, quando então ela é descomprimida e envida para a camada de aplicação. Exceto pelo fato de ter que iniciar o *handshake* do SSL e enviar as mensagens via *SSL Record Protocol,* nada mudou para os protocolos de aplicação.

Outro ponto interessante do SSL é a flexibilidade de poder ser implementada em qualquer protocolo de aplicação que utilize o TCP. Há ainda o *SSL Alert Protocol* que enviar e receber eventuais mensagens de erro, e se necessário, interromper a conexão.

2.4 Protocolos para VPN - *Virtual Private Network*

As bases fundamentais para o conceito de VPN's estão ligadas diretamente a duas tecnologias: tunelamento (túnel) e criptografia. O tunelamento tem a função de encapsular e transmitir os dados, sobre uma rede privada ou pública, entre dois pontos distintos. E a criptografia é utilizada para garantir a integridade, autenticidade e confidencialidade dos dados e principalmente das conexões envolvidas, formando assim uma base para a segurança das soluções VPN's.

A grande necessidade e a carência de segurança na comunicação remota VPN, levou a especificação e criação de diferentes padrões. Em alguns casos é possível integrar soluções com objetivo de alcançar uma situação eficiente para esses acessos.

Essas diferentes especificações geraram uma serie de protocolos, os quais atuam em camadas distintas do modelo ISO/OSI. Esses também tinham formas de tunelamento e tipos de criptografia diferentes. O que influencia diretamente no nível de segurança do acesso remoto com VPN's.

Alguns dos protocolos utilizados para VPN's são: PPTP, L2TP, IPSEC e SSL/TLS. Esses protocolos estão presentes em diverso sistema operacionais. Eles também possuem níveis de segurança distintos e alguns deles são usados com padrão por sistemas operacionais. Nas sessões a seguir, serão apresentados mais detalhes de cada um deles.

2.4.1 O protocolo PPTP

O PPTP (*Point-to-Point Tunneling Protocol*) é um protocolo de camada 2 referente ao modelo isso/OSI, é foi regulamentado pela RFC-2637 (Hamzeh; Pall; Verthein; Taarud; Little e Zorn, 1999). Ele foi originalmente criado por um grupo de empresas chamado *PPTP Forum*, constituído pela 3Com, *Ascend Communications, Microsoft, ECI Telematics e US Robotics.*

Esse grupo de empresas criou uma série de ferramentas para auxiliar no processo de comunicação segura. Para a criptografia o PPTP utiliza o MPPE (*Microsoft Point-to-Point Encryption*), e o tráfego, tem como base o protocolo PPP (*Point-to-Point Protocol).*

O PPTP não possui nativamente serviços de criptografia, ele pega os dados já criptografados pelo MPPE e faz o seu encapsulamento, como apresentado na Figura 5.

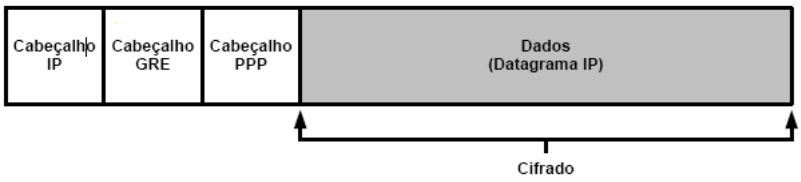

Figura 5: Encapsulamento de um datagrama IP feito pelo PPTP – Fonte: (Rezende, 2004)

Mas no PPTP, o encapsulamento é feito em vários níveis. Em primeiro lugar, o PPP é encapsulado pelo protocolo GRE (*Generic Routing Encapsulation*) que também foi desenvolvido pelo *PPTP Forum.* Continuando o processo, o GRE é encapsulado pelo protocolo IP. O GRE também pode ser usado pelo protocolos IPX e NetBIOS (Rezende, 2004).

A simplicidade de configuração e o suporte por uma diversidade de sistema operacionais são características marcantes para o PPTP.

2.4.2 O protocolo L2TP

Como definido em sua nomenclatura, o protocolo L2TP (*Layer 2 Tunneling Protocol)* como o PPTP, também é um protocolo de camada 2 em relação ao modelo OSI e regulamentado pela RFC-2661 (Townsley; Valencia; Rubens; Pall; Zorn e Palter, 1999). Ele nada mais é que uma combinação do PPTP, com o protocolo de para VPN's da CISCO, o L2F (*Layer 2 Forwarding*).

Diferentemente do PPTP, o L2TP, encapsula o PPP para protocolos X.25, Frame Relay e ATM (*Asynchronous Transfer Mode*) além claro do próprio IP. A criptografia dos pacotes L2TP acontece antes do fechamento da conexão PPP, contrário do que acontece no PPTP que criptógrafa tudo depois do fechamento da conexão.

Na comunicação como L2TP, os pacotes envolvidos nesse processo também sofrem uma série de encapsulamentos, mas o mais importante e que possibilita o aumento na segurança é o encapsulamento IPSec (*Internet Protocol Security*) que implementa a autenticação, a integridade dos dados. O IPSec será apresentado na sessão a seguir.

2.4.3 O protocolo IPSEC

O IPSec (*Internet Protocol Security*) se baseia em três pilares: prover a privacidade, integridade e autenticidade no tráfego de dados IP. Ou seja, trazer segurança de qualidade com criptografia, para a camada IP ou camadas superiores. Para isso são definidas uma série de diretrizes de segurança (Ortiz; Ferreira, 2003).

Sua estrutura esta definida pela RFC-2411 (Thayer; Doraswamy e Glenn, 1998), e para esses serviços são implementados uma da arquitetura de protocolos de segurança de tráfego de dados, de autenticação de cabeçalho AH (*Authentication Header*), de encapsulamento seguro do conteúdo dos dados ESP (*Encapsulating Security*

Payload) e de procedimentos e protocolos de gerência de chaves IKE (*Internet Key Exchange*). Essa arquitetura pode ser observada na Figura 6 (RFC-2411, 1998) .

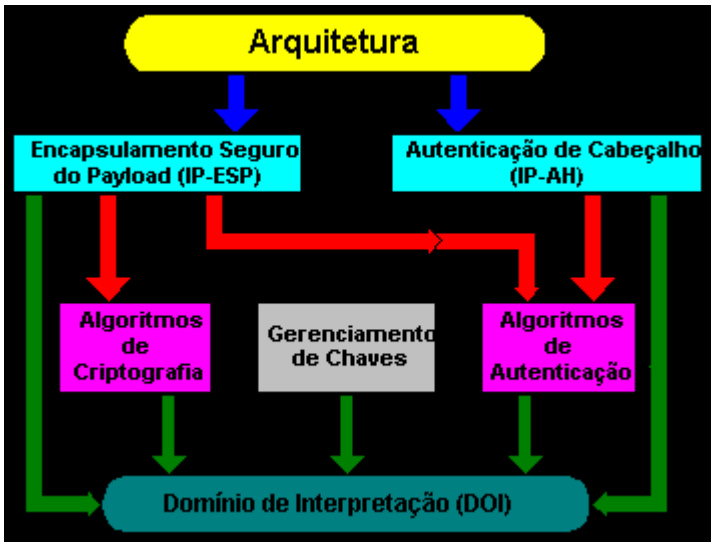

Figura 6: Arquitetura do IPSec -Fonte: (Araújo, 2004)

No IPSec pode-se trabalhar com 2 modos de envio de dados entre os *hosts:*

Modo de Transporte- A segurança e implementada individualmente para em cada *host que esta comunicando.*

Modo de Túnel- A segurança e implementada por *gateway,* ou seja, dois *hosts* fecham um túnel IPSec e o trafego entre os demais computadores da rede passa por esse túnel.

Na arquitetura apresentada na figura 6, o protocolo AH, é responsável pela autenticação e integridade, ou seja, garante a autenticidade do pacote e também que este não foi alterado durante a transmissão. O AH pode ser usado no modo transporte ou no modo túnel, como descrito anteriormente. E sua utilização evita ataques do tipo: roubo de conexão, *spoofing (*mascaramento de IP*)* e *replay* (replica o pacote e reenvia, entrando na comunicação) (Scrimger; LaSalle; Parihar; Gupta, 2002).

O protocolo ESP é responsável pela autenticação e confidencialidade, garantindo que somente os destinatários autorizados terão acesso ao conteúdo do

pacote, ele também pode ser usado no modo transporte ou no modo túnel. E sua implementação previni ataques do tipo: *replay e Sniffer* (farejar o trafego na rede)

Já o ESP provê a cifragem dos dados, para garantir que somente o destinatário possa ler o *payload* do pacote IP. Opcionalmente, também pode garantir a autenticidade e a integridade do pacote, e proteção contra ataques de *replay*.

Os protocolos AH e ESP fazem parte da estrutura IPSec, possuem funcionalidades distintas mas podem trabalhar em conjunto aumentando a segurança.

> "Os dois cabeçalhos podem ser utilizados separadamente ou podem ser combinados para prover as características de segurança desejadas para o tráfego IP. A principal diferença entre os serviços de autenticação e integridade providos pelo AH e pelo ESP está na abrangência da proteção. O AH protege todos os campos de um pacote, exceto aqueles cujos valores são alterados em trânsito. Quando oferecidos pelo ESP, esses serviços abrangem somente o próprio cabeçalho do ESP e a porção de dados do pacote" (Rezende, 2004).

Por se estruturar em várias camadas de segurança, o IPsec é um protocolo que exige um maior nível de dificuldade de implementação e também um poder maior de processamento.

2.5 Conexão ADSL

A tecnologia ADSL (*Asymmetric Digital Subscriber Line*) pertence a um ramo da família DSL (*Digital Subscriber Line*). Que é uma tecnologias desenvolvida para prover serviços de dados de alta velocidade utilizando como meio físico, pares de fios de cobre, ou seja, o par de fios usados pelo telefone.

Essa tecnologia tem como principal característica aproveitar a infra-estrutura existente das companhias telefônicas para resolver o problema do acesso, possibilitando a prestação de serviços de dados com baixo custo de implantação. Popularmente, muitos referem sobre ADSL como uma linha, mas na verdade, está

associado à modems que convertem o sinal padrão do fio de telefone par-trançado em um duto digital de alta velocidade.

Em ADSL os dados são transmitidos de forma assimétrica. A taxa de transmissão na direção central telefônica para assinante é maior, de 256 Kbps até 9 Mbps de *download,* de acordo com a distância da Central . Em contra partida, do assinante para a central, a transmissão é de16 kbps até 640 kbps de *upload.* Comparado com outros sistemas de transmissão de dados atuais, consegue atingir velocidades muito satisfatórias a um custo baixo.

O padrão ADSL funciona com um *modem* instalado no assinante, enquanto um outro modem é instalado na central telefônica. Estes dois *modems* estão permanentemente conectados. O modem divide digitalmente a linha telefônica em 3 canais separados. O primeiro canal é utilizado para transmissão de voz. O segundo canal é utilizado para o fluxo de informações no sentido assinante -> central (*upstream*) e o terceiro canal para o fluxo de dados no sentido central -> assinante (*downstream*). Esta técnica permite maiores velocidades porque raramente as pessoas fazem o mesmo número de *uploads* e *downloads*. Isto significa que o canal de *downstream* pode ser mais largo sem afetar a velocidade de transmissão de dados.

Ao invés de utilizar a infra-estrutura já existente, um novo cabeamento de outra tecnologia poderia ser lançado. Porém, levaria décadas para atingir todos os assinantes, que já dispõem de telefones em suas residências, escritórios e empresas.

A ADSL está sendo utilizada pela maioria das operadoras de serviço PSTN (telefonia fixa comutada) no Brasil. Provendo serviço de banda larga para acesso a Internet, em que os assinantes dispõem de uma conexão permanente. O Velox da Telemar e o Speedy da Telefônica são exemplos desse tipo de serviço.

2.5.1 O Protocolo PPPoE

O PPPoE (*Point-to-Point over Ethernet*) é um protocolo relativamente novo, especificado em fevereiro 1999 pela RFC-2516 (Manakos; Lidl; Evarts; Carrel; Simone e Wheeler, 1999). E é através desse protocolo que se autentica usuário para o acesso a *link* ADSL. Como visto na sessão anterior, a ADSL se trata apenas do meio físico de conexão sendo necessário um protocolo para encapsular os dados de seu computador até a central telefônica.

O protocolo PPPoE trabalha com a tecnologia PPP sobre a Ethernet, Figura 7, que é usada para ligar a placa de rede do computador ao modem, permitindo compressão e autenticação para a conexão e aquisição de um endereço IP válido para a navegação.

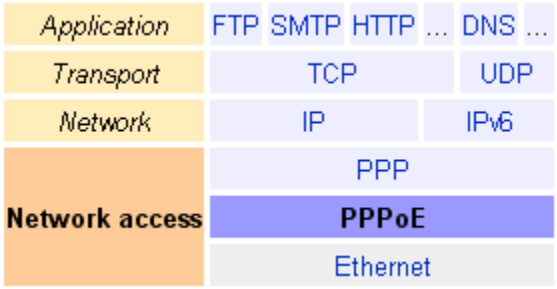

Figura 7: Modelo de camadas PPPoE - Fonte: (Point-to-Point Protocol over Ethernet, 2005)

Como grandes companhias de telecomunicações oferecem a ADSL, e precisão também do controle de acesso do assinante desse serviço. Com essa autenticação as operadoras conseguem identificar o usuário conectado e controlar suas ações.

A ANATEL, regulamentadora das telecomunicações no Brasil, não autoriza que operadoras de telecomunicações provêem acesso diretamente a Internet, sendo a função delegada a provedores de Internet licenciados. Por esse motivo o padrão ADSL brasileiro ainda inclui uma autenticação adicional, mesmo após a autenticação PPPoE,

para liberar a conexão à Internet. E esta autenticação é realizada por um provedor de acesso onde do qual se contrata um serviço adicional, apenas para ser liberado o caminho entre o assinante e a Internet.

2.5.2 O Protocolo PPPoA

O PPPoA (*Point-to-Point over ATM*) é o protocolo irmão mais velho do PPPoE, especificado em julho 1998 pela RFC-2364 (Gross; Daycel; Lin; Malis e Stephens, 1998). A diferença é que o PPPoA interliga l*inks* ADSL a redes ATM (*Asynchronous Transfer Mode*), uma tecnologia de comunicação de dados de alta velocidade usada para interligar redes locais, metropolitanas e de longa distância para aplicações de dados, voz, áudio, e vídeo.

Basicamente a tecnologia ATM fornece um meio para enviar informações em modo assíncrono através de uma rede de dados. A tecnologia ATM utiliza o processo de comutação de pacotes, que é adequado para o envio assíncrono de informações com diferentes requisitos de tempo e funcionalidades, aproveitando-se de sua confiabilidade, eficiência no uso de banda e suporte a aplicações que requerem classes de qualidade de serviço diferenciadas.

2.6 DNS -Sistema de Nomes de Domínio

Antes de 1980, a ARPANET tinha somente uma poucas centenas de computadores em rede. O mapeamento de endereço de nome de computador (*hostname*) era contido em um único arquivo chamado hosts.txt. Este arquivo era armazenado no computador host Centro de Informação de Redes da Instituto de Recursos de Stanford (*SRI-NIC Stanford Research Intitute's Network Information Center)* no Menlo Park, Califórnia. Outros computadores *hosts* na ARPANET copiavam o arquivo hosts.txt a partir do SRI-NIC para os seus *sites* se fosse necessário.

Inicialmente, este esquema trabalhava bem por que a lista hosts.txt precisava ser atualizada somente uma ou duas vezes por semana. Entretanto, em poucos anos, apareceram problemas devido ao tamanho sempre crescente da ARPANET. Os problemas incluíam os seguintes:

- O arquivo hosts.txt se tornou muito grande.
- O arquivo precisava ser atualizado mais que uma vez por dia.
- Pelo motivo de todo o tráfego de rede ter de ser roteado através do SRI-NIC, manter o hosts.txt se tornou um ponto de restrição para a rede toda.
- O arquivo hosts.txt usa uma estrutura *flat name* (espaço de nome). Isto exigiu que todos os nomes de computadores fossem únicos para toda a rede.

Estes e outros problemas conduziram o corpo de governantes da ARPANET a encontrar uma solução para o mecanismo que envolvia o arquivo hosts.txt. A decisão conduziu para a criação do DNS (*Domain Name System*), que é um banco de dados distribuído usando uma estrutura de nome hierárquico e descrito na RFC-1034 (Mockapetris, 1987) e RFC-1035 (Mockapetris, 1987).

O DNS pode ser comparado a um catálogo de telefone. O usuário procura o nome da pessoa ou organização que ela quer contatar e cruza a referência do nome para o número do telefone. Igualmente, um computador *host* contata o nome de um computador e um servidor de nome de domínio cruza a referência do nome para um endereço.

O sistema de nome de domínio (DNS) é um sistema de gerenciamento de banco de dados distribuído baseado na hierarquia cliente/servidor como apresentado na figura 8. O propósito do banco de dados DNS é traduzir nomes de computadores em endereços IP. No DNS, os clientes são chamados *resolvers* e os servidores são

chamados de *nameservers.*

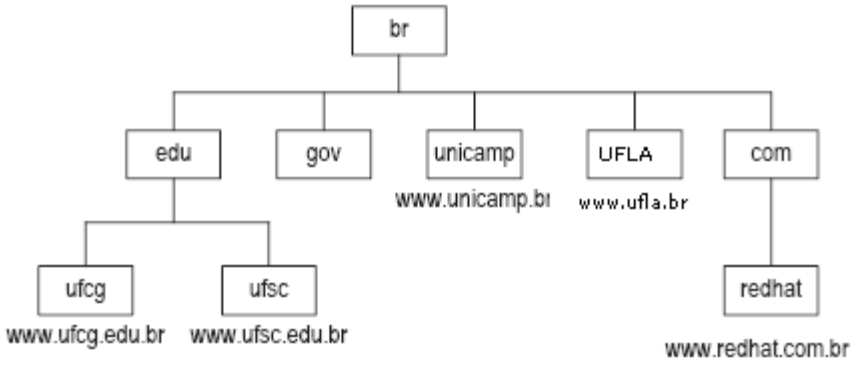

Figura 8: Árvore de Domínio - Fonte: (Uchôa;Simeone;Sica, 2003)

O DNS é igual a um catálogo de telefone. O usuário procura o nome da pessoa ou organização que ela quer contatar e cruza a referência do nome para o número do telefone. Igualmente, um computador *host* contata o nome de um computador e um servidor de nome de domínio cruza a referência do nome para um endereço IP.

2.6.1 DDNS - Sistema de Nomes de Domínio Dinâmico

Quando no acesso Internet um computador tem um IP fixo, o endereço IP é sempre o mesmo; outras pessoas/serviços que conheçam esse endereço IP sabem como encontrar o servidor. Pelo contrário quando o endereço IP varia de ligação para ligação, as outras pessoas e/ou serviços têm dificuldade em encontrar o servidor na Internet.

Na grande maioria dos casos, o acesso Internet, principalmente *links* ADSL faz-se via IP variável, e não por IP Fixo. As empresas de fornecimento de acesso Internet cobram diferentes taxas de serviço para IP fixo, normalmente muito mais caros.

Para resolver o problema de alto custo de IP's fixos há uma solução, o serviço de DNS Dinâmico. Versões mais novas de *software* de DNS trazem uma característica interessante, a atualizações dinâmicas, que tem a capacidade de modificar dados de uma zona de DNS sem precisar editar manualmente os arquivos. Simplesmente um modo associar um nome de máquina a um endereço de IP dinâmico.

No serviço de DNS dinâmico, pode se conseguir um nome estático para se ligar a aquele endereço IP variável. A única coisa a se fazer é, sempre se manter se conectado. Desta forma, o *hostname* sempre estará visível para os demais *hosts* na Internet.

2.7 O serviço No-IP

O No-IP é uma implementação de um serviço de DNS dinâmico que está disponível gratuitamente através do *site* www.No-IP.com. O serviço é bastante eficiente e simples.

Para seu usado é necessário efetivar um cadastro no *site* do No-IP. É registrado então um subdomínio, e também se instala no *host,* um programa cliente, que irá informar periodicamente ao serviço No-IP o número do IP que estamos utilizando em determinado momento.

Deve-se ainda configurar o programa, para trabalhar com o acesso *on-line* o qual possibilita a comunicação transparente entre cliente e servidor DDNS.

Além do No-IP outros serviços de DNS dinâmico estão disponíveis gratuitamente como o DynDNS www.dyndns.com e o Agente DDNS www.winconnection.com.br. Mas esses não trazem a diversidade de domínios gratuitos disponíveis pelo No-IP.

Capítulo 3

3.1 Implementando VPN em Linux com PPPOE, No-IP e SSL

Nesse capítulo serão apresentados os processos de instalação e configuração das aplicações envolvidas na solução de interligação entre matriz e filiais de empresas com um baixo custo usando o Sistema Operacional Linux. Para implementação da infra-estrutura para o *links* de Internet banda larga ADSL, o PPPoE será usado juntamente com o serviço gratuito de DNS dinâmico No-IP. E para estabelecer a VPN em modo túnel entre os pontos, o SSL/TLS foi escolhido.

Toda essa solução se baseia na distribuição Linux Mandriva Power Pack 2006 plus, que já deve estar instalada. Mas pode ser facilmente adaptada para qualquer outra distribuição Linux.

3.1.1 Instalando ADSL com PPPoE

Inicialmente será necessário ativar o serviço de banda larga nos servidores que fecharam o túnel VPN's. Para isso serão instalados os pacotes ppp-2.4.3-9mdk e rp-pppoe-3.5-5.mdk, esse último trás os aplicativos necessários para a configuração. Esse processo deve ser repetido no servidor VPN da matriz e também na filial. O superusuário deve executar os comandos a seguir:

```
# urpmi ppp
# urpmi rp-pppoe
```

Após instalados os software e ainda como superusuário, o comando adsl-setup será executado para configurar a conexão com o serviço de banda larga Velox, da operadora Telemar. Outros serviços com Speedy da Telefonica, Turbonet da GVE e

Turbo da Brasil Telecon também trabalham com essa tecnologia. No nosso ambiente de configuração e implementado na cidade de Belo Horizonte a única operadora disponível era a Telemar.

 # adsl-setup

Esse comando ativará o menu de configurações apresentado a seguir:

Welcome to the Roaring Penguin ADSL client setup. First, I Will run some checks on your system to make sure the PPPoE client is installed property...

Looks good! Now, please enter some information:

USER NAME

Enter your PPPoE user name (default xxx@xxx.xxx)

Nessa sessão deve ser definido o *login* de autenticação com o provedor.

Exemplo: telemar@3134743666

A seguir, deve ser definido a interface *ethernet* que esta ligada ao *modem* ADSL. Caso seja /dev/eth0 basta teclar [*ENTER*]

INTERFACE

Enter the Ethernet interface connected to the ADSL modem.
For Solaris, this is likely to be something like /dev/hme0.
For Linux, it will be ethn, where 'n' is a number.
(default eth0):

A próxima opção tem relação com o a conexão por demanda. Se escolhido [*yes]* a conexão será sob demanda, se [*no*] conexão permanente.

Do you want the link to come up on demand, or stay up continuously? If you want it to come up on demand, enter the idle time in seconds after which the link should be dropped. If you want the link to stay up permanently, enter 'no' (two letters, lower-case).
NOTE: Demand-activated links do not interact well with dynamic IP addresses. You may have some problems with demand- activated links.
Enter the demand value (default no):

Agora, será configurado o serviço de resolução de nomes DNS. Serão definido os DNS primário e secundário, ou caso se deseje obter essa configuração direto da operadora basta utilizar a opção [*server*] e [*ENTER*].

DNS

Please enter the IP address of your ISP's primary DNS server. If your ISP claims that 'the server will provide DNS addresses', enter 'server' (all lower-case) here. If you just press enter, I will assume you know what you are doing and not modify your DNS setup.
Enter the DNS information here:

Em seguida a senha para a autenticação com o provedor é solicitada. É necessário digitar a confirmação da senha.

PASSWORD

Please enter your PPPoE password:

Please re-enter your PPPoE password:

Na etapa seguinte, serão ofertadas possibilidades do uso de regras de *firewall* previamente configuradas através de três opções:

Número 0 - não ativa *firewall*.

Número 1 - ADSL conectando apenas um computador a Internet.

Número 2 - ADSL conectando o computador que compartilha acesso a Internet para toda rede (*gateway*).

FIREWALLING

Please choose the firewall rules to use. Note that these rules are very basic. You are strongly encouraged to use a more sophisticated firewall setup; however, these will provide basic security. If you are running any servers on your machine, you must choose 'NONE' and set up firewalling yourself. Otherwise, the firewall rules will deny access to all standard servers like Web, e-mail, ftp, etc. If you are using SSH, the rules will block outgoing SSH connections which allocate a privileged source port.
The firewall choices are:
0 - NONE: This script will not set any firewall rules. You are responsible for ensuring the security of your machine. You are STRONGLY recommended to use some kind of firewall rules.
1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation
2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway

for a LAN
Choose a type of firewall (0-2):

Dando continuidade paraa proxima opção, ela definira se a conexão ADSL será ativa durante a inicialização do computador ou não. Opções [yes] ou [no]

Start this connection at boot time. Please enter no or yes:

Na última etapa de configuração do PPPoE, serão apresentados as opções escolhidas e se tudo estiver correto basta escolher a opção [*y*] para confirmar. Se não [*n*] para refazer as configurações.

Ethernet Interface: eth0
User name: telemar@3134743666
Activate-on-demand: No
DNS addresses:Supplied by ISP's server
Firewalling: MASQUERADE
Accept these settings and adjust configuration files (y/n)?

Caso seja confirmado as configurações, aparecerão as congratulações pela configurações e também os comando que podem ativar ou para a conexão PPPoE. São eles:

 # adsl-start (para conectar)

 # adsl-stop (para desconectar)

Adjusting /etc/ppp/pppoe.conf
Adjusting /etc/ppp/pap-secrets and /etc/ppp/chap-secrets
(But first backing it up to /etc/ppp/pap-secrets-bak)
(But first backing it up to /etc/ppp/chap-secrets-bak)

Congratulations, it should be all set up!

Type 'adsl-start' to bring up your ADSL link and 'adsl-stop' to bring it down. Type 'adsl-status' to see the link status.

Todas a configurações geradas pela ferramenta “adsl-setup”, são escritas no arquivo /etc/ppp/pppoe.conf e que podem ser diretamente manipuladas por esse arquivo de configuração.

3.1.2 Instalando o DNS Dinâmico No-IP

Como o *link* ADSL recebe endereço IP dinamicamente, e esse pode ser trocado a qualquer momento, isso inviabilizaria o fechamento do túnel VPN entre a matriz e filial. Para resolução desse problema, um serviço de DDNS gratuito como o No-IP (www.No-IP.com) será utilizado para associar a essas máquinas um nome de domínio.

Para uso desse serviço, primeiramente deve se fazer um cadastro no site "http://www.No-IP.com/newUser.php", como mostra a Figura 9.

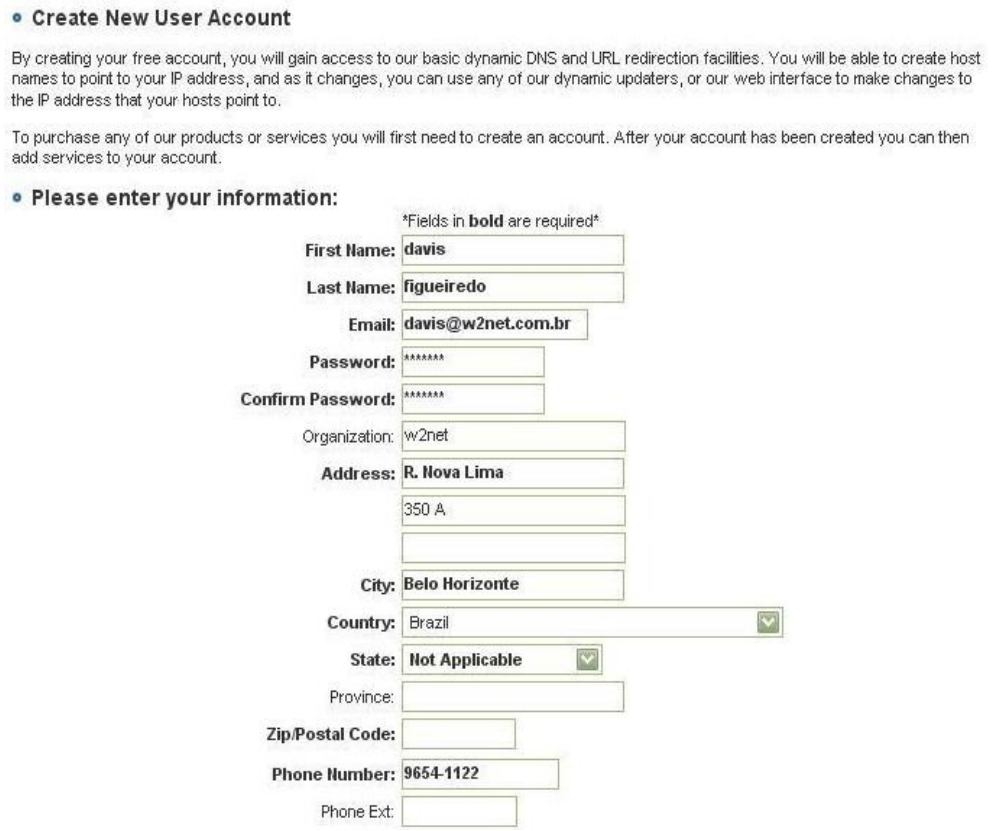

Figura 9: Criando conta no DDNS No-IP

Após o preenchimento do cadastro deve ser aceitos os termos de prestação de serviço pelo No-IP e confirmado a assinatura, como mostra a Figura 10.

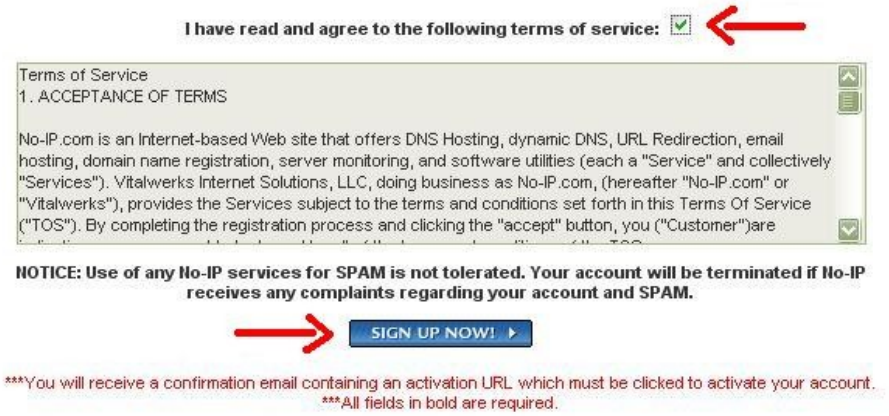

Figura 10: Aceitação dos termos e confirmação da assinatura.

Se todos os dados estiverem corretos, uma tela de congratulações aparecerá em seguida. Nela também esta indicando que será enviado um *email* para conta cadastrada, para que se possa confirmar o desejo do cadastro, como apresentado nas Figura 11 e 12.

Figura 11: *Email* do No-IP

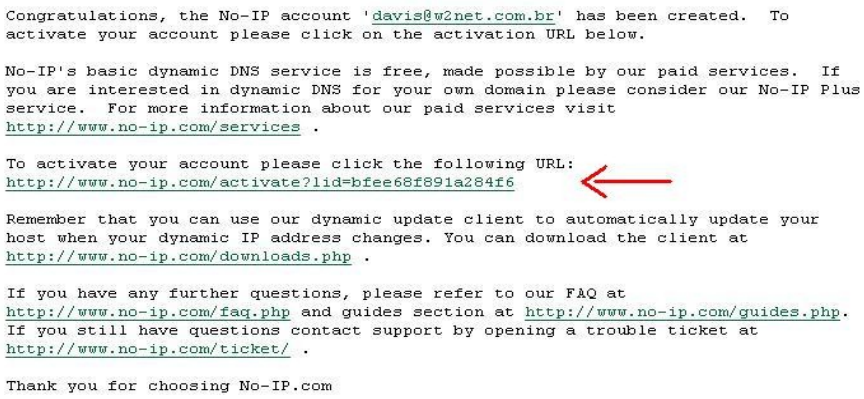

Figura 12: Ativação da conta No-IP

Com a conta No-IP cadastrada, basta acessar o "site www.No-IP.com" e fazer o *login* com a conta de *email* e senha cadastrados, como mostrado na Figura 13.

Figura 13: *Login* no No-IP

No serviço No-IP, estão disponíveis uma série de recursos e vários tipos de registro, mas no caso de uma simples resolução de nome de domínio para endereço IP para a um túnel VPN, basta adicionar um registro do tipo "A" como Figura 14.

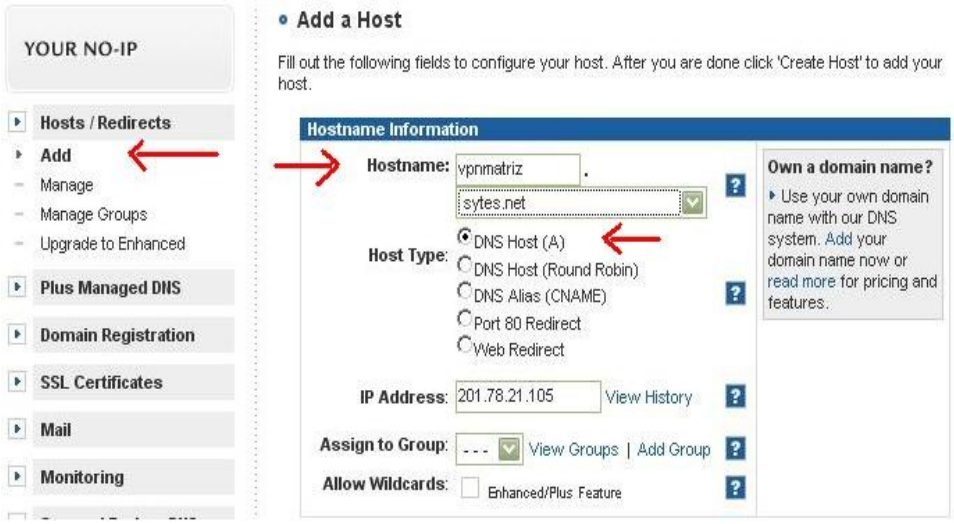

Figura 14: Adicionando um nome ou *host*

Como pode ser observado na figura 14, foi registrado um *hostname* "vnpmatriz" com um domínio cedido pelo No-IP "sytes.net" ou seja, o nome completo ficou "vpnmatriz.sytes.net" vinculado ao IP 201.78.21.105. O *hostname* e meramente sugestivo podendo ser usado qualquer outro que não esteja em uso.

O processo de criação deve ser repetido para o *host* que será identificado como filial. Ao final, quando for acessado no *menu* a opção *manage* aparecerá os 2 registros, semelhantes a Figura 15.

Figura 15: Registro dos *hosts* Matriz e Filial

Terminado esse processo, tanto a máquina vpnmatriz quanto a vpnfilial, poderão ser localizadas pelos respectivos nomes e os servidores de DNS do No-IP ficaram responsáveis pela resolução IP. Mas isso ainda não é o suficiente para uma solução automatizada. Caso o endereço de um dos pontos seja modificado, o DNS não terá a possibilidade de resolver.

Para solucionar essa pendência será necessário baixar e instalar um aplicativo cliente No-IP como figura 16 , que tem a função de monitorar a troca do IP. E caso aconteça uma troca, o programa cliente avisar ao servidor qual o novo endereço para aquele nome.

Figura 16: *Download* cliente No-IP para Linux

A software cliente No-IP deve ser desempacotado e descompactado e seu binário "noip2-Linux", pode se copiado para o diretório /sbin como demostrado a seguir.

```
# tar -zxvf noip-duc-linux.tar.gz
# cd noip-2.1.3
# cd binaries
# cp -a noip2-Linux /sbin
```

Em seguida o arquivo de configuração /etc/noip.conf será criado. Para isso a executação do comando apresentado a seguir é necessário. Nesta etapa ele pedirá a interface que esta conectada ao *modem*, caso exista mais de uma, login do usuário, a senha e o domínio registrado no *site* do serviço No-IP que vai ser vinculado a esse IP. Isso é usado para gerar o arquivo de configuração /etc/noip.conf

```
# noip2-Linux -C -c /etc/noip.conf
```

Auto configuration for Linux client of No-IP.com.

Multiple network devices have been detected.

Please select the Internet interface from this list.

By typing the number associated with it.
0 eth1
1 ppp0
1
Please enter the login/email string for No-IP.com
davis@W2net.com.br
*Please enter the password for user davisfigueiredo@gmail.com' ************

2 hosts are registered to this account.
*Do you wish to have them all updated?[N] (y/N) **ENTER***
Do you wish to have host [vpnmatriz.sytes.net] updated?[N] (y/N) ***ENTER***
Do you wish to have host [vpnfilial.sytes.net] updated?[N] (y/N) ***y***
*Do you wish to run something at successful update?[N] (y/N) **ENTER***

New configuration file '/etc/noip.conf' created.

E para finalizar o configuração, a seguinte linha com o comando "*# noip2-Linux -c /etc/noip.conf -U 1 &*" que faz a verificação de 1 em 1 minuto da configuração IP da máquina, deve ser digitada no arquivo de inicialização do sistema "/etc/rc.local". Isso se deve a necessidade de que toda vez que o equipamento for iniciado ou ocorrer uma pane de energia, o No-IP seja ativado e permaneça monitorando a interface de rede.

3.1.3 Criando a VPN com OpenVPN

O OpenVPN é um robusto e altamente flexível serviço que possibilita a implementação de VPN's suportando os protocolos SSL/TLS, túneis TCP e UDP através *proxy* ou NAT (*Network Address Translation*) e padronizado pela grande maioria dos sistema operacionais do mercado. (Yonan, 2006).

Ele implementa também segurança de rede a nível das camadas 2 e 3 do modelo OSI (*Open System Interconnection*). Sua instalação e configuração é bem simples é o pacote opevpn-2.0.1-2mdk deverá ser instalado para a configuração do túnel VPN. O pacote libzo2_2-2.01-1mdk também será útil no processo de compressão dos pacotes que serão transmitidos pela VPN. (Yonan, 2006).

```
# urpmi openvpn
# urpmi liblzo2
```

A instalação dos 2 pacotes deve ser feita nos dos pontos da VPN. O ambiente, como já dito anteriormente, propõe a ligação da rede de 2 empresa uma matiz (Web Manegatti) a sua filial (W2net). Cada uma das empresas possuem uma rede e um *gateway, que conseqüentemente será a* VPN. Os computadores que serão usados como *gateway*, possuem duas placas de rede, uma para rede interna, denominada eth1 e outra eth0 ligada ao *modem* ADSL. A VPN usará chave simétrica para o processo de criptografia. E o comando para criação da chave é o apresentado a seguir:

```
# openvpn --genkey --secret /etc/openvpn/matriz.key
```

Na seqüência de configuração, será feita uma copia exemplo do arquivo de configuração disponível pelo próprio pacote openvpn para o arquivo principal chamado matriz.conf. E em seguida nesse arquivo serão feitas implementações para a VPN proposta.

```
# cp /usr/share/openvpn/sample-config-files/static-office.conf
      /etc/openvpn/matriz.conf
```

O arquivo /etc/openvpn/matriz.conf deverá ser configurado com os seguintes parâmetros:

 # Sample OpenVPN configuration file for
 # office using a pre-shared static key.
#
 # '#' or ';' may be used to delimit comments.

 # Use a dynamic tun device.
 # For Linux 2.2 or non-Linux OSes,
 # you may want to use an explicit
 # unit number such as "tun1".
 # OpenVPN also supports virtual
 # ethernet "tap" devices.
dev tun (modo de operação da VPN, modo túnel)

 # 10.1.0.1 is our local VPN endpoint (office).
 # 10.1.0.2 is our remote VPN endpoint (home).
ifconfig 10.1.0.2 10.1.0.1 (IP`s virtuais para Matriz e filial)

 # Our up script will establish routes
 # once the VPN is alive.
up ./office.up (ativa rotas para a rede interna)

 # Our pre-shared static key
secret matriz.key (arquivo com a chave simétrica para a criptografia)

 # OpenVPN 2.0 uses UDP port 1194 by default
 # (official port assignment by iana.org 11/04).
 # OpenVPN 1.x uses UDP port 5000 by default.
 # Each OpenVPN tunnel must use
 # a different port number.
 # lport or rport can be used
 # to denote different ports
 # for local and remote.
port 1194 (Porta padrão para conexão na VPN)

 # Downgrade UID and GID to
 # "nobody" after initialization
 # for extra security.
user nobody (usuário administrador da vpn)
group nobody (grupo administrador da vpn)

 # If you built OpenVPN with
 # LZO compression, uncomment
 # out the following line.
comp-lzo (ativa a compressão dos dados)

 # Send a UDP ping to remote once
 # every 15 seconds to keep
 # stateful firewall connection
 # alive. Uncomment this
 # out if you are using a stateful
 # firewall.
ping 15 (teste de conexão)

 # Uncomment this section for a more reliable detection when a system
 # loses its connection. For example, dial-ups or laptops that
 # travel to other locations.
; ping 15
; ping-restart 45
; ping-timer-rem
persist-tun (manter o sistema sempre conectado)
persist-key

 # Verbosity level.
 # 0 -- quiet except for fatal errors.
 # 1 -- mostly quiet, but display non-fatal network errors.
 # 3 -- medium output, good for normal operation.
 # 9 -- verbose, good for troubleshooting
verb 9 (nível de detalhamento de log`s)

Para o parâmetro "***up ./office.up***" funcionar corretamente e ativar a rota para rede interna o arquivo office.up deverá ser copiado para o diretório /etc/openvpn com o comando a seguir:

 # cp /usr/share/openvpn/sample-config-files/office.up /etc/openvpn/

Para a configuração na filial, primeiramente deverá ser copiado o arquivo /etc/openvpn/matriz.key que foi gerado na matriz para o diretório /etc/openvpn da filial. Como essa chave e única para os dois pontos, a

segurança nessa tarefa é muito importante. Para isso poderá ser usado uma conexão segura SSH para fazer a copia, com no comando apresentado

#scp vpnmatriz.sytes.net:/etc/openvpn/matriz.key /etc/openvpn

A arquivo de configuração da filial é muito semelhante ao da matriz, então, também será feita uma copia exemplo do arquivo de configuração disponível pelo próprio pacote OpenVPN para o arquivo principal chamado filial.conf. E em seguida nesse arquivo serão feitas implementações para a VPN.

 # cp /usr/share/openvpn/sample-config-files/static-home.conf /etc/openvpn/filial.conf

As mudanças mais significativa a relação ao arquivo matriz.conf são em relação aos parâmetros: *remote, ifconfig e up.* Os demais são iguais ao anterior, e deverá ser configurado com os seguintes parâmetros:

 # vi /etc/openvpn/filial.conf

```
# Sample OpenVPN configuration file for
# office using a pre-shared static key.
#
# '#' or ';' may be used to delimit comments.

# Use a dynamic tun device.
# For Linux 2.2 or non-Linux OSes,
# you may want to use an explicit
# unit number such as "tun1".
# OpenVPN also supports virtual
# ethernet "tap" devices.
dev tun  (modo de operação da VPN, modo túnel)

# Our OpenVPN peer is the office gateway.
remote vpnmatriz.sytes.net (nome do gateway
                            registrado para matriz)
```

 # 10.1.0.1 is our local VPN endpoint (office).
 # 10.1.0.2 is our remote VPN endpoint (home).
ifconfig 10.1.0.2 10.1.0.1 (IP`s virtuais para filial e matriz)

 # Our up script will establish routes
 # once the VPN is alive.
up ./home.up (ativa rotas para a rede interna)

 # Our pre-shared static key
secret matriz.key (arquivo com a chave simétrica para a criptografia)

 # OpenVPN 2.0 uses UDP port 1194 by default
 # (official port assignment by iana.org 11/04).
 # OpenVPN 1.x uses UDP port 5000 by default.
 # Each OpenVPN tunnel must use
 # a different port number.
 # lport or rport can be used
 # to denote different ports
 # for local and remote.
port 1194 (Porta padrão para conexão na VPN)

 # Downgrade UID and GID to
 # "nobody" after initialization
 # for extra security.
user nobody (usuário administrador da vpn)
group nobody (grupo administrador da vpn)

 # If you built OpenVPN with
 # LZO compression, uncomment
 # out the following line.
comp-lzo (ativa a compressão dos dados)

 # Send a UDP ping to remote once
 # every 15 seconds to keep
 # stateful firewall connection
 # alive. Uncomment this
 # out if you are using a stateful
 # firewall.
ping 15 (teste de conexão)

 # Uncomment this section for a more reliable detection when a system
 # loses its connection. For example, dial-ups or laptops that

 # travel to other locations.
; ping 15
; ping-restart 45
; ping-timer-rem
persist-tun (manter o sistema sempre conectado)
persist-key

 # Verbosity level.
 # 0 -- quiet except for fatal errors.
 # 1 -- mostly quiet, but display non-fatal network errors.
 # 3 -- medium output, good for normal operation.
 # 9 -- verbose, good for troubleshooting
verb 9 (nível de detalhamento de log`s)

Como na matriz, para o parâmetro "***up ./home.up***" funcionar corretamente e ativar a rota para rede interna o arquivo home.up deverá ser copiado para o diretório /etc/openvpn com o comando a seguir:

 # cp /usr/share/openvpn/sample-config-files/home.up /etc/openvpn/

Após a configuração dos dois lados da VPN, deverá ser executado em ambos os lados, iniciando pelo lado da filial os seguintes comandos:

Carga do modulo tun/tap no kernel.

 # modprobe tun

Habilitar o roteamento no sistema.

 # echo 1 > /proc/sys/net/ipv4/ip_forward

Ativar a VPN na filial.

 # openvpn --daemon --config /etc/openvpn/filial.conf

Ativar a VPN na matriz.

 # openvpn --daemon --config /etc/openvpn/matriz.conf

Para testes o túnel, da matriz envie pacotes icpm-request para filial.

 # ping 10.1.0.1

E da filial para matriz

 # ping 10.1.0.2

Se for obtido resposta de ambos os lados a VPN foi estabelecida corretamente. E para terminar a configuração, e automatizar a conexão na inicialização das máquinas, deverá ser copiado os seguintes comandos no final do arquivo /etc/rc.local.

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
modprobe tun
echo 1 > /proc/sys/net/ipv4/ip_forward

# Se for na filial
openvpn --daemon --config /etc/openvpn/filial.conf
# Se for na filial
# openvpn --daemon --config /etc/openvpn/matriz.conf
```

É muito comum a existência de *firewall* em equipamento que implementam esse tipo de funcionalidade. Sendo assim à a necessidade de serem tratadas regras específicas para as conexões VPN com o OpenVPN.

O pacote já traz um *script* de configuração de um *firewall* para esse tipo de implementação, ele se localiza no arquivo */usr/share/openvpn/sample-config-files/firewall.sh*). As regras mais importantes são as seguintes.

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT
```

3.2 Implantação da VPN

A solução de VPN da baixo custo foi colocada em um ambiente de produção, para interligar a rede interna de duas empresas. A Figura 17 representa graficamente a estrutura instalada.

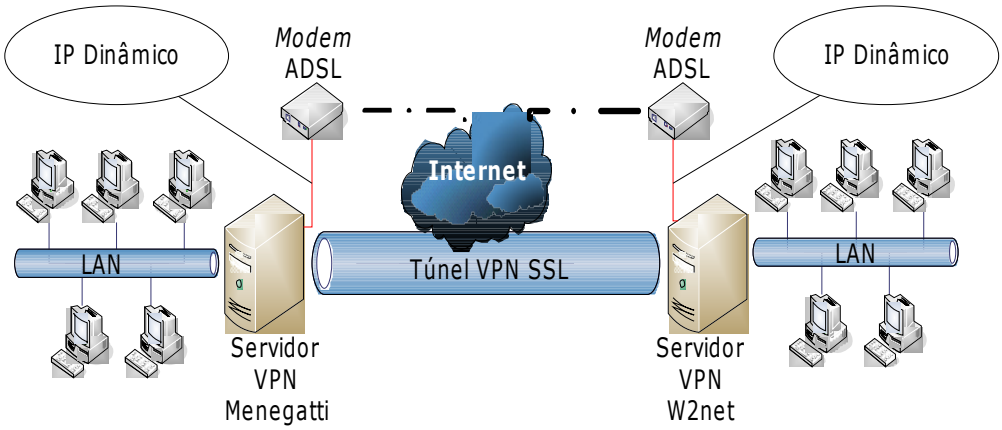

Figura 17: VPN em produção

As especificações de endereçamento, nomes e produtos usados nas empresas estão relatados a seguir:

Matriz

Empresa: Menegatti

Servidor VPN:

Sistema Operacional Linux Mandriva

Interface Local (eth1) IP (192.168.1.254/24)

Interface Publica (eth0) IP (Ip Dinâmico)

Interface Túnel (tun0) IP (10.1.0.1)

Nome registrado no No-IP: vpnmatriz.sytes.net

Rede Local:

Sistema Operacional Windows XP

Faixa de endereços: 192.168.1.50/24 a 192.168.1.150/24

Default Gateway 192.168.1.254

Link ADSL:

Telemar/Velox

Filial

Empresa: W2net

Servidor VPN:

Sistema Operacional Linux Mandriva

Interface Local (eth0) IP (192.168.2.254/24)

Interface Publica (eth1) IP (Ip Dinâmico)

Interface Túnel (tun0) IP (10.1.0.2)

Nome registrado no No-IP: vpnfilial.sytes.net

Rede Local:

Sistema Operacional Windows XP

Faixa de endereços: 192.168.2.50/24 a 192.168.2.150/24

Default Gateway 192.168.2.254

Link ADSL:

Telemar/Velox

As etapas de configuração seguiram a ordem definida nas seções anteriores. Primeiro foi instalado o Linux nos dois servidores e configurado a conexão ADSL. Com a conexão estabelecida e os dos servidores ligados a Internet, a conta No-IP (www.No-IP.com) foi criada e os registros vpnmatriz.sytes.net e vpnfilial.sytes.net foram associados aos IP's. E por fim, foram instalados aos aplicativos cliente No-IP em

ambas as máquinas. Garantindo assim a atualização do IP ao nome sempre que ele for trocado.

Para o fechamento do túnel entre a Menegatti e a W2net os passos apresentados na sessão 3.1.3 foram seguidos. E para que as estações de trabalho possam ser roteadas para o túnel, foram adicionadas rotas, como exposto a seguir.

No servidor da Menegatti o seguinte comando foi executado:

 # route add -net 192.168.2.0/24 gw 10.1.0.2

No servidor da W2net o seguinte comando foi executado:

 # route add -net 192.168.1.0/24 gw 10.1.0.1

Com as rotas definidas alguns testes foram feitos.

Exemplo 1: Na rede da Menegatti, uma estação Windows XP com IP 192.168.1.55, executa o comando ping tentando se comunicar com uma outra estação de IP 192.168.2.51. O mesmo teste e feito de forma contraria, da máquina 192.168.2.51 para192.168.1.55. E como era de se esperar, houve resposta em ambos as lados.

Teste de acesso a compartilhamentos de arquivos também foram feitos e obtiveram sucesso. Mostrando assim a viabilidade do projeto.

Capítulo 4

CONCLUSÃO

O objetivo dessa pesquisa teve como foco principal, encontrar uma solução para conexão segura entre redes de dados, que estivesse financeiramente ao alcance de pequenas e médias empresas.

Foi apresentado na pesquisa, a implementar de um túnel VPN usando o Linux como sistema operacional e o protocolo SSL/TLS para estabelecer a comunicação segura entre as duas pontas. A aplicação usada para isso foi o OpenVPN, que trazia uma configuração simples e eficiente, com portabilidade para outros sistemas não Linux.

Um outro aspecto importante a ser descrito é que, além do custo zero para licença de software. A tecnologia de ligação dos pontos utilizava *links* ADSL mais conhecidos como banda larga. Oferecido por operadoras de telecomunicação, esse *links* tem uma mensalidade muito baixa, cerca de 10% comparado ao um *link* dedicado.

Para se encontrarem na Internet, os *gateway* VPNs receberam nome, pois a ADSL oferta endereços IPs dinamicamente, que podem ser trocados repentinamente ou por quedas de energia. O serviço de DDNS No-IP com seu software de monitorando de IP dinâmico demostrou eficiência e funcionou perfeitamente, mantendo sempre as VPNs.

A solução se mostrou possível, já que pode se testar na prática a interligação de duas empresas, as quais puderam se comunicar permanentemente sobre o túnel sem nenhuma restrição ou problema.

Esse estudo pode mostrar que o Linux juntamente com o software livre, podem proporcionar uma redução de custo e possibilidade de melhoria na qualidade de serviço

usados e prestados por pequenas empresas. Dando a elas uma maior chance de crescimento e competitividade com medias e grande organizações.

Referências Bibliográficas

Andrew S.Tanenbaum. *Redes de Computadores*. CAMPUS, 2001.

Edmar Roberto Santana de Rezende. *Segurança no Acesso Remoto VPN*. 2004. 144 f. Dissertação de Mestrado (Mestrado em Ciência da Computação) – Faculdade de Ciências da Computação, Universidade Estadual de Campinas, Campinas.

Eduardo Bellincanta Ortiz e Ed'Wilson Tavares Ferreira. *VPN Vitual Private Network, Implementando Soluções em linux*. ÉRICA, 2003.

G. Gross; M, Kaycel; A. Lin; A. Malis e J. Stephens. RFC-2364 – *Point-to-Point over ATM*
http://www.ietf.org/rfc/rfc2564.txt?number=2364, acesso em janeiro de 2006.

Gorgonio Araújo. *Transações Seguras via Web*. RNP – Rede Nacional de Ensino e Pesquisa, 2004. Disponível via www em http://www.rnp.br/newsgen/9803/https.html, acesso em março de 2006.

James Yonan. *OpenVPN^{TM} 2.0.x Man Page*, 2006. Disponível em www http://openvpn.net/man.html, acesso em março de 2006.

James Yonan. *OpenVPN^{TM} 2.0. HOWTO*, 2006. *Disponível em www* http://openvpn.net/howto.html, acesso em março de 2006.

Joaquim Quinteiro Uchôa; Luiz Eduardo Simeone e Ferando Cortez Sica *Administração de Redes Linux* .UFLA/FAEPE, 2003.

Joaquim Quinteiro Uchôa. *Segurança em Redes e Criptografia*.
UFLA/FAEPE, 2005.

K. Hamzeh; G. Pall; W. Verthein; J. Taarud; W. Little e G. Zorn. RFC-2637 – *Point-to-Point Tunneling Protocol*
http://www.ietf.org/rfc/rfc2637.txt?number=2637, acesso em janeiro de 2006.

L. Manakos; K. Lidl; J. Evarts; D. Carrel; D. Simone e R. Wheeler. RFC-2516 – *Point-to-Point over Ethernet*
http://www.ietf.org/rfc/rfc2516.txt?number=2516, acesso em janeiro de 2006.

Matthew D. Wilson. *The VPN HOWTO*. www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/**VPN**-HOWTO.pdf , acesso em abril de 2006.

MSDN. *Aumentando a segurança dos dados com o SQL Server 2005* , 2005. Disponível via www em http://www.microsoft.com/brasil/msdn/Tecnologias/arquitetura/SegurancaDadosSQLServer2005.mspx?mfr=true, acesso em março de 2006.

NAI - *Network Associates Inc. An Introduction To Criptography*. Santa Clara, NAI, 1999. Disponível via www em http://www.pgpi.org/doc/guide/6.5/en/intro/, acesso em março de 2006.

P. Mockapetris. RFC-1034 D*omain Names - Concepts and Facilities*
http://www.ietf.org/rfc/rfc1034.txt?number=1034, acesso em janeiro de 2006.

P. Mockapetris. RFC-1035 *Domain Names - Iplememtation e Specification.* http://www.ietf.org/rfc/rfc1035.txt?number=1035, acesso em janeiro de 2006.

Point-to-Point Protocol over Ethernet, 2005. Disponível via www em http://en.wikipedia.org/wiki/Pppoe, acesso em março de 2006.

R. Thayer; N. Doraswamy e R. Glenn. RFC-2411 – *IP Security* http://www.ietf.org/rfc/rfc2411.txt?number=2411, acesso em janeiro de 2006.

Rob Scrimger; Paul LaSalle; Mridula Parihar e Meeta Gupta. *TCP/IP, a Bíblia.*Campos 2002

Roderick W. Smith. *Redes Linux Avancadas*. Ciência Moderna, 2003.

Sandro Melo e Clodonil H. Trigo. *Projeto de Segurança em Software Livre*. Alta Books, 2004.

T. Dierks; C. Allen. RFC-2246 – *Transport Layer Security* http://www.ietf.org/rfc/rfc2246.txt?number=2246, acesso em janeiro de 2006.

W. Townsley; A. Valencia; A. Rubens; G. Pall; G. Zorn e B. Palter. RFC-2661 – *Layer Two Tunneling Protocol* http://www.ietf.org/rfc/rfc2661.txt?number=2661, acesso em janeiro de 2006.