UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE PEREIRA DO CARMO

SOLUÇÃO SEGURA PARA UTILIZAÇÃO DE VPN BASEADA EM IP'S DINÂMICOS

VITÓRIA
2010

ALEXANDRE PEREIRA DO CARMO

SOLUÇÃO SEGURA PARA UTILIZAÇÃO DE VPN BASEADA EM IP'S DINÂMICOS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.
Orientador: Prof. Dr. Anilton Salles Garcia.

VITÓRIA
2010

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Carmo, Alexandre Pereira do, 1973-

C287s Solução segura para utilização de VPN baseada em IP's dinâmicos / Alexandre Pereira do Carmo. – 2010.

89 f. : il.

Orientador: Anilton Salles Garcia.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Extranets. 2. Redes de computadores. 3. Nomes de domínio na internet. 4. Sistemas de segurança. I. Garcia, Anilton Salles. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

ALEXANDRE PEREIRA DO CARMO

SOLUÇÃO SEGURA PARA UTILIZAÇÃO DE VPN BASEADA EM IP'S DINÂMICOS

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

Aprovada em 14 de junho de 2010.

COMISSÃO EXAMINADORA

_____

Prof. Dr. Anilton Salles Garcia - Orientador
Universidade Federal do Espírito Santo

_____

Prof. Dr. José Gonçalves Pereira Filho
Universidade Federal do Espírito Santo

_____

Prof. Dr. Carlos Alberto Malcher Bastos
Universidade Federal Fluminense

“A dificuldade na vida é tomarmos a sério
a mesma coisa durante tempo de mais”
(André Gide)

Dedicatória

À minha esposa Raquel e ao meu filho Pedro.

Os grandes amores da minha vida.

Agradecimentos

A Deus que, mesmo quando não me lembrei Dele, se fez presente em todos os momentos.

Aos meus pais José e Leoni e minhas irmãs Claudia e Chris pela formação, carinho e amizade.

Aos meus colegas da Unitera, Wilber Polonini, Rodrigo Bonfá Drago, Tiago Zanon, Tiago José Lima, Rodolfo da Silva Villaça, Adrian Bonfá Drago, que participaram ativamente deste trabalho.

Ao amigo e parceiro Maxwell Eduardo Monteiro pela força até o final da dissertação.

Ao meu orientador e amigo Anilton Salles Garcia que acreditou em mim e nesse trabalho, mais do que eu mesmo.

A minha esposa Raquel que esteve ao meu lado e me apoiou incondicionalmente, sem críticas ou cobranças, em todo este longo período do mestrado.

Ao meu filho Pedro que me fez perceber o que realmente é importante nesta vida.

Resumo

Redes privadas transportam informações de negócio importantes entre empresas, seus clientes e parceiros geograficamente distribuídos. Todos os usuários dessas redes privadas, não importando quão remoto estejam, esperam poder acessar dados e recursos da rede como se estivessem fisicamente em uma mesma rede local da empresa. Dessa forma, as redes privadas devem prover uma conectividade confiável para todos os usuários, e ao mesmo tempo devem proteger os dados e recursos privados da empresa. Um sistema de conectividade pode oferecer integridade, autenticidade e privacidade das informações trocadas entre empresas através do uso de VPN e segurança de acesso às redes através de sistemas de segurança de borda, como *Firewalls* e *Proxies*.

Em conjunto com essas tecnologias, este trabalho tem por objetivo desenvolver um sistema de conectividade de baixo custo baseado em endereçamento IP dinâmico utilizando a tecnologia ADSL como forma de acesso a à Internet. As VPNs utilizadas para interconexão dos sites possuem reconfiguração automática de seus parâmetros, ficando transparente para o usuário final a mudança de endereços por parte do provedor. O sistema também mantém um elevado nível de segurança em todos os elementos de sua arquitetura, eliminando vulnerabilidades existentes nas soluções adaptadas a este tipo de endereçamento. Além disso ele é capaz de interagir com outros equipamentos e sistemas utilizando padrões abertos de comunicação.

A abordagem proposta nesta dissertação foi implementada e testada em ambientes reais de empresas da Grande Vitória durante 4 anos. O sistema funcionou de maneira satisfatória interligando as redes de clientes de diversos ramos da economia, apresentando um alto índice de disponibilidade das conexões.

Palavras Chaves: Segurança, VPN, IP dinâmico, DYNDNS, DNSSEC

Abstract

Private networks allow important information to be shared between companies, their clients and partners located in different places or cities. All users, no matter how far they are, expect to access data and resources from the main company's network as they were physically connected to the same local net. Thus, private networks should provide a reliable and safe connection for all users, protecting the company's private data and resources. Data confidentiality, integrity, and authenticity may be get by using a VPN connection, besides applying Firewalls and Proxies to protect the whole system.

In order to solve these issues, this work proposes a new connectivity solution in a low cost, safe, efficient and flexible way. It is based on dynamic IP addressing and uses ADSL technology to access the Internet. The VPN connections have automatic reconfiguration of the parameters, so the changes on addresses made by the provider are not noticed by the final users. The framework also ensures a high safety level for the entire system, eliminating the vulnerabilities present in other solutions based on dynamic IP. Besides that, the developed framework can work with other equipments or systems by using open standards communication.

The proposed approach was implemented and tested in real networks of some companies in Vitória, during 4 years. The system performed very well connecting networks of different types of companies and clients, and achieving a high level of connection availability.

Keywords: Security, VPN, Dynamic IP, DYNDNS, DNSSEC

Sumário

Índice de Figuras

Índice de Tabelas

Nomenclatura

Siglas

Símbolo	Descrição
ADSL	*Asynchronous Digital Subscriber Loop*
AH	*Authentication Header*
ATM	*Asynchronous Transfer Mode*
CETIC.br	Centro de Estudos sobre as Tecnologias da Informação e da Comunicação
CPE	*Customer Premises Equipment*
DDNS	*Dynamic Domain Name System*
DHCP	*Dynamic Host Configuration Protocol*
DiffServ	*Differentiated Services*
DNS	*Domain Name System*
DNSSEC	*Domain Name System Security Extensions*
DoS	*Denied of Service*
ESP	*Encapsulating Security Payload*
FWA	*Fixed Wireless Access*
GRE	*Generic Routing Encapsulation*
HMAC-MD5	*Hash-based Message Authentication Code-Message-Digest Algorithm 5*
HTTP	*Hypertext Transfer Protocol*
IETF	*Internet Engineering Task Force*
IKE	*Internet Key Exchange*
IP	*Internet Protocol*
IPsec	*Internet Protocol security*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
ISP	*Internet Service Provider*
L2TP	*Layer 2 Tunneling Protocol*
MPLS	*Multiprotocol Label Switching*
NIC.br	Núcleo de Informação e Coordenação do Ponto br
NOC	*Network Operation Center*
OSI	*Open Systems Interconnection*
PC	*Personal Computer*
POP	*Point of Presence*
PPTP	*Point-to-Point Tunneling Protocol*

Siglas (continuação)

Símbolo	**Descrição**
RR	*Resource Records*
RRSet	*Resource Record Set*
SLA	*Service Level Agreements*
SOHO	*Small Office Home Office*
SSL	*Secure Sockets Layer*
TCP	*Transmission Control Protocol*
TI	Tecnologia da Informação
TLS	*Transport Layer Security*
TTL	*Time To Live*
UDP	*User Datagram Protocol*
UFES	Universidade Federal do Espírito Santo
VoIP	*Voice over Internet Protocol*
VPN	*Virtual Private Network*
VPNC	VPN Consortium
xDSL	*Digital Subscriber Loop family*

Capítulo 1: Introdução

Redes privadas transportam informações de negócio importantes entre empresas, seus clientes e parceiros geograficamente distribuídos. Todos os usuários dessas redes privadas, não importando quão remoto estejam, esperam poder acessar dados e recursos da rede como se estivessem fisicamente em uma mesma rede local da empresa. Dessa forma, as redes privadas devem prover uma conectividade confiável para todos os usuários, e ao mesmo tempo, devem proteger os dados e recursos privados da empresa. Um sistema de conectividade pode oferecer integridade, autenticidade e privacidade das informações trocadas entre empresas através do uso de VPN (*Virtual Private Network*) e segurança de acesso às redes através de sistemas de segurança de borda, como *Firewalls* e *Proxies*.

Hoje, as empresas podem criar VPNs para transportar dados privados entre redes geograficamente distribuídas através da infraestrutura da Internet. Provedores de serviço oferecem acesso rápido (alta velocidade de transmissão) à Internet através da tecnologia ADSL (*Asynchronous Digital Subscriber Loop*) e com um sistema de endereçamento IP (*Internet Protocol*), utilizando a infraestrutura de telefonia fixa. Essas facilidades possibilitam a transmissão de dados privados pela internet através de VPN *gateways*, que encapsulam os dados para manter a confidencialidade, integridade e autenticidade dos dados dentro do túnel VPN. Considerando que VPN *gateways* acessam a Internet através de uma linha telefônica via ADSL e que desempenham o papel de *firewall* de borda, pode-se implementar uma rede privada eficiente, mais segura e com um custo menor se comparado com a locação de canais dedicados.

Já existem no mercado soluções de hardware e software que implementam um ambiente de conectividade baseado em VPN *gateways*. Entretanto, tais soluções trabalham com sistema de endereçamento IP fixo para acesso à Internet, o que torna a solução mais cara. Algumas soluções podem ser adaptadas à utilização de endereçamento IP dinâmico, porém elas se mostram ineficientes, pouco práticas e com um baixo nível de segurança.

A proposta deste trabalho é desenvolver um sistema baseado em endereçamento IP dinâmico com reconfiguração automática das VPNs, ficando transparente para o usuário final a mudança de endereços por parte do provedor. O sistema também deve manter um elevado nível de segurança em todos os elementos de sua arquitetura, de forma a garantir a

disponibilidade, confidencialidade, integridade e autenticidade dos dados trafegados pela VPN.

1.1 Definição do Problema

Cada vez mais provedores de serviço oferecem acesso rápido a Internet através de tecnologias como o ADSL. Na maior parte do país, essa tecnologia é oferecida baseada em endereçamento IP dinâmico. Porém, não existem hoje soluções que implementem um ambiente de conectividade baseado em VPN *gateways*, desenvolvidos especificamente para este tipo de endereçamento IP para acesso à Internet.

Apesar de soluções baseadas em endereçamento IP fixo poderem ser adaptadas a utilização em um sistema baseado em IP dinâmico, elas se mostram inadequadas a este ambiente de conectividade.

Um dos problemas com essa abordagem é a necessidade de uma reconfiguração manual da VPN quando o provedor de serviço altera o endereço externo do VPN *gateway*. Essa necessidade diminui a praticidade e consequentemente a disponibilidade da solução adotada.

Um outro problema ocorre quando se tenta automatizar a reconfiguração da VPN utilizando serviços de DNS (*Domain Name System*) dinâmico. Este tipo de serviço, da forma como são usados, são vulneráveis a ataques do tipo DNS *spoofing*. Um atacante poderia se passar por uma das pontas da VPN, interrompendo a comunicação entre as duas pontas da VPN, resultando em negação de serviço (DoS - *Denied of Service*).

1.2 Justificativa

A construção de um sistema específico para trabalhar com endereçamento IP dinâmico, que mantenha um elevado nível de segurança e disponibilidade, possibilitará a utilização de tecnologias como o ADSL para o acesso rápido à Internet para a interligação de redes geograficamente distribuídas através de VPNs. O desenvolvimento de uma ferramenta como esta fornecerá uma solução de conectividade de baixo custo, segura e flexível e que possa ser largamente utilizada em todo o país.

1.3 Objetivo

Este trabalho tem por objetivo desenvolver um sistema baseado em endereçamento IP dinâmico com reconfiguração automática das VPNs, ficando transparente para o usuário final a mudança de endereços por parte do provedor. O sistema também deve manter um elevado nível de segurança em todos os elementos de sua arquitetura, eliminando vulnerabilidades existentes nas soluções adaptadas a este tipo de endereçamento. Além disso ele deve ser capaz de interagir com outros equipamentos e sistemas.

1.4 Metodologia

Para o desenvolvimento deste trabalho, inicialmente, foi realizado um estudo bibliográfico em busca de soluções existentes ou que pudessem ser utilizadas para o problema abordado nesta dissertação. Foram analisadas as vantagens e desvantagens dos trabalhos relacionados encontrados e, então, foi projetada uma nova arquitetura para se obter uma solução mais adequada ao cenário apresentado.

O passo seguinte foi a escolha das ferramentas e tecnologias que servissem de base para a implementação da arquitetura proposta. Fazem parte dessa escolha a forma de acesso à Internet, o sistema operacional utilizado, os protocolos mais adequados, os programas para implementação dos servidores e serviços, além das linguagens de programação.

Uma vez definidas as ferramentas e tecnologias a serem usadas, o sistema foi desenvolvido e implementado na forma de um protótipo, o qual foi, primeiramente, testado em laboratório. Após a sua estabilização foi oferecido a empresas do mercado para sua utilização em ambientes de produção.

A sua utilização em ambientes reais permitiu a validação da arquitetura através de testes de funcionalidade e medidas de tempo de disponibilidade do sistema. Além disso, foi obtida a satisfação dos clientes com a solução ofertada.

1.5 Resultados e Contribuições

A abordagem proposta nesta dissertação foi implementada e testada em ambientes reais de empresas da Grande Vitória durante 4 anos. O sistema funcionou de maneira

satisfatória interligando as redes de clientes de diversos ramos da economia. A interligação foi realizada através de VPN de forma segura e automática, atingindo um índice de 98% de disponibilidade da conexão. Vale ressaltar que os 2% restantes incluem a indisponibilidade do acesso causado pela operadora.

Outro resultado importante foi o nível de segurança atingido, onde ataques ao tráfego de informações para o estabelecimento da conexão foram evitados.

As principais características e contribuições da abordagem aqui proposta são:

- A utilização de um servidor DNS dinâmico (DDNS - *Dynamic Domain Name System*) e seguro (DNSSEC - *Domain Name System Security Extensions*) como ponto central de armazenamento e consulta dos endereços IP para o estabelecimento da VPN.
- O uso de protocolos padronizados e abertos na construção do sistema, garantindo a interoperabilidade da solução.
- A construção do sistema baseado em software livre, proporcionando flexibilidade à solução e a possibilidade de aprimoramento e implementação em outros equipamentos.
- Monitoramento, reconfiguração e estabelecimento automático das VPNs.

1.6 Estrutura da dissertação

Esta dissertação está organizada da seguinte forma. O Capítulo 1 corresponde a esta introdução. No Capítulo 2 é apresentado o referencial teórico, enquanto o Capítulo 3 discute os trabalhos relacionados. A abordagem proposta é apresentada no Capítulo 4 descrevendo-se toda a sua implementação. Os resultados são apresentados no Capítulo 5 através de um estudo de caso. Finalmente o Capítulo 6 apresenta as conclusões deste trabalho.

Capítulo 2: Referencial Teórico

Desde o início da Internet comercial no Brasil, observa-se um grande crescimento no número de pontos conectados à rede. Este crescimento é estimulado por uma queda de preço e aumento da velocidade de acesso, além de uma maior disponibilidade de aplicações *on-line* e conteúdo.

Segundo o Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (CETIC.br), que é responsável pela produção de indicadores e estatísticas sobre a disponibilidade e uso da Internet no Brasil, em 2006 quase 90% das empresas brasileiras possuíam conexões por banda larga para acesso à Internet.

O termo Banda Larga é usado neste texto de acordo com a classificação feita em [1] e [2] que consideram as conexões permanentes à Internet, com velocidades superiores a 128Kbps utilizando as principais tecnologias de acesso no Brasil: xDSL (*Digital Subscriber Loop family*), Cable Modem, FWA (*Fixed Wireless Access*)/Rádio e Satélite.

O Brasil é um país com uma infraestrutura de telefonia bastante desenvolvida e limitada infraestrutura de TV a cabo. Essas duas são as principais tecnologias de acesso por Banda Larga no país. Em função disso, ao final de 2007, 75% das conexões eram realizadas via xDSL e 22% por cable modem disponibilizado pelas operadoras de TV a cabo. As outras tecnologias presentes são menos significativas.

De acordo com a pesquisa feita pela Cisco System [1], ao final de 2007, o Brasil possuía aproximadamente 7 milhões de conexões à Internet por Banda Larga e 45 mil conexões por linhas dedicadas.

O mercado de linhas dedicadas se concentra no usuário corporativo. Grandes e médias empresas são as principais responsáveis pelo consumo deste tipo de conexão, que necessita de maior velocidade, e principalmente, maior índice de disponibilidade.

O mercado de Banda Larga, ao contrário, é voltado para os consumidores domésticos. Além dos usuários residenciais, algumas empresas de porte menor (principalmente micro e pequenas) também utilizam tal tecnologia, buscando uma melhor relação entre custo e benefício.

De 2006 para 2007 houve um crescimento em torno de 35% no número de conexões por Banda Larga. Os principais fatores responsáveis por esse crescimento são o aumento da

velocidade das conexões e o aparecimento de soluções específicas para o mercado SOHO (*Small Office Home Office)* e de médias empresas. É interessante notar, que parte desse crescimento ocorre em função da aquisição de conexões de Banda Larga por grandes empresas, com objetivo de disponibilizar acesso para seus funcionários, para que eles possam trabalhar remotamente e para conectar escritórios ou filiais de menor porte.

Em conjunto com o aumento do número de conexões, há um aumento na mesma proporção na utilização de tecnologias como a VPN, que garantam a segurança das informações trafegadas pela Internet.

2.1 VPN

O crescimento na utilização de VPNs vem encontrando dificuldade em função da falta de interoperabilidade das implementações existentes [3]. Isso ocorre devido à confusão em relação ao escopo e à própria definição de VPN, já que não existe uma definição comum ao termo.

O entendimento do conceito de uma VPN ou rede privada virtual, vem do próprio entendimento das palavras que formam este acrônimo. Uma rede privada pode ser entendida como uma rede pertencente a uma única instituição e apenas os integrantes dessa instituição possuem acesso a esta rede, bem como às informações nela trafegadas, ao contrário de uma rede pública, onde a rede é compartilhada por qualquer usuário. Já o termo rede virtual, corresponde a redes abstratas, que não possuem correspondência direta com a alocação física do meio, normalmente utilizando conexões temporárias.

No contexto deste trabalho, o termo VPN é definido como uma rede que utiliza uma infraestrutura de comunicação compartilhada, para interligar dois ou mais elementos com um alto nível de segurança, seguindo as definições descritas em [3].

As VPNs possuem uma importância fundamental no âmbito das corporações, principalmente no que se refere a custos, pois permitem a substituição de linhas dedicadas e estrutura de conexão remota, por uma infraestrutura de conexão pública e clientes VPN. De fato, a utilização de uma única conexão pública como a Internet facilita o gerenciamento dos acessos, pois não é mais necessário criar um ponto de acesso privado para cada uma das conexões remotas.

Considerando o alcance atual da Internet, junto às mais variadas formas de acesso a essa rede, associado a um aumento da flexibilidade para as organizações, é possível permitir o acesso de usuários mesmo que estejam a cada momento em um local diferente e, ainda assim, permitir que trabalhem como se estivessem presentes localmente de forma transparente.

Outra importante vantagem no modelo de utilização de VPNs é a escalabilidade da solução, pois um novo *site* ou usuário pode ser integrado à infraestrutura sem muitas configurações ou investimentos adicionais. Dado o seu controle centralizado, há um melhor gerenciamento de recursos como o aumento da velocidade do *link* de comunicação ou a utilização de uma nova faixa de endereçamento de rede, sempre que houver um crescimento do número de acessos.

Apesar das vantagens como redução de custos, flexibilidade, transparência e escalabilidade da utilização do modelo de conexão através de VPN, ele possui algumas desvantagens que devem ser analisadas antes de sua adoção.

A primeira desvantagem é a diminuição do desempenho da solução. A utilização de VPN traz um aumento do *overhead* da comunicação, pois introduz novos campos ou cabeçalhos, dependendo da implementação escolhida. Além disso, há um aumento do processamento necessário ao se utilizar mecanismos de segurança e controle do fluxo de informações.

Outra desvantagem da adoção de VPNs é que, apesar do modelo ser simples, o seu projeto e implementação podem ter um nível de complexidade alto [4]. A escolha do tipo de VPN, topologia, arquitetura, protocolos ou algoritmos utilizados merecem uma análise cuidadosa para que não dificulte ou até mesmo inviabilize a adoção do modelo. Essa escolha depende dos requisitos de cada organização, tais como segurança, custo ou qualidade de serviço.

2.1.1 Tunelamento

Tunelamento é a capacidade de encapsulamento de um pacote de dados original dentro de outro pacote de dados, de forma que o pacote de dados original seja invisível na rede que ele será transportado. O objetivo é simular uma conexão física entre as duas redes remotas através de uma outra rede.

Para estabelecer um túnel é necessário se definir os *endpoints*, ou *gateways* do túnel e o protocolo utilizado para o transporte dos pacotes originais. Podemos ver um exemplo de

tunelamento na Figura 2.1. O *gateway* da rede de origem é responsável pelo encapsulamento do pacote original dentro do pacote de transporte e enviar este pacote de transporte até o *gateway* de destino. O *gateway* de destino recebe o pacote encapsulado, retira o pacote original e envia ao destino dentro da rede local. Para executar este processo, os *gateways* em questão devem possuir a informação de que pacotes devem ser encapsulados, qual o *gateway* final e qual o protocolo que será usado para o transporte deste pacote encapsulado. Com relação às redes remotas, é transparente a forma, protocolo ou rede utilizada pelos *gateways* para envio do pacote original até o *gateway* no final do túnel

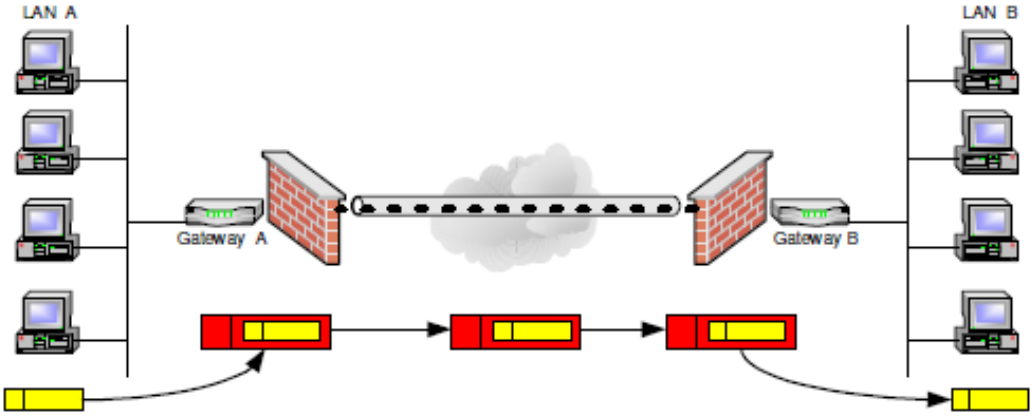

Figura 2.1: Tunelamento

Existem vários motivos para se justificar a utilização do mecanismo de tunelamento entre duas redes remotas, entre eles estão:

- O pacote original utiliza um protocolo não suportado pela rede de transporte usada na conexão entre as redes remotas. É o caso da interligação de redes IPv6 usando a infraestrutura de rede baseada no protocolo IPv4 ou a conexão de redes IP sobre ATM (*Asynchronous Transfer Mode*).
- A necessidade de um encaminhamento diferente do pacote original na rede de transporte, como ocorre para pacotes IP em túneis GRE (Generic Routing Encapsulation).
- A garantia de segurança dos dados trafegados pela rede de transporte, que é utilizado nas implementações de VPN para se garantir o aspecto privado da comunicação.

Tecnologias de tunelamento podem ser usadas nas diversas camadas do modelo OSI (*Open Systems Interconnection*) [5], tais como PPTP (*Point-to-Point Tunneling Protocol*) e L2TP (*Layer 2 Tunneling Protocol*) na camada de enlace, IPsec (*Internet Protocol security*) na camada de rede ou SSL (*Secure Sockets Layer*) na camada de apresentação. Algumas dessas tecnologias são descritas nas próximas seções deste trabalho.

2.1.2 Segurança da Informação

As redes de computadores, notadamente a Internet, mudaram a forma de se usar os sistemas de informação. Os serviços oferecidos e as possibilidades de utilização são muito mais amplos do que em sistemas fechados. Da mesma forma cresceram proporcionalmente os riscos à privacidade e integridade da informação. Devido a isso, mecanismos de segurança da informação devem ser projetados de maneira a prevenir acessos não autorizados aos recursos e dados desses sistemas [6].

A segurança da informação é a proteção dos sistemas de informação contra a negação do serviço ao usuário autorizado, e contra o acesso e modificação não autorizada da informação armazenada, em processamento ou em trânsito. A segurança da informação deve ser capaz de prevenir, detectar, deter e documentar eventuais ameaças aos sistemas de informação. Segundo a [7], [8] e [9], a segurança da informação é caracterizada pela preservação de três atributos básicos da informação:

- Confidencialidade: propriedade da informação que limita seu acesso a entidades legítimas, ou seja, àquelas autorizadas pelo proprietário da informação.
- Integridade: propriedade que garante que a informação manipulada mantenha todas as características originais estabelecidas pelo proprietário da informação.
- Disponibilidade: propriedade que garante que a informação esteja sempre acessível para o uso legítimo, ou seja, por aqueles usuários autorizados pelo proprietário da informação.

É importante destacar alguns pontos com relação a esses atributos. A confidencialidade não garante a restrição de acesso aos dados trafegados por uma estrutura pública, por exemplo. Um usuário não autorizado pode conseguir o acesso a esses dados, mas não à informação contida neles. Já o atributo integridade não pode ser confundido com

confiabilidade. Uma informação pode ser imprecisa, mas ainda assim deve se manter-se íntegra, sem que sofra alterações não autorizadas.

Alguns autores [8], [9], [10], [11], [12], [13], [14] defendem que para a informação ser considerada segura, ela ou os sistemas que a suportam devem possuir também os seguintes atributos:

- Autenticidade: propriedade da informação em que é possível se atestar a veracidade da origem da informação. É a certeza que uma informação provém da fonte anunciada.
- Não Repúdio: impossibilidade de negação de uma operação ou serviço que criou ou modificou uma informação. Não é possível negar o envio ou recepção de uma informação.
- Auditoria: identificação de diversos passos ou processos que uma informação foi submetida, tais como os participantes, horários, locais e ações realizadas.

Para o estabelecimento de uma conexão em uma VPN, é necessária a autenticação das duas entidades que precisam estabelecer a comunicação. No entanto, não basta apenas identificar as partes envolvidas na comunicação, é necessário definir que recursos estarão disponíveis para cada uma das partes. Dessa forma, vários níveis de controle de acesso podem ser implementados, seja pelo controle de que entidades podem estabelecer uma conexão VPN ou pelo controle dos recursos ofertados a cada uma destas entidades.

A decisão sobre o controle de acesso a estes recursos pode ser implementada pelas próprias entidades envolvidas na comunicação ou por um elemento externo independente.

2.1.3 Tipos de Túneis

VPNs podem interligar redes privadas utilizando-se a Internet como rede pública. Porém além desse tipo de interligação de uma rede a outra, VPNs podem ser usadas para conexão de um único *host* à uma rede privada através da Internet, ou mesmo para a comunicação de um computador a um único outro computador dentro de uma rede corporativa remota.

Para se estabelecer uma conexão entre esses vários elementos, existem diferentes tipos de túneis.

Host-to-Host

Essa é a forma mais simples de se estabelecer um canal seguro de comunicação. Esse tipo de túnel é estabelecido entre dois computadores para troca de informações apenas entre eles, como mostrado na Figura 2.2. Um exemplo desse tipo de túnel é uma conexão estabelecida entre um cliente e um servidor Web através do protocolo SSL. Apenas o computador com o *browser* cliente e a máquina com o servidor Web terão acesso às informações dessa conexão e são responsáveis por iniciar e finalizar o túnel entre eles. Os dois *hosts* podem estar inclusive na mesma rede e ainda assim optarem por estabelecer um túnel deste tipo com o objetivo de proteger a comunicação de *insiders* [15].

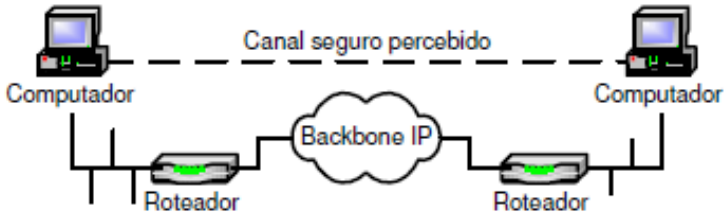

Figura 2.2: Host-to-Host

Host-to-Gateway

Esse tipo de túnel é estabelecido entre um *host* e um *gateway* de VPN. Por trás do gateway, existirá uma ou mais redes intituladas de redes internas. O objetivo desse tipo de conexão é possibilitar que essas redes internas se comuniquem com o *host* remoto utilizando o túnel estabelecido pelo *gateway* e vice-versa. Um exemplo desse tipo de túnel é mostrado na Figura 2.3.

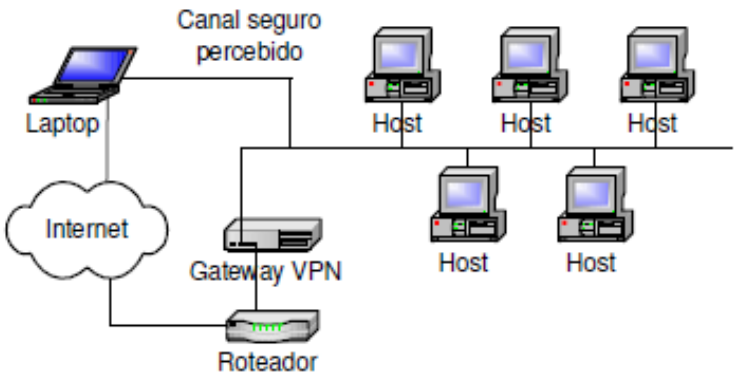

Figura 2.3: Host-to-Gateway

Um pacote originado do *host* é encapsulado, colocado no túnel e encaminhado em direção ao *gateway.* Esse irá receber o pacote, desencapsulá-lo e o enviará ao destino final já fora do túnel estabelecido. Nesse tipo de conexão, o *gateway* é responsável pelo estabelecimento do túnel e encaminhamento dos pacotes, mas não participa da comunicação.

Túneis estabelecidos através de *gateways* facilitam a administração e o controle de acesso a rede, pois permitem o gerenciamento centralizado da solução e fornecem maior transparência a comunicação, pois os *hosts* das redes internas se comunicam com o *host* remoto como se ele estivesse na mesma rede local.

Para VPNs que utilizam a Internet onde o endereço do host é um endereço dinâmico, esse tipo de conexão é chamada de *RoadWarrior-to-Gateway* [16]. Nesse caso, o túnel só pode ser estabelecido pelo *host*, pois não é possível ao *gateway* saber previamente o endereço remoto do *host*. Esse tipo de comunicação é muito comum para que usuários que estejam em trânsito possam acessar a rede corporativa remotamente.

Gateway-to-Gateway

A Figura 2.4 mostra esse tipo de túnel que possibilita a duas ou mais redes, mesmo estando separadas geograficamente, se comunicarem através de um túnel estabelecido entre dois *gateways* sobre a Internet.

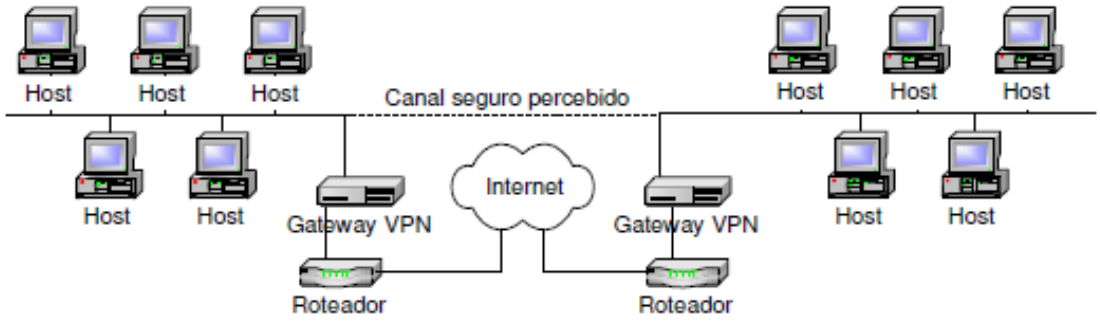

Figura 2.4: Gateway-to-Gateway

Um pacote com origem de um *host* na rede local, é enviado normalmente com o destino à rede remota. Esse pacote será interceptado pelo *gateway* local, será encasulado e encaminhado pelo túnel até o *gateway* remoto. O *gateway* remoto irá receber o pacote, desencapsulá-lo e o encaminhará ao *host* de destino. Para os *hosts* de origem e destino, a utilização do túnel é totalmente transparente. Na verdade, se o túnel for substituído por uma ligação física direta, nada será modificado na forma de comunicação entre estes *hosts*. Um exemplo disso é a utilização de um link físico interligando duas redes e, quando ocorre uma falha neste *link* é utilizada uma VPN pela Internet como redundância da comunicação.

2.1.4 Topologia

Na seção anterior foram descritos os tipos de túneis que podem ser utilizados por duas entidades para o estabelecimento de uma VPN. Para uma solução corporativa, onde vários túneis podem ser estabelecidos, a escolha de uma topologia adequada é fundamental para o sucesso do projeto.

Para escolha dessa topologia, é determinante a definição dos tipos de túneis que serão estabelecidos entre as entidades envolvidas na comunicação, dos recursos utilizados por cada uma delas e o controle de acesso a esses recursos. Em um ambiente com poucas conexões, a escolha da topologia pode não ser tão relevante, porém para um ambiente corporativo, com alta capilaridade, essa escolha passa a ter fundamental importância.

Segundo [17], a topologia adotada deve possuir escalabilidade, que implica em vários fatores, tais como facilidade de gerenciamento, custo, possibilidade de expansão, disponibilidade e flexibilidade. Para se tirar o melhor proveito da solução é essencial a escolha correta da topologia.

São apresentados a seguir os principais modelos de topologias adotados em VPNs. Cada modelo possui vantagens e desvantagens e sua adoção vai depender de cada situação que se apresente. Algumas soluções podem envolver inclusive a adoção restrita de uma topologia ou até mesmo um modelo híbrido derivado.

Hub-and-Spoke

Nesse modelo existe um nó central, chamado de *hub*, que concentrará as conexões de vários elementos remotos, chamados de *spoke*. A Figura 2.5 mostra os elementos remotos se comunicando diretamente com o nó central. É possível a comunicação entre os elementos remotos, desde que a comunicação com origem em um destes elementos passe pelo nó central antes de seguir em direção a um outro elemento remoto.

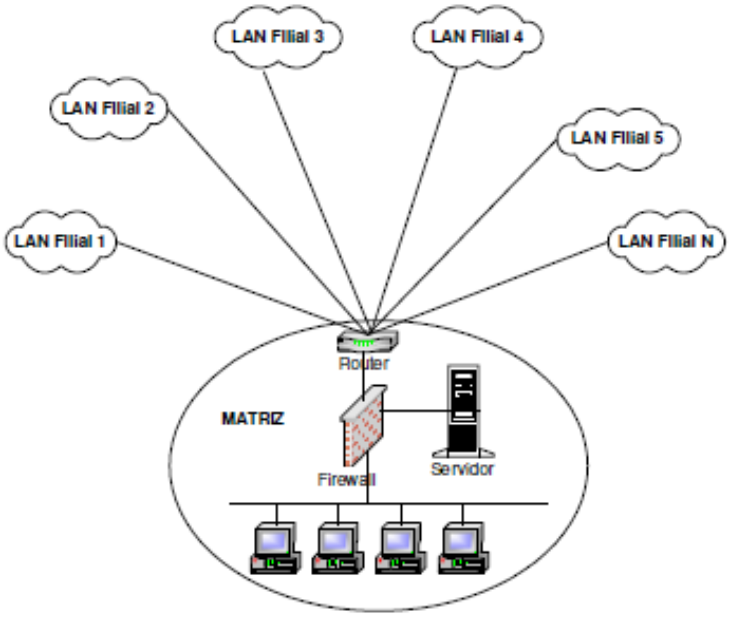

Figura 2.5: Hub-and-Spoke

Esse modelo é muito utilizado em cenários nos quais várias filiais precisam acessar recursos na matriz ou uma empresa possui trabalhadores em trânsito e fazem acesso remoto à empresa via Internet. Esse modelo também é utilizado quando se quer ter um controle do acesso de redes remotas no nó central.

Uma das grandes vantagens desse modelo é a facilidade de gerenciamento pois todo ele é feito de forma centralizada. Porém esse modelo possui uma série de desvantagens, como a diminuição do desempenho da comunicação, pois no caso de comunicação entre os *spokes*, um pacote deve ser enviado ao *hub* por um túnel, será processado e reencaminhado por outro túnel até o *spoke* de destino. Além disso a *hub* se torna um ponto único de falha na comunicação.

Full-Mesh

Nesse modelo de topologia, cada um dos nós possui uma ligação com cada um dos outros nós envolvidos na comunicação, como mostrado na Figura 2.6. Com isso há uma otimização do roteamento, já que para cada comunicação estabelecida, os pacotes são enviados diretamente para o destino, eliminando *hops* desnecessários. O maior problema desse modelo é a dificuldade de gerenciamento e expansão da solução, já que o número de túneis cresce a uma taxa de n(n-1)/2 com o aumento do número de nós (n). Um cenário de utilização do modelo é um ambiente em que os recursos estão distribuídos entre a matriz e suas filiais, e é necessário o acesso de cada nó para todos os outros nós. Além disso, o desempenho e disponibilidade da solução é um fator crítico.

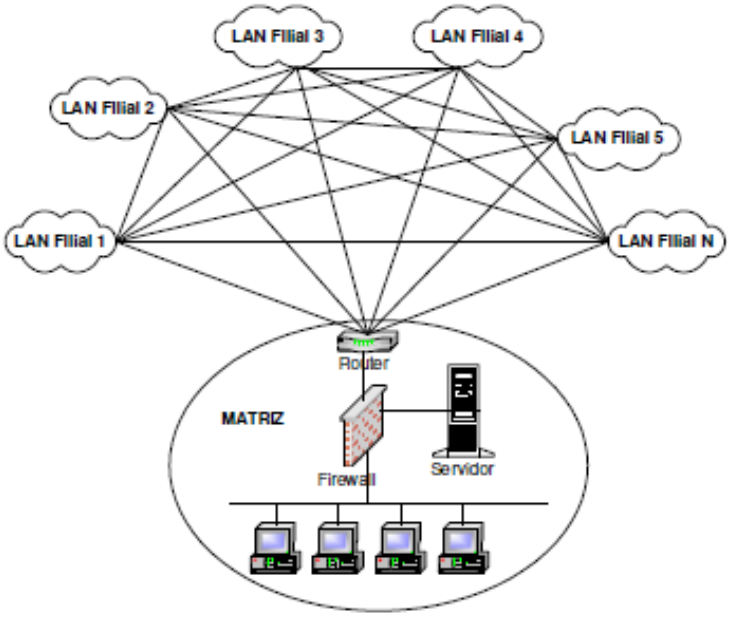

Figura 2.6: Full Mesh

Partial-Mesh

Esse modelo pode ser considerado um modelo híbrido entre os dois modelos anteriores. Os nós envolvidos na comunicação podem estabelecer túneis para mais de um nó diferente, porém não necessariamente para todos. Isso permite a diminuição do número total de túneis, elimina o ponto único de falha, pois podem existir múltiplos caminhos para comunicação entre dois nós, e para os acessos onde é necessário um maior desempenho pode ser estabelecido um túnel direto entre os nós.

Um exemplo do modelo híbrido pode ser observado na Figura 2.7 cujo cenário representa um ambiente corporativo onde os recursos principais estão distribuídos entre a matriz e algumas filiais. O modelo de comunicação entre esses nós será o *full-mesh*. A corporação possui um número grande de outras filiais que possuirão túneis estabelecidos entre elas, a matriz e as filiais principais num modelo do tipo *partial-mesh*. Além disso, a corporação possui acesso de funcionários remotos que conectarão a matriz através do modelo *hub-and-spoke*. Esse cenário ilustra bem que a escolha da topologia depende essencialmente do cenário apresentado.

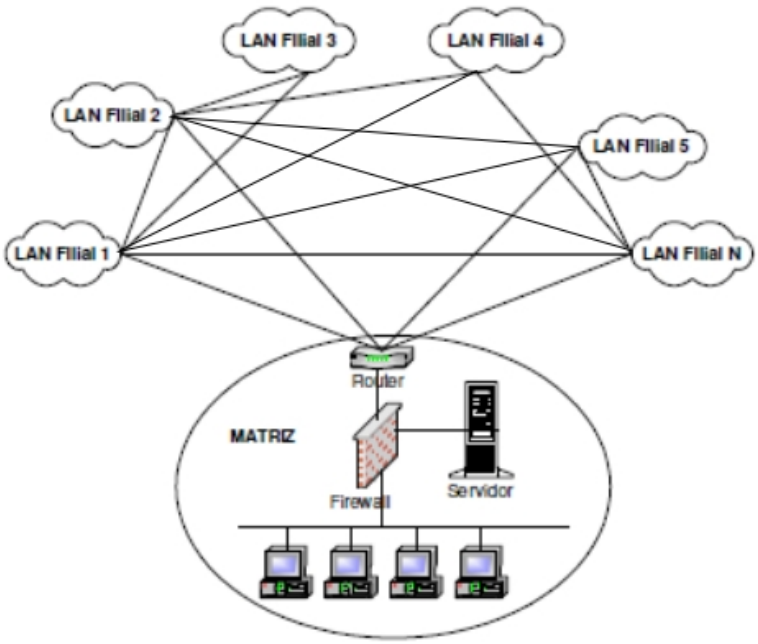

Figura 2.7: Partial Mesh

2.1.5 Tipos de VPN

Existem vários tipos de VPN. Cada um desses tipos possui características distintas e, por isso, são mais adequados para solução de problemas diferentes. Os objetivos de cada solução nem sempre são concorrentes e muitas vezes podem ser utilizadas em conjunto em função de requisitos específicos, seja para um aumento de segurança, diminuição de custo ou uma maior disponibilidade da conexão.

Em função disso é feita uma análise das vantagens e desvantagens dos tipos de VPN mais comumente oferecidas no mercado e o ambiente mais adequado a sua utilização.

Secure vs Trusted VPN

Inicialmente as instituições contratavam circuitos de um determinado provedor para interligação de *sites* remotos. O cliente poderia definir o seu próprio endereçamento e sua política de utilização. A segurança oferecida pelo provedor era a garantia de que ninguém teria acesso a esse circuito. Esses circuitos são implementados sobre roteadores e *switches*, agregando VPNs de vários clientes diferentes, onde qualquer falha de configuração ou comprometimento dos equipamentos pode permitir o acesso à informação trafegada ou outros tipos de ataque. Este tipo de VPN, onde o cliente deve confiar plenamente na operação e mecanismos de segurança do provedor, sem ter ao menos a opção de auditá-lo, é chamado de *trusted* VPN.

Com a chegada da Internet como um meio viável de interligação de *sites* remotos, percebeu-se que as *trusted* VPNs não possuíam um nível de segurança adequado e, em função disso, as instituições passaram a utilizar autenticação e criptografia nos elementos controlados por elas.

Dessa forma, foi possível prover segurança real [18] aos pacotes de dados colocados em um túnel cifrado com destino a rede remota. VPNs baseadas em protocolos que se utilizam de criptografia e autenticação são chamadas de *secure* VPNs, pois a segurança é oferecida pelo próprio protocolo.

O “VPN Consortium”, conhecido com VPNC, define algumas propriedades para cada tipo de VPN, conforme [18]:

Trusted:

- Ninguém a não ser o provedor pode afetar a criação e modificação de um caminho na VPN.

- Ninguém a não ser o provedor pode alterar, inserir ou apagar dados de um caminho/circuito estabelecido para uma determinada VPN.
- O roteamento e endereçamento utilizados em uma *trusted* VPN devem ser estabelecidos antes da VPN ser criada.

Secure:

- Todo tráfego na VPN deve ser cifrado e autenticado.
- As propriedades de segurança da VPN devem ser de comum acordo entre todas as partes participantes da VPN.
  Ninguém de fora da VPN pode afetar as propriedades de segurança definidas.

O VPNC define também alguns protocolos para cada tipo de VPN, conforme descrito em [18], tendo em vista as propriedades da organização, que é testar e avaliar a interoperabilidade entre fabricantes segundo uma série de níveis preestabelecidos

Para as *trusted* VPNs existem os circuitos ATM, *Frame-Relay* [19] e MPLS (*Multiprotocol Label Switching*) [20].

Para as *secure* VPN, o IPSec, o L2TP /IPSec e os protocolos SSL e TLS (*Transport Layer Security*)são utilizados.

Os protocolos SSL e TLS são utilizados para estabelecimento de canais seguros pelos *browsers* e servidores Web, porém pouco utilizados para estabelecimento de VPNs entre *sites* remotos, principalmente pelo seu baixo desempenho. Para esses casos de VPN corporativa, que é o foco de estudo deste trabalho, o protocolo padrão utilizado pelo mercado para o estabelecimento de *secure* VPNs, é o IPsec [15], [21], [22] e [23].

O IPsec foi desenvolvido pelo IETF (*Internet Engineering Task Force)* originalmente para o IPv6 (*Internet Protocol version 6*) e depois foi integrado também ao IPv4 (*Internet Protocol version 4*) . O protocolo possui três componentes principais:

- AH (*Authentication Header*): fornece serviço de autenticação ao pacote IP, não alterando o conteúdo do *payload* do pacote. A autenticação é feita sobre todo o pacote [24].
- ESP (*Encapsulating Security Payload*): fornece cifragem de pacotes, sendo que, opcionalmente, pode haver a autenticação do que foi cifrado. A cifragem é realizada somente sobre o conteúdo interno ao cabeçalho ESP [25].

- IKE (*Internet Key Exchange*): negocia parâmetros de conexão para os outros dois, incluindo chaves [26].

O tunelamento com ciframento é feito usando-se o cabeçalho ESP à frente do pacote original, acrescentando-se a esse conjunto um cabeçalho IP externo.

A classificação entre o tipo *secure* e *trusted* está definida em função da utilização ou não de protocolos de criptografia no estabelecimento da VPN. Normalmente o tipo *trusted* VPN é oferecido pelos provedores de acesso a clientes que buscam certo nível de serviço. Nesses casos, o cliente quer a garantia por parte do provedor que seus dados se deslocarão por um caminho específico e que possuirão propriedades como latência, largura de banda, prioridade e tempo de resposta bem definidos. Isto é possível pois todo o controle da infraestrutura de comunicação está sob a responsabilidade do provedor. O problema desse tipo de VPN é que é impossível para o cliente se certificar que seus dados estão realmente seguros.

VPNs do tipo *secure*, são adequadas às instituições que possuem dados sensíveis e por isso precisam ter certeza que eles não serão capturados ou modificados por um atacante, seja ele um *insider* [15] no provedor ou vindo de uma rede externa, e que chegarão ao destino de forma privada e sem alterações.

É importante notar que as *secure* e *trusted* possuem propósitos diferentes, mas não são exclusivas mutuamente. Um outro tipo de VPN que pode ser utilizada é a VPN híbrida.

As VPNs híbridas, geralmente utilizam uma *secure* VPN sobre uma *trusted* VPN, geralmente visando a segurança fornecida pela primeira e QoS oferecida pela segunda [18], podendo inclusive ter parte do caminho como *secure* e parte como *trusted* VPN. A utilização de tecnologias VPN/MPLS e DiffServ (*Differentiated Services*) estão sendo cada vez mais utilizadas no mercado para que se tenha garantia de nível de serviço em *secure* VPNs [27].

Alguns países ou órgãos reguladores exigem que determinados tipos de dados, como transações médicas e financeiras utilizem uma tecnologia de *secure* VPN mesmo quando existe a infraestrutura de uma *trusted* VPN [28].

CPE (Customer Premises Equipment) vs Baseadas em Rede

As IP VPNs podem ser classificadas em função do posicionamento e quem é responsável pela administração do *gateway* VPN.

Em VPNs baseadas em CPE, o *gateway* VPN responsável por iniciar e terminar o túnel seguro de comunicação se encontra dentro da organização. Soluções baseadas em CPE podem ser gerenciadas pela própria empresa ou ter sua gerência terceirizada a uma empresa de segurança.

No modelo em que o CPE é gerenciado dentro da empresa, toda arquitetura, acessos e administração do equipamento, ficam a cargo da equipe interna de TI (Tecnologia da Informação). Isto exige um alto nível de experiência e capacitação da equipe responsável pelo gerenciamento da solução. Grandes instituições conseguem manter equipes desse nível, resultando em uma ambiente de VPN com um alto índice de segurança. Para pequenas e médias instituições, é uma tarefa árdua e nem sempre bem sucedida, e, muitas vezes, acaba comprometendo a segurança de todo o ambiente.

Para acessos remotos o papel de *gateway* VPN é exercido por um *software* cliente, estabelecendo um túnel do tipo *Host-to-Gateway*. O usuário, após se autenticar no gateway remoto, recebe a política de segurança definida pela instituição para comunicação para este tipo de conexão, de forma a não comprometer a rede da organização.

CPEs podem possuir a sua gerencia terceirizada, onde a administração do *gateway* de VPN é realizada por uma empresa de segurança. Esse é um modelo interessante para empresas que não possuem recursos para escolher, administrar e se atualizar na constante evolução das soluções de VPN que estão sempre surgindo no mercado. A definição da solução mais adequada é fundamental para a segurança de seu ambiente.

A opção pela terceirização da gerência do *gateway* VPN deve ser analisada com cuidado, pois traz alguns riscos e implicações. Não é pelo fato de possuir um canal seguro entre os *gateways* remotos que os seus dados e sua conexão estão seguros [29]. As instituições erram ao delegar a responsabilidade pela definição de sua política de segurança à empresa terceirizada. O terceiro não conhece as particularidades de cada cliente, qual a criticidade de cada ativo dentro da instituição ou quem deve ter acesso a uma determinada informação. Na verdade, em muitas empresas, a definição da política de segurança não é uma responsabilidade apenas da área de TI.

Além disso, a instituição deve definir muito bem o SLA (*Service Level Agreements*) com a empresa de segurança contratada, tendo dados concretos que refletem sua política de segurança. Um SLA não mensurável é inútil [30].

As VPNs podem também ser do tipo baseado em rede. A sua principal característica é que todos os dispositivos envolvidos na construção de uma VPN são sistemas compartilhados de propriedade do ISP (*Internet Service Provider*). Esses dispositivos estão todos localizados dentro de seu *backbone*, começando a VPN na borda do *backbone* do ISP. Na empresa fica somente um roteador comum e através de um *link* qualquer, o *site* interessado em integrar a VPN deve se conectar ao POP (*Point of Presence*) mais próximo. Todo o gerenciamento, construção, manutenção e separação entre VPNs distintas cabe ao ISP.

Uma grande vantagem desse tipo de VPN é a simplicidade da solução para o cliente. Toda a complexidade inerente ao gerenciamento da solução fica a cargo do ISP. Novos sites podem ser incluídos no ambiente de VPN apenas adicionando um novo *link* de comunicação. A qualidade de serviço é um outro ponto forte, pois nem sempre é possível garantir a qualidade de serviço junto ao ISP para dados cifrados.

Apesar do nível de segurança ser maior do que as *trusted* VPNs, pois os dados trafegam cifrados dentro do *backbone*, a comunicação entre a instituição e o ISP não é cifrada, o que compromete o nível de segurança do fluxo de dados. Ao contrário da VPN baseada em CPE, que mantém a segurança fim a fim dos dados trafegados entre os *sites* remotos.

Outra vantagem das VPNs baseadas em CPE é a flexibilidade oferecida pelo modelo para adição de novos sites, modificações na arquitetura ou no tipo de solução adotada. Para VPNs baseadas em rede, o ISP obriga a instituição a usar tecnologias de comunicação suportada pelos seus equipamentos, inviabilizando muitas vezes o acesso remoto a rede local.

Outra desvantagem apontada para uma solução baseada em CPE é o custo do equipamento. Calculando-se a economia no custo mensal, o custo do próprio equipamento pode ser faturado nessa conta e rapidamente restituído, em comparação ao custo mensal das VPNs baseadas em rede, sejam *gateways* adquiridos pela própria instituição ou oferecidos por uma empresa terceirizada que os oferecerá dentro do serviço.

2.1.6 Arquitetura de *Firewall*

Segundo [31] *firewall* é definido como “um componente ou conjunto de componentes que restringem o acesso entre uma rede protegida e a Internet, ou entre outros conjuntos de redes”. Todo o tráfego entre as redes protegidas pelo *firewall* é vigiado, filtrado e algumas vezes modificado.

Um *gateway* VPN é um importante componente utilizado na composição de um *firewall*, porém não é o único. O *gateway* de VPN deve ser utilizado em conjunto com outras tecnologias de forma adequada para possibilitar o cumprimento da política de segurança da instituição.

É possível encontrar no mercado, produtos que se utilizam de diversas dessas tecnologias para compor uma solução de controle de acesso entre a rede interna e a Internet, além do *gateway* VPN. Neste trabalho, são abordadas duas das tecnologias mais utilizadas na composição dessas soluções: o filtro de pacotes e *proxy*.

A filtragem de pacotes é o processo de seletivamente permitir ou bloquear o tráfego de pacotes entre duas redes, fazendo uso de um conjunto de regras de filtragem. Essas regras são baseadas em informações existentes nos cabeçalhos de cada pacote [31]. Os cabeçalhos utilizados na filtragem podem ser de diversas camadas e protocolos distintos, desde o nível de enlace até o nível de aplicação. Algumas implementações de filtros de pacotes também fazem algum trabalho na parte de dados dos pacotes.

Filtro de pacotes são extremamente efetivos na implantação de uma política de segurança. Há uma série de ataques que se consegue evitar com a filtragem de pacotes. De uma forma geral, evitam-se todos os ataques aos serviços que não são permitidos pelas regras configuradas no filtro.

Filtros de pacotes apresentam sempre um bom desempenho. O baixo custo computacional dos filtros de pacotes permite ainda que eles sejam facilmente adequados ao crescimento do número de máquinas na rede interna (escalabilidade). Outra vantagem importante é a transparência, pois filtros não exigem nenhuma configuração especial no *software* cliente e/ou servidores dentro da rede interna. Os usuários só percebem que há alguma filtragem de pacotes quando tentam fazer algo não permitido pela política de segurança implementada [31], [32].

Filtros de pacotes carecem, principalmente em roteadores, de boas ferramentas de administração. As regras de filtragem podem ser muito difíceis de configurar e exigem o conhecimento de uma intricada sintaxe específica para o dispositivo. Além disso, a ordem em que as regras são definidas pode acarretar mudanças significativas no resultado do filtro de pacotes e, em muitos sistemas, a falta de ferramentas de edição torna ainda mais complicada a tarefa do administrador de segurança de editar as regras de filtragem a serem implementadas [33].

O *proxy* tem como objetivo principal controlar todas as comunicações, para algum determinado serviço ou conjunto de serviços, entre as máquinas internas e externas [31]. Com essa tecnologia, quando um cliente na rede interna, por exemplo, deseja estabelecer uma conexão com algum servidor remoto, a comunicação deve ser feita sempre via um servidor *proxy* apropriado. Portanto, esse esquema envolve pelo menos duas conexões: uma entre o cliente interno e o servidor *proxy* e outra entre o servidor *proxy* e o servidor remoto. Os dados de uma conexão são repassados para a outra conexão e vice-versa pelo agente *proxy*, sem que, no entanto, os cabeçalhos de rede e transporte atravessem a máquina *proxy*.

Para servidores remotos é como se todos os clientes estivessem na máquina *proxy*. Os clientes internos, por sua vez, se comunicam com o servidor *proxy* como se esse fosse o próprio servidor remoto. Obviamente, como primeira desvantagem dessa tecnologia, tem-se uma perda de desempenho quando comparada com as tecnologias de filtro de pacotes [32].

Um servidor *proxy* pode aumentar a segurança por examinar todo o fluxo de dados da comunicação na camada de aplicação. Realizando filtragem nesse nível de comunicação, é possível bloquear ataques relacionados às vulnerabilidades próprias dos serviços e protocolos de aplicação [32]. Além disso, pode-se implementar *cache* de dados, prover autenticação de usuários e criar *logs* mais completos dos dados trocados. Neste caso, o *proxy* é conhecido como *proxy* de aplicação ou dedicado.

Como o *proxy* dedicado precisa conhecer uma aplicação em particular, deve existir um *proxy* para cada serviço desejado, transformando o suporte a novas aplicações um grande problema (extensibilidade). Para alguns serviços e protocolos pode ser difícil ou até mesmo impossível implementar *proxies* dedicados [34]. Além disso, com o crescimento de uso da Internet, vários novos serviços e protocolos têm aparecido rapidamente, o que torna complexa a tarefa de criar *proxies* com a mesma rapidez, tornando o filtro de pacotes a opção viável para controle de acesso, nestes casos.

Outro problema com os *proxies* é o seu elevado consumo de recursos do dispositivo. Eles podem tornar-se um gargalo de comunicação, limitando também o crescimento do número de máquinas da rede (perda de escalabilidade) [32]. Além disso, uma máquina *proxy* pode ter sua segurança comprometida também por ataques à falhas de segurança em sua implementação, que em geral, é mais complexa do que o filtro de pacotes.

Em função das desvantagens apresentadas pelos *proxies*, como baixo desempenho, extensibilidade e escalabilidade, a sua utilização em um sistema de *firewall* deve ser restrita

aos serviços que precisam de uma maior segurança, com um controle de acesso mais apurado. Hoje, cada vez mais serviços são oferecidos sobre o HTTP (*Hypertext Transfer Protocol*), tornando-o o protocolo de aplicação mais utilizado na Internet. Consequentemente o *proxy* HTTP também é o mais utilizado em sistemas de *firewall*. Para os serviços que não utilizarão um *proxy* e para a própria proteção do *proxy*, o filtro de pacotes é essencial.

A construção de uma política de segurança de uma rede deve se basear em conceitos mais genéricos do que seja uma rede segura, nos serviços que devem ser protegidos, nos serviços externos que devem estar disponíveis para os usuários internos e vice-versa. Do estudo desses fatores deve resultar uma topologia de equipamentos de segurança que seja reflexo da política de segurança anteriormente estabelecida. A colocação de um equipamento ou serviço VPN deve gerar uma análise de todas as alterações de tráfego que serão introduzidas, com o objetivo de assegurar as premissas da política de segurança.

Como o *gateway* VPN deve receber tráfego da rede, não há qualquer sentido em que ele aceite qualquer outro tipo de tráfego, além daquele de controle, que não seja cifrado. Essa rejeição ou descarte de outros tipos de tráfegos é parte fundamental da política de autodefesa do próprio *gateway* VPN.

O posicionamento do *gateway* VPN em relação ao filtro de pacotes é crucial, pois os filtros não podem aplicar regras de filtragem a pacotes cifrados. Existem várias opções de colocação: em frente ao filtro de pacotes, atrás do filtro de pacotes, no filtro de pacotes, paralelo ao filtro de pacotes, ao lado do *firewall*.

A posição aconselhada no artigo [35] é a colocação da VPN ao lado do filtro de pacotes, em uma interface dedicada. Nessa configuração, todos os pacotes que chegam ao *gateway* VPN passam antes por um filtro de pacotes ou de estados, o que fornece uma proteção contra ataques diretos. Após passarem pelo gateway, terem os cabeçalhos de tunelamento retirados e serem decifrados, os pacotes originais podem passar agora por um processo de filtragem, o que não podia ser feito ao entrarem no filtro por estarem completamente cifrados.

Uma desvantagem dessa solução é a necessidade de outro dispositivo para executar as funções de *gateway* VPN, aumentando-se o custo. Uma alternativa a essa disposição é a utilização do *gateway* VPN no filtro de pacotes, o que eliminaria a necessidade de um outro dispositivo, e consequentemente diminuiria o custo da solução. Essa opção mantém a

proteção ao *gateway* VPN, e a filtragem dos pacotes antes e depois do processo de cifragem/decifragem dos pacotes.

O problema é o acúmulo de funções no mesmo dispositivo, que traz uma diminuição de desempenho. Por isso, esta alternativa é indicada para pequenas e médias empresas, que possuam um baixo tráfego de dados passando pelo *firewall*.

2.2 DNS

Do ponto de vista da Internet, os *hosts* são muito bem identificados pelo endereço IP. No entanto, do ponto de vista do usuário, esses números são difíceis de serem memorizados. É mais fácil para o usuário guardar o nome **www.unitera.com.br** do que o seu respectivo endereço IP **201.38.61.145**.

A fim de facilitar a vida do usuário foi criado o serviço de DNS [36], que associa nomes simbólicos a endereços IP, tornando mais fácil ao usuário final a utilização da Internet como um todo.

Um nome é meramente um identificador que consiste de uma sequência de caracteres escolhidos num alfabeto. O endereço IP é considerado um nome de baixo nível. Para os usuários, no entanto, é preferível a utilização de endereços de alto nível.

2.2.1 Espaço de Nomes Plano

No início da Internet, como uma forma de estabelecer um sistema de nomes, foi criado um único banco de dados que contivesse a relação <nome, IP> de todas as máquinas da Internet. Esse banco de dados era armazenado em um único arquivo chamado de HOSTS.TXT, que era gerenciado e distribuído por uma única entidade. Todos os *hosts* consultavam uma cópia desse arquivo para conseguir traduzir os nomes para endereços IP. Qualquer máquina que fosse adicionada a Internet deveria ser cadastrada nesse servidor de nomes central.

Para uma rede pequena funcionava bem. Porém com o crescimento da Internet, esse sistema passou a ter alguns inconvenientes:

- O processo de alteração da base de dados pelos administradores do sistema passou a se tornar inviável devido a alta taxa de mudanças no espaço de endereçamento.

- Cada nome associado a um *host* deveria ser único, por isso a colisão de nomes dentro do espaço de endereçamento plano passou a ser um problema.
- Já não era mais possível manter a consistência dos dados armazenados em cada cópia dos arquivos, devido a irregularidade e aumento da taxa de mudanças do banco.

Todos estes problemas foram consequência de uma abordagem deficiente do ponto de vista da escalabilidade.

Em 1984 mudou-se do modelo de espaço de nomes plano para o atual modelo de DNS baseado no espaço de nomes hierárquico.

2.2.2 Nomes Hierárquicos

A descentralização do mecanismo de nomes, através da delegação de autoridade de partes do espaço de nomes e a distribuição de responsabilidade do mapeamento entre nomes e endereços IP, é a forma de acomodar um sistema de nomes que cresce rapidamente como ocorre com a Internet. A partição do espaço de nomes deve ser feita de forma a suportar mapeamento eficiente e garantir controle autônomo dos nomes designados.

O sistema de nomes hierárquico funciona pela partição do espaço de nomes no nível mais alto, enquanto a autoridade dos nomes nas subdivisões é passada para um agente responsável. Por exemplo, pode-se escolher particionar o espaço de nomes baseado no nome da organização. Cada organização teria autoridade sobre os nomes internos a ela, inclusive podendo criar subdivisões internas e dividindo as responsabilidades.

2.2.3 Domínio de Nomes na Internet

O DNS tem dois aspectos conceitualmente diferentes:

- Abstrato: Especifica a sintaxe dos nomes e regras para delegar autoridade sobre os mesmos
- Concreto: Especifica a implementação de um sistema distribuído que eficientemente mapeia nomes em IPs.

Um nome de domínio consiste de uma sequência de sub-nomes separados por ponto “.” Por exemplo, ele.ufes.br, denomina o domínio ele (Departamento de Engenharia Elétrica) da UFES (Universidade Federal do Espírito Santo) do Brasil.

O nível de domínio mais alto está repartido nos descritos na tabela X:

COM	Comercial
EDU	Educacional
GOV	Governamental
MIL	Militar
NET	Centros de suporte da rede
ORG	Organizações não governamentais
código do pais	Para cada país

Tabela 2.1: Domínios de Nível Mais Alto

Cada um desses domínios terá um responsável que poderá dividi-lo em sub-domínios, delegando autoridade para outras pessoas. Note que um conjunto de máquinas de um mesmo domínio não implica em localização geográfica igual. Uma máquina do domínio ele.ufes.br pode estar no Laboratório de Automação Inteligente da UFES e outra estar na Argentina.

Como mostrado na figura 2.8, delegação indica uma transferência de responsabilidade na administração a partir daquele ponto na árvore DNS.

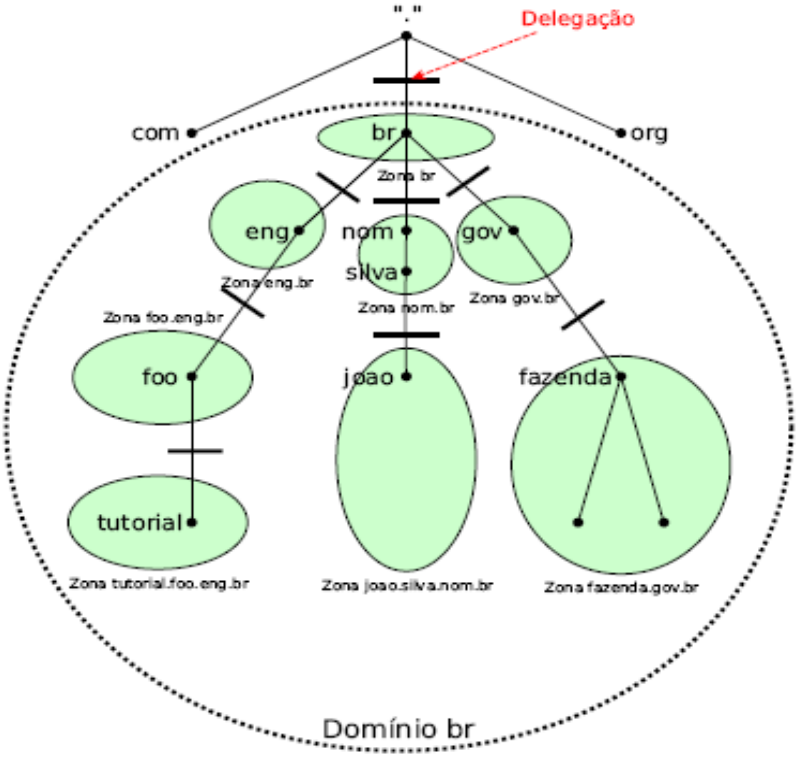

Figura 2.8: Delegação de Responsabilidade do DNS

No Brasil, o órgão ao qual foi delegada a responsabilidade pela implementação dos registros dos domínios com a terminação ".br" é o Núcleo de Informação e Coordenação do Ponto br (NIC.br). Toda a informação pertinente a registro e alteração de domínios, pode ser encontradas em http://registro.br.

2.2.4 Consulta ao DNS

Assuma que um usuário invoca uma aplicação e fornece o nome da máquina com a qual a aplicação deve se comunicar. Antes de usar protocolos como UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*) para se comunicar, a aplicação deve encontrar o endereço IP da máquina destino.

A aplicação passa o nome da máquina para o RESOLVER local. O RESOLVER é responsável por enviar as questões para o servidor.

Primeiro ele pergunta ao servidor local para que este consulte o endereço. Neste ponto existem três possibilidades:

- O servidor local conhece o endereço, porque o endereço está na parte do servidor local da base de dados mundiais.
- O servidor local conhece o endereço porque alguém já pediu pelo mesmo endereço recentemente e o servidor o guardou no *cache* local.
- O servidor não conhece o endereço mas sabe como encontrá-lo.

Neste último caso, o servidor local encaminha a pegunta a um servidor responsável pelo domínio de mais alto nível. No caso o domínio “.”.

O servidor responsável pelo domínio “.” não possui toda a base de dados do DNS armazenada localmente, o que torna impossível a resposta direta a consulta feita pelo servidor local. Porém ele pode indicar o IP do servidor ao qual ele delegou a responsabilidade pelo domínio hierarquicamente abaixo.

O servidor local novamente encaminha a pergunta ao servidor informado pelo responsável pelo domínio hierarquicamente superior. Essa interação continua até que o servidor consultado seja responsável pelo domínio ao qual o *host* procurado pertença. Neste caso ele responderá qual o endereço IP ao servidor local que repassará a resposta ao *host* solicitante. Um exemplo de consulta é demonstrado na figura. 2.9

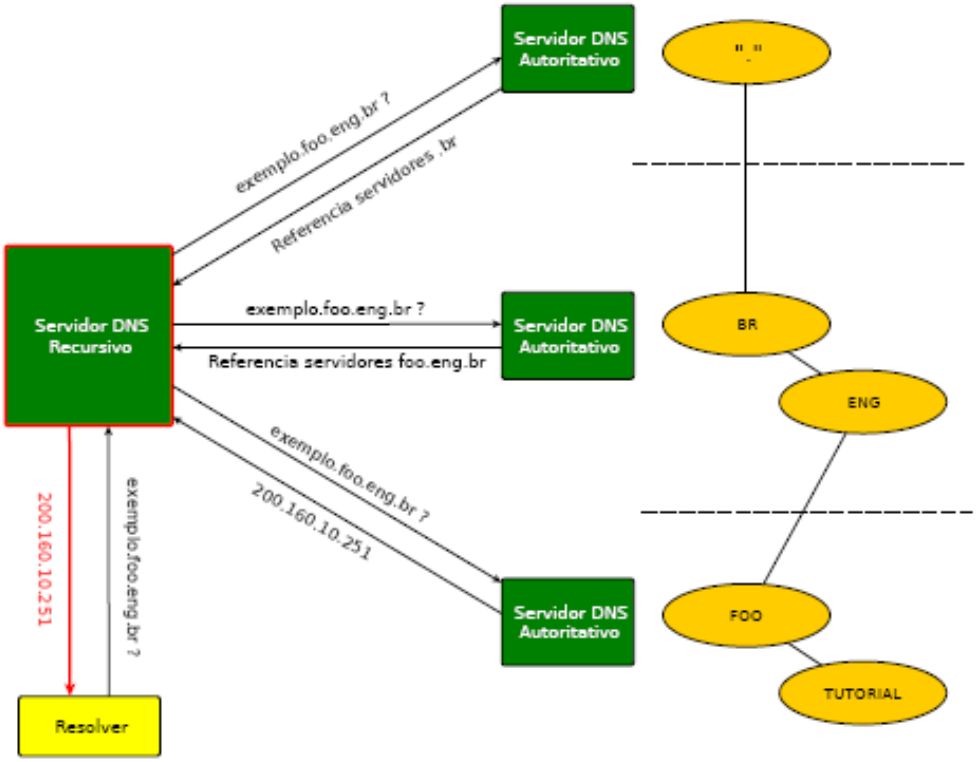

Figura 2.9: Consulta DNS

2.2.5 Formato da Mensagem

Mensagens DNS são unidades de dados que são transmitidas entre os servidores e RESOLVERS. As mensagens consistem em um cabeçalho e 4 secções como mostrado na figura 2.10.

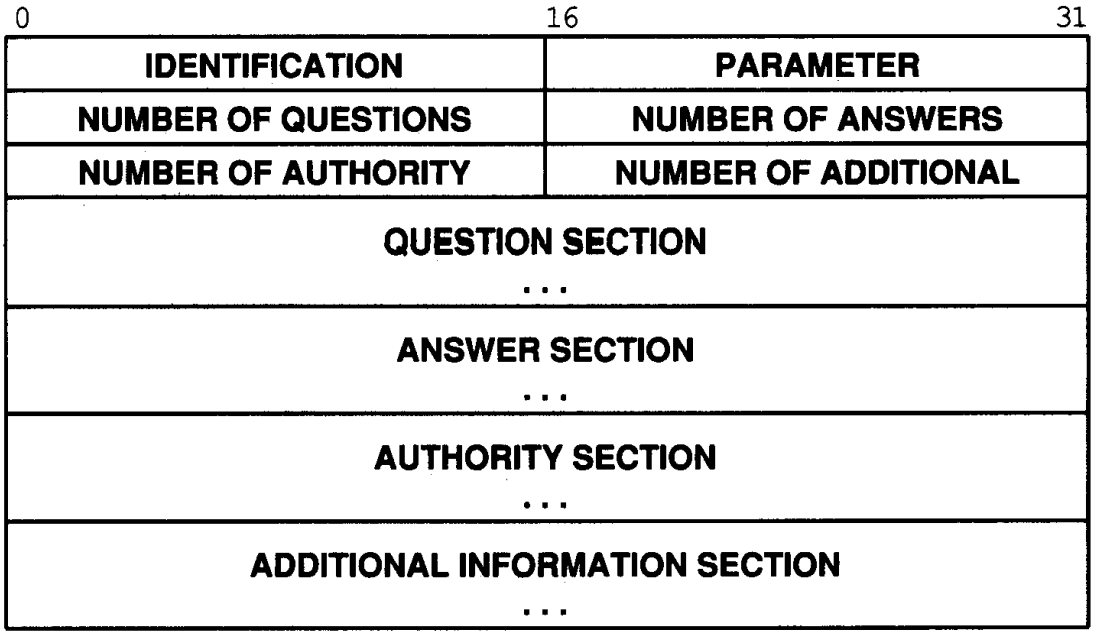

Figura 2.10: Cabeçalho de uma Mensagem DNS

O cabeçalho possui os seguintes campos:

- IDENTIFICATION – Número identificador gerado pelo programa servidor ou cliente.
- PARAMETER - Indica se a mensagem é uma pergunta ou uma resposta, além de outros parâmetros, como por exemplo, se houve erros na mensagem.
- QUESTION, ANSWER, AUTHORITY e ADDITIONAL – Servem para diferentes propósitos e as secções referentes a cada um dos números do cabeçalho, nem sempre são todas preenchidas. Neles são informados, por exemplo o objeto da consulta, o TIPO de objeto e o tempo de vida TTL (*Time To Live*) do mesmo. O tempo de vida é importante para designar quanto tempo um computador vai armazenar esta informação em sua cache.

O DNS é um banco de dados hierarquicamente distribuído que possui a relação <nome,IP>, porém este é apenas um tipo de informação armazenada neste banco. Outras informações são armazenadas e também podem ser consultadas usando o sistema de resolução de nomes. Algumas dessas informações são necessárias para o correto funcionamento do sistema, outras servem para fornecer novas funcionalidades.

Cada uma dessas informações está associada a um tipo de objeto diferente. Esses tipos de objetos são chamados de *Resource Records* (RR).

A Tabela 2.2 mostra uma lista de alguns RR mais utilizados no sistema de DNS:

SOA	Indica autoridade para os dados deste domínio
NS	Lista um servidor de nomes para este domínio
A	Mapeamento de nomes para endereço IP
PTR	Mapeamento reverso, ou de endereço IP para nome
CNAME	Nomes Canônicos, para Aliases
TXT	Informações Textuais
WKS	Well-Known Services
HINFO	Host Information
MX	Mail Exchanger, recebedor de e-mail do domínio

Tabela 2.2: Resource Records

2.2.6 Relação entre Servidores DNS

Quando um determinado domínio é delegado, este deve possuir um servidor primário e um ou mais servidores secundários. Isso é importante para manter a disponibilidade do sistema em caso de falha do servidor primário.

A relação entre servidores DNS de um mesmo domínio é feita de forma que toda a informação seja inserida, alterada ou retirada do primário e os secundários simplesmente consultam o primário e copiam essas informações. A pergunta que vem a mente neste momento é: de quanto em quanto tempo os secundários devem fazer estas consultas?

A resposta é: depende de qual frequência as informações deste domínio são alteradas. Se há alterações constantes, os secundários devem vir frequentemente ao primário atualizar suas informações. Na realidade, esses tempos são designados quando se estabelece um domínio num servidor DNS.

Ao se estabelecer um domínio primário num servidor DNS é preciso definir a temporização deste domínio com relação ao secundário ou em relação ao tempo de *cache* que os registros deste domínio devem ter. Os tempos em questão estão descritos na Tabela 2.3:

Refresh	De quanto em quanto tempo o secundário consultará o primário para saber se houve ou não modificação. Se houve modificação, o secundário atualiza os seus dados
Retry	Caso o secundário não consiga estabelecer conexão com o primário, com qual frequência ele ficará retentando. Observe que *retry* precisa ser menor que *refresh*.
Expire	Caso o secundário não consiga estabelecer conexão com o primário, quanto tempo este servidor ainda responderá pelo domínio em questão
TTL *default*	Este é o tempo que outras máquinas na Internet manterão em seus *caches* as informações obtidas deste servidor DNS sobre o domínio em questão. Observe que este tempo será para outras máquinas e não para o secundário do domínio

Tabela 2.3: Temporização de Registros DNS

Uma observação importante a se fazer é que um mesmo servidor DNS pode servir a vários domínios diferentes, tanto como primário quanto secundário. Essa temporização mostrada anteriormente diz respeito a um único domínio, ou seja, cada domínio deve ter sua temporização designada independentemente.

CACHE

Todo o processo de resolução de nomes no sistema de DNS possui um desempenho bem menor do que uma simples consulta a uma base de dados local como no antigo sistema baseado no arquivo HOSTS.TXT. Entretanto, o sistema DNS prevê uma forma de acelerar o tempo de resposta de uma consulta.

O servidor de nomes local precisa trocar várias mensagens para conseguir responder a uma consulta. Durante as sucessivas tentativas de resolução, o servidor mapeia informações sobre o espaço de nomes hierárquico do DNS. Essas informações já descobertas são armazenadas no *cache* local e podem ser usadas para futuras consultas. Na próxima tentativa de resolução de nomes, o servidor verifica se a informação já está armazenada no *cache*, em caso positivo, o tempo de resolução de nomes tem uma diminuição considerável.

Os três primeiros tempos da TABELA 2.3 dizem respeito à temporização entre servidores primário e secundário. O quarto tempo da tabela se refere a relação entre o servidor responsável pela zona DNS, que define o TTL, e o servidor local que pergunta.

A definição do tempo em que a informação deve permanecer no *cache* é complicada. Se for grande demais, a informação pode ficar desatualizada. Se for pequeno demais, pode-se causar o aumento do tráfego, o aumento de utilização do servidor final e o aumento do tempo de resposta.

2.2.7 DNS Dinâmico

O Sistema de DNS foi originalmente desenvolvido para suportar consultas a endereços estáticos de *hosts*. Mesmo que haja mudanças, é esperado que a frequência dessas mudanças seja baixa e que estas sejam editadas diretamente na base de dados do servidor primário.

Infelizmente, esse sistema não contempla uma classe de dispositivos: aqueles que possuem os seus endereços atribuídos dinamicamente por um servidor DHCP (*Dynamic Host Configuration Protocol*) a cada inicialização e/ou tempo pré-determinado. Esses dispositivos somente seriam alcançáveis pelo seu endereço atual e não poderiam oferecer os seus serviços na Internet de uma forma contínua através de um nome. Dessa forma, o protocolo DNS recebeu um adendo através da RFC2136 [37] que definiu uma forma eficaz de mapear dinamicamente um nome em um endereço junto ao protocolo DNS. Este adendo ficou conhecido como DNS dinâmico ou DDNS.

Deve-se considerar o DNS dinâmico como um grande avanço para a dissociação do endereço IP de um dispositivo de sua função, já que o mesmo permite que um nome no espaço de nomes DNS mude de endereço automaticamente.

O DNS dinâmico utiliza os mesmos campos do Formato das Mensagens DNS, e os mesmos formatos de seção, mudando apenas o uso destas seções.

Várias empresas, tais como DynDNS (www.dyddns.com), NoIP (www.noip.com) ou SuperDNS (www.superdns.com.br), fornecem o serviço de DNS dinâmico para os usuários da Internet. Principalmente para o mercado SOHO. A grande maioria funciona através de *scripts* que verificam periodicamente a mudança do IP. Assim que essa mudança é identificada, o *script* se conecta ao servidor DNS, se autenticado através de um usuário e senha pré-configurado, e atualiza este novo endereço.

Muitos equipamentos do mercado SOHO vêm hoje com a possibilidade de configuração do DDNS. Esses equipamentos possuem *scripts* prontos que possibilitam o acesso a um conjunto dessas empresas que fornecem o serviço de DNS dinâmico.

2.2.8 DNSSEC

O DNS é utilizado como infraestrutura básica para comunicação pela Internet. Isso o torna um ponto crítico sob o ponto de vista da segurança da comunicação de um sistema.

Um ataque a um sistema de DNS é uma das formas mais efetivas de se comprometer um sistema com comunicação baseada na Internet. Um atacante que consiga alterar uma resposta DNS passa a ter amplo leque de alternativas de ataque ao sistema alvo. Deve-se considerar que essa alteração poderá fazer com que os usuários de um domínio sejam direcionados para o endereço de servidores do atacante tornando vulneráveis serviços, tais como Web, correio eletrônico, VoIP (*Voice over Internet Protocol*) e VPN.

O sistema de DNS foi projetado como um banco de dados distribuído que funciona eficientemente, porém a segurança não foi um foco de atenção em seu desenvolvimento, o que acabou tornando-o vulnerável a uma série de ataques como descrito em [38].

Com o objetivo de elevar o nível de segurança do DNS foi criado o DNSSEC que utiliza criptografia nos dados armazenados e trocados entre as várias entidades do sistema.

DNSSEC é um conjunto de extensões ao DNS onde foram adicionados novos *Resource Records* que poderão ser usados por clientes DNS para validar a autenticidade e integridade de uma resposta DNS.

Para um administrador de uma zona DNS, o DNSSEC é utilizado no processo de assinatura de RRSets (*Resource Record Set*) com a chave privada, na publicação das assinaturas de cada RRSet no arquivo de zona e na publicação da chave pública. Além disso, é necessário que a sua chave pública seja assinada pelo administrador da zona hierarquicamente superior.

Já o cliente utiliza o DNSSEC para verificar a autenticidade e integridade das respostas DNS através das chaves públicas publicadas.

Foram criados 4 novos *Resource Records*: DNSKEY, RRSIG, NSEC e DS.

DNSKEY

Cada zona DNS possui um par de chaves: pública e privada, gerada pelo administrador. A chave privada permanece armazenada em segredo pelo administrador e a chave pública é armazenada no arquivo de configuração da zona através do RR DNSKEY.

RRSIG

Um RRSet é um conjunto de RRs de uma zona DNS que possuem um mesmo nome, classe, mas os dados são diferentes . A assinatura de um RRset provê maior granularidade ao DNSSEC, pois não é preciso assinar e realizar a assinatura da zona DNS inteira.

NSEC

O registro NSEC foi projetado para mostrar que não existe nenhum registro entre dois registros válidos. Caso um atacante tente personificar um registro de uma determinada zona DNS, o cliente pode realizar uma consulta por este registro e recebe como resposta um registro NSEC informando dois registros válidos e mostrando que não existe o registro consultado entre eles.

O problema do Registro NSEC é que ele pode ser utilizado para enumerar todos os registros de uma determinada zona DNS.

Para resolver isso, foi criado um novo registro NSEC3 que contém apenas o *hash* dos registros e não mais os dados.

DS

É um *hash* da chave DNSKEY de uma zona hierarquicamente inferior. Esse registro indica que a Zona delegada está assinada e qual é a sua chave. O Registro DS é um ponteiro para a cadeia de confiança, a qual garante a autenticidade das delegações de uma zona DNS.

Além das extensões criadas, a RFC 3007 [39] descreve a forma de se fazer atualizações dinâmicas seguras em um DNSSEC.

Foram criados dois novos registros para requisições e respostas autenticadas. O registro TSIG [40] que utiliza o sistema de algoritmo simétrico com chave compartilhada e o registro SIG(0) [41], [42] que utiliza algoritmo assimétrico com chave pública e privada. Em ambos os casos é incluída uma assinatura no final de uma mensagem DNS.

Para um *host* atualizar um registro no servidor DNS, ele envia uma requisição de atualização cifrada e assinada utilizando um dos métodos TSIG ou SIG(0). O servidor DNS recebe o pedido de atualização, verifica em sua política se o solicitante possui o direito de atualizar aquele registro, confere a autenticidade da mensagem e, por fim, atualiza o registro.

O registro SIG(0) possui a vantagem de facilitar a distribuição das chaves, já que utiliza o esquema de chaves pública e privada, porém aumenta a complexidade e diminui o desempenho do sistema comparando com a utilização do registro TSIG.

Ambos os métodos poderiam ser usados, mas neste trabalho optou-se por utilizar o registro TSIG em função de sua simplicidade.

2.3 Conclusão

O objetivo deste capítulo foi introduzir o conceito de VPN e DNS e situar a utilização destas tecnologias em um cenário de interligação segura de pequenas empresas por VPN pela Internet, baseado em um sistema de endereçamento IP dinâmico. A tecnologia de acesso a Internet escolhida para ser utilizada neste trabalho será a ADSL em função de sua maior utilização no Brasil. Neste cenário, o sistema deve ser capaz de estabelecer uma topologia de VPN do tipo *Partial-Mesh*, pois ela oferece uma maior flexibilidade com relação ao fluxo de informação e necessidades de desempenho das conexões.

Uma análise nos tipo de VPN nos mostra que é necessária a utilização de VPN do tipo *Trusted* com o protocolo IPsec, baseadas em CPE. A VPN *Trusted* é a única que utiliza criptografia no transporte de dados e o IPsec por possuir melhor desempenho e ser padrão de mercado. Além disso a VPN baseada em CPE oferece proteção fim-a-fim da comunicação.

Como arquitetura dos gateways neste trabalho será usado o gateway VPN junto ao filtro de pacotes e Proxy Web, pois possui um elevado nível de segurança, sem grande perda de desempenho para o cenário proposto e com baixo custo de implementação.

O sistema também utilizará do serviço de DNS como infraestrutura básica para identificação e autenticação dos gateways remotos que possuem endereçamento IP dinâmico. Para isso é necessário a implementação das funções de DDNS e DNSSEC no mesmo servidor, em especial os registros do tipo TSIG para requisições e respostas DNS autenticadas.

Capítulo 3: Trabalhos Relacionados

Para a construção do sistema proposto, foi realizado um estudo inicial das possíveis formas de implementação da tecnologia de VPN e como elas podem ser utilizadas em conjunto com as formas de acesso à Internet disponíveis .

Uma análise da tecnologia de VPN usada no contexto de acesso remoto é realizada em [43], [44] e [45] que serviu como um importante ponto de partida para uma análise mais aprofundada do problema. Uma outra pesquisa interessante é feita por [46] que mostra como integrar a VPN numa configuração de *firewall*.

Porém esses trabalhos mostram uma arquitetura de VPN do tipo *host-to-gateway* ou *RoadWorrior-to-Gateway*, onde a VPN é estabelecida entre um *host* e um *gateway*.

Também é mostrado uma adaptação do modelo *RoadWorrior-to-Gateway* usando IP dinâmico para estabelecimento de uma VPN do tipo *gateway-to-gateway*. Neste caso, pelo menos um dos *gateways* possui endereço IP fixo e os túneis VPN são sempre estabelecidos a partir do *gateway* com endereço dinâmico.

Esse modelo é bastante usado e confiável, pois normalmente as tecnologias de conexão à Internet com endereçamento IP fixo possui um alto índice de disponibilidade. Inclusive as operadoras ficam sujeitas a multas contratuais pelo não cumprimento das metas de disponibilidade.

Esse modelo também é de fácil implementação, mas possui algumas limitações pois só permite a interligação de VPNs do tipo *hub-and-spoke.* Além disso, não é possível utilizá-la numa situação onde todos os *gateways* possuem endereço IP dinâmico.

Para se estabelecer uma VPN nessa situação seria necessário identificar o endereço IP de ambas as pontas, realizar a configuração e estabelecer o túnel do tipo *gateway-to-gateway* baseado em dois endereços IPs fixos. Porém quando pelo menos um dos endereços muda a VPN fica indisponível até que uma reconfiguração seja realizada.

O maior problema nessa situação não é realizar a reconfiguração e o novo estabelecimento do túnel, mas sim como identificar que um dado endereço mudou e qual é este novo endereço, principalmente se a mudança ocorreu no *gateway* remoto.

Uma solução para esse problema surgiu com a definição de extensões para atualização dinâmica de registros DNS [37]. Essas extensões permitem que um *host* modifique um

determinado registro dentro de uma zona DNS. O objetivo é permitir que ao mudar o endereço de um *host* que possui IP dinâmico, ele próprio seja capaz de atualizar o seu registro dentro da zona DNS.

A utilização do DDNS permite que através de consultas ao servidor DNS, um *gateway* descubra o novo endereço do *gateway* remoto. Além disso ele próprio é capaz de informar a sua mudança de endereço.

Produtos de mercado passaram a utilizar DNS dinâmico para reconfiguração das VPNs entre dois *gateways* com endereçamento IP dinâmico [47]. Esses produtos normalmente possuem *scripts* que monitoram o seu endereço IP e o endereço IP do *gateway* remoto. Se um dos endereços foi modificado, o *gateway* altera os arquivos de configuração e tenta estabelecer novamente a VPN. Caso tenha sido modificado o seu próprio endereço, ele envia a atualização ao servidor DNS onde está cadastrado para que o *gateway* remoto identifique a alteração.

O problema dessa abordagem é a forma de atualização dos registros DNS no servidor. O servidor permite que um determinado registro de uma zona DNS seja alterada apenas por determinados *hosts*, mas a autenticação deste *host* é feita através de usuário e senha que trafega em texto claro entre o *gateway* e o servidor.

Essa forma de autenticação permite que seja realizado um ataque do tipo DNS *spoofing*. Um atacante poderia capturar a informação de usuário e senha e modificar o registro no servidor DNS. Quando o *gateway* remoto realizar a consulta daquele registro, tentará estabelecer um túnel com o endereço modificado pelo atacante, o que causará indisponibilidade da VPN caracterizando um ataque de DoS. Além disso, o atacante poderá obter informações que possibilitem outros tipos de ataques.

Em função dos problemas de segurança com a utilização do DNS dinâmico, fabricantes de *gateways* lançaram soluções que permitem a utilização de IP dinâmico para todos os *gateways* VPN como visto em [48] e [49]. Quando um *gateway* tem seu endereço IP modificado, ele atualiza este novo endereço em um sistema centralizado de gerenciamento da solução, como pode ser visto na figura 3.1. Os dados da atualização são enviados de forma cifrada, o que confere integridade e autenticidade à transação, evitando assim os ataques de DNS *spoofing* e DoS vistos anteriormente.

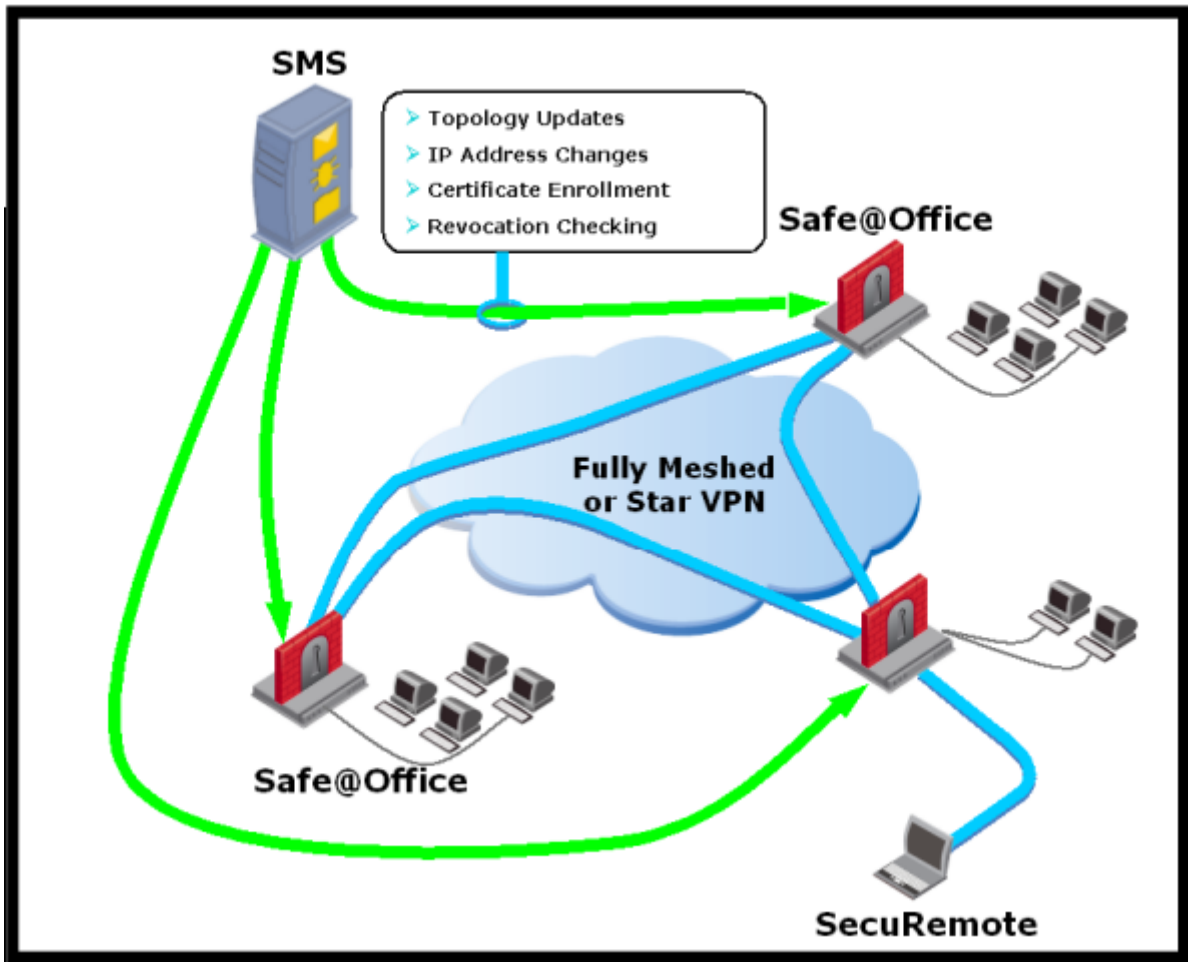

Figura 3.1: Sistema Centralizado de Gerenciamento – Retirada de [48]

O objetivo dos fabricantes é fornecer ao cliente uma solução que integre os diversos mecanismos de segurança um uma visão unificada. A arquitetura do sistema deve permitir o compartilhamento informações entre os seus diversos elementos de forma a aumentar o nível global de segurança. Um exemplo de uma arquitetura como esta esta descrita em [49] e pode ser vista na figura 3.2.

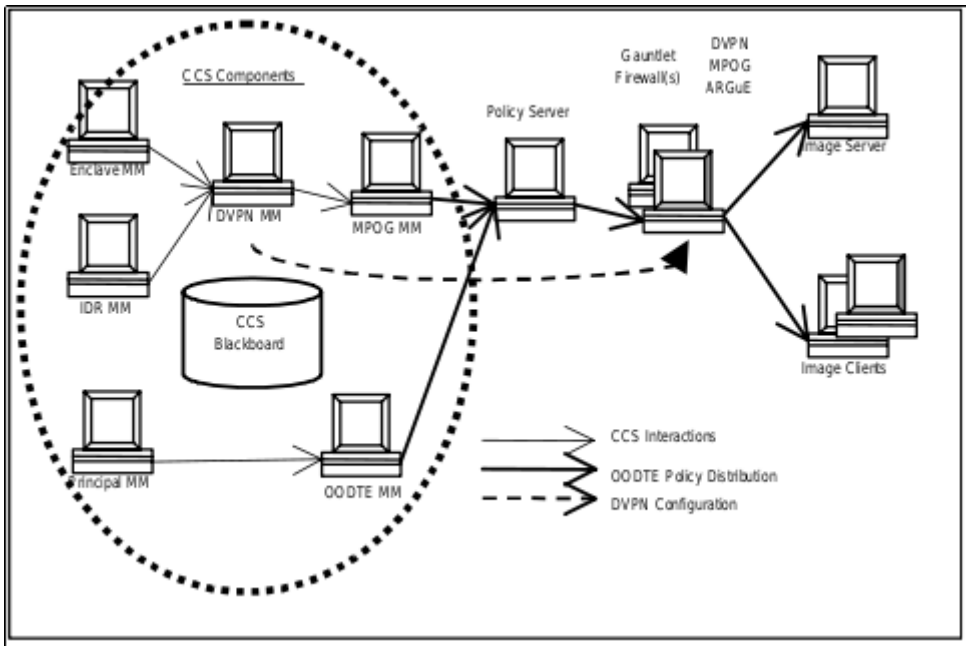

Figura 3.2: Arquitetura NAI – Retirada de [49]

Apesar de essa solução propiciar o funcionamento com segurança para o cenário proposto de interligação de redes através de VPN, utilizando endereçamento IP dinâmico, as soluções apresentadas possuem um problema. A tecnologia utilizada na solução é baseada em serviço e protocolos proprietários. Apenas com a utilização dos *gateways* e do sistema de gerenciamento centralizado do mesmo fabricante é possível a implementação da solução.

Uma solução com gerenciamento descentralizado é proposto por [50]. Neste trabalho foi criado um espaço de nomes e endereços IP virtuais que serão usados dentro do contexto do estabelecimento de VPNs dinâmicas. Também foram criados componentes para roteamento, filtragem e um driver para captura dos dados relativos a dados VPN trafegados. A arquitetura simplificada do trabalho pode ser visto na figura 3.3

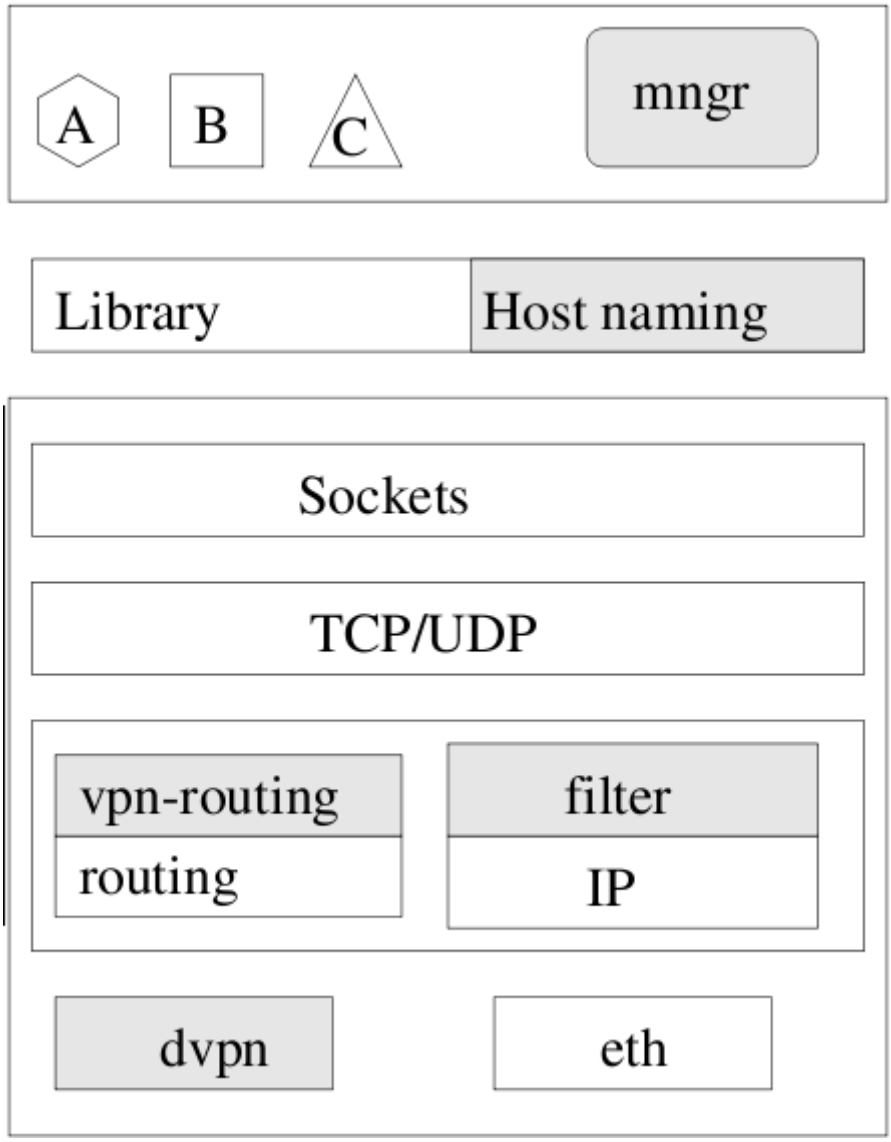

Figura 3.3: Arquitetura DVPN – Retirada de [50]

A solução fornece um alto nível de segurança, pois permite o controle granular de todo o tráfego de informações referente a VPN além da característica de gerenciamento descentralizado tornar o sistema mais robusto e tolerante a falhas. Uma indisponibilidade em qualquer um dos nós não compromete a conexão dos outros *gateways*. Porém a necessidade de modificações no sistema operacional, a complexidade dos componentes e uma diminuição do desempenho da solução dificultam a sua implementação em larga escala para o cenário proposto neste trabalho.

A utilização do serviço de DNS pode ser um risco como visto em [51], [52] e [38], devido a falhas de configuração e ausência de elementos de segurança no protocolo.

Entretanto, a utilização do DNSSEC [41], como nos trabalhos [53] e[54], torna promissor o uso desta tecnologia como base para implementação de uma arquitetura segura para o sistema proposto.

Uma visão da utilização do DNSSEC em um futuro próximo é abordada em [55]. E o uso do DNSSEC em um contexto de atualizações dinâmicas foi abordado em [39].

O trabalho [16] descreve um projeto onde o autor utiliza o protocolo IPsec e DNSSEC para estabelecer túneis VPN entre dois *hosts* quaisquer. O objetivo é estabelecer uma comunicação cifrada entre dois pontos sem que seja feita nenhuma configuração prévia entre as partes envolvidas.

Mas, da mesma forma como [56], não é previsto nos trabalhos a modificação dos IPs no estabelecimento dos túneis.

Diferentemente dos trabalhos citados anteriormente, a abordagem proposta nesta dissertação de mestrado, consegue implementar um sistema de VPN baseado em endereçamento IP dinâmico, mantendo um alto nível de segurança e com a flexibilidade que a utilização de padrões abertos proporciona.

São características importantes do sistema proposto:

- A utilização de um servidor DNS dinâmico (DDNS) e seguro (DNSSEC) como ponto central de armazenamento e consulta dos endereços IP para o estabelecimento da VPN.
- O uso de protocolos padronizados e abertos na construção do sistema para garantir a interoperabilidade da solução.
- A construção do sistema baseado em *software* livre, o que proporciona flexibilidade à solução e a possibilidade de aprimoramento e implementação em outros equipamentos.
- Reconfiguração e estabelecimento automático das VPNs.

Características essas que não foram encontradas em conjunto nos trabalhos mostrados neste capítulo e representam a principal contribuição desta dissertação.

Os detalhes da abordagem proposta neste trabalho são mostrados no capítulo seguinte.

Capítulo 4: Proposta e Implementação

Hoje, as empresas podem criar VPNs para transportar dados privados entre redes geograficamente distribuídas através da infraestrutura da Internet. Provedores de serviço oferecem acesso rápido (alta velocidade de transmissão) à Internet através da tecnologia ADSL e com um sistema de endereçamento IP (Internet Protocol), utilizando a infraestrutura de telefonia fixa. Essas facilidades possibilitam a transmissão de dados privados pela Internet através de VPN *gateways*, que encapsulam os dados para manter a privacidade, autenticidade e integridade dos dados dentro do túnel VPN. Dado que os VPN *gateways* acessam a Internet através de uma linha telefônica via ADSL e que desempenham o papel de *firewall* de borda, pode-se implementar uma rede privada eficiente, mais segura e com um custo menor se comparado com a locação de canais dedicados. A Figura 4.1 ilustra um ambiente de conectividade desse tipo.

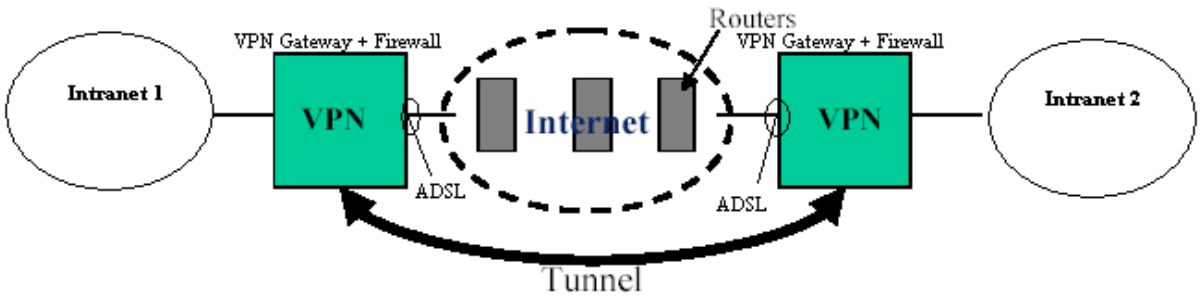

Figura 4.1: VPN

Existem no mercado soluções de *hardware* e *software* que implementam o ambiente de conectividade mostrado na Figura 4.1, entretanto tais soluções possuem alguns problemas como visto no capítulo anterior. O primeiro problema é a diminuição da praticidade e, consequentemente, da disponibilidade da solução ao se utilizar a reconfiguração manual dos parâmetros de VPN. O segundo problema é com a segurança da comunicação ao se utilizar o serviço de DDNS para atualizar os parâmetros de configuração da VPN. O terceiro problema é o custo e a falta de flexibilidade ao utilizar soluções proprietárias para implementação do ambiente proposto.

A solução aqui apresentada, implementa o ambiente da Figura 4.1 com sistema de endereçamento IP dinâmico e reconfiguração automática das VPNs, ficando transparente para o usuário final a mudança de endereços por parte do provedor. Um *firewall* também é incorporado no VPN *gateway* a fim de aumentar o nível de segurança contra invasão de intrusos na rede. A utilização de *firewalls* nesse tipo de ambiente é extremamente importante dado o grande número de ataques provenientes da Internet. Toda solução é baseada em protocolos abertos e a implementação utiliza apenas *software* livre. Isto garante a interoperabilidade dos equipamentos com outros *gateways* e servidores.

4.1 Arquitetura

A Figura 4.2 mostra a arquitetura simplificada da solução. O sistema é composto por 2 *gateways* e um servidor de DNS. Os *gateways* acessam a Internet através de uma conexão ADSL de IP dinâmico. Para estabelecer a VPN, todos os *gateways* fazem uma consulta ao servidor DNS sobre o *gateway* da outra ponta da VPN. De posse do IP respondido pelo servidor, o serviço de VPN é configurado e iniciado. Uma vez estabelecido o túnel entre dois *gateways*, qualquer máquina de uma das redes pode falar com qualquer máquina da outra rede, desde que a conexão seja permitida pela política aplicada.

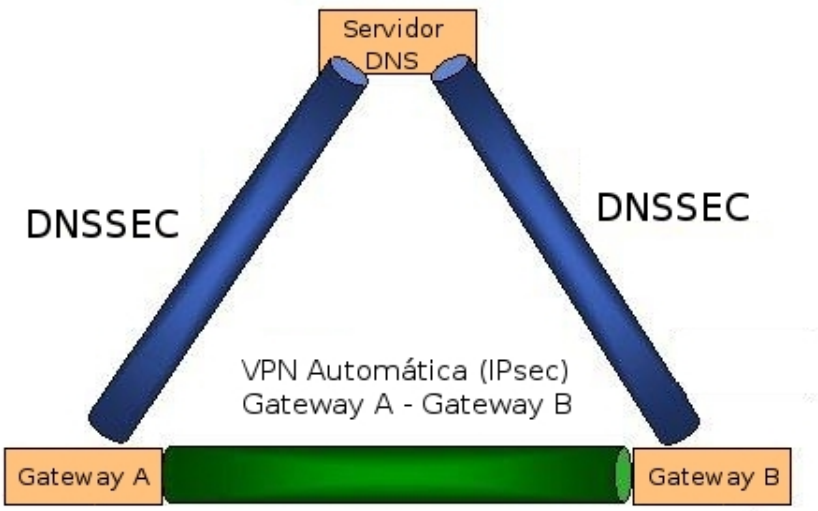

Figura 4.2: Arquitetura

Os *gateways* VPN monitoram o IP válido de sua interface fornecido pela operadora. Caso ele identifique que houve modificação no endereço, a VPN é reconfigurada, o servidor de DNS é atualizado e a VPN é reiniciada com as novas configurações. Este processo de automação das configurações e estabelecimento da VPN, evita que o sistema tenha que ser atualizado manualmente sempre que houver uma mudança no endereçamento IP, aumentando consequentemente a disponibilidade da solução.

Para atualização do endereço, os *gateways* enviam uma atualização do registro do tipo A referente ao endereço do próprio *gateway* que está em uma zona DNS armazenada no servidor. Essa zona DNS possui todos os registros que poderão ser atualizados dinamicamente. Para garantir que apenas os *gateways* possam atualizar os seus próprios registros, o servidor DNS é configurado para aceitar apenas requisições autenticadas, usando algoritmos e chaves preestabelecidas

O servidor DNS é um equipamento essencial para o funcionamento do sistema. O servidor precisa ter acesso dedicado à Internet com o endereço IP fixo. Assim o servidor pode servir de referência para todos os *gateways* VPN atualizarem as informações que são modificadas dinamicamente pelos demais *gateways* e também atualizar seus próprios registros.

É necessário que o servidor DNS funcione como DDNS e DNSSEC simultaneamente. Isto garante a autenticidade e a integridade na atualização e requisições dos registros no servidor. Dessa forma, resolve-se a vulnerabilidade do sistema a ataques do tipo DNS *spoofing* como visto no capítulo anterior, diminuindo a chance de ocorrer uma negação de serviço na VPN .

A figura 4.3 mostra a arquitetura dos *gateways* VPN utilizados na solução. Os *gateways* possuem um conjunto de componentes e configurações específicas que tem por objetivo o aumento tanto do nível de segurança do próprio equipamento quanto dos dados trafegados. Uma primeira camada de segurança foi introduzida através de configurações específicas no próprio Sistema Operacional. O primeiro passo para se ter este aumento de segurança foi realizado através de configurações específicas no sistema operacional. Estas configurações tornam o sistema operacional menos suscetível a falhas de configuração dos demais componentes do *gateway* ou a operação errada do ambiente.

O sistema de arquivos possui um verificador de integridade que analisa periodicamente mudanças não previstas. Caso ocorra uma mudança em algum arquivo ou

diretório em que esta modificação não era prevista, este evento é registrado e o administrador do sistema é avisado.

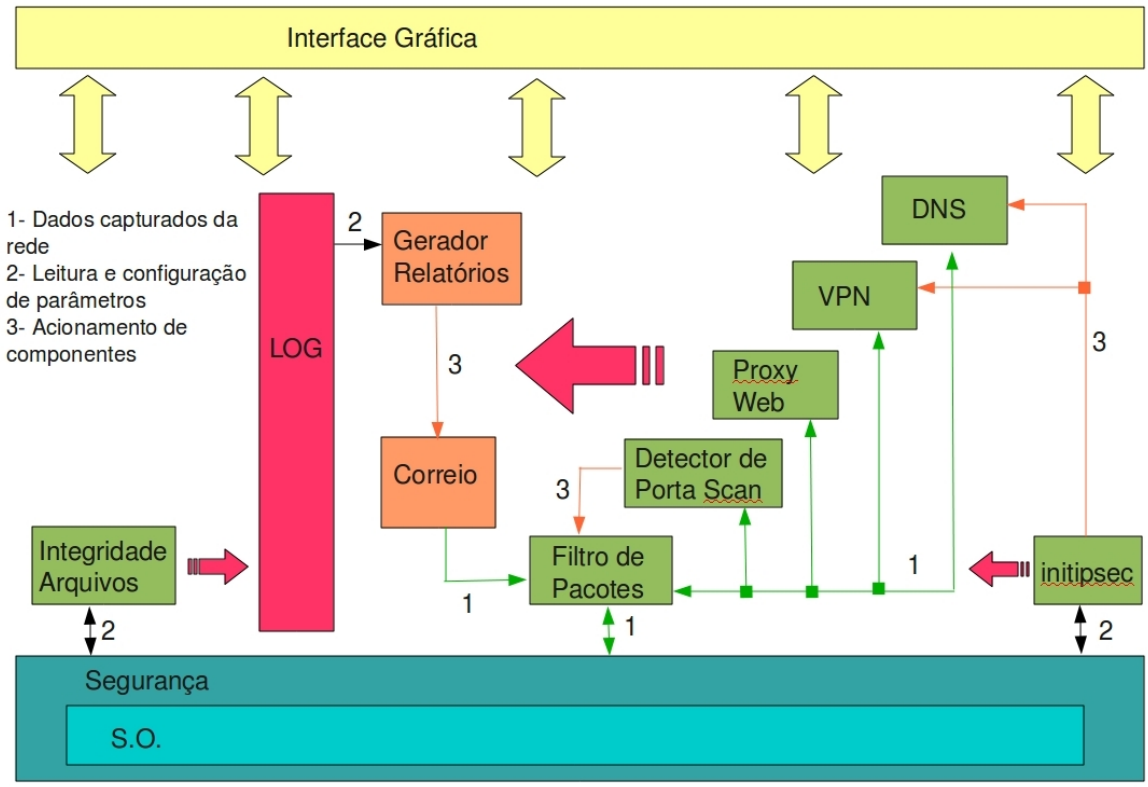

Figura 4.3: Arquitetura do Gateway

Para o controle dos dados que passam pelo *gateway* é utilizado um filtro de pacotes. Este filtro deve ter suas regras configuradas de acordo com a política de segurança da instituição. Os pacotes são capturados pelo sistema operacional e enviados para o filtro de pacotes. Ele é responsável por avaliar se este pacote será bloqueado, enviado a outro componente do sistema, ou encaminhado para outro *host*, repassando ao sistema operacional. É importante frisar que todo tráfego de entrada ou saída da rede passa necessariamente pelo filtro de pacotes.

Normalmente o início de um ataque a um *gateway* se dá através de uma verificação de que portas TCP/UDP estão respondendo a requisições externas. A partir desta verificação é feita uma análise se há vulnerabilidades nos serviços correspondentes a estas portas. Para evitar este tipo de ataque, todo o tráfego que passa pelo filtro de pacotes também é

direcionado para o detector de *Port Scan*. Caso um s*can* seja identificado, o detector adiciona uma regra de bloqueio da origem do ataque no filtro de pacotes.

Como vimos no Capítulo 2, o tráfego *web* normalmente é o de maior volume de acesso dentro de uma instituição, e também o que possui o maior número de vulnerabilidades conhecidas. Por isso se faz necessário um nível maior de proteção a este tráfego, conseguido através de um *proxy web*.

Os componentes de VPN e DNS são responsáveis pelo estabelecimento da VPN entre os *gateways* de forma segura. O controle, monitoramento e automatização destes elementos é feito pelo initipesec. O algorítimo implementado neste componente é executado periodicamente e verifica alguns parâmetros do sistema operacional e dos servidores de VPN e DNS. Como resultado da análise ele realiza a reconfiguração e acionamento destes componentes. Este algorítimo e sua implementação serão analisados detalhadamente no decorrer deste capítulo.

Todos os componentes da arquitetura do gateway, geram registros no log do sistema. Um gerador de relatórios, faz uma leitura destes registros e sintetiza em relatórios que são enviados por e-mail ao administrador do sistema ou acessados através da interface gráfica do sistema. Além de fornecer acesso aos relatórios gerados, a interface gráfica foi criada de forma a possibilitar a configuração e leitura de todos os parâmetros necessários ao pleno funcionamento dos componentes descritos na figura 4.3, além do acionamento e parada destes componentes.

Todo o sistema é baseado em *software* livre adicionando flexibilidade ao sistema, pois permite a interoperabilidade dos *gateways* com qualquer equipamento que possua solução similar e a possibilidade de implementação desta solução com outros equipamentos. Ao contrário de outras soluções proprietárias, não é necessário que os *gateways* utilizados ou o elemento centralizador dos registros sejam do mesmo fabricante.

4.2 Algoritmo

O algoritmo utilizado para resolver o problema dos endereçamento IP dinâmico e reconexão automática segue os seguintes passos: o *gateway* verifica se está conectado à Internet através de uma conexão ADSL, se não estiver, tenta fazer uma nova conexão. Caso esteja conectado, inicia o processo de autenticação. Havendo falha no processo de

autenticação o algoritmo é finalizado. Após o acesso à Internet estar autenticado, o *gateway* verifica se o endereço IP local é diferente do último endereço IP recebido pelo provedor. Caso o endereço seja diferente, o *gateway* enviará uma atualização para o servidor DNS contendo o novo endereço IP. O próximo passo é consultar no servidor DNS os endereços de todos os *gateways* VPN. Se não for possível consultar o servidor DNS o algoritmo é finalizado. Caso tenham mudado quaisquer dos endereços, o *gateway* atualiza os arquivos de configuração. Nesse ponto o algoritmo verifica se as VPN estão estabelecidas. Caso não estejam, seja pela reconfiguração ou por outra falha qualquer, os túneis são reiniciados. Passado um intervalo de tempo preestabelecido, o algoritmo é executado novamente. A Figura 4.4 mostra o algoritmo como um fluxograma.

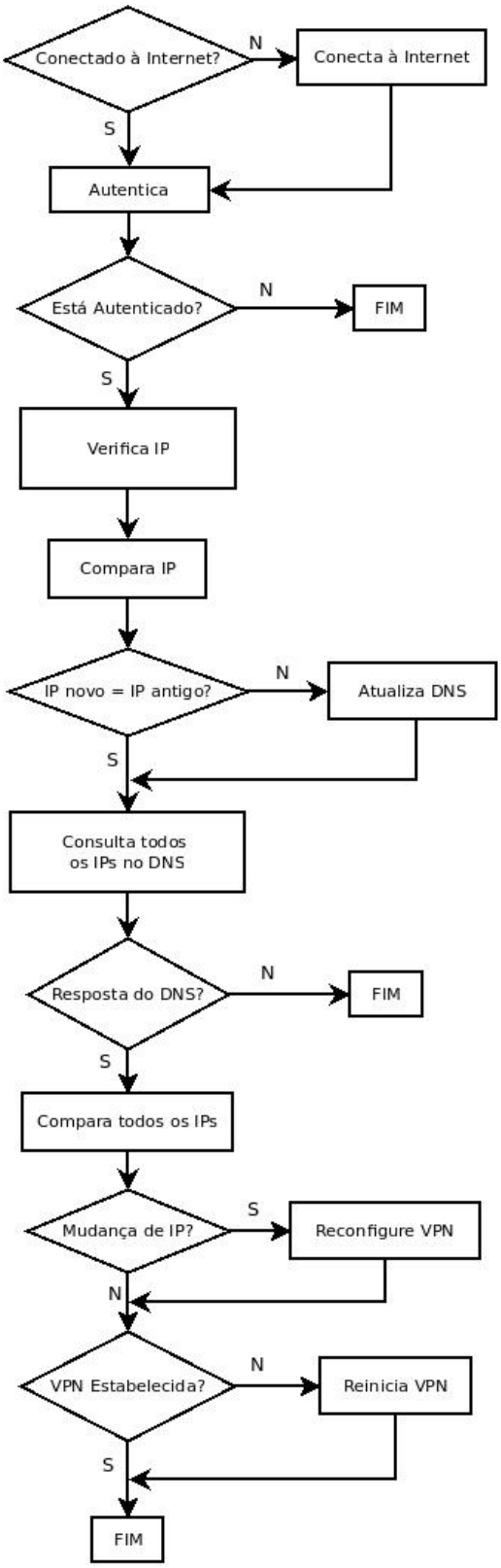

Figura 4.4: Algoritmo

O fato de o algoritmo ser executado periodicamente implica em em alguns cuidados em sua construção. Considerando que o *gateway* já esteja conectado e com as VPNs estabelecidas, ao se executar novamente o algoritmo, este não deve interferir no funcionamento normal do sistema.

Outro ponto importante ocorre na atualização do registro DNS. Logo após a atualização são consultados todos os endereços dos *gateways* VPN, inclusive do próprio *gateway*, pois caso haja alguma falha e a atualização não tenha sido feita, uma nova comparação é realizada com o endereço requisitado ao servidor DNS. Nessa comparação, não será detectada a mudança de endereço e não será feita a reconfiguração das VPNs. Numa nova execução do algoritmo a mudança de endereço será detectada novamente e feita uma nova tentativa de atualização e reconfiguração das VPNs.

4.3 Implementação

4.3.1 Configuração do cliente

Para implementar o algoritmo descrito, foi criado o programa initipsec. Este programa é executado, por padrão, a cada minuto, de forma a fazer as devidas verificações se algum endereço IP (próprio ou de algum *gateway* remoto) foi modificado. Na construção dos *scrpts*, os *gateways* VPN são identificados como MBOX. Os nomes destes *gateways* são compostos por MBOX seguido de um número de identificação. O servidor de DNS será identificado a partir de agora como MBOX1.

O initipsec faz uso de uma série de arquivos que são usados na configuração de parâmetros internos, de sistema e programas externos. A seguir, são apresentados os arquivos utilizados pelo programa e uma breve descrição.

- `initipsec.conf` – arquivo de configuração do programa initipsec
- `addr.conf` – arquivo contendo os endereços IP atuais dos MBOXs
- `ipsec.wrk` – arquivo de apoio para configuração da VPN
- `ipsec2.wrk` – arquivo de apoio para configuração da VPN
- `dnsupdate` – script utilizado para atualização do endereço IP no servidor DNS.

initipsec.conf

O arquivo */etc/mbox/initipsec.conf* armazena as configurações do initipesc e deve ser criado antes de sua execução. Este arquivo pode conter comentários e deve listar algumas variáveis, assim como seus valores. As variáveis são utilizadas para indicar os gateways remotos e outros parâmetros para configuração das VPNs. Segue a descrição das variáveis deste arquivo de configuração.

- **MBOX1:** armazena o endereço IP do servidor de DNS dinâmico;
- **ENDPOINTS:** armazena os nomes dos *endpoints* dessa máquina MBOX, ou seja, os nomes dos *gateways* remotos com os quais esse MBOX se conecta. Os nomes dos *endpoints* são separados apenas por uma vírgula;
- **SIDE:** a cada máquina em uma conexão atribui-se uma posição a ela, esquerda ou direita. Esta variável armazena casa posição das outras máquinas as quais o MBOX se conecta. Esta variável é utilizada para reescrever o arquivo de configuração da VPN. Os valores, **left** ou **right**, são separados apenas por uma vírgula;
- **TTL:** time-to-live, usado na atualização do registro no servidor de DNS. O padrão utilizado é de 2 minutos;
- **HOSTNAME:** nome deste MBOX;
- **SLEEPTIME**: tempo, em segundos, entre cada execução do *initipsec*. O Padrão utilizado é de 1 minuto.
- **DOMAIN**: domínio em que este MBOX está registrado.
- **KEYFILE**: arquivo com a chave para atualização do registro no servidor DNS.

Cada linha do arquivo, excetuando-se as linhas de comentários, deve terminar com um ponto e vírgula (*;*). Comentários podem ser adicionados iniciando a linha com o caractere tralha (#).

A seguir, encontra-se um exemplo do arquivo *initipsec.conf*.

_____/etc/mbox/initipsec.conf_____

```
# Comentarios começam/terminam com o carácter sharp.  #
# Este arquivo possui a seguinte estrutura:           #
# <nome da variavel>=<valor>;                         #
# exemplo:                                            #
# LOCALHOST=127.0.0.1;                                #
```

```
# Os nomes das variaveis devem estar em letras maius- #
# culas e so podem ser constituidas por letras e      #
# numeros. Cada nome de variavel pode ter ate 20      #
# caracteres.                                         #
MBOX1=200.217.17.28;
ENDPOINTS=zeus,mbox2,mbox3;
SIDE=right,right,left;
TTL=120;
HOSTNAME=mbox0;
SLEEPTIME=60;
DOMAIN=mbox.com.br
KEYFILE=/etc/Kmbox.maxima-ti.com.br.+157+41388.private;
```

_____/etc/mbox/initipsec.conf_____

addr.conf

O arquivo */etc/mbox/addr.conf* armazena o endereço IP atual deste MBOX, do NextHop que indica o roteador pelo qual passam os túneis VPN, do servidor de DNS, e também dos endereços IP de cada máquina a que este MBOX se conecta através de uma VPN.

O arquivo não necessita ser previamente criado, pois o initipsec irá criá-lo sempre que não for encontrado.

O initipsec reescreve este arquivo sempre que qualquer um dos endereços IP mudar, ou se forem adicionados novos endpoints a este MBOX através do arquivo de configuração `initipsec.conf`, descrito anteriormente.

O arquivo segue o seguinte modelo:

```
<nome do endpoint>=<endereço IP>;
```

O arquivo pode ser apagado a qualquer momento e por qualquer motivo, como por exemplo, para forçar o initipsec atualizar todos os endereços IP e os arquivos de configuração da VPN IPSec (*`/etc/ipsec.conf`* e *`/etc/ipsec.secrets`*).

Abaixo se encontra um exemplo do arquivo *`/etc/mbox/addr.conf`*.

___/etc/mbox/addr.conf___

```
MeuIP=10.0.0.30;
```

```
NextHop=10.0.0.30;
zeus=200.165.47.57;
mbox1=200.217.17.28;
```

___/etc/mbox/addr.conf___

ipsec.wrk e ipsec2.wrk

Esses arquivos de configuração são utilizados como base para modificação dos arquivos de configuração da VPN IPsec. Os aquivos ipsec.wrk e ipsec2.wrk são usados em conjunto com os arquivos initipsec.conf e addr.conf, para recriar respectivamente os arquivos de configuração ipsec.conf e ipsec.secrets, sempre que ocorre uma mudança de endereço IP em um dos gateways envolvidos no estabelecimento dos túneis VPN.

__/etc/mbox/ipsec.wrk__

```
# /etc/ipsec.conf - FreeS/WAN IPsec configuration file

# More elaborate and more varied sample configurations can be found
# in FreeS/WAN's doc/examples file, and in the HTML documentation.



# basic configuration
config setup
      # THIS SETTING MUST BE CORRECT or almost nothing will work;
      # %defaultroute is okay for most simple cases.

      interfaces="ipsec0=ppp0"

      # Debug-logging controls:  "none" for (almost) none, "all" for
lots.

      klipsdebug=none
      plutodebug=none

      # Use auto= parameters in conn descriptions to control startup
actions.

      plutoload=%search
      plutostart=%search
```

```
            # Close down old connection when new one using same ID shows
up.
            #uniqueids=yes


      # sample VPN connection
      conn zeus-mbox2
            # Left security gateway, subnet behind it, next hop toward
right.

            left=internet
            leftsubnet=192.168.11.0/24
            leftnexthop=interhop
            # Right security gateway, subnet behind it, next hop toward
left.

            right=gegenstation
            rightsubnet=10.0.0.0/24
            rightnexthop=200.217.31.155

            # To authorize this connection, but not actually start it, at
startup,
            # uncomment this.

            keyingtries=0
            auth=ah
            authby=rsasig
            leftrsasigkey= chave pública de mbox2
            rightrsasigkey= chave pÃºblica de zeus
            auto=add


      # sample VPN connection
      conn mbox2-mbox3
            # Left security gateway, subnet behind it, next hop toward
right.

            left=internet
```

```
            leftsubnet=192.168.11.0/24
            leftnexthop=interhop
            # Right security gateway, subnet behind it, next hop toward
left.

            right=otherside
            rightsubnet=192.168.10.0/24
            rightnexthop=200.217.31.155

            # To authorize this connection, but not actually start it, at
startup,
            # uncomment this.

            keyingtries=0
            auth=ah
            authby=rsasig
            leftrsasigkey= chave pública de mbox2
            rightrsasigkey=chave pública de mbox3
            auto=start
```

__/etc/mbox/ipsec.wrk__

Deve-se ter muita atenção com a convenção adotada ao atribuir as variáveis em cada arquivo de configuração ipsec.wrk e ipsec2.wrk nas máquinas MBOX. Tem-se que analisar quem é left e quem é right em cada conexão. Nesse exemplo, Zeus é right em ambas as conexões.

ipsec2.wrk

Da mesma forma aqui devemos observar quem é left e right numa dada conexão.

__/etc/mbox/ipsec2.wrk__

```
      internet gegenstation: RSA {
            Modulus:    PublicExponent: 0x03
            # everything after this point is secret
            PrivateExponent:
            Prime1:
            Prime2:
```

```
        Exponent1:
        Exponent2:
        Coefficient:
        }
    internet otherside: RSA {
        Modulus:
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent:
        Prime1:
        Prime2:
        Exponent1:
        Exponent2:
        Coefficient:
        }

    # do not change the indenting of that "}"
```

__/etc/mbox/ipsec2.wrk__

dnsupdate

O script dnsupdate foi criado com o objetivo de realizar a alteração do registro do próprio MBOX no servidor DNS. O script, que tem a finalidade de executar o nsupdate usando as chaves do tipo TSIG.

__/etc/mbox/dnsupdate__

```
    #!/bin/sh

    NEW_IP_ADDRESS=`ifconfig ppp0|grep "inet addr"|cut -d':' -f2|cut -d'
' -f1`
    #echo $NEW_IP_ADDRESS

    nsupdate -v -k $KEYFILE > /dev/null << EOF
    server $SERVER
    zone $ZONE
    update delete $HOSTNAME A
    update add $HOSTNAME $TTL A $NEW_IP_ADDRESS
    show
```

```
send
EOF
```

__/etc/mbox/dnsupdate__

É importante dizer que para esses scripts executarem corretamente nas respectivas máquinas clientes, os mesmos devem estar com seus relógios sincronizados com mbox1. Para essa sincronização foi configurado em todos os MBOX o ntpd.

4.3.2 Configuração do Servidor

A configuração do IPsec para o estabelecimento da VPN é realizado através do programa initipsec. Ele também é responsável pela atualização do registro tipo A na zona DNS.

Para o funcionamento do sistema foi montado um servidor DNS que utilizasse as tecnologias de DDNS e DNSSEC. Este servidor é responsável por receber as requisições de atualização de registro de cada um dos *gateways* VPN que são realizados através do *script* initipsec em cada cliente. O servidor também é responsável por verificar a integridade e autenticidade da requisição.

Neste projeto foi registrada uma única zona DNS chamada de mbox.com.br em REGISTRO.br de uso exclusivo pelo projeto. O REGISTRO.br é a entidade que registra os domínios de Internet no Brasil. Este servidor também é o servidor master desta zona DNS.

Estamos utilizando como servidor de DNS o BIND 9.2.1-9, padrão da instalação do Red Hat 8.0. A partir do BIND 9.2, todas as versões já suportam o DNSSEC e o DYNDNS.

4.3.3 Segurança para Update de DNS Dinâmico (DYNDNS e DNSSEC)

Para implementarmos a segurança na atualização de registros no servidor DNS foi utilizado o sistema de Chaves Simétricas. Essa escolha se deve ao fato do BIND nesta versão suportar apenas esse tipo de Chave para a realização do Update Seguro. O sistema de chave adotada foi o TSIG com o algorítimo de chaves HMAC-MD5 (*Hash-based Message Authentication Code-Message-Digest Algorithm 5*).

Como dito anteriormente, a configuração do ambiente de DYNDNS e DNSSEC do projeto MBOX é bem simples e eficiente. A configuração do servidor será realizada a partir

do arquivo */chroot/named/etc/named.conf* além do arquivo de zona do domínio mbox.com.br. Não é necessário a configuração do zona reversa pois esta está sob responsabilidade do provedor de acesso Internet dos clientes.

Para gerarmos as chaves TSIG utilizadas para atualização de DYNDNS pelos MBOX, basta digitar o comando:

```
#dnssec-keygen a HMAC-MD5 b 512 HOST <keyname>
```

<keyname> é o nome da chave gerada. No presente caso, é utilizado *mbox.máxima-ti.com.br* como o nome da chave.

A chave gerada será armazenada no arquivo:

`K<keyname>+157+<keyid>.private`

Este arquivo é utilizado no script de atualização */etc/dnsupdate*.

<keyid> é uma identificação, única, da chave.

4.3.4 Configuração do servidor DNS

A configuração do servidor depende de qual tipo de chave é utilizada.

O arquivo */chroot/etc/named/named.conf* dever ser editado para configurar o servidor para receber update de DNS dinâmico. Assim, o primeiro passo para isso é configurar o servidor para usar chaves de autenticação dos dados de update cifrados, neste caso chaves do tipo TSIG. Isso é feito adicionando-se no arquivo named.conf o trecho abaixo:

```
key <keyname> {
      algorithm HMAC-MD5;
      secret <keydata>;
};
```

Onde <keyname> o nome da chave escolhida no momento de criação da mesma e <keydata> a chave em si que é o conjunto de caracteres que segue o nome key dentro do arquivo de chave `K<keyname>+157+<keyid>.private` que foi gerado anteriormente.

Por fim, basta dar as permissões para quais máquinas poderão realizar update de DNS das zonas utilizando chave. Ou seja, além da máquina ter que possuir a chave para fazer o

update de DNS, a mesma ainda deve ser autorizada pelo servidor para fazer isso. Para garantir uma maior segurança, cada MBOX vai possuir uma chave especifica e poderá atualizar apenas seu próprio registro.

Para tanto basta adicionar no arquivo named.conf um trecho similar ao que segue abaixo:

```
zone “example.com” {
      type master;
      file “master/example.com”;

      update-policy {
            grant foo.example.com name foo.example.com. A;
      };
};
```

São mostrados a seguir os arquivos de chaves TSIG e o arquivo /chroot/named/etc/named.conf da máquina servidora de DNS do projeto MBOX, chamada de mbox1.mbox.com.br.

Vale lembrar que os arquivos de configuração do BIND de mbox1.mbox.com.br se encontram em /chroot/named/etc/namedb. Neste mesmo diretório se encontra um arquivo, que no caso do projeto MBOX, chamado mbox.com.br.zone.jnl que é criado e utilizado pelo próprio BIND para atualizar o arquivo mbox.com.br.zone uma vez que este não é atualizado todas as vezes que um update de DNS é feito e sim apenas de tempos em tempos de acordo com esse arquivo .jnl.

named.conf

__/chroot/named/etc/named.conf__

```
## named.conf - configuration for bind

options {
      directory "/etc/namedb/";
      pid-file "/var/run/named.pid";
};
```

```
key "zeus.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};
key "mbox000.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};
key "mbox002.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};
key "mbox003.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};
key "mbox004.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};
key "mbox005.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};
key "mbox006.mbox.com.br" {
      algorithm HMAC-MD5;
      secret " blablablabla ";
};

zone  "." {
      type hint;
      file  "named.ca";
};

zone  "0.0.127.in-addr.arpa" {
      type master;
      file  "0.0.127.in-addr.arpa.zone";
};
```

```
#zone  "0.0.10.in-addr.arpa" {
#     type master;
#     file  "0.0.10.in-addr.arpa.zone";
#};

zone  "17.217.200.in-addr.arpa" {
      type master;
      file  "17.217.200.in-addr.arpa.zone";
};

zone  "localhost" {
      type master;
      file  "localhost.zone";
};

zone  "mbox.com.br" {
      type master;
      file  "mbox.com.br.zone";
      update-policy {
          grant mbox000.mbox.com.br name mbox000.mbox.com.br A;
          grant mbox002.mbox.com.br name mbox2.mbox.com.br A;
          grant mbox003.mbox.com.br name mbox3.mbox.com.br A;
          grant mbox004.mbox.com.br name mbox4.mbox.com.br A;
          grant mbox005.mbox.com.br name mbox5.mbox.com.br A;
          grant mbox006.mbox.com.br name mbox6.mbox.com.br A;
          grant zeus.mbox.com.br name zeus.mbox.com.br A;
          };
};
```

__/chroot/named/etc/named.conf__

mbox.com.br.zone

__/chroot/named/etc/namedb/mbox.com.br.zone__

```
$ORIGIN .
$TTL 86400  ; 1 day
mbox.com.br       IN SOA      mbox.com.br. root.mbox.com.br. (
                         365        ; serial
                         28800      ; refresh (8 hours)
                         7200       ; retry (2 hours)
                         604800     ; expire (1 week)
```

```
                        86400      ; minimum (1 day)
                        )
                    NS    ns1.mbox.com.br.
                    NS    ns2.mbox.com.br.
                    MX    0 smtp.mbox.com.br.
$ORIGIN mbox.com.br.
$TTL 120    ; 2 minutes
mbox0               A     200.149.94.27
mbox000             A     200.141.249.70
$TTL 86400  ; 1 day
mbox1               A     200.217.17.28
$TTL 120    ; 2 minutes
mbox002             A     200.97.5.93
mbox003             A     200.141.250.184
mbox004             A     200.149.157.178
mbox005             A     200.165.7.26
mbox006             A     200.149.82.9
$TTL 86400  ; 1 day
ns1                 A     200.217.17.28
ns2                 A     200.179.160.2
smtp                A     127.0.0.1
$TTL 1      ; 1 second
www                 A     200.217.17.28
$TTL 120    ; 2 minutes
zeus                A     200.165.46.212
```

__/chroot/named/etc/namedb/mbox.com.br.zone__

Kmbox.mxima-ti.com.br+157+31388.private

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: blÃ¡blÃ¡blÃ¡...
```

Uma vez que o as configurações do BIND foram realizadas corretamente, basta iniciá-lo com o comando:

```
#/etc/init.d/named start
```

Capítulo 5: Estudo de Caso

O projeto do MBOX foi implementado e funcionou plenamente em ambientes reais da região metropolitana da Grande Vitória-ES-Brasil durante um período de 4 anos, entre 2004 e 2008. Ele foi utilizado por empresas de vários setores da economia, tais como empresas siderurgia, turismo e instituições de ensino. Foram implementados 22 gateways durante este período.

Uma arquitetura simplificada do funcionamento da solução de conectividade segura MBOX é mostrada na Figura 5.1. O ambiente é composto de vários MBOX interligados entre si por VPNs. Cada MBOX é o VPN *gateway* das redes internas das empresas que trocam informações entre si.

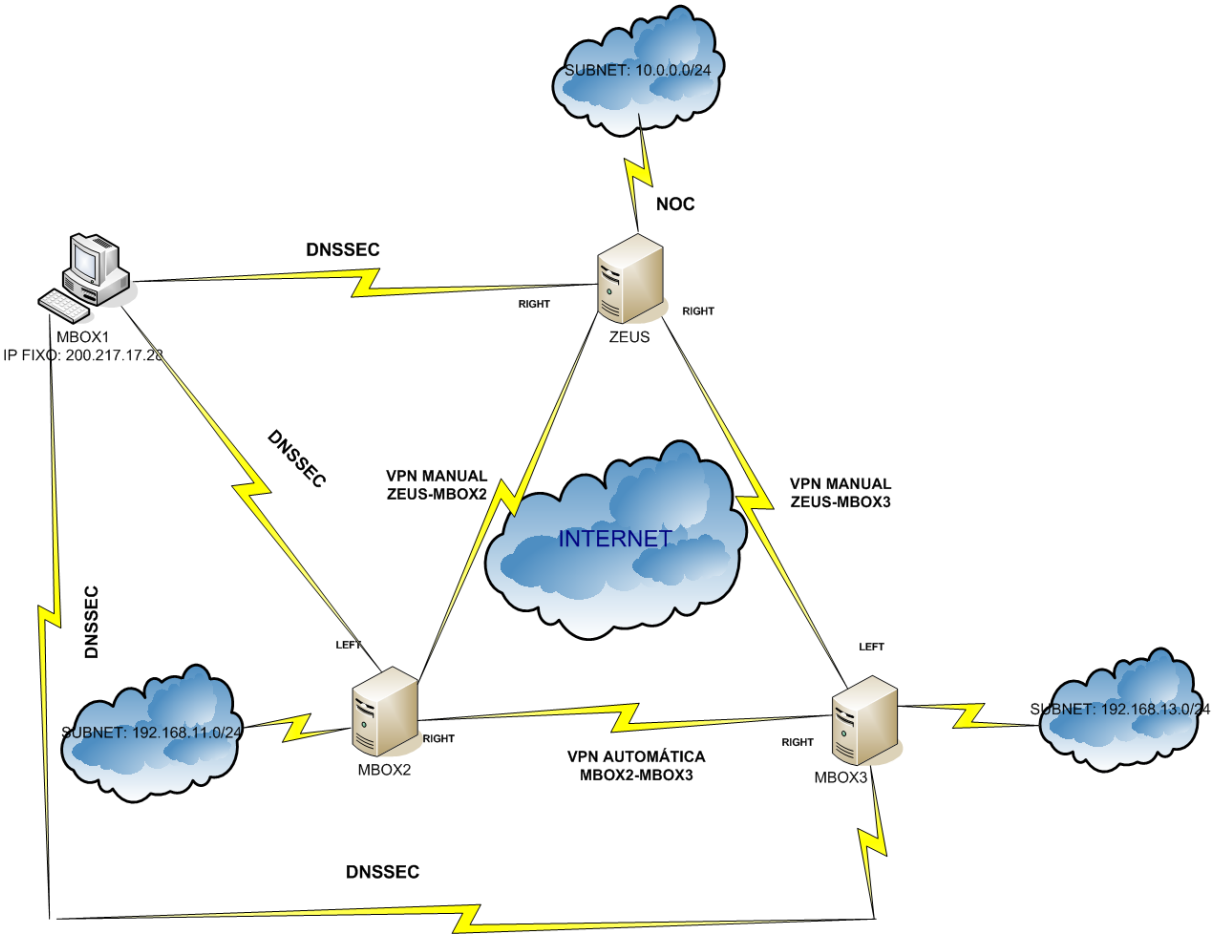

Figura 5.1: Arquitetura MBOX

Na Figura 5.1 podem-se observar os seguintes componentes:

- **MBOX2 e MBOX3**: dispositivos utilizados como VPN *gateways* nas redes em que se deseja prover conectividade. Utilizam modem ADSL para conectarem-se à Internet e não têm endereço IP fixo. Executam aplicativos como *webproxy*, filtro de pacotes, e outros que desempenham funções de controle e segurança. Considerando que MBOX2 e MBOX3 sejam VPN *gateways* de duas filiais, a VPN entre as filiais deve sempre estar sempre estabelecida.
- **MBOX de suporte – ZEUS**: dispositivo utilizado pela empresa que oferece a solução de conectividade para estabelecer conexões com qualquer MBOX através de uma VPN sob demanda (VPN Manual). As VPNs são estabelecidas para permitir o suporte remoto nas redes internas dos clientes. Também utiliza modem ADSL para conectar-se à Internet e não tem endereço IP fixo.
- **MBOX1**: dispositivo essencial para o funcionamento de toda a arquitetura. Possui endereço IP fixo, conexão com a Internet através de um *link* dedicado. É um servidor de DNS dinâmico (DYNDNS) e sua função é mapear o nome dos outros MBOXs com o IP que eles possuem em qualquer momento, pois o provedor da conexão ADSL fornece um endereço IP dinâmico para os MBOXs. Quando os MBOXs detectam que tiveram seus endereços IP ou o endereço de outro VPN *gateway* alterados pelo provedor, e fazem a alteração dos registros no MBOX1 usando DNSSEC. A configuração das VPNs é automaticamente modificada e a VPN é restabelecida.

O hardware que suporta os dispositivos MBOXs é um PC (*Personal Computer*) padrão baseado no sistema operacional GNU Linux Red Hat 8. A seguir são descritas as configurações de software para implementar a solução e na TABELA 5.1 estão as informações das versões utilizadas no primeiro protótipo:

- **Conectividade com endereçamento IP dinâmico**: as VPNs são criadas utilizando-se o IP Security Protocol (IPSec) a partir do software Linux FreeS/WAN. O servidor DNS BIND foi utilizado para suportar o DYNDNS e o DNSSEC. O serviço de conexão ADSL utilizado foi o Velox da Telemar S.A. provido em Vitória-ES-Brasil.
- **Segurança de borda nos MBOXs (VPN *gateways*)**: o software Bastille aumenta a segurança do sistema operacional, o Tripwire mantém a integridade dos arquivos do

sistema, o IPTable e o Squid foram usados para implementar as regras de *firewall* e *webproxy* respectivamente, o PSAD (*Port Scanner Attack Detector)* como um detector de ataques do tipo *port scan* e o Logcheck para análise de *logs* e envio de relatórios via e-mail utilizando para isso o Sendmail.

- **Configuração remota**: os MBOXs podem ser configurados a partir de um centro remoto de operações de rede (NOC – *Network Operation Center*), através de interfaces Web implementadas a partir de tecnologias Java (Servlets e JSP).

Componente	Função
Linux Red Hat 8.0	Sistema Operacional
FreeS/WAN 2.0	VPN baseada em IPsec
Bind 9.2	Servidor DDNS e DNSSEC
Bastile	Configuração de segurança para o sistema operacional
Tripware	Integridade do sistema de arquivo
Squid	WebProxy
PSAD	Detector de Port Scan
Logcheck	Analisador de logs
Sendmail	Servidor de correio eletrônico
Servlets e JSP	Interface gráfica

Tabela 5.1: Componentes da Solução

Todos os MBOXs foram monitorados no esquema 24x7 a partir do NOC da empresa Unitera que atuou no desenvolvimento e ofereceu a solução como um serviço aos seus clientes. Durante os primeiros seis meses de operação foram medidas 25 quedas por mês em média. Todas as vezes o sistema realizou a atualização e reconfiguração automática das VPNs sem a necessidade de intervenção humana. Testes realizados mostram um período de indisponibilidade entre 2 e 4 minutos.

O Nagios é um *software* livre de monitoramento e que foi utilizado para medir a disponibilidade dos MBOXs. A Figura 5.2 mostra o monitoramento do MBOX2 realizado em um período de 8 dias. Pode-se observar uma disponibilidade de aproximadamente 98%. Foram medidas 12 trocas de endereço, sendo que em duas dessas trocas houve um período de indisponibilidade do acesso a Internet causado pela operadora que totalizou 3h e 22minutos.

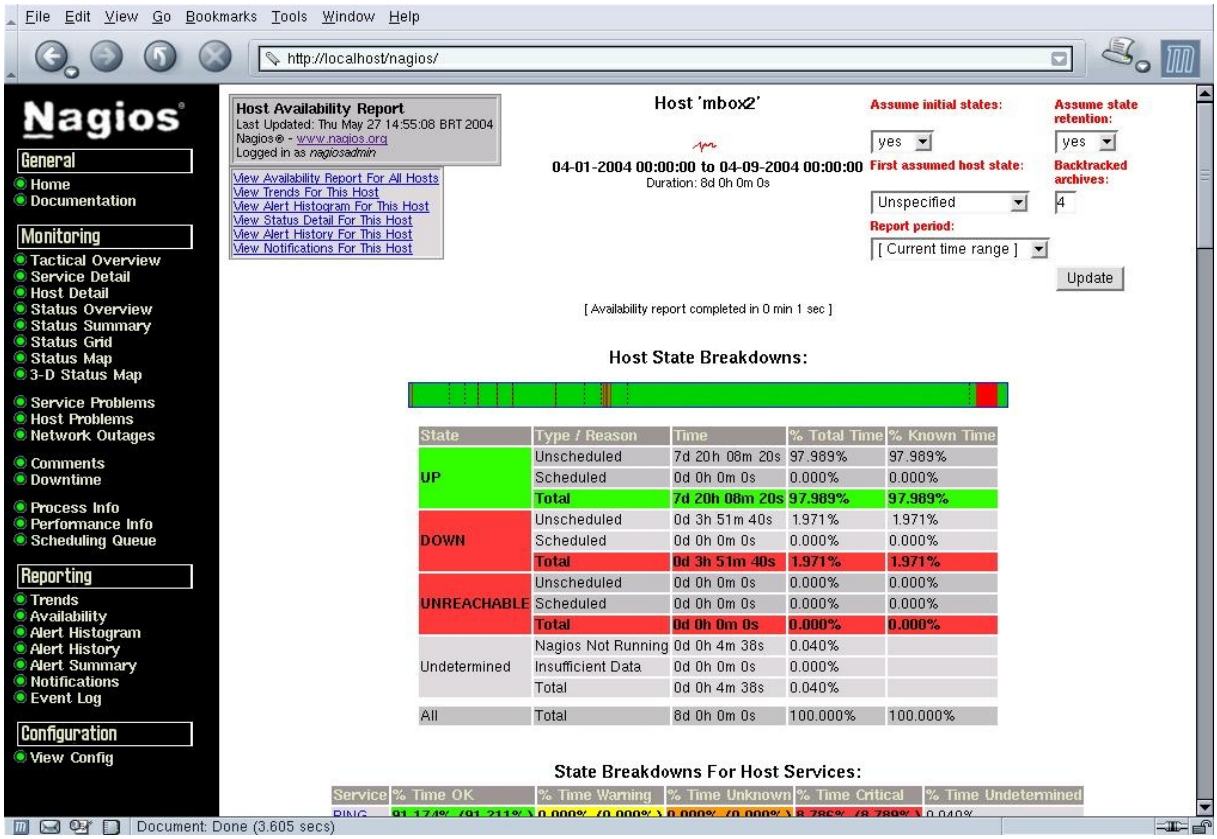

Figura 5.2: Nagios

Para tornar o Mbox um produto de fácil instalação por parte do usuário. Foi desenvolvida uma interface gráfica acessível via Web *browser* [57]. Essa interface permite ao usuário a instalação e configuração de todas as funcionalidades oferecidas. A Figura 5.3 apresenta um exemplo da Interface gráfica.

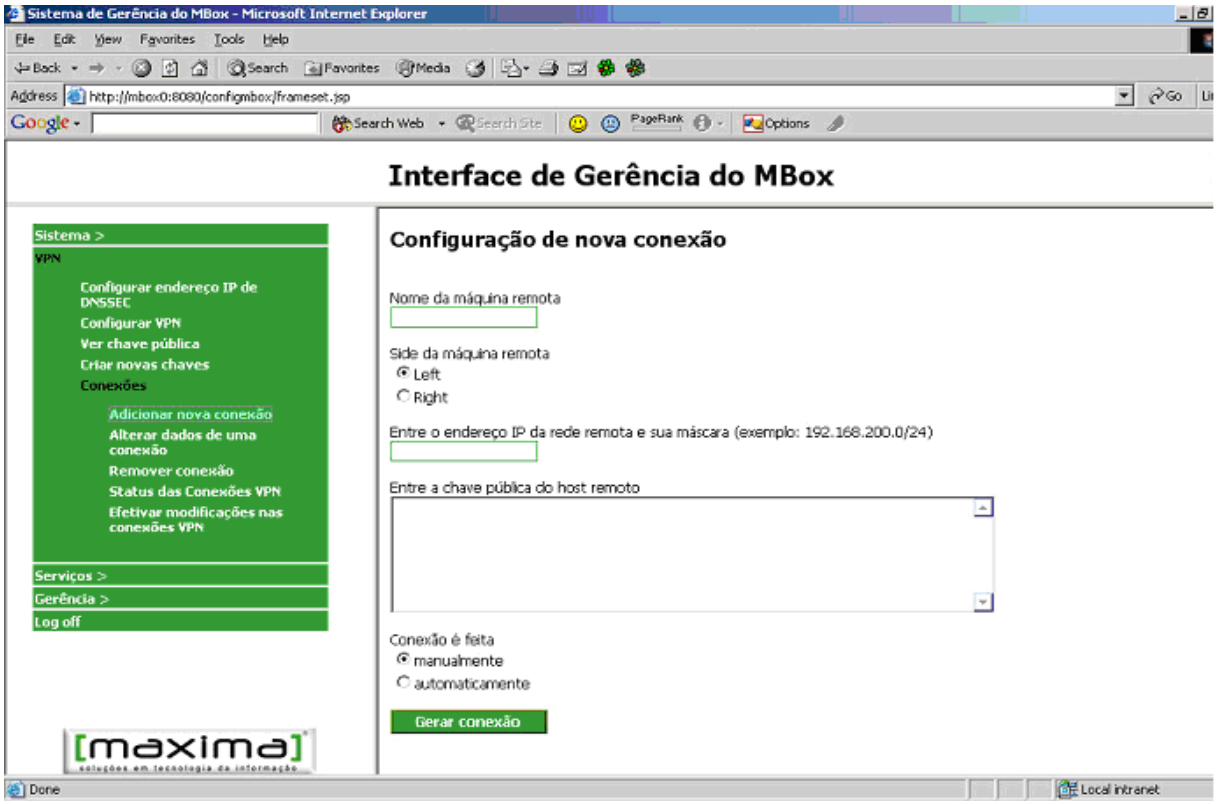

Figura 5.3: Interface Gráfica

O projeto do MBOX também foi utilizado em um modelo integrado de um sistema de gerenciamento como visto em [58]. Este sistema é composto por um servidor de gerenciamento de redes central, localizado no NOC que armazena os dados históricos para geração de estatísticas dos elementos gerenciados. Esses dados são coletados através do protocolo SNMP (*Simple Network Management Protocol*) e armazenados localmente em cada MBOX. Cada MBOX remoto processa as informações localmente tentando antecipar problemas de desempenho e falhas e os envia através de *traps* ao servidor central.

Capítulo 6: Conclusão e Trabalhos Futuros

Neste trabalho foi proposto e desenvolvido um sistema baseado em endereçamento IP dinâmico com reconfiguração automática das VPNs, ficando transparente para o usuário final a mudança de endereços por parte do provedor.

Em outros trabalhos relacionados, são observados problemas como a necessidade de intervenção manual para reconfiguração ou a automação do processo de restabelecimento das conexões VPN, usando métodos de atualização que não garantem a integridade e autenticidade das informações trafegadas. Ambos os casos podem causar a indisponibilidade da conexão. Na primeira situação, a indisponibilidade é causada pela dificuldade de monitoramento e reconfiguração da VPN. Já no segundo caso, pode ser causada por uma alteração indevida nos registros e atualizações.

Algumas soluções utilizam um sistema proprietário de gerenciamento centralizado. Neste caso, há a impossibilidade de integração com outros *gateways* que não usam o mesmo sistema de armazenamento das informações de atualização.

Existem trabalhos que utilizam um servidor DNSSEC para garantir a segurança da atualização das informações e interoperabilidade entre *gateways*. Apesar de serem utilizados em cenários diferentes do proposto neste trabalho, eles serviram de base para o desenvolvimento desta dissertação.

Para solucionar os problemas identificados nos trabalhos anteriores, foi desenvolvido um sistema de monitoramento e reconfiguração automática de VPN utilizando um servidor DNS dinâmico (DDNS) e seguro (DNSSEC) como elemento centralizador das informações dinâmicas.

Tal solução é capaz de manter um elevado nível de segurança em todos os elementos de sua arquitetura, eliminando vulnerabilidades existentes nas soluções adaptadas a este tipo de endereçamento. Além disso, é capaz de interagir com outros equipamentos e sistemas.

O sistema foi testado em um ambiente real apresentando bom desempenho e atingindo um alto índice de disponibilidade.

Como trabalho futuro, este sistema deveria ser adaptado e testado em um cenário que utilize outras formas de acesso, tal como 3G, que estão cada vez mais sendo oferecidas por empresas de telefonia móvel.

Outro trabalho interessante seria uma avaliação de desempenho do sistema ao se modificar o TTL dos registros dinâmicos atualizados no servidor DNS.

Capítulo 7: Referências Bibliográficas

[1]Perez M., - *Barômetro,Cisco de Banda Larga Brasil 2005-2010* - Relatório Técnico, setembro de 2007. Disponível em: http://www.cisco.com/web/BR/barometro/barometro.html. Acesso em: 25 de Março de 2008

[2]CETIC.br, - *TIC EMPRESAS - Pesquisa sobre o Uso das Tecnologias da Informação e da Comunicação no Brasil* – Disponível em: <http://www.cetic.br/pesquisas-indicadores.htm>. Acesso em: 25 de Maço de 2008

[3]Gleeson B., Lin A., Armitage G., Malis A. - *A Framework for IP Based Virtual Private Networks* – Request for Comments 2764, Fevereiro de 2000

[4]Castro R., Geus P. - *Uma Análise de soluções VPN em Redes Corporativas de Alta Capilaridade* – Dissertação de Mestrado – Instituto de Computação, Unicamp, Campinas, Outubro de 2004

[5]ISO - *International Standard 7498 – Information Processing Systems - OSI: Basic Reference Model Addendum 1: Connectionless-mode Transmission*, Maio de 1986.

[6]Laureano M., - *Segurança da Informação* – Monografia de Pós Graduação- PUC, Paraná, Outubro 2005

[7]NBR ISO/IEC 17799 - Associação Brasileira de Normas Técnicas – Tecnologia da Informação - *Código de Prática para Gestão da Segurança da Informação*. Rio de Janeiro, 2003.

[8]Krause, Micki e Tipton, Harold F. - *Handbook of Information Security Management* - Auerbach Publications, 1999.

[9]Albuquerque, Ricardo e Ribeiro, B.. - *Segurança no Desenvolvimento de Software – Como desenvolver sistemas seguros e avaliar a segurança de aplicações desenvolvidas com base na ISO 15.408* - Editora Campus. Rio de Janeiro, 2002.

[10]DIAS, C. - *Segurança e Auditoria da Tecnologia da Informação* - Axcel Books. Rio de Janeiro, 2000.

[11]WADLOW, T. *Segurança de Redes*. Editora Campus. Rio de Janeiro, 2000.

[12]Shirey, R. – *Internet Security Glossary*. - Request For Comments 2828, Maio 2000.

[13]SÊMOLA, M. - *Gestão da Segurança da Informação – Uma Visão Executiva*. Editora Campus. Rio de Janeiro, 2003.

[14]SANDHU, Ravi S., SAMARATI P. - *Authentication, Acess Control, and Intrusion Detection* - IEEE Communications, 1994.

[15]Nakamura E.; Geus P. - *Segurança de Redes em Ambientes Cooperativos* - Editora Berkeley, São Paulo, Brasil, 2002.

[16]Richardson M., Redelmeier D. H. - *Opportunistic Encryption using the Internet Key Exchange (IKE)* – Rquest for Comments 4322, Dezembro de 2005

[17]Regan K., - *Secure VPN Design Considerations* – Network Security, Maio 2003.

[18]VPN Consortium - *VPN Technologies: Definitions and Requirements* - Disponível em: http://www.vpnc.org/vpn-technologies.html. Acesso em: 20/01/2008.

[19]Frame Relay Forum - *The basic Guide to Frame Relay* - Networking. 1998.

[20]Harrison J. - *VPN Technologies - A Comparison*. - Data Connection Limited, 2003. Disponível em: http://www.dataconnection.com. Acesso em: 27/02/2008.

[21]Alterson G. - *Comparing BGP/MPLS and IPSec VPNs*. - SANS Institute - GIAC Security Essentials (GSEC), 2002.

[22]Kent S., Seo K. - *Security Architecture for the Internet Protocol* – Request for Comments 4301, Dezembro e 2005

[23]Cisco Systems - *IPSec* - Cisco Systems White Paper, 2000.

[24]Kent S. - *IP Authentication Header* – Request for Comments 4302, Dezembro 2005

[25]Kent S. - *IP Encapsulating Security Payload (ESP)* – Request for Comments 4303, Dezembro 2005

[26]Kent S. - *Internet Key Exchange (IKEv2) Protocol* – Request for Comments 4306, Dezembro 2005

[27]Boava A., Magalhães M. - *Estrategia de projeto de VPN MPLS com QoS* – Dissertação de Mestrado, Instituto de Computação, Unicamp, Junho de 2004

[28]Cisco Systems. - *Deploying Complex and Large Scale IPSec VPNs* - Networkers 2003 - Session SEC-2012, 2003.

[29]Strahler O. - *Network Based VPNs* - SANS Institute - GIAC Security Essentials (GSEC), 2003.

[30]Metzler J. - *Crafting SLAs for Private IP Services* - Webtorials - IT Bussiness Brief, Fevereiro 2004.

[31]Chapman D. B.; Zwicky E. D., Cooper S. - *Building Internet Firewalls* - Editora O'Reilly, Segunda Edição, 2000.

[32]Avolio F. M. - *Application Gateways and Stateful Inspection: Brief Note Comparing and Contrasting* - Trusted Information System, Inc. Janeiro de 1998.

[33]Chapman D. B. - *Network (In) Security through IP packet filtering*. - Proceedings of the Third USENIX Unix Security Symposium, pp. 63-67, Setembro de 1992

[34]CheckPoint Software Technologies Ltd. - *Stateful Inspection Firewall Technology*, Relatório Técnico, 1998.

[35]KING, C. M. - *The 8 Hurdles to VPN Deployment* - Information Security, Março de 1999.

[36]Mockapetris P. - *Domain Names – Concepts and Facilities* – Request for Comments 1034, Novembro de 1987

[37]Vixie P., et all - *Dynamic Updates in the Domain Name System (DNS UPDATE)* – Request for Comments 2136, Abril de 1997

[38]Atkins D., Austein R. - *Threat Analysis of the Domain Name System (DNS)* – Request for Comments 3833, Agosto de 2004

[39]Wellington B. - *Secure Domain Name System (DNS) Dynamic Update* – Request for Comments 3007, Novembro de 2000

[40]Vixie P., et all - *Secret Key Transaction Authentication for DNS (TSIG)* – Request for Comments 2845, Maio de 2000

[41]Eastlake 3rd, D. - *Domain Name System Security Extensions* – Request for Comments 2535, Março de 1999

[42]Eastlake 3rd, D. - *DNS Request and Transaction Signatures (SIG(0)s)* - Request for Comments 2931, Setembro de 2000

[43]Nakamura E., Geus P**.** - *Análise de Segurança do Acesso Remoto VPN* – SSI 2000, 24 a 26 de outubro de 2000, Sao Jose dos Campos, SP, Brasil

[44]Benvenuto M., Keromytis A. - *EasyVPN: IPsec Remote Access Made Easy* – LISA ´03, San Diego, CA, EUA, October 16-31, 2003

[45]Resende E., Geus P. – *Uma solução segura e escalável para Acesso Remoto VPN* - SSI 2002, 12 a 16 de outubro de 2002, Sao Jose dos Campos, SP, Brasil

[46]Figueiredo F., Geus P. – *Colocação do VPN na Configuração do Firewall* - SSI 2001, 24 a 26 de outubro de 2001, São José dos Campos, SP, Brasil

[47]SonicWALL - *VPN Site to site with dynamic IP-Addresses* – Relatório Técnico, Agosto de 2003

[48]SofaWare - *Dynamic VPN Service* – Relatório Técnico, Junho de 2004

[49]Sames D. L., Tally G. W. - *An Integrated Approach to Security Management and Response* - Advanced Security Research Journal Volume III Num. I, Março de 2001

[50]Rodeh O., Birman K., Hayden M., Dolev D. - Dynamic Virtual Private Networks - Cornell University Technical Report, TR98-1695, August 1998.

[51]Schuba, C. - *Addressing Weaknesses in the Domain Name System Protocol* - Master's thesis, Purdue University Department of Computer Sciences, August 1993

[52]Holmblad J. - *The Evolving Threats to the Availability and Security of the Domain Name System* SANS Institute, October 5, 2003

[53]Wouters P. – *Secure DNS, Het beveiligingen van DNS in de praktijk* - c't magazine, Março de 2003, Amsterdam, Holanda

[54]Josefsson S. - *Network Application Security Using The Domain Name System* - Stockholm University, Junho de 2000

[55]Hinshelwood D. - *DNS, DNSSEC and the Future* - SANS Institute, May 2003

[56]Rolando C. F. - *Alternativa de Infraestructura de Seguridad Basada en IPsec y DNSsec* – FLIP06, Junho de 2006, Lima, Perú

[57]Carmo A. P. - *MBOX – Manual do Usuário* – Vitória, junho de 2005

[58]Villaca, R. S.; Ramos, A.S.; Drago, R. B.; Garcia, A. S. - *A Web-Based Pro-Active Fault and Performance network Management Architecture* - ICT2004, Fortaleza, Junho 2004