UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

SAMUEL DAMIN BALDESSAR

ESTUDO DE UMA SOLUÇÃO PARA INTERLIGAÇÃO DE REDES USANDO SSL

CRICIÚMA, JULHO DE 2009

SAMUEL DAMIN BALDESSAR

ESTUDO DE UMA SOLUÇÃO PARA A INTERLIGAÇÃO DE REDES USANDO SSL

Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

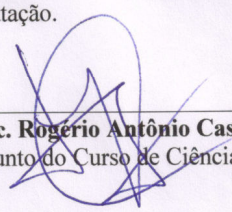
Orientador:Prof. M.Sc. Paulo João Martins

CRICIÚMA, JULHO DE 2009

SAMUEL DAMIN BALDESSAR

Estudo de uma solução para interligação de redes usando SSL

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

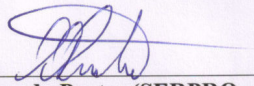


Prof. MSc. Rogério Antônio Casagrande
Coordenador Adjunto do Curso de Ciência da Computação

Banca Examinadora:

Prof. MSc. Paulo João Martins (UNESC)
Orientador

Prof. Esp. André Faria Ruaro (TI CriciumaTrans)



Prof. MSc. Ricardo Portes (SERPRO – Porto Alegre)

Dedico esse trabalho a todos que,

de forma direta ou indireta , profissional ou pessoal,

colaboraram no seu desenvolvimento.

AGRADECIMENTOS

Aos meus pais Ivo e Iracema, pela educação que me deram, sempre buscando mostrar o caminho da honestidade, respeito e trabalho, apoiando e incentivando nas horas difíceis;

a minha noiva Taise e minha irmã Suzana pela ajuda no profissional e pessoal e por compreenderem as vezes que estive ausente física ou mentalmente;

aos professores do curso, principalmente meu orientador Paulo João Martins,
que colaborou desenvolvimento deste trabalho;

e principalmente a Deus e a Nossa Senhora de Caravaggio por ter-me dado a oportunidade de viver e estar concluindo mais essa etapa.

Muito obrigado a todos!

RESUMO

Com o crescimento da demanda por serviços de interligação e acesso remoto a redes corporativas, buscou-se alternativas ao uso de *links* dedicados que tem um custo relativamente alto. O alcance global e alta disponibilidade de acesso, motivam o desenvolvimento de mecanismos para interligação de redes privadas que usam a Internet como meio de transmissão. O objetivo do presente trabalho, estudar o conceito de *Virtual Private Network* (VPN) nos aspectos de segurança, funcionalidade, custos e benefícios. Para que isso fosse possível primeiramente fez-se estudo das tecnologias de rede e de VPN disponíveis. A partir deste estudo, escolheu-se a solução OpenVPN, que tem como base o protocolo TLS/SSL. O principal motivo dessa escolha foi buscar uma alternativa às soluções de VPN baseadas no protocolo IPSec, tecnologia que domina mercado e com maior literatura disponível. Ao final, será apresentada uma proposta de implementação de uma VPN utilizando o OpenVPN, e será feita a apresentação das dificuldades encontradas e dos resultados obtidos.

Palavras-chave: Redes de Computadores, Segurança da Informação, Rede Privada Virtual, OpenVPN.

ABSTRACT

With the growth of the demand for interconnection services and remote access to corporate networks, it has been searched alternatives to the use of dedicated links, which have a relatively high cost. The global range and high availability of access, motivate the development of mechanisms for the interconnection of private networks that use the Internet as a way of transmission. The aim, of the present work, was to study the concept of Virtual Private Network (VPN) regarding safety, functionality, costs and benefits. To make this possible, firstly a study on the available network technologies and VPN was done. From this study, the solution chosen was the OpenVPN, which has as basis, the TLS / SSL protocol. The main reason for this choice, was to seek an alternative to VPN solutions based on IPSec Protocol, technology which dominates the market and with more literature available. At the end, it will be presented a proposal for the implementation of a VPN using OpenVPN, and it will be shown the difficulties encountered and results achieved.

Keywords: Computer Networking, Information Security, Virtual Private Network, OpenVPN.

LISTA DE FIGURAS

Figura 1. Rede Geograficamente Distribuída.....20

Figura 2. Criptografia Simétrica.....28

Figura 3. Posicionamento da SSL na pilha de protocolos habitual..... 30

Figura 4. Acesso Remoto via Internet.....50

Figura 5. Conexão de *LANs*.....51

Figura 6. Conexão de *Hosts* em uma mesma Intranet(sem intermediação).....52

Figura 7. Conexão de *Hosts* em uma mesma Intranet(com intermediação).....53

Figura 8. Diagrama do Experimento.....60

Figura 9. Seleção de componentes do OpenVPN.....63

Figura 10. Interface TAP-Win32 Adapter V9…….……64

Figura 11. Posicionamento do Firewal.....73

Figura 12. Status do OpenVPN no Windows……..…...76

Figura 13. Acesso a compartilhamento via VPN.....77

LISTA DE TABELAS

Tabela 1. Comparação entre protocolos de *tunneling*.....46

Tabela 2 . Conteúdo de uma chave estática Tabela 67

Tabela 3 . Arquivo servidor.conf.....68

Tabela 4 . Arquivo cliente.conf.....70

Tabela 5 . Arquivo cliente.conf usando no-ip.....72

Tabela 6 . Arquivo servidorwindows.conf para clientes windows.....74

Tabela 7 . Arquivo cliente.ovpn..... 75

Tabela 8 . Arquivos de configuração com chave estática otimizados.....80

Tabela 9 . Parâmetro alterados no arquivo openssl.cnf.....81

Tabela 10 . Criação de arquivos e diretório.....82

Tabela 11 . Arquivo servidor.conf modo TLS.....85

Tabela 12 . Arquivo cliente.conf modo TLS.....87

Tabela 13 . Teste de desempenho.....89

Tabela 14 . Custos envolvidos.....91

LISTA DE SIGLAS

ADSL	*Asymmetric Digital Subscriber Line*
AH	*Autentication Header*
ATM	*Asynchronous Transfer Mode*
CPU	*Central Processing Unit*
DES	*Data Encryption Standard*
DHCP	*Dynamic Host Configuration Protocol*
DLL	*Dynamic-link library*
DNS	*Domain Name System*
DoS	*Denial of Service*
ESP	*Encapsulation Security Payload*
GRE	*Generic Routing Encapsulation*
HTTPS	*HyperText Transfer Protocol Secure*
IETF	*Internet Engineering Task Force*
IKE	*Internet Key Exchange*
IPSec	*IP Protocol Security*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
IPX	*Internetwork Packet Exchange*
ISDN	*Integrated Services Digital Network*
ISP	*Internet Service Provider*
LAN	*Local Area Networking*
LZO	*Lempel-Ziv-Oberhumer* (data compression algorithm)

L2F	*Layer 2 Forwarding*
L2TP	*Layer 2 Tunneling Protocol*
MPLS	*MultiProtocol Label Switching*
NAS	*Network-attached storage*
NAT	*Network Address Translation*
NetBEUI	*NetBIOS Extended User Interface*
OSI	*Open Systems Interconnection*
PKI	*Publik Key Infraestruture*
PPP	*Point-to-Point Protocol*
PPTP	*Point-to-Point Tunneling Protocol*
RAS	*Remote Access Servicesk*
RFC	*Request for Comments*
SA	*Security Association*
SSL	*Secure Sockets Layer*
TCP	*Transmission Control Protocol*
TAP	*Terminal Access Point Interface*
TLS	*Transport Layer Security*
TUN	*Tunneling Interface*
UDP	*User Datagram Protocol*
VPN	*Virtual Private Network*
WAN	*Wide Area Networking*

SUMÁRIO

1 INTRODUÇÃO **14**
1.1 OBJETIVO GERAL 15
1.2 OBJETIVOS ESPECÍFICOS 15
1.3 JUSTIFICATIVA 15
1.4 ESTRUTURA DO TRABALHO 16
2 REDES DE COMPUTADORES **17**
2.1 TOPOLOGIA DE REDES 18
2.2 REDES LOCAIS 19
2.3 REDES GEOGRAFICAMENTE DISTRIBUÍDAS 19
2.4 REDES SEM FIO 20
2.5 INTERNET 21
3 REDES PRIVADAS VIRTUAIS **23**
3.1 CONCEITOS BÁSICOS 24
3.1.1 Criptografia **25**
3.1.1.1 Criptografia Simétrica 27
3.1.1.2 Criptografia assimétrica 29
3.1.2 Tunelamento **29**
3.1.2.1 Tipos de Tunelamento 31
3.2 PROTOCOLOS PARA VPN 32
3.2.1 *Point-to-Point Protocol* **33**
3.2.2 *Point-to-Point Tunneling Protocol* **34**
3.2.3 *Layer 2 Forwarding* **35**
3.2.4 *Layer 2 Tunneling Protocol* **36**
3.2.5 *IP Protocol Security* **37**
3.2.5.1 *Autentication Header* 39
3.2.5.2 *Encapsulation Security Payload* 40
3.2.5.3 *Internet Key Exchange* 41
3.2.5.4 Avaliação do *IPSec* – Vantagens e Desvantagens 42
3.2.6 *Transport Layer Security / Secure Sockets Layer* **44**
3.3 COMPARAÇÃO ENTRE PROTOCOLOS DE *TUNNELING* 46
3.4 VPNS IMPLEMANTADAS PELOS ISPS 47
3.5 ALGUMAS CONSIDERAÇÕES RELEVANTES SOBRE AS VPNS 47
3.6 AMEAÇAS X VULNERABILIDADES 49
4 CENÁRIOS DE ACESSO REMOTO **50**
4.1 ACESSO REMOTO VIA INTERNET 50
4.2 CONEXÃO DE LANS 51
4.3 CONEXÃO DE *HOSTS* DE UMA MESMA INTRANET (SEM INTERMEDIAÇÃO) 51
4.4 CONEXÃO DE *HOSTS* DE UMA MESMA INTRANET (COM INTERMEDIAÇÃO) 52
4.5 CENÁRIO ESCOLHIDO 53
5 TRABALHOS CORRELATOS E SOFTWARES VPN **53**
5.1 O OPENVPN 55
5.1.2 Uso do protocolo TLS para autenticação e negociação de chaves 56
6 ESTUDO DE UMA SOLUÇÃO PARA INTERLIGAÇÃO DE REDES USANDO SSL **58**
6.1 IDENTIFICAÇÃO DOS RECURSOS NECESSÁRIOS PARA IMPLEMENTAÇÃO 59

6.1.1 Identificação da versão e distribuição Linux a ser usada 59
6.2 DIAGRAMA DO EXPERIMENTO 60
6.3 INSTALAÇÃO DO OPENVPN 61
6.3.1 Instalando o OpenVPN no Linux 61
6.3.2 Instalando o OpenVPN no Windows 63
6.3.3 Windows x Linux 65
6.4 CONFIGURANDO E EXECUTANDO O OPENVPN 65
6.4.1 Configuração do servidor por meio de chaves estáticas 66
6.4.2 Configuração do cliente por meio de chaves estáticas 70
6.4.3 Finalizando a configuração 71
6.4.3.1 Posicionamento do Firewall 73
6.4.4 Configuração por meio de chaves estáticas Windows 74
6.4.5 Parâmetros de otimização 78
6.5 CONFIGURAÇÃO POR MEIO DE CERTIFICADOS X509 80
6.5.1 Criando a Autoridade Certificadora 81
6.5.2 Gerando parâmetros Diffie-Hellman 83
6.5.3 Gerando certificados públicos e privados 84
6.5.4 Configurando o servidor no modo TLS 85
6.5.5 Configurando o cliente no modo TLS 86
6.5.6 Revogando Certificados 87
6.6 TESTE DE DESEMPENHO 88
6.7 RESULTADOS OBTIDOS 89
6.8 CUSTOS RELACIONADOS 91
CONCLUSÃO 92
REFERÊNCIAS 94
APÊNDICE A – ARQUIVO *SERVIDOR.CONF* COMENTADO 96
APÊNDICE B – INSTALAÇÃO DO UBUMTU SERVER 98

1 INTRODUÇÃO

Organizações têm a necessidade de compartilhamento de recursos de hardware e software, o que demanda o uso de redes locais. Porém, ocorre uma grande necessidade de redes locais diferentes e distantes umas das outras, se conectarem para que uma possa usufruir dos recursos da outra e vice-versa, como no caso de uma matriz e uma filial de uma empresa por exemplo.

Uma alternativa para tal conexão seria o uso de VPN (*Virtual Private Network*) que é uma rede privada de dados construída por meio da utilização de uma rede pública, como, por exemplo, a Internet. Ou seja, utiliza-se uma rede pública para conectar diferentes nós, ou redes, ao invés de *links* dedicados entre os mesmos (PETERSON; DAVIE, 2004).

Utilizar a Internet para a conexão de *hosts* de redes privadas é uma boa alternativa para a redução de custos, porém falha quanto à segurança da informação em relação à confidencialidade (TANENBAUM, 1997).

As informações que trafegam entre redes corporativas muitas vezes são confidenciais. No caso se usar uma rede pública como *backbone*, mecanismos de segurança devem ser implemantados.

Uma solução seria utilizar criptografia para o tráfego de dados por meio da rede pública, ou seja, se estes dados forem capturados durante o trajeto pela rede pública eles não poderão ser decifrados.

O estudo propõe usar uma solução VPN, que implemente mecanismos de segurança para a conexão de *hosts* de redes privadas fisicamente distantes e que tenha um custo relativamente baixo.

1.1 OBJETIVO GERAL

Implantar uma Rede Privada Virtual (VPN) para interligar redes distintas buscando redução de custos e um nível aceitável de segurança usando software livre.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

a) Compreender a tecnologia VPN;
b) Utilizar *links* ADSL com cliente DNS dinâmico, como alternativa a *links* dedicados;
c) Pesquisar e avaliar soluções de baixo custo de VPN;
d) Verificar a viabilidade da utilização de VPN usando uma solução de baixo custo;
e) Compreender os protocolos disponíveis para o desenvolvimento de VPNs ;
f) Verificar as vantagens e desvantagens no uso desse sistema.

1.3 JUSTIFICATIVA

A expansão da Internet vem ocorrendo de forma muito acelerada. A cada dia, cresce a disponibilidade do acesso de banda larga. Paralelo a isso, a necessidade de interconexão de redes privadas também aumenta. Neste contexto, conexões seguras são indispensáveis, já que as informações que trafegam na rede privada são confidenciais e precisam de um meio seguro de transmissão.

Uma solução seria a utilização de *links* dedicados, que são eficientes, porém seu custo é relativamente alto. (PETERSON; DAVIE, 2004).

O alto custo dos *links* dedicados motiva o desenvolvimento de mecanismos de acesso remoto baseados na Internet. Esse tipo de acesso remoto utiliza a tecnologia de Redes Privadas Virtuais (VPN), possibilitando que uma infra-estrutura de rede pública, como a Internet, seja utilizada como *backbone* para a comunicação entre o usuário remoto e a rede privada. (KUROSE; ROSS, 2006).

A facilidade de acesso e o alcance mundial da Internet, no entanto, possibilitam a existência de vários outros cenários de acesso remoto VPN e a tarefa de satisfazer os requisitos de segurança das várias classes de usuários de acesso remoto apresenta vários desafios.

A segurança é um fator crucial, e o conhecimento das novas ameaças juntamente com a adoção de um conjunto de mecanismos de segurança capazes de atender aos requisitos impostos nos diversos cenários de acesso remoto VPN, torna-se vital para que a solução de acesso remoto alcance seus objetivos.

Todas essas vantagens e novas possibilidades representam um forte incentivo para que organizações migrem do acesso dedicado para um modelo de acesso remoto VPN.

1.4 ESTRUTURA DO TRABALHO

A seguir é apresentada uma síntese dos capítulos constantes nesse trabalho.

O primeiro capítulo demonstra uma visão geral do presente trabalho, definindo o problema a ser resolvido e mostrando os objetivos a serem alcançados com ele.

O segundo capítulo apresenta um breve histórico sobre as redes de computadores, demonstrado algumas tecnologias e conceitos importantes.

O terceiro capítulo fala sobre as tecnologias de VPN existentes, os protocolos envolvidos e os tipos de criptografia e de tunelamento.

O quarto capítulo descreve os diversos cenários de acesso remoto em que as VPN's podem ser utilizadas.

O quinto capítulo apresenta alguns trabalhos correlatos a este, ferramentas de VPN disponíveis e o OpenVPN.

O sexto capítulo demonstra o trabalho desenvolvido desde a instalação até os resultados obtidos referente ao uso do OpenVPN para interligação de redes.

2 REDES DE COMPUTADORES

Comunicação é o ato de transmissão de informações de um lugar a outro. Comunicação sempre foi, desde o início dos tempos, uma necessidade humana. A necessidade de comunicação motivou o desenvolvimento das redes de computadores.

Com o surgimento dos computadores, logo se teve a necessidade de interligá-los de algum modo, para o compartilhamento de informações e recursos. A capacidade de compartilhar informações e recursos é o que torna a rede uma ferramenta valiosa. Antes do seu surgimento, as atividades de computação comercial eram difíceis, caras e frustrantes (TANENBAUM, 1997).

Uma rede de computadores é formada por um conjunto de módulos processadores (MPs) capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação. As informações podem ser dados, programação, mensagens instantâneas, dentre outros e os recursos podem ser impressoras, dispositivos de armazenamento, *firewal*ls e *gateways* de Internet (SPURGEON, 1997).

De acordo com Tanenbaum (1997) existem dois tipos de tecnologia de transmissão: as de redes de difusão e as redes ponto a ponto.

As redes de difusão têm apenas um canal de comunicação, compartilhado por todas as máquinas. Os pacotes enviados por uma das máquinas são recebidos por todas as

outras. Um campo de endereço dentro do pacote especifica seu destinatário. Quando recebe um pacote, a máquina o analisa. Se o pacote tiver sido endereçado à própria máquina, ela o processa, se for destinado a outra máquina, o pacote é ignorado.

As redes ponto a ponto consistem em muitas conexões entre pares de dispositivos de rede individuais. Partindo da origem, talvez um pacote desse tipo de rede tenha de visitar um ou mais dispositivos intermediários para chegar ao seu destino. Como em geral é possível ter diferentes rotas com diferentes tamanhos, encontrar boas rotas é algo importante em redes ponto a ponto e os algoritmos de roteamento desempenham um papel importante. Embora haja algumas exceções, geralmente as redes locais de menor porte tendem a usar os sistemas de difusão e as maiores, o sistema ponto a ponto.

Um trabalho de suma importância para as LANs foi criado na década 70 , no Centro de Controle de Palo Alto,CA da Xerox *Corporation*. Foram desenvolvidos um grupo de padrões e protocolos chamado *Ethernet*. A importância de tal criação foi muito grande a ponto desse padrão ser utilizado fortemente até os dias atuais (SPURGEON ,2000).

2.1 TOPOLOGIA DE REDES

Topologia refere-se ao *layout* físico e lógico da rede. Os pontos no meio físico onde são conectados computadores, impressoras, dentre outros recebem a denominação de nós. Estes nós sempre estão associados a um endereço lógico, para que possam ser reconhecidos pela rede, não importando qual dispositivo esteja conectado (KUROSE; ROSS, 2006).

A seguir, uma divisão entre topologia lógica e topologia física, sugerido por (TANENBAUM, 1997):

a) topologia lógica: descreve a maneira como a rede transmite informações de um equipamento para outro, o formato dos dados, o método de transferência, dentre outros. É a forma como os protocolos operam no meio físico. O exemplo mais conhecido e mais usado é a topologia lógica *Ethernet*.

b) topologia física: descreve como o cabeamento e componentes de hardware do meio físico serão dispostos na rede, como é feita a distribuição da mídia de conexão (cabeamento de cobre, fibra óptica, *wireless*, entre outros). São exemplos: a topologia em barramento, em estrela e em anel.

2.2 REDES LOCAIS

De uma forma geral, uma Rede Local ou *Local Area Network* (LAN) pode ser definida como qualquer rede que conecta dois ou mais dispositivos, localizados dentro de uma área geograficamente limitada. Fisicamente, quanto maior a distância de um nó da rede ao outro, maior a taxa de erros que ocorrerão devido à degradação do sinal (KUROSE; ROSS, 2006).

As LANs têm um tamanho físico restrito, o que significa que o pior tempo de transmissão é limitado e conhecido com antecedência. Conhecendo esse limite, o administrador simplifica o gerenciamento da rede e a torna mais estável (SPURGEON ,2000).

2.3 REDES GEOGRAFICAMENTE DISTRIBUÍDAS

Uma rede geograficamente distribuída, ou *Wide Area Network* (WAN), abrange uma ampla área geográfica, as vezes se estendendo por países ou continentes. Ela contém um

conjunto de máquinas (*hosts*) com as mais diversas finalidades, desde provedores de acesso, até *desktops* de usuários domésticos ou de empresas. Esses *hosts* se conectam por uma sub-rede que tem a função de transportar mensagens de um *host* para o outro (TANENBAUM, 1997).

O mesmo autor afirma que na maioria das redes geograficamente distribuídas, a sub-rede consiste em dois componentes distintos: linhas de transmissão e elementos de comutação. As linhas de transmissão transportam os *bits* entre as máquinas e os elementos de comutação são computadores, ou hardwares dedicados, usados para conectar duas ou mais linhas de transmissão, normalmente chamados de roteadores. A Figura 1 mostra o esquema de uma Rede Geograficamente Distribuída.

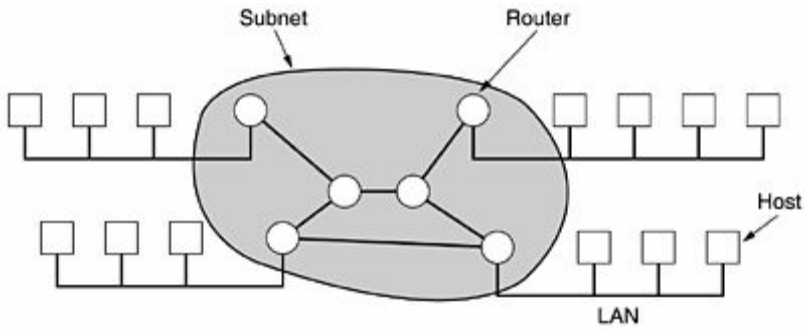


Figura 1. Rede Geograficamente Distribuída
Fonte:TANENBAUM,A (1997).

2.4 REDES SEM FIO

Redes sem fio, ou redes *wireless*, podem ser definidas como redes que utilizam o ar como meio de transmissão de dados, voz e vídeo.

A comunicação digital sem fios não é uma idéia nova. Em 1901, o físico italiano Guglielmo Marconi demonstrou como funcionava um telégrafo sem fio. Dentro de um navio, transmitiu informações para o litoral por meio de código *morse* (que pode ser considerado

binário). Os modernos sistemas digitais sem fios têm um desempenho melhor, mas a idéia básica é a mesma (TANENBAUM, 1997).

Nos últimos anos, o mundo vem se tornando cada vez mais móvel, com a telefonia celular e as redes de dados sem fio se tornando mais comuns. Em conseqüência, tipos tradicionais de redes de computadores (cabeadas) têm perdido espaço. Se usuários estão conectados a uma rede por meio de cabos físicos, seus movimentos serão muito reduzidos, o que motiva o desenvolvimento de novas tecnologias de rede sem fio, que agregam cada vez mais velocidade e qualidade de comunicação às redes (KUROSE, ROSS, 2006).

2.5 INTERNET

Internet é um sistema de comunicação que interliga todos os continentes. É formado por várias redes de computadores interligadas, ou seja, é uma rede de redes. A Internet é uma poderosa ferramenta que torna possível o acesso à informação de forma rápida e fácil, além de permitir a transferência de dados (GUIZZO, 2002).

Apesar de tanto questionamento sobre a segurança na Internet hoje em dia, ela surgiu exatamente pela busca de segurança nas informações militares americanas na década de 1960.

Após passagem pelos ambientes acadêmicos, ganhou consolidação e padronização, depois que o TCP/IP tornou-se protocolo oficial em 1º de janeiro de 1983. (TANENBAUM, 1997).

Mas o tempo passou, e grandes incidentes de segurança, juntamente com a demanda por novos recursos, tornaram evidente a necessidade de se transformar a Internet em um lugar seguro.

Com a popularização da Internet, a necessidade de manter seguros os dados transmitidos na rede se tornou evidente, já que muitos realizam movimentações financeiras on-line, transmitem dados sigilosos, dentre outras atividades. Para tal, foram criadas diversas medidas para proteger os dados de ataques ou reduzir as chances de um ataque ser bem sucedido. Uma boa maneira de proteger dados em trânsito na Internet consiste na utilização de técnicas de criptografia e tunelamento, sendo ambas disponibilizadas na tecnologia de redes privadas virtuais (RICCI, 2007).

Segurança na Internet é um tema que demanda um estudo aprofundado. Nos próximos capítulos, serão apresentados conceitos e técnicas de segurança para tráfego de dados na Internet, principalmente sobre a temática das redes privadas virtuais.

3 REDES PRIVADAS VIRTUAIS

Centenas de milhares de empresas e instituições geograficamente distribuídas ainda não possuem qualquer solução segura para interligação de suas redes ou para acesso de usuários remotos. (GUIMARÃES; LINS; OLIVEIRA, 2006).

A facilidade de acesso, o alcance mundial e o baixo custo da Internet motivam o desenvolvimento de mecanismos de segurança para tráfego de dados na rede. A idéia de utilizar uma rede pública como a Internet em vez de *links* dedicados (muito mais caros) para implementar redes corporativas é denominada *Virtual Private Network* (VPN). Uma VPN utiliza protocolos de segurança e tecnologias de criptografia, tunelamento e autenticação para garantir características de uma rede dedicada a quem utiliza uma infra-estrutura de rede não-confiável, como a Internet, para interligação de suas redes ou para acesso de usuários remotos a determinados recursos providos pela rede (SILVA, 2005).

O processo de envio de uma informação por uma VPN é o seguinte: primeiramente o cliente é autenticado pelo servidor *Gateway* VPN ao fazer o pedido para o estabelecimento de uma conexão, também podendo ser autenticado por algum outro servidor de autenticação definido na rede. Posteriormente, a informação é criptografada, possibilitando o envio de formatos de dados ilegíveis (textos cifrados) e, logo em seguida, estas informações são encapsuladas em pacotes IP. Neste momento, o *Gateway* VPN acrescenta um novo cabeçalho IP, contendo o endereço do *Gateway* VPN origem e o destino, encapsulando o pacote original. À medida que os pacotes chegam ao seu destino, vão sendo reconstituídos e decodificados para um formato legível (GUIMARÃES; LINS; OLIVEIRA, 2006).

Pessoas e empresas interessadas no uso da tecnologia VPN, devem preocupar-se com vários aspectos de segurança envolvidos na interligação de suas redes por meio de uma infra-estrutura não confiável.

A seguir serão apresentados alguns dos conceitos básicos que fundamentam as Redes Privadas Virtuais. Tais conceitos influem diretamente no nível de segurança provido pela solução de acesso remoto VPN, e por isso são necessários ao entendimento desta tecnologia.

3.1 CONCEITOS BÁSICOS

Buscando prover uma comunicação segura num meio de transmissão público, as VPNs se fundamentam em dois conceitos básicos: a criptografia e o tunelamento (RICCI, 2007).

A criptografia é utilizada para proteger a informação transmitida, dados são cifrados pelo emissor utilizando cálculos matemáticos complexos e decifrados no destino. A criptografia visa garantir a confidencialidade, a autenticidade e a integridade das conexões. Pode ser considerada a base para a segurança das soluções VPN.

Já o tunelamento é utilizado para o estabelecimento de uma comunicação virtual entre dois ou mais pontos utilizando uma infra-estrutura de rede pública como *backbone*, formando assim, um "túnel de criptografia" responsável pelo encapsulamento, transmissão e desencapsulamento dos dados entre estes pontos (SILVA, 2005).

As VPNs possuem protocolos de comunicação próprios, que atuam em conjunto com outros protocolos (TCP/IP por exemplo), fazendo com que um túnel virtual seja estabelecido e os dados trafeguem por ele criptografados na rede pública. Dentre eles, podemos destacar o *Point-to-Point Protocol* (PPP), o *Point-to-Point Tunneling Protocol* (PPTP), o *Layer Two Tunneling Protocol* (L2TP), *Secure IP* (IPSec) e o *Secure Sockets Layer / Transport Layer Security* (SSL/TLS), que serão abordados no decorrer deste capítulo.

3.1.1 Criptografia

A palavra criptografia significa “escrita secreta”, cujo nome vem do grego *Kryptos*, que significa oculto ou secreto, e *graphen*, que significa escrita. A criptografia pode ser definida como a arte ou ciência que utiliza códigos e cifras para ocultar a informação (TANENBAUM, 1997).

O processo de criptografia pode ser descrito da seguinte forma: um emissor gera uma mensagem original, e utilizando-se de uma chave e um algoritmo de criptografia, gera um texto cifrado que é transmitido para um receptor. Ao chegar para o receptor, esse texto passa pelo processo inverso, chamado de decifragem, resultando na mensagem original **(**NORTHCUTT; NOVAK; MCLACHLAN, 2001).

As técnicas de criptografia estão diretamente ligadas a privacidade no mundo virtual seguro. Tentar impedir que alguém capture um pacote que trafega em vários equipamentos como computadores, roteadores e *switches*, é extremamente difícil, senão impossível (GUIMARÃES; LINS; OLIVEIRA, 2006).

Como já foi dito anteriormente, as VPN’s permitem que uma rede pública, seja utilizada para a comunicação entre pontos privados. Apesar das vantagens oferecidas por esta tecnologia, o uso de uma rede pública para a transmissão de dados confidenciais pode trazer sérias implicações de segurança. Dessa forma, é imprescindível que a VPN seja capaz de prover um conjunto de funcionalidades que garanta requisitos como (SILVA, 2005):

a) **confidencialidade:** como a conexão VPN é feita por meio de uma infra-estrutura de rede pública, compartilha-se a conexão virtual com um grupo de pessoas que está no submundo da rede, tentando capturar dados que não lhes pertencem. Logo, mesmo dificultando, não se pode garantir 100% de privacidade, más é importante garantir que, mesmo em posse da

informação capturada, o dado não seja visível, preservando assim a confidencialidade da comunicação;

b) **integridade**: assegura que os dados não sejam alterados durante a transmissão. Além da possibilidade dos dados que trafegam pela rede pública serem interceptados, existe a possibilidade desses dados serem modificados por um interceptador. Assim, é necessário garantir de alguma forma que qualquer adulteração nesses dados seja detectada, garantindo integridade das informações;

c) **Autenticidade**: verifica se a destinatário com quem se está trocando informações sigilosas é realmente quem deveria ser, evitando uma possível falsificação de identidade por parte de um usuário mal intencionado, garantindo a autenticidade das partes envolvidas. Essas garantias podem ser obtidas por meio de assinaturas digitais[1];

d) **Não-Repúdio**: ”Impede que uma entidade (computador, pessoa , entre outros) envolvida em uma transação negue a sua participação no evento”(SILVA, 2005 , p.29). Um exemplo prático disso pode ser aplicado em uma indústria. A equipe de produção tem a garantia de que a informação recebida pela equipe de vendas não será negada no futuro, ou seja, há garantia da fonte da mensagem, de forma que, posteriormente, a equipe de vendas não poderá afirmar que não foi ela quem enviou. O não-repúdio é uma etapa posterior à autenticidade (ou um atributo opcional da autenticidade), que só pode ser obtida via criptografia assimétrica, a qual será vista no decorrer do trabalho.

[1] Assinatura digital é um código binário baseado em 2 aspectos: o documento em si e uma informação que ligue a uma certa pessoa ou conjunto de pessoas . Essa ligação é denominada autenticação. As assinaturas digitais tentam resolver o problema da autenticidade de documentos digitais, tal que o reconhecimento de firmas (assinaturas) em documentos de papel (GUIMARÃES; LINS; OLIVEIRA, 2006).

Além dos requisitos básicos apresentados, é interessante que a solução VPN ofereça mecanismos adicionais de segurança. Controlar o acesso de usuários, gerar relatórios periódicos, além de outros mecanismos, podem reforçar a segurança na comunicação (SILVA, 2005).

É Importante ressaltar que o segredo da criptografia não está no algoritmo empregado, e sim na chave de criptografia. Os melhores sistemas criptográficos são aqueles de domínio público, podendo, portanto ser extensamente analisados pelos cientistas e validados quanto a possíveis falhas ou fraquezas, sendo posteriormente revistos num processo permanente de melhoria (TANENBAUM, 1997). O mesmo autor afirma que existem dois tipos de chaves: chave simétrica ou chave secreta, e chave assimétrica ou chave pública que serão demonstrados a seguir.

3.1.1.1 Criptografia Simétrica

Chaves Simétricas são algoritmos que utilizam uma chave única compartilhada entre as pontas autorizadas para criptografar e descriptografar os dados transmitidos ou armazenados (RICCI, 2007).

Ao utilizar um algoritmo criptográfico simétrico, somente as pontas devidamente autorizadas (possuidoras da chave secreta**)** conseguirão entender o conteúdo transmitido. Caso um invasor capture a informação, o conteúdo estará protegido, pois sem a chave secreta para descriptografar os dados capturados, ele nada pode fazer (RICCI, 2007).

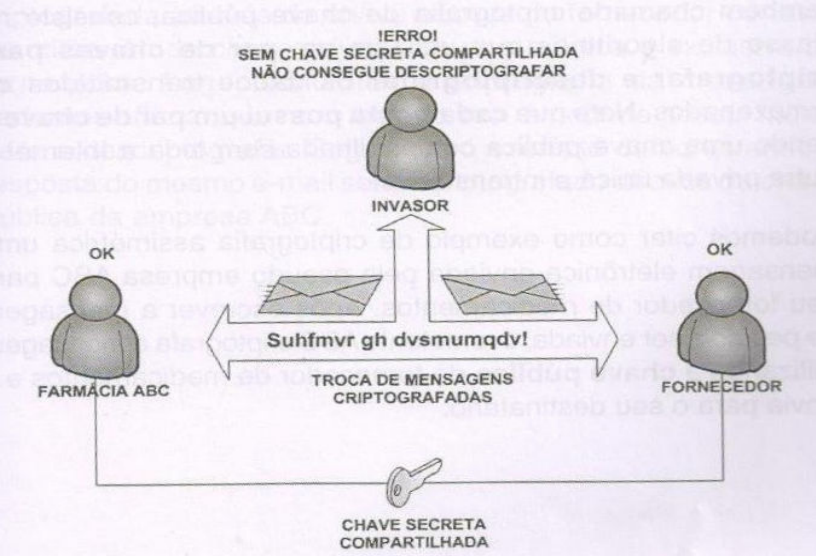


Figura 2. Criptografia Simétrica
Fonte:RICCI,B (2007).

A chave é compartilhada pelos dois pontos, ou seja, o destinatário sabe qual é a chave que utilizará para voltar a informação à sua forma original (MARCELO, 2007).

O segredo reside na chave e o problema é conseguir fazer com que somente o emissor e o destinatário de uma mensagem criptografada possam conhecer a chave secreta. Para futuras alterações nessa chave, as duas partes devem comunicar-se de um modo seguro para que nenhuma parte mal intencionada conheça a chave (MARCELO, 2007).

A grande vantagem neste tipo de chave é sua velocidade em relação à chave assimétrica, pois utiliza a mesma chave para criptografar e descriptografar os dados, consumindo menos recursos de processamento que o complexo sistema assimétrico e seu par de chaves matematicamente dependentes. (RICCI, 2007).

3.1.1.2 Criptografia assimétrica

O conceito de criptografia assimétrica é bem mais amplo que o de criptografia simétrica. Basicamente a chave é dividida em duas partes: uma parte é privada e única para o usuário; e outra parte pública que pode ser distribuída livremente com quem se deseja abrir uma comunicação com privacidade e confidencialidade (RICCI, 2007).

Normalmente, a parte privada fica armazenada usando-se um algoritmo de chave simétrica como o DES[2], porque o tamanho da chave é bem maior que o das chaves simétricas.

Quando se quer enviar uma informação criptografada para alguém, usa-se a parte pública da chave do destinatário para criptografar e enviar o dado. O destinatário utilizará a parte privada da chave para descriptografar a mensagem e retorná-la à forma original. Caso ele queira enviar algo para o emissor, irá criptografar com a chave pública e enviar pela Internet, onde irá usar a chave privada do emissor para retornar a informação à forma original (SILVA, 2005).

O RSA (*Rivest Shamir Adleman*) é o algoritmo assimétrico mais popular em uso, e possui chaves de 512, 768, 1024, 2048 bits. Este algoritmo é utilizado pela maioria das aplicações de criptografia assimétrica em uso atualmente.

3.1.2 Tunelamento

Tunelamento é o processo de encapsular um tipo de pacote dentro de outro para facilitar o transporte de uma informação dentro da rede (RICCI, 2007).

---

[2] O algoritmo de criptografia DES foi desenvolvido na década de 70 pelo *National Bureau of Standarts* com ajuda da *National Security Agency*. O propósito era criar um método padrão para proteção de dados. A IBM criou o primeiro rascunho do algoritmo, chamando-o de LUCIFER. O DES tornou-se oficialmente norma federal americana em janeiro de 1977 (TANEMBAUM, 1997).

O tunelamento resolve o problema de transferir pacotes de protocolos diferentes do TCP/IP pela Internet. Pacotes do tipo NetBEUI ou IPX por exemplo podem ser encapsulados por pacotes IP. Desta forma, o tunelamento fornece um mecanismo para que outros protocolos, além do IP, possam ser transmitidos por meio de uma VPN (GUIMARÃES; LINS; OLIVEIRA, 2006).

O tunelamento é totalmente transparentes para os *hosts, e* nenhum *software* ou configuração especial são exigidos. Os *hosts* não possuem conhecimento do fato de que os pacotes estão sendo criptografados ou que estão sendo enviados por uma rede pública. A figura 3 demonstra o estabelecimento de um túnel VPN.

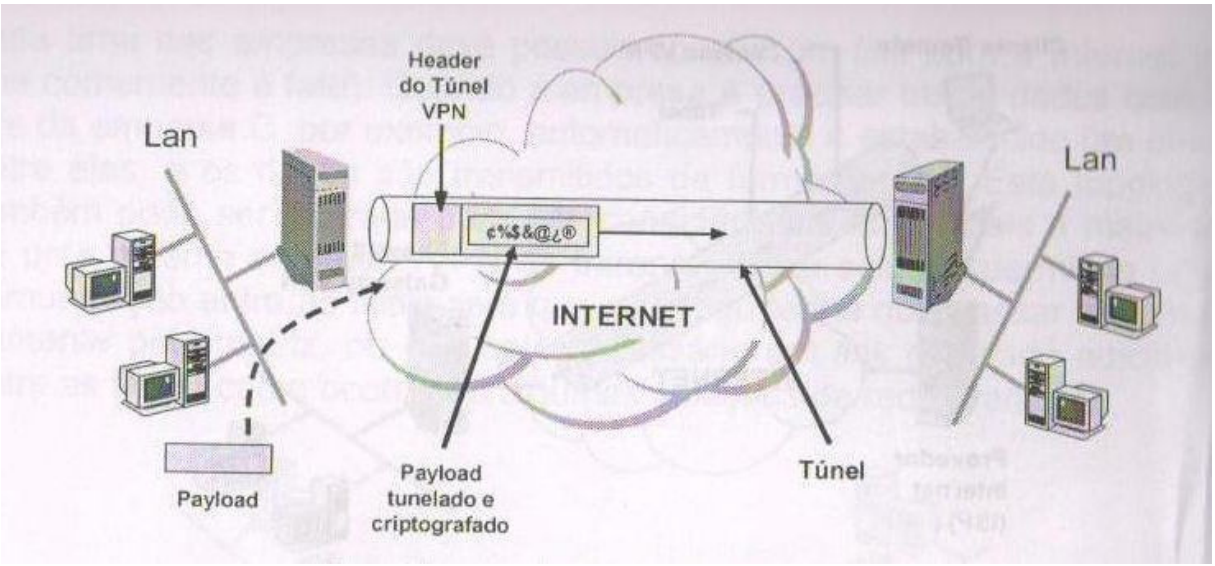


Figura 3. Estabelecimento de um túnel VPN
Fonte: GUIMARÃES; LINS; OLIVEIRA, (2006).

Embora os endereços dos *hosts* sejam mascarados para o mundo virtual, eles não possuem anonimato completo, pois os endereços dos *Gateways* estão disponíveis nos pacotes.

Caso um pacote seja capturado por um terceiro, ele ainda pode determinar quem está se comunicando com quem, embora não consiga ler o conteúdo (REZENDE, 2004).

É importante frisar que a criptografia, o encapsulamento e o tunelamento não tornam os pacotes enviados inacessíveis, eles ainda podem ser capturados e analisados. Contudo, se for utilizado um algoritmo de criptografia corretamente implementado e adequadamente forte, a informação será protegida.

3.1.2.1 Tipos de Tunelamento

Diante dos conceitos básicos apresentados, é importante ressaltar que o tunelamento pode ser considerado de dois tipos (SILVA, 2005):

a) **tunelamento voluntário –** este tipo de tunelamento ocorre quando a própria estação de trabalho ou servidor de rede utiliza algum software cliente de tunelamento para estabelecer uma conexão com o Servidor VPN. Este tipo de tunelamento é comum quando clientes remotos se conectam a Internet, para posteriormente utilizarem o software cliente. Neste modo, o túnel VPN termina no cliente;

b) **tunelamento compulsório –** este tipo de tunelamento ocorre quando existe um Servidor de Autenticação para acesso à rede. Neste caso, o estabelecimento do túnel VPN com o servidor VPN do site remoto e a configuração de autenticação são de responsabilidade do Servidor de Autenticação. Para o cliente, o tunelamento VPN é transparente, pois os clientes possuem acesso às informações das outras redes por meio do Servidor de Autenticação, não sendo necessário nenhum software cliente VPN para o estabelecimento do túnel.

3.2 PROTOCOLOS PARA VPN

Existem diversos protocolos disponíveis para a construção de redes VPN. Desta forma, no processo de implementação de uma VPN é essencial a definição da solução VPN e consequentemente o melhor protocolo para se realizar o tunelamento (SILVA, 2005).

Cada situação deve ser analisada. A aplicabilidade de cada protocolo depende dos requisitos e necessidades dos clientes, do problema que está sendo apresentado e da solução que se deseja obter (GUIMARÃES; LINS; OLIVEIRA, 2006).

Outro fator importante é o controle (quem detém e por que o controle é necessário) e de como é feita cada implementação destes protocolos (tunelamento voluntário ou compulsório). A segurança nas conexões é garantida por mecanismos de autenticação e controle de acesso usando canais criptografados (RICCI, 2007).

As VPNs possuem protocolos de comunicação próprios, que atuam em conjunto com outros protocolos (TCP/IP, por exemplo).

Os protocolos para VPN que mais se destacam são:

a) *Point-to-Point Protocol* (PPTP);

b) *Point-to-Point Tunneling Protocol* (PPTP);

c) *Layer 2 Forwarding* (L2F);

d) *Layer Two Tunneling Protocol* (L2TP);

e) *Secure IP* (IPSec);

f) *Secure Sockets Layer / Transport Layer Security* (SSL/TLS).

A seguir serão apresentadas as principais características técnicas de cada protocolo. A escolha do melhor protocolo para a VPN a ser implementada culmina num projeto mais eficiente e menos sujeito a erros.

3.2.1 *Point-to-Point Protocol*

O Protocolo Ponto-a-Ponto (PPP) é um dos protocolos de enlace de dados mais populares de se interligar *hosts*, por meio de linha serial ou discada com Provedores de acesso (ISP). O usuário remoto configura uma conexão PPP com o servidor de acesso remoto (Remote Access Server – RAS). Depois que a comunicação é estabelecida, o usuário remoto consegue enviar e receber diversos protocolos (inclusive o IP) encapsulados em frames PPP (SILVA, 2005).

Um usuário doméstico, por exemplo, pode se conectar a um Provedor de Acesso à Internet por meio de uma linha discada e se tornar temporariamente um *host* da Internet. A estação do usuário, usando uma linha discada e um modem, realiza uma conexão física (comutação de circuitos) com o modem do RAS (Servidor de Acesso Remoto) pertencente ao provedor. Após o estabelecimento desta conexão física, o computador do usuário torna-se um *host* da Internet e pode enviar e receber pacotes IP, da mesma forma que outros hosts fisicamente conectados à Internet (SILVA, 2005).

Em resumo, as principais características do protocolo PPP são (GUIMARÃES; LINS; OLIVEIRA, 2006):

a) capacidade de encapsular diversos protocolos. Os pacotes dentro do *frame* PPP não precisam ser obrigatoriamente IP. O PPP pode encapsular protocolos como o IPX e o NEtBEUI, por exemplo;
b) o PPP trata a detecção de erros;
c) permite que endereços IP sejam negociados em tempo de conexão (uso do DHCP), isto é, aceita atribuição dinâmica de endereços IP.
d) permite realizar autenticação de usuários.

Este protocolo não é objeto maior de pesquisa neste trabalho, pois é mais utilizado para conexões de Clientes Remotos com a Internet, e este trabalho procura abordar a interligação entre redes que já estão inseridas na Internet. Porém, é necessário o conhecimento básico do funcionamento deste protocolo para um melhor entendimento dos demais protocolos de tunelamento para construção de redes VPN.

3.2.2 *Point-to-Point Tunneling Protocol*

O protocolo PPTP, ou *Point-to-Point Tunneling Protocol*, é um protocolo que foi desenvolvido por um consórcio de empresas de tecnologia da Informação, incluindo a *US Robotics* (parte da 3Com), Microsoft, *Ascend Communication* (parte da *Lucent*) e *ECI Telematics*, mas, foi amplamente popularizado por meio das implementações realizadas pela *Microsoft* nos seus Sistemas *Windows* (NORTHCUTT; NOVAK; MCLACHLAN, 2001).

Este protocolo buscou atender aos interesses de fornecedores de *hardware* que participaram da sua concepção, fornecedores de Servidores de Acesso Remoto e aos interesses da *Microsoft*, fornecedora de *software*, para o desenvolvimento de soluções em conectividade por meio do uso da Internet como rede Privada Virtual e segura (RICCI, 2007).

A idéia básica do PPTP era dividir as funções do acesso remoto de tal modo que indivíduos e empresas pudessem utilizar a infra-estrutura da Internet para prover uma conectividade segura entre clientes remotos e redes privadas (REZENDE, 2004).

Em resumo, a comunicação PPTP envolve três processos (GUIMARÃES; LINS; OLIVEIRA, 2006):

a) **processo de conexão e comunicação PPP** – processo em que o cliente remoto usa PPP para se conectar ao RAS ou a um provedor de Internet,

utilizando linha telefônica ou algum serviço ISDN de comunicação. Neste caso, o PPP é utilizado para iniciar e terminar conexões físicas, autenticar usuários e criar datagramas PPP contendo pacotes criptografados;

b) **processos de conexão de controle PPTP –**processo que cria um controle de conexão desde o cliente até o servidor PPTP. Essa conexão utiliza TCP e é chamada de túnel PPTP;

c) **processo de tunelamento de dados PPTP –** processo que cria os datagramas IP contendo os pacotes PPP criptografados e os envia por meio do túnel PPTP até o servidor PPTP, que finalmente desmonta os pacotes recebidos e descriptografa os pacotes PPP para que sejam enviados à rede corporativa.

O protocolo PPTP não inclui privacidade e gerência de chave de criptografia, o que é um problema quando se fala em VPN. Conclui-se, portanto, que este mecanismo de tunelamento também não é adequado para conexões VPN entre redes corporativas, pois a maioria das aplicações da VPN PPTP é destinada aos usuários *roaming*, ou seja, aqueles que se deslocam constantemente, não se aplicando a conexões entre redes (SILVA, 2005).

3.2.3 *Layer 2 Forwarding*

Foi um dos protocolos pioneiros para VPNs. De forma semelhante ao PPTP, o L2F foi concebido para ser um protocolo de *tunneling* entre usuários remotos e redes locais. No entanto, o L2F não depende do protocolo IP, sendo, por isso, capaz de trabalhar diretamente com outros protocolos (SILVA, 2005).

O L2F foi desenvolvido na mesma época de desenvolvimento do PPTP pela Cisco, a Nortel e a Shiva *Corporation* (parte da Intel). Estas empresas estavam desenvolvendo

a proposta do L2F, que tinha como objetivo permitir que provedores de acesso ou empresas de telecomunicações, oferecessem ao mercado, um serviço de acesso remoto discado para redes privadas. Desta forma, as empresas não precisariam adquirir modens ou equipamentos de acesso remoto, podendo pagar pelo serviço (REZENDE, 2004).

3.2.4 *Layer 2 Tunneling Protocol*

O *Layer 2 Tunneling Protocol* (L2TP) , ou Protocolo de Tunelamento da Camada 2, foi proposto pela IETF (*Internet Engineering Task Force*) com o objetivo de estabelecer um padrão para o encapsulamento de *frames* PPP para a construção de redes VPN de acesso remoto (*dial-up*) (SILVA, 2005).

Os protocolos PPTP e o L2F foram propostos para padronização junto a IETF que resolveu criar um novo protocolo com as melhores características técnicas dos protocolos PPTP e do L2F (*Layer 2 Forwarding*) para se tornar padrão. Porém, os desenvolvedores do PPTP (Microsoft em especial) não aceitaram essa imposição e resolveram continuar sozinhos o desenvolvimento e aprimoramento do PPTP, enquanto os desenvolvedores do L2F decidiram parar e assumir o L2TP como padrão (SILVA, 2005).

Como conseqüência dessa convergência, o L2TP oferece as melhores funções e características destes dois protocolos, além dos benefícios adicionais como o túnel multiponto, que permite que um único cliente inicialize várias VPNs. Outra característica do L2TP é que ele suporta qualquer protocolo roteado como o IP, IPX e qualquer tecnologia e protocolo de *backbone* WAN (Frame Relay por exemplo) (RICCI, 2007).

É importante descrever as principais características do protocolo L2TP (GUIMARÃES;LINS;OLIVEIRA, 2006):

a) o L2TP foi desenvolvido para suportar os dois modos de tunelamento, voluntário e compulsório. No modo voluntário, o túnel é iniciado pelo cliente remoto. Já no modo compulsório, o túnel é automaticamente criado, exigindo que o NAS do provedor esteja pré-configurado com informações do túnel e de autenticações dos usuários;
b) o L2TP herdou os mecanismos de segurança (criptografia e autenticação) do PPP. Portanto, ele não autentica o pacote que irão sair do cliente remoto, somente autentica o usuário remoto. Ele também não provê mecanismos de gerência de chaves;
c) ao contrário do PPTP, o L2TP utiliza o protocolo UDP para fazer a manutenção de túnel VPN;
d) geralmente é utilizado em conjunto com o IPSec com o intuito principal de oferecer autenticação de pacotes e suporte a NAT .

O protocolo L2TP apresenta alguns problemas de segurança, que inviabilizam seu uso em cenários onde existe uma rede não confiável, como a Internet, entre os extremos de um túnel VPN. Seu uso deve sempre ser combinado com outros protocolos capazes de suprir a sua ausência de serviços de segurança (REZENDE, 2004).

3.2.5 *IP Protocol Security*

O IPSec foi proposto pela *Internet Engineering Task Force* (IETF) em 1998 em meio a intermináveis discussões sobre qual seria a melhor camada para se inserir a criptografia da Internet – em uma camada *fim-a-fim* ou em uma camada de rede. A camada escolhida foi a camada 3 do modelo OSI (camada de rede), correspondente a camada IP na arquitetura TCP/IP (GUIMARÃES;LINS;OLIVEIRA, 2006).

IPSec é um conjunto de protocolos que define a arquitetura e as especificações para prover serviços de segurança dentro do protocolo IP. O IPSec foi padronizado implementando mecanismo de criptografia para o IPv4 e IPv6. Também define um conjunto de serviços de segurança, incluído integridade dos dados, autenticação, confidencialidade (criptografia) e limite de fluxo de tráfego, oferecendo proteção à camada de rede e às camadas superiores (TANEMBAUM,1997).

Em resumo, pacotes IP privados, são tratados com funções de segurança de dados como criptografia, autenticação e integridade, e então são encapsulados. Após estarem protegidos em outros pacotes IP, os pacotes podem ser transmitidos. Funções de gerenciamento de chaves também fazem parte das funções do IPSec (SILVA, 2005).

O IPSec possui uma arquitetura aberta no sentido de possibilitar a inclusão de outros algoritmos de autenticação, dando a possibilidade de escolha do nível de segurança desejado aos projetistas da VPN (REZENDE, 2004).

Os requisitos de segurança podem ser divididos em três grupos (GUIMARÃES;LINS;OLIVEIRA, 2006):

a) negociação do nível de segurança;

b) autenticação e Integridade;

c) confidencialidade.

Os dois últimos são independentes entre si, podendo ser utilizados de forma conjunta ou isoladamente, de acordo com as necessidades colocadas pelo projeto.

Para implementar estas características, o IPSec é composto de 3 mecanismos adicionais (REZENDE, 2004):

a) *Autentication Header* (AH);

b) *Encapsulation Security Payload* (ESP);

c) *Internet Key Exchange* (IKE).

Esses mecanismos serão mostrados a seguir, e serão relacionados com os requisitos de segurança citados acima.

3.2.5.1 *Autentication Header*

O *Autentication Header* (AH) consiste no padrão criado pela IETF responsável por fornecer um mecanismo de integridade e autenticação dos pacotes IP. A segurança é garantida por meio da inclusão de informação para autenticação no pacote a qual é obtida por meio de um algoritmo aplicado sobre o conteúdo dos campos do pacote IP, excluindo-se aqueles que sofrem mudanças durante o transporte. Provê proteção contra ataques do tipo *replays*[3], e torna-se responsável por garantir que a origem, o destino e os dados não foram alterados durante seu tráfego na Internet (RICCI, 2007).

Porém, vale destacar que:

> Embora a autenticação aconteça no pacote IP, nem todos os campos podem ser autenticados, porque alguns campos do cabeçalho serão alterados no decorrer da transmissão. Esses campos são considerados mutantes, ou variáveis, sendo eles: Tipo do Serviço, Offset, Flags,Tempo de vida do pacote e Checksum.(SILVA, 2005 , p. 79).

O componente *authentication header* garante a integridade dos pacotes, porém não garante a confidencialidade dos dados transmitidos, ou seja, apesar de validar a origem, o destino e os dados transmitidos, não garante que caso o pacote seja capturado ao longo de sua transmissão, seu conteúdo possa ser extraído e a informação capturada. Isso acontece porque o AH não possui o recurso de criptografia. A confidencialidade é tratada pelo protocolo ESP que será visto a seguir (RICCI, 2007).

---

[3] Ataque que ocorre quando uma pessoa mal intencionada captura pacotes válidos e autenticados pertencentes a uma conexão , replica-os e os reenvia como se fosse a entidade que iniciou a conexão (RICCI,2007).

3.2.5.2 *Encapsulation Security Payload*

O ESP consiste no conjunto de serviços de segurança responsável por fornecer integridade, autenticação e confidencialidade aos dados transmitidos (GUIMARÃES; LINS; OLIVEIRA, 2006).

O ESP ao ser implementado torna-se responsável por garantir a integridade dos dados trafegados acrescida a confidencialidade, ou seja, este torna-se responsável por garantir que os dados não foram alterados durante seu tráfego na Internet além de os tornar ilegíveis por meio da utilização de criptografia. O uso de criptografia impede que pacotes capturados ao longo de sua transmissão tenham seu conteúdo secreto extraído (RICCI, 2007).

Similar ao *authentication header* (AH), o cabeçalho ESP é inserido imediatamente após o cabeçalho IP e imediatamente antes dos protocolos de camadas mais altas (RICCI, 2007).

O ESP também provê a autenticação da origem dos dados, integridade da conexão e serviço *anti-reply*. A confidencialidade independe dos demais serviços e pode ser implementada de 2 modos - transporte e túnel. No primeiro modo, o pacote da camada de transporte é encapsulado dentro do ESP, e, no túnel, o datagrama IP é encapsulado inteiro dentro do cabeçalho ESP (REZENDE, 2004).

Cabe observar que, como o pacote IP é um datagrama e portanto não possui garantia de entrega, cada pacote deve conter informações necessárias para estabelecer o sincronismo da criptografia, permitindo que a descriptografia ocorra no destino sem problemas. Porém, se nenhum algoritmo de criptografia for utilizado, o ESP poderá oferecer somente autenticação (SILVA, 2005).

3.2.5.3 *Internet Key Exchange*

Consiste no padrão responsável por especificar uma metodologia segura de troca de chaves entre duas pontas visando fazer com que essas se autentiquem e entrem em acordo quanto ao meio utilizado para assegurar os dados transmitidos, ou seja, este protocolo é utilizado junto a duas pontas IPSec para que essas estabeleçam uma relação de confiança entre si antes de transmitirem dados confidenciais (RICCI, 2007).

Os protocolos de AH e ESP especificam sob quais serviços de segurança em cada pacote IP os dados serão manipulados, de acordo com a *Security Association* (SA)[4] negociada entre as entidades participantes (GUIMARÃES;LINS;OLIVEIRA, 2006).

Uma SA pode ser configurada manualmente por um administrador de segurança em cada gateway, ou mais interessante, pode ser negociada dinamicamente por meio de um protocolo de gerência de chave como IKE (REZENDE, 2004).

Essa negociação dinâmica é necessária, porque não se sabe quando será preciso negociar uma SA para estabelecer um túnel VPN e, porque uma SA não deve ter um tempo de vida infinito, ou seja, é recomendável que se troque a SA de tempos em tempos e, consequentemente, as chaves de criptografia. Quanto mais tempo se utilizar a mesma chave de criptografia, maiores serão as chances de algum invasor descobri-la (SILVA, 2005).

O IKE é baseado no protocolo *Internet Security Association and Key Managent Protocol* (ISAKMP) que define como duas entidades instituirão um canal de comunicação seguro entre elas, trocando informações de chaves e negociando serviços de segurança.

---

[4] O conceito de Security Association (Associação de segurança - SA) é um dos conceitos fundamentais do IPSec. Uma SA define tipos de medidas de segurança que devem ser aplicadas aos pacotes baseados e quem está enviando os pacotes, para onde estão indo e que tipo de dados estão conduzindo. O conjunto de serviços de segurança oferecidos pela SA depende do protocolo de segurança, de suas opções escolhidas e do modo na qual a SA irá trabalhar (SILVA, 2005).

3.2.5.4 Avaliação do *IPSec* – Vantagens e Desvantagens

O protocolo IPSec pode proteger qualquer protocolo que rode sobre o IP e qualquer meio físico onde o IP possa rodar, podendo proteger uma grande variedade de aplicações e protocolos rodando sobre uma infra-estrutura física complexa, sendo o aproveitamento da infra-estrutura IP já existente a sua grande vantagem. (GUIMARÃES; LINS; OLIVEIRA, 2006).

O IPSec possui vantagens sobre outros protocolos que também implementam VPN em nível de enlace (nível 2) PPTP, L2TP e o L2F. Em resumo, as maiores vantagens são (GUIMARÃES; LINS; OLIVEIRA, 2006):

a) **transparente a sub-rede:** não é necessário que a Sub-rede implemente IPSec, sendo necessário apenas a implantação nas extremidades na VPN;
b) **simplicidade:** com o processo de criptografia em nível de aplicação, cada programa pode possuir sua própria implementação de segurança;
c) **flexibilidade:** nem todo trafego precisa estar sobre a ação dos serviços de segurança implementados pelo IPSec ;
d) **fácil Implementação:** pode ser implementado nos roteadores e *gateways* não alterando clientes e outros servidores da rede;
e) **gerenciamento manual e automático de chaves:** Além do gerenciamento manual de chaves, a IETF definiu como norma o protocolo IKE para o gerenciamento automático de chaves .

Também é bom ressaltar que o IPSec possui algumas limitações, que são (GUIMARÃES; LINS; OLIVEIRA, 2006):

a) não pode ser seguro se o sistema não for – garantir a segurança das máquinas que implementam o IPSec é fundamental;

b) não é fim-a-fim – as informações no próprio *host* ou em algum *host* interno podem ser violadas se o invasor estiver na rede local;

c) o IPSec autentica máquinas, más não autentica usuários – o IPSec pode garantir que a comunicação entre as máquinas transcorra de forma segura e pode saber quais máquinas se conectaram ao servidor, porem se quiser ter o controle por usuário será necessário usar uma aplicação complementar ;

d) não previne ataques DoS (Denial of Service) – esses ataques podem causar o travamento de um sistema importante, por isso é necessária a utilização de um *firewal*;

e) não evita a análise do tráfego de rede – alguns campos não criptografados dos *Headers* podem ser monitorados, como, por exemplo, os endereços de destino e origem e o tamanho do pacote. Tais campos não representam a informação em si, mas podem ajudar numa possível tentativa de roubo de informação.

O IPSec aliado aos recursos de protocolos como o AH e o ESP, e ao uso do protocolo IKE para compartilhar seguramente chaves secretas demonstra-se como uma excelente solução de segurança, quando usando corretamente, as chances de um invasor descobrir algum ponto falho na conexão são mínimas (SILVA, 2005).

3.2.6 *Transport Layer Security / Secure Sockets Layer*

Com o surgimento da *Web* para o público, empresas e bancos enxergaram um novo mercado, amplo e de alcance global. Porém, no começo, eram apenas páginas estáticas, sem preocupação com a privacidade, a integridade e a autenticidade dos dados (TANENBAUM, 1997).

Em 1995, a *Netscape Communications Corp*, introduziu um pacote de segurança chamado *Secure Sockets Layer* (SSL) para atender a demanda por conexões seguras usadas para os mais devidos fins (comércio eletrônico por exemplo).Em 1996, a *Netscape* submeteu a SSL à IETF para padronização. O resultado foi a *Transport Layer Security* (TLS), descrita na RFC 2246. As mudanças feitas na SSL foram relativamente pequenas (SILVA, 2005).

O protocolo TLS é uma versão atualizada do SSL versão 3, sendo que seu nome é constantemente associado ao SSL devido as muitas semelhanças.

O Protocolo SSL fornece confidencialidade e integridade na comunicação entre um cliente e um servidor, garantindo a integridade dos dados pelo uso de assinaturas digitais e privacidade por meio do uso de criptografia (FANELI; MARCHEZINI, 2007).

Depois que a conexão segura é estabelecida, a principal tarefa da SSL é manipular a criptografia e a compactação. Quando o HTTP é usado sobre a SSL, denomina-se *Secure HTTP* (HTTPS). O protocolo SSL não se limita ao uso apenas com navegadores da Web, mas essa é sua aplicação mais comum (MARCELO, 2007).

O TLS/SSL, trata-se de uma nova camada colocada entre a camada de aplicação e a camada de transporte. O posicionamento da SSL na pilha de protocolos habitual é ilustrado na Figura 3.

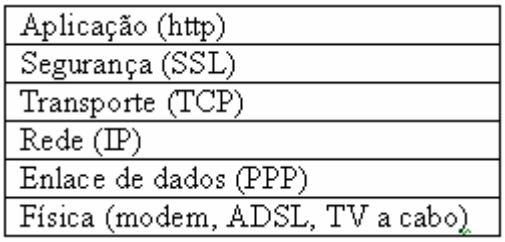


Figura 3. Posicionamento da SSL na pilha de protocolos habitual
Fonte:TANENBAUM,A (1997).

Um dos pontos fortes do TLS/SSL é que ele atua no topo dos *sockets* TCP/IP, o que torna muito mais fácil a construção de aplicações de rede utilizando o TLS/SSL do que programando diretamente sobre *sockets* (GUIMARÃES; LINS; OLIVEIRA, 2006).

O protocolo TLS/SSL foi projetado para dar suporte a diversos algoritmos de criptografia e assinatura digital existentes no mercado, além de dar suporte a projetos futuros (FANELI; MARCHEZINI, 2007).

Diferentemente do IPSec, que fornece segurança a cada datagrama IP envolvido na comunicação, o TLS/SSL é implementado na aplicação de rede (ex.: O *browser Internet Explorer*), sendo portanto parte do seu código. Isto possibilita à aplicação determinar quais dados deverão ser protegidos, o que representa uma grande vantagem, pois elimina o *overhead* resultante da proteção de certos dados que não precisam ser protegidos e consequentemente aumenta o desempenho da aplicação (FANELI; MARCHEZINI,2007).

As VPNs baseadas no protocolo *Security Sockets Layer* (SSL) vem se tornando uma tendência, e ao mesmo tempo, gerando discussões sobre a sua segurança e robustez. Muitos continuam defendendo o IPSEC como forma segura de criar VPNs, mas outros apontam que sua complexidade de implementação e configuração o tornam suscetível a erros que podem comprometer o maior benefício de uma VPN que é a confidencialidade dos dados (MARCELO, 2007).

O TLS/SSL suporta também o restabelecimento de conexão. Caso uma sessão de um cliente for interrompida por qualquer motivo, existe a possibilidade de se dar

continuidade a sessão anterior fornecendo o identificador de sessão. Isto é muito útil em aplicações cujas sessões precisam ser reiniciadas constantemente. Um exemplo prático disso é o uso de *links* ADSL para implementar VPNs.

3.3 COMPARAÇÃO ENTRE PROTOCOLOS DE *TUNNELING*

A tabela abaixo apresenta um comparativo entre as potencialidades dos principais protocolos para VPN.

Tabela 1-Comparação entre protocolos de tunneling

Propriedades	Descrição	PPTP	L2F	L2TP/IPSec	IPSec	TLS/SSL
Autenticação de Usuário	Consegue autenticar os usuários que queiram estabelecer uma conexão.	SIM	SIM	SIM	Implementação em andamento.	SIM
Autenticação de Computadores	Autentica computadores envolvidos na conexão.	SIM	SIM	SIM	SIM	SIM
Suporte a NAT	Passa por meio de um NAT para esconder os pontos finais da conexão.	SIM	SIM	NÃO	NÃO	SIM
Suporte a Multi-Protocolo	Define um método padrão para o tráfego IP e não IP	SIM	SIM	SIM	Implementação em andamento.	SIM
Atribuição Dinâmica de Endereço IP	Define uma negociação de endereçamento IP entre o servidor VPN e seus clientes. Isso elimina configurações manuais do protocolo IP.	SIM	SIM	SIM	Implementação em andamento.	SIM
Encriptação	Pode criptografar o trá-fego corrente.	SIM	SIM	SIM	SIM	SIM
Uso de PKI	Usa infra-estrutura de chave pública para imple mentar a criptografia e a autenticação.	SIM	SIM	SIM	SIM	SIM
Autenticação de Pacotes	Provê um método de autenticação que garante que os pacotes não foram alterados durante a transmissão.	NÃO	NÃO	SIM	SIM	SIM

Fonte: FANELI, A; MARCHEZINI, V (2007).

3.4 VPNS IMPLEMANTADAS PELOS ISPS

Nos últimos anos, com o surgimento de tecnologias que implementam VPNs baseadas no protocolo IP, surgiu interesse cada vez maior das empresas, principalmente as concessionárias de telecomunicações, em oferecer serviços de VPN para um grande número de clientes sobre os mesmos *backbones*, de maneira escalável e gerenciável. Infelizmente no Brasil não existem empresas que efetivamente possuem o serviço. A Embratel, por exemplo, possui um serviço de abrangência nacional chamado **IP VPN**, que é implementado em MPLS[5], e não possui mecanismo de conexão com a Internet, funcionando somente em circuito dedicado (GUIMARÃES; LINS; OLIVEIRA, 2006).

3.5 ALGUMAS CONSIDERAÇÕES RELEVANTES SOBRE AS VPNS

Algumas considerações referentes a custo e desempenho se fazem necessárias para o sucesso na implantação de um projeto VPN.

A capacidade de processamento dos dispositivos que compõe uma VPN é um fator importante a ser considerado, pois a criptografia exige grande capacidade de processamento, em ambas as pontas da VPN que precisam criptografar e descriptografar os dados transmitidos. As soluções para o problema de necessidade de processamento são a utilização de criptografia por hardware dedicado, que tem um custo relativamente alto, e as soluções por software, que demandam maior poder de processamento (GUIMARÃES; LINS; OLIVEIRA, 2006).

---

5 MPLS, ou MultiProtocol Label Switching, é uma tecnologia de encaminhamento de pacotes baseada em rótulos (labels), que funciona basicamente com a adição de um rótulo nos pacotes IP na entrada do *backbone* e, a partir daí, todo o encaminhamento pelo *backbone* passa a ser feito com base neste rótulo e não mais no endereço IP, simplificando o processo de roteamento e permitindo a criação de VPNs por meio da criação de tabelas de *labels* exclusivas de cada VPN

Se existirem limitações críticas de tempo para a transmissão das informações, o uso de VPNs por meio da Internet pode ser inadequado, pois podem ocorrer problemas de desempenho e atrasos na transmissão sobre os quais a empresa não terá controle (SILVA, 2005).

Apesar da redução de custos (que pode não ocorrer), é preciso muita atenção com a segurança quando se constrói uma WAN utilizando a Internet como meio de transporte. Usar criptografia nas VPNs é fundamental, já que os dados farão a maior parte do trajeto por meio de vários roteadores e *hosts* desconhecidos, em território pouco familiar e eventualmente inseguro (RICCI, 2007).

Quando a informação é cifrada no lado do emissor, uma chave é necessária para decifrá-la no lado do receptor. Os dispositivos que implementam a VPN em cada lado da conexão devem gerenciar esta troca de chaves de forma automática e transparente (SILVA, 2005).

O uso de VPNs para acesso remoto tem crescido consideravelmente. Desenvolvido como uma alternativa aos acessos remotos tradicionais por usuário e senha, o acesso remoto VPN tem grande aplicabilidade em um ambiente corporativo. O acesso de usuários remotos à rede por meio de VPNs é um pouco mais complexo de se desenvolver, pois neste caso é necessário algum mecanismo para autenticar o usuário e uma forma qualquer de negociar a troca de chaves. Algoritmos de chaves públicas e assinaturas digitais são utilizados para estas finalidades (GUIMARÃES;LINS;OLIVEIRA, 2006).

Ressalta-se também que a implementação de uma VPN pode consumir bastante tempo e tornar-se desvantajosa economicamente. Se não houver um planejamento adequado, preocupando-se com a gerência das chaves e a resolução de problemas encontrados, o projeto pode tornar-se inviável em relação a custo, a tempo e a eficácia da solução implantada. É importante que se tenha conhecimento das redes que se pretende interligar, assim como as

suas configurações, pois qualquer imperfeição pode resultar em mais tempo gasto para corrigi-la (MARCELO, 2007).

3.6 AMEAÇAS X VULNERABILIDADES

Apesar de a VPN ser um grande aliado das empresas, de administradores de rede e de segurança, ela por si só não resolve todos os problemas de segurança de uma corporação, nem é a solução para todos os problemas. A empresa que não se preocupa com segurança ou que não está focada no momento, corre o risco de levar prejuízos financeiros pelo tempo que ficar sem determinado serviço, pelo vazamento de informações confidenciais e pela perda de rendimento de funcionários que acessam sites indevidamente.

As ameaças que podem ser citadas são (SILVA, 2005):

a) *hackers;*
b) antigos funcionários ou funcionários insatisfeitos;
c) parceiros *extranet*;
d) usuários que querem ter posse da informação para uso pessoal ou beneficio próprio.

A vulnerabilidade que a empresa fica exposta se dá pela falta de uma política de segurança, sistemas desatualizados, gestão inadequada dos softwares e dispositivos existentes praticados por pessoas sem o conhecimento necessário para tal. Comprovadamente, a maioria dos ataques, ou captura de informações confidenciais, acontecem dentro da empresa ou através de engenharia social.

4 CENÁRIOS DE ACESSO REMOTO

Acesso remoto consiste basicamente em oferecer a computadores remotos, devidamente autenticados, o mesmo nível de acesso obtidos por computadores fisicamente localizados em uma rede privada local.

Existem vários cenários possíveis de acesso remoto, que apesar de possuírem requisitos específicos, na maioria dos casos, possuem muitos pontos em comum. Entender esses requisitos se faz necessário para que se possa avaliar efetivamente qual a melhor opção na implementação de um projeto (REZENDE, 2004).

4.1 ACESSO REMOTO VIA INTERNET

Sua definição técnica consiste na categoria de VPN que permite a um computador localizado em uma rede pública qualquer conectar-se ao perímetro de uma rede privada por meio da autenticação em um servidor de acesso, o qual, por sua vez, concede ao cliente acesso aos recursos da rede privada local (RICCHI, 2007).

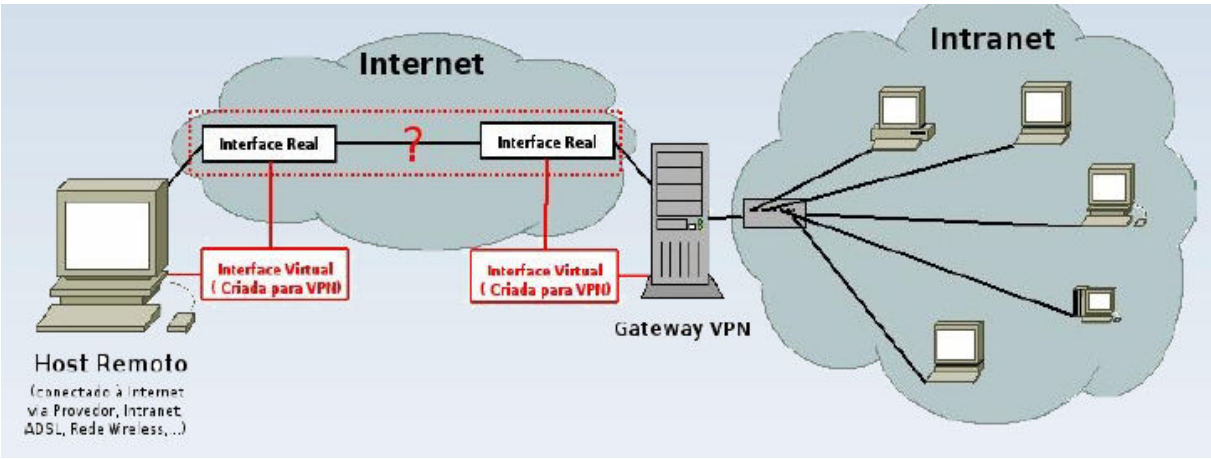


Figura 4. Acesso Remoto via Internet
Fonte: GALVÃO (2007).

4.2 CONEXÃO DE LANS

Também conhecida como VPN *site-to-site, lan-to-lan*, ou *router-to-router*. Permite conectar redes privadas distintas de maneira segura utilizando como meio uma rede pública qualquer. Geralmente é utilizada para substituir um circuito dedicado.

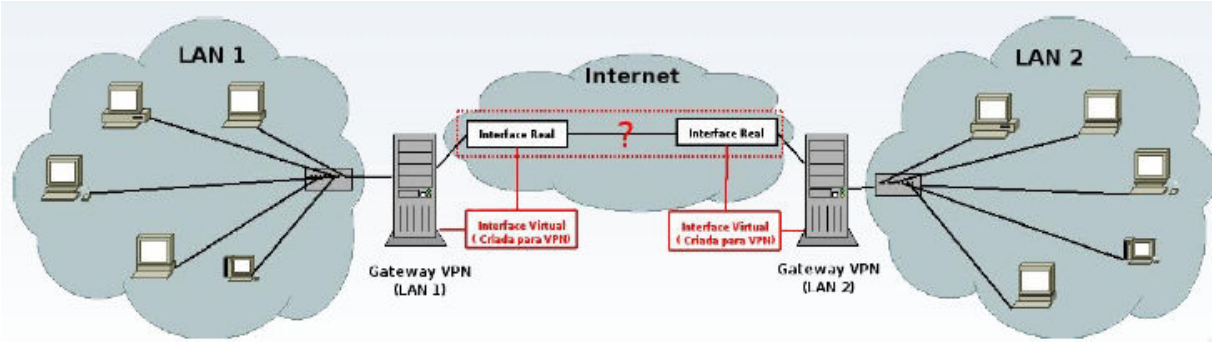


Figura 5. Conexão de LANs
Fonte: GALVÃO (2007)

4.3 CONEXÃO DE *HOSTS* DE UMA MESMA INTRANET (SEM INTERMEDIAÇÃO)

Sua definição técnica consiste na categoria de VPN que permite dois ou mais computadores localizados em uma mesma rede privada possam se comunicar de forma segura, sendo que os demais computadores da Intranet não conseguem visualizar a informação transmitida, e para tal comunicação não se faz necessário o uso de um *gateway* VPN.

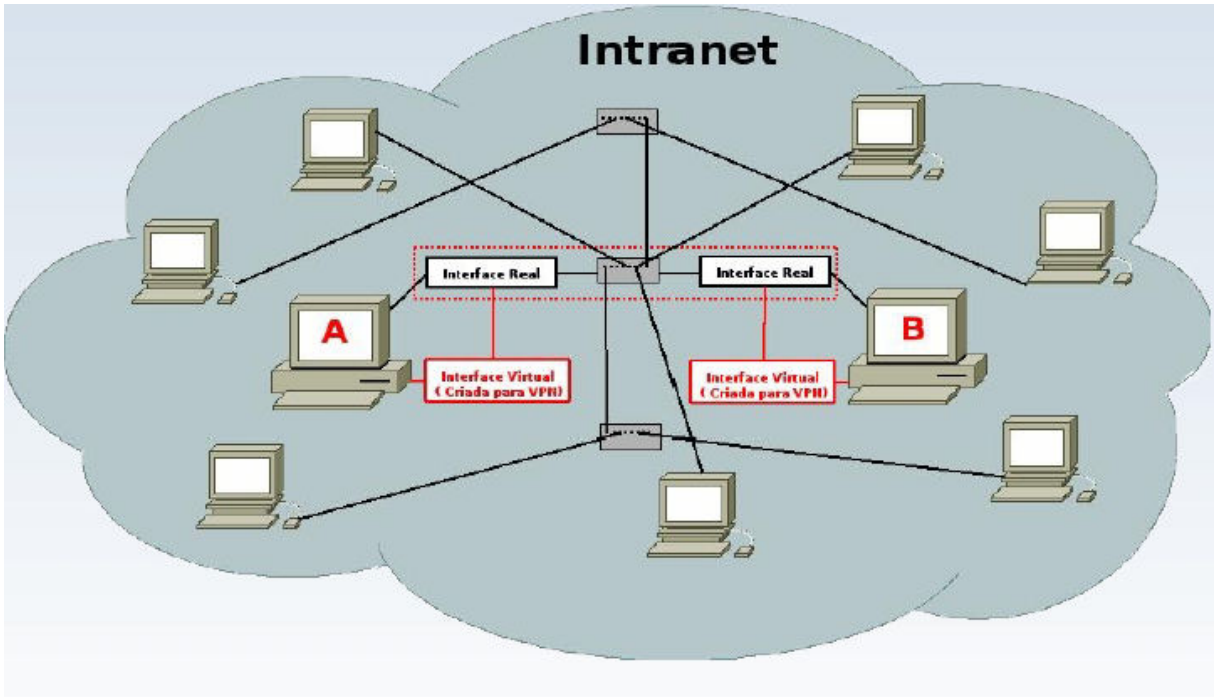


Figura 6. Conexão de *Hosts* em uma mesma Intranet (sem intermediação)
Fonte: GALVÃO (2007).

4.4 CONEXÃO DE *HOSTS* DE UMA MESMA INTRANET (COM INTERMEDIAÇÃO)

A única diferença do modelo apresentado anteriormente, é que nesse modelo, se faz necessário o uso de um *gateway* VPN que gerencia todo o processo de comunicação. Permite dois ou mais computadores localizados em uma mesma rede privada possam se comunicar de forma segura, sendo que os demais computadores da Intranet não conseguem visualizar a informação transmitida.

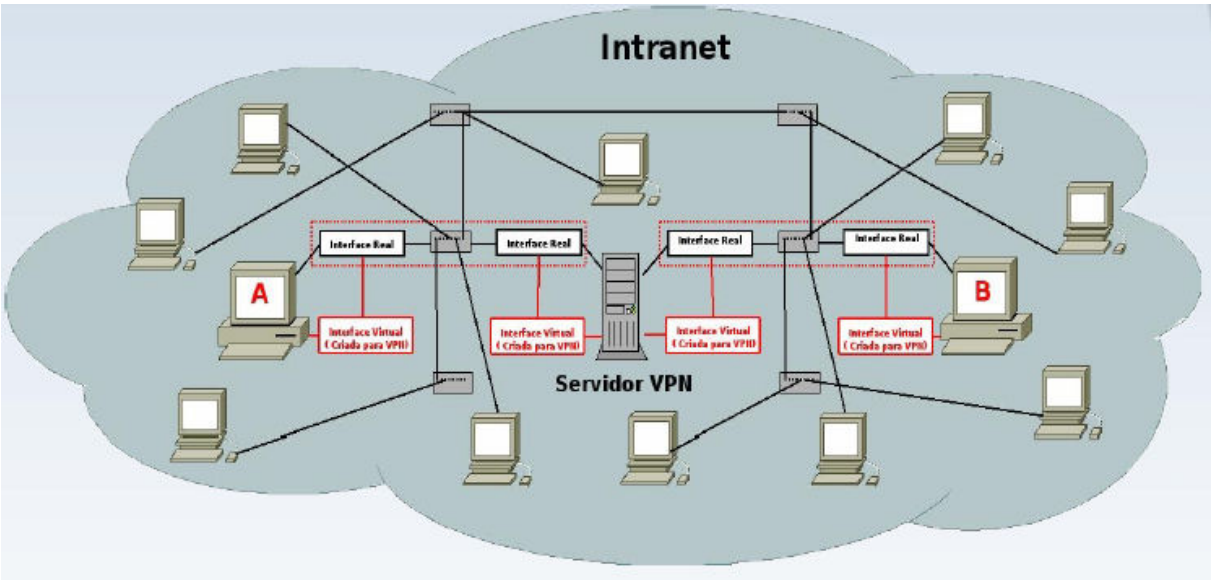


Figura 7. Conexão de *Hosts* em uma mesma Intranet (com intermediação)
Fonte: GALVÃO (2007).

4.5 CENÁRIO ESCOLHIDO

O cenário escolhido para a implementação do trabalho usa a topologia *lan-to-lan*. Dois computadores, em redes distintas, farão o papel de *gateway* VPN, criptografando, encapsulando e transmitindo pacotes. Esse processo também pode ser feito por hardware dedicado, porém uma solução de hardware eficiente tem um custo alto e pode ficar obsoleta com o passar dos anos, o que motiva o uso de soluções de software (SILVA, 2005).

5 TRABALHOS CORRELATOS E SOFTWARES VPN

Por trazer grandes benefícios e novas possibilidades, que representam um forte incentivo para que organizações e usuários migrem para um modelo de acesso remoto VPN, existem diversos estudos em diferentes linhas relacionadas com as tecnologias existentes.

A exemplo de outros sistemas que são mantidos com o código fonte aberto (como por exemplo o “Linux”) uma comunidade virtual de *software* livre ajuda constantemente a

melhorar o OpenVPN. Códigos fonte e maiores informações podem ser obtidos em http://openvpn.net.

Existem outros sistemas de código fonte aberto disponíveis dentre eles pode-se destacar (GALVÃO,2007):

a) FreeSwan – disponível em www.freeswan.org;
b) OpenSwan – disponível em www.openswan.org ;
c) Vtun – *Virtual Tunnel* disponivel em http://vtun.sourceforge.net;
d) Cipe – *Crypto IP Encapsulation* disponível em http://sourceforge.net/projects/cipelinux;
e) Vpnd – *Virtual Private Network Daemon* disponível em http://vpnd.dotsrc.org;
f) Tinc – disponível em http://www.tincvpn.org;
g) Secvpn – *Secure Virtual Private Network* http://alioth.debian.org/projects/secvpn;
h) Yavipin – disponível em http://yavipin.sourceforge.net.

Na Universidade Estadual de Campinas, Edmar Roberto Santana de Rezende, autor do trabalho "Segurança no Acesso Remoto VPN", concluído em Fevereiro de 2004, realizou um amplo estudo dos diversos aspectos envolvidos na elaboração de uma solução segura e viável de acesso remoto VPN. Por meio desta análise, foi possível identificar os principais requisitos e avaliar algumas soluções existentes que compõe esse complexo cenário.

Também na Universidade Estadual de Campinas, um trabalho intitulado "Uma Análise de Soluções VPN em Redes Corporativas de Alta Capilaridade" foi desenvolvido por

Robledo de Andrade e Castro, tendo como principal objetivo analisar as principais abordagens e protocolos disponíveis, visando classificar de maneira clara os propósitos e limitações de cada abordagem, e focando na utilização do *IP Security* (IPSec) para prover uma VPN de baixo custo e principalmente segura.

Na Universidade do Extremo Sul Catarinense o Bacharel Lucas Urgioni Niehues, autor do trabalho "Ameaças Digitais: Um Estudo dos Riscos Envolvidos no Uso da Internet, seus Impactos e Formas de Proteção" apresentou as principais ameaças digitais existentes, seus sintomas e procedimentos de defesa. O trabalho não tem o foco específico em VPN, porém procura apresentar mecanismos de segurança da informação na Internet, que são buscados na implementação de VPNs.

5.1 O OPENVPN

Desenvolvido por James Yonan e publicado como software livre, permite que as pontas da VPN se autentiquem criando um túnel de criptografia. Baseia-se no protocolo SSL/ TLS e está disponível para Solaris, Linux, OpenBSD, FreeBSD, NetBSD, Mac Os X e Windows 2000/XP (não funciona no Windows 98).

O OpenVPN Não é compatível com o IPSec e opera no conceito cliente/servidor. Utiliza a biblioteca de funções OpenSSL para a geração de criptografia. Preferencialmente, o OpenVPN trabalha sobre o protocolo UDP (TCP também a torna operacional). Utiliza duas interfaces para a conexão, a TUN e a TAP. A TUN simula a camada 3 ou de rede e a TAP simula uma interface *Ethernet* na camada 2 ou de enlace (MARCELO, 2007).

Suas principais características são (GALVAO, 2007):

a) utiliza os protocolos SSL/TLS;

b) flexibilidade (uso de TCP ou UDP) ;

c) implementa todos os cenários apresentados;

d) clientes também para Windows ;

e) uso de chaves ao invés de usuário/senha;

f) plataformas: Linux, Windows (a partir do 2000), OpenBSD,FreeBSD, NetBSD, MacOS X e SunOS/Solaris.

O OpenVPN pode operar de três formas com relação ao nível de utilização de criptografia (MARCELO, 2007):

a) nenhuma criptografia – é apenas criado o túnel no protocolo escolhido (TCP ou UDP) sem o uso de criptografia;

b) criptografia de chaves estáticas – onde cada uma das pontas precisa ter uma cópia da chave privada e essa chave não muda;

c) no modo TLS – o *Transport Layer Security*, ou simplesmente TLS, é um protocolo criptográfico onde as chaves são trocadas periodicamente por meio de certificados digitais criados por uma autoridade certificadora.

Existem outras ferramentas de VPN SSL disponíveis, algumas livres, outras pagas, porém o OpenVPN é a ferramenta que possui o maior número de funcionalidades , além de ser *freeware* e multiplataforma.

No decorrer do trabalho serão apresentadas todas as etapas para o desenvolvimento de uma VPN utilizando o OpenVPN, além de bibliotecas e arquivos de configuração necessários.

5.1.2 Uso do protocolo TLS para autenticação e negociação de chaves

O TLS é a última evolução da família de protocolos SSL e foi desenvolvida originalmente pela Netscape para o seu primeiro web *browser* seguro. O TLS e seus predecessores SSL têm se espalhado pela web ao longo dos anos e têm sido extensivamente analisados à procura de falhas. Isso tem permitido sua evolução e fortalecimento, de forma

que nos dias de hoje TLS/SSL é considerado um dos mais robustos e maduros protocolos disponíveis (FANELI; MARCHEZINI, 2007).

5.1.3 Diferenciais que motivam a escolha do OpenVPN

Os principais pontos fortes do OpenVPN incluem portabilidade para diversas plataformas conhecidas do universo computacional (PC, Mac, Sun, etc), grande estabilidade, escalabilidade para centenas ou milhares de clientes, fácil instalação e suporte para IP dinâmico e NAT (FANELI; MARCHEZINI, 2007).

6 ESTUDO DE UMA SOLUÇÃO PARA INTERLIGAÇÃO DE REDES USANDO SSL

Nos últimos anos têm ocorrido grandes mudanças na área de tecnologia da informação. Novos recursos de *hardware* e *software* foram criados ou melhorados. A infra-estrutura de comunicações também evolui muito, permitindo o acesso a Internet de alta velocidade a um custo acessível.

Inicialmente, as redes locais eram utilizadas apenas para compartilhamento de recursos, sem muita preocupação com segurança.

Com a crescente necessidade de comunicação entre redes locais, tornou-se necessário o desenvolvimento de métodos para interligação dessas redes de forma segura e confiável. Um desses métodos baseia-se no conceito de *Virtual Private Network* (VPN), onde mecanismos segurança baseados no uso de criptografia são adicionados para que se possa usar a Internet como meio de comunicação para interligação de redes ou para acesso de usuários remotos.

Antes do surgimento do conceito de VPN, a única forma de interligar redes era por meio de *links* dedicados. Os links dedicados possuem um alto custo e caso se deseje interligar mais uma rede, um novo link deve ser contratado.

Para implementar uma VPN de forma eficiente, vários fatores devem ser considerados. A VPN deve ter um nível aceitável de segurança e ser estável.

O objetivo deste trabalho é analisar os aspectos de segurança, funcionalidade, custos e benefícios envolvidos na implementação de uma VPN. Serão utilizadas ferramentas de *software* livre em ambiente Linux e Windows. Para o desenvolvimento, escolheu-se a solução OpenVPN, que tem como base o protocolo TLS/SSL. O principal motivo dessa escolha foi buscar uma alternativa às soluções de VPN baseadas no protocolo IPSec,

tecnologia que domina mercado e com maior literatura disponível. Ao final, serão apresentadas as dificuldades encontradas e os resultados obtidos.

6.1 IDENTIFICAÇÃO DOS RECURSOS NECESSÁRIOS PARA IMPLEMENTAÇÃO

A configuração de uma VPN requer a escolha da ferramenta que será usada para implementar a VPN com seus respectivos protocolos e também a escolha do sistema operacional dos *gateways* . Existem várias ferramentas disponíveis no mercado, algumas proprietárias e outras *open source*.

Este trabalho focou-se em três fatores básicos: custos de implementação, segurança e usabilidade .

Por isso procurou-se encontrar uma ferramenta *open source*, que fosse multi-plataforma, que implementasse mecanismos de segurança eficientes e que fosse estável.

Dentre as softwares VPN disponíveis no mercado, o OpenVPN, baseado no protocolo TLS/SSL foi o escolhido. O que motivou essa escolha, foi sua facilidade de configuração, disponibilidade como *freeware,* além de compatibilidade com vários sistemas operacionais.

O OpenVPN também funciona bem em redes baseadas em NAT e IP dinâmico, que estão presentes na maioria das redes com conexões ADSL (MARCELO, 2007).

6.1.1 Identificação da versão e distribuição Linux a ser usada

O OpenVPN é desenvolvido para plataforma Linux, portanto qualquer distribuição pode ser usada. Apesar disso, Marcelo(2007) sugere o uso da distribuição Fedora. Mesmo assim, optou-se pela distribuição Linux Ubuntu server versão 8.4, devido a sua

simplicidade, estabilidade e familiaridade devido ao grande tempo de uso durante a graduação.

6.2 DIAGRAMA DO EXPERIMENTO

O cenário da VPN a ser implementada , consiste na topologia *lan-to-lan,* onde duas redes locais serão interligadas usando a Internet com backbone. Esse número de redes pode aumentar muito, devido a escalabilidade do OpenVPN. Um *firewall* filtrará os pacotes de entrada e saída. A interface de conexão com a Internet usa *modens* ADSL com NAT e IP dinâmico. O *gateway* VPN rodará o *software* OpenVPN, responsável pela Criptografia e tunelamento dos pacotes transmitidos pela Internet.

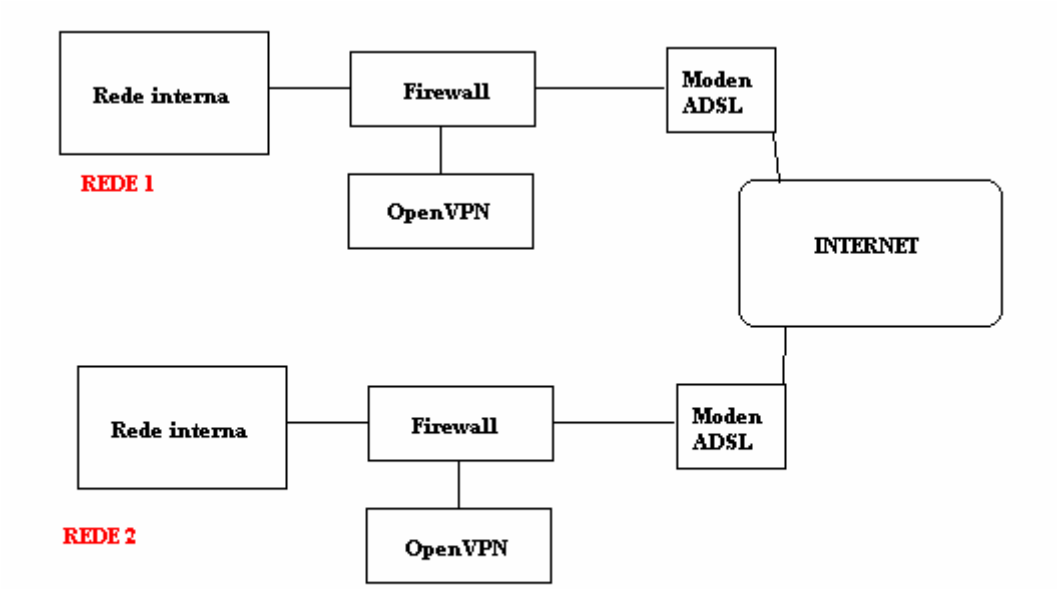


Figura 8. Diagrama do Experimento

6.3 INSTALAÇÃO DO OPENVPN

Os arquivos de instalação do OpenVPN estão disponíveis na internet em http://www.openvpn.net/index.php/downloads.html em versões Windows e Linux. A versão escolhida foi a 2.1 por ter mais recursos e estar disponível para instalação automática no Ubuntu server.

Apesar de ainda não estar finalizada, a versão 2.1 está em constante atualização e apresenta vários recursos extras que motivam sua escolha:

a) funciona em sistemas 64bits, e no Windows vista;
b) apresenta recursos extras, como suporte automático a *smartcards,* que guardam certificados digitais para conexões VPN usando TLS*;*
c) suporta múltiplas conexões;
d) compartilha a porta 443 , permitindo que o servidor OpenVPN utilize a mesma porta do servidor HTTPS.

O OpenVPN implementa extensões de segurança de rede nas camadas OSI 2 ou 3 usando o protocolo padrão da indústria TLS/SSL, que apresenta flexibilidade nos métodos de autenticação de clientes (suporta certificados digitais, *smart cards*, etc) e permite políticas de controle de acesso por usuários ou grupos usando regras de *firewall* aplicadas à interface virtual VPN (TUN/TAP).

6.3.1 Instalando o OpenVPN no Linux

Se computador onde o OpenVPN vai ser instalado estiver conectado a Internet, a instalaçãoa instalação pode ser feita por linha de comando: no Ubuntu server 8.04 , basta entrar com o usuário *root* e digitar o comando:

```
apt-get install openvpn.
```

Além do pacote openvpn, a biblioteca de compactação LZO II e outros 3 pacotes são baixados e instalados automaticamente: openssl ,openssl-blacklist e openvpn-blacklist.

Caso o computador não possua acesso a internet ou a versão do linux utilizada for outra, basta conseguir o arquivo **openvpn-2.1_rc16.tar.gz** e seguir os seguintes passos:

a) descompactar o arquivo em um diretório qualquer;

root@server:/tmp # tar –xzvf openvpn-2.1_rc16.tar.gz

b) em seguida basta digitar;

root@server:/tmp/openvpn/ # ./configure

root@server:/tmp/openvpn/ # make

root@server:/tmp/openvpn/ # ./make install

c) para verificar se o OpenVPN está instalado corretamente;

openvpn –version

Lembrando que, por estar em constante atualização, novas versões do OpenVPN 2.1 podem estar disponíveis para *download* no site, porém todas tornam a VPN operacional , sendo que algumas pequenas modificações são feitas em cada liberação. Foram executados testes com duas versões diferentes do OpenVPN 2.1 e a VPN funcionou pefeitamente.

6.3.2 Instalando o OpenVPN no Windows

A instalação em ambiente Windows requer alguns cuidados especiais. O inicio da instalação é como qualquer instalação Windows, basta ir pressionando *next* e a aceitar o termo de uso da licença GNU-GPLdo OpenVPN clicando em *I agree* . A próxima tela, é a de instalação das DLLs e bibliotecas necessárias para o perfeito funcionamento do OpenVPN, bem como suporte à interface TAP, necessária no ambiente Windows. Apesar de nem todos os itens serem obrigatórios, é interessante deixar todos marcados.

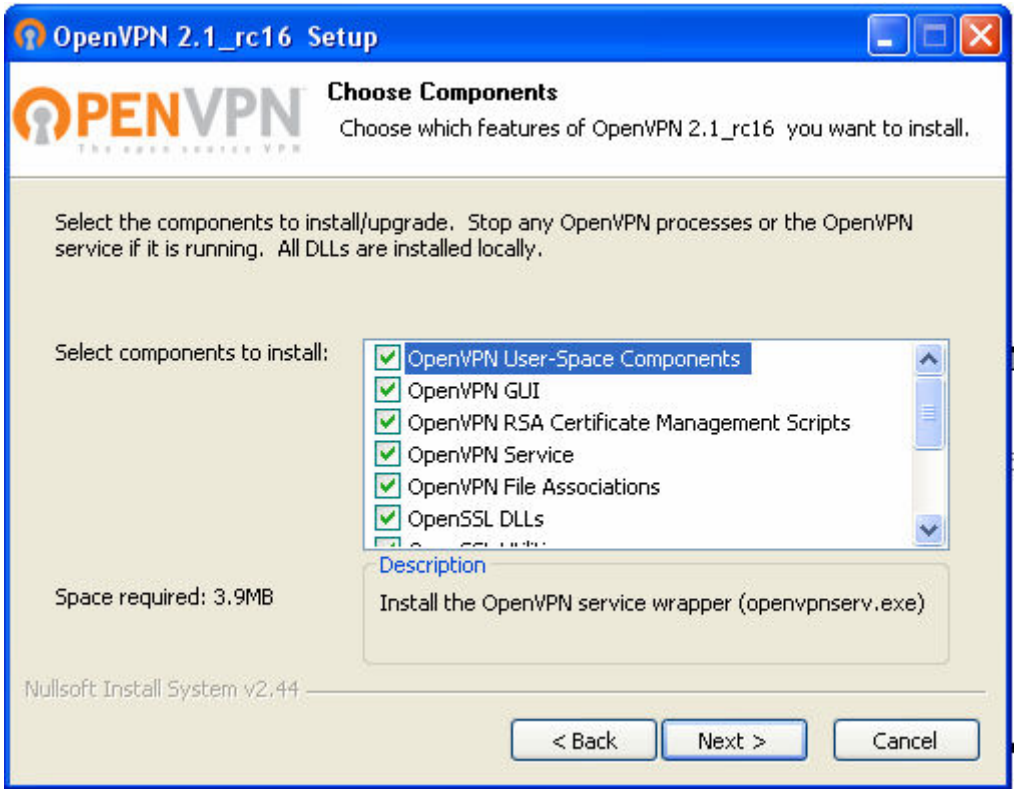


Figura 9. Seleção de componentes do OpenVPN
Fonte: http://www.openvpn.net/index.php/downloads.html

Em determinado ponto da instalação, será exibida uma mensagem avisando que o driver TAP/Win32 não passou pelo processo de certificação de compatibilidade com o

Windows. Deve-se então, escolher a opção "Continuar assim mesmo" para concluir a instalação, caso contrário o OpenVPN não ficará operacional.

Com a instalação finalizada, é importante abrir os dispositivos de rede do Windows e verificar se a nova interface rede TAP foi instalada. Esta conexão TAP-Win32 Adapter V9 é a interface virtual utilizada pelo OpenVPN para a conexão com *hosts* remo tos.

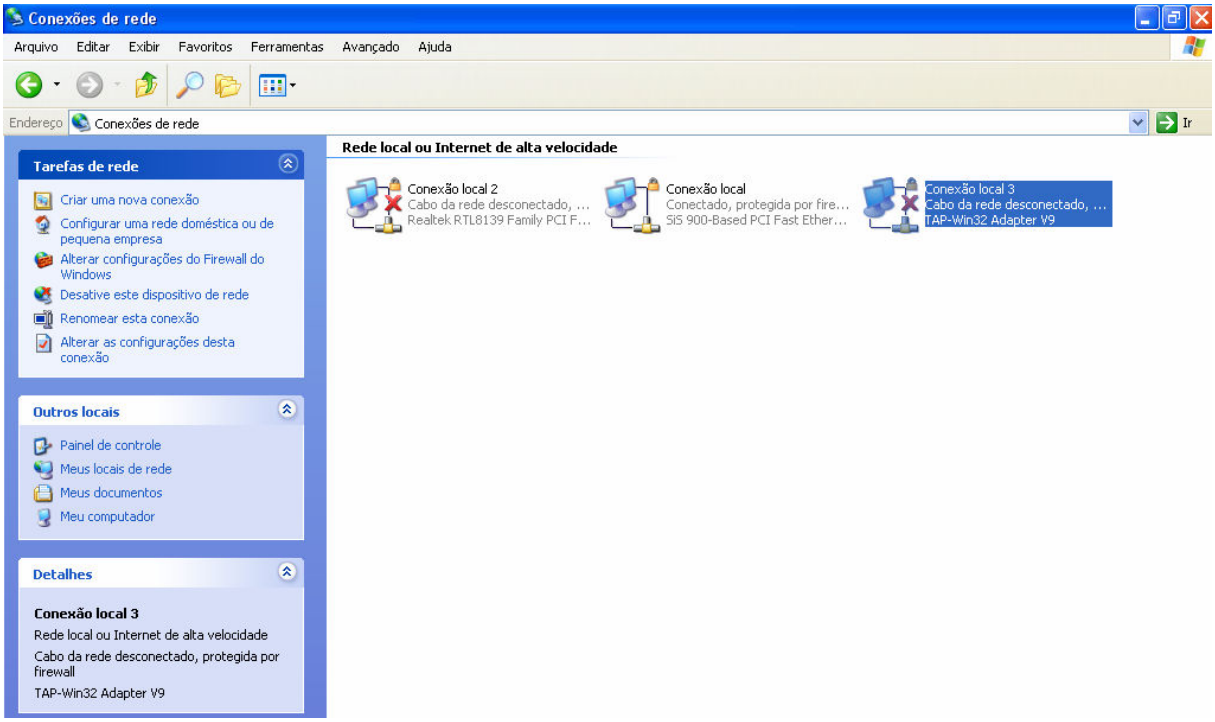


Figura 10. Interface TAP-Win32 Adapter V9
Fonte: http://www.openvpn.net/index.php/downloads.html

Analisando a figura 9, percebe-se que uma nova interface de rede foi criada, essa interface é virtual e é por meio dela que o acesso aos recursos da rede configurada na outra ponta da VPN torna-se possível. Essa interface recebe o endereço de IP local definido no arquivo de configuração do OpenVPN, e o Windows a interpreta como sendo uma interface física, tornando possível imprimir numa impressora da outra rede por exemplo, desde que tudo esteja configurado corretamente.

6.3.3 Windows x Linux

Ao instalar o OpenVPN nos dois sistemas operacionais, nota-se que a forma de interface na rede VPN é diferente. O linux tem suporte em nível de kernel a interfaces TUN e TAP, e o Windows tem suporte somente a TAP. Segundo alguns projetos já realizados, contata-se o seguinte (MARCELO, 2007):

a) ligação Linux – Linux – interface utilizada TUN;
b) ligação Linux – Windows – interface utilizada TAP;
c) ligação Windows – Windows – interface utilizada TAP.

A definição de qual interface será escolhida, bem como outras configurações serão definidas na próxima etapa, onde os arquivos de configuração devem ser otimizados para que a VPN fique estável e relativamente segura. Um parâmetro errado e o desempenho e confiabilidade podem ser prejudicados.

6.4 CONFIGURANDO E EXECUTANDO O OPENVPN

Como já foi visto anteriormente, o OpenVPN pode operar de três formas com relação ao nível de utilização de criptografia(MARCELO, 2007):

a) nenhuma criptografia – é apenas criado o túnel no protocolo escolhido (TCP ou UDP) sem o uso de criptografia;
b) criptografia de chaves estáticas – onde cada uma das pontas precisa ter uma cópia da chave privada e essa chave não muda;

c) no modo TLS – o *Transport Layer Security*, ou simplesmente TLS, é um protocolo criptográfico onde as chaves são trocadas periodicamente por meio de certificados digitais criados por uma autoridade certificadora.

A forma mais simples de configurar o OpenVPN, obtendo um nível aceitável de segurança, é utilizando chaves estáticas. Nesse modo, é gerado um arquivo no servidor contendo a chave de criptografia, que tornará os dados ilegíveis para que não possuir a chave criptográfica e por ventura tentar capturá-los. O arquivo precisa estar no servidor e ser copiado para a máquina cliente para que a VPN funcione.

Para VPNs onde a segurança é de extrema prioridade, é recomendável utilizar o modo TLS, por meio de uma estrutura baseada em certificados X509, onde as chaves são trocadas periodicamente.

Para iniciar o projeto, será configurada uma VPN básica, com chave estática e com um nível aceitável de segurança. Não será demonstrada a configuração sem criptografia porque não existe segurança nesse modo. Caso se queira usar uma VPN sem criptografia, basta retirar o parâmetro *secret* do arquivo de configuração.

O primeiro exemplo, será entre duas maquinas rodando Linux, com chaves estáticas e com parâmetros básicos, posteriormente será usado um servidor Linux com clientes Windows. Ao final , o ambiente será melhorado adicionando-se alguns parâmetros para estabilizar a VPN.

6.4.1 Configuração do servidor por meio de chaves estáticas

Primeiramente deve-se definir qual máquina será o servidor. A implementação será num servidor Linux. Feito isso, é preciso gerar a chave de criptografia no servidor. O

próprio software permite a geração da chave. No *prompt* do Shell basta digitar o seguinte comando:

openvpn --genkey --secret chave.key

O parâmetro -- *genkey* gera a chave de criptografia, necessária para a configuração da VPN.

O parâmetro -- *secret* escreve o resultado num arquivo chamado *chave.key*, que poderia ter qualquer nome.

Chave.key terá em seu conteúdo uma chave com 2048 bits descrita abaixo:

Tabela 2 – Conteúdo de uma chave estática

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
d72d2dba45278df38e389a28d9157e29
5db494116932ff1a0934136ed452b356
b8d8856a281a0824d2ef5d3ded49839b
2baaa84ebffb4c33467508b1b5a321dd
356f6d273fc391573c8e9e0655760e6d
435a9eb52fe92f14780a887eab1b8fae
c5e783216efc16c178a33fdf94d036af
9b58f34d14abaeea1e23960cb7930333
1f948ad67cdc192c6e51e7ca17aa600d
729d9b07393281a3b8275d05a19233dd
3e046072656566d6c4e628695b5f70e5
470f8aa843304513caedb3ce0c9f7469
e6f87dbacd229e0360b97833f8e768bb
b7e0a1f981a3aebcd08aa92d4245f616
85cd5d5bbdc8267c103cd8f0eb993186
aef9865139efeba75b64a9981910ae2b
-----END OpenVPN Static key V1-----
```

Todas as pontas da VPN devem possuir uma cópia desse arquivo. Para padronizar todos os arquivos necessários serão copiados na pasta /etc/openvpn. Deve-se tomar cuidado para que o conteúdo da chave estática não caia em mão erradas, por isso é importante limitar o acesso físico ao computador em que ela esta gravada e também tomar cuidado no modo em que ela será copiada para a outra ponta, se possível transferir por sftp, evitando email ou outras formas de envio inseguras.

Após gerar o arquivo chave.key, deve-se copiá-lo para a pasta /etc/openvpn.

No diretório /etc/openvpn, também fica o arquivo que inicia a VPN, esse arquivo possui a extensão *.conf* e no servidor será chamado de *servidor.conf*. Nele serão definidos uma série de parâmetros para utilização de criptografia e conexão entre os *hosts*.

Tabela 3 – Arquivo servidor.conf

`proto udp`
`dev tun`
`ifconfig 192.168.0.1 192.168.0.2`
`secret /etc/openvpn/chave.key`
`port 5900`

O parâmetro *proto* define qual protocolo será utilizado. Por questões de desempenho o protocolo escolhido foi UDP, porém o uso do TCP também é possível.

O grande problema que inviabiliza o uso do TCP é a redução de desempenho. Ao utilizar uma porta TCP, o OpenVPN tem que fazer a checagem e retransmissão dos pacotes perdidos, gerando overhead, e diminuído o desempenho .

O protocolo UDP deve ser utilizado sempre que possível, pois os pacotes são transmitidos diretamente, aumentado o desempenho. Normalmente, vem a dúvida se o uso do UDP pode gerar inconsistência de dados. A resposta é não, porque o OpenVPN é responsável apenas pela criação do link de dados, e sobre ele tem o protocolo TCP/IP, além das diferentes camadas do sistema operacional que se encarregam da correção de erros e retransmissão de pacotes perdidos.

O parâmetro *dev* identifica o tipo da interface virtual para a conexão da VPN. Se um dos *gateways* VPN estiver rodando Windows a interface escolhida deve ser do tipo TAP, independente do Sistema Operacional do outro *gateway*. Em ligações cujas pontas rodam Linux, a interface de ser do tipo TUN. Lembrando que o Linux também suporta interfaces TAP, porém interfaces do tipo TUN tem melhor desempenho em ambientes Linux.

O parâmetro *ifconfig,* tem 2 argumentos, o primeiro, no caso 192.168.0.1, define o endereço da interface virtual do servidor e o segundo, no caso 192.168.0.2 define o endereço da interface virtual do cliente. As interfaces podem ser do tipo TUN ou TAP conforme for decidido.

O parâmetro *secret* identifica o endereço do arquivo contendo a chave privada para a encriptação dos dados que serão transmitidos.

O parâmetro *port* define qual porta será utilizada. É importante que se configure a mesma porta no arquivo de configuração do cliente e do servidor. O OpenVPN por padrão utiliza a porta 1194 UDP, e, diferentemente de outras soluções VPN, partilha várias conexões na mesma porta padrão o que facilita a configuração de *firewalls* e direcionamento de portas nos *gateways* de Internet ou *modens* ADSL.

Para que OpenVPN funcione, é preciso que se direcione o tráfego da porta escolhida para a máquina em que o OpenVPN está instalado. No caso do trabalho, utilizou-se uma linha ADSL para a conexão com a Internet. Existem várias marcas de modens ADSL e cada uma tem um jeito diferente de liberação de portas, mas o processo é sempre o mesmo:

a) definir o endereço de IP da maquina em que o OpenVPN está instalado.
b) definir a porta ou intervalo de portas, na qual o OpenVPN foi configurado;
c) definir o protocolo de comunicação.

Após a definição dos parâmetros, a VPN precisa ser inicializada. A inicialização é bem simples, basta executar o comando:

```
openvpn  -- config /etc/openvpn/servidor.conf  &
```

O comando é executado em *background* para evitar qualquer tipo de interrupção no console ativo. Mesmo assim, surgirão umas linhas de mensagem com informações

referentes a versão, endereço da chave e nome da interface virtual criada, que somem ao digitar *enter*.

Se tudo estiver correto, será criado um novo adaptador de rede, que pode ser visualizado digitando *ifconfig* no console, o último adaptador da lista, geralmente chamado de tun0, é o utilizado pelo OpenVPN. Vários adaptadores podem ser usados simultaneamente, desde que vários arquivos de inicialização sejam usados. A cada novo arquivo inicializado, é somado 1 ao nome da interface para diferenciar, logo se tiver 2 arquivos inicializando, 2 interfaces serão criadas, tun0 e tun1.

6.4.2 **Configuração do cliente por meio de chaves estáticas**

No cliente, o arquivo será chamado cliente.conf, dentre as poucas alterações em relação ao arquivo do servidor, destaca-se o parâmetro *remote*, que especifica o endereço do *host* remoto.

No exemplo abaixo o *host* remoto está na rede local, se o servidor tiver um IP válido fixo na Internet basta digitá-lo no arquivo de configuração. Também há possibilidade de se cadastrar num serviço de DNS dinâmico, como o no-ip.com por exemplo, e digitar seunome.no-ip.org no parâmetro *remote*.

Vale salientar, que se o servidor estiver em outra rede, deve-se fazer o redirecionamento dos pacotes da porta selecionada no *gateway* de Internet ou modem ADSL, para o endereço do servidor na rede local.

Tabela 4 – Arquivo cliente.conf

`proto udp`
`remote 10.1.1.33`
`dev tun`
`ifconfig 192.168.0.2 192.168.0.1`
`secret /etc/openvpn/chave.key`
`port 5900`

Após a definição dos parâmetros, a VPN precisa ser inicializada. O comando é basicamente o mesmo do servidor, mudando apenas o nome do arquivo:

openvpn -- config /etc/openvpn/cliente.conf &

Nota-se também, que os argumentos do parâmetro *ifconfig* estão invertidos em relação ao servidor, o que indica que 192.168.0.2 será o endereço de IP da interface tun0 do cliente e 192.168.0.1 o endereço da interface tun0 servidor.

Para testar se tudo está funcionando no cliente, basta executar o comando:

ping 192.168.0.1

Se tudo estiver funcionando, será exibido vários retornos com o tempo de transmissão, para que o ping pare de enviar requisições ao servidor é só digitar Ctrl + C.

6.4.3 Finalizando a configuração

Após testado com sucesso na rede local, realizou-se um teste entre 2 redes com as características descritas abaixo:

servidor - 10.1.1.0/255.255.255.0

cliente - 192.168.1.0/255.255.255.0

Ambas as redes usam *links* ADSL para acesso a Internet. Para não precisar mudar constantemente o IP do servidor no arquivo de configuração do cliente, uma conta no site no-ip.com foi criada com o *host* testeopenvn.no-ip.org.

No Ubuntu Server, é preciso instalar uma aplicação que ira atualizar o IP no site no-ip a cada nova conexão do *modem* ADSL. A instalação é por linha de comando, e a configuração é bem simples, pedindo apenas o E-mail e Senha cadastrados no site. Para instalar o cliente no-ip:

apt-get install no-ip

Os arquivos de configuração são os mesmos, mudando apenas o parâmetro *remote* no cliente.

Tabela 5 – Arquivo cliente.conf usando no-ip

proto udp
remote testeopenvpn.no-ip.org
dev tun
ifconfig 192.168.0.2 192.168.0.1
secret /etc/openvpn/chave.key
port 5900

Apesar de existir uma conexão via túnel, as redes internas 10.1.1.0 e 192.168.1.0, não têm comunicação. As únicas maquinas que se comunicam são o servidor e o cliente do OpenVPN que criaram uma rede virtual na faixa 192.168.0.0. Para que as duas redes possam ficar visíveis e uma possa usufruir dos recursos da outra, é preciso que haja uma definição de rotas.

No servidor:

route add –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.0.2

O comando adiciona uma rota apontando como o *gateway* da rede 192.168.1.0 o endereço do cliente definido na configuração do OpenVPN. Para a configuração no cliente, o comando é semelhante:

route add –net 10.1.1.0 netmask 255.255.255.0 gw 192.168.0.1

No caso do cliente, é criada uma rota apontando como *gateway* da rede local do servidor (10.1.1.0) o endereço virtual da interface TUN usada na VPN.

Configuradas as rotas, é preciso acrescentar umas linhas no *firewall* liberando o acesso:

Firewall do servidor:

iptables –t nat –A POSTROUTING –o tun+ -j MASQUERADE

iptables –A FORWARD –i tun0 –s 10.1.1.0/16 –d 192.168.1.0/16 –j ACCEPT

Firewall do cliente:

iptables –t nat –A POSTROUTING –o tun+ -j MASQUERADE

iptables –A FORWARD –i tun0 –s 192.168.1.0/16–d 10.1.1.0/16 –j ACCEPT

A primeira linha libera o mascaramento e a segunda linha permite a passagem dos pacotes entre as 2 redes. Lembrando que para automatizar o processo, todos os comandos devem ser gravados no arquivo /etc/rc.local.

6.4.3.1 Posicionamento do Firewall

Para aumentar a segurança, o *firewall* das duas redes deve estar posicionado ao lado do *gateway* VPN. Nessa topologia, o *firewall* repassa o trafego criptografado que chega da rede externa ao *gateway* VPN, que o decifra e reenvia para o firewall. De posse do conteúdo puro, o *firewall* realiza os filtros e o envia para a rede local.

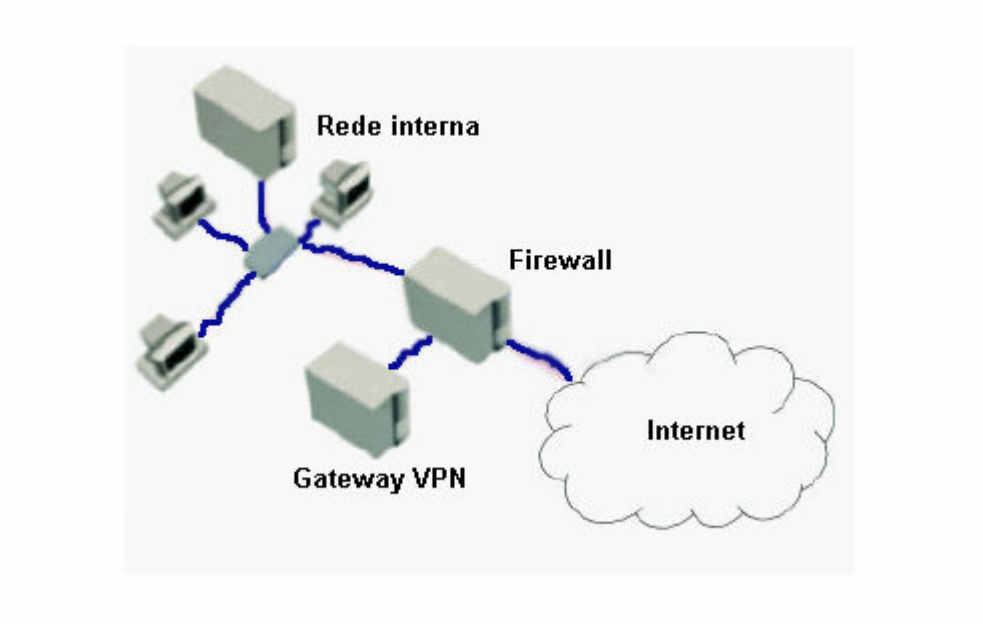


Figura 11. Posicionamento do Firewall

O *gateway* VPN também é protegido de ataques internos, pois o *firewall* realiza filtros no conteúdo da rede local. Pacotes mal intencionados destinados ao *gateway* VPN , seja da rede local ou externa passam antes pelo *firewall*.

6.4.4 Configuração por meio de chaves estáticas Windows

O OpenVPN funciona em Windows 2000, 2003, XP e Vista, não funcionando em versões mais antigas. No exemplo a seguir, será usado um servidor Linux e Cliente Windows.

Para usar o Windows, é necessário que a interface para a conexão seja TAP, necessitando apenas de algumas modificações no arquivo do servidor:

Tabela 6 – Arquivo servidorwindows.conf para clientes windows

`proto udp`
`dev tap`
`ifconfig 192.168.2.1 255.255.255.0`
`secret /etc/openvpn/chave.key`
`port 5901`

Além da mudança da interface para TAP, uma nova faixa de IP foi criada para a VPN e a porta também mudou. Em vez de especificar o endereço da outra ponta, especificou-se um endereço de IP do servidor e uma mascara para a rede. Isso permite por meio de um único arquivo, vários clientes possam se conectar ao servidor VPN.

Feito as mudanças, basta salvar o arquivo na pasta /etc/openvpn e executar o comando.

openvpn -- config /etc/openvpn/servidorwindows.conf &

Nessa situação será criada a interface tap0, que estabelecerá a comunicação com os clientes Windows.

O arquivo chave.key também precisa estar presente na maquina com Windows. Para facilitar a configuração, ele será copiado para “C:\Arquivos de programas\OpenVPN\config” . Essa pasta é a pasta padrão de configuração do OpenVPN no

Windows, sendo que todos os arquivos de configuração que estiverem nessa pasta serão iniciados automaticamente quando este serviço for executado, desde a extensão do arquivo seja .ovpn.

Tabela 7 – Arquivo cliente.ovpn

`proto udp`
`remote testeopenvpn.no-ip.org 5901`
`dev tap`
`ifconfig 192.168.2.2 255.255.255.0`
`secret "C:\\Arquivos de programas\\OpenVPN\\config\\chave.key"`

Os parâmetros são muito semelhantes, o que muda é que a porta pode ser definida no parâmetro *remote* após o endereço do *host*.

No parâmetro *ifconfig*, deve-se especificar um endereço na mesma faixa de IP da rede iniciada no servidor.

A localização do arquivo contendo a chave de criptografia deve ser colocada com duplicidade de barras, senão as mesmas serão interpretadas como comando do *shell*, além de obrigatoriamente ser delimitada por aspas duplas.

Para testar o arquivo configurado, basta executar o utilitário “OpenVPN GUI”que todos os arquivos gravados na pasta *config* com a extensão .ovpn serão executados automaticamente. Também pode-se clicar com o botão direito sobre o arquivo e clicar na opção "Start OpenVPN on this configuration file" que o arquivo será executado.

Uma janela de status aparecerá com informações referentes ao processo de conexão. Nela, pode-se desconectar e conectar o cliente Windows do servidor. Essa janela é ocultada assim que o processo de conexão for finalizado, e pode ser aberta clicando no ícone.

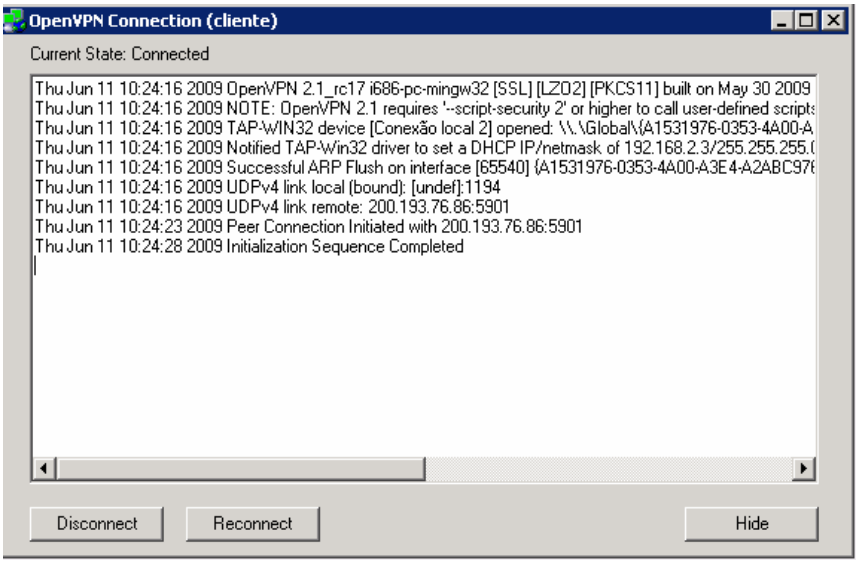


Figura 12. Status do OpenVPN no Windows

Apenas por motivos de teste, foi configurado o Samba no servidor para testar o compartilhamento de arquivos via VPN. O Samba é um utilitário que viabilizada o compartilhamento de arquivos e impressoras entre máquinas Windows e Linux. Em ambiente Microsoft, foi possível acessar uma pasta compartilhada no servidor e gravar um arquivo lá. Lembrando que o compartilhamento deve ter permissão de escrita.

No Windows pasta abrir uma janela do explorer e digitar \\IP ou nome do servidor no caso \\192.168.2.1 exatamente do mesmo modo em que se faz em uma rede local. A figura 11 demonstra o exemplo citado.

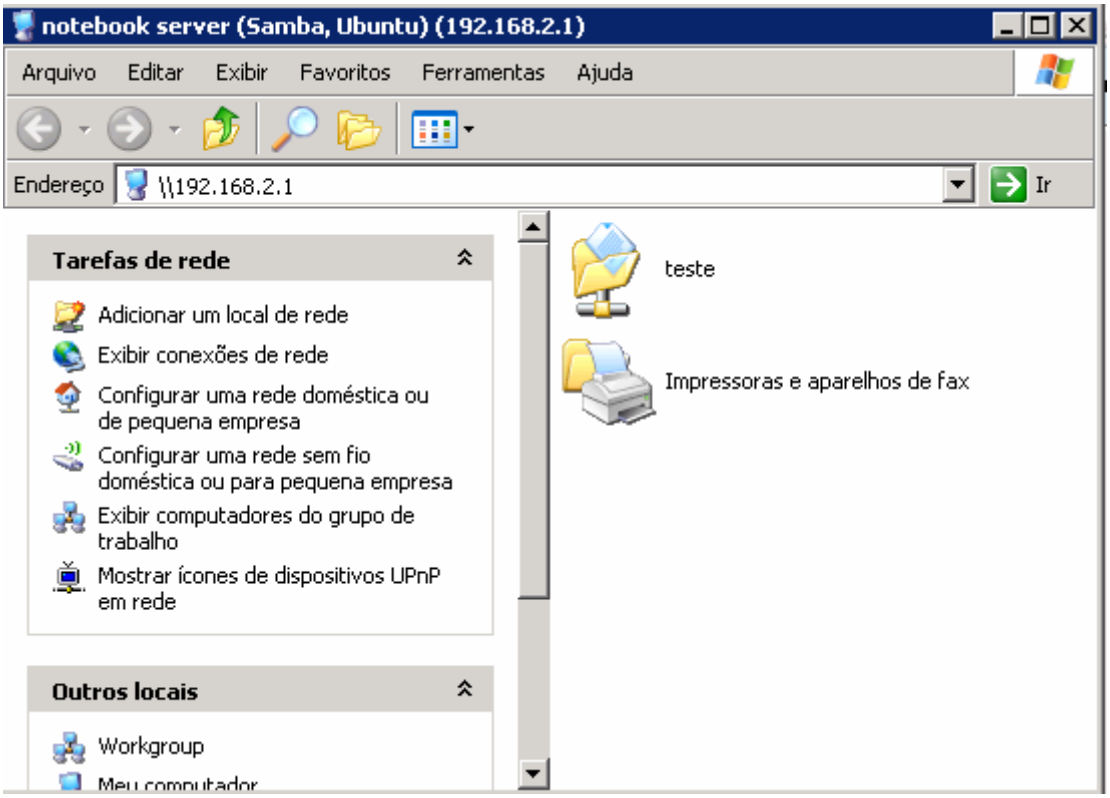


Figura 13. Acesso a compartilhamento via VPN

Até agora foram apresentadas configurações básicas, que tornaram a VPN operacional, porém com algumas limitações. Verificou-se problemas de estabilidade, principalmente porque os endereços IP das conexões ADSL mudam constantemente, obrigando o cliente a reiniciar o serviço. Vale lembrar, que o OpenVPN também pode compactar os dados para aumentar o desempenho em caso de grande tráfego de dados ou para uso em redes congestionadas.

Uma série de modificações se fazem necessárias para otimizar a VPN. Essas modificações são feitas por meio de uma série de parâmetros que tratam as mais diversas situações.

6.4.5 Parâmetros de otimização

Como foi verificado até agora, problemas com queda de sinal fazem com que o cliente seja obrigado a reiniciar o serviço. Existe um parâmetro que faz com que o servidor e o cliente monitorem a conexão, enviando *pings* periódicos um para o outro. Caso não haja resposta, a VPN é reiniciada automaticamente depois de determinado tempo. Esta opção é muito útil para uso em redes com constante queda de sinal, e sobre *links* ADSL. Segue exemplo:

keepalive 10 120

O primeiro argumento especifica o intervalo de requisições, ou seja, um *ping* é enviado a cada 10 segundos sem atividade. O segundo argumento define o tempo em segundos para a VPN seja reiniciada, ou seja, a VPN é reiniciada depois de 120 segundos sem respostas nos *pings*. Esse parâmetro deve ser configurado no cliente e no servidor.

O parâmetro *keepalive* é o mais recomendado, porém existem outros parâmetros que também tratam de controle de fluxo e não serão usados (MARCELO, 2007):

a) **ping** <segundos> - emite um ping para a outra ponta do túnel após um número específico de segundos sem tráfego;
b) **ping-restart** <segundos> - depois de um número específico de segundos sem tráfego a conexão do túnel é reiniciada;
c) **ping-exit** <segundos> - se num período específico de segundos o túnel ficar inativo, a conexão entre as pontas é finalizada;
d) **inactive** <segundos> - faz com que dispositivos tun/tap sejam desativados depois de um numero especifico de segundos inativo.

e) **resolv-retry** <segundos>**-** o OpenVPN tenta resolver o nome do *host* depois de um período especifico de segundos. Se não conseguir, não tentará novamente.

Outras duas opções, que ajudam a resolver problemas de conectividade, são a ***persist-key*** e a ***persist-tun***. Elas fazem com que o OpenVPN mantenha a interface TUN aberta e as chaves carregadas durante o processo de restauração do link em caso de queda de sinal. Esses parâmetros tornam o processo de reconexão mais rápido e eficiente:

persist-key

persist-tun

Existe também um parâmetro, denominado ***float*** que mantém o túnel aberto mesmo que o endereço IP da outra ponta mude. Sem o uso desse parâmetro, a mudança de IP faria com que a conexão fosse encerrada e o túnel fosse interrompido até que fosse reiniciado. Lembrando que o túnel pode ser reiniciado manualmente ou pelo uso do ***keepalive***. Para funcionar, esse parâmetro deve ser incluído na configuração do servidor e na do cliente:

float

O uso de *links* ADSL como *gateways* de Internet pode se tornar um problema. Em determinados horários o desempenho cai consideravelmente. Para o uso em redes congestionadas o OpenVPN pode compactar os dados transmitidos, aumentando o desempenho da VPN. A compactação de dados utilizada pelo OpenVPN não consome muitos recursos de processamento, o que viabiliza seu uso. Para ativar a compactação de dados, basta inserir o parâmetro a seguir no arquivo de configuração do cliente e do servidor:

comp-lzo

Lembrando que para a compactação funcionar, o pacote LZO deve estar instalado.

Com as diversas melhorias feitas, os arquivos de configuração do cliente e servidor Linux ficaram assim:

Tabela 8 – Arquivos de configuração com chave estática otimizados

servidor.conf	**Cliente.conf**
`proto udp`	`proto udp`
`dev tun`	`remote testeopenvpn.no-ip.org`
`ifconfig 192.168.0.1 192.168.0.2`	`dev tun`
`comp-lzo`	`ifconfig 192.168.0.2 192.168.0.1`
`float`	`comp-lzo`
`keepalive 10 120`	`float`
`persist-key`	`keepalive 10 120`
`persist-tun`	`persist-key`
`secret /etc/openvpn/chave.key`	`persist-tun`
`port 5900`	`secret /etc/openvpn/chave.key`
	`port 5900`

Até agora foram apresentados exemplos de arquivos de configuração que usam chaves estáticas para criptografar os dados transmitidos. Esse modo é o mais usado devido a facilidade de configuração, porém tem recursos limitados. Existe a possibilidade de usar uma configuração mais elaborada, utilizando certificados X509. Este método é denominado *Publik Key Infraestruture* (PKI) e permite criar VPN mais seguras.

6.5 CONFIGURAÇÃO POR MEIO DE CERTIFICADOS X509

O OpenVPN permite uma série de outras opções, que podem em muito melhorar a performance e a segurança em conexões remotas privadas.

Em sua implementação mais segura, o OpenVPN suporta autenticação baseada em certificados X509. Essa é autenticação é bi-direcional, ou seja, o servidor verifica a autenticidade do cliente e o cliente a autenticidade do servidor.

Para que a autenticação bi-direcional se torne possível, é necessário criar uma estrutura PKI, com uma Autoridade Certificadora responsável por gerar os certificados dos clientes e do servidor.

Existem duas opções para construir uma PKI. A primeira seria trabalhar diretamente com a biblioteca OpenSSL, disponível no Linux, e que é quem realmente gerencia a PKI. A segunda é utilizar a interface **easy-rsa,** disponibilizada pelo OpenVPN para simplificar os comandos da OpenSSL. A opção escolhida foi a primeira, pois trabalha diretamente com comandos do OpenSSL, o que representa a oportunidade de adquirir novos conhecimentos, já que essa biblioteca pode ser usada para outros fins.

6.5.1 Criando a Autoridade Certificadora

Ao contrário do que o nome sugere, criar uma Autoridade Certificadora é relativamente simples. Primeiramente é preciso verificar se o utilitário OpenSSL está instalado. Como foi verificado anteriormente, uma versão mais atual do OpenSSL é instalada automaticamente quando o OpenVPN é instalado.

Verificando que o OpenSSL está presente, é preciso editar algumas opções no arquivo de configuração, que no caso do Ubumtu Server encontra-se em /etc/ssl/openssl.cnf:

Tabela 9 – parâmetro alterados no arquivo openssl.cnf

1	dir = /etc/openvpn
2	Database = $dir/index.txt
3	new_certs_dir = $dir/aux
4	certificate = $dir/my-ca.crt
5	serial = $dir/serial
6	private_key = $dir/my-ca.key
7	default_days = 730

Explicando as mudanças feitas:

a) linha 1: especifica o local onde serão armazenados os certificados, no caso especificamos o diretório do openvpn;

b) linha 2: arquivo indexador do banco de dados do SSL que precisar ser criado em branco no diretório selecionado;

c) linha 3: especifica o diretório onde serão armazenados os novos certificados, precisa ser criado em /etc/openvpn;

d) linha 4: especifica o certificado da autoridade certificadora, que ainda não foi criado;

e) linha 5: é preciso criar um arquivo com um numero que possa ser incrementado;

f) linha 6: especifica o endereço da chave privada que ainda não foi criada e não deve sair do servidor;

g) linha 7: especifica do tempo de duração dos certificados, no caso 730 dias.

Alguns arquivos e diretórios configurados, precisam ser criados no diretório /etc/openvpn :

Tabela 10 – criação de arquivos e diretório

`cd /etc/openvpn`
`touch index.txt`
`echo 01 > serial`
`mkdir aux`

Com os arquivos criados, chegou a hora de gerar a chave privada e o certificado da Autoridade Certificadora, esses arquivos serão chamados de *my-ca.key* e *my-ca.c*rt e deverão ser gravados no diretório /etc/openvpn:

openssl req –nodes –new -509 –keyout my-ca.key –out my-ca.crt –days 730

Serão solicitadas algumas informações que serão gravadas no certificado referentes a País, Estado, Cidade, nome da empresa, nome do setor, nome e e-mail do administrador.

Se não ocorrer nenhum erro, a Autoridade Certificadora responsável pela geração dos certificados esta gerada, caso apareça algum erro deve-se verificar a sintaxe do comando e se os arquivos e diretórios foram criados em /etc/openvpn.

6.5.2 Gerando parâmetros Diffie-Hellman

O Diffie-Hellman foi o primeiro algoritmo de chave pública, descoberto em meados da década de 70.

Permite que haja a troca de chaves públicas entre duas ou mais partes, permitindo que as entidades que recebam a chave pública, utilizem esta chave para criptografar o conteúdo de uma mensagem que será enviada à parte que fornecer a chave pública.

Esse conteúdo criptografado não poderá ser aberto por partes que possuam a chave pública, e sim apenas pela parte que enviou a chave pública, pois a mesma possui a chave privada necessária para descriptografar a mensagem.

Para gerar os parâmetros Diffie-Hellman basta executar um comando no OpenSSL:

openssl dhparam –out dh.pem 1024

Esse comando irá gerar um arquivo chamada dh.pem que deve estar presente no servidor e nos clientes.

Os arquivos criados até o momento são:

a) **my-ca.crt** - Certificado mestre da Autoridade Certificadora;

b) **my-ca.key** - Chave privada mestre da Autoridade Certificadora;

c) **dh.pem-** arquivo do Diffie-Hellman.

Com isso, a PKI foi inicializada e os arquivos mestre da Autoridade Certificadora estão criados. Pode-se agora partir para a criação dos certificados/chaves privadas para o servidor e os clientes.

6.5.3 Gerando certificados públicos e privados

Chegou a hora de gerar os certificados públicos e privados que serão utilizados na VPN. Para gerar os arquivos do servidor:

openssl req –nodes –new –keyout servidor.key –out servidor.csr

openssl ca –out servidor.crt –in servidor.csr

O primeiro comando cria um pedido de certificado e o segundo comando faz com que a Autoridade Certificadora assine o certificado, concordando que o mesmo é válido. Serão solicitadas as mesmas informações referente a País, Estado dentre outras, que foram solicitadas na criação da Autoridade Certificadora. Além dessas informações, o OpenSSL solicita uma senha que será usada cada vez que o certificado for usado. No caso do uso de certificados em VPNs, o uso de senhas não é interessante, pois reduz a praticidade, devido a que cada vez que o certificado for usado será pedida a senha, então o campo senha deve ficar em branco.

No segundo comando, o OpenSSL solicitará confirmação de assinatura do certificado e confirmação de criação do conjunto certificado/chave privada . Deve-se confirmar as 2 perguntas, caso contrário a geração do certificado será abortada:

```
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit?
[y/n]y
```

O comando gerou os arquivos *servidor.key* e *servidor.crt* que serão usados pelo servidor e devem estar no diretório /etc/openvpn. O arquivos *servidor.crs* pode ser descartado, pois é usado somente na geração do certificado.

Os certificados públicos e privados dos clientes também devem ser gerados no servidor. O servidor pode gerar vários certificados usando a mesma sintaxe mudando apenas o nome do arquivo:

openssl req –nodes –new –keyout cliente.key –out cliente.csr

openssl ca –out cliente.crt –in cliente.csr

As mesmas perguntas e mensagens aparecerão. O processo de criação é semelhante. Após serem gerados os arquivos necessários, chegou a hora de configurar servidor e o cliente.

6.5.4 Configurando o servidor no modo TLS

Uma vez definida a estrutura de PKI e geradas as chaves e certificados necessários, algumas mudanças deverão ser feitas no arquivo de configuração do servidor, para que ele possa trabalhar no modo TLS.

Tabela 11 – Arquivo servidor.conf modo TLS

proto udp
dev tun
ifconfig 192.168.0.1 192.168.0.2
comp-lzo
Float
keepalive 10 120
persist-key
persist-tun
tls-server
dh /etc/openvpn/dh.pem
ca /etc/openvpn/my-ca.crt
cert /etc/openvpn/servidor.crt
key /etc/openvpn/servidor.key
port 5900

A *tls-server* identifica que é um arquivo de configuração de servidor. Nela são especificados quatro novos parâmetros:

a) **dh**- define a localização do arquivo com os parâmetros Diffie-Hellman usado para troca de chaves públicas, este parâmetro está presente apenas no servidor;

b) **ca**- define a localização do arquivo contendo o certificado da Autoridade Certificadora. Este arquivo deve estar presente, no servidor e nos clientes;
c) **cert** – define a localização do arquivo contendo o certificado do servidor. Este arquivo deve estar presente apenas no servidor, sendo que o cliente terá seu próprio arquivo de certificado gerado pela Autoridade Certificadora;
d) **key** – define a localização do arquivo contendo a chave privada do servidor.

Feitas as mudanças no arquivo de configuração e colocados os arquivos nos devidos lugares é só reiniciar o OpenVPN:

/etc/ini.d/openvpn restart

Caso apareça algum erro, deve-se verificar a localização de todos os arquivos. Para facilitar e padronizar todos os arquivos são gravador em /etc/openvpn.

6.5.5 Configurando o cliente no modo TLS

As alterações no arquivo *cliente.conf*, são poucas em relação ao arquivo *servidor.conf*. Tirando as mudanças já observadas anteriormente no capitulo da conexão por chaves estáticas, a única diferença em relação ao servidor é que na linha *tls-client* o parâmetro **dh** que direciona para arquivo contendo os parâmetros *Diffie-Hellman* não esta presente.

Tabela 12 – Arquivo cliente.conf modo TLS

proto udp
dev tun
remote testeopenvpn.no-ip.org
ifconfig 192.168.0.2 192.168.0.1
comp-lzo
Float
keepalive 10 120
persist-key
persist-tun
tls-client
ca /etc/openvpn/my-ca.crt
cert /etc/openvpn/cliente.crt
key /etc/openvpn/cliente.key
port 5900

Lembrando, que os arquivos *my-ca.crt*, *cliente.key*, e *cliente.conf* devem ser copiados do servidor para o diretório /etc/openvpn.

Depois de tudo configurado , basta reiniciar o OpenVPN e fazer um teste de ping para verificar eventuais problemas.

6.5.6 Revogando Certificados

O OpenVPN permite a revogação dos certificados que eventualmente ofereçam risco a segurança da informação na empresa. Esta opção é muito importante no caso de roubo de uma maquina que tenha o certificado armazenado ou no caso de um funcionário que se afaste da empresa. Se os certificados não forem revogados, qualquer um em posse deles poderá se conectar à VPN e, possivelmente, acessar à rede interna da empresa.

No servidor, os novos certificados ficam armazenados na pasta criada durante a configuração em /etc/openvp/aux. Nesse diretório, vão ter vários arquivos com a extensão .pem . Para saber qual o certo:

fgrep " usuário"."domínio" *

A saída do comando vai retornar algumas informações sobre o certificado pesquisado, inclusive o nome do arquivo. Para revogar o certificado:

openssl ca -revoke arquivo.pem

Ao terminar o processo, será exibida uma mensagem que o certificado foi revogado e que a base de dados do OpenSSL foi atualizada.

Neste ponto encerra-se a implementação da VPN proposta, a seguir serão demonstrados os resultados obtidos e sugestões para trabalhos futuros.

6.6 TESTE DE DESEMPENHO

Para a execução do teste, foi copiada uma pasta de 6mb com arquivos dos mais diferentes formatos para um compartilhamento no servidor. Para tal, considerou-se 5 situações:

a) uso de criptografia com chave estática e com compactação;
b) uso de criptografia com chave estática e sem compactação;
c) uso de criptografia com certificados digitais e com compactação;
d) uso de criptografia com certificados digitais e sem compactação;
e) sem criptografia e com compactação.

O cenário escolhido foi o apresentado no decorrer do trabalho, com 2 *hosts* de redes distintas interligados por VPN, e usando *links* ADSL.

Escolheu-se um horário de pouco tráfego, para que o resultado não fosse influenciado por aplicações concorrentes de banda nas 2 redes.

Tabela 13 – Teste de desempenho

Situação	**Tempo (MIN)**
uso de criptografia com chave estática e com compactação	5:30
uso de criptografia com chave estática e sem compactação	6:43
uso de criptografia com certificados digitais e com compactação	5:45
uso de criptografia com certificados digitais e sem compactação	7:32
sem criptografia e com compactação	5:15

A partir deste teste, conclui-se que o uso de compactação sempre melhora o desempenho da VPN , independente do *overhead* de processamento criado pelo uso da criptografia. Esse *overhead* fica claro observando-se a pouca diferença de tempo entre o melhor desempenho com criptografia e o desempenho sem criptografia.

Não foi testada a situação sem criptografia e sem compactação, pois a VPN busca implementar mecanismos de segurança por meio de criptografia e esse cenário dificilmente será usado.

O tamanho das chaves também pode influenciar no resultado final. As chaves utilizadas nos testes são as mesmas criadas anteriormente.

A diferença de tempo dos resultados com e sem compactação demonstrou que o uso de compactação por meio da biblioteca LZO melhora o desempenho consideravelmente.

6.7 RESULTADOS OBTIDOS

A pesquisa consistiu-se na instalação de um sistema *open source* denominado OpenVPN que interligou duas redes distintas, utilizando a Internet como meio de comunicação. Este cenário serviu para análises de vantagens e desvantagens no uso desse

sistema. Pode-se realizar vários testes de desempenho e verificações de funcionalidade desse sistema.

O uso da Internet como meio de comunicação, utilizando técnicas de criptografia e tunelamento mostrou-se uma excelente alternativa. De início, com o uso de chaves estáticas e poucos parâmetros de controle a VPN ficou um pouco instável, principalmente devido a mudanças no endereço de Ip do servidor e do cliente que usavam *links* ADSL para a conexão com Internet, obrigando que o serviço fosse reiniciado a cada vez que o IP mudasse.

Utilizando parâmetros para controle de mudança de endereço de Ip e restabelecimento de conexão, o problema de estabilidade foi resolvido, faltando apenas configurar um cliente de DNS dinâmico, que garantiu que o servidor estivesse disponível a cada troca de Ip, não havendo mais a necessidade de reconfigurar o OpenVPN.

Como a segurança é o principal foco das VPNs, o uso de criptografia permitiu que os dados trafegados na Internet se tornassem ilegíveis a quem não tivesse a chave para descriptografá-los. Mesmo assim, foram adicionados mecanismos adicionais de segurança, como certificados digitais que aumentaram ainda mais o nível de segurança e autenticação bi-direcional, onde o servidor verifica a autenticidade do cliente e o cliente a autenticidade do servidor.

Além do uso de criptografia para melhorar a segurança e o uso de DNS dinâmico para melhorar a estabilidade, também foi configurada a compactação dos dados por meio da biblioteca LZO. O uso de compactação é muito útil em redes muito congestionadas ou com pouca largura de banda. O seu uso aumentou o desempenho na transferência de arquivos em horários considerados de pico, onde o uso de banda é muito concorrido.

Os pontos negativos verificados no uso do OpenVPN foram: escassez de manuais e livros em português e a complexidade de configuração devido a ser toda baseada na criação e modificação de arquivos de texto.

6.8 CUSTOS RELACIONADOS

Estimativa de custos envolvidos para implementação dessa solução.

Tabela 14 – Custos envolvidos

Equipamento/Serviço R$	**Valor R$**
01 Micro Pentium 4 1.7 GHz	1400,00
01 Micro Semprom 2800+	1200,00
01 Sistema Operacional Linux Ubuntu Server 8.4 TLS	0,00
01 Software OpenVPN	0,00
01 mão-de-obra para intalaçao 5 *Horas	600,00
01 Capacitação 10 *Horas	1200,00
TOTAL:	**4400,00**

* Custo estimado de um profissional da área.

Obs: não foram considerados custos referentes estrutura de redes, tais como *modens* ADSL, *Switches* e cabeamento estruturado, pois empresas interessadas em usar esse sistema normalmente já têm a estrutura de rede pronta.

Como já foi dito anteriormente, os *links* dedicados tem um custo alto. Em pesquisa realizada com algumas empresas, constatou-se que o custo de um link de 1024kbps fica em torno de 1400,00 por mês, dependendo da região.

Links adsl de 4096kbps custam em torno de 140,00 reais por mês, representando uma economia mensal de 1000%. Essa relação custo x benefício aumenta ainda mais se o número de filiais for maior, pois um novo link dedicado é necessário para cada nova ponta da VPN.

Diante dos valores apresentados, fica evidente que a implementação de uma VPN, pode reduzir consideravelmente os custos.

CONCLUSÃO

As pesquisas feitas ao longo do desenvolvimento deste trabalho permitiram constatar que VPN é uma tecnologia com amplo potencial de mercado. A necessidade de expansão das empresas, torna a procura por serviços de VPN cada vez maior. Existem várias soluções proprietárias, cada qual com seus prós e contras.

Neste trabalho explorou-se os aspectos envolvidos na implantação de Redes Privadas Virtuais, provando que é possível diminuir boa parte dos custos usando a solução OpenVPN, sem perder em qualidade e desempenho.

Os objetivos desse trabalho foram atingidos, pois necessitou-se compreender os fundamentos teóricos referentes aos conceitos de criptografia, tunelamento, protocolos e topologias. Foi possível visualizar com clareza a utilização dos diversos mecanismos de segurança e as questões envolvidas na sua escolha. A implantação de um sistema para interligação de redes usando SSL foi bem sucedida. Foi possível verificar os benefícios e as dificuldades dessa tecnologia.

Apesar de ter-se conseguido alcançar os objetivos do trabalho, alguns temas não foram totalmente abordados, ficando a sugestão para trabalhos futuros: implementar uma VPN utilizando o OpenVPN em modo *bridge*, que permita o tráfego em *broadcast*, o que pode sacrificar um pouco o desempenho, devido ao uso adicional de banda, mas permite uma série de recursos encontrados somente redes locais, como por exemplo a instalação automática de impressoras. Outra sugestão seria implementar uma VPN usando o protocolo IPSec e compará-la com a implementada nesse trabalho apontando as diferenças, os pontos positivos e os pontos negativos.

O estudo efetuado proporcionou crescimento pessoal e profissional. O empenho nas pesquisas e testes, juntamente conhecimento adquirido na área de segurança em redes, tornaram o trabalho gratificante.

REFERÊNCIAS

FANELI, André Luiz Perciano; MARCHEZINI, Vanderlei Carlos. **OPENVPN:** IMPLEMENTAÇÃO DE VPN ATRAVÉS DE SSL. Vitória: Biblioteca da Faculdade Salesiana de Vitória, 2007. 133 p. Disponível em: <http://www.multicast.com.br/sergio/arquivos/monografia-pos-seguranca-openvpn-com-ssl-no-linux.pdf>. Acesso em: 10 março 2009.

GALVÃO, Ricardo Kléber Martins. OpenVPN: Rápido e Prático. In: ENCONTRO POTIGUAR DE SOFTWARE LIVRE, 3., 2007, Natal. **Anais... .** Natal, Rn: Cefet-rn, 2007. p. 1 - 28. Disponível em: <http://rn.softwarelivre.org/portal/files/rk-openvpn.pdf>. Acesso em: 18 maio 2009.

GUIMARÃES, Alexandre Guedes; LINS, Rafael Dueire; OLIVEIRA, Raimundo. **Segurança com Redes Privadas Virtuais.** 2. ed. Rio de Janeiro:Brasport, 2006.

GUIZZO, Erico. **Internet**: o que é, o que oferece, como conectar-se. São Paulo: Ática, 2002.

KUROSE, James F.; ROSS, Keith W. **Redes de Computadores e a Internet.** Uma abordagem top-down . 3. ed São Paulo: Pearson Addison Wesley, 2006.

LUNARDI, Marco A.**Redes de computadores**:prático e didático. Rio de Janeiro: Ed. Moderna, 2007.

MARCELO, Antonio. **OPEN VPN.** Rio de Janeiro: Brasport, 2007. 81 p. (Guia rápido do administrador de rede).

NORTHCUTT, Stephen; NOVAK, Judy; MCLACHLAN, Donald. **Segurança e prevenção em redes**. São Paulo: Berkeley, 2001.

OPENVPN - AN OPEN SOURCE SSL VPN SOLUTION. Disponível em: < http://openvpn.net >. Acesso em: 16/03/09.

PETERSON, Larry L; DAVIE, Bruce S. **Redes de Computadores**: uma abordagem de sistemas. Rio de Janeiro: Elsevier, 2004.

REZENDE, Edmar Roberto Santana de. **Segurança no Acesso Remoto VPN.** 2004. 121 f. Dissertação (Mestrado) - Unicamp, Campinas, 2004. Disponível em: <http://www.las.ic.unicamp.br/paulo/teses/20040227-MSc-Edmar.Roberto.Santana.de.Rezende-Seguranca.no.acesso.remoto.VPN.pdf>. Acesso em: 18 abril 2009.

RICCI, Bruno. **Redes Segura**:vpn em linux. Rio de Janeiro: Moderna, 2007.

SILVA, Lino Sarlo da. **Virtual Private Network - VPN.** 2. ed. São Paulo: Novatec, 2005. 233 p.

SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 20., 2002 MAIO 20-24. BÚZIOS, RJ; PIRMEZ, LUCI; CARMO, LUIZ FERNANDO RUST DA COSTA; MACÊDO, RAIMUNDO J; SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. Anais ... Búzios: [s.n.], 2002. 157.

SOARES, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Sérgio. **Redes de Computadores.** Das LANs, MANs e WANs às Redes ATM. 2. ed. Rio de Janeiro: Ed. Campus, 1995.

SPURGEON,Charles E.**Ethernet**:o guia definitivo. Rio de Janeiro: Campus, 2000.

TANENBAUM, Andrew S. **Redes de computadores**. 4.ed. Rio de Janeiro: Campus, 1997.

T. Dierks; C. Allen. RFC-2246 – *Transport Layer Security* http://www.ietf.org/rfc/rfc2246.txt?number=2246, acesso em: 10/02/2009.

APÊNDICE A – Arquivo *servidor.conf* comentado

```
#Arquivo de configuração do servidor OpenVPN
# O Caractere # é usado para delimitar comentários
#
#Define o protoloco a ser usado (TCP ou UDP)
proto udp
#
#Define o tipo de Interface a ser usada (TUN ou TAP)
#Quando se trabalha com Windows usar TAP
#Quando se trabalha com Linux, TUN , apesar de que TAP também
seja operacional
dev tun
#
#define o endereço do servidor e do cliente
#pode-se usar a sintaxe
#ifconfig 192.168.0.1 255.255.255.0
#para que vários clientes possam ser conectados na mesma vaixa
ifconfig 192.168.0.1 192.168.0.2
#
#ativa a compactação dos dados transmitidos por meio da
#biblioteca lzo se for ativada do servidor, deve ser ativada
#nos clientes
comp-lzo
#mantém o túnel aberto mesmo que o endereço IP da outra ponta
#mude
float
#
#O comando keepalive serve para estabilizar a conexão
#O primeiro argumento especifica o intervalo de requisições,
#ou seja, um ping é enviado a cada 10 segundos sem atividade.
#O segundo argumento define o tempo em segundos para a VPN
#seja reiniciada,
#Esse parâmetro deve ser configurado no cliente e no servidor.
keepalive 10 120
#
#Fazem com que o OpenVPN mantenha a interface TUN aberta e as
#chaves carregadas durante o processo de restauração do link
#em caso de queda de sinal.
persist-key
persist-tun
#
#dh- define a localização do arquivo com os  parâmetros
#Diffie-Hellman  usado para troca de chaves públicas, este
#parâmetro está presente apenas no servidor;
#ca- define a localização do arquivo contendo o certificado
#da Autoridade Certificadora. Este arquivo deve estar
#presente, no servidor e nos clientes;
```

```
#cert – define a localização do arquivo contendo o
#certificado do servidor. Este arquivo deve estar presente
#apenas no servidor, sendo que o cliente terá sem próprio
#arquivo de certificado gerado pela Autoridade Certificadora
#key – define a localização do arquivo contendo a chave
#privada do servidor.
tls-server
dh /etc/openvpn/dh.pem
ca /etc/openvpn/my-ca.crt
cert /etc/openvpn/servidor.crt
key /etc/openvpn/servidor.key
#
# Define qual porta será utilizada
port 5900
```

APÊNDICE B – Instalação do Ubumtu Server

A distribuição Linux Ubuntu server versão 8.4 pode ser obtida solicitando o cd que é entregue gratuitamente via correio ou realizando *download* do cd de instalação. Ambas as opções estão disponíveis no site http://www.ubuntu-br.org.

Após a obtenção do cd de instalação, deve-se configurar o *setup* da máquina para iniciar por meio do *cd-rom*, reiniciá-la e executar os passos abaixo:

a) responder algumas questões que nada interferem no uso do sistema, como escolha do *layout* de teclado e ajuste da hora;
b) tipo de instalação: qualquer tipo deverá funcionar, porém, partindo do pressuposto que quanto mais software instalado maior a possibilidade de erros, usar a instalação mínima;
c) particionamento:a melhor opção é o particionamento automático, pois o instalador define swap e sistema de arquivos automaticamente. Caso haja outros sistemas operacionais instalados na maquina o particionamento manual é necessário. Uma *swap* com 512mb já é suficiente e o sistema de arquivos recomendado é o EXT3. O tamanho da partição EXT3 varia de acordo com a quantidade de aplicações se deseja instalar.
d) configuração da rede: deve-se escolher um nome para a máquina e configurar um IP da rede fixo, para facilitar a liberação de portas no roteador;
e) a próxima tela é para a criação de um usuário não root, onde se define o nome de usuário e senha;
f) por ultimo, o instalador apresenta uma serie de serviços, como por exemplo servidor de DNS, servidor de email, servidor de impressão dentre

outros, caso uma das opções for escolhida o instalador instala e configura todos os pacotes necessários, caso não for, somente pacotes padrão serão instalados.

Após a instalação do sistema operacional , é interessante definir uma senha para o usuário *root*, para que todas as instalações e configurações necessárias possam ser feitas sem maiores problemas. Para definir a senha de *root* basta digitar o comando *sudo passwd* , o sistema ira pedir a senha do usuário cadastrado durante a instalação e posteriormente a nova senha de *root*.