FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA DO ESTADO DO RIO DE JANEIRO

INSTITUTO SUPERIOR DE TECNOLOGIA EM CIÊNCIA DA COMPUTAÇÃO

LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA

OUTROS TRABALHOS EM:
www.projetoderedes.com.br

Uma Implementação de VPN

Bruno Alves Fagundes

Petrópolis – RJ
2007

Bruno Alves Fagundes

Uma Implementação de VPN

Monografia de Graduação apresentada ao Instituto Superior de Tecnologia em Ciências da Computação de Petrópolis de como parte das exigências do Curso FORTIC – Tecnologia da Informação e da Comunicação.

Orientador:
Fábio Borges de Oliveira

Petrópolis
Rio de Janeiro – Brasil
2007

Bruno Alves Fagundes

Uma Implementação de VPN

Monografia de Graduação apresentada ao Instituto Superior de Tecnologia em Ciências da Computação de Petrópolis de como parte das exigências do Curso FORTIC – Formação em Tecnologia da Informação e da Comunicação.

Fábio Borges de Oliveira
(Orientador)

Antônio Tadeu Azevedo Gomes
LNCC/MCT

Ercilia de Stefano
COPPE/UFRJ

José Rodrigues Fernandes
IST/FESO/Petrobrás

Petrópolis
Rio de Janeiro – Brasil

2007

RESUMO

A interconexão de redes por um meio público é uma alternativa muito atraente para empresas que necessitam compartilhar seus recursos com outras empresas ou funcionário, no entanto trafegar esses dados sem nenhuma forma de proteção é um risco muito alto. A tecnologia de VPN permite esta conexão de forma segura e utiliza a Internet para este fim.

Este trabalho irá apresentar alguns conceitos de segurança que servem como base para o entendimento do funcionamento de uma VPN. Serão abordadas as principais formas de criptografia no meio computacional, assim como também serão levados em consideração outras formas de segurança como firewall, por exemplo.

Ao final desta introdução tecnológica este trabalho apresentará os principais protocolos utilizados para criar e gerenciar os túneis, apresentando seus pontos positivos e negativos.

ABSTRACT

The net interconnection through a public channel is a very attractive alternative for the companies which need to share their resources with others companies or employees. The VPN technology allow this connection and the Internet can be used for this end.

This work will show some security conceptions that are useful to understand a VPN work. The main ways of cryptography in the computational world won't be taken for granted, as well as other means of security (firewall, for example).

At the end of the technological introduction this work will present the main used protocols to create and to manage the tunnels, presenting their positive and negative points.

LISTA DE FIGURAS

Figura 1 – Atacante Passivo e Atacante Ativo 12
Figura 2 – Processo de criptografia 14
Figura 3 – Criptografia simétrica 15
Figura 4 – Criptografia simétrica / número de chaves 16
Figura 5 – Criptografia assimétrica 18
Figura 6 – Assinatura digital 22
Figura 7 – Men-in-the-middle attack 25
Figura 8 – Filtro de pacotes 29
Figura 9 – Screening Router 29
Figura 10 – Servidor Proxy 30
Figura 11 – Proxy (Camada de aplicação) 31
Figura 12 – Firewall Stateful 32
Figura 13 – Arquitetura dual home-host 33
Figura 14 – Screened host 34
Figura 15 – Arquitetura Screened Subnet 35
Figura 16 – Túnel VPN 37
Figura 17 – Transporte subjacente 38
Figura 18 – Topologia Host-host 39
Figura 19 – Topologia Host-gateway 39
Figura 20 – Gateway-gateway 40
Figura 21 – Conexão PPTP 42
Figura 22 – Tunelamento PPTP 43
Figura 23 – Processo de comunicação PPTP 44
Figura 24 – Conexão L2F 46
Figura 25 – Conexão L2TP 47
Figura 26 – Encapsulamento L2TP 49
Figura 27 – IPSec modo transporte 50
Figura 28 – IPSec modo túnel 50
Figura 29 – Campos do cabeçalho AH 51
Figura 30 – Pacote ESP 52
Figura 31 – Conexão SSL 55
Figura 32 – Camada de Atuação do SSL 56
Figura 33 – Mensagem do protocolo handshake 56
Figura 34 – Funcionamento do protocolo SSL Record 58
Figura 35 – Cenário de implementação 1 59
Figura 36 – Cenário de implementação 2 60
Figura 37 – Instalação do cliente OpenVPN para Windows 71
Figura 38 – Ícone do OpenVPN para Windows 71

LISTA DE ABREVIATURAS

3DES	3 Data Encryption Standard
AES	Advanced Encryption Standard
AH	Authentication Header
ANSI	American National Standards Institute
ATM	Asynchronous Transfer Mode
CHAP	Challenge Handshake Authentication Protocol
DES	Data Encryption Standard
DMZ	Demilitarized Zone
DoS	Denial of Service
ECC	Elliptic Curve Cryptography
ESP	Encapsulation Secure Payload
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange Protocol
IP	Internet Protocol
IPS	Intrusion-prevention system
IPSec	IP Secure
Ipv4	IP Versão 4
Ipv6	IP Versão 6
IPX	Internetwork Packet Exchange
ISAKMP	Internet Security Association and Key Management Protocol
ISDN	Integrated Services Digital Network
L2F	Layer Two Forwarding
L2TP	Layer Two Tunneling Protocol
LAN	Local Area Network
LZO	Lempel-Ziv-Oberhumer

MAC	Message Authentication Code
MD5	Message Digest 5
MS-CHAP	Microsoft Challenge Handshake Authentication Protocol
NAS	Network Access Server
NetBEUI	NetBIOS Extended User Interface
NIST	National Institute of Standards and Technology dos E. U.
OSI	Open Systems Interconnection
PAP	Password Authentication Protocol
PFS	Perfect Forward Secrecy
PoP3	Post Office Protocol versão 3
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
RC4	Rivest Cipher 4
RSA	Ron Rivest, Adi Shamir e Len Adleman.
SA	Security Association
SAD	Security Association Database
SHA-1	Secure Hash Algorithm 1
SHA-2	Secure Hash Algorithm 2
SLIP	Serial Line IP
SMLI	Stateful Multi-Layer Inspection
SMTP	Simple Mail Transfer Protocol
SPD	Security Police Database
SPI	Security Index Parameter
SSL	Security Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network

Sumário

Sumário 9
Capítulo 1 11
Introdução 11
Capítulo 2 12
Segurança 12
2.1 - Segurança da Informação 13
2.2.1 - Confidencialidade 13
2.2.2 - Integridade 13
2.2.3 - Autenticidade 14
2.2.4 - Disponibilidade 14
2.3 - Criptografia 14
2.3.1 Criptografia simétrica 15
2.3.2 Criptografia assimétrica 18
2.4 Message Digest (Hash) 20
MD5 (Message Digest 5) 21
SHA-1 (Secure Hash Algorithm) 21
MAC (Message Authentication Code) 22
2.5 Assinatura Digital 22
2.6 Formas de Ataques 23
2.6.1 Criptoanálise 23
2.6.2 Força Bruta (Brutal Force Attack) 24
2.6.3 Homem Intermediário (Men-In-The-Middle Attack) 24
2.6.4 Reenvio (Replay Attack) 25
Capítulo 3 26
Firewall 26
3.1 O que um firewall pode fazer 27
3.2 O que um firewall não pode fazer 27
3.3 Arquitetura de Firewall 28
3.3.1 Filtro de pacotes 28
3.3.2 Servidor Proxy 30
3.3.4 Firewall Stateful 31
3.4 Topologias 32
3.4.1 Dual Home-host 32
3.4.2 Screened host 33
3.4.3 Screened Subnet 34
Capítulo 4 37
VPN – Virtual Private Network 37
4.1 Elementos de uma conexão VPN 38
4.2 Topologias 39
4.2.1 Host-host 39
4.2.2 Host-gateway 39
4.2.3 Gateway-gateway 40
4.3 Protocolos 40
4.3.1 - PPTP (Point-to-Point Tunneling Protocol) 40

4.3.2 - L2F (Layer Two Forwarding) *45*
4.3.3 - L2TP (Layer Two Tunneling Protocol) *46*
4.3.4 - IPSec (IP Security) *49*
4.3.5 - SSL (Secure Socket Layer) *55*
Capitulo 5 ..... **59**
Cenário de Implementação **59**
Cenário 1: gateway-gateway 59
Cenário 2: host-gateway 60
5.1 OpenVPN 61
5.2 Instalação 62
5.3 Softwares desenvolvidos 74
Capítulo 6 ..... **75**
Conclusão **75**

Capítulo 1

Introdução

Inicialmente quando surgiram as redes de computadores não havia uma preocupação tão grande quanto à segurança como hoje, não dava para imaginar, naquela época, que nos dias de hoje praticamente tudo funcionaria com o auxilio de máquinas. E em um ambiente corporativo elas são simplesmente indispensáveis.

Com o surgimento das redes de computadores e os avanços tecnológicos, alcançamos diversas formas de conectarmos nossas máquinas, as redes de computadores nos permitem uma grande redução de custos com infra-estrutura e equipamentos.

Atualmente é muito comum termos a necessidade de conectar várias redes próximas ou não. Como solução temos os canais dedicados e as VPNs, cada um com seus prós e contras, o que mais motiva a instalação de uma VPN é o seu baixo custo, se comparada com uma solução de canal dedicado. Uma VPN permite que redes se comuniquem de forma segura através de um meio público, geralmente é utilizada a Internet. Esta tecnologia cria túneis virtuais e transmite os dados criptografados através deles.

O objetivo deste trabalho é apresentar as principais características de uma VPN assim como os principais protocolos disponíveis, ao seu final será descrita uma implementação em Linux utilizando o SSL (Secure Socket Layer).

Este trabalho está divido em 6 capítulos, onde o capítulo 1 possui uma introdução, o capítulo 2 apresenta conceitos de segurança e algumas formas de ataque, o capítulo 3 descreve sobre firewall e as principais arquiteturas existentes, o capítulo 4 apresenta os conceitos de VPN e os protocolos mais utilizados, o capítulo 5 descreve a implementação de uma VPN em Linux utilizando o OpenVPN que cria túneis SSL e o capítulo 6 uma avaliação sobre o trabalho.

Capítulo 2

Segurança

Diariamente novas formas de se atacar um usuário são descobertas, a criatividade dos atacantes não têm limites, são e-mails falsos sobre informações de contas bancárias, promoções com links suspeitos, notificações de que seu CPF está irregular etc. Na maioria desses casos o atacante depende da ingenuidade do usuário para poder ter acesso às informações confidenciais, esses tipos de ataques são chamados de ataques de engenharia social, onde o atacante faz com que o usuário passe todas as informações sem se negar. Mas, existem outras formas de ataque que um usuário pode sofrer, o atacante pode tomar conhecimento de informações importantes de um usuário sem utilizar técnicas de engenharia social, desta forma o atacante invade o "ambiente" onde o usuário está e então com acesso ao seu micro rouba as informações que deseja.

Existem duas maneiras de um atacante atuar, de forma passiva, ou seja, ele apenas "escuta" o meio e busca informações ou de forma ativa onde o atacante além de capturar as mensagens ele também altera seu conteúdo retransmitindo ou até mesmo gerando novas mensagens assumindo assim a identidade do usuário. Veja a figura 1.

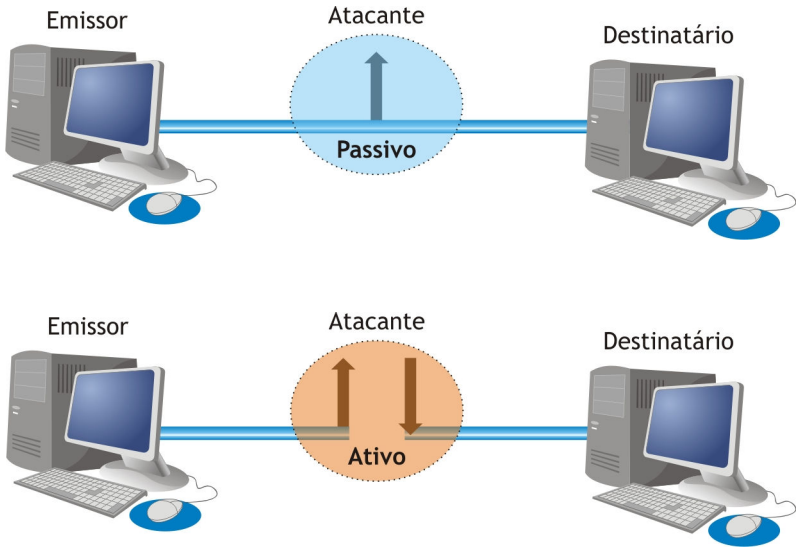

Figura 1 – Atacante Passivo e Atacante Ativo

2.1 - Segurança da Informação

A segurança dos dados é certamente a principal preocupação de um administrador e para chegar a uma solução eficaz é necessário fazer uma análise detalhada do ambiente que deve ser protegido.

Como as empresas estão cada vez mais dependentes das redes de computadores para realizar suas atividades, a preocupação com a segurança dos dados cresce a cada dia.

Quando falamos de segurança da informação presumimos que três premissas devem ser atendidas: *Confidencialidade, Integridade e Autenticidade*. Em alguns casos, como serviços pela rede, também temos que garantir a *Disponibilidade.*

2.2.1 - Confidencialidade

Como as informações trafegam em um meio público é muito simples capturar um pacote e ter acesso ao seu conteúdo, para assegurar-se de que o conteúdo de um pacote não será "legível" para um intruso, mas somente para o real destinatário, utilizamos técnicas de criptografia. Onde a informação original é criptografada em sua origem e enviada para o destinatário, este por sua vez, recebe o pacote criptografado, descriptografa-o e então tem acesso ao conteúdo do pacote.

O elemento-chave na segurança de uma criptografia é o tamanho da chave utilizada, uma vez que a maioria dos algoritmos é de domínio público.

2.2.2 - Integridade

Integridade é garantir que a mesma informação que foi enviada pelo remetente chegará idêntica ao destinatário, inibindo assim qualquer tentativa de se alterar o conteúdo das mensagens. Esta premissa é garantida através de funções de hash que são calculadas no remetente e enviadas junto com a mensagem, o destinatário executa a função hash

novamente na mensagem recebida e compara com o resultado envido. Se os dois valores de hash forem iguais não houve alteração de conteúdo.

2.2.3 - Autenticidade

É a garantia de que a mensagem realmente foi enviada pelo emissor esperado. Assim o destinatário tem a certeza do remetente.

2.2.4 - Disponibilidade

É fazer com que um determinado serviço esteja disponível quando for solicitado. De nada adianta ter um serviço que não esteja disponível quando for solicitado.

2.3 - Criptografia

A criptografia existe há muito tempo e consiste na arte ou ciência de codificar uma mensagem para que somente pessoas autorizadas possam ter acesso ao conteúdo. Com o passar dos tempos e o desenvolvimento da matemática e da informática muitas outras formas de criptografia foram desenvolvidas.

A palavra criptografia vem do grego (Kryptos = oculto e Grafia = escrita) [7]. A criptografia hoje está cada vez mais no cotidiano. Para qualquer transação bancária através da Internet um bom nível de segurança é considerado pré-requisito, pois estamos transmitindo dados valiosos através de um meio inseguro. Um exemplo do processo de criptografia pode ser descrito através da figura 2.

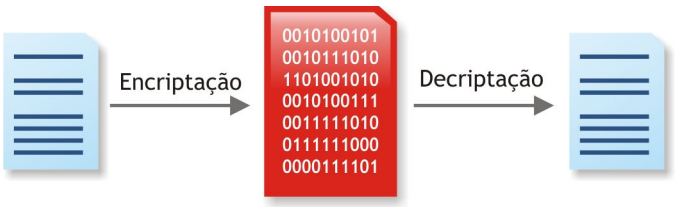

Figura 2 – Processo de criptografia

A segurança da criptografia está relacionada à chave utilizada, esta chave consiste em uma seqüência de bits e quando maior ela for maior poderá ser a segurança obtida.

Dependendo do tipo de chave utilizada a criptografia pode se classificar em simétrica e assimétrica.

2.3.1 Criptografia simétrica

Este método de criptografia consiste na utilização de uma única chave para cifrar e decifrar as mensagens e geralmente são algoritmos bem rápidos. Esta chave é trocada ou combinada entre os extremos da conexão antes da transmissão dos dados. A figura 3 demonstra este processo.

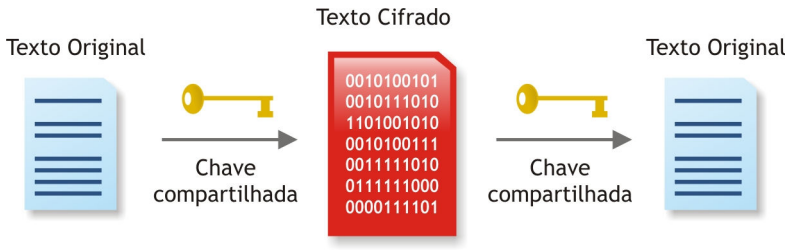

Figura 3 – Criptografia simétrica

A mensagem é cifrada com a chave compartilhada entre as duas partes e então é enviada. O receptor faz o processo inverso: pega a mensagem cifrada e a decifra com a chave compartilhada e assim tem acesso à mensagem original.

Uma desvantagem deste processo e que quanto mais usuários participarem da transmissão maior será o número de chaves que todos deverão armazenar.

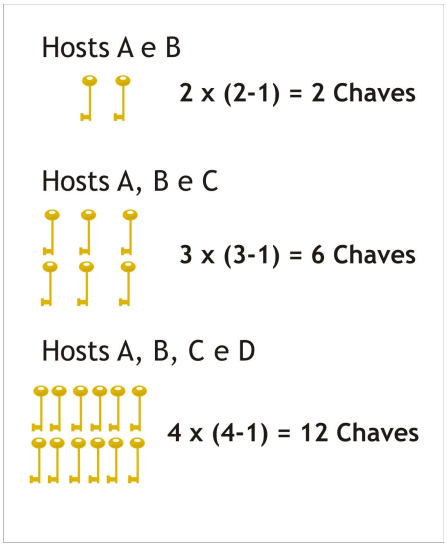

Figura 4 – Criptografia simétrica / número de chaves

A figura 4 mostra que quanto mais usuários existirem na comunicação maior será o número de chaves necessárias para que todos possam se comunicar de forma segura. Para cada par de hosts transmitindo é necessário uma chave distinta. Isso faz com que um grupo com 4 host necessitem de 4x(4-1) = 12 chaves. Como a segurança da transmissão depende da segurança da chave quanto mais chaves existirem maior é o risco de problemas.

Os algoritmos de chave simétrica podem ser divididos em duas categorias: de bloco e de fluxo:

Algoritmos de Bloco – Cifram os dados a partir de blocos, ou seja, se o dado a ser cifrado é um texto, esse texto será dividido em blocos e a criptografia será aplicada em cima de cada bloco. Um problema com estes algoritmos é que se o mesmo bloco de texto simples aparecer em dois lugares, ele encriptará o mesmo texto, gerando assim, um padrão de repetição.

Algoritmos de Fluxo – Cifram os dados byte a byte. O dado a ser criptografado não é cifrado por blocos, como o anterior e sim, serialmente. A informação vai sendo criptografada do início ao fim, sem separações.

Algoritmos de criptografia simétrica:

DES – o Padrão para Criptografia de Dados (*Data Encryption Standard*) foi adotado como padrão pelo governo dos EUA em 1977, e como padrão ANSI em 1981. O DES é um algoritmo de bloco e tem diferentes modos de operação, dependendo da finalidade com que é usado. Foi o algoritmo mais utilizado internacionalmente, sendo o primeiro cujo conhecimento se tornou público. Possui uma entrada de 64 bits, chaves de 56 bits, e saída de 64 bits; utiliza a mesma chave para descriptografar. Utilizado em aplicações típicas que necessitam de bom desempenho. Sua fraqueza é o pequeno tamanho da chave (56 bits), facilitando o ataque por força bruta.

3DES – Foi uma maneira de tornar o DES pelo menos duas vezes mais seguro, usando o algoritmo de criptografia três vezes, com três chaves diferentes. Atualmente, o AES substituiu o Triple-DES por ser muito mais rápido e seguro.

RC4 – Inventado em 1987 pela RSA, nunca teve o seu algoritmo de funcionamento interno publicado. Esse segredo possuía interesses financeiros e não de segurança. A empresa esperava que o mantendo em segredo, ninguém mais o implementaria e o comercializaria. É uma cifragem muito utilizada hoje em dia, até fazendo parte no protocolo de comunicação SSL (*Security Socket Layer*). Embora o tamanho da sua chave possa variar, ela é normalmente utilizada com 128 bits. A cifra foi desenvolvida por Ron Rivest, e o fornecedor garante que o algoritmo é imune a algumas técnicas de criptoanálise (quebrar um texto ilegível ou quebrar a chave) como a diferencial (método probabilístico para ataque do tipo texto legível conhecido - o analista conhece o texto claro, o texto criptografado e o algoritmo, mas não conhece a chave). O problema mais comum é o uso de chaves com comprimento pequeno.

AES – O AES surgiu como uma proposta para substituir o DES, para isso foi realizado um concurso, pelo NIST (National Institute of Standards and Technology dos E. U.), onde os algoritmos deveriam atender as seguintes características: Algoritmo publicamente definido; ser uma cifra simétrica de bloco; permitir que o tamanho da chave possa aumentar; Implementável tanto em hardware quanto em software e

disponibilizado livremente ou em acordo com termos ANSI. E após três anos o algoritmo Rijndael escrito por Vincent Rijmen e Joan Daemen foi o vencedor. Baseado em uma rede de permutação em blocos de 128, 192 e 256 bits e chaves de 128, 192 e 256 bits. O algoritmo é baseado em um trabalho anterior de Rijmen e Daemen chamado Square, que por sua vez é derivado do algoritmo Shark também de ambos.

2.3.2 Criptografia assimétrica

Também conhecida como criptografia de chave pública ela consiste na utilização de duas chaves, uma pública e outra privada, desta forma um usuário que pretenda transmitir seus dados, deverá manter sua chave privada em um local muito seguro, pois esta chave é que vai garantir a autenticidade das mensagens enviadas pelo usuário e possibilitar a leitura das mensagens recebidas.

Já a chave pública, não necessita de tanta segurança, pois ela será de conhecimento de todos. Como as chaves são complementares se a mensagem foi cifrada por uma das chaves somente a outra chave poderá decifrá-la.

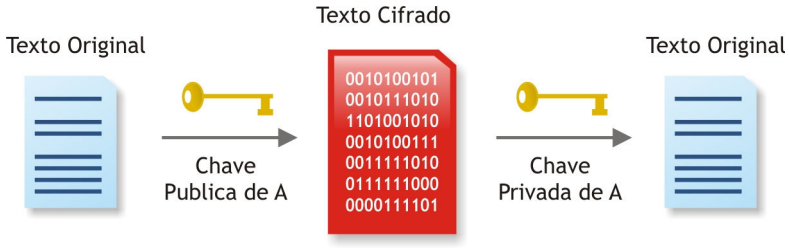

Figura 5 – Criptografia assimétrica

Vamos imaginar a seguinte situação: Inicialmente queremos mandar uma mensagem para o usuário A, então obtemos a chave pública do usuário A e ciframos a mensagem a ser enviada e após este processo a enviamos para o destinatário. O destinatário ao receber a mensagem pega sua chave privada e decifra a mensagem recebida. A criptografia assimétrica também é utilizada para realização do processo de assinatura digital como apresentado na sessão 2.5.

Diferente da criptografia simétrica cada usuário participante da comunicação deve ter apenas que se preocupar com a segurança da sua chave privada.

Uma desvantagem é que os métodos assimétricos são mais lentos do que os métodos de criptografia simétrica.

Algoritmos assimétricos

ECC – A tecnologia ECC (Elliptic Curve Cryptography) possui segurança comparável à tecnologia RSA, com a vantagem de trabalhar com chaves bem menores, ideal para aplicações onde o processador, memória e tráfego em rede são recursos críticos. Pode atuar com chaves de 160, 224, 256, 384 ou 512 bits o que são equivalentes às chaves de 1024, 2048, 3072, 7682, 15360 bits para o RSA respectivamente [4]. A segurança de um sistema de criptografia baseado em curvas elípticas baseia-se no problema do logaritmo discreto em tais curvas.

RSA – O RSA (Rivest-Shamir-Adleman), inventado por Ron Rivest, Adi Shamir e Leonard Adleman em 1977, apresenta a vantagem de permitir a utilização de chaves criptográfica de tamanho variável, permitindo a utilização de chaves grandes, favorecendo a segurança, ou pequenas, favorecendo a eficiência. O tamanho do bloco de dados a ser encriptado é também variável, desde que seu tamanho seja menor que o da chave utilizada. O criptograma gerado é sempre do mesmo tamanho que a chave.

Tanto o RSA como o ECC utilizam criptografia em blocos e possuem uma segurança muito forte, devido ao alto poder computacional necessário para se tentar quebrar uma chave. Podem tanto ser usados para cifrar informações como para servir de base para um sistema de assinatura digital.

Algoritmo de troca de chaves:

Diffie-Hellman – Um sistema para troca de chaves criptográficas entre partes. Na verdade, não é um método de cifrar ou decifrar, é um método para troca de chave

secreta compartilhada por meio de um canal de comunicação público. Com efeito, as duas partes estabelecem certos valores numéricos comuns e cada uma delas cria uma chave. As transformações matemáticas das chaves são trocadas. Cada parte calcula então uma terceira chave (a chave de sessão) que não pode ser descoberta facilmente por um atacante que conheça os valores trocados.

Uma característica curiosa deste algoritmo, proposto por W. Diffie e M. E. Hellman em 1976 é o fato de não serem realizadas operações de encriptação e assinatura de mensagens.

Este algoritmo torna possível a determinação de uma chave compartilhada entre duas entidades que desejem trocar dados de forma segura, mas que só possam trocar dados por meio de um canal público. Sua função é, portanto, o estabelecimento de uma chave secretamente através de um canal inseguro. Uma vez que esta tenha sido estabelecida, algum algoritmo de criptografia simétrico ou assimétrico pode ser utilizado para promover a troca segura de dados.

Este algoritmo apresenta a desvantagem de não prover autenticação, tornando possível o estabelecimento de uma chave compartilhada com uma entidade que esteja pretendendo se passar por outra. Esta deficiência torna possível um ataque do tipo men-in-the-middle.

2.4 Message Digest (Hash)

Os termos *hash* e *message digest* são utilizados para identificar funções com as características abaixo:

Mapeiem um valor de entrada em um valor de saída;
Não haja qualquer correlação direta entre os valores de entrada e saída;
Recebam valores de entrada de tamanhos arbitrários e mapeiem-nos em valores de saída de tamanho fixo.

Principais propriedades de um hash:

É desejável que a partir dele não se pode descobrir a mensagem original;
É improvável que duas mensagens que produzam o mesmo resumo (colisão);

Colisões – é quando o algoritmo produz para duas mensagens diferentes o mesmo hash.

MD5 (Message Digest 5)

Criado em 1991 por Ron Rivest, é basicamente um MD4 melhorado. Foi desenvolvido para máquinas de 32 bits e pode ser facilmente implementado, teve seu algoritmo publicado em abril de 1992.

O MD5 recebe uma mensagem de tamanho aleatório e retorna um resumo de 128 bits. Antes de iniciar os cálculos o algoritmo complementa a mensagem, se necessário, para que o tamanho da mensagem mais 64 bits (esses 64 bits correspondem a uma representação binária do tamanho original da mensagem) possam ser divididos por 512. Pois o algoritmo trabalha com blocos de 512 bits, após esta operação e que o MD5 faz uso de um buffer de quatro registradores de 32 bits cada para iniciar os cálculos e produzir um resumo.

SHA-1 (Secure Hash Algorithm)

Este algoritmo foi aprovado pelo governo dos Estados Unidos da América em 1995 para ser utilizado em todos os departamentos e agências federais para autenticação de documentos digitais.

Assim como o MD5 o SHA-1 também surgiu a partir de estudos feitos sobre o MD4, portanto possui uma estrutura semelhante ao MD5, pois trabalha com blocos de 512 bits e faz o mesmo processo de complemento da mensagem original para que possa ser dividida por 512. Após o processo de complemento do tamanho da mensagem o SHA-1

faz uso de dois buffers de cinco words de 32 bits para produzir um resumo de 160 bits ao final de seus cálculos.

MAC (Message Authentication Code)

É um pequeno bloco de dados que é gerado a partir de uma chave compartilhada, este bloco é anexado à mensagem e enviado para a outra extremidade da comunicação, esta extremidade realiza o mesmo cálculo sobre a mensagem utilizando a chave compartilhada e compara os dois resultados MAC. Assim é possível certificar a integridade e autenticidade da mensagem.

2.5 Assinatura Digital

A assinatura digital é a forma de garantir ao receptor da mensagem o seu real emissor. Para que possamos realizar o processo de assinatura digital temos que fazer uso de um sistema de chaves públicas, a figura 6 ilustra este processo.

Figura 6 – Assinatura digital

O emissor de posse da mensagem original, também chamada de texto pleno, inicialmente calcula o hash da mensagem e então o criptografa com sua chave privada, logo em seguida ele envia a mensagem e o hash criptografado para o destinatário.

O destinatário ao receber a mensagem para verificar se o autor é realmente quem diz ser, vai até uma autoridade certificadora e busca a chave pública do remetente, com ela

o destinatário poderá descriptografar o hash, recalculá-lo e então verificar se o remetente é autentico.

Este tipo de procedimento não previne que um invasor tenha conhecimento do conteúdo da mensagem, apenas garante quem é o remetente. Uma boa combinação é além do emissor criptografar o hash com sua chave privada ele cifra toda a mensagem com a chave pública do destinatário, desta forma somente o destinatário com sua chave privada poderá ter acesso ao conteúdo da mensagem.

2.6 Formas de Ataques

2.6.1 Criptoanálise

Criptoanálise é a ciência que abrange os princípios e métodos para se chegar a desvendar um criptograma sem as chaves criptográficas empregadas na sua geração.

Um atacante pode tentar obter acesso a dados cifrados por duas formas: obtendo a chave, ou tentando deduzir a mensagem a partir do criptograma. Ambas fazem uso de métodos de criptoanálise.

Há basicamente três métodos de ataque para se decifrar um criptograma: utilizando apenas o criptograma, conhecendo-se a mensagem, ou escolhendo-se a mensagem.

Em ataques utilizando apenas o criptograma (*ciphertext-only*), o criptoanalista deve determinar a chave a partir de criptogramas interceptados. Em alguns casos o método de criptografia, o idioma, o assunto ou as prováveis palavras utilizadas nas mensagens podem ser conhecidos facilitando o ataque.

Em ataques cuja mensagem é conhecida (*known-plaintext*), o criptoanalista possui acesso a pares mensagem-criptograma e procura reconhecer padrões entre diferentes pares.

Em ataques do tipo mensagem escolhida (*chosen-plaintext*), o criptoanalista possui acesso a algum mecanismo que lhe fornece o criptograma referente a uma mensagem de sua escolha.

Um algoritmo criptográfico é considerado incondicionalmente seguro se não existe análise sistemática capaz de obter acesso a mensagens, independente da quantidade de criptogramas aos quais se tenha acesso e do conhecimento do algoritmo.

2.6.2 Força Bruta (Brutal Force Attack)

Consiste na tentativa de descobrir a chave através de testes com chaves geradas seqüencialmente, aleatoriamente ou seguindo algum padrão. A melhor forma de se evitar este tipo de ataque é utilizar chaves razoavelmente grandes, o que aumentaria o número de possibilidades, por exemplo: com uma chave de 256 bits teremos aproximadamente 10^{77} possibilidades de chaves diferentes, mesmo assim o atacante poderá descobrir a senha nas primeiras tentativas. Havendo tempo disponível este ataque sempre obterá sucesso.

2.6.3 Homem Intermediário (Men-In-The-Middle Attack)

Neste tipo de ataque, também conhecido como Ataque Brigada de Incêndio (*Bucket Brigade Attack*) e ilustrado na Figura 7, duas entidades A e B desejam estabelecer uma comunicação segura por meio de criptografia assimétrica. No entanto, um atacante X, intercepta e altera de forma conveniente as informações trocadas entre A e B, de modo a impedir a comunicação direta entre eles.

Ao interceptar a chave pública de A enviada por A para B, X envia para B sua própria chave pública, pretendendo ser A. O mesmo ocorre para a chave pública de B enviada para A. Deste modo, X consegue estabelecer comunicações seguras com A e B, fazendo estas duas crerem estar se comunicando diretamente.

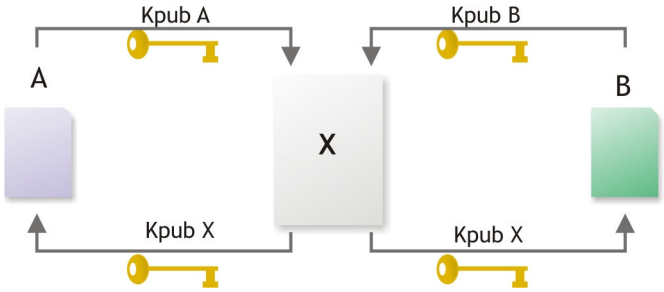

Figura 7 – Men-in-the-middle attack

Para impedir este tipo de ataque é importante que as chaves públicas envolvidas sejam fornecidas por alguma entidade confiável.

2.6.4 Reenvio (Replay Attack)

Este tipo de ataque consiste na retransmissão indevida por parte do atacante de alguma mensagem anteriormente interceptada. Como exemplo, pode-se imaginar um cliente (A) que deve solicitar ao banco (B) a transferência de certo valor para um empregado (X). X intercepta então a mensagem (encriptada) com a solicitação, e mesmo sem compreender seu conteúdo, mas sabendo que esta lhe favorece, reenvia, possivelmente diversas vezes, a mesma mensagem para B. Algumas formas de se contornar esta vulnerabilidade são:

- A anexação da data e hora de envio, devidamente encriptados, de modo que pedidos mais antigos sejam desconsiderados. Mensagens reenviadas dentro desse espaço de tempo seriam, no entanto, consideradas válidas. É necessário sincronismo entre os relógios internos dos equipamentos envolvidos, o que muitas vezes não é uma tarefa simples;
- A anexação de números de seqüência às mensagens, também encriptados, de modo a rejeitar mensagens com números de seqüência repetidos;
- O desenvolvimento de protocolos de autenticação mais complexos.

Capítulo 3

Firewall

Firewall é um elemento de software ou hardware que tem a função de regular o tráfego de uma rede, permitindo ou não a passagem de determinados pacotes usando como base as regras de segurança que lhe foram estabelecidas.

O firewall hoje é um elemento muito importante na construção de uma rede segura. Quando pensamos em compartilhar dados de nossa rede com outras temos que ter alguns cuidados especiais. Por exemplo: se uma empresa pretende disponibilizar um serviço de consulta a um banco de dados para seus clientes e este banco de dados está na rede local da empresa isso fará com que a rede local fique exposta à rede pública, permitindo assim que um atacante tenha acesso a outras informações da empresa além do serviço disponibilizado. A instalação de um firewall entre a empresa e a Internet ajudaria a barrar estas tentativas de invasão.

Para se implantar um firewall é muito importante ter uma política de segurança eficiente, pois é com base nesta política que o firewall será implementado.

Um firewall funciona como um porteiro da sua rede. É ele quem vai decidir quais pacotes podem entrar na ou não na rede, assim como também é ele quem determina quais pacotes podem ou não sair dela. Ele está associado a um conjunto de regras específicas que podem remeter a ações como bloquear, negar, rejeitar ou aceitar os pacotes. Estes critérios de avaliação utilizam parâmetros como protocolo, endereço IP de origem, endereço IP de destino, número da porta de origem, número da porta de destino etc.

3.1 O que um firewall pode fazer

O firewall é o elemento da rede que está exposto à rede pública, desta forma qualquer ataque deverá primeiro passar por ele, sendo assim ele pode ser considerado um checkpoint, ou seja, o foco para a tomada de decisões referentes à segurança.

Com a implantação de um firewall você acaba obrigando a adoção de uma política de segurança ou a revisão de uma já existente além de poder monitorar de forma mais eficiente todo o tráfego da Internet e paralelamente poderá definir perfis para os usuários e reduzir a exposição da rede na Internet.

3.2 O que um firewall não pode fazer

Um firewall não pode proteger contra conexões que não passem por ele. De nada adianta a implantação de um firewall se os usuários da rede interna podem acessar a internet diretamente, sem passar pelo firewall, através de conexões discadas, por exemplo.

Um firewall não pode proteger contra usuários internos mal intencionados. Nada pode ser feito quando um usuário da rede deseja prejudicar o seu funcionamento, por exemplo, ele pode copiar informações sigilosas através de CDs, disquetes ou pendrives ou até mesmo instalar programas que criem brechas para que um atacante possa agir.

Por mais bem configurado que um firewall seja ele não tem como proteger contra ameaças totalmente novas, mesmo que o administrador bloqueie tudo que não é permitido, um atacante pode explorar uma nova vulnerabilidade de um serviço que até o momento parecia seguro e ter acesso à rede interna.

Um firewall não pode proteger a rede contra vírus, para isso é necessário um software específico de antivírus.

3.3 Arquitetura de Firewall

Antes de apresentar algumas arquiteturas de firewall vamos primeiro demonstras alguns conceitos importantes:

Bastion host

Um bastion host é qualquer máquina da rede configurada para exercer um papel crítico de segurança na rede interna, ele representa a presença pública da rede na Internet. Será o ponto de ligação entre a rede local e a rede pública.

Como o bastion host será o possível alvo dos ataques, ele deve possuir uma estrutura simples que permita uma fácil manutenção.

DMZ

DMZ é a sigla para DeMilitarized Zone ou "Zona Desmilitarizada", também é conhecida como Perimeter Network ou "Rede Perimetral" ela está situada entre a rede confiável (rede local) e a rede pública (geralmente a Internet).

3.3.1 Filtro de pacotes

É o tipo mais simples de firewall, ele consiste em determinar a ação que um pacote receberá baseado em informações como:

- Endereço IP de origem;
- Endereço IP de destino;
- Protocolos TCP, UDP ou ICMP;
- Portas TCP ou UDP de origem;
- Portas TCP ou UDP de destino;
- Tipo de mensagem ICMP;
- Tamanho do pacote.

Este firewall atua na camada de Rede do modelo OSI.

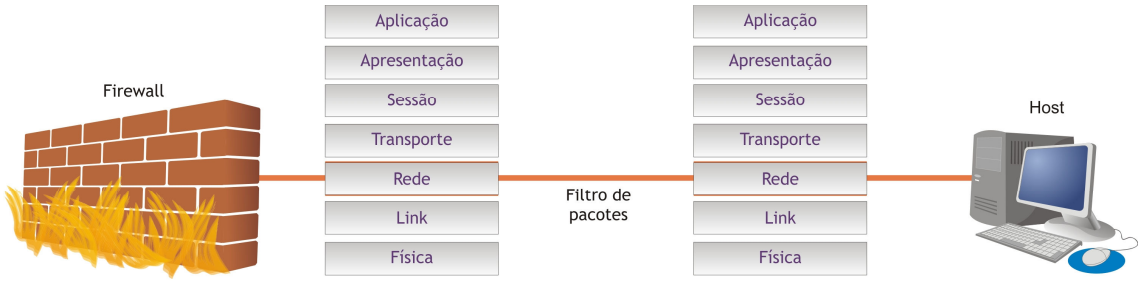

Figura 8 – Filtro de pacotes

Este tipo de filtragem é muito comum ser implementado diretamente em roteadores e são chamados de screening routers, ficam posicionados em pontos de ligação entre redes.

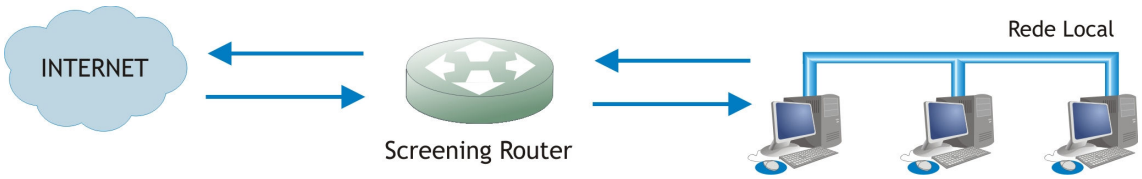

Figura 9 – Screening Router

Vantagens:

- Transparência para os usuários;
- Variedade de software para filtro de pacotes;
- Consome pouco recurso computacional, pois verifica cabeçalhos apenas da camada de rede.

Desvantagens:

- As ferramentas disponíveis não são perfeitas, pois alguns protocolos não são bem adaptados;
- É difícil testar uma política aplicada;

3.3.2 Servidor Proxy

Um servidor Proxy atua como um intermediário entre o cliente da rede interna e o serviço da rede externa. Esta conexão ocorre da seguinte forma:

O cliente da rede interna faz a solicitação, esta solicitação chega ao Proxy, ele a armazena e faz a mesma solicitação ao serviço de Internet, o serviço responde ao Proxy e então ele envia a resposta ao cliente da rede interna, assim o processo se torna totalmente transparente para o cliente e para o serviço da internet, pois ambos acreditam estar se comunicando diretamente.

Tendo posse deste controle o servidor Proxy pode restringir qualquer requisição de um serviço externo por um cliente interno. A figura 10 ilustra o funcionamento de um serviço de Proxy.

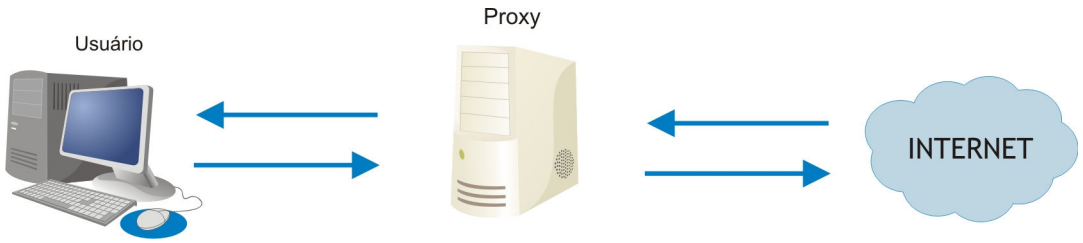

Figura 10 – Servidor Proxy

Existem também servidores Proxy que visam o melhor desempenho da rede, estes são chamados de Proxy de cache. Consiste em armazenar uma cópia das últimas requisições feitas pelos clientes internos desta forma quando houver uma reincidência o servidor Proxy poderá responder diretamente para o cliente interno sem necessitar consultar novamente o serviço solicitado.

Um firewall do tipo Proxy atua na camada de aplicação do modelo OSI, é uma implementação mais segura do que o filtro de pacotes, porém consome mais recursos visto que é necessário que o pacote percorra todas as camadas para então sofrer a ação.

Figura 11 – Proxy (Camada de aplicação)

Vantagens:

- Possuem bons registros de log;
- Pode fornecer cache;
- Pode fornecer autenticação no nível de usuário;
- Permite aos usuários acesso direto a serviços da internet.

Desvantagens:

- É necessário um servidor para cada serviço;
- O desenvolvimento de novas aplicações para servidores proxy não acompanha o desenvolvimento dos serviços da Internet.

3.3.4 Firewall Stateful

Foi desenvolvida inicialmente pela CheckPoint, desenvolvedora do Firewall-1, e posteriormente foi implementada em sistemas de código aberto como Ipf & ipfilter. Este tipo de firewall é considerado um dos mais seguros, pois une as características dos outros dois tipos apresentados: filtro de pacote e Proxy.

Pertence a terceira geração de firewalls, pois suportam a tecnologia SMLI - Stateful Multi-Layer Inspection, ou seja: Inspeção Total de todas as Camadas do modelo OSI. Esta tecnologia permite que o firewall examine cada pacote em todas suas camadas do modelo OSI, desde a de transporte até a de aplicação, sem necessidade de processar a mensagem.

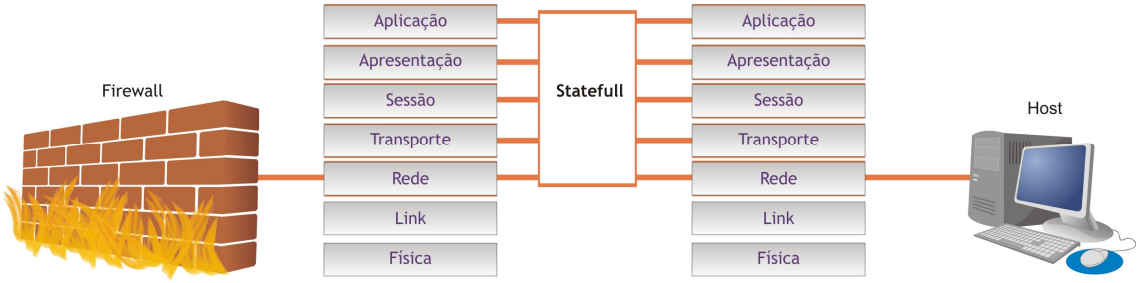

Figura 12 – Firewall Stateful

Esta tecnologia utiliza um método de comparação dos pacotes e padrões pré-definidos de pacotes amigáveis, isso junto com algoritmos de verificação de tráfego otimizados pode oferecer a velocidade de um firewall de filtro de pacotes e a mesma segurança de um firewall do tipo proxy.

Vantagens:

- É totalmente transparente aos usuários;
- É mais rápido que o firewall proxy;
- Oferece recursos de filtragem de pacotes.

Desvantagens:

- Por ser uma tecnologia mais nova não está disponível para qualquer aplicação.

3.4 Topologias

3.4.1 Dual Home-host

Esta arquitetura é formada por um computador com duas interfaces de rede, uma para rede segura (rede local) e outra para a rede pública (internet). Este computador vai isolar as duas redes, onde apenas os serviços do firewall têm acesso a ambas as redes, ninguém da rede segura consegue acessar a rede pública e ninguém da rede pública consegue acessar alguém da rede segura. Para que isso ocorra uma conexão deverá chegar ao firewall e ele fará a conexão com a máquina desejada aplicando as regas de filtragem aos pacotes.

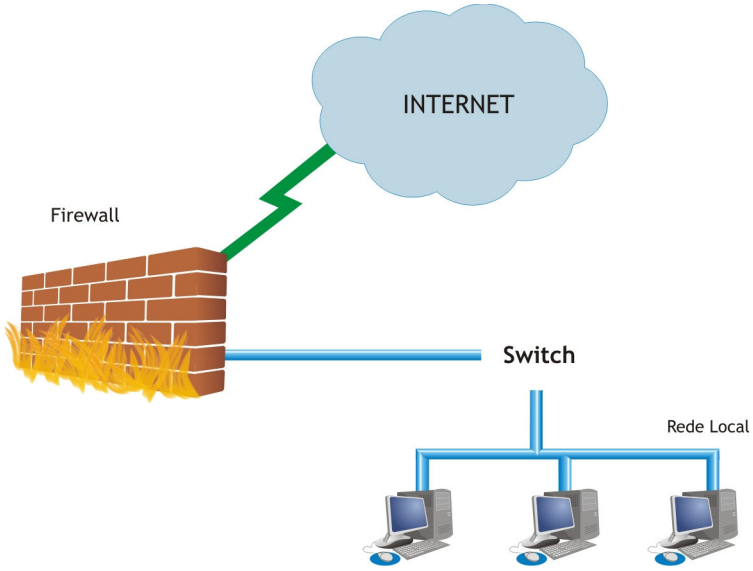

Figura 13 – Arquitetura dual home-host

Esta arquitetura é indicada para um sistema com as seguintes características:

- Pequeno tráfego para Internet;
- Tráfego pela internet não é crítico;
- Não provê serviços a usuários externos;
- Os dados da rede interna não possuem muito valor.

3.4.2 Screened host

Nesta arquitetura os serviços são providos a partir de um bastion host, que está conectado apenas a rede interna, o que separa a rede interna da rede pública é um roteador do tipo screening router, que implementa o filtro de pacotes, fazendo com que todo tráfego externo que passa pelo roteador seja direcionado para o bastion host, assim os serviços externos só conseguem conexão com o bastion host e não com a rede interna.

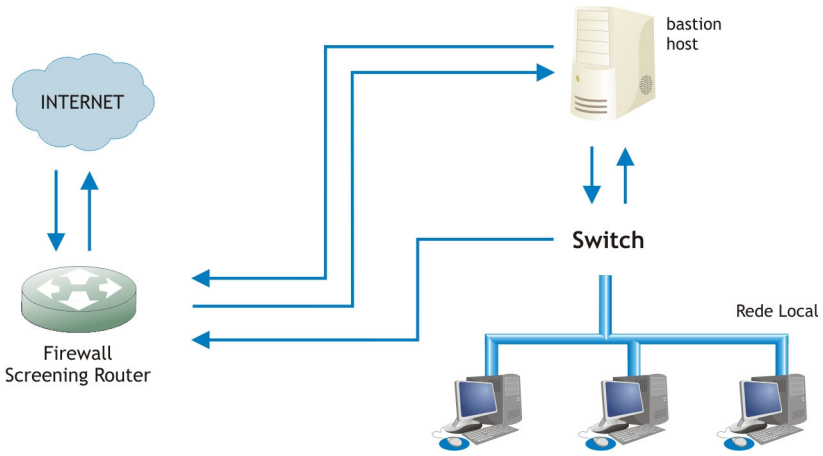

Figura 14 – Screened host

É uma arquitetura fácil de implementar e de dar manutenção e o acesso a serviços proxy é rápido pois o servidor está na mesma rede.

Uma desvantagem é que há apenas uma barreira que protege a rede interna, o roteador, se este for comprometido toda segurança da rede também será.

Esta arquitetura é apropriada para sistemas com as características:

- Poucas conexões chegam pela Internet;
- A rede interna deve ser relativamente bem protegida;
- Não é recomendado quando o bastion host é um web server.

3.4.3 Screened Subnet

Esta arquitetura conta com a presença de uma DMZ ou Rede perimetral, que fica situada entre a rede interna e a rede pública. É considerada uma das mais seguras implementações de firewall, pois conta com três níveis de segurança. A figura 15 ilustra esta segurança.

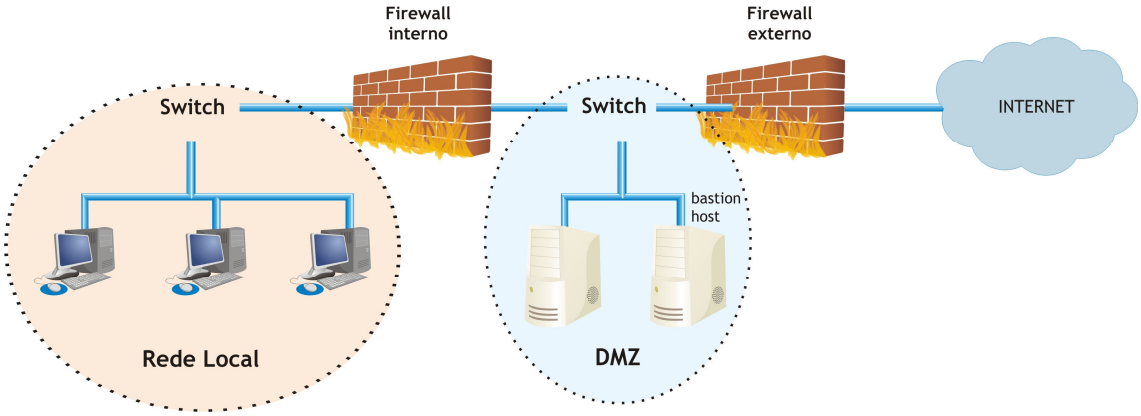

Figura 15 – Arquitetura Screened Subnet

O Roteador Externo (1º nível)

Este equipamento ira permitir a entrada somente de requisições para serviços disponíveis na DMZ e conseqüentemente terá uma filtragem menos rigorosa do que o roteador interno. As regras para proteção da rede interna devem ser praticamente as mesmas em ambos os roteadores.

DMZ (2º nível)

Nesta rede estarão situados os servidores que irão prover serviços à rede externa e o bastion host para permitir a interconectividade da rede interna com a DMZ

O Roteador Interno (3º nível)

Este roteador executa a maior parte da filtragem dos dados, pois ele só irá permitir a saída de requisições para o bastion host e a entrada de pacotes originados nele, isolando a rede interna e tornando o bastion host o único ponto de acesso à rede.

Imaginando que um atacante fosse invadir esta arquitetura ele primeiro teria que passar pelo roteador externo e mesmo que conseguisse, ficaria preso em algumas das máquinas da DMZ para poder chegar à rede local teria ainda que passar pelo bastion host e depois

pelo outro roteador. É altamente recomendado que duas equipes diferentes configurem os roteadores uma o roteador externo e outra o roteador interno, pois possivelmente se for encontrada uma falha em um deles o outro também poderá estar comprometido.

Capítulo 4

VPN – Virtual Private Network

As VPNs surgiram como alternativa para se conectar redes fisicamente distantes através de um meio público não seguro. Vamos supor que uma empresa abrisse uma filial em outro estado, para que a filial compartilhasse os mesmos recursos da rede local da matriz seria necessário fazer a contratação de um canal dedicado junto a uma operadora de telecomunicações, este custo é muito elevado o que acaba inviabilizando esta solução. Como alternativa para este problema temos as VPNs que possuem um custo muito inferior ao de um canal dedicado, esta diferença de valores pode ser considerado o principal fator de desenvolvimento das VPNs.

A sigla VPN quer dizer Virtual Private Network, ou seja, Redes Privadas Virtuais, onde o termo redes corresponde às redes de computadores, Privada corresponde à forma como os dados trafegam, neste caso os dados são criptografados o que garante a privacidade das informações, e o termo Virtual por não fazerem, necessariamente, parte do mesmo meio físico.

Ao criar uma conexão VPN estamos criando um túnel entre as extremidades da conexão assim os dados trafegam seguros de uma ponta até a outra.

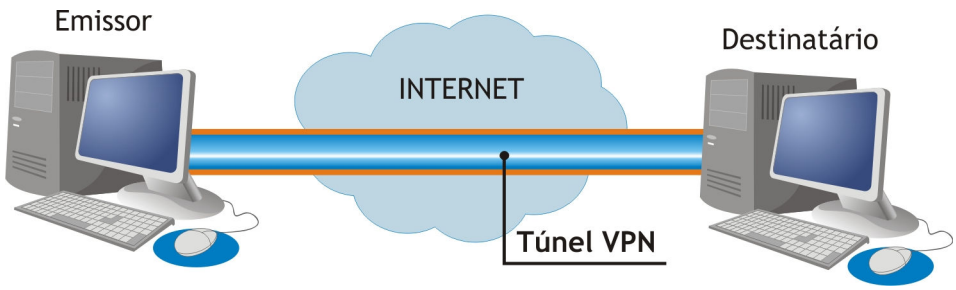

Figura 16 – Túnel VPN

A VPN deve dispor de recursos para permitir o acesso de clientes remotos autorizados aos recursos da LAN corporativa, viabilizar a interconexão de LAN's de forma a possibilitar acesso de filiais, compartilhamento de recursos e informações, e finalmente, assegurar privacidade e integridade dos dados que trafegam pela Internet.

4.1 Elementos de uma conexão VPN

Os principais elementos de uma conexão VPN são: tunelamento, autenticação das extremidades e o transporte subjacente [8].

Tunelamento – O tunelamento se dá pela forma como os dados trafegam pela conexão VPN. A idéia do túnel surge quando para enviar os dados uma das extremidades da conexão, primeiro criptografa e depois encapsula o pacote original dentro de um novo pacote.

Autenticação das Extremidades – Ao utilizar a autenticação das extremidades em uma conexão VPN garantimos que somente usuários válidos estão participando da transmissão, através de protocolos de autenticação, que em sua maioria implementam algoritmos de hash. O que garante a integridade das mensagens.

Transporte Subjacente – Devido ao protocolo TCP/IP ser a base da Internet, ele é amplamente utilizado para a comunicação entre redes. Entretanto, ele é muito inseguro. Por isso, uma VPN utiliza a infra-estrutura da rede já existente do TCP/IP para transmitir os seus pacotes pela Internet adicionando alguns cabeçalhos, o que possibilita a instalação destes em qualquer parte da rede [1].

Figura 17 – Transporte subjacente

4.2 Topologias

4.2.1 Host-host

Comunicação entre dois hosts com acesso a internet separados fisicamente podendo estar ou não em uma rede. A finalidade para este tipo de conexão seria para sincronismo de dados.

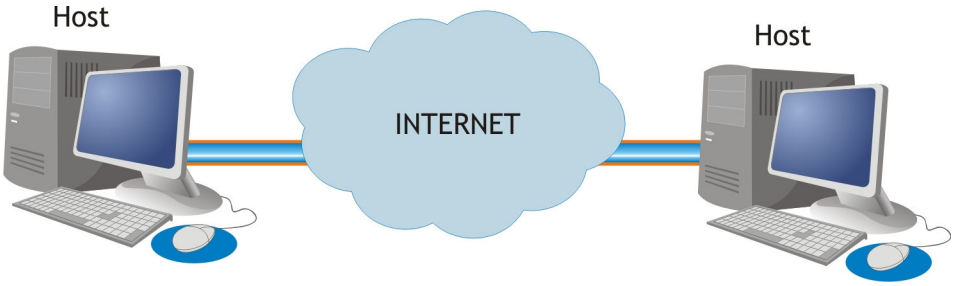

Figura 18 – Topologia Host-host

4.2.2 Host-gateway

Conexão de um host a uma rede. É uma conexão típica de empresas que possuem vendedores fora das suas dependências e estes precisam acessar a rede da empresa enquanto estão fora, como por exemplo, ao realizar uma venda ele se conecta a rede da empresa e ao confirmar a venda no sistema da empresa o vendedor já poderia disparar processos para o setor de produção e iniciar o processo de fabricação dos produtos o que reduziria o tempo de entrega.

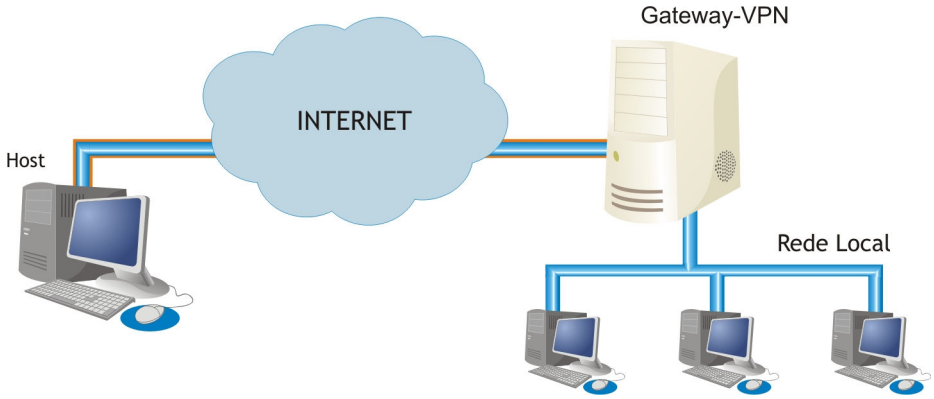

Figura 19 – Topologia Host-gateway

4.2.3 Gateway-gateway

Conexão entre duas empresas onde os gateways de VPN estarão sempre conectados, dessa forma as empresas, geograficamente distantes, podem compartilhar os mesmos recursos de rede com um custo muito reduzido.

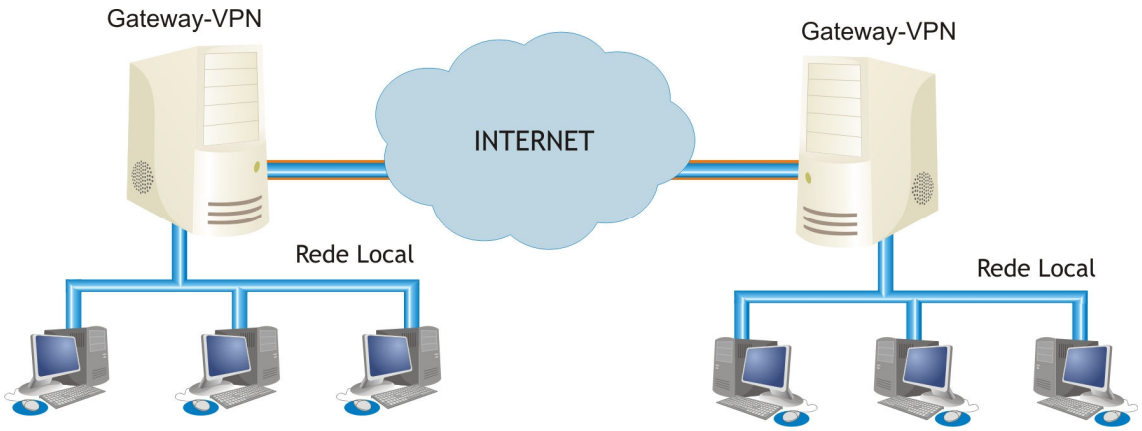

Figura 20 – Gateway-gateway

4.3 Protocolos

Os protocolos são os responsáveis pela abertura e gerenciamento das sessões de túneis VPN.

4.3.1 - PPTP (Point-to-Point Tunneling Protocol)

Desenvolvido por um fórum de empresas denominado PPTP Fórum, entre os membros estavam empresas como a Microsoft, Ascend Communications, ECI Telematics, US Robotics e outras. Esse fórum tinha por objetivo facilitar o acesso de computadores remotos a uma rede privada através da Internet ou outra rede baseada em IP, foi um dos primeiros protocolos de VPN que surgiram.

Está incorporado no Windows a partir do NT 4.0 e um cliente em um patch para Windows 95. Agrega as funcionalidades do PPP para que o acesso remoto seja tunelado para um destino. Ele encapsula pacotes PPP utilizando uma versão modificada do

protocolo de encapsulamento genérico de roteamento (GRE). Permitindo ao PPTP flexibilidade em lidar com outros tipos de protocolos como IPX, NetBEUI etc.

O protocolo se baseia nos mecanismos de autenticação do PPP, os protocolos PAP, CHAP e MS-CHAP.

PAP (Password Authentication Protocol)
É um protocolo de autenticação de texto em formato simples. Os dados são transmitidos sem nenhuma forma de criptografia, ou seja, se o pacote for interceptado o atacante tem acesso a todo o conteúdo do pacote.

CHAP (Challenge Handshake Authentication Protocol)
É um protocolo de autenticação por desafio de resposta. Ele utiliza o MD5 para responder a um desafio emitido pelo servidor de acesso remoto.

É uma melhoria do PAP, pois a senha nunca é enviada na primeira mensagem. A senha é usada para criar uma string hash de desafio. O servidor que sabe a senha duplica a operação e compara os resultados.

MS-CHAP (Microsoft Challenge Handshake Authentication Protocol)
Muito semelhante ao CHAP, sua grande diferença é que no CHAP, a versão de texto simples da senha deve estar disponível para validar a resposta de desafio. No MS-CHAP, o servidor de acesso remoto exige só a string de hash da senha para validar a resposta do desafio.

O PPTP apresenta algumas limitações:

- Não prover forte criptografia;
- Não suportar métodos de autenticação por token.

Funcionamento

Existem três elementos envolvidos em uma conexão PPTP. O cliente, o servidor de acesso à rede e o servidor PPP.

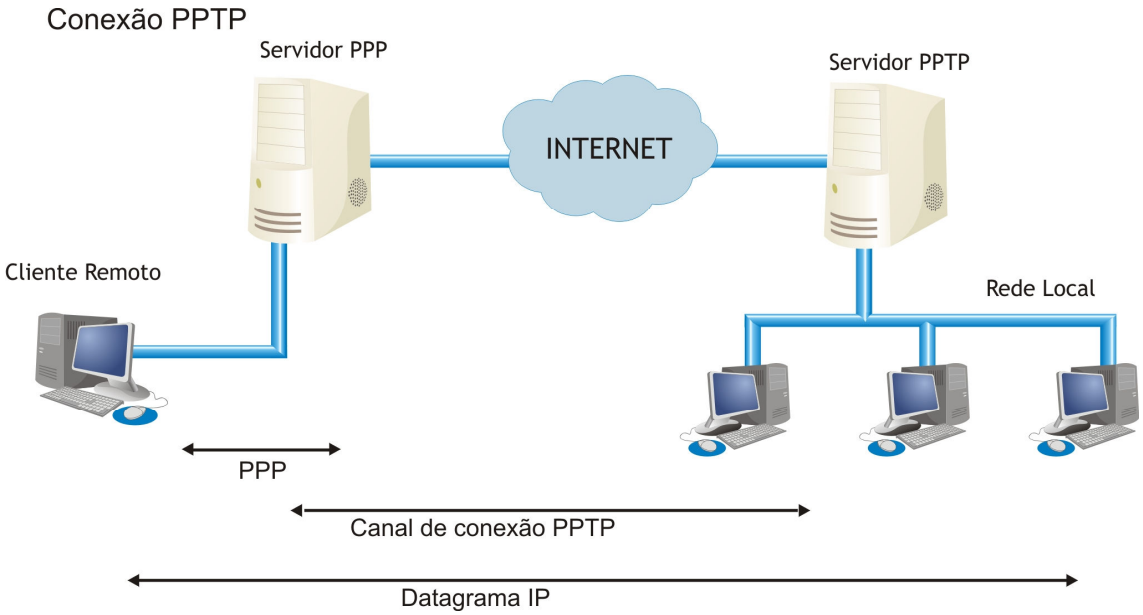

Figura 21 – Conexão PPTP

A comunicação criada pelo PPTP tipicamente envolve três processos, cada um exige que os anteriores sejam satisfeitos.

1. Conexão de comunicação PPP;
2. Conexão de controle PPTP;
3. Tunelamento de dados PPTP.

Conexão de comunicação

O cliente PPTP utiliza o PPP para se conectar ao ISP utilizando uma linha telefônica ou ISDN padrão. Nesta etapa o PPP é utilizado para estabelecer a conexão, criptografar e autenticar os dados.

Conexão de controle PPTP

Utilizando a conexão estabelecida pelo PPP cria-se uma conexão de controle desde o cliente até o servidor PPTP na Internet. É nesta etapa que todos os parâmetros de configuração da conexão são definidos entre as extremidades do túnel. Esta conexão utiliza pacotes TCP e é chamada de Túnel PPTP.

Tunelamento dos dados

Os pacotes de dados são primeiro criptografados e encapsulados com um cabeçalho PPP. O quadro PPP resultante é então encapsulado com um cabeçalho GRE. Este quadro por fim é encapsulado com um cabeçalho IP que contém os endereços de origem e destino correspondentes às extremidades da conexão PPTP.

Figura 22 – Tunelamento PPTP

Encapsulamento de um datagrama IP feito pelo PPTP.

Datagrama PPTP

Existem dois tipos básicos de datagramas PPTP:

Datagramas IP – contém mensagens de dados;

Datagramas TCP – contém mensagens de controle, estas mensagens são responsáveis pelo gerenciamento do túnel.

A figura 23 descreve o processo de comunicação PPTP analisando os pacotes que estão sendo trafegados.

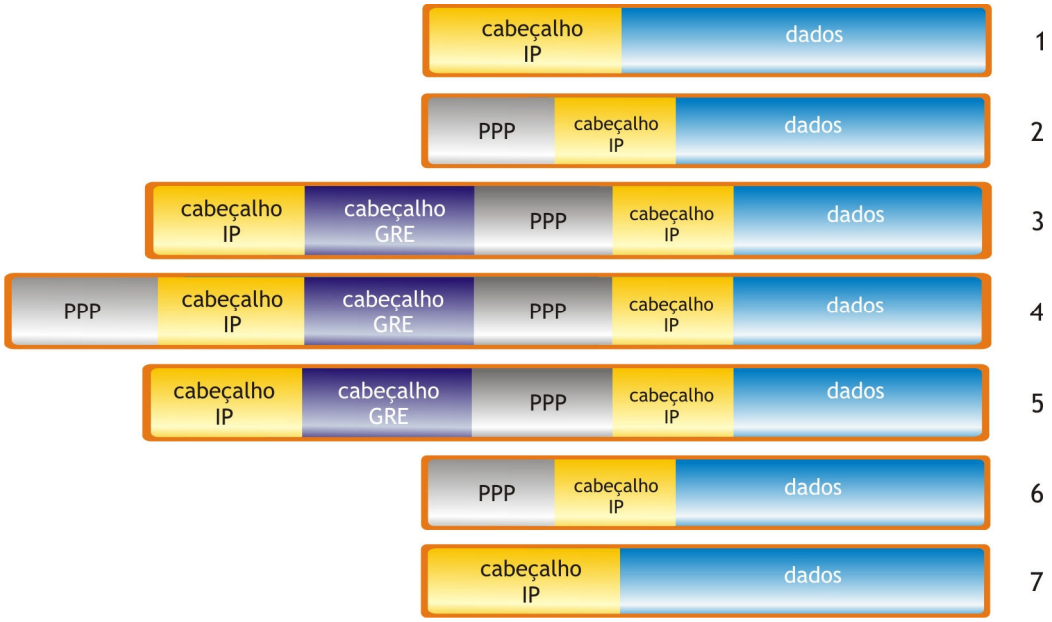

Figura 23 – Processo de comunicação PPTP

1- Datagrama IP original;
2- Datagrama IP dentro de um quando PPP;
3- Quadro PPP encapsulado usando o encapsulamento de roteamento genérico (GRE) com cabeçalho IP;
4- Quadro PPP com todo pacote construído. Esse quadro é enviado para o servidor de acesso a rede;
5- Retirando o cabeçalho PPP, o pacote é enviado através da Internet para o servidor PPTP utilizando o túnel PPTP;
6- Os cabeçalhos IP e GRE são removidos e o servidor PPTP recebe o quadro PPP;
7- O servidor retira o cabeçalho PPP e coloca o datagrama IP dentro da rede interna.

Desvantagens

1. Processo de negociação dos parâmetros de conexão é feita com criptografia muito fraca, permitindo que um atacante modifique os parâmetros ou obtenha alguns dados como endereço IP, nome e versão do software do usuário e em alguns casos até o hash da senha.

2. As mensagens do canal de controle são transmitidas sem qualquer forma de autenticação ou proteção de integridade. O que expõe o canal a ataques de seqüestro de conexão.

3. No período de negociação dos parâmetros de conexão é possível que um atacante inicie vários processos de negociação falsos, visto que não existe autenticação do cliente nesse momento o que poderia acarretar em um ataque de negação de serviço (DoS – Denial of Service).

4.3.2 - L2F (Layer Two Forwarding)

Desenvolvido pela empresa CISCO ele surgiu nos primeiros estágios da tecnologia VPN.

Uma diferença entre o L2F e o PPTP é que o L2F possui tunelamento independente do IP, sendo capaz de trabalhar diretamente com outros meios como ATM e Frame Relay. O L2F sempre assume que a rede privada do cliente estará atrás de um gateway, podendo ser um roteador ou um firewall.

Utiliza o PPP para autenticação de usuários remotos, mas também pode incluir suporte para autenticação via RADIUS, TACACS e TACACS+.

Existem dois níveis de autenticação de usuário: um no IPS antes de estabelecer o túnel e outro quando se estabelece a conexão com o gateway.

Funcionamento

Primeiro o usuário estabelece uma conexão PPP com o servidor de acesso a rede (NAS) do ISP, então o NAS estabelece um túnel L2F com o gateway. Finalmente o gateway autentica o nome do usuário e a senha e estabelece a conexão PPP ou SLIP (Serial Line IP).

A autenticação é feita quando uma sessão VPN-L2F é estabelecida, o cliente, o NAS e o gateway da Internet usam um sistema triplo de autenticação via CHAP.

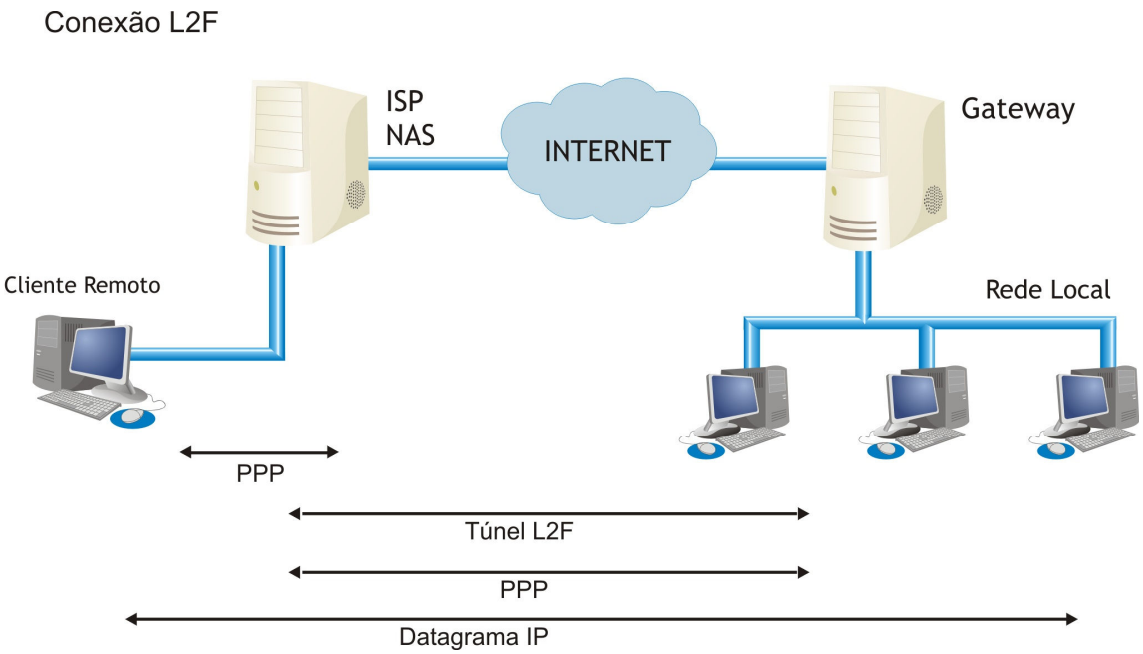

Figura 24 – Conexão L2F

A grande desvantagem do L2F é não definir criptografia e encapsulamento de dados.

4.3.3 - L2TP (Layer Two Tunneling Protocol)

Em uma tentativa de se criar um padrão para protocolos de tunelamento o IETF reuniu neste protocolo as melhores características dos dois protocolos existentes o PPTP e o L2F. Após alguns meses de estudo, concluiu-se que um novo protocolo seria o ideal para um padrão, mas os desenvolvedores do PPTP não aceitaram muito bem essa idéia e continuaram a desenvolvê-lo, enquanto que os desenvolvedores do L2F decidiram parar o projeto e adotar o L2TP.

Ele oferece a flexibilidade e a escalabilidade do IP com a privacidade do Frame Relay ou ATM, permitindo que os serviços de rede sejam enviados nas terminações dos túneis.

O L2TP realiza o encapsulamento de pacotes PPP, podendo então fazer uso dos mecanismos de autenticação PPP. Também provê suporte para autenticação do Túnel, permitindo que ambos os extremos do túnel sejam autenticados

Este protocolo foi desenvolvido para suportar dois modos de tunelamento: o tunelamento voluntário e o compulsório [8].

Voluntário – é iniciado pelo computador remoto, sendo mais flexível para usuários em trânsito que podem discar para qualquer provedor de acesso, pois o provedor não participa da criação dos túneis.

Compulsório – é criado automaticamente e iniciado pelo servidor de acesso à rede sob a conexão discada. Isto necessita que o servidor de acesso à rede seja pré-configurado para saber a terminação de cada túnel baseado nas informações de autenticação de usuário.

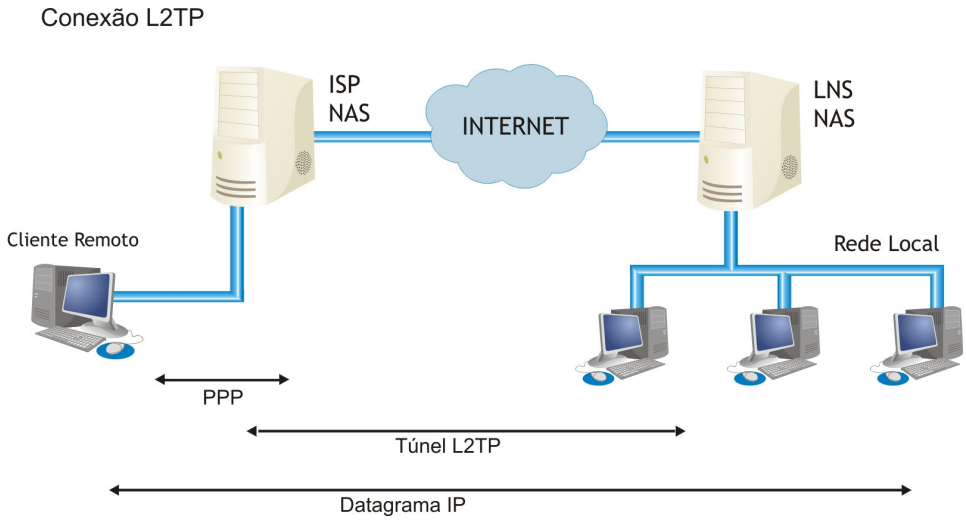

Figura 25 – Conexão L2TP

Funcionamento

Um concentrador de acessos L2TP localizado no Servidor de Acesso a Rede (NAS) troca mensagens PPP com o servidor de rede L2TP para criação dos túneis. O L2TP passa os pacotes através do túnel virtual entre as extremidades da conexão.

Os quadros enviados pelo usuário são aceitos pelo ISP, encapsulados em pacotes L2TP e encaminhados pelo túnel. No gateway de destino os quadros L2TP são desencapsulados e os pacotes originais são processados para interface apropriada.

Datagramas L2TP

Existem dois tipos de datagramas:

- Mensagens de controle – utilizadas para manter, gerenciar e excluir túneis e chamadas;
- Mensagens de dados – são utilizadas para encapsular pacotes PPP a serem transmitidos pelo túnel.

O L2TP suporta tunelamento dos seguintes quadros:

- Frame Relay;
- Ethernet;
- VLAN;
- HDLC (High-Level Data Link Control)
- PPP

Autenticação

Devido ao uso do PPP para links dial-up o L2TP inclui mecanismos de autenticação dentro do PPP, os protocolos PAP e CHAP. Outros sistemas de autenticação também podem ser utilizados como RADIUS e o TACACS. Porém o L2TP não possui processos para gerenciamento de chaves criptográficas.

Diante deste problema, usá-lo em uma rede insegura como a Internet não é recomendado. Para isso deve ser combinado com outros protocolos que supram essas ausências como o IPSec, por exemplo.

Encapsulamento L2TP

Figura 26 – Encapsulamento L2TP

Cabeçalho do L2TP

Quando dados passam por uma interface de entrada de um túnel L2TP, eles são encapsulados por um cabeçalho adicional L2TP, este cabeçalho e composto pelos seguintes parâmetros:

- Cabeçalho de entrega (Delivery header) – é o cabeçalho necessário para transmitir pacotes L2TP através da rede;
- Cabeçalho L2TP – contém informações que permitem identificar de maneira única um túnel no local em que será desencapsulado;
- Informações (payload) – o que será transportado pelo L2TP podendo ser tanto um quadro da camada de enlace quanto um pacote da camada de rede.

4.3.4 - IPSec (IP Security)

Em 1995 como uma resposta as carências de segurança existentes no protocolo IP o Grupo de Trabalho de Segurança IP do IEFT desenvolveu o IPSec, como uma alternativa para a nova geração do Ipv4, o Ipv6 [6].

Este conjunto de protocolos fornece principalmente serviços de integridade, autenticação, controle de acesso e confidencialidade permitindo interoperabilidade com protocolos de camadas superiores como TCP, UDP, ICMP etc.

O IPSec apresenta três características principais [2][6].

- Cabeçalho de Autenticação (Authentication Header) AH – prove a integridade dos pacotes garantindo sua origem;
- Cabeçalho de Encapsulamento do Payload (Encapsulation Header Payload) ESP – garante a confidencialidade dos pacotes;
- Protocolo de negociação e troca de chaves (Internet Key Exchange) IKE – permite a troca de chaves de forma segura.

O IPSec pode trabalhar de dois modos diferentes.

Modo Transporte

É o modo nativo do IPSec, nele há a transmissão direta dos dados protegidos entre os hosts. Toda autenticação e cifragem são realizadas no payload. Utilizado em clientes que implementam o IPSec.

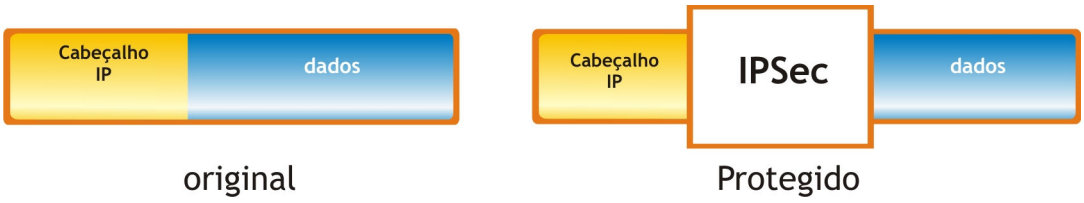

Figura 27 – IPSec modo transporte

Modo Túnel

É mais utilizado por gateways que manipulam tráfego de hosts que não têm suporte ao IPSec. O pacote original é encapsulado em um novo pacote com a criptografia do IPSec incluindo o cabeçalho original, então é enviado para o outro gateway IPSec que desencapsula e o encaminha ao destinatário.

Figura 28 – IPSec modo túnel

Authentication Header (AH)

A utilização do protocolo AH previne ataques do tipo replay, spoofing e hijacking isso porque o protocolo faz uso de mecanismos de autenticação. A figura 29 descreve os campos do protocolo AH.

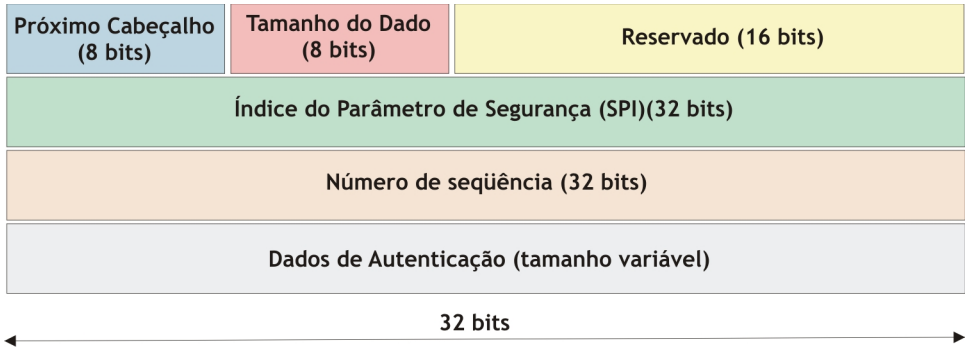

Figura 29 – Campos do cabeçalho AH

Para proteger um pacote o AH insere um cabeçalho AH dentro do pacote a ser protegido, utiliza um número seqüencial que é zerado a cada estabelecimento de uma nova associação segura e adiciona funções de hash (MD5, SHA1 ou SHA2) ao AH [2][6].

Encapsulation Security Payload (ESP)

Além de fornecer as características do AH, este protocolo também oferece a confidencialidade como podemos observar na figura 30. Ele adiciona um cabeçalho ESP logo após o cabeçalho AH (caso este esteja sendo utilizado) e criptografa toda a parte correspondente aos dados (payload) com um algoritmo que foi negociado durante o estabelecimento da SA.

Figura 30 – Pacote ESP

Security Association (SA)

Security Associations ou Associações de Segurança, são muito importantes dentro do IPSec, são elas que contém todas as informações que serão necessárias para configurar as conexões entre as entidades do IPSec.

Elas são criadas durante o processo de negociação dos parâmetros da conexão, uma SA contém informações como algoritmo de criptografia, chaves secretas ou seqüências de números, funções hash, modo de funcionamento (túnel ou transporte), porta de comunicação e outros.

A identificação de uma SA é feita por três parâmetros: endereço IP, protocolo de segurança (AH ou ESP) e por um índice denominado SPI (Security Parameter Index) que é definido na fase de negociação, fase esta que ocorre antes da criação da SA.

Bancos de dados de Segurança

Existem dois bancos de dados utilizados pelo IPSec, o SPD (Security Police Database) e o SAD (Security Association Database). Os SPDs possuem as políticas de segurança, as quais os pacotes irão se submeter, estas políticas são definidas pelo administrador do sistema. Estas políticas serão utilizadas pelas SAs durante o processamento do pacote IP.

Informações armazenadas nos SADs

- Índice dos parâmetros de segurança (SPI);
- Tipo de protocolo AH ou ESP;
- Modo túnel ou modo transporte;
- Seqüencial do pacote IP dentro da SA;
- Número máximo de unidades de transmissão;
- Endereço IP de origem da SA;
- Endereço IP de destino da SA;
- Algoritmo de autenticação e sua chave;
- Algoritmo de criptografia e sua chave;
- Tempo de vida das chaves;
- Tempo de vida da SA.

Os SPDs submetem os pacotes a uma lista de regras e o pacote que atenderá pelo menos uma destas regras sofrerá a ação determinada pelo administrador. Esta ação pode ser: recusar o pacote, aceitar o pacote e aplicar o IPSec sobre ele ou deixá-lo entrar sem aplicar o IPSec.

O SPD trata o tráfego de entrada e de saída independentemente, em que cada interface de rede tem políticas específicas e processa o pacote relativo a cada direção.

Gerenciamento de chaves

O gerenciamento das SAs no IPSec pode ser feito de maneira automática ou de maneira manual, sendo que para redes de grande porte, torna-se inviável gerenciar todas as SAs manualmente. Existem protocolos que implementam o gerenciamento de chaves automaticamente de maneira satisfatória, no entanto o principal deles é o IKE (Internet Key Exchange Protocol) que combina o ISAKMP (Internet Security Association and Key Management Protocol) para definir o método de distribuição de chaves, com o OAKLEY para definir como as chaves serão determinadas.

O protocolo IKE utiliza o ISAKMP para estabelecer um túnel entre as entidades da conexão IPSec. A criação deste túnel é dividida em duas etapas.

1º Etapa: é estabelecida uma associação de segurança do ISAKMP, diferente da SA do IPSec, é neste momento em que as entidades da conexão trocam informações como parâmetros das chaves e um valor randômico que previne ataques do tipo replay. Por fim estas informações são autenticadas por um dos métodos: Segredo compartilhado, Assinatura digital ou troca de chaves públicas.

Segredo compartilhado – a chave é gerada a partir de um desafio que é escolhido entre as entidades. Depois que a chave é criada ela passa por um processo de hash para garantir a autenticidade na hora de usa-la, durante a negociação.

Assinatura digital – É necessário utilizar uma autoridade certificadora para que as entidades possam ter acesso aos certificados digitais, este é o lugar onde estão armazenadas as assinaturas de cada usuário, a partir desta assinatura que a autenticação será baseada.

Criptografia de chaves públicas – as entidades irão trocar as chaves criptográficas.

2º Etapa: também chamada de Modo Rápido, nesta fase se utiliza o canal criado na fase anterior assim os dados já estão protegidos então as entidades negociam a SA do IPSec ao final desta negociação estes dados são gravados nos SADs de cada extremidade.

Existe uma alternativa para aumentar ainda mais a segurança nesta etapa, e o Perfect Forward Secrecy (PFS) esta opção faz com que a chave seja derivada do algoritmo de Diffie-Hellman aumentando a segurança e reduzindo a performance.

4.3.5 - SSL (Secure Socket Layer)

Originalmente o protocolo SSL (Secure Socket Layer) foi desenvolvido pela Netscape Communications para garantir a segurança entre aplicações cliente/servidor evitando influências externas, falsificação dos dados e "escutas". Ao ser padronizado recebeu o nome de TLS (Transport Layer Security) o TLS 1.0 é o mesmo que o SSL 3.0.

Conforme descrito por T. Dierks [3], os objetivos do protocolo SSL em ordem de prioridade são:

1- Garantir o sigilo e a segurança dos dados de uma conexão entre duas partes através do uso de criptografia;
2- Permitir que programadores pudessem desenvolver aplicações utilizando o SSL/TLS independente de sua plataforma de trabalho, garantindo sua interoperabilidade;
3- Prover uma estrutura adequada para incorporar novos métodos de criptografia e chave pública quando necessário, sem a necessidade de criar um novo protocolo ou uma nova biblioteca de segurança interna, garantindo assim a extensibilidade do protocolo;
4- Disponibilização de um esquema opcional para armazenamento temporário de dados das sessões estabelecidas, o que ajuda a diminuir o tráfego entre as partes, logo influencia em um melhor desempenho do protocolo.

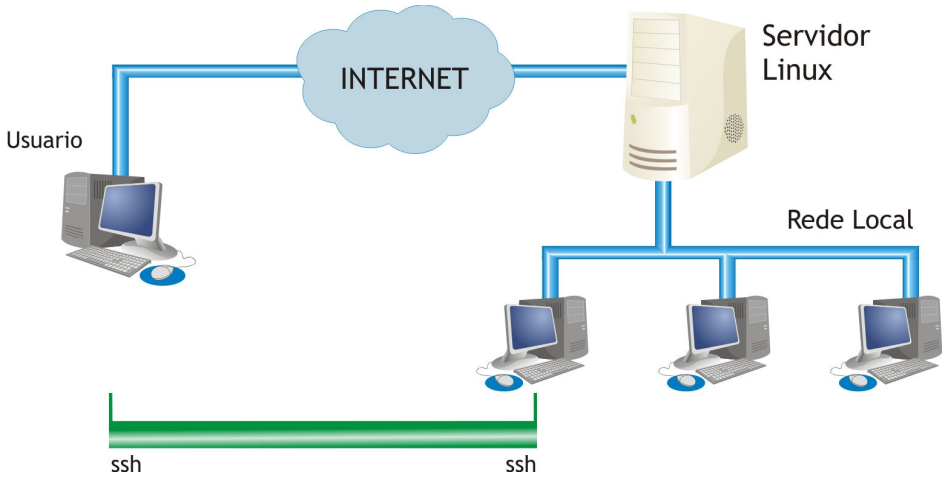

Figura 31 – Conexão SSL

O SSL atua entre as camadas Transporte (TCP) e Aplicação, podendo rodar sobre outros protocolos como o http, Telnet, FTP, SMTP e outros de forma transparente.

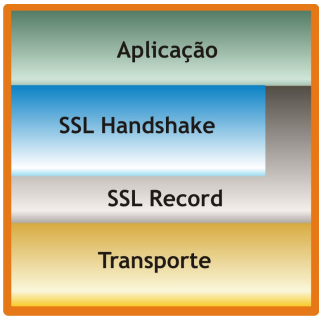

Figura 32 – Camada de Atuação do SSL

O Protocolo SSL é dividido em duas camadas, na camada inferior temos o SSL Record Protocol utilizado para encapsular os protocolos de nível superior além de prover serviços de autenticação, encriptação, fragmentação e compressão da mensagem. Já na camada superior temos alguns protocolos auxiliares que provêem os recursos de segurança.

SSL Handshake protocol

É o protocolo responsável pela autenticação do cliente e do servidor, além de fornecer os parâmetros para o funcionamento do SSL Record Protocol. Todas as mensagens de Handshake são trocadas usando um MAC (Message Authentication Code) para dar mais segurança desde o início do processo.

O protocolo de handshake é constituído de duas fases, numa é feita uma escolha de chave que será utilizada entre o cliente e o servidor, a autenticação do servidor e a troca da chave mestra, já a segunda é feita uma autenticação do cliente, mas esta fase pode não ser requerida. Abaixo a figura 33 demonstra o formato das mensagens deste protocolo:

Figura 33 – Mensagem do protocolo handshake

Handshake type – indica o tipo de mensagem de handshake que será enviada;
Tamanho – indica o tamanho do corpo em bytes
Corpo – os dados da mensagem.

Change Cipher Spec Protocol

Este protocolo é o responsável pela troca dos algoritmos de criptografia utilizados. Ele consiste em uma variável que indica qual o método de criptografia será utilizado. A alteração desta variável pode ser realizada tanto pelo cliente quanto pelo servidor.

Alert Protocol

É o protocolo responsável por supervisionar os erros existentes durante as transações do SSL, a cada erro encontrado o Alert Protocol envia uma mensagem de alerta para a outra extremidade da conexão, caso este número de erros se torne muito elevado, a conexão é interrompida.

SSL Record

A comunicação deste protocolo se dá através do estabelecimento de uma sessão, caracterizado por um Estado de Sessão e um Estado de Conexão, estes estados são criados após o protocolo de handshake concluir suas funções. A figura 25 representa o funcionamento do protocolo

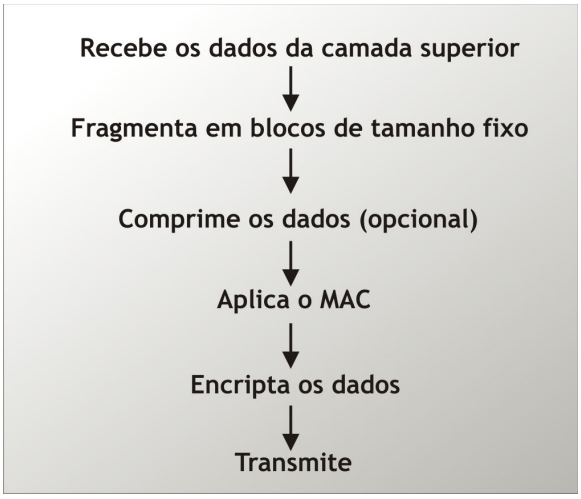

Figura 34 – Funcionamento do protocolo SSL Record

O protocolo recebe os dados da camada superior e os fragmenta em tamanhos fixos para que possam ser melhor "manuseados", posteriormente, dependendo dos parâmetros recebidos da fase de negociação do protocolo de handshake os dados são ou não compactados, em seguida aplica-se um MAC com uma das funções de hash como MD5 ou SHA-1, por exemplo. Agora os dados são encriptados com o algoritmo definido e finalmente transmitidos.

A outra extremidade da conexão executa a operação inversa, junta os fragmentos e entrega a mensagem completa para os protocolos da camada superior.

Vantagens:

- Um dos protocolos mais convenientes e utilizados para implementação de transações seguras;
- Simples implantação;
- Trabalho independente das aplicações utilizadas e, após o handshake inicial, comporta-se como um canal seguro;
- Possui uma padronização do IETF.

Capítulo 5

Cenário de Implementação

Para a demonstração da utilização de uma VPN criamos dois cenários conforme mostrado nas figuras 35 e 36, para a segurança da nossa aplicação vamos utilizar o método de chaves públicas baseado em SSL/TLS + RSA.

Cenário 1: gateway-gateway

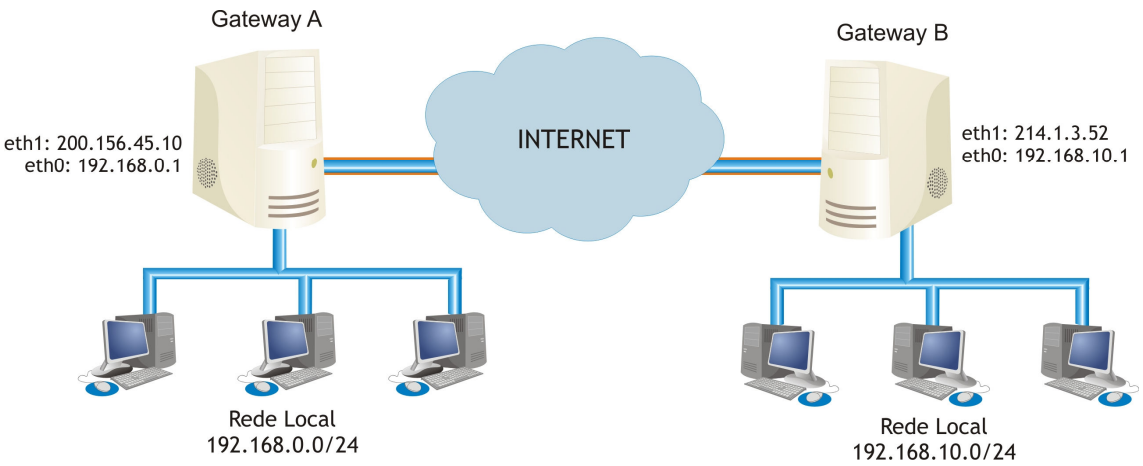

Figura 35 – Cenário de implementação 1

Vamos demonstrar a utilização de uma VPN que tipicamente é criada para atender a necessidade de manter duas redes sempre conectadas como, por exemplo, uma empresa matriz ligada a uma filial, esta filial precisa que seus dados sejam enviados para a matriz com muita rapidez e de maneira segura, portanto vamos criar uma VPN entre os servidores. **Servidor A** (servidor da matriz) e **Servidor B** (servidor da filial). A rede da matriz é 192.168.0.0/24 e a rede da filial é 192.168.1.0/24.

Nos servidores estão instaladas distribuições Linux (Slackware 10.2) e possuem duas interfaces de rede: Eth0 (para rede local) e Eth1 (para acesso a Internet).

Estamos utilizando endereços IP fixos para acesso a Internet onde as duas empresas, matriz e filial, possuem endereços reais. Mas vamos comentar uma forma de acesso através de conexões ADSL com IP dinâmico.

Cenário 2: host-gateway

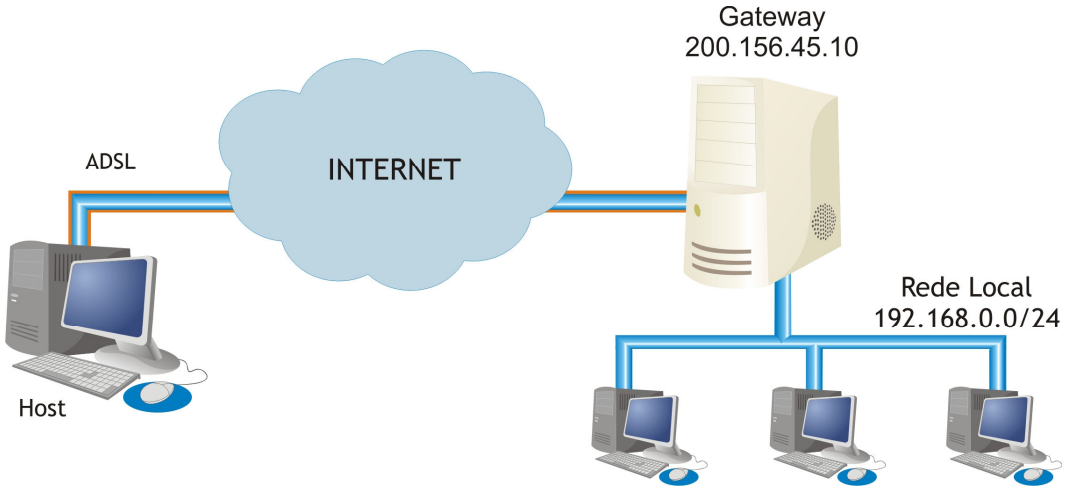

Figura 36 – Cenário de implementação 2

Neste cenário pretendemos apresentar uma solução para o seguinte problema: Imagine uma empresa que possui um grupo de vendedores externos e para melhorar seu atendimento a empresa sempre oferece os menores prazos de entrega. Para realizar este tipo de operação os vendedores precisam entrar no sistema da empresa e realizar seus pedidos diretamente no servidor da matriz. Para isso, nosso servidor de VPN ficará sempre conectado esperando as conexões dos clientes.

Quando um vendedor precisa realizar um pedido, para ele será necessário conectar-se a Internet e criar uma conexão com o servidor da matriz.

Nesta demonstração o servidor possui uma distribuição Linux instalada (Slackware 10.2) com das interfaces de rede: Eth0 (para rede local) e Eth1 (para acesso a Internet). O cliente está rodando Windows XP Sp2 com uma conexão ADSL ou dial-up para acesso a Internet.

5.1 OpenVPN

O OpenVPN é uma solução VPN baseada em Security Sockets Layer (SSL). Ele está disponível para as plataformas: Linux, Windows 2000/XP ou superior, OpenBSD, FreeBSD, NetBSD, Mac OS X e Solaris. É uma solução robusta e muito simples de se instalar em qualquer arquitetura de rede, pois não envolve diretamente o sistema operacional. Possui suporte para autenticação com chaves públicas, chaves estáticas e permite a criação de túneis sem autenticação. Para garantir a criptografia da comunicação o OpenVPN faz uso do OpenSSL, pode ser configurado para utilizar tráfego TCP ou UDP – em uma definição bem simples do seu funcionamento o OpenVPN simplesmente criptografa o pacote e o envia dentro de um novo pacote UDP/TCP [9].

Ele necessita que o sistema operacional tenha suporte ao driver TUN/TAP para que possa funcionar corretamente. No Linux, nas versões de kernel anteriores a 2.2, será necessário realizar a instalação deste driver e recompilar o kernel. Para usuários do kernel 2.4.7 ou superior o suporte a esse driver já está incluído.

Para aplicações mais robustas com tráfego intenso é possível implementar o "load-balance" onde diversos servidores serão os responsáveis pelo acesso a rede interna, com um sistema inteligente o OpenVPN pode gerenciar qual servidor será utilizado para criar o túnel ou os servidores vão sendo utilizados por ordem, ou seja, esgotou o número de conexões permitidas para aquele servidor o túnel será criado em outro.

Ainda permite a utilização da biblioteca LZO para compactação dos dados a serem transmitidos. O OpenVPN pode trabalhar com dois tipos de túneis:

Túneis IP roteáveis – utilizado para rotear o tráfego ponto-a-ponto sem broadcast. São túneis muito fáceis de configurar e apresentam um desempenho muito satisfatório.

Túneis Ethernet Bridge – utilizado para situações onde há necessidade de broadcast sobre o túnel. São túneis um pouco mais complexos de se configurar e precisam da

criação de um dispositivo virtual para realizar a ponte. O pacote do OpenVPN possui scripts pré-configurados para auxiliar na criação desse dispositivo.

5.2 Instalação

A primeira coisa a se fazer é verificar se o kernel tem suporte ao driver TUN/TAP, para isso execute o comando:

```
# locate if_tun.h
```

O retorno deve ser algo parecido com:

```
/usr/include/linux/if_tun.h
```

O site oficial do projeto OpenVPN é http://www.openvpn.net, a versão utilizada para esta demonstração foi a 2.0.7.

O OpenVPN, como foi mencionado anteriormente, faz uso da biblioteca LZO para compactação dos dados a serem transmitidos. Esta biblioteca está disponível para download no endereço: http://www.oberhumer.com/opensource/lzo/download, a versão utilizada para os testes foi a 1.0.8.

Os arquivos utilizados foram:

LZO: lzo-1.0.8.tar.gz
OpenVPN: openvpn-2.0.7.tar.gz

Instalando a biblioteca LZO

Descompacte o pacote da biblioteca.

```
$ tar –xzvf lzo-1.0.8.tar.gz
$ cd lzo-1.0.8
$ ./configure
$ make
$ su
# make install
```

Instalando o OpenVPN

```
$ tar –xzvf openvpn-2.0.7.tar.gz
$ cd openvpn-2.0.7
$ ./configure
$ make
$ su
# make install
```

Agora vamos ativar o roteamento de pacotes IP no sistema:

```
$ su
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Nesta etapa precisamos criar os certificados para os nossos servidores então vamos utilizar o OpenSLL para isso.

O próprio OpenVPN possui um arquivo previamente configurado para nos auxiliar na criação da autoridade certificadora e os certificados digitais.

Criando a Autoridade Certificadora (AC)

Acesse o diretório onde o OpenVPN foi descompactado como por exemplo: /home/usuario/openvpn-2.0.7. Dentro deste diretório existe uma pasta chamada easy-rsa, nela teremos uma série de scripts pré-configurados para facilitar nossa configuração.

Edite o arquivo **vars**.

```
# cd easy-rsa
# vi vars
```

Agora vamos alterar algumas informações do arquivo

export KEY_DIR=”/etc/openvpn/keys”
Define a pasta onde serão gerados os certificados e as chaves.

export KEY_SIZE=1024

Define o tamanho da chave, em bits, que será utilizada, 1024 bits é um tamanho considerável seguro, mas caso deseje-se uma segurança maior pode ser utilizado o valor de 2048.

export CA_EXPIRE=3650

export KEY_EXPIRE=3650

Estes dois parâmetros definem o tempo de validade (em dias) dos certificados que serão gerados, após este número de dias o certificado estará espirado.

Salve as alterações no arquivo e execute os seguintes comandos:

```
# . ./vars
# ./clean-all
# ./build-ca
```

O primeiro comando carrega as configurações do arquivo vars para a memória já o comando clean-all vai limpar qualquer outra informação de uma possível autoridade certificadora existente e por fim o comando build-ca irá criar nossa autoridade certificadora.

Será solicitado um nome para a Autoridade certificadora utilizamos "CA-VPN" entre outras informações. Ao final deste processo serão gerados quatro arquivos:

```
ca.crt
ca.key
index.txt
serial
```

Gerando parâmetros Diffie-Hellman

Agora precisamos gerar os parâmetros Diffie-Hellman, para a troca de chaves durante o estabelecimento das conexões.

```
# ./build-dh
```

Este processo poderá demorar alguns minutos. Será gerado um arquivo dh1024.pem (onde 1024 corresponde ao tamanho da chave "*export KEY_SIZE=1024*").

Criando os certificados

Para criar os certificados digite o comando:

```
# ./build-key-server matriz
# ./build-key filial
# ./build-key cliente
```

Para o campo **Common Name** digite **matriz**, **filial e cliente** respectivamente. Quando for solicitado *"Certificate is to be certifified until Apr 25 18:04:24 2017 GTM (3650 days). Sing the certificate? [yes | no]"* e *"1 out of 1 certificate requests certified, commit? [yes | no]"* escolha a opção yes para ambos.

Serão gerados os seguintes arquivos:

```
matriz.crt
matriz.csr
matriz.key
filial.crt
filial.csr
filial.key
cliente.crt
cliete.csr
cliente.key
```

Mas para nossa implementação serão necessários apenas os arquivos com extensão .crt e .key.
Para maior segurança é recomendado que os dados da CA fiquem em outro computador ou em uma mídia segura que possa ser acessado quando for necessário.

Para adicionar novos usuários a nossa VPN basta executar o script build-key no computador onde a CA está configurada. Existe também a possibilidade de gerar certificados com senha, para isso deve executar o script build-key-pass e após solicitar os dados para geração do certificado será solicitado uma senha que futuramente será digitada pelo usuário sempre que ele se conectar a VPN.

Scripts de configuração (Gateway – Gateway)

Servidor A - Matriz

Para realizar a configuração vamos utilizar como base o arquivo tls-office.conf contido do diretório sources do OpenVPN [9]. Segue listagem abaixo:

Conteúdo do arquivo /etc/openvpn/matriz.conf

```
# Usar como interface o driver tun.
dev tun

# 10.10.0.1 IP do túnel da matriz.
# 10.10.0.2 IP do túnel da filial.
ifconfig 10.10.0.1 10.10.0.2

# Arquivo que contém as rotas.
# Ele será executado quando o túnel estiver concluído.
up ./etc/openvpn/matriz.up

# Informa que é o servidor da transmissão
tls-server

# Parâmetros Diffie-Hellman
dh /etc/openvpn/dh.pem

# Certificado da CA
ca /etc/openvpn/ca.crt

# Certificado público da Matriz
cert /etc/openvpn/matriz.crt

# Certificado privado da Matriz
key matriz.key

# Porta UDP para transmissão.
port 5000

# Envia um ping via UDP para a parte
# remota a cada 15 segundos.
ping 15

# logs.
log /var/log/openvpn.log

# nível de detalhes do log (pode variar de 0 até 9)
verb 3
```

Arquivo de rotas *matriz.up*

```
#!/bin/bash
route add -net 192.168.10.0 netmask 255.255.255.0 gw $5
```

O parâmetro $5 representa o endereço IP do gateway VPN.

Executando o OpenVPN no servidor.

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

Servidor B - Filial

A configuração do servidor da filial será muito semelhante a do servidor da matriz e também utilizaremos o arquivo tls-office.conf como base para nossa configuração. Segue listagem abaixo:

Conteúdo do arquivo /etc/openvpn/filial.conf

```
# Usar como interface o driver tun.
dev tun

# Ip da Matriz.
# Caso o servidor não possua um IP fixo
# poderá ser utilizado algum serviço de DNS dinâmico
remote 200.156.45.10

# 10.10.0.2 IP do túnel da filial.
# 10.10.0.1 IP do túnel da matriz.
ifconfig 10.10.0.2 10.10.0.1


# Arquivo que contém as rotas.
# Ele será executado quando o túnel estiver concluído.
up ./etc/openvpn/filial.up


# Informa que é o cliente da transmissão
tls-client

# Certificado da CA
ca /etc/openvpn/ca.crt

# Certificado público da Matriz
cert /etc/openvpn/filial.crt
```

```
# Certificado privado da Matriz
key /etc/openvpn/filial.key

# Porta UDP para transmissão.
port 5000

# Envia um ping via UDP para a parte
# remota a cada 15 segundos.
ping 15

# logs.
log /var/log/openvpn.log

# nível de detalhes do log (pode variar de 0 até 9)
verb 3
```

Arquivo de rotas *filial.up*

```
#!/bin/bash
route add -net 192.168.0.0 netmask 255.255.255.0 gw $5
```

Da mesma forma que na configuração do servidor o parâmetro $5 representa o endereço IP do gateway VPN.

Executando o OpenVPN na filial.

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

Testando a conexão VPN.

Para testar a conexão, basta executarmos o comando ping de uma extremidade da conexão para outra. Supondo que na rede da Matriz tenhamos uma maquina cujo IP é 192.168.0.27 então, de qualquer maquina da rede da Filial executamos o comando ping para esta máquina.

ping 192.168.0.27
Resposta de 192.168.0.27: bytes=32 tempo<43ms TTL=64
Resposta de 192.168.0.27: bytes=32 tempo<21ms TTL=64

E o mesmo processo será feito de uma máquina da rede da Matriz para a rede da Filial.

ping 192.168.10.43

Resposta de 192.168.10.43: bytes=32 tempo<64ms TTL=64

Resposta de 192.168.10.43: bytes=32 tempo<39ms TTL=64

Scripts de configuração (Host – Gateway)

Ao se conectar com o Gateway VPN o host recebera um IP da rede local, e assim passará a fazer parte da rede. Aqui utilizaremos as ferramentas de bridge-utils[5] disponíveis junto ao pacote do OpenVPN

Gateway da Matriz - Servidor

Conteúdo do arquivo /etc/openvpn/servidor.conf

```
# Usar como interface o driver tap.
dev tap0

# Especifica que o servidor irá se comportar como uma bridge
# 192.168.0.100 255.255.255.0
# IP  e marcara da bridge depois que o túnel for criado
# 192.168.0.192 192.168.0.254
# Faixa de IPs que estarão disponíveis para os clientes
Server-bridge    192.168.0.100    255.255.255.0    192.168.0.192
192.168.0.254

# Parâmetros Diffie-Hellman
dh /etc/openvpn/dh.pem

# Certificado da CA
ca /etc/openvpn/ca.crt

# Certificado público da Matriz
cert /etc/openvpn/matriz.crt

# Certificado privado da Matriz
key matriz.key

# Porta para conexão
port 1194
# Permite que o cliente veja as máquinas da matriz
# no ambiente de rede do Windows
client-to-client
```

```
# Ativa a compactação LZO
comp-lzo

# Determina o número máximo de conexões simultâneas
max-clients 15

# logs.
log /var/log/openvpn.log

# nível de detalhes do log (pode variar de 0 até 9)
verb 3
```

Executando o OpenVPN no servidor.

```
# openvpn --config /etc/openvpn/servidor.conf --daemon
```

Para termos certeza de que o serviço está rodando devemos verificar o log do OpenVPN.

```
# tail /var/log/openvpn.log
```

Instalando o cliente Windows

Existe um cliente para Windows disponível para download no em http://openvpn.se este cliente cria um ícone na barra de tarefas do Windows e através dele o usuário dera conectar e desconectar os túneis. A instalação é muito simples e segue o padrão da maioria dos softwares para Windows.

1- Baixar o cliente OpenVPN (http://openvpn.se/download.html) a versão utilizada é a 2.0.7 o arquivo de intalação é: openvpn-2.0.7-gui-1.0.3-install.exe;
2- Execute o arquivo;
3- Na janela que se abrirá para instalação clique no botão “Next”;
4- Clique no botão “I agree” para aceitar os termos de utilização do software;
5- Deixe a marcação padrão e clique no botão “Next”;
6- Clique sobre o botão “Install”e aguarde o final da instalação. Se durante este processo aparecer uma janela conforme a figura 37 clique no botão “SIM”;

Figura 37 – Instalação do cliente OpenVPN para Windows

7- Após a copia dos arquivos clique sobre o botão “Next >” e depois sobre o botão “Finish. Ao concluir a instalação aparecerá ao lado do relógio um ícone conforme a figura 38. Isto representa que sua instalação foi bem sucedida.

Figura 38 – Ícone do OpenVPN para Windows

Após concluir a instalação devemos criar o arquivo de configuração do cliente.

Conteúdo do arquivo cliente.ovpn

```
# Usar como interface o driver tap.
dev tap0

# Determina que é o cliente na transmissão
client

# Protocolo
proto upd


# Ip da parte remota
# Caso o servidor não possua um IP fixo
# poderá ser utilizado algum serviço de DNS dinâmico
remote 200.156.45.10
```

```
# Não permite alteração do IP e porta após a conexão
nobind

# Número de tentativas de conexão com o servidor
resolve-retry infinite

# Certificado da CA
ca ca.crt

# Permite ao cliente receber configurações do servidor
pull

# Certificado público da Matriz
cert matriz.crt

# Certificado privado da Matriz
key matriz.key

# Porta para conexão
port 1194

# Permite que o cliente "enxergue" as máquinas da matriz
client-to-client

# logs.
log /var/log/openvpn.log

# nível de detalhes do log (pode variar de 0 até 9)
verb 3
```

Ao concluir a instalação do cliente OpenVPN acesse a pasta onde o software foi instalado (a instalação padrão é "C:\Arquivos de Programas\OpenVPN");

Abra o diretório "config" e copie o arquivo de configuração para esta pasta junto com os certificados e a chave secreta.

Certificado da CA: ca.crt
Certificado do usuário: cliemte.crt
Chave Privada: cliente.key
Arquivo de configuração: cliente.ovpn
Agora basta clicar sobre o ícone do OpenVPN, localizado próximo ao relógio do Windows, com o botão direito do mouse e clicar sobre a opção "Connect". Após concluir alguns testes a conexão estará realizada.

Testando a conexão

Da mesma forma que na topologia gateway-gateway, vamos utilizar o comando ping para realizar os testes em nossa conexão. Supondo que na nossa rede exista um servidor http, cujo IP é 192.168.0.128, então da maquina do cliente executamos o comando ping.

ping 192.168.0.128
Resposta de 192.168.0.128: bytes=32 tempo<60ms TTL=128
Resposta de 192.168.0.128: bytes=32 tempo<54ms TTL=128

Gerenciando usuários

A gerência dos usuários pode ser feita de duas formas [10]. Podemos criar um túnel para cada usuário, configurando uma porta para cada cliente e consequentemente um daemon no servidor para cada conexão, outra forma e deixarmos o OpenVPN gerenciar os túneis utilizando as ferramentas de bridge e a partir daí podemos criar rotas inválidas para os usuários que não possuem mais o acesso à rede ou então podemos revogar o seu certificado.

Para criar rotas invalida vamos adicionar a linha abaixo no arquivo de configuração do servidor, esta linha aponta para um diretorio que contém os arquivos com rotas inválidas para os clientes.

```
client-config-dir ccd
```

Crie o diretório ccd, no servidor, e dentro dele um arquivo conforme abaixo:

```
# cat ccd/cliente
ifconfig-push 10.0.0.5 10.0.0.25
```

Assim quando o usuário cliente se conectar ao servidor VPN ele pegará uma rota inválida, pois este comando seta novos IPs para conexão VPN.

Outra forma de bloquear um cliente de se conectar ao servidor é revogarmos seu certificado, para isso utilizamos um script pré-configurado do OpenVPN que está na pasta easy-rsa. Ao executar o comando ele irá atualizar ou criar o arquivo crl.pem.

```
# revoque-full cliente
```

Após revogar o certificado devemos inserir a linha abaixo no arquivo de configuração do servidor, servidor.conf, caso ele esteja rodando deverá ser reiniciado.

```
crl-verify crl.pem
```

Assim quando um usuário tentar se conectar ao servidor receberá uma mensagem de erro informando uma falha no processo de handshake.

5.3 Softwares desenvolvidos

Para complementar o serviço de VPN foi desenvolvido um sistema para distribuição dos manuais, softwares, senhas e certificados. Este sistema foi desenvolvido em PHP utilizando o banco de dados MySQL e o apache como servidor.

O sistema consiste em um cadastro de usuários que possuem acesso ao serviço de VPN, após realizar o cadastro do usuário no servidor de VPN e gerar os certificados o operador insere no banco de dados às informações referentes ao usuário criado. Os certificados são manualmente copiados para um diretório que está acessível pelo servidor apache. Quando o usuário se autenticar no sistema ele terá disponível os links para baixar os manuais e softwares para instalação assim como seu certificado, certificado do servidor e sua senha do certificado.

Esta senha é gerada de forma aleatória contendo obrigatoriamente letras, números e totalizando oito caracteres.

Capítulo 6

Conclusão

Ao final deste trabalho, conclui-se que uma solução VPN é a forma mais econômica de se conectar redes através de um meio público. Dentre os protocolos apresentados os que se demonstraram eficazes para elaboração de uma VPN foram: o IPSec e o SSL, os demais apresentaram falhas ou deficiências de segurança, portanto foram considerados inadequados.

Optou-se por utilizar o OpenVPN, por ser um software livre, possui fácil implementação e utiliza túneis SSL. O SSL é amplamente utilizado para transações segurar pela Internet e está em constante atualização, conforme mencionado anteriormente.

Ao final da implementação foi possível perceber que conseguimos atender as características desejáveis para uma VPN segura utilizando o OpenVPN, ele possui métodos de criptografia eficientes, suporte para gerenciamento de chaves, suporte a múltiplos protocolos, gerenciamento de endereços dos clientes e autenticação.

Esta solução foi implementada no Laboratório Nacional de Computação Científica utilizando uma topologia host-gateway visando permitir que usuários externos obtivessem acesso aos recursos da rede interna. A implementação realizada neste trabalho foi baseada nesta experiência.

Mas somente a VPN não é suficiente, para permitir que usuários externos tenham acesso aos recursos da rede de forma segura, é extremamente recomendado possuir outras forma de proteção como firewall e uma política de segurança bem elaborada.

Bibliografia

[1] KOLESNIKON, Oleg e HATCH, Brian. “Building Linux Virtual Private Networks (VPNs)”
EUA, Editora New Riders, 2002.

[2] S. DA SILVA, Lino. “Virtual Private Network – VPN”
2º Ed., São Paulo, Editora Novatec, 2005

[3] T. Dierks, C. Allen, The TLS Protocol Version 1.0. RFC 2246, 1999.

[4] http://www.certicom.com

[5] http://www.openvpn.org

[6] http://www.wikipedia.com

[7] http://www.absoluta.org/cripty/cripty_h.htm

[8] http://www.rnp.br/newsgen/9811/vpn.html

[9] http://epx.com.br/artigos_3rd/howto.php

[10] http://www.istf.com.br/vb/showthread.php?t=8624