Rafael Moreno Ribeiro do Nascimento

Estudo de Caso para implantação de VPN na Unimontes

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Orientador
Prof. MsC. Denilson Vedoveto Martins

Lavras
Minas Gerais - Brasil
2009

Rafael Moreno Ribeiro do Nascimento

Estudo de Caso para implantação de VPN na Unimontes

Monografia de Pós-Graduação "*Lato Sensu*" apresentada ao Departamento de Ciência da Computação para obtenção do título de Especialista em "Administração em Redes Linux"

Aprovada em *novembro de 2009*

Prof. MsC. Joaquim Quinteiro Uchôa

Prof. MsC. Herlon Ayres Camargo

Prof. MsC. Denilson Vedoveto Martins
(Orientador)

Lavras
Minas Gerais - Brasil
2009

Agradecimentos

A Deus;

À minha linda namorada Laura, pela paciência, carinho e apoio, que foram fundamentais para a execução deste trabalho;

À minha família, aos professores e a todos que de alguma forma contribuíram para que este trabalho fosse concluído;

Ao meu orientador Denilson pela atenção e apoio em todas as etapas deste trabalho.

Sumário

Lista de Figuras

Resumo

Este trabalho tem como objetivo mostrar o uso de VPN sobre a internet como a alternativa de baixo custo, gerenciável e segura para interligar os *Campi* da Unimontes. A monografia apresenta técnicas de criptografia, tunelamento e comutação de pacotes, incluindo exemplos e diversos algoritmos na revisão bibliográfica, assim como alternativas para o serviço, vantagens e desvantagens para um completo entendimento do leitor. Finalmente, será mostrado como ocorreu a implementação e na conclusão será apontado o que foi possível com o uso deste serviço.

Palavras-Chave: VPN; Internet; Criptografia; Tunelamento; Comutação de pacotes.

Capítulo 1

Introdução

A necessidade de comunicação entre pessoas e empresas vem crescendo muito, uma vez que as distâncias estão aumentando cada vez mais, porém esta necessidade de se compartilhar recursos e informações, deixa as pessoas cada vez mais dependentes dos meios de comunicação.

Os computadores e os meios de comunicação vêm se fundindo cada vez mais, reduzindo custos, aumentando as possibilidades e integrando um maior número de usuários, independente da localização física dos mesmos.

A internet é um recurso que está revolucionando as comunicações, devido à facilidade de uso e ao custo cada vez menor de se manter um *link*. Porém, a *internet* é um meio de comunicação muito promíscuo com uma diversidade de usuários com necessidades diferentes, bem ou mal intencionados.

As redes de computadores vêm crescendo ultimamente, expandindo-se além das redes locais, surgindo cada vez mais organizações que precisam interligar entre cidades as suas empresas, aumentando as possibilidades de integração de serviços e facilitando a comunicação. Existem alguns problemas advindos de se ter interligação de redes entre cidades, relacionados a questões de segurança e custos.

Para suprir estas necessidades, foram criados alguns recursos de comunicação, com características diferentes para demandas diferentes, recursos como circuitos dedicados que devem ser contratados das operadoras de telecomunicação, com um alto grau de confiabilidade e segurança, mas com um alto custo. Foram criadas também, soluções como as VPNs(*Virtual Private Networks*), que surgiram como uma alternativa a estes circuitos dedicados, provendo uma segurança satisfatória

garantida por verdadeiros circuitos virtuais criados sobre uma rede pública como a internet.

O objetivo deste TCC (Trabalho de Conclusão de Curso) é apresentar a solução de uma VPN na Unimontes(Universidades Estadual de Montes Claros), suprindo as demandas existentes e as futuras demandas de compartilhamento de recursos de forma segura.

Para o desenvolvimento deste trabalho, foi escolhido como tema uma demanda que estava reprimida na instituição - a interligação entre os Campi. Após alguns dias de pesquisa na internet, foi escolhida a VPN (*Virtual Private Network*) como solução. A VPN iria suprir a demanda de forma segura e com um custo baixo de implementação e manutenção, pois seria implementada pela própria equipe e não haveria preocupações com o custo de licenças de *software*, porque a solução seria totalmente feita utilizando software livre.

A monografia está organizada da seguinte maneira: no segundo capítulo, é discutido o cenário da instituição em que a solução será implantada, comentando sobre a referida instituição e sobre as demandas que fizeram surgir a necessidade da VPN. No terceiro capítulo, são explicados conceitos de redes baseadas no padrão TCP/IP, padrão que foi utilizado na implementação da solução. No quarto capítulo, são mostrados conceitos de criptografia, importantes para que o leitor possa entender o funcionamento da VPN. No quinto capítulo, é apresentado o conceito de tunelamento, o que é importante para que o leitor entenda como as VPNs são estabelecidas. No sexto capítulo, é explicado o que são VPNs, bem como as topologias existentes, as vantagens, desvantagens e alternativas ao uso de VPNs. O sétimo capítulo trata da implementação da ferramenta, explicando as técnicas utilizadas diretamente nos arquivos de configuração do OpenVPN, bem como os comandos utilizados para a criação das chaves.

Capítulo 2

Cenário

A Unimontes (Universidade Estadual de Montes Claros) tem em sua abrangência 30% do Estado, constituindo-se de 12 *Campi* espalhados por Minas Gerais.

A universidade tem um sistema integrado com a Secretaria de Planejamento do Estado de Minas Gerais (Seplag), no qual a universidade presta contas das despesas, frequência dos recursos humanos, utilização de seus veículos e movimentação do almoxarifado. Este sistema é ligado via circuito *frame-relay* com a Companhia de Tecnologia da Informação do Estado de Minas Gerais (PRODEMGE), porém a universidade possui vários Campi, como dito anteriormente, sendo muito oneroso manter um *link frame-relay* em cada cidade, uma vez que este é um serviço que as operadoras cobram um preço alto para a contratação, fato este que exigiu que a Gerência de Tecnologia da Informação (GTI) da universidade procurasse uma forma em que se centralizasse as conexões no Campus sede da Unimontes em Montes Claros.

Atualmente, a Universidade conta com uma secretaria, responsável pelas atividades referentes ao registro e controle acadêmico, tendo como principais atividades: matrícula, emissão de históricos e declarações, processos de transferências, expedição de diplomas, divulgação de resultados, protocolo de requerimentos diversos. Para tanto, conta com um banco de dados centralizado no Campus Montes Claros, sendo que este abrange todos os *Campi*. Devido ao acesso a este banco de dados ser feito exclusivamente com uma aplicação que funciona na rede local do *Campus*, os servidores responsáveis pelas secretarias dos Campi externos precisam se deslocar até o *Campus* de Montes Claros para fazerem a inserção de dados no sistema, gerando despesas com deslocamento e atrasando todo o processo de integração entre as secretarias.

A universidade possui um sistema de biblioteca da instituição que exige o acesso diretamente ao banco de dados. Como a Universidade possui bibliotecas em várias cidades, seria necessário, devido à engenharia de software utilizada no sistema, que o banco de dados estivesse conectado diretamente à internet, o que não seria uma atitude muito prudente do ponto de vista da segurança.

Devido a estas demandas, fez-se necessário prover o acesso ao banco de dados da rede interna a partir dos *Campi* das outras cidades, e ao circuito *frame-relay* de forma segura e com um orçamento reduzido.

Capítulo 3

Redes de computadores

A comunicação em rede é feita através de protocolos que estão organizados em camadas, independentes umas das outras, cada uma com a sua função específica, conforme veremos a seguir.

3.1 Classificação das redes

3.1.1 LANs (*Local Area Networks*)

Pode-se caracterizar uma rede local como sendo uma rede que permite a interconexão de equipamentos de comunicação de dados numa pequena região (SOARES G. LEMOS, 1995), limitada a um prédio, ou um conjunto de prédios, como um *campus* universitário.

3.1.2 MANs *(Metropolitan Area Networks)*

Quando a distância de ligação entre os vários módulos processadores começa a atingir distâncias metropolitanas, chamamos esses sistemas de redes metropolitanas (*Metropolitan Area Networks*). Uma rede metropolitana apresenta características semelhantes às das rede locais, sendo que as MANs, em geral, cobrem áreas maiores (SOARES G. LEMOS, 1995).

3.1.3 WANs *(Wide Area Networks)*

As redes geograficamente distribuídas, surgiram da necessidade de se compartilhar recursos especializados por uma maior comunidade de usuários geograficamente dispersos. Por terem um custo de comunicação bastante elevado, tais redes são em geral públicas, isto é, o sistema de comunicação, chamado sub-rede de comunicação, é mantido, gerenciado e de propriedade de grandes operadoras (públicas ou privadas), e seu acesso é público (SOARES G. LEMOS, 1995).

3.2 Modelo TCP/IP

O padrão TCP/IP *(Transmission Control Protocol/Internet Protocol)* foi criado pelo Departamento de Defesa Americano (DoD) para garantir a preservação da integridade dos dados, assim como para manter a comunicação de dados no advento de uma guerra.

Se bem planejada e corretamente implementada, uma rede baseada na combinação de protocolos (suite) TCP/IP pode ser independente, confiável e muito eficiente.

Graficamente mostrado abaixo e explicado detalhadamente a seguir:

Figura 3.1: Modelo TCP/IP de 5 camadas

3.2.1 Camada de aplicação

É responsável pela definição dos protocolos necessários para a comunicação ponto a ponto pelas aplicações, bem como pelo controle e especificações da interface com o usuário (FILIPPETTI, 2008). Sua comunicação é estabelecida via *socket*, que é a interface de comunicação entre a camada de aplicação e a de transporte. O desenvolvedor de uma aplicação tem pouco acesso a parâmetros da camada de transporte, no máximo escolhendo qual protocolo será utilizado, tamanho do *buffer* ou segmentos.

As aplicações que funcionam nos servidores, também conhecidas como serviços em alguns sistemas operacionais, são representadas por portas, isto é, uma comunicação é estabelecida utilizando-se o endereço IP (*Internet Protocol*) e a porta. Quando o sistema operacional percebe que o cliente está requisitando uma porta específica, ele disponibiliza a aplicação que utiliza aquela determinada porta. As portas são padronizadas para cada tipo de aplicação, sendo que a lista das referidas portas pode ser encontrada em www.iana.org.

Exemplos de aplicações e portas:

80 http (*Hypertext Transfer Protocol*);

25 SMTP (*Simple Mail Transfer Protocol*);

22 SSH (*Secure Shell*).

3.2.2 Camada de transporte

A camada de transporte fornece comunicação lógica entre processos e aplicações. É uma camada que funciona apenas nos sistemas finais, sem a preocupação com detalhes de como as mensagens chegaram até ela.

A camada de transporte possibilita que vários sistemas utilizem uma mesma conexão utilizando multiplexação e demultiplexação. A tarefa de entregar os dados contidos em um segmento da camada de transporte à porta correta é denominada demultiplexação. Já o trabalho de reunir, no hospedeiro de origem, porções de dados provenientes de diferentes portas, encapsular cada porção de dados com informações de cabeçalho para criar segmentos e passar esses segmentos à camada de rede é denominada multiplexação (KUROSE, 2006).

A camada de transporte também possibilita dois tipos de serviços, sendo um orientado à conexão (serviço confiável à aplicação solicitante) TCP - *Transmission*

Control Protocol, e um não orientado à conexão (serviço não confiável a aplicação solicitante) UDP - *User Datagram Protocol*.

3.2.2.1 Características do protocolo UDP:

Melhor controle no nível da aplicação sobre quais dados são enviados e quando a aplicação passa os dados ao UDP, imediatamente o protocolo empacotará os dados e os enviará para a camada de rede, enquanto que o TCP tem um mecanismo de controle de congestionamento que limita o remetente da camada de transporte quando os enlaces ficam muito congestionados.

Não há estabelecimento de conexão: enquanto o protocolo TCP usa uma apresentação de três vias, o UDP simplesmente envia mensagens sem nenhuma negociação anterior.

Não há estados de conexão: enquanto o TCP mantém uma conexão, verificando questões como congestionamento, *buffers* de envio e recebimento, sequência e reconhecimento. O UDP não faz nenhum dos tratamentos anteriores, possibilitando a conexão com um número maior de clientes simultâneos.

Pequena sobrecarga de cabeçalho de pacotes: enquanto o segmento TCP utiliza 20 *bytes* de sobrecarga de cabeçalho, o UDP usa apenas 8 *bytes*, fato este que acaba possibilitando uma economia de banda.

3.2.2.2 Características do protocolo TCP:

O TCP tem como característica principal ser um protocolo de transferência confiável de dados, utilizando técnicas como controle de fluxo, números de sequência, reconhecimentos e temporizadores. Garantindo que os dados sejam entregues ao destinatário corretamente e em ordem. As técnicas utilizadas para que este mecanismo funcione são mostradas a seguir:

• Números de sequência e números de reconhecimento:

O número de sequência é o campo que identifica a sequência em que os dados estão durante a transferência no fluxo de *bytes* entre cliente e servidor, enquanto o número de reconhecimento é o campo que especifica o próximo *byte* que o receptor espera receber.

• Controle de fluxo:

Os controle de fluxo é um serviço de compatibilização de velocidade entre o remetente e o destinatário da transmissão.

• Temporizadores:

Também conhecido como *TCP Timers*, os temporizadores são os campos responsáveis por definir sempre que uma ação ou resposta são exigidos do *host* remoto.

3.2.3 Camada de rede

A camada de rede está presente em cada um dos hospedeiros e roteadores na rede, diferentemente das camadas de transporte e aplicação.

Na camada de rede é que são realizadas as funções de endereçamento, repasse (transferência de um enlace de entrada para um enlace de saída dentro de um único roteador, ou seja, apenas informa o caminho até o próximo roteador). O roteamento consiste na rota completa abrangendo todos os roteadores, isto é inclui do primeiro até o último.

Assim como a camada de transporte, a camada de rede tem a capacidade de prover um serviço orientado ou não à conexão. As redes com serviços orientados à conexão são chamadas redes de circuitos virtuais (redes CV), já as redes com serviços não orientados à conexão são chamadas redes de datagramas[1].

3.2.4 Camada de enlace

A camada de enlace é responsável pela conexão entre os nós da rede. Entenda-se nós como sendo os roteadores em uma WAN *(Wide Area Network)* ou pela conexão entre placas de rede em uma LAN *(Local Area Network)*. A unidade de dados nesta camada é o quadro, local onde são encapsulados os dados na conexão. Alguns exemplos de protocolos de camada de enlace são, ethernet, o protocolo mais amplamente utilizado em todo o mundo para redes locais, o protocolo 802.11 para LANs sem fio e ATM para redes WAN.

Entre as ações realizadas por um protocolo de camada de enlace ao enviar e receber quadros, estão detecção de erros, retransmissão, controle de fluxo e acesso aleatório (KUROSE, 2006).

[1] É a estrutura de dados unitária de transmissão em uma rede (de computadores ou telecomunicações)

3.2.5 Camada física

A camada física é responsável pelas características físicas dos equipamentos, como a quantidade de conectores, metragens e características elétricas. Nessa camada são definidos por exemplo os tipos de cabos e os conectores específicos para estes cabos, além de serem definidas as distâncias suportadas para os cabos, a tensão que será suficiente para alimentar os equipamentos e o nível de interferência eletromagnética suportada pelos equipamentos.

Na camada física, serão definidas as características do cabeamento horizontal, ou seja, o cabeamento responsável pela ligação com as estações de trabalho e do cabeamento vertical ou *backbone* que é o cabeamento responsável pela interligação entre os pontos da rede. Para o perfeito funcionamento dos mesmos é necessário seguir os padrões de cada cabeamento.

3.3 Rede de circuito virtual

Uma rede de circuito virtual é uma conexão virtual criada em uma rede de forma que todos os roteadores intermediários entre os sistemas inicial e final façam parte da conexão com um roteamento pré-definido, onde todos os saltos da rede possuem o identificador do circuito virtual. Exemplos de circuitos virtuais são as redes ATM *Asynchronous Transfer Mode*, *Frame Relay* e X.25.

3.4 Rede de datagramas

Em uma rede de datagramas, cada pacote que transita por ela contém em seu cabeçalho o endereço de destino. Assim, o pacote segue um caminho hierárquico, não existindo uma rota pré-definida, sendo que cada pacote é analisado durante seu percurso por cada roteador e verificado para qual roteador este deve ser repassado até chegar ao seu destino final. A internet, também chamada de rede pública, é um exemplo de rede de datagramas.

Capítulo 4

Criptografia

A sociedade está cada vez mais dependente dos meios de comunicação via *internet*, aumentando cada vez mais a sua utilização. Da mesma forma, vem aumentando o número de fraudes a estes meios de comunicação, ameaçando que a informação seja entregue de forma confiável ao seu destinatário, como podemos observar na Figura 4.1.

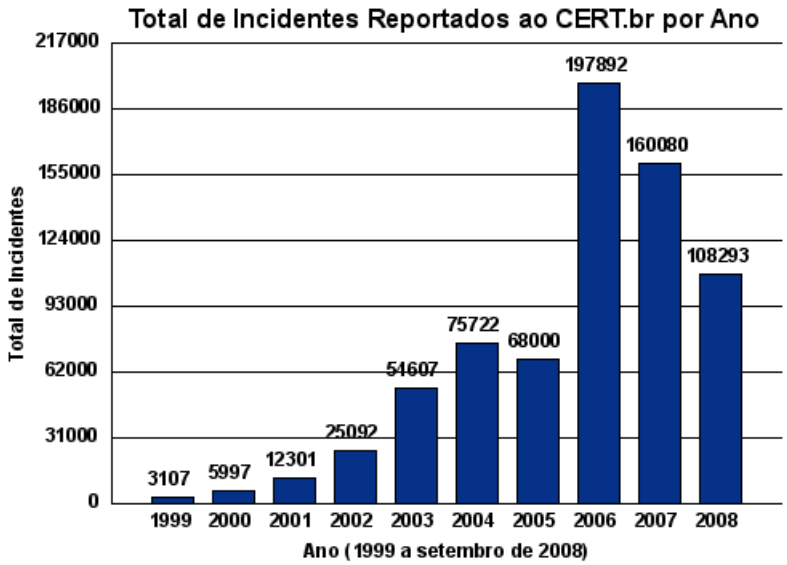

Figura 4.1: Gráfico de incidentes

A criptografia é uma forma de garantir que as informações sejam trafegadas nos meios de comunicação, mantendo-se a integridade, confidencialidade e disponibilidade.

De acordo com (APPLIED, 1996), a palavras criptografia vem do grego (*Kryptos* = escondido, oculto e Grafia = Escrita). A criptografia é a ciência que usa a matemática para encriptar e decriptar dados, permitindo que informações sigilosas ou mensagens sejam transmitidas por redes e trafeguem livremente sem serem lidas por indivíduos ou instituições que não sejam o destinatário final da mensagem. Dessa forma, ela garante confidencialidade, autenticidade, integridade e não-repúdio.

O processo de criptografia pode ser descrito da seguinte forma: um emissor gera a mensagem original, chamada de texto plano, e, utilizando uma chave e um algoritmo de cifragem, gera um texto cifrado, ou seja, incompreensível para quem não tem autorização de lê-lo. Ao chegar ao receptor, este texto passa pelo processo inverso, chamado decifragem, resultando no texto plano original, observado na Figura 4.2.

Figura 4.2: Processo de cifragem e decifragem de arquivo

4.1 Criptografia Simétrica

A criptografia simétrica é um método criptográfico de chave única, onde o emissor e o destinatário possuem a mesma chave. Trata-se de um método bastante rápido para criptografar e decriptografar a informação, porém exige um canal seguro para que a chave possa transitar inicialmente, uma vez que tendo o intruso acesso à chave, ele poderá ler todas as mensagens trocadas, podendo até comprometê-las. A figura 4.3 ilustra o seu funcionamento.

A criptografia simétrica não garante a identidade de quem enviou ou recebeu a mensagem.

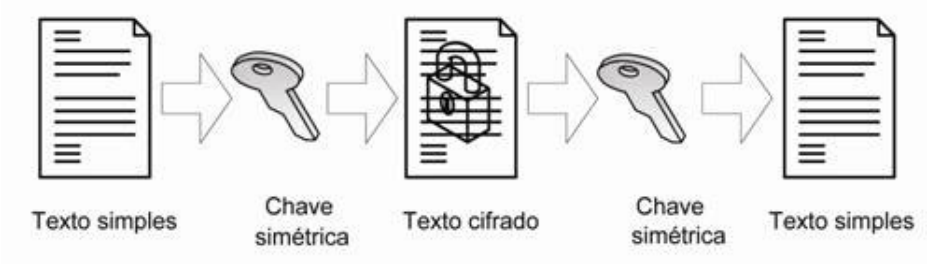

Figura 4.3: Processo de criptografia simétrica

4.1.1 Algoritmos simétricos:

4.1.1.1 DES (*Data Encryption Standart*) e *Triple* DES

O DES é um algoritmo simétrico (o processo de encriptação é o mesmo de decriptação) de 64 *bits*. Para cada 64 *bits* de texto simples na entrada do algoritmo, surgem 64 *bits* de texto criptografado na saída. Apesar de a chave do DES ser representada como um número de 64 *bits*, o seu tamanho é 56 *bits*, pois todos os oitavos bits são usados somente como *bits* de paridade.

A transformação de texto simples em texto cifrado no DES dá-se em 18 estágios. No primeiro estágio, é feita uma transposição da chave no texto de 64 *bits*, ao passo que no último estágio ocorre uma transposição inversa do primeiro estágio. Estes dois estágios não afetam na segurança do DES, podendo ser retiradas do DES a fim de facilitar a implementação do DES em *software*. Contudo, essa modificação no algoritmo o coloca fora do padrão do DES. As 16 etapas intermediárias são transformações idênticas (HINZ, 2000).

4.1.1.2 IDEA

O algoritmo de IDEA, inicialmente chamado de IPES (*Improved Proposed encryption standart*), foi idealizado por Lai e Massey em 1991, com intuito de ser eficiente em aplicações por *software* LaiI (1990), citado em Moreno (2005).

O IDEA possui chave secreta de 128 *bits* e tanto na entrada de texto legível, como na saída de texto cifrado são 64 *bits*. É um algoritmo capaz de criptografar e de decriptografar, porem são utilizadas duas fases diferente: 1 – é baseada na realização de oito iterações utilizando subchaves distintas, 2 – uma transformação final (FERNANDES, 2007).

4.1.1.3 RC2 e RC4

O RC2 é uma cifra de bloco simétrico de 64 *bits* de comprimento de chave variável; o RC4 é uma cifra de fluxo de comprimento de chave variável. O tamanho da chave para qualquer dos algoritmos pode ter de 1 a 2048 *bits* de comprimento. Em geral, os algoritmos são usados com chaves de 40 *bits* e de 128 *bits*. Uma chave de 40 *bits* é muito pequena para proteger qualquer item de qualquer valor, mas é amplamente usada devido a regras de exportação anteriores dos EUA, que impediam a exportação de produtos usando chaves mais longas.

Esses algoritmos foram desenvolvidos por Ronald Rivest e são segredos comerciais da RSA *Data Securety*. O algoritmo RC4 foi revelado em 1994 por uma postagem anônima da Usenet; o RC2 teve o mesmo destino em 1996. Ambos os algoritmos parecem ser razoavelmente fortes. Quando implementados em *software*, eles são cerca de dez vezes mais rápidos que o DES (ZWICKY, 2000).

4.1.1.4 *Blowfish*

O *Blowfish* é uma cifra de bloco simétrico de 64 *bits* com uma chave de comprimento variável. A chave pode ter de 32 a 448 *bits* de tamanho. O *Blowfish* foi criado por Bruce Schneier e foi lançado em 1994. O algoritmo parece ser forte. Ele foi projetado para uso em microprocessadores de 32 *bits* usando operações matemáticas simples. Ele tem requisitos de memória maiores que os outros algoritmos, o que o torna menos atraente para cartões inteligentes e outros dispositivos pequenos. O *Blowfish* não é patenteado e implementações em C estão no domínio público. Quando implementado em *software*, o *Blowfish* é executado em aproximadamente cinco vezes a velocidade do 3DES (ZWICKY, 2000).

4.2 Criptografia Assimétrica ou de chave pública

A criptografia de chave assimétrica resolve o problema da criptografia da chave simétrica de transferência inicial da chave. Na criptografia de chave assimétrica existem duas chaves, uma pública e uma privada, de forma que a chave pública é distribuida entre os elementos da comunicação e é responsavel por criptografar a mensagem. Por outro lado, a chave privada é responsavel por decriptografar as mensagens.

4.2.1 Funcionamento

Chave privada

A chave privada pode criar chaves públicas, podendo a chave privada decriptografar a mensagem. Referida chave deve ser mantida em segredo.

Chave pública

A chave pública não pode criar uma chave privada, podendo apenas criptografar a mensagem, sem a possibilidade da operação inversa.

A dificuldade de quebra está no tempo que levaria conseguir uma chave privada a partir da chave pública.

O funcionamento da criptografia de chave privada é ilustrado na Figura 4.4.

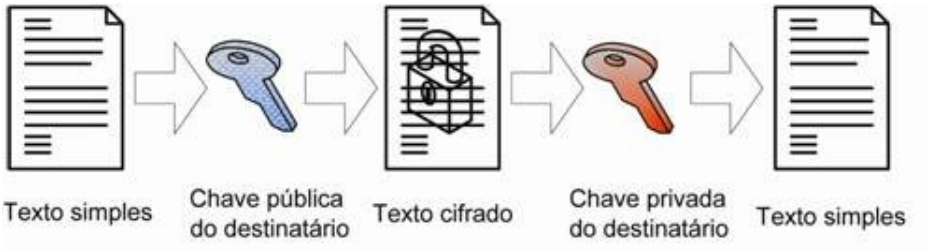

Figura 4.4: Processo de criptografia assimétrica

4.2.2 Algoritmos assimétricos ou de chave pública:

4.2.2.1 Algoritmo RSA

Um dos algoritmos mais seguros de encriptação de informações atuais originado dos estudos de Ronald Rivest, Adi Shamir e Leonard Adleman, um trio de matemáticos brilhantes que mudaram a história da criptografia. Sua patente expirou em 21 de setembro de 2000.

O princípio do algoritmo é construir chaves públicas e privadas utilizando números primos.

Uma chave é uma informação restrita que controla toda a operação dos algoritmos de criptografia. No processo de codificação, uma chave é quem dita a transformação do texto puro (original) em um texto criptografado.

Pré-Codificação

Para usarmos o método RSA, devemos converter uma mensagem em uma sequência de números. Essa etapa é chamada de pré-codificação.

Inicialmente, escolhemos uma chave boa e convertemos a mensagem, de letras para símbolos ou números. Então precisamos determinar 2 primos distintos, que nomearemos p e q; que são denominados parâmetros RSA. Seja n = pq; onde p e q são primos, n é chamado de modulo RSA. A ultima etapa da pré-codificação consiste em separar o número, encontrado a partir da conversão da mensagem em números, em blocos cujos valores sejam menores que n.

A maneira de escolher os blocos não e única e não precisa ser homogênea (todos os blocos com o mesmo número de dígitos), mas não podemos, por exemplo, começar um bloco com zero, pois isto traria problemas na hora de montar a seqüência recebida (ASSIS, 2003).

4.2.2.2 *ElGamal*

O algoritmo de criptografia ElGamal foi imaginado pela dificuldade em se calcular logaritmos discretos[1]. O cálculo da exponenciação é simples, com o algoritmo *Square-and-Multiply*, porém não existe algoritmo de tempo polinomial que calcule o logaritmo discreto no caso geral. Por isso, ElGamal se utilizou disso e propôs esse algoritmo, que não foi bem aceito por não ser uma função bijetora[2], ou seja, a decriptação pode gerar resultados diferentes (ASSIS, 2003).

4.2.2.3 *Diffie-Hellman*

Diffie-Hellman é um algoritmo de troca de chaves que pode usar tamanhos de chaves variáveis (teoricamente ilimitados).

Esse algoritmo foi criado por Whitfield Diffie e Martin Hellman em 1976. Ele usa a exponenciação e a aritmética modular como base de seus cálculos; isso é bastante seguro, mas envolve números muito grandes e cálculos relativamente lentos. Uma das características mais importantes do *Diffie-Helman* é que ele pode ser usado para gerar um segredo que tem encaminhamento de segredo perfeito.

[1] Um logaritmo discreto é uma noção relacionada na teoria finita de grupos. Para alguns grupos finitos, acredita-se que logaritmo discreto seja muito difícil de ser calculado, enquanto exponenciais discretas são bem fáceis.

[2] É uma função injetora e sobrejetora.

Diffie-Helman foi patenteado, mas a patente expirou em 1997, e assim, o algoritmo pode ser usado livremente (ZWICKY, 2000).

4.2.3 Funções de *hash*

Uma função de *hash* é uma equação matemática que utiliza uma palavra ou texto para criar um código chamado message digest (resumo de mensagem). Especificamente, deve ser impraticável encontrar:

- Texto que dá um *hash* a um dado valor. Ou seja, mesmo que você conheça o message digest, não conseguirá decifrar a mensagem .

- Duas mensagens distintas que dão um hash ao mesmo valor.

A capacidade de descobrir uma mensagem que dê um *hash* a um dado valor possibilita a um agressor substituir uma mensagem falsa por uma mensagem real que foi assinada. Permite ainda que alguém rejeite de forma desleal uma mensagem, alegando que, na realidade, ele ou ela assinou uma mensagem diferente, dando um hash ao mesmo valor e violando assim a propriedade de não-repúdio das assinaturas digitais. A capacidade de descobrir duas mensagens distintas que dêem um *hash* ao mesmo valor possibilita um tipo de ataque no qual alguém é induzido a assinar uma mensagem que dá um hash ao mesmo valor como sendo outra mensagem com um conteúdo totalmente diferente.

4.2.4 Assinaturas digitais

Uma assinatura digital é algo equivalente a uma assinatura normal em papel, uma vez que possui características únicas que identificam o indivíduo que a gerou no documento a que foi proposto assinar.

Na forma convencional de assinatura, as técnicas utilizadas para falsificar assinaturas ainda são muito primárias, baseando-se em cores de tinta de caneta ou qualidades de papel que dificultam a fotocópia. Nenhum destes sistemas é à prova de falhas.

Os algoritmos de assinatura digital fornecem mecanismos muito mais resistentes a falsificações. Enquanto a tecnologia de assinatura digital combina a criptografia de chave pública e o *hash* criptográfico, a criptografia de chave pública oferece um meio para você provar sua identidade e o *hash* criptográfico fornece

um caminho para garantir que as informações às quais você associou sua identidade não foram modificadas (ZWICKY, 2000).

4.2.5 Certificado Digital

O certificado digital é composto pela assinatura digital, chave pública, alguns valores, atributos e uma data de vencimento. Alguns valores são utilizados para a aplicação e outros para a pessoa que emitiu, dentre os quais podemos exemplificar telefone, endereço, empresa, setor, entre outros. Um certificado digital deve ser assinado por uma autoridade certificadora que será o vínculo de confiança entre o emissor e o recebedor do certificado. A autoridade certificadora poderia confirmar a autenticidade do certificado ou revogar quando necessário (ZWICKY, 2000).

4.3 Tipos de ataques:

Criptoanálise (Cryptanalysis): A ciência – ou a arte – de ler o tráfego criptografado sem conhecimento prévio da chave.

Força bruta (Brute force): Experimentando cada chave possível.

Repetição (Replay): Nesses ataques selecionam uma mensagem legítima e a injetam novamente na rede mais tarde.

Espionagem passiva (Passive eavesdropping): Um invasor passivo simplesmente ouve o fluxo do tráfego.

Ataque ativo (Active attack): Em um ataque ativo, o inimigo pode inserir mensagens e – em algumas variantes – excluir ou modificar mensagens legítimas.

Homem-no-meio (Man-in-the-middle): O inimigo permanece entre você e a parte com quem você deseja se comunicar, bem como personifica cada um de vocês para a outra parte.

Recortar e colar (Cut-and-paste): Dadas duas mensagens criptografadas com a mesma chave, às vezes é possível combinar partes de duas ou mais mensagens para produzir uma nova mensagem. Você talvez não saiba o que ela informa, mas pode utilizá-la para enganar seu inimigo, para que ele faça algo para você.

Redefinição de data/hora (Time-reseting): Nos protocolos que utilizam a data/hora atual, esse ataque tentará confundi-lo com relação a data/hora correta.

Ataque de aniversário (Birthday attack): Um ataque nas funções de *hash*, em que o objetivo é descobrir duas mensagens quaisquer que reproduzam o mesmo valor. Se uma pesquisa exaustiva demorasse 2^n passos, um ataque de aniversário levaria somente $2^n/2$ tentativas.

Ataques contra o oráculo (Oracle attack): Um invasor pode conseguir algum benefício enviando consultas a uma das partes, utilizando os participantes do protocolo como oráculos (CHESWICK, 2003).

Capítulo 5

Tunelamento

O tunelamento é definido como "um conceito arquitetônico no qual uma ou mais camadas de protocolo são repetidas, de maneira que uma topologia virtual é criada sobre a topologia física". Túneis na maioria das vezes são seguros, protegidos por criptografia. Existem túneis bons, que podem auxiliar na criação de um "canal"seguro entre duas pontas e existem túneis maus, que dificultam ainda mais a administração de uma rede corporativa. Protocolos de túneis devem ser documentados e usar portas específicas (CHESWICK, 2003).

Quando um túnel utiliza uma porta de um determinado serviço, como por exemplo a porta 80, serviço http[1], porta que é permitida em muitos *firewalls*, ele força uma comunicação "invisível", dificultando o controle do tráfego da rede, limitando a utilidade dos *firewalls*, como é o caso do *Simple Object Access Protocol* (SOAP) da Microsoft, sendo este um exemplo de um tunel mau. Nós concentraremos neste trabalho em falar de túneis bons e como estes podem auxiliar o desenvolvimento de sistemas seguros.

Na sequência, serão apresentados alguns protocolos utilizados em VPNs para um embasamento teórico no assunto.

[1] HTTP (acrônimo para *Hypertext Transfer Protocol*, que significa Protocolo de Transferência de Hipertexto) é um protocolo de comunicação (na camada de aplicação) utilizado para transferir dados por *intranets* e pela *World Wide Web*

5.1 Protocolos de tunelamento

5.1.1 Protocolo PPTP (*Point to Point Tunneling Protocol*)

O protocolo PPTP permite o estabelecimento de uma conexão VPN entre o cliente e o servidor de forma transparente ao provedor de acesso.

Funcionamento

Inicialmente, o cliente utiliza o protocolo PPP[2] (*Point to Point Protocol*) para estabelecer a conexão com o servidor. Uma vez autenticado e conectado, os quadros recebem um cabeçalho GRE[3] (*Generic Routing Encapsulation*) para transporte dos dados. Após esta etapa, é criada uma conexão de controle entre cliente e servidor, utilizando o protocolo TCP por onde são estabelecidos os parâmetros de configuração entre os extremos do tunel. Desta forma, é criado o tunel PPTP. A estrutura de um quadro de tunelamento PPTP pode ser observada na Figura 5.1.

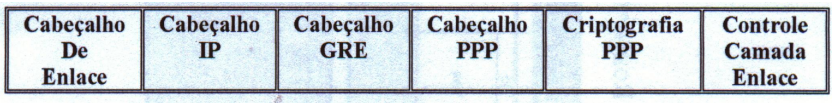

Figura 5.1: Quadro de encapsulamento PPTP

5.1.2 L2F (*Layer 2 Forwarding Protocol*)

O L2F (*Layer 2 Forwarding Protocol*) protocolo desenvolvido pela CISCO [4] como um dos primeiros protocolos de tunelamento, possui tunelamento independente do protocolo IP, o que permite que ele possa trabalhar com redes ATM e Frame-Relay.

Não suporta criptografia, fato este que torna os dados que são trafegados pela rede utilizando este protocolo vulneráveis (BALLESTEROS, 2008).

[2]Protocolo que permite a criação de uma conexão entre dois dispositivos de rede.

[3]Protocolo que encapsula uma ampla variedade de protocolos dentro de túneis IP, criando um link virtual entre roteadores e pontos remotos sobre redes IP.

[4]Companhia na área de telecomunicações

5.1.3 L2TP (*Layer 2 Tunneling Protocol*)

É um protocolo desenvolvido pela IETF [5](*Internet Engineering Task Force*) unindo as melhores práticas do PPTP e do L2F.

Apresenta seu próprio encapsulamento de pacotes, é muito flexível pode ser utilizado em redes não IP como *Frame Relay* e ATM. O L2TP é também um protocolo escalável, suportando múltiplas sessões por um mesmo túnel. Porém é um protocolo que não suporta criptografia e tampouco gerenciamento de chaves de criptografia, tornando as VPNs que utilizam o protocolo L2TP inseguras, portanto para uma utilização segura do mesmo é necessário que se utilize juntamente com o protocolo IPSec (BALLESTEROS, 2008).

5.1.4 Protocolo IPsec(*IP Security*)

Implementado na camada de rede, o IPsec é um protocolo que fornece segurança baseada em criptografia de um extremo ao outro. Sendo um requisito para o IPv6, o IPsec pode ser implementado como opção no IPv4. O IPsec pode fornecer segurança para qualquer protocolo na camada de rede.

Os serviços de segurança que o IPsec fornece são:

- Controle de acesso;
- Autenticação de origem de dados;
- Integridade da mensagem;
- Proteção de retransmissão;
- Confidencialidade.

Os protocolos que formam o IPsec são:

O AH (*Authetication Header* - Cabeçalho de autenticação) proporciona integridade de mensagens e autenticação de origem de dados.

O ESP (*Encapsulating Security Payload*) proporciona confidencialidade e proteção contra a análise do fluxo do tráfego.

[5]Uma comunidade internacional ampla e aberta (técnicos, agências, fabricantes, fornecedores, pesquisadores) preocupada com a evolução da arquitetura da Internet e seu perfeito funcionamento.

O ISAKMP (*Internet Security Association and Key Management Protocol*), protocolo responsável pela gestão de chaves, isto é, pela criação, eliminação e alteração das chaves.

O IPsec pode operar de dois modos, quais sejam, o modo transporte, onde a autenticação e criptografia apenas ocorre na camada de transporte e o modo túnel, no qual a criptografia e autenticação ocorrem em todo o pacote IP, ficando visível apenas o cabeçalho IP externo, com o último endereço de origem e destino, todo o conteúdo interno estando cifrado (ZWICKY, 2000).

5.1.5 *Secure Sockets Layer (SSL)/ Transport Layer Security (TLS)-Based Solution*

Atuando entre as camadas de transporte e aplicação (TCP), o protocolo SSL/TLS provê uma conexão segura com três propriedades básicas: privacidade, autenticidade e confiabilidade, sendo que o SSL é um protocolo proprietário da netscape, enquanto o TLS é um protocolo desenvolvido a partir do SSL, padronizado pelo IETF (*Internet Engineering Task Force*).

Para garantir que a conexão seja privada, durante a negociação inicial (*Handshake*) é utilizada a criptografia simétrica, tal como DES, DES-3, RC4, etc, garantindo a segurança das aplicações.

Assegurando a identidade do cliente e do servidor, pode ser utilizada autenticação com criptografia assimétrica, como RSA, DSS, etc.

Para manter a conexão confiável, existe um mecanismo de checagem de integridade de mensagem usando MAC (*Message Authentication Code* - Código d Autenticação de Mensagem), com as funções hash, SHA e MDS (ZWICKY, 2000).

Capítulo 6

VPN (*Virtual Private Network*)

As LANs (*Local Area Networks*) ficaram muito limitadas para as novas demandas das instituições. Com isso, surgiu a necessidade de interligar as redes, uma vez que era interessante que serviços específicos ficassem centralizados em apenas um local, muitas instituições possuíam conexão com a rede pública (*internet*) e poderiam utiliza-la para conectarem seus serviços, porém as conexões estariam abertas e facilmente teriam suas informações interceptadas por alguem. A partir dessa demanda, surgiu a VPN, como uma forma de utilizar a rede pública como se ela fosse privada.

As VPNs utilizam técnicas de *tunneling*[1] para permitir o acesso entre as redes e de criptografia para garantir a privacidade destas conexões, possibilitando que mesmo interceptadas, as informações coletadas não teriam utilidade.

Alguns dos dispositivos que implementam uma VPN são: roteadores, equipamentos específicos e *softwares* instalados em *gateways*[2].

As VPNs possuem seus próprios protocolos de comunicação que atuam em conjunto com o TCP/IP, fazendo com que o túnel virtual seja estabelecido e os dados trafeguem criptografados (CENTRO UNIVERSITáRIO DO PARá, 2002).

[1] A capacidade de criar túneis entre duas máquinas por onde certas informações passam.

[2] Um Gateway, ou porta de ligação, é uma máquina intermediária geralmente destinada a interligar redes, separar domínios de colisão, ou mesmo traduzir protocolos.

6.1 Topologias

6.1.1 Host-to-host

Tendo um *host* como um computador conectado à internet, esta topologia é formada pela interligação entre dois ou mais *hosts*, tendo o serviço de VPN provido por um *gateway* intermediário entre as redes em que estão subordinados. Um exemplo disso é um sistema cliente/servidor em que um cliente fica em uma rede e o servidor em outra, como mostrado na figura 6.1.

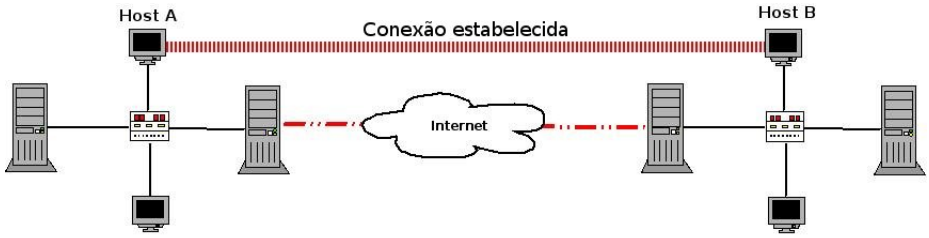

Figura 6.1: Topologia host to host

6.1.2 Host-Rede

Esta topologia permite que um *host* tenha acesso à rede da instituição através da *internet*. Essa VPN, necessariamente, é criada pelo *gateway* de borda da instituição que limitará se o *host* deverá estar restrito a serviços específicos da rede ou terá acesso a toda rede, como podemos observar na figura 6.2.

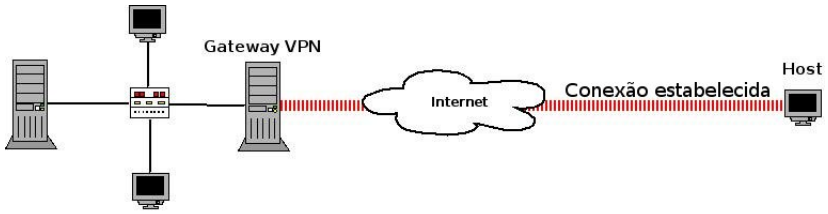

Figura 6.2: Topologia host-rede

6.1.3 *Gateway-to-gateway*

Nesta topologia, duas ou mais redes ficam interligadas. Essa é a implementação mais difícil de ser feita, uma vez que exige uma grande quantidade de recursos como largura de banda de internet e uma equipe maior de segurança, já que existirá uma expansão do perímetro[3] da rede, como podemos observar na figura 6.2 o princípio de seu funcionamento.

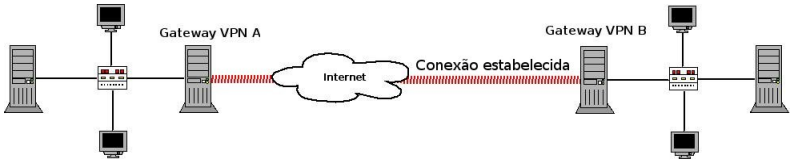

Figura 6.3: Topologia geteway-to-gateway

6.2 Vantagens do uso de VPN

A maior vantagem da utilização da VPN - Rede Privada Virtual, para interligar redes é o custo. É mais econômico utilizar a rede pública, a internet, em relação a manter um circuito dedicado ou um *pool de modems*[4].

Uma rede privada virtual oferece criptografia em todos os pontos da conexão, possibilitando que todo o tráfego da rede se torne incompreensível para alguém mal intencionado que deseje interceptar e obter alguma informação através daquele tráfego de rede, independente do protocolo, seguro ou não, que será utilizado sobre ele (ZWICKY, 2000).

6.3 Desvantagens de redes privadas virtuais

As redes privadas virtuais envolvem conexões perigosas. Por exemplo, se um usuário está utilizando uma rede virtual privada para ter acesso à internet, este pode ser invadido por alguem que esteja na ponta da conexão. Outro exemplo é que uma vez expandida a rede para outras redes, alguém mal intencionado em alguma destas redes pode ganhar acesso à rede central e ter acesso a informações que não

[3] Limite de abrangência da rede

[4] Um pool de modems é um grupo de modems que são usados para receber ligações.

deveria, exigindo-se, neste caso, que seja revista toda a segurança de perímetro (ZWICKY, 2000).

6.4 Alternativas ao serviço VPN

6.4.1 Frame-Relay

O *Frame-Relay* é um serviço que permite a ligação entre duas redes distintas independente da localização geográfica, sendo um serviço contratado de uma operadora de telecomunicação que será responsável pela criação deste circuito.

O *Frame-Relay* pode funcionar sobre um PVC (*Permanent Virtual Circuit* - Circuito Virtual Permanente), ou sobre um SVC (*Switched Virtual Connections* - Conexão Virtual Comutada). Abordaremos neste trabalho apenas a baseada em PVC por ser a mais utilizada.

O *Frame-Relay* é um protocolo que funciona nas camadas 1 e 2 do modelo TCP, ou seja, qualquer informação que for enviada na camada 3 não será tratada por este serviço. Referido serviço permite a comunicação entre DTEs[5] *Data Terminal Equipments* e DCE[6] *Data Circuit Terminating Equipment*. Ele provê uma comunicação orientada à conexão em nível de enlace, com a criação de um PVC (*Permanent Virtual Circuit* - Circuito Virtual Permanente). Os PVCs são circuitos lógicos criados entre dois DTEs, sendo que cada PVC possui DLCI (*Data Link Connection Identifier*), que é o identificador do circuito virtual. Como princípio básico de funcionamento, temos que o caminho completo até o destino é estabelecido antes de qualquer transmissão de informação (FILIPPETTI, 2008).

6.4.2 Redes ATM (*Asynchronous Transfer Mode*)

Com o objetivo de desenvolver um protocolo que permitisse a transmissão de voz, vídeo, e-mail e texto em tempo real, dois grupos, o ATM Forum (ATM FORUM, 2004) e a União Internacional de Telecomunicações (ITU, 2004), envolveram-se no desenvolvimento dos padrões para redes ATM. Eles definiram desde as especificações de camada de aplicação para ATM até o enquadramento de dados ATM no nível de *bit* por várias camadas físicas, de fibra, de cobre e de rádio.

[5] Terminais definidos pelo usuários.

[6] Equipamentos responsáveis por receber e transmitir dados pelos meios de transmissão

Principais características das redes ATM

O ATM suporta diversos modelos de serviço, sendo com taxa constante de *bits*, taxa variável de bits, serviço de taxa disponível e serviço com taxa não especificada de *bits*. A arquitetura ATM é dividida em 3 camadas, conforme se vrifica na figura 6.4.

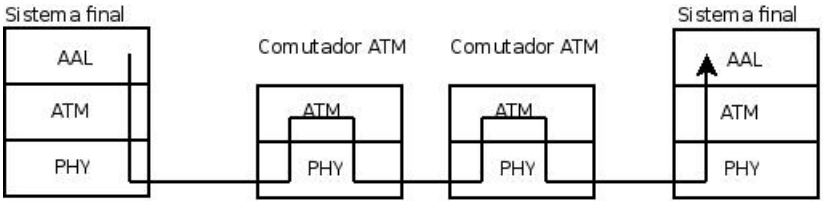

Figura 6.4: As três camadas ATM

As três camadas ATM

A camada de adaptação ATM (AAL - *ATM Adaptation Layer*) está presente apenas nos dispositivos de borda da rede ATM. Tal camada tem as funções análogas à camada de transporte do modelo TCP e esta presente apenas na redes ATM.

No lado remetente, a AAL recebe dados passados por uma aplicação ou protocolo de nível mais alto (tal como IP, se a rede ATM estiver sendo utilizada para conectar dispositivos IP). Já no lado receptor, ela passa dados para o protocolo ou aplicação de camada mais alta.

A camada ATM está no centro da estrutura ATM e define a estrutura da célula ATM, que é tão importante para as redes ATM quanto o datagrama para a rede IP. Essa camada define o significado das células ATM e a estrutura dos campos das células. Os primeiros 5 *bytes* da célula constituem o cabeçalho ATM, enquanto os demais 48 *bytes* são a carga útil ATM.

A camada física ATM, localizada no nível mais baixo das camadas ATM, é responsável pelas tensões, temporização de *bits* e enquadramento do meio físico.

6.4.3 VPN x Circuitos dedicados (*Frame-Relay ou ATM*)

A utilização de um circuito dedicado possibilitaria um desempenho interessante, uma vez que o circuito seria direcionado apenas para aquela aplicação, porém o custo de manutenção de um circuito destes é bem alto.

O fator custo levou à escolha da VPN como solução para conexão entre as redes, uma vez que teríamos uma economia por não contratarmos outro serviço de telecomunicação, posto que a implementação de VPN escolhida funciona sobre a infra-estrutura atual de internet de todos os *campi*.

Capítulo 7

Implementando a VPN com o OpenVPN

7.1 Motivação para a utilização da solução OpenVPN

O IPsec é um padrão de mercado para a utilização de VPNs e está disponível em muitas plataformas de *hardware* e *software*, tendo sido por esse motivo cogitada a possibilidade de uso de IPsec como solução, porém em muitas das redes da instituição não existem IPs fixos, devido à utilização de serviços de ADSL[1](*Asymmetric Digital Subscriber Line*). Outra característica que inviabilizou a utilização do IPsec foi o fato de as conexões serem estabelecidas por máquinas que têm acesso à internet através de NAT[2](*Network Address Translation*). As características anteriormente citadas levaram à escolha do OpenVPN como solução de VPN para a Unimontes (FEILNER, 2006).

[1] É um formato de DSL, uma tecnologia de comunicação de dados que permite uma transmissão de dados mais rápida através de linhas de telefone do que um modem convencional pode oferecer, serviço fornecido por operadoras a um baixo custo.

[2] também conhecido como masquerading é uma técnica que consiste em reescrever os endereços IP de origem de um pacote que passam por um router ou firewall de maneira que um computador de uma rede interna tenha acesso ao exterior (rede pública).

7.2 A ferramenta OpenVPN

O OpenVPN é uma solução VPN que implementa suas conexões nas camadas 2 e 3 e usa o padrão SSL/TLS para encriptação(FEILNER, 2006). A aplicação OpenVPN é um projeto *Open Source* e licenciado sob a GPL (*General Public License*).

7.3 A instalação da solução

Para instalar o aplicativo OpenVPN, tomando como referência a distribuição debian, deve-se seguir as seguintes etapas:

1. Instalar os pacotes openvpn e openssl:

```
apt-get install openvpn
apt-get install openssl
```

2. Para criar os certificados e as chaves:

Entrar no diretório /usr/lib/ssl, e editar o arquivo openssl.cnf alterando os dados da CA, colocando as informações necessárias, como mostrado na Figura 7.1.

3. Para alterar o tempo de duração da CA:

Entrar no diretório misc:

```
cd misc
```

Edite o arquivo CA.pl, conforme mostrado na Figura 7.2 e altere o tempo de duração do certificado da CA.

4. Para criar a CA e as chaves da mesma:

No diretório /usr/lib/ssl/misc,

criando a CA:

```
./CA.pl - newca
```

Seguindo os passos anteriores a CA foi criada dentro do diretório .demoCA.

criando as chaves:

```
./CA.pl - newreq
```

```
countryName                     = {\it{Country Name (2 letter code)}}
countryName_default             = BR
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = {\it{State or Province Name (full name)}}
stateOrProvinceName_default     = MG

localityName                    = {\it{Locality Name (eg, city)}}
localityName_default            = Montes Claros

0.organizationName              = {\it{Organization Name (eg, company)}}
0.organizationName_default      = Unimontes

organizationalUnitName          = {\it{Organizational Unit Name (eg, section)}}
organizationalUnitName_default  = Gerencia de Tecnologia da Informacao

commonName                      = {\it{Common Name (eg, YOUR name)}}
commonName_max                  = 64
commonName_default              = Autoridade Certificadora da Unimontes

emailAddress                    = {\it{Email Address}}
emailAddress_max                = 64
emailAddress_default            = ca@unimontes.br
```

Figura 7.1: Arquivo de configuração openssl.cnf

```
\$CADAYS="-days 3650";

#O nosso certificado da CA esta configurado para durar 10 anos.
#Edite o arquivo CA.pl e altere o tempo de duracao do certificado comum.

\$DAYS="-days 365";

#O nosso certificado comum esta configurado para durar 1 ano.
```

Figura 7.2: Arquivo de configuração CA.pl

Com as informações cadastradas, serão criados os seguintes arquivos:

• newkey.pem: contém a chave privada.

• newreq.pem: contém a chave pública.

5. A assinatura de um certificado pela CA:

Dentro do diretório /usr/lib/ssl/misc, com o arquivo newreq.pem presente, para assinar um certificado, executar o comando:

```
./CA.pl - sign
```

Será pedida a *passfrase*(senha no formato de frase) da autoridade certificadora para que seja feita a assinatura. Será criado o arquivo newcert.pem, com a chave pública assinada.

7.3.1 Certificado Raiz

Será disponibilizado o certificado raiz, que é o certificado da CA, pois os clientes precisarão instalar os certificados em seus clientes VPN. O certificado raiz é o arquivo /usr/lib/ssl/misc/demoCA/cacert.pem. Renomeie-o para cliente01.crt e envie para o cliente da VPN.

Possibilitando o uso da chave sem senha

```
openssl rsa -in newkey.pem -out key.pem
```

A *passfrase* da chave privada será solicitada. O arquivo key.pem que será gerado conterá a chave privada sem a senha.

Configuração

Renomear as chaves para melhorar a identificação das mesmas. Levando em conta que o nome do gateway seja unimontes.

```
mv newcert.pem unimontes.crt
mv key.pem unimontes.key
```

Renomear uma cópia do certificado raiz da CA que assinou a chave pública, geralmente o nome desse certificado é cacert.pem, renomear uma cópia, preservando o original.

```
mv cacert.pem rede.crt
```

Será necessário uma terceira chave. Essa só existirá do lado do *gateway* e será utilizada para troca inicial de informações, quando ainda estará sendo fechado o canal criptográfico. Vamos gerar uma chave de 1024 *bits* de tamanho, com o seguinte comando:

```
openssl dhparam -out janauba.dh 1024 (janauba.dh sendo o nome do cliente)
```

Para criar mais uma chave para ser utilizada como segurança adicional no TLS. Essa chave existirá tanto no cliente quanto no *gateway*. Utilize o comando:

```
openvpn --genkey --secret unimontes.tlskey
```

Instalação dos certificados e das chaves

Criados os certificados, vamos colocá-los nos diretórios corretos dentro do *gateway*.

```
rede.crt (copia de cacert.crt) ---> /etc/ssl/certs/
unimontes.crt (antigo newcert.pem) ---> /etc/ssl/certs/
unimontes.key (antiga key.pem) ---> /etc/ssl/private/
unimontes.dh (chave Diffie-Hellman) ---> /etc/ssl/private/
unimontes.tlskey (seguranca TLS) ---> /etc/ssl/private/
```

A seguir, altere as permissões dos arquivos gerados para ter mais segurança:

```
chmod 644 /etc/ssl/certs/rede.crt
chmod 644 /etc/ssl/certs/unimontes.crt
chmod 400 /etc/ssl/private/unimontes*
```

7.3.2 A configuração do OpenVPN e o *Firewall*

Por questões de segurança não será colocada a versão original da configuração da instituição, usando no lugar desta a configuração recomendada pelo OpenVPN Howto.

Em ambos os lados cliente e servidor:

Abrir a porta UDP 1194 e permitir a utilização de pacotes de dispositivos TUN/TAP:

As regras do IPtables da Figura 7.3 podem ser obtidas em: http://svn.openvpn.net/projects/openvpn/trunk/openvpn/sample-config-files/firewall.sh

7.3.3 Arquivo de configuração do gateway

O arquivo /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz é um exemplo, comentado, de arquivo de configuração do OpenVPN. É com base nele que deverá ser gerado o arquivo de configuração da vpn no gateway (em /etc/openvpn/).

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT

iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT

# Allow packets from private subnets
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT

#Keep state of connections from local machine and private subnets
iptables -A OUTPUT -m state --state NEW -o eth0 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state NEW -o eth0 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

#Masquerade local subnet
iptables -t nat -A POSTROUTING -s \$PRIVATE -o eth0 -j MASQUERADE
```

Figura 7.3: Arquivo de configuração do firewall

Na Figura 7.4, um exemplo de arquivo de configuração do gateway:

Explicando o arquivo de configuração acima:

• port: uma porta escolhida para servir a conexão.

• local: contém o IP do gateway na internet.

• dev: contêm o canal que será usado no tunelamento.

• proto: permite escolher o protocolo de transporte a ser utilizado.

• crl-verify: arquivo com certificados revogados ou que não estão mais em uso.

• ifconfig-pool-persist: opção que faz com que o OpenVPN passe a armazenar uma lista dos endereços IP usados por cada cliente da VPN e faça o possível para atribuir sempre os mesmos endereços em futuras conexões dos clientes.

• cipher: escolha do método de encriptação.

• client-config-dir: rota inválida para clientes não autorizados.

Inicialização do serviço

```
port 1194
proto tcp-server
dev tun0
ca keys/unimontes/rede.crt
cert keys/unimontes/unimontes.crt
key keys/unimontes/unimontes.key
dh keys/unimontes/unimontes.dh
server 10.254.254.0 255.255.255.0
crl-verify keys/unimontes/crl.pem
ifconfig-pool-persist servers/server/logs/ipp.txt
cipher BF-CBC
user nobody
group nogroup
status servers/server/logs/openvpn-status.log
log-append servers/server/logs/openvpn.log
verb 2
mute 20
max-clients 100
management 127.0.0.1 9999
keepalive 10 120
client-config-dir /etc/openvpn/servers/server/ccd
tls-server
comp-lzo
persist-key
persist-tun
ccd-exclusive
push "route 10.0.0.0 255.255.0.0"
push "route 10.10.10.0 255.255.255.0"
push "route 192.168.0.0 255.255.0.0"
push "dhcp-option WINS 10.0.0.27"
```

Figura 7.4: Arquivo de configuração do gateway

Para inicializar o serviço usaremos o comando /etc/init.d/openvpn start.

7.3.4 Configuração dos clientes OpenVPN em máquinas Windows XP

Instalação do cliente do OpenVPN

O projeto OpenVPN disponibiliza para download o OpenPVN GUI, que deve ser instalado e deve ser feita a instalação padrão, sem remover ou acrescentar componentes.

Os arquivos deverão ficar por padrão no diretório:

```
C:\Arquivos de Programas\OpenVPN\config\
```

Os arquivos de configuração do cliente são os da Figura 7.5.

```
cliente01.ovpn
rede.crt
unimontes.dh
cliente01.crt
cliente01.key
```

Figura 7.5: Arquivos de configuração do cliente

No arquivo cliente01.ovpn, mostrado na Figura 7.6 fica a configuração do cliente, o seu conteúdo é o seguinte:

```
client
proto tcp-client
dev tun
ca rede.crt
dh unimontes.pem
cert cliente01.crt
key cliente01.key
remote 222.222.222.222 1194
cipher BF-CBC
verb 2
mute 20
keepalive 10 120
comp-lzo
persist-key
persist-tun
float
resolv-retry infinite
nobind
```

Figura 7.6: Arquivo cliente01.ovpn

Para iniciar o OpenVPN, deve-se clicar com o botão direito do mouse sobre o ícone e escolha a opção “connect”.

7.4 Resultados do projeto:

Neste primeiro momento foi instalado com êxito a VPN funcionando com os sistemas do estado em Janaúba e o sistema da secretaria geral. O sistema do estado

esta funcionando corretamente, com uma velocidade bem razoável, já o sistema da secretaria apresentou certa lentidão devido às características do mesmo, uma vez que este faz consultas muito extensas no banco de dados, fazendo com que o mesmo apresente até então baixo desempenho, a situação esta sendo estudada pela empresa desenvolvedora do sistema. A Figura 7.7 mostra o sistema da secretaria funcionando com a VPN.

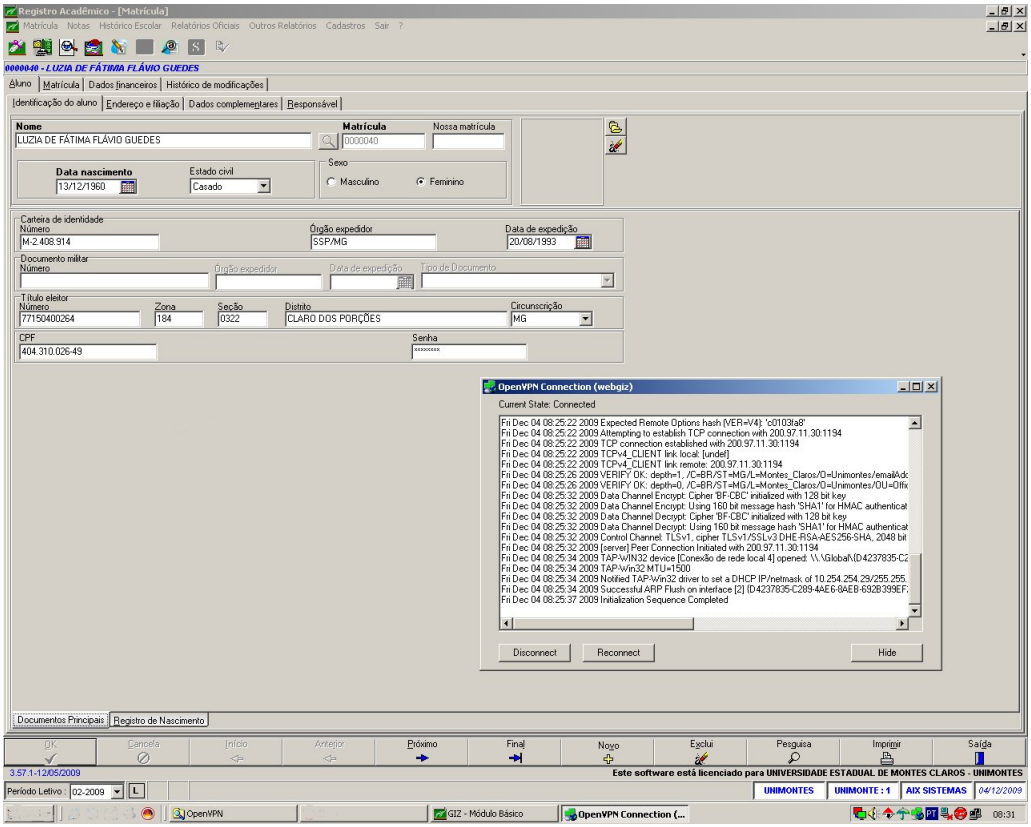

Figura 7.7: Sistema da secretaria funcionando com a VPN

Capítulo 8

Conclusão

A rede da Unimontes não pode mais se restringir à rede local no Campus sede, uma vez que se faz necessária a integração dos vários serviços dependentes de informática com todos os Campi da instituição.

A escolha da solução para a instituição foi feita levando em conta aspectos de segurança, custo e o tempo de implementação. Os aspectos da segurança que foram levados em conta foram o perímetro da rede e a privacidade das informações. O custo estimado mensal para a instalação de um circuito dedicado em cada Campus seria de R$900,00, sendo que seriam necessários 9 circuitos, porém conseguiu-se a referida instalação apenas com o custo das horas de projeto e implementação do serviço da equipe para a implantação do serviço da VPN. A universidade estava precisando do serviço para poder cumprir metas com o Estado e com isso pleitear mais recursos além de ter uma administração mais esclarecida na sua prestação de contas.

A utilização da internet trouxe a possibilidade de usufruir dos serviços que alternativamente seria necessária a contratação de circuitos virtuais pelas operadoras, tendo uma economia considerável para a instituição, possibilitando a contratação de melhores serviços para o acesso à internet para os Campi da Universidade.

Com a VPN foi possível suprir as demandas existentes da Universidade, integrando as secretarias e possibilitando que os sistemas do Estado funcionem em vários Campi, permitindo que a rede seja flexível o suficiente para acompanhar o crescimento da instituição, de forma segura e gerenciável.

Referências Bibliográficas

APPLIED Cryptography: Protocols, Algorithms, and Source Code in C. 2. ed. [S.l.]: John Wiley & Sons, 1996.

ASSIS, J. M. de. *Implementando VPN em Linux - Monografia*. Pós-Graduação em Administração em Redes Linux, 2003. Disponível em: <http://www.ginux.ufla-.br/files/mono-JoaoAssis.pdf>.

ATM FORUM. *ATM Forum*. 2004. Disponível em: <http://www.atmforum-.com>.

BALLESTEROS, H. M. S. *VPN (Virtual Private Network)*. Mestrado e Doutorado em Redes de Computadores, 2008. Disponível em: <http://www.gta.ufrj.br/grad-/08_1/vpn/tiposenlace.html>.

ZWICKY, D. Elizabeth, Chapman, D. Brent, S. Cooper *BUILDING Internet Firewalls*. 2. ed. [S.l.]: O'Reilly and Associates, 2000.

CENTRO UNIVERSITáRIO DO PARá. *Implementação de uma VPN em Linux utilizando o protocolo IPSec - Monografia*. Curso de bacharelado em Ciência da Computação, 2002. Disponível em: <http://www.abusar.org/manuais/VPN-alan-rafael.pdf>.

FERNANDES, J. *Criptografia e modelo criptográfico do sistema informatizado de eleiçoes do Brasil - Monografia*. 2007. Disponível em: <www.esab.edu.br-/arquivos/monografias/TccRedesJF.pdf>.

FILIPPETTI, M. A. *CISCO CCNA 4.1 (Exame 640-802): Guia completo de estudo*. 3. ed. [S.l.]: Visual Books, 2008.

CHESWICK William R.; BELLOVIN, Steven M.; RUBIN, Aviel D. *FIREWALLS e segurança na Internet*. 2. ed. [S.l.]: Bookman, 2003.

HINZ, M. A. M. *Um estudo descritivo de novos algoritmos de criptografia - Monografia*. 2000. Disponível em: <http://www.ufpel.tche.br/prg/sisbi/bibct-/acervo/info/2000/Mono-MarcoAntonio.pdf>.

ITU. *Iternational Telecommunications Union*. 2004. Disponível em: <www.itu.int>.

KUROSE, J. F. *Redes de computadores e a Internet*. 3. ed. [S.l.]: Addison Wesley, 2006.

MICROSOFT. *Aumentando a segurança dos dados com o SQL Server*. 2005. Disponível em: <http://www.microsoft.com/brasil/msdn/Tecnologias-/arquitetura/SegurancaDadosSQLServer2005.mspx?mfr=true>.

FEILNER, Markus. *OPENVPN Building and Integrating Virtual Private Networks*. 1. ed. [S.l.]: Packt, 2006.

SOARES G. LEMOS, S. C. *Redes de Computadores: Das LANs, MANs e WANs as Redes ATM*. [S.l.]: CAMPUS, 1995.