Um Modelo de Segurança de Redes para Ambientes Cooperativos

Emilio Tissato Nakamura

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Um Modelo de Segurança de Redes para Ambientes Cooperativos

Emilio Tissato Nakamura[1]

Setembro de 2000

Banca Examinadora:

- Prof. Dr. Paulo Lício de Geus (Orientador)
- Prof. Dr. Adriano Mauro Cansian
  IBILCE, UNESP
- Prof. Dr. Ricardo Dahab
  Instituto de Computação, UNICAMP
- Prof. Dr. Célio Cardoso Guimarães (Suplente)
  Instituto de Computação, UNICAMP

1. financiado por Robert Bosch Ltda

Ficha Catalográfica elaborada pela Biblioteca Central da UNICAMP

Nakamura, Emilio Tissato
N145m Um Modelo de segurança de redes para ambientes cooperativos / Emilio Tissato Nakamura. – Campinas, SP: [s.n.], 2000.

Orientador: Paulo Lício de Geus.
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Redes de computação - Medidas de segurança. 2. Internet (Redes de computadores). 3. Redes de computação - Protocolos. 4. Criptografia. 5. UNIX (Sistema operacional de computador) - Medidas de segurança. 6. TCP/IP (Protocolos de redes de computação. I. Geus, Paulo Lício de. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título

Um Modelo de Segurança de Redes para Ambientes Cooperativos

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Emilio Tissato Nakamura e aprovada pela Banca Examinadora.

Campinas, setembro de 2000

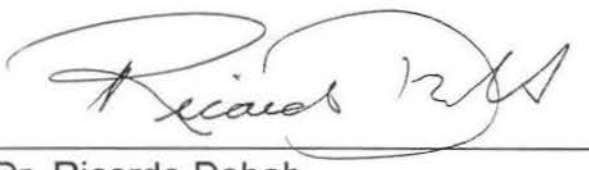
Paulo Lício de Geus (Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 25 de setembro de 2000, pela Banca Examinadora composta pelos Professores Doutores:

Prof. Dr. Adriano Mauro Cansian
UNESP



Prof. Dr. Ricardo Dahab
IC – UNICAMP

Prof. Dr. Paulo Lício de Geus
IC – UNICAMP

Resumo

A principal característica de um ambiente cooperativo é a complexidade das conexões lógicas. Isso traz como conseqüência uma série de implicações de segurança. Diferentes níveis de acesso são agora utilizados para recursos, antes disponíveis apenas internamente, a partir de múltiplas e heterogêneas conexões. Isso faz com que a abordagem clássica dos *firewalls*, de criar uma separação entre a rede da organização e a rede pública, já não baste. Diversos conceitos e tecnologias têm que ser utilizados para que a segurança seja provida em ambientes cooperativos. Mais do que isso, um modelo de segurança também é necessário.

Assim, este trabalho tem como objetivo apresentar tais conceitos e tecnologias de segurança, e propor um modelo para que eles possam proteger de fato um ambiente cooperativo. Política de segurança, *firewalls*, sistemas de detecção de intrusões, redes privadas virtuais, infra-estrutura de chaves públicas e autenticação são os tópicos abordados neste trabalho. O modelo de segurança proposto é formado por três elementos: i) o *firewall* cooperativo, que sugere uma arquitetura de segurança para a integração das diferentes tecnologias e conceitos; ii) um modo de minimizar os problemas resultantes da complexidade das regras de filtragem e iii) um modelo de cinco níveis hierárquicos de defesa, que visa facilitar a compreensão dos problemas de segurança existentes, e assim minimizar as possibilidades de erros na definição da estratégia de defesa da organização.

Abstract

The main characteristic of a cooperative environment is the complexity of logical connections. This brings about a series of security implications. Different access levels are now used for resources, previously available only internally, from multiple and heterogeneous connections. This makes the classical approach to firewalls, that of creating a division between the organization's and the public networks, no longer enough for the task. Diverse concepts and technologies need to be employed to provide security to cooperative environments. More than that, a security model is also needed.

As such, this work has as its main goal to present such security concepts and technologies and to propose a framework so that they may in fact protect a cooperative environment. Security policy, firewalls, intrusion detection systems, virtual private networks, public key infrastructure and authentication are the topics discussed in this work. The security model proposed is made of three elements: i) the cooperative firewall, which suggests a security architecture for the integration of different concepts and technologies; ii) a way of minimizing the problems resulting from the complexity of filtering rules and iii) a 5-level hierarchical defense model, which intends to ease the understanding of existing security problems, so that the process of defining the organization's defense strategy is made less error-prone.

Dedico aos meus pais, as pessoas que mais me ajudaram, que mais me apoiaram e que mais me incentivaram, não só durante a realização deste trabalho, mas também em todos os momentos de minha vida.

Agradecimentos

Ao professor Paulo Lício de Geus, por toda orientação, apoio e atenção fornecidas durante o desenvolvimento deste trabalho.

Ao Eduardo e à Cecília, pelo apoio e ajuda em Campinas.

Aos meus irmãos.

A todos os meus amigos, pelos grandes momentos que passamos juntos.

Aos colegas e funcionários do Instituto de Computação.

À Grace, por todo amor, paciência e compreensão.

Ao pessoal do grupo de redes e telecomunicações (AST41) da Bosch, pelo auxílio e apoio.

À Robert Bosch, pelo apoio financeiro concedido para a realização deste trabalho.

A Deus, pela oportunidade de realização de mais este importante passo em minha vida.

Conteúdo

Resumo v

Abstract vi

Dedicatória vii

Agradecimentos viii

Conteúdo ix

Lista de Tabelas xv

Lista de Figuras xvi

1 Introdução 18

2 O Ambiente Cooperativo 20
- 2.1 A Informática Como Parte dos Negócios . . . 20
- 2.2 Ambientes Cooperativos . . . 21
- 2.3 Problemas nos Ambientes Cooperativos . . . 21
- 2.4 Segurança em Ambientes Cooperativos . . . 23
- 2.5 Conclusão . . . 25

3 A Necessidade de Segurança 26
- 3.1 A Segurança de Redes . . . 26
- 3.2 Mais Evolução, Mais Preocupação com Segurança . . . 27
- 3.3 Segurança como Parte dos Negócios . . . 28
- 3.4 Como a Segurança é Vista Hoje . . . 30
- 3.5 Investimentos em Segurança . . . 32
- 3.6 Mitos sobre Segurança . . . 34
- 3.7 Riscos e Considerações quanto à Segurança . . . 34
- 3.8 Segurança vs Funcionalidades . . . 35
- 3.9 Segurança vs Produtividade . . . 37
- 3.10 Uma Rede Totalmente Segura . . . 37
- 3.11 Conclusão . . . 38

Os Riscos que Rondam as Organizações **40**
4.1 Os Potenciais Atacantes 40
4.1.1 Script Kiddies 41
4.1.2 Cyberpunks 42
4.1.3 Insiders 42
4.1.4 Coders 46
4.1.5 White Hat 46
4.1.6 Full Fledged 47
4.2 Terminologias do Mundo Hacker 47
4.3 Os Pontos Explorados 48
4.4 O Início de um Ataque 50
4.5 Ataques para Obtenção de Informações 52
4.5.1 Engenharia Social 52
4.5.2 Packet Sniffing 53
4.5.3 Scanning de Vulnerabilidades 55
4.5.4 Port Scanning 57
4.5.5 Firewalking 62
4.5.6 IP Spoofing 62
4.6 Ataques de Negação de Serviços 63
4.6.1 Bugs em Serviços, Aplicativos e Sistemas Operacionais 63
4.6.2 SYN Flooding 64
4.6.3 Fragmentação de pacotes IP 65
4.6.4 Smurf e Fraggle 67
4.6.5 Teardrop e Land 68
4.7 Ataque Ativo contra o TCP 68
4.8 Ataques Coordenados 69
4.9 Ataques no Nível de Aplicação 73
4.9.1 Buffer Overflow 73
4.9.2 Ataques Web 74
4.9.3 Problemas com o SNMP 75
4.9.4 Vírus, Worms e Cavalos de Tróia 77
4.9.5 War Dialing 80
4.10 Conclusão 80

Política de Segurança **82**
5.1 A Importância 82
5.2 O Planejamento 83
5.3 Os Elementos 84
5.4 Considerações Sobre a Segurança 85
5.5 Os Pontos a Serem Tratados 87
5.6 A Implementação 89
5.7 Os Maiores Obstáculos para a Implementação 89
5.8 Política para Senhas 93
5.9 Política Para Firewall 96
5.10 Política Para Acessos Remotos 96
5.11 Política de Segurança em Ambientes Cooperativos 97

5.12 Conclusão 102

6 Firewalls **104**
6.1 Definição e Função 104
6.2 Funcionalidades 106
6.2.1 Filtros 106
6.2.2 Proxies 106
6.2.3 Bastion Hosts 107
6.2.4 Zona Desmilitarizada 107
6.2.5 Network Address Translation (NAT) 107
6.2.6 Virtual Private Network (VPN) 108
6.2.7 Autenticação/Certificação 108
6.3 A Evolução Técnica 108
6.3.1 Filtro de pacotes 110
6.3.2 Filtros de Estados 111
6.3.3 Proxy 115
6.3.3.1 Proxy Transparente 116
6.3.4 Firewalls Híbridos 117
6.3.5 Proxies Adaptativos 118
6.3.6 Firewalls Reativos 119
6.3.7 Firewalls Individuais 119
6.3.8 A Melhor Tecnologia de Firewall 121
6.4 As Arquiteturas 122
6.4.1 Dual-Homed Host Architecture 123
6.4.2 Screened Host Architecture 123
6.4.3 Screened Subnet Architecture 124
6.4.4 Firewall Cooperativo 126
6.5 O Desempenho 126
6.6 O Mercado 128
6.7 A Avaliação do Firewall 130
6.8 Teste do Firewall 131
6.9 Problemas Relacionados 132
6.10 O Firewall Não é a Solução Total de Segurança 134
6.11 Conclusão 136

7 Sistema de Detecção de Intrusões **138**
7.1 Objetivos 138
7.2 Características 139
7.3 Tipos 140
7.3.1 Host-Based Intrusion Detection System 140
7.3.2 Network-Based Intrusion Detection System 142
7.3.3 Real-Time Activity Monitoring 142
7.3.4 Deception Systems 143
7.4 Metodologias de Detecção 143
7.4.1 Knowledge-Based Intrusion Detection 143
7.4.2 Behavior-Based Intrusion Detection 145
7.4.3 Computer Misuse Detection System 145

7.5 Padrões 146
7.5.1 Intrusion Detection Exchange Format 146
7.5.2 Common Intrusion Detection Framework 147
7.6 Localização do IDS na Rede 147
7.7 Problemas do IDS 149
7.8 Conclusão 150

A Criptografia e a PKI **152**
8.1 O Papel da Criptografia 152
8.2 A Segurança dos Algoritmos Criptográficos 153
8.2.1 A Segurança pelo Tamanho das Chaves 155
8.3 As Maiores Falhas nos Sistemas Criptográficos 158
8.4 Os Ataques aos Sistemas Criptográficos 159
8.5 Certificados Digitais 161
8.6 Public Key Infrastructure 162
8.6.1 Funções da PKI 163
8.6.2 Componentes da PKI 165
8.6.2.1 A Autoridade Certificadora 166
8.6.3 Desafios da PKI 167
8.6.4 Padrões PKI 168
8.7 Conclusão 169

Virtual Private Network **170**
9.1 Objetivos e Configurações 170
9.2 Implicações 174
9.3 Os Fundamentos da VPN 175
9.4 O Tunelamento 177
9.5 Os Protocolos de Tunelamento 177
9.5.1 PPTP e L2TP 178
9.5.2 IPSec 179
9.5.2.1 Os Dois Modos do IPSec 181
9.5.2.2 A Negociação dos Parâmetros do IPSec 183
9.5.2.3 O Gerenciamento das Chaves 184
9.6 Gerenciamento e Controle de Tráfego 187
9.7 Obstáculos 188
9.8 Conclusão 190

0 Autenticação **192**
10.1 A Identificação e a Autorização 192
10.1.1 Autenticação Baseada Naquilo que o Usuário Sabe 193
10.1.1.1 Senhas 194
10.1.2 Autenticação Baseada Naquilo que o Usuário Possui 195
10.1.2.1 Dispositivos de Memória 195
10.1.2.2 Dispositivos Inteligentes 196
10.1.2.3 Autenticação Baseada Naquilo que o Usuário é 197
10.2 Controle de Acesso 198
10.3 Single Sign-On (SSO) 200

10.4 Conclusão ... 203

11 As Configurações de um Ambiente Cooperativo **204**
11.1 Os Cenários até o Ambiente Cooperativo ... 204
11.2 Configuração VPN/Firewall ... 226
11.2.1 Em Frente ao Firewall ... 227
11.2.2 Atrás do Firewall ... 228
11.2.3 No Firewall ... 229
11.2.4 Paralelo ao Firewall ... 230
11.2.5 Na Interface Dedicada do Firewall ... 231
11.3 Conclusão ... 232

12 O Modelo Proposto **234**
12.1 Os Aspectos Envolvidos no Ambiente Cooperativo ... 234
12.1.1 Usuários Internos X Usuários Externos X Usuários Remotos ... 234
12.1.2 O Desafio no Ambiente Cooperativo ... 235
12.1.3 A Complexidade das Conexões ... 236
12.2 As Regras de Filtragem ... 237
12.2.1 Exemplos de Filtragem – Laboratório de Administração e Segurança (LAS) 238
12.3 Manipulação da Complexidade das Regras de Filtragem ... 250
12.3.1 IPTables ... 250
12.3.2 Netfilter ... 252
12.3.3 O IPTables no Ambiente Cooperativo ... 254
12.4 Integrando Tecnologias – Firewall Cooperativo ... 255
12.5 Níveis Hierárquicos de Defesa ... 258
12.5.1 1° Nível Hierárquico de Defesa ... 261
12.5.2 2° Nível Hierárquico de Defesa ... 262
12.5.3 3° Nível Hierárquico de Defesa ... 263
12.5.4 4° Nível Hierárquico de Defesa ... 263
12.5.5 5° Nível Hierárquico de Defesa ... 264
12.5.6 Os Níveis Hierárquicos de Defesa na Proteção dos Recursos ... 264
12.6 Conclusão ... 266

13 Conclusão **268**

Bibliografia **272**

Lista de Tabelas

8.1 Resistências comparativas entre os algoritmos de chave simétrica e assimétrica. . . 155
8.2 O espaço de chaves e o tempo de processamento necessário. 156
8.3 Estimativas para ataques de força bruta. 157
8.4 Fatoração de chaves do algoritmo assimétrico. 157

Lista de Figuras

2.1 O ambiente cooperativo - diversidade de conexões. 22
2.2 O perigo das triangulações. 23
2.3 Os diferentes níveis de acesso somados ao perigo das triangulações. 24
4.1 As partes envolvidas em um ataque coordenado. 70
5.1 A triangulação que dribla a política de segurança de uma organização. 98
5.2 Modelo de segurança convencional, representada pelo firewall. 99
5.3 Modelo de segurança convencional, representada pelo firewall com DMZ. 100
5.4 Modelo "Bolsões de Segurança", representada pelo firewall cooperativo. 101
6.1 Filtro de estados trabalhando na chegada de pacotes SYN. 112
6.2 Filtro de estados trabalhando na chegada dos demais pacotes. 113
6.3 Filtro de estados trabalhando na chegada de pacotes ACK. 114
6.4 Um hacker pode acessar a rede da organização através do cliente VPN. 120
6.5 A arquitetura host dual-homed. 124
6.6 A arquitetura screened host. 125
6.7 A arquitetura screened subnet. 126
6.8 Uma variação da arquitetura screened subnet. 127
7.1 A localização do IDS na rede da organização. 148
8.1 A arquitetura do modelo PKIX. 166
9.1 Gateway-to-gateway VPN, onde o túnel VPN é criado entre 2 redes. 171
9.2 Gateway-to-gateway VPN, onde o usuário utiliza um provedor VPN. 172
9.3 Client-to-gateway VPN, com provedor de acesso e software VPN. 173
9.4 Client-to-gateway VPN, onde os usuários utilizam um software VPN. 174
9.5 Remote-access VPN, através de provedor Internet e software VPN. 175
9.6 Remote-access VPN, através de provedor VPN, de onde o túnel é criado. 176
9.7 Overhead que pode ocorrer no cabeçalho de um pacote L2TP. 178
9.8 O protocolo L2TP sendo utilizado através de um provedor VPN. 179
9.9 O protocolo PPTP sendo utilizado através de um software cliente. 180
9.10 O protocolo L2TP sendo utilizado através de um software cliente. 180
9.11 A cifragem e a autenticação no modo transporte do IPSec. 182
9.12 No modo transporte o IPsec é incorporado na pilha TCP/IP. 182
9.13 A cifragem e a autenticação no modo túnel do IPSec. 183
9.14 No modo túnel o IPsec é implementado no gateway. 184
9.15 As fases até a negociação do SA. 185
9.16 O estabelecimento de uma conexão VPN baseada em IPSec. 187
11.1 A rede interna de uma organização. 205
11.2 A comunicação entre organizações através de conexão dedicada. 206

11.3 A necessidade do firewall nas conexões com a Internet. 20
11.4 A organização provendo serviços para os usuários externos. 20
11.5 As duas barreiras que formam a DMZ do firewall. 20
11.6 O firewall composto por 3 interfaces de rede. 20
11.7 O servidor de banco de dados na DMZ. 21
11.8 O servidor de banco de dados na rede interna da organização. 21
11.9 O utilização de uma segunda DMZ para o servidor de banco de dados. 21
11.10Duas DMZs em um único componente de firewall. 21
11.11A arquitetura da organização com os acessos à Internet e à filial. 21
11.12Os riscos envolvidos em múltiplas conexões. 21
11.13Múltiplas conexões envolvendo a Internet. 21
11.14Mecanismos de segurança não equivalentes entre matriz e filial. 21
11.15Acesso à Internet da filial através de linha dedicada. 21
11.16Acesso à Internet da filial através de VPN. 21
11.17Acesso à Internet em conjunto com VPN. 22
11.18Aumento da complexidade das conexões. 22
11.19Localização do CA na DMZ. 22
11.20Localização do CA na segunda DMZ. 22
11.21A arquitetura de segurança com o IDS. 22
11.22A VPN na frente do firewall. 22
11.23A VPN atrás do firewall. 22
11.24A VPN no firewall. 23
11.25A VPN paralela ao firewall. 23
11.26A VPN na interface dedicada do firewall. 23
12.1 O esquema utilizado pelo LAS. 23
12.2 Os canais utilizados pelos usuários internos. 24
12.3 Os canais utilizados pelos usuários vindos da Internet. 24
12.4 A arquitetura e equipamentos utilizados pelo LAS. 24
12.5 O funcionamento do iptables. 25
12.6 O funcionamento do netfilter. 25
12.7 O iptables no ambiente cooperativo. 25
12.8 O "muro" em um ambiente cooperativo. 25
12.9 A arquitetura do firewall cooperativo. 25
12.10A granularidade dos níveis hierárquicos de defesa. 25
12.11As ações em cada nível hierárquico de defesa. 26
12.12O 1° nível hierárquico de defesa. 26
12.13O 2° nível hierárquico de defesa. 26
12.14O 3° nível hierárquico de defesa. 26
12.15O 4° nível hierárquico de defesa. 26
12.16O 5° nível hierárquico de defesa. 26

Capítulo 1

Introdução

Tem sido visto ultimamente que a necessidade de segurança é um fato que vem transcendendo o limite da produtividade e da funcionalidade. Enquanto isso, a sua dependência para com o sucesso da organização vem acompanhando os passos da globalização e do crescimento acelerado da economia digital.

Alguns eventos que demonstram esse aumento da importância com a segurança podem ser exemplificados, tais como a rápida disseminação dos vírus, que estão cada vez mais sofisticados. Isso pôde ser observado mais recentemente no mês de maio de 2000, com o ataque do vírus "I Love You". Outro notório evento recente foi a exploração em larga escala de ferramentas de ataques coordenados e distribuídos, que afetaram e causaram grandes prejuízos a *sites* como a Amazon Books, o Yahoo, a CNN, a eBay, a UOL e o ZipMail. Somou-se ainda ataques a *sites* de comércio eletrônico, tendo como principal caso o roubo de informações sobre clientes da CDNow, inclusive dos números de cartões de crédito. Casos de "pichações" de *sites* Web também são um fato corriqueiro, demonstrando a rápida popularização dos ataques a sistemas de computadores, que teve como marco os ataques e a captura do *hacker* Kevin Mitnick, que se encontra atualmente livre após o cumprimento de uma pena na prisão.

Porém, os ataques que vêm causando os maiores problemas para as organizações são aqueles que acontecem a partir da sua própria rede, ou seja, os ataques internos. Somado a isto está o fato das conexões entre as redes das organizações alcançarem níveis de integração cada vez maiores. Os ambientes cooperativos, formados a partir de conexões entre organizações e filiais, fornecedores, parceiros comerciais, distribuidores, vendedores ou usuários móveis, resultam na necessidade de um novo tipo de abordagem quanto à segurança. Em oposição à idéia inicial dos

firewalls, que era de proteger a rede da organização isolando-a das redes públicas, os *firewalls cooperativos* têm como objetivo não apenas proteger a rede da organização contra os ataques vindos da rede pública, mas também contra os ataques que podem ser considerados internos, que podem vir a partir de qualquer ponto do ambiente cooperativo.

Este trabalho tem como objetivo caracterizar um ambiente cooperativo, demonstrar os problemas que existem nesse tipo de ambiente, apresentar algumas tecnologias, técnicas e conceitos de segurança disponíveis, e propor um modelo de segurança que realiza a integração entre os conceitos e as tecnologias apresentadas, de modo a proteger do modo mais eficiente possível o ambiente cooperativo contra os ataques a que estão passíveis. Além disso, um modelo que visa minimizar os erros na definição e na implementação das medidas de segurança e das regras de filtragem também será apresentado.

O trabalho é dividido em 3 partes, sendo que a parte I, composta pelos capítulos 2, 3 e 4, faz a ambientação do problema que motivou a pesquisa. A parte II, formada pelos capítulos entre 5 e 10, apresenta as técnicas, conceitos e tecnologias que podem ser utilizadas na luta contra os problemas de segurança vistos na parte I. Já a parte III apresenta o modelo de segurança proposto. Essa parte é formada pelos capítulos 11 e 12. O capítulo 2 faz a apresentação de um ambiente cooperativo, e as necessidades de segurança são demonstradas no capítulo 3. Os riscos que rondam as organizações, representadas pelas técnicas de ataques mais utilizadas, são discutidos no capítulo 4. A política de segurança, os *firewalls*, os sistemas de detecção de intrusões, a criptografia, as redes privadas virtuais e a autenticação dos usuários são discutidas, respectivamente, nos capítulos 5, 6, 7, 8, 9 e 10. Já o capítulo 11 discute as configurações que podem fazer parte de um ambiente cooperativo, enquanto o capítulo 12 discute os aspectos de segurança envolvidos nesse tipo de ambiente e o modelo proposto. O modelo é composto pela arquitetura do *firewall* cooperativo, o modo de minimizar a complexidade das regras de filtragem, e o modelo hierárquico de defesa, destinado a facilitar a compreensão dos problemas de segurança inerentes, resultando assim em menos erros na definição da estratégia de segurança da organização.

Capítulo 2

O Ambiente Cooperativo

Este capítulo inicial mostra a dependência cada vez maior da informática para o sucesso das organizações, o que faz com que um novo ambiente de extrema importância surja, o ambiente cooperativo. Como conseqüência, diversos problemas passam a ocorrer nesse ambiente, principalmente com relação à segurança dos seus recursos.

2.1 A Informática Como Parte dos Negócios

O mundo moderno e globalizado faz com que as organizações busquem o mais alto nível de competitividade, onde novos mercados são disputados vorazmente. O concorrente agora pode estar em qualquer parte do mundo, e para superá-los, é necessário, mais do que nunca, produzir produtos de qualidade, prestar bons serviços e manter um bom relacionamento com os clientes, sejam eles internos ou externos, tudo com a maior eficiência possível.

Uma das principais características na busca da competitividade global é a velocidade, qualidade e eficiência das comunicações. Atualmente é necessário que a infra-estrutura de telecomunicações, que permite a comunicação entre pessoas e recursos, seja bem projetada e bem dimensionada. Isso é vital para a sobrevivência de uma organização.

A própria infra-estrutura de rede e a informática podem ser consideradas como sendo uma das responsáveis pela rápida globalização. Em menor escala, essa infra-estrutura no mínimo contribuiu e possibilitou o avanço da globalização. Se antes a Revolução Industrial pôde ser vista, agora a Revolução Digital faz parte da vida de todos. A informática é hoje considerada parte do processo de negócios de qualquer organização, de modo que é um fato determinante para o seu sucesso, seja no processo de criação de um produto, no atendimento aos clientes ou na venda

de produtos. Imagine uma falha em algum dos componentes da informática, que pode afetar negativamente os negócios da organização.

2.2 Ambientes Cooperativos

No mundo globalizado e de rápidos avanços tecnológicos, as oportunidades de negócios vêm e vão com a mesma rapidez desses avanços. Todos vivenciam uma época de grandes transformações tecnológicas, econômicas e mercadológicas. Grandes fusões estão acontecendo, implicando também na fusão de suas infra-estruturas de telecomunicações, o que pode resultar em sérios problemas relacionados à segurança.

Além das fusões entre as organizações, as parcerias comerciais e as formas de comunicação avançam de tal modo, que a infra-estrutura de rede, de vital importância para os negócios, passa a ser uma peça chave para todos nesse novo modelo de negócios.

Esse contexto atual, de grandes transformações comerciais e mercadológicas, somado à importância cada vez maior do papel da Internet nesse contexto, faz com que um novo ambiente surja, onde múltiplas organizações trocam informações através de uma rede integrada. Informações técnicas, comerciais e financeiras, necessárias para o bom andamento dos negócios, agora trafegam por essa rede, que conecta matriz com filiais, clientes, parceiros comerciais, distribuidores e usuários móveis. A complexidade dessa rede atinge níveis consideráveis, o que implica em cuidados e medidas que devem ser tomados, principalmente com relação à proteção das informações que fazem parte dessa rede. Esse ambiente onde a rápida e eficiente troca de informações entre matrizes, filiais, parceiros comerciais e usuários móveis é um fator determinante de sucesso, é chamado de ambiente cooperativo. A formação de um ambiente cooperativo (figura 2.1), com as evoluções que ocorrem nas conexões das organizações, pode ser visto com detalhes no capítulo 11.

2.3 Problemas nos Ambientes Cooperativos

Uma característica dos ambientes cooperativos é a complexidade que envolve a comunicação entre diferentes tecnologias (cada organização utiliza a sua), diferentes usuários, diferentes culturas, e diferentes políticas internas. A suíte de protocolos TCP/IP e a Internet permitiram o avanço em direção aos ambientes cooperativos, ao tornar possíveis as conexões entre as dife-

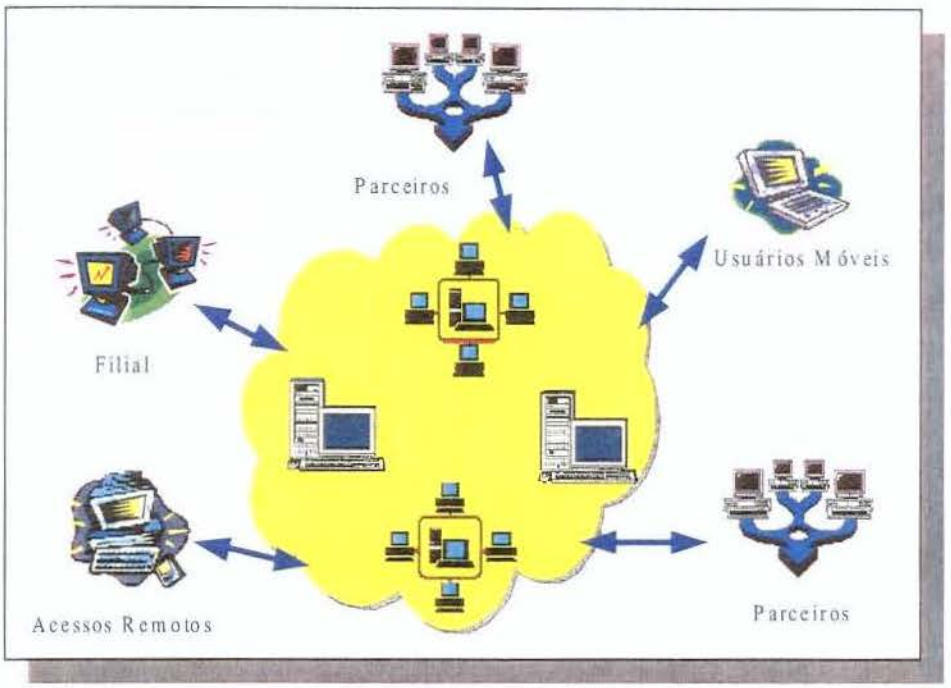


Figura 2.1: O ambiente cooperativo - diversidade de conexões.

rentes organizações de modo simples. Porém, essa interligação teve como conseqüência uma enorme implicação quanto à proteção dos valores de cada organização.

Algumas situações que demonstram o grau de complexidade existente nos ambientes cooperativos podem ser vistos quando é analisado, por exemplo, uma conexão entre 3 organizações (A, B e C). Como proteger os valores da organização A, evitando que um usuário da organização B acesse informações que pertencem somente à organização A? Pode-se supor uma situação onde usuários da organização B não podem acessar informações da organização A, porém usuários da organização C podem acessar as informações da organização A. Como evitar que usuários da organização B acessem informações da organização A através da organização C? Como pode ser visto na figura 2.2, isso constitui um caso típico de triangulação, onde uma rede é utilizada como ponte para uma outra rede.

Além das triangulações, uma outra situação que ocorre em um ambiente cooperativo é quando as entidades da organização A podem acessar todos os recursos da organização, enquanto os usuários da organização cooperada B podem acessar somente determinados recur-

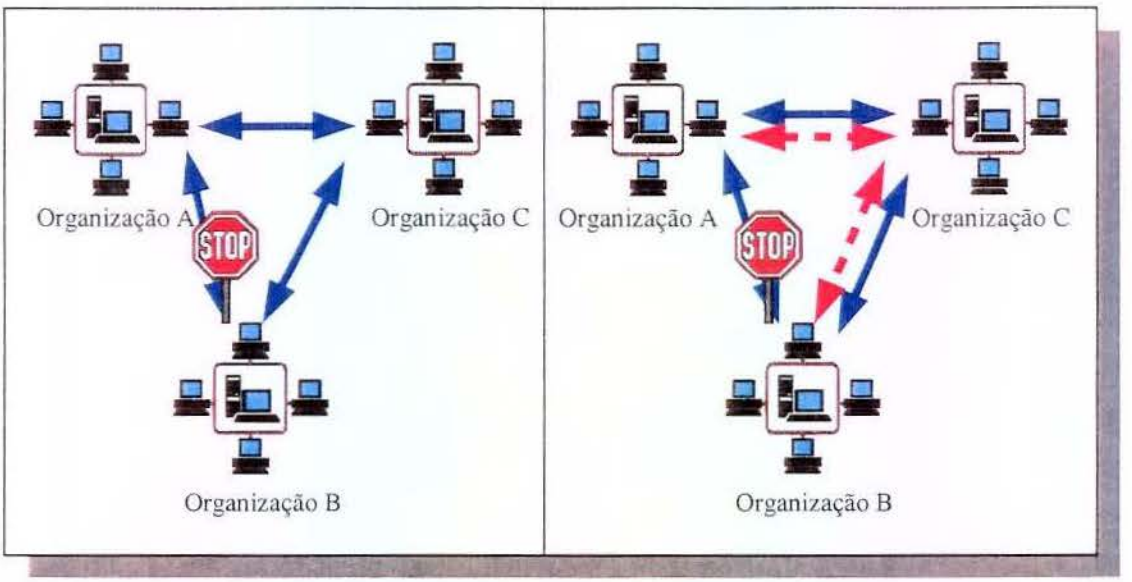


Figura 2.2: O perigo das triangulações.

sos, como por exemplo, informações sobre produtos e do setor financeiro. Além disso, os usuários da Internet não podem acessar nenhum recurso da organização A, enquanto a organização C possui acesso irrestrito aos recursos da organização A. Essa situação demonstra os problemas que surgem em controlar os acessos em diferentes níveis, que pode tornar-se mais complexa ainda se diferentes usuários da organização B acessam diferentes recursos da organização A. Somado a isso, pode-se ver ainda o problema de triangulação, de modo mais forte ainda, com os usuários da Internet podendo chegar à organização A caso a organização B ou C possuam acesso à Internet (figura 2.3).

A divisão entre os diferentes tipos de usuários, os desafios a serem enfrentados no ambiente cooperativo, e a complexidade que envolve esses ambientes são analisados, com detalhes, no capítulo 11.

2.4 Segurança em Ambientes Cooperativos

Os problemas a serem resolvidos nos ambientes cooperativos estão se tornando cada vez mais comuns em organizações hoje em dia. O ambiente cooperativo é complexo, e a segurança neces-

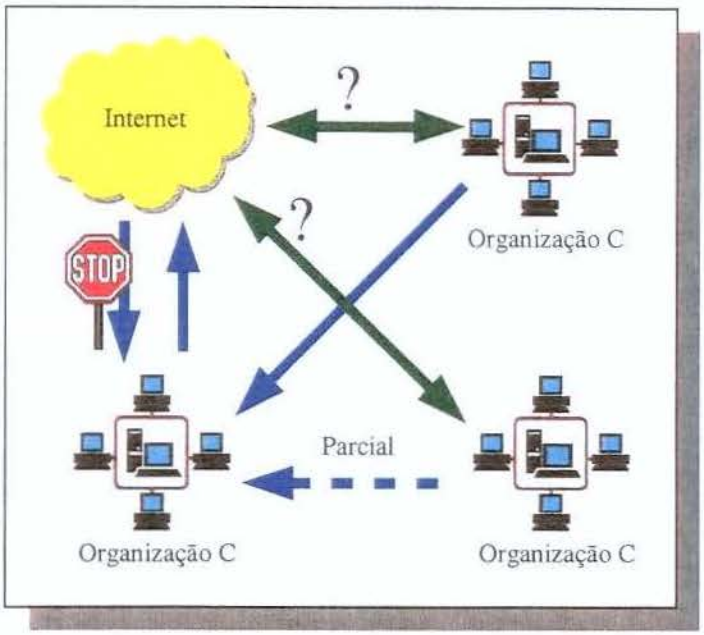


Figura 2.3: Os diferentes níveis de acesso somados ao perigo das triangulações.

sária a ser implementada é igualmente complexa, envolvendo aspectos humanos, tecnológicos e sociais.

Este trabalho irá enfocar com maior ênfase os aspectos tecnológicos relacionados à segurança em ambientes cooperativos, porém não significando que eles tenham maior relevância com relação aos outros. Todos os aspectos são de extrema importância, e devem ser considerados na implantação da segurança nos ambientes cooperativos. O que não pode ser esquecido é que a segurança reside mais nos processos e nas pessoas, e menos na tecnologia [ALE 98].

O trabalho visa identificar os pontos dessa infra-estrutura de rede a serem protegidos, apontar quais os perigos existentes, discutir tecnologias relacionadas à segurança e propor um modelo de segurança que englobe técnicas, metodologias e tecnologias de segurança. Embora haja um sem número de tecnologias e técnicas de segurança, que serão apresentados no decorrer do trabalho, o administrador de segurança passa por grandes dificuldades em saber o que fazer para proteger a sua rede, ficando muitas vezes completamente perdido quanto às ações a serem tomadas. O *firewall* cooperativo, o modo de definir as regras de filtragem e o modelo hierárquico de

defesa visam justamente auxiliar no processo de proteção da rede através da apresentação das técnicas, tecnologias e arquiteturas mais adequadas para cada situação.

Algumas questões que serão abordadas neste trabalho são:

- Qual a importância e a necessidade de uma política de segurança?
- Como implementar e garantir um nível de hierarquia entre as comunicações das diversas organizações do ambiente cooperativo?
- A importância e a necessidade da educação dos usuários;
- Quais as fronteiras entre as organizações no ambiente cooperativo?
- Qual tecnologia utilizar para garantir a proteção dos valores da organização? *Firewall*, criptografia, autenticação de dois fatores, biometria, *Single Sign-On* (SSO), *Public Key Infrastructure* (PKI), *IP Security* (IPSec), *Virtual Private Network* (VPN)?
- Quais são os maiores problemas envolvendo *firewalls* e o ambiente cooperativo?
- Como resolver os problemas de regras de filtragem inerentes no ambiente cooperativo?
- Como integrar as diversas tecnologias disponíveis?
- Qual o modelo de segurança proposto?
- Enfim, como garantir a segurança nesse ambiente cooperativo?

2.5 Conclusão

Este capítulo discutiu a importância da informática para os negócios de todas as organizações. A necessidade cada vez maior de conexões resulta em uma complexidade bastante grande nas configurações de redes de todos os envolvidos. Com isso um ambiente cooperativo é formado, que traz junto de si uma série de implicações de segurança, principalmente quanto aos limites entre as redes e aos perigos de triangulações. A formação de um ambiente cooperativo será vista com detalhes no capítulo 11, na parte III do trabalho, que apresenta ainda o modelo de segurança de redes proposto, no capítulo 12.

Capítulo 3

A Necessidade de Segurança

Neste capítulo a segurança possui todo o foco, onde questões sobre investimentos em segurança e os seus mitos, bem como a relação da segurança com as funcionalidades, a produtividade e os riscos envolvidos são discutidos. Pode-se ver também a segurança de redes e a impossibilidade de se ter uma rede totalmente segura.

3.1 A Segurança de Redes

Atualmente a informática é um instrumento utilizado pelo homem para que ele possa realizar seus trabalhos de modo mais fácil, mais rápido, mais eficiente e mais competitivo, produzindo assim os melhores resultados. A rede é um dos elementos principais, permitindo a conexão entre os computadores, de modo que a flexibilidade, a facilidade e a disponibilidade dos recursos resultem em uma maior produtividade, e conseqüentemente, em maiores lucros dentro de uma organização.

A manutenção dessa estrutura de redes passa assim a ser essencial para o bom andamento das organizações, fazendo com que ela precise ser protegida. A segurança, portanto, significa muito mais do que a proteção contra hackers, maus funcionários ou vírus. A segurança significa permitir que as organizações busquem os seus lucros, que são conseguidos através de novas oportunidades de negócios, que são resultados da flexibilidade, facilidade e da disponibilidade dos recursos de informática. Assim, a segurança deve ser considerada uma parte crítica da infra-estrutura necessária para que os negócios da organização possam ser realizados.

Essa importância pode ser vista através das novas oportunidades de negócios surgidas no mundo digital, que condicionam seus sucessos à eficiência da estratégia de segurança. Algumas dessas oportunidades que podem ser exploradas são [FOO 98]:

- ***E-marketing*** - *site* Web;
- ***E-sales*** - venda de produtos e serviços pela rede;
- ***E-service*** - como as referências cruzadas de livros de interesse dos clientes pela Amazon Books;
- ***E-support*** - como a Federal Express, que dá a situação atual da carga em tempo real;
- ***E-supply*** - construção e integração da cadeia de fornecimento entre seus fornecedores e clientes;
- ***E-engineering*** - desenvolvimento de produtos de modo colaborativo.

A segurança assim deve ser vista como o elemento que irá permitir que novas oportunidades possam ser exploradas de modo concreto, de modo que sem ela não existem negócios, pelo menos a longo prazo. A maior indicação de perigo está no fato de pesquisas mostrarem que os negócios on-line trazem junto com si um aumento de 57% nas brechas envolvendo vazamento de informações, enquanto os acessos não autorizados vindos do exterior aumentam em mais de 24% [BRI 99B], resultados da exposição na Internet.

3.2 Mais Evolução, Mais Preocupação com Segurança

Nos tempos do *mainframe*, os aspectos de segurança eram simples, relacionados basicamente com o nome de usuário e a sua senha [DID 98]. Nos dias atuais, o alto grau de conectividade e a grande competitividade trouxeram também outros tipos de problemas. Os avanços tecnológicos vêm trazendo grandes oportunidades de negócios, porém quanto maiores esses avanços, maiores as vulnerabilidades que aparecem e que devem ser tratadas com a máxima atenção. Alguns colocam a culpa na própria indústria, que não estaria dando a devida atenção aos aspectos de segurança de seus produtos, estando mais interessada em finalizar rapidamente o produto, para colocá-lo no mercado antes que seus concorrentes.

De fato, essa falta de cuidados com relação à segurança na implementação dos produtos pode ser observada em alguns fatos:

- O surgimento da suíte de protocolos TCP/IP e o advento da Internet fez com que o escopo das invasões crescesse em proporções mundiais, onde qualquer um pode atacar qualquer outro;
- A criação de linguagens macro nos aplicativos como o Word ou o Excel fez surgir uma nova geração de vírus, que se espalham com uma velocidade nunca antes vista (também através de *e-mails*), já que qualquer tipo de arquivo de dados pode estar infectado, não mais somente os arquivos executáveis e discos de *boot*;
- A Web e as linguagens criadas para a Internet, como o JavaScript ou o ActiveX, são de difícil controle e podem causar sérios problemas caso contenham códigos maliciosos e serem executados em uma rede interna;
- O avanço nas pesquisas de clonagem podem resultar em mais problemas envolvendo a segurança, em principal contra a biometria (capítulo 10), que vem sendo desenvolvida para eliminar problemas existentes nas tecnologias tradicionais de autenticação (capítulo 10).

3.3 Segurança como Parte dos Negócios

Nas décadas de 70 e 80, a informática fazia parte da retaguarda dos negócios das organizações, onde o foco principal era a confidencialidade dos dados. Era a época dos *mainframes*, e a proteção era voltada para os dados. Entre as décadas de 80 e 90, com o surgimento dos ambientes de rede, a integridade passou a ser de suma importância, e a proteção era feita não em cima dos dados, mas sim das informações. A informática fazia parte da administração e da estratégia da organização. A partir da década de 90, o surgimento das redes IP fez com que o foco fosse mudado para a disponibilidade. A informática agora faz parte direta dos negócios, e o conhecimento e o capital intelectual é que devem ser protegidos.

Pode-se definir dados como sendo um conjunto de *bits* armazenados, como nomes, endereços, data de nascimento. Um dado passa a ser uma informação quando ele passa a ter um sentido, como as informações referentes a um cliente especial. O conhecimento é o conjunto de informações que agrega valor ao ser humano e à organização, valor esse que resulta em uma vantagem competitiva, tão importante no mundo atual.

Nesse mundo globalizado, onde as informações atravessam fronteiras com velocidade espantosa, a proteção do conhecimento é de vital importância para a sobrevivência das organizações. As dimensões dessa necessidade passam a influenciar diretamente os negócios. Uma falha, uma comunicação com informações falsas, ou um roubo ou fraude de informações podem trazer graves conseqüências para a organização, como a perda de mercado, negócios e conseqüentemente de capital. Deste modo, a proteção, não só das informações e de seu capital intelectual, mas também de todos os recursos envolvidos na infra-estrutura de rede, deve ser tratada com a sua devida distinção. E como o conhecimento é o principal capital das organizações, proteger esse conhecimento significa proteger o seu próprio negócio. Assim, a segurança passa a fazer parte do processo de negócio das organizações.

Um problema existente é que muitos processos de negócios não foram concebidos no contexto de um ambiente distribuído e de redes, e muitos outros foram desenvolvidos sem o enfoque na segurança, na abordagem "funcionando está ótimo". Isso resulta em uma aplicação de "*band-aid*" para os problemas de segurança, sem uma arquitetura de segurança racional. Essa abordagem de remendos é considerada melhor do que não ter nenhuma abordagem, porém ela cria um falso senso de segurança, que na realidade, por ser superficial e utilizar técnicas parciais, pode aumentar a vulnerabilidade da organização. Sem um plano e arquitetura de segurança bem definidos, as tecnologias de segurança podem ser mal interpretadas e mal utilizadas, como o *firewall*, que se mal configurado e mal utilizado, não possui função alguma na rede [DYK 98].

Um exemplo da estreita relação entre a segurança e os negócios pode ser visto no seguinte exemplo: na medida em que as organizações se movem para a Web, vendendo seus produtos diretamente ao consumidor via meios eletrônicos, a segurança passa a ser o coração dessa venda. A transmissão do número do cartão de crédito deve ser segura, os dados do consumidor devem ser protegidos, e os dados do cartão de crédito recebidos devem ser muito bem armazenados. Assim, a segurança passa a ser, em primeiro momento, o principal responsável pelo negócio, o elemento que permite que a venda realmente aconteça. Se antes o setor comercial era o responsável pelas decisões de vendas, hoje, no mundo eletrônico, o profissional de segurança possui um papel importante, fazendo parte direta nos negócios da organização. É ele o responsável pela segurança das transações eletrônicas, passando assim de uma posição técnica obscura para a linha de frente dos negócios da organização [SEC 98].

Assim, a segurança das informações e os negócios estão estreitamente ligados. Hoje o profissional de segurança está partindo para um trabalho mais orientado à essa nova realidade, onde ele tem que ouvir pessoas, de modo a entender e saber como aplicar as tecnologias, de acordo com a organização e as suas necessidades [BRI 99].

3.4 Como a Segurança é Vista Hoje

Apesar da segurança ser hoje essencial para os negócios das organizações, a dificuldade em entender a sua importância ainda é bastante grande. No mundo atual, muitas vezes, quase que invariavelmente, a única segurança existente é a obscuridade. E isso é realmente muito ruim para a organização, pois mais cedo ou mais tarde alguém irá descobrir que um grande tesouro está à sua completa disposição. É apenas uma questão de tempo para que isso aconteça, causando grandes prejuízos, sejam elas financeiras, morais ou de reputação. E todos sabem que uma boa reputação pode demorar anos para ser construída, mas pode ser destruída em questão de instantes. É claro que esse aspecto depende da área de atuação da organização. Por exemplo, para um banco, um incidente de segurança, por menor que ela seja, fará com que os seus clientes percam a confiança nos serviços prestados, e eles procurarão outros para movimentarem seus recursos financeiros. A grande questão portanto está na confiança. Bancos trabalham com isso, de forma que o grande negócio deles é a confiança que obtém de seus clientes.

E é justamente nela que se baseia ou se basearão os negócios da maioria das organizações. Tudo isso como resultado da globalização da economia mundial e da grande estrutura de redes que vêm sendo construído pelas organizações. Pode ser visto que a convergência para as redes é um processo natural, pois ela permite que os negócios sejam realizados de modo mais eficiente, dinâmico e produtivo, o que faz com que as relações entre as organizações e seus clientes, fornecedores, parceiros e funcionários dependam cada vez mais dessa estrutura. Assim, os ambientes cooperativos nascem e crescem, desenvolvendo um novo modelo de negócios baseado nas redes. E esses ambientes cooperativos necessitam de um grande grau de confiança para que funcionem de maneira adequada. Do mesmo modo que os bancos dependem da confiança que recebem de seus clientes, o mesmo ocorrerá com as demais organizações.

A organização que faz parte de um ambiente cooperativo deve entender que a segurança agora é essencial para o sucesso de seus negócios. Se nos bancos a relação de confiança era entre a instituição e seus clientes, agora essa relação ocupa dimensões ainda maiores, onde a

confiança não deve apenas existir entre a organização e seus clientes, mas também entre a organização e seus fornecedores, parceiros, distribuidores e funcionários. Isso porque um incidente de segurança em um único ponto desse ambiente pode comprometer todo o ambiente cooperativo. Por exemplo, se em uma cadeia do processo de negócio um fornecedor sofrer algum incidente de segurança, esse incidente pode se alastrar por todos os outros pontos do ambiente cooperativo. Isso pode resultar em um rompimento das relações de confiança entre os pontos do ambiente cooperativo, já que a falha de um pode trazer prejuízos para todos.

A segurança de redes ainda é um campo novo, e muitos ainda não conseguem enxergar a sua importância, imaginando apenas que as soluções são caras e não trazem nenhum retorno financeiro. Isso faz com que os gerentes prefiram aplicar seus recursos em novas soluções que podem trazer vantagens visíveis aos olhos de todos.

Esse é o maior problema da segurança - a solução de segurança é imensurável e não resulta em soluções onde todos podem notar que alguma coisa foi feita. Pelo contrário, a segurança tem o papel de justamente evitar que alguém perceba que alguma coisa está errada na infra-estrutura tecnológica da organização. O fato é que ninguém percebe a existência de segurança, apenas a inexistência dela, quando um incidente acontece e resulta em prejuízos gigantescos. A segurança hoje possui esse conceito, a de que é um artigo caro e dispensável, e só é necessário quando algum ataque acontece e traz prejuízos à organização.

O que é realmente necessário é que o ambiente cooperativo seja analisado de acordo com a sua importância e com os grandes benefícios que esse ambiente pode trazer à organização. E é impossível que um ambiente cooperativo seja considerado sem que as questões relacionadas à segurança sejam discutidas.

O grande ideal é de que a segurança passe a ser um processo transparente dentro das organizações. Alguma coisa parecida com o que aconteceu com a qualidade. Todos começaram a buscar a qualidade em seus negócios, de tal forma que hoje, quando qualquer serviço for prestado, ou qualquer produto for vendido, eles devem possuir qualidade sem que isso seja ao menos discutido. Não é mais questão de avaliar se pode, mas sim de que deve possuir qualidade. O mesmo caminho deverá ser seguido agora com relação à segurança. A questão não deve ser se existe ou não a segurança, mas sim em que nível ela se encontra. Ela deve fazer parte do processo de negócios, já que se não existe a segurança, então não existe o negócio. O "funcionando está bom", todo mundo sabe fazer. Agora, o "funcionando com segurança", será o grande

diferencial entre as boas organizações, confiáveis, e as más, que não receberão a confiança necessária para o seu sucesso, e tenderão ao fracasso.

Apesar de bastante cru, seguir essa idéia de que a segurança e o ambiente cooperativo devem andar juntos, trará grandes benefícios à economia global.

3.5 Investimentos em Segurança

Um dos principais problemas para a implantação da segurança é o seu orçamento, comumente pequeno ou praticamente inexistente. O principal ponto a ser considerado é que, como foi visto no tópico anterior, os gerentes geralmente não possuem a visão necessária para enxergar a importância de uma boa estratégia de segurança. Felizmente isso começou a mudar com o advento dos vírus Melissa e ExploreZip, que causou problemas para diversas organizações, porém em uma área bem específica, a de anti-vírus. Um acontecimento mais recente foram os ataques distribuídos, que tornaram inacessíveis grandes *sites* como a Amazon, Yahoo, UOL, E-Bay, Zipmail, entre outros. A segurança geralmente é vista como um elemento supérfulo dentro das organizações, criando-se diversos mitos quanto ao assunto, como podem ser vistos na seção 3.6. Como as próprias organizações possuem orçamentos apertados, a segurança acaba ficando em segundo plano, geralmente vindo à tona apenas quando são extremamente necessários, ou seja, apenas quando a organização sofre algum incidente de segurança, como um ataque ao banco de dados e divulgação pública de material confidencial.

Essa visão reativa, com as decisões de segurança sendo tomadas apenas após um incidente, traz uma série de conseqüências negativas para a organização, principalmente no que se refere à perda de credibilidade e conseqüente perda de mercado. Isso acaba resultando em um grande problema para os administradores de segurança, que acabam não possuindo os recursos necessários para as soluções de segurança. O que é necessário é fazer com que os gerentes passem a enxergar a segurança da organização como um elemento essencial para o seu sucesso nesse mundo onde as redes fazem uma grande diferença no mercado. Fazer a gerência entender que a solução de segurança não gera gastos, mas sim é um investimento que permite a continuidade de seus negócios, é o ponto chave dentro dessa estratégia.

Essa visão parece estar mudando aos poucos, como pode ser observado na pesquisa realizada em 1999 [BRI 99B], que mostra que o nível de segurança das organizações têm aumentado

(85% melhoraram o nível de segurança), porém ao mesmo tempo os incidentes de segurança também continuam a aumentar (23% contra 12% em 1998).

A pesquisa mostra também uma tendência de aumento gradual no nível de investimentos com segurança, na medida em que as organizações começam a tratá-la como o habilitador de negócios em potencial, passando de um mal necessário para um componente no novo modelo de "organização virtual". Na pesquisa, a percentagem das organizações que destinavam até $50 mil diminuiu de 52% em 1998 para 44% em 1999, enquanto cresceu a percentagem das organizações que destinam mais de $1 milhão (de 8% em 1998 para 11% em 1999, e para 13% para 2000). A média dos orçamentos cresceu 21,7% de 1998 para 1999. As organizações que gastam mais de $500 mil com segurança cresceu de 15% em 1998 para 19% em 2000 [BRI 99B].

Apesar do crescimento do orçamento com a segurança, apenas 34% acham que o orçamento é suficiente, sendo que 63% dizem que a falta de orçamento é um obstáculo para a proteção adequada dos dados e recursos da organização (58% em 1998) [BRI 99B].

Dividido em setores industriais, a indústria da aviação é a que utiliza o maior número de produtos e serviços de segurança (19 de 23). As instituições educacionais formam o setor que utiliza o menor número de soluções (7 de 23), sendo que apenas 58% usam *firewall* (média geral da indústria de 82%) e 10% usam VPN (28% de média para toda a indústria). O setor militar utiliza pouco o *firewall* (75%) e o IDS (53%), porém é o que mais utiliza a *Public Key Infrastructure* (PKI) (41% contra 21% da indústria em geral). Quem mais utiliza *smart cards* é a indústria da aviação (42%), ao contrário do setor médico/biológico (10%). A biometria é a solução de segurança menos utilizada, por enquanto, com apenas 7% utilizando a solução. O setor do governo e o setor de tecnologia são os setores que mais utilizam a biometria, com 10% e 11%, respectivamente [BRI 99B].

Os valores relacionados à segurança são difíceis de serem quantificadas, pois o que está em jogo são, além dos recursos considerados tangíveis (horas de trabalho para a recuperação, equipamentos, softwares), os recursos intangíveis (valor do conhecimento, imagem da organização). Além disso, os cálculos sempre são feitos em cima de suposições, como "SE o sistema for atingido, teremos $$$ de prejuízos, então o melhor é investir $$$ para a proteção dos recursos da organização".

O foco é identificar os valores das informações da organização, e então calcular e avaliar os impactos nos negócios. Essa identificação permite entender exatamente o que ocorre se a orga-

nização sofre danos nessas informações [BRI 99]. Assim, a análise de risco e o uso de uma metodologia para quantificar e qualificar os níveis de segurança de cada recurso da organização são necessários. Isso auxilia na criação da proposta e das justificativas de investimentos para a implantação de um sistema de segurança adequado.

3.6 Mitos sobre Segurança

Alguns mitos sobre segurança utilizados pelos gerentes para "tapar os olhos com relação à segurança" são [JOH 98]:

- "Isso não acontecerá conosco";
- "Nós utilizamos os melhores sistemas, então eles devem ser seguros";
- "Nós utilizamos as últimas versões dos sistemas dos melhores fabricantes";
- "Nossos fornecedores irão nos avisar caso alguma vulnerabilidade seja encontrada";
- "Nós tomamos todas as precauções de modo que testes não são necessários";
- "Problemas de segurança é com o departamento de IT";
- "A companhia de IT contratada irá cuidar da segurança";
- "Nós não precisamos nos preocupar com a segurança, pois segurança é um luxo para quem tem dinheiro".

3.7 Riscos e Considerações quanto à Segurança

Diversos aspectos devem ser levados em consideração quando uma rede passa a constituir uma parte crítica dentro de uma organização. Alguns dos riscos existentes e algumas considerações a serem feitas são [AVO 94]:

- A falta de uma classificação das informações quanto ao seu valor e à sua confiabilidade, que servem de base para a definição de uma estratégia de segurança adequada, resultam em um fator de risco para a organização, além de dificultar o dimensionamento das perdas resultantes em um ataque;

- O controle de acesso mal definido faz com que os usuários que são autenticados no início da conexão possuam acesso irrestrito a quaisquer partes da rede interna, inclusive a partes do sistema que não são necessários para a realização de suas tarefas;
- A dificuldade de controle do administrador sobre todos os equipamentos da rede interna faz com que eles não possam ser considerados confiáveis. *Bugs* nos sistemas operacionais ou nos softwares utilizados por esses equipamentos podem abrir brechas na rede interna, como pode ser visto na seção 4.6.1;
- A Internet deve ser considerada um ambiente hostil, e portanto não confiável. Assim, todos os usuários da Internet devem ser considerados não confiáveis;
- As informações que trafegam pela rede estão sujeitas a serem capturadas;
- Senhas que trafegam pela rede estão sujeitas a serem capturadas;
- *E-mails* que são enviadas para fora da rede interna podem ser capturadas, lidas, modificadas e falsificadas;
- Qualquer conexão entre a rede interna e qualquer outro ponto pode ser utilizado para ataques à rede interna;
- Telefones podem ser grampeados e as informações que trafegam pela linha, sejam voz ou dados, podem ser gravados;
- A segurança pela obscuridade torce para que o invasor não saiba dos problemas com segurança e dos valores disponíveis na rede interna. Até quando?

3.8 Segurança vs Funcionalidades

Até há pouco tempo atrás as organizações implementavam suas redes apenas com o objetivo de prover funcionalidades que permitiam promover a evolução de seus processos organizacionais. A preocupação com a segurança praticamente não existia, porém o mundo atual exige que as redes das organizações sejam voltadas para o seu próprio negócio, requerendo portanto a segurança. Em um primeiro momento, a falta de um planejamento em segurança pode parecer bom, pois tudo funciona adequadamente, porém os problemas de segurança usualmente aparecem depois, o que pode resultar em custos estratosféricos para que esses problemas sejam resolvidos, principalmente em grandes ambientes [FIST 98].

A importância da segurança na rede da organização cresce mais rapidamente ainda quando se leva em consideração o rápido aumento da complexidade das conexões. Um ponto fundamental quando se discute o assunto é que a segurança é inversamente proporcional às funcionalidades, ou seja, quanto maiores as funcionalidades, como serviços, aplicativos e demais facilidades, menor é a segurança desse ambiente. Isso pode ser explicado porque a segurança é comprometida através de:

- Exploração de vulnerabilidades em sistemas operacionais, aplicativos, protocolos e serviços;
- Exploração dos aspectos humanos das pessoas envolvidas;
- Falha no desenvolvimento e implementação da política de segurança.

Esses tópicos serão vistos com maiores detalhes no capítulo 4. Quando as vulnerabilidades que podem existir em sistemas operacionais, aplicativos, protocolos e serviços são analisadas, pode-se considerar que ela é resultante de *bugs*, que são decorrentes de falhas em seu código, em seu *design* ou em sua configuração. Assim, quanto maior o número de sistemas, maiores as responsabilidades dos administradores, e maior a probabilidade de existência de *bugs*, que podem ser explorados. As obrigações dos administradores quanto à manutenção da segurança devem estar claramente definidas na política de segurança da organização, como é o caso do acompanhamento das novidades e boletins dos sistemas que estão sendo utilizados na organização, principalmente quanto a relatórios de segurança e instalação de *patches* de correção. Estes e outros pontos referentes à política de segurança serão discutidos no capítulo 5.

Um estudo da IDC propôs uma fórmula para determinar os pontos de vulnerabilidades de uma rede: o número de pontos de vulnerabilidades é igual ao número de recursos críticos da organização multiplicado pelo número de usuários que possuem acesso a esses recursos. Assim, se um servidor NT possui 10000 arquivos e 100 usuários, existem 1 milhão de pontos de acessos vulneráveis. Como a prevenção de todas as brechas são impraticáveis, o objetivo é balancear a segurança com os riscos, minimizando os impactos que uma brecha de segurança pode causar à organização. Os gastos com as brechas de segurança estão em torno de $256.000 por ano por organização, segundo a pesquisa [BRI 99B].

3.9 Segurança vs Produtividade

A administração da segurança de uma organização é uma tarefa complexa, na medida em que ela deve ser dimensionada sem que a produtividade dos usuários seja afetada. Geralmente, a segurança é antagônica à produtividade dos usuários, no sentido de que, como visto no tópico anterior, quanto maiores as funcionalidades, maiores as vulnerabilidades existentes. Isso leva os administradores a restringirem ao máximo os serviços externos que os usuários podem acessar, de modo a minimizar ao máximo os riscos.

O problema é que uma política de segurança muito restritiva geralmente afeta a produtividade do usuário. Por exemplo, se o FTP for bloqueado com o objetivo de prevenir a entrada de cavalos-de-tróia, e o usuário necessita utilizar esse serviço para que seu trabalho continue, ele certamente buscará maneiras de driblar essa restrição do *firewall*. O usuário poderá instalar um modem em seu equipamento, ou tentar achar brechas que furem o bloqueio do *firewall*. Quando isso acontece, os objetivos não são alcançados, pois a segurança é comprometida pelas ações dos usuários, e a produtividade desses usuários é comprometida, já que eles perdem tempo tentando achar maneiras de driblar o *firewall*.

Por isso é importante uma política de segurança bem definida e bem balanceada, tanto com relação aos serviços externos que os usuários podem acessar, quanto com relação aos serviços internos que os usuários externos podem acessar. O objetivo é criar uma política que defina idealmente apenas os serviços realmente necessários.

Outro ponto a ser considerado na definição desses serviços que serão permitidos é quanto a serviços como o RealAudio, RealVideo, ICQ e *chats*, que constituem um problema, já que comprometem o nível de produtividade da organização, além de consumir uma grande largura de banda da rede. Alguns deles, como o ICQ, ainda trazem maiores vulnerabilidades à rede interna da organização.

3.10 Uma Rede Totalmente Segura

A segurança é um aspecto complexo, que envolve aspectos humanos, sociais e tecnológicos, de modo que afirmar que uma organização está 100% segura é na realidade um grande erro. Simplesmente não existe um modelo de segurança à prova de *hackers*. Será visto no capítulo 4 que os *hackers* podem atuar de diversas maneiras, e mesmo os próprios funcionários maliciosos de

uma organização podem fazer esse papel de *hacker* (*insiders*). Se a segurança envolve não apenas aspectos tecnológicos, mas também aspectos técnicos (um bom administrador de segurança), aspectos sociais (funcionários inescrupulosos que roubam informações confidenciais da própria organização), aspectos humanos (funcionários inocentes que sofrem com a engenharia social) e aspectos educacionais (funcionários devem saber pelo menos como escolher senhas boas), então com toda essa complexidade o objetivo das organizações é a de tentar proteger ao máximo os recursos da organização, e não protegê-los totalmente.

Diversos aspectos contribuem para se medir essa "máxima proteção", entre elas definir os recursos que devem ser protegidos, definir quem irá administrar a segurança, e principalmente, qual o valor que será utilizado como investimento para a segurança.

No mínimo, essa segurança inclui uma política e procedimentos compreensivos, controle dos usuários, e autenticação de todos os acessos ao sistema, das transações e das comunicações. Inclui também a proteção dos dados, além do constante monitoramento e a evolução do nível de segurança geral. Outro ponto importante é que as novas tecnologias devem ser utilizadas antes que os *hackers* utilizem essas tecnologias contra a organização.

Assim, a segurança total pode levar à loucura, e a segurança parcial assume, por definição, os riscos. As organizações portanto devem definir o nível de segurança de acordo com as suas necessidades, já assumindo os riscos [DYK 98]. Isso faz com que o plano de contingência seja um dos pontos essenciais dentro do esquema de segurança de uma organização.

O objetivo não é construir uma rede totalmente segura, mas sim um sistema altamente confiável. O sistema deve ser capaz de anular os ataques mais casuais de *hackers* e também ser capaz de tolerar acidentes, como o de um tubarão que rompe os cabos de transmissão. As falhas benignas devem ser toleradas pelos sistemas. Essas vulnerabilidades devem ser colocadas em um lugar onde não podem causar problemas. Assim, uma rede nunca será totalmente segura, mas deve-se procurar meios de torná-la, pelo menos, mais confiáveis, como descrito em "Trust in Cyberspace" [KRO 99].

3.11 Conclusão

Com a rápida evolução que pode ser acompanhada no mundo dos negócios, onde as conexões entre organizações significam vantagens competitivas, a segurança de redes passa a ser fundamental. Porém, captar investimentos para a implementação de uma estratégia de segurança

envolve diversos desafios, onde riscos e mitos de segurança devem ser combatidos. As funcionalidades envolvidas com o andamento dos negócios, bem como a produtividade dos usuários, são afetadas com as medidas de segurança adotadas, de modo que elas devem ser bem avaliadas e estudadas para que não causem impactos significativos para os envolvidos. A segurança é necessária, porém a sua estratégia de implementação deve ser bem definida, medindo-se custos e benefícios, já que a segurança total não é possível.

Capítulo 4

Os Riscos que Rondam as Organizações

Este capítulo apresenta os riscos a que as organizações estão sujeitos. Os possíveis atacantes e os métodos, técnicas e ferramentas utilizados por eles são apresentados, mostrando que as preocupações com a segurança devem ser tratadas com a máxima atenção e cuidado, para que a continuidade dos negócios das organizações não seja afetada. É contra esses riscos que as organizações têm que lutar, principalmente através das técnicas, tecnologias e conceitos a serem discutidos na parte II deste trabalho.

4.1 Os Potenciais Atacantes

O termo genérico para quem realiza um ataque em um sistema de computadores é *hacker*. Essa generalização porém possui diversas ramificações, já que os ataques aos sistemas possuem objetivos diferentes, e os seus sucessos dependem do grau de segurança dos alvos, ou seja, sistemas bem protegidos são mais difíceis de serem atacados, exigindo maior habilidade dos *hackers*.

Os *hackers*, por sua definição original, são aqueles que utilizam seus conhecimentos para invadir sistemas, não com o intuito de causar danos à vítima, mas sim como um desafio às suas habilidades. Eles invadem os sistemas, capturam ou modificam arquivos para provar a sua capacidade, e compartilham a sua proeza com seus colegas. Eles não possuem a intenção de causar danos à vítima, apenas demonstrar que o conhecimento é o poder. Exímios programadores e conhecedores dos segredos que envolvem as redes e os computadores, eles geralmente não gostam de ser confundidos com *crackers*. Com o advento da Internet, os diversos ataques pelo

mundo foram atribuídos a *hackers*, mas eles refutam essa idéia dizendo que *hacker* não é *cracker*. *Crackers* são elementos que invadem sistemas para roubar informações e causar danos à vítima. Antes, os *crackers* eram vistos como aqueles que quebravam códigos e proteções de softwares.

Atualmente porém, com o crescimento da Internet e a conseqüente facilidade em se obter informações e ferramentas de ataques, a definição de *hackers* mudou. A própria imprensa mundial tratou de mudar esse conceito. Agora, qualquer incidente de segurança é atribuído a *hackers*, em seu sentido genérico. A palavra *cracker* não é mais vista nas reportagens, a não ser como *cracker* de senhas, software que é utilizado para se descobrir senhas cifradas.

Estudos sobre *hackers* foram realizados, e o psicólogo canadense Marc Rogers chegou ao seguinte perfil do *hacker*: obsessivo de classe média, branco, homem entre 12 e 28 anos, com pouca habilidade social e possível história de abuso físico e social. Uma nova classificação de tipos de invasores também foi apresentado no *RSA Data Security Conference* [MOD 99]:

- ***Kiddies*** - iniciantes;
- ***Cyberpunks*** - mais velhos, mas ainda anti-sociais;
- ***Insiders*** - empregados insatisfeitos;
- ***Coders*** - os que atualmente escrevem sobre suas "proezas";
- **Profissionais** - contratados;
- ***Full Fledged*** - cyber-terroristas.

É importante lembrar, porém, que não são só os *hackers* que causam problemas de segurança nos sistemas. Os usuários, sejam eles autorizados ou não, mesmo sem intenções malévolas, podem causar danos ou negar serviços de redes através de seus erros e de sua própria ignorância [HTTP 01].

4.1.1 Script Kiddies

Também conhecidos como *newbies*, os *script kiddies* trazem diversos problemas para as organizações. Geralmente são inexperientes e novatos, que pegam os programas que se encontram prontos na Internet. Devido à grande facilidade em se conseguir esses programas, os *script kiddies* são considerados perigosos para um grande número de organizações, que são aquelas

que não possuem uma política de segurança bem definida, e portanto sempre possuem alguma brecha de segurança, principalmente as geradas pela falta de atualização de um *patch* do servidor. Isso é o suficiente para que os *script kiddies* executem os programas pegos na Internet contra seus servidores e causem estragos consideráveis.

É devido principalmente aos *script kiddies* que as organizações começaram a prestar mais atenção em seus problemas de segurança. Eles são a imensa maioria dos "*hackers*" na Internet, e um grande número de incidentes de segurança são causados pelos *script kiddies*. Os seus limitados conhecimentos podem ser vistos em relatos onde servidores registravam tentativas de ataques em ambientes Windows utilizando-se comandos específicos do Unix.

4.1.2 Cyberpunks

São os *hackers* dos tempos românticos, aqueles que se dedicam à invasões de sistemas por puro divertimento e desafio. Possuem extremo conhecimento, e são obcecados pela privacidade de seus dados, e portanto utilizam a criptografia em todas as suas comunicações. A preocupação principal é contra o governo, que com o *Big Brother* podem estar acessando as informações privadas dos cidadãos. Os *hackers* mais paranóicos, que acreditam em teorias de conspiração, tendem a virar *cyberpunks*.

Geralmente são eles que encontram novas vulnerabilidades em serviços ou sistemas, prestando assim um favor às organizações, quebrando sistemas e publicando as vulnerabilidades encontradas. Isso contribui para que a indústria de software corrija seus produtos, e melhor do que isso, passem a desenvolvê-los com maior enfoque na segurança. Infelizmente, porém, a indústria ainda prefere corrigir seus produtos a adotar uma metodologia de desenvolvimento com enfoque na segurança.

4.1.3 Insiders

Os *insiders* são os maiores responsáveis pelos incidentes de segurança nas organizações. Pesquisas indicam que mais de 60% dos ataques são feitos pelos *insiders*, ou seja, a maioria dos ataques vêm a partir da própria rede interna, através de funcionários, ex-funcionários ou pessoas que conseguem se infiltrar dentro das organizações. Uma série de questões estão envolvidas nesse tema, desde a engenharia social, até a relação do funcionário com o chefe, passando pelo suborno e a espionagem industrial.

De acordo com a pesquisa do *American Society for Industrial Security* (ASIS) realizada em 1997, mais de 56% das 172 companhias pesquisadas sofreram tentativas de apropriação indevida de informações privadas, e num período de 17 meses, mais de 1100 incidentes de roubo de propriedade intelectual foram documentados, resultando em prejuízos na ordem de $44 bilhões, o que é 5 vezes maior do que os valores da pesquisa anterior [DEN 99].

Essas estimativas cresceram para $100 bilhões em 1998, sendo que uma das razões para o crescimento da espionagem industrial é que a economia hoje é baseada no conhecimento, onde as informações são as grandes responsáveis pelas vantagens competitivas. Isso faz com que as conseqüências de um roubo sejam potencialmente desastrosas, influenciando até mesmo na própria sobrevivência da organização [SEC 98-1]. De fato, o capital intelectual encabeça a economia eletrônica atual, e alguns exemplos de que a espionagem industrial é um fato pode ser visto nos exemplos dos roubos de projetos e fórmulas da General Electrics, Kodak, Gilette e Schering-Plough [ULS 98].

A espionagem industrial, atribuída geralmente a *insiders*, é considerada uma nova modalidade de crime organizado, assim como as máfias e os cartéis de drogas. Em um nível mais alto, o que se vê é o surgimento de organizações especializadas em espionagem industrial, sendo que o próprio governo de alguns países, como o Japão, França e Israel, financiam esses trabalhos, institucionalizando a prática. Na França, por exemplo, a agência de inteligência *Direction Generale de la Securite Extrieure* (DGSE) têm o trabalho facilitado, principalmente em hotéis, onde geralmente possuem grampos e câmeras escondidas, de modo que segredos de executivos de organizações concorrentes correm o risco de serem roubados e revelados. As maiores companhias americanas avisam seus executivos sobre esses perigos [SEC 98-1].

Um caso envolvendo companhias de investimento mostra a importância da segurança contra a espionagem industrial e contra os ataques a sistemas de computadores no competitivo mundo atual. A Reuters Holdings PLC e a Bloomberg LP são concorrentes no mercado de investimentos, e o uso de computadores é essencial para a análise dos investimentos e das tendências do mercado. O sistema da Bloomberg era considerado melhor que o da Reuters, razão pela qual ganhava cada vez mais o mercado. Isso fez com que a Reuters fundasse a Reuters Analytics para o desenvolvimento de um produto de análise competitivo. Em janeiro de 1998 a Reuters Analytics decidiu utilizar uma conduta diferente da habitual, ou seja, contratou consultores para invadir os computadores da Bloomberg, o que resultou em acesso à informações que continham códigos

das operações e documentos descrevendo as funcionalidades do sistema. A Bloomberg não descobriu quais métodos foram utilizados para a invasão, porém sabe-se que ex-funcionários da Bloomberg que estavam trabalhando na Reuters Analytics estavam envolvidos [DEN 99].

No nível interno das organizações, os próprios funcionários são as suas maiores ameaças, pois eles possuem o tempo e a liberdade necessários para procurar alguma coisa nas mesas das pessoas, ler memorandos privados, copiar documentos, abusar da amizade de colegas e copiar facilmente uma grande base de dados, que pode valer milhões, em um disco Zip. O fato mais marcante é que essas pessoas conhecem as operações, a cultura e detalhes da organização, que facilitam a espionagem. Com isso eles sabem onde estão os segredos, quem são os concorrentes e sabem, principalmente, como apagar seus rastros. Esses fatos fazem com que os *insiders* sejam difíceis de serem identificados [SEC 98-1].

A identificação dos *insiders* pode ser difícil, mas geralmente eles são funcionários descontentes com os seus trabalhos, que sentem que tem os seus trabalhos subestimados pelo chefe. Eles são geralmente mal-tratados, e querem mostrar os seus reais valores e fazerem alguma coisa para se sentirem importante. Esse tipo de funcionário pode ser facilmente manipulado por concorrentes, e eles sabem como persuadir os funcionários que se encontram em posição não muito confortável dentro da organização [SEC 98-1]. Um outro tipo de *insider* são aqueles que buscam alguma atividade excitante para quebrar a rotina de trabalho chato. Os *insiders* são de extrema importância, pois a organização pode estar perdendo espaço, mercado e imagem para o concorrente sem saber o motivo. Será que não houve espionagem e roubo de algumas informações, que chegaram nas mãos dos concorrentes?

Um cuidado especial deve ser tomado com relação a ex-funcionários, que são muitas vezes os elementos mais perigosos. Se um funcionário foi demitido, ele pode querer vingança. Se saiu da organização sob bons termos, pode querer demonstrar seus conhecimentos e valores para seu novo chefe, que pode ser um concorrente da ex-organização [SEC 98-1].

Funcionários terceirizados também podem constituir um risco, já que, se por um lado não possuem acesso a informações confidenciais, passam a conhecer os procedimentos, os hábitos e os pontos fracos da organização, que podem então ser explorados posteriormente. Os funcionários terceirizados podem ainda passar a aceitar subornos para divulgação de informações consideradas confidenciais, ou mesmo subornar os funcionários da organização em busca de segredos industriais [SEC 98-1].

O controle do pessoal de segurança e de limpeza também é importante, já que geralmente eles possuem acesso irrestrito a todos os locais, inclusive à sala de CPU. Como a sala de CPU deve ser limpa por alguém, a engenharia social poderia ser utilizada aqui para que o acesso a áreas restritas seja obtido [SEC 98-1].

Alguns exemplos a seguir comprovam os perigos que as organizações correm com os *insiders* [DEN 99]:

- Funcionários confiáveis - Em março de 1999, um cientista nuclear americano do Los Alamos National Laboratory foi acusado de vender segredos da tecnologia de armas nucleares para a China desde 1980. Em outro caso, de 1994, um funcionário do Ellery Systems em Colorado, Estados Unidos, utilizou a Internet para transferir um software avaliado em $1 milhão para um concorrente da China.

- Funcionários subornados ou enganados - Um espião alemão, Karl Hinrich Stohlze, seduziu uma funcionária de uma empresa de bio-tecnologia de Boston para conseguir informações confidenciais dessa empresa, que incluía métodos de pesquisas de DNA e informações sobre o status dos projetos da companhia. A funcionária foi demitida, porém não processada. Apesar disso, o espião alemão continua trabalhando, agora na Europa;

- Funcionários antigos - Em 1993, Jose Ignacio Lopez e mais 7 outros funcionários deixaram a General Motors para se transferirem para a Volkswagen. Junto eles levaram 10000 documentos privados da GM, que incluía segredos sobre novos modelos de carros, futuras estratégias de vendas e listas de compras. Em 1996, Lopez foi processado e a GM foi indenizada em $100 milhões.

Através desses exemplos, pode-se verificar que a segurança é muitas vezes um problema social, e não um problema tecnológico, demonstrando que os aspectos humanos, sociais e pessoais não podem ser esquecidos na definição da estratégia de segurança.

Apesar de parecer uma prática anti-ética e extremamente ilegal, nem todas as maneiras de se conseguir informações competitivas são contra a lei. A obtenção de informações de outras organizações constitui o trabalho de diversos profissionais, e existe até mesmo uma organização desses profissionais, o *Society of Competitive Intelligence Professionals* (SCIP). O antigo CEO da IBM, Louis Gerstner, formou em abril de 1998 12 grupos de inteligência para a obtenção de informações privilegiadas, que são colocadas em um banco de dados central, que é acessado pelos principais executivos da IBM. O trabalho desse tipo de profissionais está no limiar entre o ético

e o anti-ético, e uma das regras desses profissionais é nunca mascarar a sua verdadeira identidade [DEN 99].

4.1.4 Coders

Os *coders* são os *hackers* que resolveram compartilhar seus conhecimentos escrevendo livros ou conferindo palestras e seminários sobre suas proezas. Ministrar cursos também faz parte das atividades dos *coders*, que parecem ter sido influenciados pelo aspecto financeiro.

4.1.5 White Hat

São também conhecidos como "*hackers* do bem", "*hackers* éticos", samurais ou *sneakers*, que utilizam seus conhecimentos para descobrir vulnerabilidades nos *sites* e aplicar as correções necessárias, trabalhando de maneira profissional e legal dentro das organizações. Eles vêem a si próprios como guerreiros que protegem os sistemas das organizações que os contratam contra os *hackers* não-éticos, sendo assim os responsáveis pelos testes de invasões, onde simulam ataques para medir o nível de segurança da rede da organização. Uma série de considerações devem ser analisadas antes de contratar um *white hat*, como definir os limites de uma invasão, a fim de se evitar que dados confidenciais sejam expostos, deixar claro em contrato que as informações obtidas permanecerão em sigilo, e garantir que todas as correções sejam implementadas.

A utilização desses profissionais pode ser importante para a segurança de uma organização, porém deve-se tomar bastante cuidado com relação aos limites da utilização de seus serviços. Um *white hat* pode achar uma série de vulnerabilidades no sistema, e pode querer cobrar para fazer as correções necessárias. Como novas vulnerabilidades vão sendo descobertas com o tempo, e como novas funcionalidades que vão sendo implantadas na rede trazem consigo uma série de novas brechas, uma nova análise de segurança é sempre necessária, o que sempre acaba gerando mais custos. A segurança portanto é um processo constante, de modo que o mais interessante talvez seja manter um administrador de segurança dentro da própria organização. Essa pode ser a solução mais plausível, pois depois de uma consultoria, das simulações, das análises e das correções, é sempre necessário uma adequação da política de segurança, fazendo com que os custos com a utilização de um *white hat* sempre acabem sendo maiores do que os previstos, como se formassem uma grande bola de neve.

Diversas fontes, como a [RAD 99], mostram que utilizar um *hacker* para cuidar da segurança pode ser perigoso, justamente devido à própria cultura *hacker*. Um exemplo disso é sobre uma agência governamental americana, que contratou um *white hacker* para cuidar da segurança interna. Quando o *hacker* finalizou o serviço, a agência descobriu que ele tinha divulgado as vulnerabilidades encontradas em *sites* Web de *hackers* e *bulletin boards*. O pior é que muitas dessas vulnerabilidades não tinham sido sequer corrigidas.

4.1.6 Full Fledged

São os cyber-terroristas, *black hat* ou *crackers*. Esse grupo utiliza seus conhecimentos para invadir sistemas e roubar informações secretas das organizações. Geralmente tentam vender as informações roubadas de volta para a sua vítima, ameaçando a organização de divulgação das informações roubadas, caso o valor desejado não seja pago. Esse tipo de prática é conhecido como chantagem ou *blackmail*, e a exposição pública das informações poderia trazer conseqüências indesejáveis para a vítima.

O *black mail* foi utilizado recentemente no caso de invasão do *site* de comércio eletrônico CD Universe. Os *hackers* conseguiram invadir a base de dados do *site*, onde conseguiram capturar milhares de números de cartões de créditos de seus clientes. Eles exigiam milhões de dólares para que não divulgassem esses números.

Além do *blackmail*, qualquer ação maliciosa que visa prejudicar e causar prejuízos às suas vítimas, podem ser consideradas de autoria de *full fledged*.

4.2 Terminologias do Mundo *Hacker*

Algumas terminologias utilizadas no mundo *hacker* são [RAD 99]:

- ***Carding*** - prática ilegal de fraudes com números de cartões de créditos, que são utilizados pelos *hackers* para fazer compras para si próprios e para seus amigos. O comércio eletrônico tornou-se um terreno de grande perigo devido aos *cardings*, o que vem fazendo com que a segurança das transações eletrônicas com cartões de créditos tenha uma evolução natural, como é o caso do protocolo SET;
- ***Easter Egg*** - uma mensagem, imagem ou som que o programador esconde em um software, como brincadeira. Geralmente deve-se seguir procedimentos para ativar essa parte do código dos softwares;

- ***Media Whore*** - na cultura *hacker*, quem deixa o mundo "*underground*" para ganhar atenção da mídia são traidores. São aqueles que buscam a glória e a fama pessoal;
- ***Phreaking*** - é o *hacking* de sistemas telefônicos, geralmente para fazer ligações gratuitas ou para espionar ligações alheias;
- ***Suit*** - pela cultura dos *hackers*, *suit* são "os outros", ou seja, os funcionários de organizações, que trabalham sempre bem vestidos. Oficiais do governo são também chamados de *suit*;
- ***Tentacle*** - também conhecido como *aliases*, são as identidades usadas pelos *hackers* para executar suas "proezas" sem serem identificados;
- ***Trojan Horse*** - os cavalos-de-tróia, que são softwares legítimos que possuem códigos escondidos, que executam atividades não previstas. O usuário utiliza o software normalmente, mas ao mesmo tempo executa outras funções ilegais, como enviar mensagens e arquivos para o *hacker*;
- **Vírus** - programa que destrói dados ou sistemas de computador. Esses programas se replicam e são transferidos de um computador para outro;
- ***Worm*** - similar ao vírus, que se auto-replica, espalhando-se de uma rede para outra. Diferente do vírus, o *worm* pode causar danos sem a necessidade de ser ativado pelo usuário;
- ***War Dialer*** - programa que varre números telefônicos em busca de modens ou aparelhos de fax;
- ***Warez*** - softwares piratas que são distribuídos ilegalmente pela Internet;
- ***White Hat*** - também conhecido como *ethical hacker* ou *true hacker*, é o *hacker* que não é malicioso. Eles utilizam seus conhecimentos para satisfazerem suas curiosidades, e não para danificar computadores ou realizar outras atividades relacionadas aos crimes cibernéticos (seção 4.1.5).

4.3 Os Pontos Explorados

Duas técnicas principais são exploradas pelos *hackers* para a invasão de um sistema, que são a engenharia social e a invasão técnica. A engenharia social será melhor discutida na seção 4.5.1, enquanto as invasões técnicas serão discutidas nas seções a seguir. Essas invasões exploram deficiências na concepção, configuração ou gerenciamento dos sistemas, e continuarão a existir na

medida em que o mercado é centrado nas características dos produtos, e não na segurança. Esse comportamento adotado pelos fabricantes, de preferirem consertar furos de segurança a construir sistemas conceitualmente seguros, é discutido por Peter Shipley da KPMG Peat Marwick em [HAL 98].

Os ataques técnicos podem explorar uma série de condições, como as que podem ser vistas a seguir [HAL 98]:

- Exploração de vulnerabilidades, que podem ser *bugs* na implementação ou no *design* do sistema operacional, de serviços, aplicativos e protocolos. Ataques a senhas também são comuns, que podem ser através da captura através da rede (*packet sniffing*) ou através do *cracking*. Protocolos como o *Internet Control Message Protocol* (ICMP) podem ser explorados em ataques como o *Smurf* e *ping-of-death*. O UDP pode ser explorado pelo *Fraggle*, enquanto o TCP pode sofrer ataques conhecidos como *SYN flood*. Esses ataques serão discutidos com maiores detalhes nas seções a seguir;

- Mal uso de ferramentas legítimas, que em vez de serem utilizadas para auxiliar no gerenciamento e na administração, são utilizadas pelos *hackers* para a obtenção de informações ilícitas para a realização de ataques. Alguns exemplos são o comando *nbtstat* do Windows NT, que fornece informações que podem ser utilizadas para o início de um ataque contra usuários do sistema (identidade do controlador do domínio, nome NetBIOS, nome IIS, nomes de usuários), o *port scanning*, que é utilizado para identificar portas ativas do sistema, e conseqüentemente dos serviços providos por cada porta, e o *packet sniffing*, utilizado para diagnosticar problemas de rede, mas que pode ser utilizado para capturar pacotes que trafegam na rede em busca de informações como senhas, arquivos e *e-mails*;

- Configuração, administração e manutenção imprópria, onde a complexidade na definição de rotas e das regras de filtragem do *firewall* introduzem novos pontos de ataques no sistema. Outros pontos são a utilização da configuração padrão, a administração preguiçosa (sem senha), e a exploração da relação de confiança entre equipamentos;

- Projeto do sistema ou capacidade de detecção ineficiente, como por exemplo, um sistema de detecção de untrusão (IDS) que fornece informações falsas ou erradas.

Foi visto que a grande maioria dos *hackers* são novatos, que utilizam ferramentas e informações que já existem na Internet, e que a espionagem industrial cresce a cada dia. De fato, hoje não é necessário grandes conhecimentos para invadir um sistema, sendo possível até mesmo

adquirir CDs (www.hackerscatalog.com) com uma interface GUI de fácil utilização para a realização dos ataques. O sucesso de um ataque depende mais do número e variedade das tentativas de ataque, de forma que o nível de segurança da organização será tão grande quanto aos objetivos do invasor, ou seja, se um *hacker* tiver como objetivo atacar uma rede, ele terá sucesso mais rapidamente se a rede da organização não tiver um nível de segurança adequado.

O fato é que a maioria dos ataques constitui um briga de gato e rato, já que as ferramentas de defesa existentes só protegem os sistemas dos ataques já conhecidos. Isso faz com que, se por um lado os administradores de segurança procuram fechar as brechas existentes, por outro lado os *hackers* vêm atualizando constantemente o seu leque de técnicas de ataque, que podem não ser detectados pelos administradores e pelas suas ferramentas de defesa [RAN 01].

Assim, o que deve se ter em mente é que a segurança é um processo evolutivo, uma constante luta do administrador de segurança contra os *hackers* e contra os usuários internos que buscam maneiras de utilizar recursos proibidos na rede e que até mesmo causam transtornos através de seus erros.

4.4 O Início de um Ataque

O primeiro passo para um ataque é a obtenção de informações sobre o sistema a ser atacado. Essas informações podem ser obtidas através das seguintes técnicas [HTTP 01]:

- ***Dumpster Diving* ou *Trashing*** – atividade na qual o lixo é remexido em busca de informações sobre a organização ou a rede da vítima, como nomes de contas e senhas, informações pessoais e informações confidenciais. Essa técnica foi utilizada inclusive no Brasil, principalmente em bancos, onde os lixos eram remexidos em busca de informações. Elas eram assim trabalhadas e cruzadas com outras informações de clientes, resultando no acesso à conta desses usuários. Isso faz com que um picotador de papéis seja um acessório importante dentro das organizações, já que não se pode acreditar que ninguém irá ler algum papel jogado no lixo;
- ***Engenharia Social*** – técnica onde se tenta iludir a vítima assumindo-se uma falsa identidade, normalmente de administrador de rede, gerente de segurança, empregado de alto escalão ou outra pessoa requisitando informações pessoais. Essa técnica explora o fato

dos usuários estarem sempre dispostos a ajudar e a colaborar com os serviços da organização. Os aspectos da engenharia social serão discutidos com mais detalhes na seção 4.5.1;

- ***War Dialing*** – técnica de localização de modens, que podem ser utilizados para se descobrir o tipo de sistema e também como porta de entrada para a rede da organização. Maiores detalhes dessa técnica podem ser vistos na seção 4.9.5;
- **Ataque Físico** – método menos comum, onde equipamentos, softwares ou fitas magnéticas são roubados. O incidente mais conhecido é do Kevin Poulsen, que roubou diversos equipamentos do provedor de acesso de diversas organizações, o que resultou na descoberta de diversas informações confidenciais dessas organizações;
- ***Scanning* de Portas e de Vulnerabilidades** – técnica utilizada para capturar informações sobre as portas abertas nos sistemas, e também para determinar de maneira fácil e simples as vulnerabilidades existentes nos sistemas. Esses dois tipos de ferramentas serão discutidas, respectivamente, nas seções 4.5.4 e 4.5.3.

Após a obtenção das informações, o *hacker* pode atacar o sistema através de uma das três formas [HTTP 01]:

- Monitorando a rede;
- Penetrando no sistema;
- Inserindo códigos maliciosos ou informações falsas no sistema.

O resultado desses ataques podem ser de cinco tipos [HTTP 01]:

- Monitoramento não autorizado;
- Descoberta e revelação de informações confidenciais;
- Modificação não autorizada de servidores e da base de dados da organização;
- Negação ou corrupção de serviços;
- Fraude ou perdas financeiras.

4.5 Ataques para Obtenção de Informações

Conhecer o terreno e coletar informações do alvo, se possível sem ser notado ou descoberto, é o primeiro passo para a realização de um ataque de sucesso. É através da obtenção dessas informações que o ataque pode ser planejado e executado com sucesso. As seguintes técnicas e ferramentas, que serão discutidas nas próximas seções, podem ser utilizadas para a obtenção dessas informações: engenharia social, *packet sniffing*, *scanning* de vulnerabilidades, *port scanning* e *firewalking*.

4.5.1 Engenharia Social

A engenharia social é a técnica que explora as fraquezas humanas e sociais, ao invés de explorar a tecnologia. Ela tem como objetivo enganar e lubridiar pessoas, a fim de que elas revelem senhas ou outras informações que possam comprometer a segurança da organização. Um ataque de engenharia social clássico consiste em se passar por um alto funcionário, que tem problemas urgentes de acesso ao sistema. O *hacker* faz assim o papel de um ator que ataca o lado mais fraco de um esquema de segurança, o ser humano. Esse ataque é difícil de ser identificado, pois o que está em jogo é a confiança, a psicologia e a manipulação das pessoas. Kevin Mitnick, um dos *hackers* mais famosos e que se livrou da prisão em fevereiro de 2000, utilizava a engenharia social em mais de 80% de seus ataques.

Um caso de um ataque onde a engenharia social foi explorada ocorreu em outubro de 1998, envolvendo a *America On-Line* (AOL). Um indivíduo conseguiu obter dados da AOL e assim solicitar mudanças no registro de domínio DNS, de forma que todo o tráfego para a AOL foi desviado para um outro domínio [HTTP 02].

Uma das técnicas de engenharia social consiste em visitar escritórios e tentar fazer com que a secretária se distraia, enquanto o *hacker* analisa documentos que estão em cima da mesa ou no computador. Utilizar o método de entrar pela portas do fundo ou pela garagem para ter acesso a salas restritas também faz parte da engenharia social, bem como se disfarçar de entregador de flores ou de pizzas.

Outro ataque, que exige um prazo mais longo para o seu sucesso, é criar um software com *bugs* inseridos de propósito. O *hacker* poderia entregar esse software para a organização a fim de que testes fossem realizados com ele. O *hacker* pede gentilmente para que liguem para ele

em caso de falhas no software, se prontificando a resolver os problemas encontrados. A vítima assim ligaria para o *hacker*, que teria assim o acesso ao computador da vítima para a correção da falha que ele mesmo implantou, além do acesso para a realização das tarefas referentes ao ataque, tais como a instalação de *backdoors* ou de bombas lógicas.

4.5.2 Packet Sniffing

Também conhecida como *Passive Eavesdropping*, essa técnica consiste na captura de informações valiosas diretamente através do fluxo de dados na rede. Diversos softwares podem ser encontrados, inclusive o *snoop*, que vem no Solaris, e o *tcpdump*, que vem no Linux, que são geralmente utilizados para auxiliar na resolução de problemas de rede.

As informações que podem ser capturadas pelos *sniffers* são referentes aos pacotes que trafegam no segmento de rede em que o software funciona, sendo que diversos softwares possuem filtros que permitem a captura de pacotes específicos referentes a determinados endereços IPs, determinados serviços ou determinados conteúdos.

Senhas que trafegam em claro pela rede, como os de serviços como FTP, Telnet e POP, podem ser facilmente capturados desta maneira. *E-mails* também podem perder a sua confidencialidade através da utilização de *sniffers*.

As medidas de segurança que podem ser tomadas para minimizar as implicações de segurança são a divisão da rede em mais segmentos, através da utilização de *switches* ou de roteadores. Porém, como essa medida não elimina totalmente a possibilidade de captura de pacotes em um mesmo segmento, a solução é o uso de protocolos que utilizam a criptografia, como o SSH no lugar do Telnet, ou o IPSec, além da utilização da criptografia em informações confidenciais que trafegam pela rede, como por exemplo, em *e-mails*.

Existem diversas técnicas para verificar se um *sniffer* está sendo executado em um determinado segmento de rede. Um dos métodos é o administrador acessar cada equipamento dessa rede e verificar se existe ou não o processo rodando. O problema é que se um *hacker* estiver executando um *sniffer*, ele tomará o cuidado de esconder esse processo da lista de processos, impossibilitando assim a sua detecção. Outro método é a criação de tráfego de senhas pré-determinadas, de modo que o *hacker* pode ser detectado e identificado através da utilização dessa senha. Esse método porém não é muito eficiente, uma vez que o *hacker* pode fazer grandes estragos antes de utilizar essa senha pré-determinada, principalmente porque ele terá em seu

poder não apenas essa senha pré-determinada, mas também a de usuários legítimos. David Wu apresenta em [WU 98] 3 outras técnicas para realizar a detecção remota de *sniffers* na rede, sem a necessidade de acessar cada equipamento do segmento:

- *MAC Detection* – tira proveito de um erro na implementação do TCP/IP de diversos sistemas operacionais, que utiliza apenas o endereço IP para entregar os pacotes, não conferindo o endereço MAC quando a interface está no modo promíscuo. Assim, a técnica utiliza pacotes *ICMP echo request* com endereços IPs de um *host* mas com endereços MAC falsos. Se alguém estiver utilizando um *sniffer*, ele estará em modo promíscuo, e irá responder ao pedido de *ping*, sendo assim detectado. Essa técnica não funciona com sistemas operacionais que implementam o protocolo TCP/IP corretamente;

- *DNS Detection* – tira proveito do fato da maioria dos *sniffers* utilizarem o DNS reverso. Um tráfego com endereço falso seria colocado na rede, e se o *sniffer* capturar esse pacote, o pedido de DNS reverso seria enviado ao servidor DNS, que detectaria assim a existência de *sniffers* na rede. Essa técnica identifica quantos *sniffers* estão na rede, não sendo possível porém detectar quais são esses equipamentos. Essa técnica pode ainda detectar *sniffers* entre segmentos separados por roteadores;

- *Load Detection* – a idéia é que os equipamentos que estão rodando *sniffers* possuem maior grau de processamento, e assim levam mais tempo para responder às requisições. Essa técnica faz uma análise estatística dos tempos de respostas a requisições de serviços, baseadas nos tempos de respostas com pouco tráfego na rede e com o tráfego a ser pego pelos *sniffers*. Esses tempos são então comparados, de modo que se a diferença for muita, então o equipamento está utilizando maior processamento, que pode ser resultado da utilização de *sniffers*. O tipo de pacote a ser utilizado nos testes, porém, deve ser escolhido cuidadosamente. O *ICMP echo request*, por exemplo, não serve, pois a resposta sai do equipamento a partir da própria pilha TCP/IP, antes de chegar ao nível de usuário, não sendo possível portanto medir o grau de processamento do equipamento. A mesma situação ocorre com os pedidos de conexão SYN. Sendo assim, é necessário utilizar um método que utilize o nível de usuário, como é o caso de comandos FTP. Essa técnica não funciona de modo eficiente em redes com alto tráfego, pois as medidas são mais difíceis de serem apuradas e comparadas, já que os dois tempos tornam-se bastante equivalentes.

4.5.3 Scanning de Vulnerabilidades

Os *scanners* de vulnerabilidades realizam diversos tipos de testes na rede em busca de brechas de segurança, sejam elas em protocolos, serviços, aplicativos ou sistemas operacionais. Alguns riscos existentes na rede que esses *scanners* podem analisar, através da checagem de roteadores, servidores, *firewalls*, sistemas operacionais e outras entidades IP, são [INF 99-2]:

- Compartilhamento de arquivos que não são protegidos por senhas;
- Má configuração;
- Software desatualizado;
- Pacotes TCP que podem ter o seu número de seqüência adivinhados;
- *Buffer overflows* em servidores;
- Falhas no nível de rede do protocolo;
- Configurações de roteadores potencialmente perigosos;
- Evidências de higiene pobre dos servidores Web;
- Checagem de cavalos de tróia, como o Back Orifice ou o Netbus;
- Checagem de senhas fáceis de serem adivinhadas (*Password guessing*);
- *War dialing*;
- *Port scanning*;
- SNMP;
- *Denial-of-Service*;
- Sistemas de detecção de intrusões (*Intrusion Detection System*);
- Configuração da política dos navegadores *Web*.

Os riscos citados serão discutidos nas próximas seções, e demonstram que os *scanners* de vulnerabilidades são uma peça importante para análise de riscos e também para a auditoria da política de segurança das organizações. Essa importância pode ser enfatizada principalmente porque o *scanning* pode ser utilizado para demonstrar os problemas de segurança que existem na rede, de forma a alertar a gerência para a necessidade de um melhor planejamento com rela-

ção à proteção dos valores da organização. As consultorias de segurança utilizam constantemente essa ferramenta para justificar a necessidade de uma melhor proteção e assim vender seus serviços, aproveitando-se de uma importante funcionalidade dos *scanners*, que é a sua capacidade de *reporting*, que realiza uma avaliação dos riscos encontrados pelo *scanning* [INF 99-2].

Como o *scanner* é a implementação de um conjunto de vulnerabilidades que podem ser exploradas, é fundamental que a sua base de dados seja constantemente atualizada com as novas técnicas de ataques e as novas vulnerabilidades, para que a proteção da rede não seja prejudicada. A atualização da base de dados é similar à atualização de anti-vírus, ou seja, uma base mais antiga pode não detectar as brechas de segurança mais novas.

Uma pesquisa mostra os 12 maiores problemas identificados em um *scanning* da rede [INF 99-2]:

1. Servidores rodando serviços desnecessários, permitindo, por exemplo, o SNMP ou o FTP anônimo;

2. Software e firmware com configurações *default*, sem *patches*, desatualizados ou vulneráveis;

3. Vazamento de informações através de serviços como o SNMP, SMTP, *finger*, *rusers*, *systat*, *netstat*, *banners* Telnet, *Server Message Block* (SMB) do Windows, além da configuração de zonas de transferências para servidores sem nome;

4. Relações de confiança inapropriadas, para serviços como *rlogin*, *rsh*, *rexec*;

5. *Firewalls* ou listas de controle de acesso dos roteadores mal configurados;

6. Senhas fracas;

7. Servidores Web mal configurados, como *scripts* CGI, FTP anônimo e SMTP;

8. Exportação de serviços de compartilhamento de arquivos inapropriados, como o NetWare File Services ou o NetBIOS;

9. Servidores Windows NT mal configurados ou sem aplicação de *patches*;

10. *Logging*, monitoramento e capacidade de detecção inadequados;

11. Pontos de acesso remoto inseguros;

12. Falta de política, procedimentos, padrões ou guias compreensivos.

Um ponto importante a ser considerado é que, assim como os *scanners* auxiliam os administradores de segurança na proteção da rede, indicando as vulnerabilidades a serem enfrentadas, eles podem também ser utilizados pelos *hackers* para que brechas de segurança sejam detectadas e exploradas. Uma contra-medida que pode ser adotada é a utilização de sistemas de detecção de intrusões, *Intrusion Detection Systems* (IDS), que realizam o reconhecimento de padrões de *scanning* e alertam o administrador de segurança quanto ao fato. O IDS será discutido no capítulo 7.

4.5.4 Port Scanning

Os *port scans* são ferramentas utilizadas para a obtenção de informações referentes aos serviços que são acessíveis, que são definidas através do mapeamento das portas TCP e UDP. Com as informações obtidas através do *port scanning*, evita-se o desperdício de esforços com ataques a serviços inexistentes, de modo que o *hacker* pode se concentrar em utilizar técnicas que exploram os serviços específicos, que podem ser de fato exploradas.

O *nmap* é um dos *port scans* mais utilizados, e pode ser utilizado para realizar a auditoria do *firewall* e do IDS, além de determinar se o sistema possui falhas de implementação na pilha TCP/IP, que podem ser exploradas em ataques do tipo DoS. Além de mapear as portas abertas dos sistemas, ele pode ainda identificar, através do método de *stack fingerprinting*, que é discutido em [FYO 98], o sistema operacional utilizado pelo alvo. Existem também opções para informar sobre a seqüência dos pacotes TCP, sobre os usuários que estão rodando os serviços de cada porta, o nome DNS, e se o endereço pode se tornar vítima do *Smurf* (seção 4.6.4). Algumas características que tornam o *nmap* bastante poderoso são o *scanning* paralelo, a detecção do estado de *hosts* através de *pings* paralelos, o *decoy scanning*, a detecção de filtragem de portas, o *scanning* de RPC (não *portmapper*), o *scanning* através do uso de fragmentação de pacotes, e a flexibilidade na especificação de portas e alvos. Além disso, o *nmap* informa o estado de cada porta identificada como sendo aberta (aceita conexões), filtrada (existe um *firewall* que impede que o

nmap determine se a porta está aberta ou não) ou não filtradas. Os métodos de *scanning* utilizados pelo *nmap* são [FYO 97][FYO 99]:

- ***UDP*** – esse método envia um pacote UDP de 0 *byte* para cada porta do alvo. Se ele recebe uma mensagem *ICMP port unreachable*, então a porta está fechada. Caso contrário, o *nmap* assume a porta como estando aberta;

A -> T O atacante (A) envia um pacote UDP ao alvo (T);

A <- T Se T retorna uma mensagem *ICMP port unreachable*, a porta está fechada;

A ! T Se A não recebe nenhuma mensagem, a porta provavelmente está aberta, e pode ser utilizada para o ataque.

- ***TCP connect()*** – é a forma mais básica de *scanning* TCP. A *system call connect()* é utilizada para abrir uma conexão com as portas do alvo. Se a porta está aberta, a *system call* funcionará com sucesso, caso contrário a porta não está aberta. Uma vantagem desse método é que não é necessário nenhum privilégio especial para a sua utilização. Em contrapartida, esse método é facilmente detectado, pois basta apenas verificar as conexões em cada porta;

A -> T O atacante (A) tenta fazer uma conexão com o alvo (T);

A <- T Se T aceita a tentativa de conexão de A, então a porta está aberta, e pode ser utilizada para o ataque;

T ! A Se T não aceita a tentativa de conexão vinda de A, então a porta está fechada.

- ***TCP SYN (half open)*** – esse método não abre uma conexão TCP completa. Um pacote SYN é enviado como se ele fosse abrir uma conexão real. Caso um pacote SYN-ACK seja recebido, a porta está aberta, enquanto um RST como resposta indica que a porta está fechada. Caso o SYN-ACK seja recebido, o *nmap* envia o RST para fechar o pedido de conexão antes que ela seja efetivada. A vantagem dessa abordagem é que poucos irão detectar esse *scan* de portas. É necessário ter privilégio de super usuário para utilizar esse método;

A -> T O atacante (A) envia um pacote SYN ao alvo (T);

A <- T Se T retorna um pacote SYN-ACK, a porta está aberta;

A <- T Se T retorna um pacote RST para fechar o pedido de conexão de A, a porta está fechada;

- ***ICMP (ping sweep)*** – envia pacotes *ICMP echo request* para os *hosts*. Porém, como alguns *sites* bloqueiam esses pacotes, este método é bastante limitado. O *nmap* envia também um pacote TCP ACK para a porta 80. Se ele obtém um pacote RST de volta, a máquina está funcionando;

A -> T O atacante (A) tenta envia pacotes *ICMP echo request* para o alvo (T), junto com um pacote TCP ACK para a porta 80;

A <- T Se A recebe de T um pacote *ICMP echo reply*, então a porta está aberta. Caso A receba um pacote TCP RST, T está funcionando;

T ! A Se A não recebe nenhum pacote de volta, então T não está funcionando.

- ***FIN*** – modo *stealth*. Alguns *firewalls* são capazes de registrar a chegada de pacotes SYN em determinadas portas, detectando assim o método TCP SYN. O modo *stealth* elimina essa possibilidade de detecção. Portas fechadas enviam um pacote RST como resposta a pacotes FIN, enquanto portas abertas ignoram esses pacotes. Esse método não funciona sobre Windows 9x/NT, pois a Microsoft não seguiu o rfc 973;

A -> T O atacante (A) envia um pacote FIN para o alvo (T);

A <- T Se A recebe um pacote RST de T, então a porta está fechada;

T ! A Se A não recebe nenhum pacote de resposta de T, então a porta está provavelmente aberta, podendo ser portanto explorada.

- ***Xmas Tree*** – modo *stealth*. Portas fechadas enviam um pacote RST como resposta a pacotes FIN, enquanto portas abertas ignoram esses pacotes. Os *flags* FIN, URG e PUSH são utilizados. Esse método não funciona sobre Windows 9x/NT, pois a Microsoft não seguiu o rfc 973;

A -> T O atacante (A) envia um pacote FIN com os *flags* FIN, URG e PUSH ligados para o alvo (T);

A <- T Se A recebe um pacote RST de T, então a porta está fechada;

T ! A Se A não recebe nenhum pacote de resposta de T, então a porta está provavelmente aberta, podendo ser portanto explorada.

- ***Null scan*** – modo *stealth*. Portas fechadas enviam um pacote RST como resposta a pacotes FIN, enquanto portas abertas ignoram esses pacotes. Nenhum *flag* é ligado. Esse método não funciona sobre Windows 9x/NT, pois a Microsoft não seguiu o rfc 973;

A -> T O atacante (A) envia um pacote FIN sem nenhum *flag* ligado para o alvo (T);

A <- T Se A recebe um pacote RST de T, então a porta está fechada;

T ! A Se A não recebe nenhum pacote de resposta de T, então a porta está provavelmente aberta, podendo ser portanto explorada.

- ***RPC scan*** – combina vários métodos de *port scanning*. Ele pega todas as portas TCP e UDP abertas encontradas, e envia comandos NULL SunRPC na tentativa de eles serem portas RPC. É como se o comando 'rpcinfo –p' estivesse sendo utilizado, mesmo se um *firewall* estiver sendo utilizado ou se estiver protegido pelo TCP wrapper. O modo *decoy* não irá funcionar nesse método de *scanning*;
- ***ACK sweep***;
- ***SYN sweep***;
- ***FTP proxy (bounce attack)*** – o protocolo FTP permite que um servidor seja utilizado como um *proxy* entre o cliente e qualquer outro endereço. O ataque FTP Bounce é utilizado geralmente para enviar *e-mails* e mensagens, desviar de *firewalls* ou encher servidores com arquivos inúteis. O *nmap* utiliza essa característica para realizar o *scanning* TCP a partir desse servidor FTP. Caso o servidor FTP tenha permissão de leitura e escrita, é possível enviar dados para as portas abertas encontradas pelo *nmap*;
- ***Reverse-ident*** – se o *host* estiver utilizando o *identd*, então é possível identificar o dono dos serviços que estão com as portas abertas.

Os métodos utilizados pelo *nmap* para a detecção do sistema operacional [FYO 99] são listados a seguir, e podem ser vistos com detalhes em [FYO 98]:

- *TCP/IP fingerprinting;*
- *Stealth scanning;*
- *Dynamic delay;*
- *Retransmission calculations;*

Para uma organização se proteger contra a ação desses *scanners*, os *Intrusion Detection Systems* (IDS), que serão discutidos no capítulo 7, podem ser utilizados. Esse tipo de sistema faz o reconhecimento de padrões de *scanning*, de forma a alertar o administrador de segurança contra

tentativas de mapeamento da rede da organização. Porém, como pode ser visto em [ARK 99], diversas técnicas de *scanning* são utilizadas para driblar os IDS:

- *Random Port Scan* – dificulta o IDS no reconhecimento do *scanning*, ao não realizar a varredura seqüencialmente;
- *Slow Scan* – dificulta a detecção ao utilizar um *detection threshold*, que é o número de pacotes que pode ser identificado por um IDS, menor. Assim, o atacante pode, por exemplo, enviar apenas 2 pacotes por dia ao alvo para que o *scanning* seja realizado sem que ele seja detectado.
- *Fragmentation Scanning* – a fragmentação de pacotes pode dificultar a detecção de uma varredura, porém a maioria dos IDS já solucionou esse problema;
- *Decoy* – utiliza uma série de endereços falsificados, de modo que o IDS pensa que o *scanning* está partindo desses vários *hosts*, sendo praticamente impossível identificar a origem real da varredura. Um método que era utilizado para a identificação de um endereço *decoy* era verificar o campo *Time to Live* (TTL) dos pacotes. Se eles seguissem um padrão já determinado, então esse endereço poderia ser considerado *decoy*. O *nmap* utiliza um valor de TTL aleatório entre 51 e 65, dificultando assim a sua detecção;
- *Coordinated Scans* – dificulta a detecção ao utilizar diversas origens de varreduras, cada uma em determinadas portas. É geralmente utilizada por um grupo de atacantes.

Além de cumprir com o papel a que se destina, um *port scanning* pode trazer uma série de consequências para os seus alvos, a maioria deles relacionada com a implementação incorreta da pilha TCP/IP [SEC 98-3]:

- O IOS da Cisco trava quando o *UDP Scanning* é utilizado, quando a porta de *syslog* do roteador (UDP 514) é testada;
- O Check Point Firewall-1 é incapaz de registrar o *FIN Scan*;
- O *inetd* é desabilitado em alguns sistemas operacionais, entre eles o Solaris 2.6, Linux, HP-UX, AIX, SCO e FreeBSD, quando o método de *scanning* TCP SYN é utilizado;
- O TCP SYN *scanning* faz com que o "*blue screen of death*" seja mostrado no Windows 98;
- Afeta o *RPC portmapper* em alguns sistemas.

4.5.5 Firewalking

O *firewalking* é uma técnica implementada em um analisador de pacotes similar ao *traceroute*, e pode ser utilizado para se obter informações sobre uma rede remota protegida por um *firewall*. Essa técnica permite que pacotes passem por portas em um *gateway*, além de determinar se um pacote com várias informações de controle pode passar pelo *gateway*. É possível ainda mapear roteadores encontrados antes do *firewall* [GOL 98].

O *firewalking* utiliza características do *traceroute* para obter informações sobre as regras de filtragem dos *firewalls*, e também para criar um mapa da topologia da rede. Com algumas opções do próprio *traceroute* é possível obter essas informações. Por exemplo, se um *firewall* permite somente o tráfego de pacotes ICMP (o *traceroute* utiliza o UDP normalmente), basta utilizar a opção –I para que as informações passem pelo *firewall*. O *traceroute* permite também que o *trace* seja realizado através de uma porta específica, o que pode ser utilizado em redes onde o *firewall* só permite o tráfego de pacotes DNS, por exemplo [GOL 98].

Uma medida de proteção contra o *firewalking* é a proibição de tráfego de pacotes ICMP (os usuários da rede passam a não poder utilizar serviços ICMP, impedindo assim o diagnóstico de problemas da rede), a utilização de servidores *proxy* ou a utilização do *Network Address Translation* (NAT) [GOL 98].

4.5.6 IP Spoofing

O *IP spoofing* é um ataque na qual o endereço real do atacante é mascarado, de forma a evitar que ele seja encontrado. Essa técnica é bastante utilizada para tentativas de acessos a sistemas onde a autenticação é baseada em endereços IPs, como a rede de confiança em uma rede interna. Uma organização pode proteger a sua rede contra o *IP spoofing* de endereços IPs da rede interna através da aplicação de filtros de acordo com as interfaces de rede.

Essa técnica é utilizada também em ataques do tipo DoS, onde pacotes de resposta não são necessários. O *IP spoofing* não permite que as respostas sejam obtidas, pois esses pacotes são direcionados para o endereço IP forjado, e não para o endereço real do atacante. Para que um ataque tenha a sua origem mascarada e os pacotes de resposta possam ser obtidos pelo atacante, será necessário aplicar outras técnicas em conjunto, como ataques DoS ao endereço IP da vítima forjada e mudanças nas rotas dos pacotes.

4.6 Ataques de Negação de Serviços

Os ataques de negação de serviços (*Denial-of-Service Attack* – DoS) fazem com que recursos sejam explorados de maneira agressiva, de modo que usuários legítimos ficam impossibilitados de utilizar esses recursos. Uma técnica típica é o *SYN flooding* (seção 4.6.2), que causa o *overflow* da pilha de memória através do envio de um grande número de pedidos de conexões, que não podem ser totalmente completados e manipulados. Outra técnica é o envio de pacotes específicos que causam o queda do serviço, que pode ser exemplificada pelo *Smurf* (seção 4.6.4). As seções a seguir mostram como o DoS pode ser explorado pelos atacantes.

4.6.1 Bugs em Serviços, Aplicativos e Sistemas Operacionais

Um dos maiores responsáveis pelos ataques de negação de serviços são os próprios desenvolvedores de softwares. Diversas falhas nas implementações e na concepção de serviços, aplicativos, protocolos e sistemas operacionais abrem brechas que podem ser exploradas em ataques contra a rede da organização. Esses tipos de falhas podem também serem utilizados para que códigos maliciosos sejam executados, o que pode resultar em acessos não autorizados aos recursos.

O grande número de *bugs* que podem ser explorados faz com que empresas como a Microsoft se comporte de uma maneira mais responsável quanto à rapidez na disponibilização de *patches* de segurança contra os *bugs* em seus softwares. Porém, o fato é que os *bugs* aparecem com maior rapidez e em maior número do que a possibilidade das empresas de corrigi-los e fixá-los [DID 98-2].

Alguns *bugs* e condições encontrados em softwares que podem ser explorados são:

- *Buffer Overflows* (seção 4.9.1);
- Condições inesperadas - manipulação de entradas através de diferentes camadas de códigos, como por exemplo, um comando Perl que recebe parâmetros através da Web e faz o sistema operacional executar comandos específicos;
- Entradas não manipuladas - código que não define o que fazer com entradas não válidas e estranhas;
- *Race Conditions* - quando mais de um processo tenta acessar os mesmos dados ao mesmo tempo, podendo causar assim confusões e inconsistências nas informações.

Um exemplo recente de *bug* pode ser visto em [BAR 99], que mostra a descoberta de uma falha conceitual no Unix, o que os torna vulneráveis. Essa falha, que atinge todos os sabores de Unix, inclusive o Linux, com exceção do BSDI, ocorre quando diversas conexões são feitas, porém sem pedidos de requisição. Assim, os diversos *daemons* não podem responder às conexões, e a tabela de processos do sistema, que pode trabalhar com um número entre 600 e 1500 processos simultâneos, causa a parada do servidor.

Um outro exemplo de *bug* pode ser visto em [LOP 99], que demonstra um ataque que explora o *cache* do mapeamento dos objetos utilizados nas *Dynamic Link Libraries* (DLLs) pelo Windows NT. Esses objetos do *cache* se localizam no espaço de nomes interno do sistema, e são criados com permissões para que o grupo *Everyone* possa controlar totalmente esses objetos. Com isso é possível substituir esses objetos. Quando um processo é criado, e a DLL está no *cache*, ele é simplesmente mapeado no espaço do processo em vez de ser carregado. Assim, é possível que um usuário com baixos privilégios substitua esse objeto do *cache* e ele seja utilizado por um processo com altos privilégios que executa o código contido nesse DLL de tróia. Os passos e os reparos para se evitar essa vulnerabilidade são descritos no artigo.

O perigo das vulnerabilidades padrões dos sistemas operacionais também deve ser considerado, como a que pode ser encontrada no Solaris (*fingerd* permite '*bouncing*' das consultas), no Windows NT (sistema de *hashing* das senhas extremamente fraca) e no IRIX (riscos de segurança em abundância através das configurações iniciais, como a existência de contas de usuários *default*) [FIST 99].

4.6.2 SYN Flooding

A característica dos ataques *SYN flooding* é que um grande número de requisições de conexão (pacotes SYN) são enviados, de tal maneira que o servidor não é capaz de responder a todas essas requisições. A pilha de memória sofre um *overflow*, e as requisições de conexões de usuários legítimos são desprezadas.

Os ataques *SYN flooding* podem ser evitados comparando-se as taxas de requisições de novas conexões e o número de conexões em aberto. Mensagens de alertas e ações pré-configuradas podem ser tomadas quando a taxa chega a um padrão determinado. A injeção de pacotes específicos que podem fazer com que o serviço caia pode ser evitado monitorando-se os números

de seqüências dos pacotes, que estão em uma faixa esperada caso venha de um atacante específico [CIS 98-2].

Alguns dos métodos que podem ser utilizados contra os ataques *SYN flooding* são: em conexões de baixa velocidade (até 128 Kbps), utiliza-se um *time-out* e uma taxa máxima de conexões semi-abertas. Os pacotes então são descartados de acordo com esses valores determinados. Em conexões de maior velocidade, a melhor solução é desabilitar ou bloquear temporariamente todos os pacotes *SYN* ao *host* atacado após uma determinada taxa de conexão. Isso mantém o resto do sistema em funcionamento, ao mesmo tempo em que desabilita novas conexões ao *host* que está sendo atacado [CIS 98-2].

Outras soluções contra o *SYN flooding* podem ser vistas em [CIS 96], que são o aumento do tamanho da fila de pedido de conexões, que na realidade não elimina o problema, e a diminuição do *time-out* do *three-way handshake*, que também não elimina, porém minimiza o problema.

4.6.3 Fragmentação de pacotes IP

A fragmentação de pacotes está relacionada ao *Maximum Transfer Unit* (MTU), que especifica a quantidade máxima de dados em um pacote que podem passar através de um meio físico da rede. Por exemplo, a rede Ethernet limita a transferência a 1500 octetos de dados, enquanto o FDDI permite 4470 octetos de dados por pacote. Em um ambiente como a Internet, onde existe uma grande variedade física de redes, definir um MTU pequeno resulta em ineficiência, já que esses pacotes podem passar por uma rede que pode transferir pacotes maiores. Já definir um MTU grande, maior do que o da rede com MTU mínimo, resulta em fragmentação desses pacotes, já que os dados desse pacote não cabem nos pacotes que trafegam por essa rede com MTU mínimo. Os fragmentos resultantes trafegam pela rede, e quando chegam ao seu destino final eles são reagrupados, baseados em *offsets*, resultando assim no pacote original. Todo esse processo de fragmentação e reagrupamento (desfragmentação) é feito de modo automático e transparente para o usuário, definidos no protocolo IP. O fato do reagrupamento ocorrer somente no destino final implica em uma série de desvantagens, como a ineficiência, já que algumas redes físicas podem possuir MTU maior do que os pacotes fragmentados, passando a transmitir assim pacotes menores do que os possíveis. Uma outra desvantagem é a perda de pacotes, já que, se um fragmento for perdido, todo o pacote também será perdido [COM 95]. Uma desvantagem

ainda maior é a possibilidade de se tirar proveito dessa característica para a realização de ataques.

A possibilidade de ataques com fragmentação de pacotes IP ocorre devido ao modo como essa fragmentação e reagrupamento são implementados. Tipicamente, os sistemas não tentam processar o pacote, até que todos os fragmentos sejam recebidos e reagrupados. Isso abre a possibilidade de *buffer overflow* na pilha TCP, quando há o reagrupamento de pacotes maiores do que os permitidos. Isso causa problemas como o travamento do sistema, caracterizando assim ataques do tipo *Denial-of-Service*. Esse problema foi verificado inicialmente no fim de 1996, através do envio de pacotes ICMP *Echo Request*, o *ping*. Chamado de *Ping o'Death*, diversos sistemas travavam quando recebiam um pacote *ping* de tamanho grande (maior que 65535 *bytes*), devido ao estouro do *buffer* da pilha TCP/IP, já que não era possível reagrupar um pacote tão grande [KEN 97]. A única solução para o *Ping o'Death* é a instalação de *patches* que impedem que o *kernel* tenha problemas com *overflows* no momento do reagrupamento dos fragmentos IP. O *ping* foi inicialmente utilizado, devido à sua facilidade de utilização, porém outros pacotes IP grandes, sejam elas TCP (*Teardrop*) ou UDP, podem causar esse tipo de problema. Atualmente os sistemas já corrigiram esse problema através de atualizações e instalações de *patches*.

A característica do reagrupamento ser possível somente no *host* destino, de acordo com a especificação IP, faz com que o *firewall* ou o roteador não realize a desfragmentação, o que pode causar problemas peculiares. Um atacante pode, por exemplo, criar um pacote como sendo o primeiro fragmento, e especificar uma porta que é permitida pelo *firewall*, como a porta 80. O *firewall* assim permite a passagem desse pacote e dos fragmentos seguintes para o *host* a ser atacado. Um desses pacotes subseqüentes pode possui o valor *offset* capaz de sobrescrever a parte inicial do pacote IP que especifica a porta TCP. O atacante assim modifica a porta IP inicial de 80 para 23, por exemplo, de modo a conseguir acesso Telnet ao *host* a ser atacado [COH 99].

Os ataques de fragmentação do IP não podem ser evitados através de filtros de pacotes. *Hosts* que utilizam NAT estático também estão vulneráveis a esses ataques, além dos *hosts* que utilizam NAT dinâmico e que possuem uma comunicação ativa com a Internet [CIS 98].

A fragmentação IP é também utilizado como um método de *scanning*, como o utilizado pelo *nmap*. O *nmap* envia pacotes de *scanning* fragmentados, de modo que torna a sua detecção pelo *firewall* ou pelo IDS mais difícil.

4.6.4 Smurf e Fraggle

Huegen analisa em [HUE 98] os ataques *Smurf* e *Fraggle*. O *Smurf* é um ataque no nível de rede, onde um grande tráfego de pacotes *ping* (*ICMP echo*) é enviado para o endereço IP de *broadcast* da rede, com o endereço IP de origem da vítima falsificado. Assim, com o *broadcast*, cada *host* da rede recebe a requisição de *ICMP echo*, passando todos a responder para o endereço de origem, que é falsificado. A rede sofre, pois todos os seus *hosts* respondem à requisição ICMP (amplificador). E a vítima, que teve o seu endereço IP falsificado, recebe os pacotes de todos esses *hosts*. O *Fraggle* é o primo do *Smurf*, que utiliza pacotes *UDP echo* ao invés de pacotes *ICMP echo*.

Para evitar ser o intermediário do ataque ou o amplificador, o roteador deve ser configurado de modo a não receber ou deixar passar pacotes *broadcasts* através de suas interfaces de rede. Essa medida, porém, elimina também a possibilidade de utilizar o *ICMP echo* para o endereço de *broadcast* da rede, que é uma ferramenta útil para o diagnóstico da rede.

Os *hosts* também podem ser configurados de modo a não responderem a pacotes *ICMP echo* para o endereço de *broadcast*. No caso do ataque *Fraggle*, os pacotes *echo* e *chargen* devem ser descartados. Essas medidas também acabam impedindo o diagnóstico da rede, como o que ocorre na medida anterior.

Alguns equipamentos, como roteadores da Cisco, possuem o *Committed Access Rate* (CAR), que pode limitar o tráfego de determinados pacotes a uma banda determinada, o que permite limitar o tráfego de pacotes *ICMP echo* e *echo-replay* para uma banda limitada, que não comprometa a rede. O CAR pode impedir também o ataque de *TCP SYN Flooding* [HUE 98].

[FER 98] mostra um método para impedir ataques DoS que utilizam endereços IPs falsos. O objetivo é impedir que provedores de acessos ou organizações sejam utilizados como pontes de ataques, ou mesmo impedir que seus usuários realizem ataques externos. Esse método evita ataques *IP spoofing* desde a sua origem, e realmente é uma medida importante, pois é de responsabilidade do administrador de redes impedir que a sua rede seja envolvida em um ataque. O método permite que somente pacotes com endereço de origem da rede interna sejam enviadas para a rede externa, impedindo assim que pacotes com endereços falsos passem pela rede.

4.6.5 Teardrop e Land

O *Teardrop* é uma ferramenta utilizada para explorar os problemas de fragmentação IP nas implementações do TCP/IP, como foi visto na seção 4.6.3.

O *Land* é uma ferramenta utilizada para explorar vulnerabilidades TCP/IP onde um pacote é construído de modo que o pacote SYN possua o endereço de origem e a porta iguais ao do destino, ou seja, é utilizado o *spoofing*. A solução é criar regras de filtragem para se evitar o *IP spoofing* de endereços internos da rede [CIA 98-19].

4.7 Ataque Ativo contra o TCP

Joncheray mostra em [JON 95] um ataque ativo que utiliza o redirecionamento de conexões TCP para uma determinada máquina, caracterizando um ataque *man-in-the-middle*, conhecido também como *session hijacking*, ou seqüestro de conexões. Esse tipo de ataque permite driblar proteções geradas por protocolos de autenticação como o SKEY (*one-time password*) ou o Kerberos (identificação através de *tickets*). Um ataque ativo pode comprometer a segurança desses protocolos, já que os dados não trafegam de modo cifrado, nem assinados digitalmente. Ataques ativos são considerados difíceis de serem realizados, porém Joncheray mostra que com os mesmos recursos de um ataque passivo (*sniffers*), é possível realizar um ataque dessa natureza.

Uma conexão TCP entre dois pontos é realizada de modo *full duplex*, sendo definida através de 4 informações: endereço IP do cliente, porta TCP do cliente, endereço IP do servidor e porta TCP do servidor. Todo *byte* que é enviado por um *host* é marcado com um número de seqüência de 32 bits, que é reconhecido (*acknowledgment*) pelo receptor utilizando esse número de seqüência. O número de seqüência do primeiro *byte* é computado durante a abertura da conexão, e muda para cada conexão, de acordo com regras designadas a evitar a sua reutilização em múltiplas conexões.

O ataque se baseia na exploração do estado de desincronização nos dois lados da conexão TCP, que não podem trocar dados entre si, já que, embora ambos os *hosts* mantenham uma conexão estabelecida, os pacotes não são aceitos devido ao número de seqüência inválidos. Um terceiro *host*, do invasor, então cria os pacotes com números de seqüência válidos, se colocando entre os dois *hosts*. Ele envia os pacotes válidos para os dois *hosts*, caracterizando assim um ataque *man-in-the middle*.

O problema desse ataque é a grande quantidade de pacotes TCP ACK (*ACK Storm*), já que quando o *host* recebe uma pacote inválido, o número de seqüência esperado é enviado para o outro *host*, que por sua vez é inválido, então envia um novo pacote com o número de seqüência esperado, que por sua vez é inválido para o *host*. Isso cria um suposto *loop* infinito de pacotes ACK. Porém os pacotes que não carregam dados não são retransmitidos se o pacote é perdido. Isso significa que, se um dos pacotes no *loop* é negado, então o *loop* termina. A negação de um pacote é feita pelo IP, que possui uma taxa de pacotes não nulos, fazendo com que os *loops* terminem. Quanto mais congestionado a rede, maior o número de *loops* que terminam.

Dois métodos de desincronização de conexões TCP são apresentados por Joncheray: o *early desynchronization* (quebra da conexão em um estágio inicial no lado servidor e criação de uma nova conexão com número de seqüência diferente) e o *null data desynchronization* (envio de uma grande quantidade de dados para o servidor e para o cliente, que não devem afetar nem serem visíveis pelo cliente e pelo servidor).

4.8 Ataques Coordenados

A evolução mais evidente com relação aos ataques são os ataques coordenados. Esse tipo de ataque faz com que diversos *hosts* distribuídos sejam coordenados pelo *hacker* para a realização de ataques simultâneos aos alvos. Isso resulta em um ataque extremamente eficiente, onde a vítima fica praticamente sem defesa, sem conseguir ao menos descobrir a origem dos ataques, já que o ataque vem a partir de *hosts* intermediários, que são controlados pelo *hacker*.

Esse tipo de ataque possui 4 níveis hierárquicos, conforme a figura 4.1.

O *hacker* define alguns sistemas *master*, que se comunicam com os *daemons*, que realizam os ataques à vítima. Pode-se observar que os *masters* e os *daemons* são ambos vítimas do *hacker*, que através da exploração de vulnerabilidades conhecidas instalam os processos que serão utilizados no ataque. Apesar de ser uma tecnologia nova, as ferramentas de ataques coordenados possuem tamanha sofisticação, que se aproveitam das melhores tecnologias de ataques existentes, como a utilização de criptografia para o tráfego de controle entre o *hacker*, *masters* e *daemons*, e também para as informações que ficam armazenadas nesses *hosts*, como a lista dos *daemons*. Os *scannings* para a detecção dos *hosts* vulneráveis também são realizados de modo distribuído, e a instalação dos processos é feita de maneira automática, inclusive com uma implementação que faz com que esse processo esteja sempre em execução, mesmo que ele seja remo-

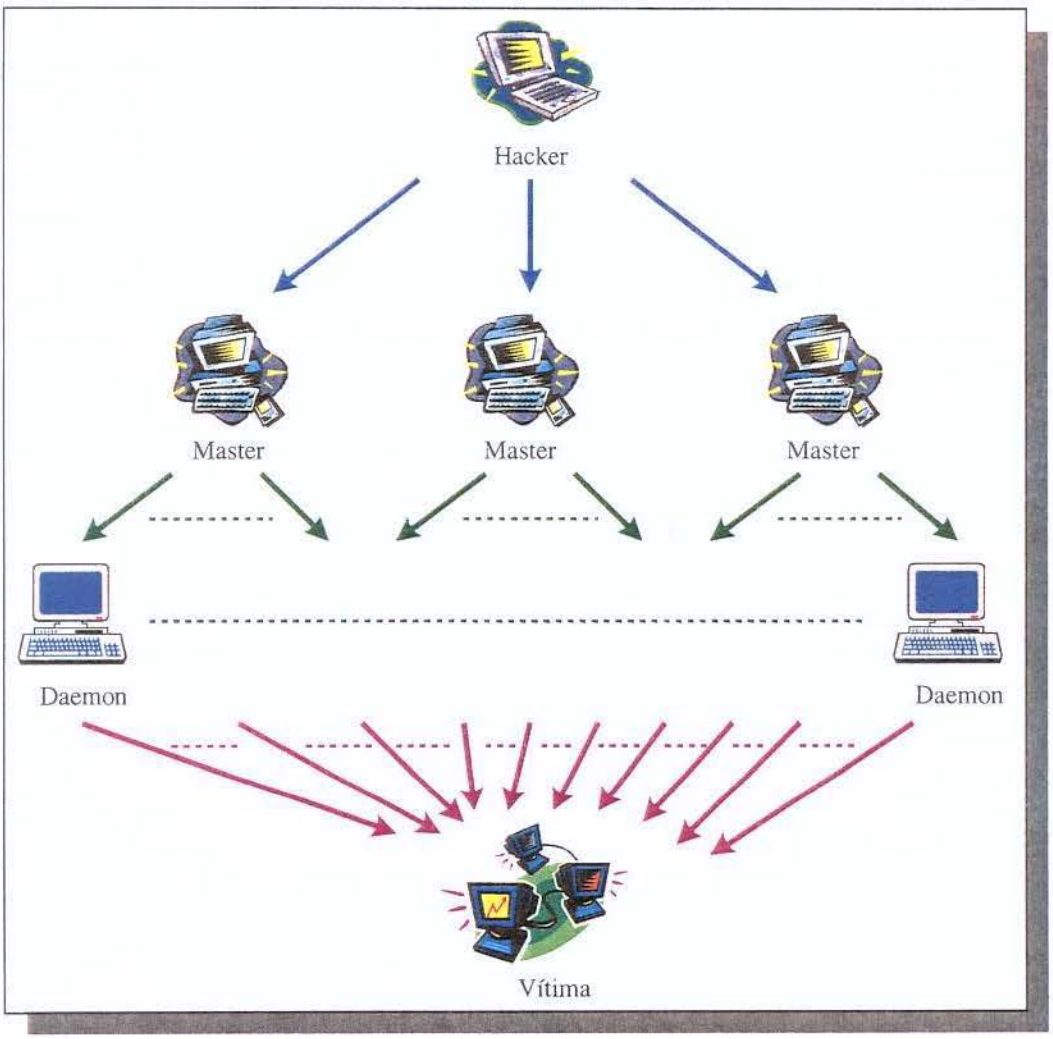


Figura 4.1: As partes envolvidas em um ataque coordenado.

vido ou o sistema seja reinicializado. Métodos para esconder as evidências das instalações dos *daemons* também são utilizados.

O primeiro ataque coordenado por um governo foi noticiado pela BBC News [NUT 99]. Aparentemente, o governo da Indonésia atacou o domínio do Timor Leste, devido a motivos políticos. Isso demonstra um novo estilo de guerra, onde táticas envolvendo computadores fazem parte da política oficial do governo, sendo utilizadas como uma arma em potencial para a desestabilização das atividades do governo.

As novas ferramentas para a realização de ataques distribuídos mostram que essa nova tecnologia, que está sendo desenvolvida a partir de tecnologias já existentes, está atingindo a maturidade, de forma que através de um simples comando um atacante pode fazer com que o alvo seja atacado a partir de uma série de pontos da Internet. Algumas das ferramentas utilizadas para esse tipo de ataque são o *trinoo* (trin00), o *Tribe Flood Network* (TFN), o *Stacheldraht* e o *Tribe FloodNet 2K* (TFN2K). De fato, essa maturidade pôde ser observada nos recentes ataques envolvendo grandes *sites* da Internet, como os da Amazon, Yahoo, UOL, Zipmail e do Cadê, que resultaram em grande repercussão na imprensa.

O *trinoo* é uma ferramenta utilizada para ataques coordenados de DoS que utiliza o UDP. Ele consiste de um pequeno número de servidores (*master*) e um grande número de clientes (*daemons*). No ataque, o *hacker* se conecta ao *master* e o instrui para realizar o ataque nos endereços IPs determinados. O *master* então se comunica com os *daemons*, passando-lhes instruções de ataques em determinados IPs em determinados períodos de tempo. O *trinoo* não utiliza o *IP spoofing*, e todas as comunicações com o *master* requerem uma senha [CER 99-1]. A rede *trinoo*, com pelo menos 227 sistemas, entre as quais 114 na Internet2, foi utilizada no dia 17 de agosto de 1999 para atacar a Universidade de Minnessota, tornando-a inacessível por 2 dias. A análise detalhada do *trinoo* pode ser vista em [DIT 99-01], que traz informações sobre os algoritmos utilizados, os pontos falhos e os métodos de detecção através de assinaturas, a serem implementados em IDS.

O TFN realiza ataques coordenados de DoS através de pacotes TCP, tendo a capacidade de realizar também o *IP spoofing*, *TCP SYN Flooding*, *ICMP echo request flood* e *ICMP directed broadcast* (*smurf*). O ataque ocorre quando o *hacker* instrui o cliente (*master*) para enviar instruções de ataques para uma lista de servidores TFN (*daemons*). Os *daemons* então geram o tipo de ataque DoS contra os alvos. As origens dos pacotes podem ser alteradas de modo aleatório, e os pacotes também podem ser modificados [CER 99-1]. Uma análise detalhada da ferramenta, de seu funcionamento e da assinatura que permite a sua detecção podem ser vistas em [DIT 99-02].

O *Stacheldraht* é uma outra ferramenta para ataques distribuídos, que combina características do *trinoo* e do TFN, adicionando a comunicação cifrada entre o atacante e os *masters Stacheldraht*, além de acrescentar a atualização automática dos agentes. A ferramenta é composta pelo *master* (*handler*) e pelo *daemon* ou *bcast* (*agent*). Uma análise detalhada da ferramenta pode ser encontrada em [DIT 99-03].

O TFN2K é uma evolução do TFN, que inclui características como técnicas que fazem com que o tráfego do TFN2K seja difícil de ser reconhecido ou filtrado, através da utilização de múltiplos protocolos de transportes (UDP, TCP e ICMP). O TFN2K possui ainda a possibilidade de executar comandos remotos, esconder a origem real do tráfego, e confundir as tentativas de encontrar outros pontos da rede TFN2K, através de pacotes *decoy*. Além disso, o TFN2K inclui ataques que causam o travamento ou a instabilidade nos sistemas, através do envio de pacotes mal formados ou inválidos, como os utilizados pelo *Teardrop* e pelo *Land* [CER 99-2].

Um dos sistemas que sofrem abusos em ataques coordenados é o MacOS 9, que pode ser utilizado como um amplificador de tráfego, ou seja, ele contém uma característica que permite que um tráfego seja amplificado em um fator de aproximadamente 37.5, sem a necessidade de utilizar o endereço de *broadcast*, como é o utilizado pelo *Smurf*. Os detalhes do problema com o MacOS 9 são analisados em [COP 99].

A prevenção contra os ataques coordenados é difícil, pois as ferramentas geralmente são instaladas em redes já comprometidas, resultando em um fator de escalabilidade bastante grande. Os ataques realizados mostram que os problemas são pertinentes da própria Internet, ou seja, uma rede pode ser vítima da própria insegurança da Internet. Uma das maneiras de contribuir para a diminuição desses incidentes são a prevenção contra instalações não-autorizadas das ferramentas de ataques coordenados, a prevenção dentro das organizações para que pacotes com *IP spoofing* não saiam dos limites da organização, e o monitoramento da rede à procura de assinaturas das ferramentas através de IDS.

A onda de ataques distribuídos está trazendo uma mudança na concepção de segurança, ao mostrar claramente que a segurança de uma organização depende da segurança dos outros, que podem ser atacados para servirem de base para novos ataques. Garantir que a rede da organização não seja utilizada como um ponto de ataque passa a ser essencial para minimizar esse tipo de ataque coordenado. A CERT [CER 99-3] apresenta uma série de medidas que devem ser tomadas de imediato, a curto prazo e a longo prazo pelos gerentes, administradores de sistemas, provedores Internet e centros de resposta a incidentes (*incident response teams*), a fim de evitar maiores estragos no futuro.

4.9 Ataques no Nível de Aplicação

Esse tipo de ataque explora vulnerabilidades em aplicativos, servidores e protocolos do nível de aplicação, e serão vistos nas seções a seguir. Os tipos de ataques mais comuns são aqueles que exploram o *buffer overflow*, comuns em aplicativos que realizam a interação do usuário com o sistema. Ataques *Common Gateway Interface* (CGI) utilizados pela Web também são um caso típico, como poderá ser visto na seção 4.9.2.

Além disso, protocolos como o FTP podem ser explorados em ataques como o *FTP Bounce*, assim como o SMNP. Serviços também podem ser explorados, como acontece com o *sendmail*, que através da utilização de comandos não documentados permite que o *hacker* obtenha acesso privilegiado ao sistema.

Outro tipo de ataque no nível de aplicação são os vírus e cavalos-de-tróia, que são a ameaça mais comum e mais visível diante os olhos dos gerentes, que por isso geralmente recebem a atenção necessária. Eles serão analisados na seção 4.9.4.

4.9.1 Buffer Overflow

O *buffer overflow* é o método de ataque mais utilizado, segundo os boletins do CERT, desde 1997. Segundo o CERT, mais de metade dos boletins são relativos a *buffer overflows*. Nesse tipo de ataque, o *hacker* explora *bugs* de implementação, onde o controle do *buffer* (memória temporária para armazenamento dos dados) não é feito. Assim, o *hacker* pode enviar mais dados do que o *buffer* pode manipular, enchendo assim o espaço da pilha de memória. Os dados podem ser perdidos ou jogados, e quando isso acontece o *hacker* pode reescrever no espaço interno da pilha do programa, para fazer com que comandos arbitrários sejam executados. Com um código apropriado, é possível ganhar acesso de super-usuário no sistema [ROT 99-B].

Por exemplo, um *hacker* pode enviar uma URL com grande número de caracteres para o servidor Web. Se a aplicação remota não faz o controle de *strings* longos, o programa pode entrar em pane. Assim, o *hacker* pode colocar códigos maliciosos na área de armazenamento da memória, que pode ser executado como parte de um argumento [ROT 99-B].

Qualquer programa pode estar sujeito a falhas de *buffer overflow*, como sistemas operacionais (Windows NT, Unix), protocolos (TCP/IP, FTP) e aplicações (Microsoft Exchange, Microsoft Internet Information Server, Netscape Communicator). Eles são difíceis de serem detectados, e

portanto a proteção contra eles geralmente é reativo, ou seja, o administrador que sofrer um ataque desse tipo deve reportar o incidente a um órgão especializado, como o CERT, CIAC e também ao fabricante da aplicação. Após isso ele deve aplicar os *patches* correspondentes, assim que eles estiverem disponíveis. As medidas reativas, em detrimento da ação proativa, serão necessárias até que uma metodologia de programação com foco em segurança seja utilizada pelas empresas de softwares.

Uma das formas de programação que permite a atuação de modo proativo é a utilização de localizações aleatoriamente do *buffer* de memória, de modo que o *hacker* não tenha idéia da posição onde colocar o seu código malicioso. O primeiro produto a utilizar essa técnica é o SECURED, da Memco (www.memco.com/products/app_Security.html) [ROT 99-B].

Um exemplo da exploração do *buffer overflow* ocorreu no *site* de leilão on-line eBay, que foi invadido em março de 1999, através da exploração de uma condição de *buffer overflow* em um programa com *SUID root*. O *hacker* pôde instalar assim um cavalo de tróia que interceptava a digitação do administrador, o que possibilitava que nomes de acesso e senhas fossem facilmente capturadas. Com o acesso de super-usuário, o *hacker* podia realizar qualquer operação no *site*, como mudar preços dos produtos em leilão, manipular ofertas e propostas, e tudo mais o que ele desejasse [ROT 99-B].

4.9.2 Ataques Web

Bugs em servidores Web, navegadores Web, *scripts Common Gateway Interface* (CGI) e *scripts Active Server Pages* (ASP) são as vulnerabilidades mais exploradas, mais simples e mais comuns de serem vistas. É através deles que os *hackers* conseguem modificar arquivos dos servidores web, resultando em modificações no conteúdo das páginas Web, e conseqüente degradação da imagem das organizações. São os ataques que ganham destaque nos noticiários, existindo na própria Internet *sites* que divulgam quais foram os *sites* "hackeados".

Além dos ataques mais comuns, que exploram os *bugs* em implementações de scripts CGI, [KIM 99] descreve duas novas tendências de ataques que exploram vulnerabilidades em CGI. O *Poison Null* [PHR 99] permite que conteúdos dos diretórios possam ser vistos, sendo que em alguns casos é possível ler e modificar arquivos dos servidores Web. O mecanismo utilizado mascara comandos da checagem de segurança do CGI, escondendo esses comandos atrás de um

"*null byte*" – um pacote de dados que o *script* CGI não detecta, a menos que ele seja programado especificamente para tratá-lo.

O ataque *Upload Bombing* afeta *sites* que oferecem recursos de *upload*, como aqueles que recebem currículos ou arquivos com desenhos. Esse ataque tem como objetivo encher o disco rígido do servidor com arquivos inúteis. Isso acontece porque os *scripts* não verificam o tamanho dos arquivos a serem submetidos ao *site*, o que impede a proteção do espaço de armazenamento do *site* [KIM 99].

Outro ataque Web conhecido é o *Web Spoofing* ou o *Hyperlink Spoofing*, que pode ser visto em [ODW 97]. O usuário é iludido a pensar que está em uma página que, na verdade, é uma página falsificada. O usuário entra em uma página segura, protegida pelo protocolo SSL, e ele é induzido a fornecer suas informações pessoais ao falso servidor. Uma maneira de se evitar isso é sempre verificar o certificado digital da página. Uma série de propostas são apresentadas pelo autor, como a definição de um objeto da página Web a ser certificada, que pode ser uma imagem (logo da companhia, por exemplo), a URL ou um texto. Isso possibilitaria que o servidor pudesse ser facilmente verificado e conferido pelo usuário no momento de sua entrada em uma página protegida pelo SSL. [FEL 97] trata do mesmo assunto, explicando as dificuldades em identificar pistas de que uma página é falsa. Essa dificuldade se deve principalmente a linguagens como o JavaScript, que permite controlar diretamente objetos do *browser*, como as propriedades da página. Uma das soluções propostas é a de desabilitar o JavaScript e olhar sempre para a barra de endereços (URL), se possível, para verificar se o endereço atual é o correto.

4.9.3 Problemas com o SNMP

McClure e Scambray descrevem em [MCC 99] os problemas que envolvem o *Simple Network Management Protocol* (SNMP) (portas UDP 161 e 162). Segundo eles, o SNMP não possui mecanismos de travamento de senhas, permitindo os ataques de força bruta. A sua configuração *default* pode anular os esforços de segurança pretendidos pelos *TCP wrappers* do Unix ou pelo *RestrictAnonymous Key* do NT. Além disso, o nome da comunidade dentro de uma organização constitui um único ponto de falha, que, caso descoberto, coloca à disposição dos *hackers* informações da rede inteira.

O SNMP pode prover diversas informações, tais como informações sobre o sistema, tabelas de rotas, tabelas *Address Resolution Protocol* (ARP) e conexões UDP e TCP, passando por cima

até mesmo dos sistemas "anti-portas" dos sistemas. O SNMP do Windows NT (*Management Information Base* (MIB)) pode fornecer informações que normalmente são bloqueadas pela chave *RestrictAnonymous*, tais como os nomes dos usuários, dos serviços que estão rodando e os compartilhamentos dos sistemas. Essas informações facilitam o planejamento de ataques pelos *hackers*, de modo que eles devem estar muito bem protegidos.

Algumas defesas contra a exploração do SNMP propostas são [MCC 99]:

- Desabilitar e remover todos os serviços e *daemons* SNMP desnecessários;
- Tratar os nomes da comunidade como senhas, criando dificuldades para que ela seja previsível;
- Restringir as informações para certos *hosts*, como por exemplo, somente para o administrador do sistema. Uma outra medida importante é, em vez de aceitar pacotes SNMP de qualquer *host*, aceitar pacotes SNMP apenas de alguns *hosts* específicos.

O SNMP foi publicado pela primeira vez em 1988, e já no início da década de 90 surgiram diversas deficiências funcionais e de segurança. Em janeiro de 1993 foi lançado a versão 2 do SNMP, que aumentou o desempenho e o suporte descentralizado a arquiteturas de gerenciamento de redes, além de adicionar funcionalidades para o desenvolvimento de aplicações. Porém, o que faltou na versão 2 foram características de segurança, que serão cobertas pela versão 3 do protocolo, que está sendo proposta desde janeiro de 1998. A versão 3 não é uma arquitetura completa, e sim um conjunto de características de segurança que devem ser utilizadas em conjunto com o SNMPv2 [STA 99].

O SNMPv3 provê três características de segurança que a versão 2 não tinha: autenticação, privacidade e controle de acesso. Os dois primeiros são parte do *User-based Security Model* (USM) e o controle de acesso é definido no *View-based Access Control Model* (VACM) [STA 99]:

- Autenticação – faz a autenticação das mensagens e assegura que elas não sejam alteradas ou artificialmente atrasadas ou retransmitidas. A autenticação é garantida através da inclusão de um código de autenticação nas mensagens, que é calculada através de uma função que inclui o conteúdo da mensagem, a identidade do emissor e do receptor, o tempo de transmissão e uma chave secreta conhecida apenas pelo emissor e receptor. A chave secreta é enviada pelo gerenciador de configuração ou de rede para as bases de dados dos diversos gerenciadores e agentes SNMP;

- Privacidade – os gerenciadores e agentes podem cifrar suas mensagens através de uma chave secreta compartilhada, baseada no DES;
- Controle de acesso – permite que diferentes gerenciadores tenham diferentes níveis de acesso ao *Management Information Base* (MIB).

4.9.4 Vírus, Worms e Cavalos de Tróia

Os vírus, *worms* e cavalos de tróia são uma ameaça constante para as organizações, que resultam em diversos tipos de problemas, mais sérios atualmente devido à possibilidade de ser incluído em um ataque distribuído, como foi visto na seção 4.8.

Os *worms* se diferem dos vírus ao se espalharem rapidamente, sem a necessidade de uma interação com o usuário, como ocorre com os vírus. Já os cavalos de tróia, como o Netbus e o Back Orifice, são softwares que aparentam realizar alguma tarefa útil, porém por trás de tudo realizam atividades maliciosas.

Os 3 tipos de vírus existentes são [CAR 99-2]:

- Vírus de setor de *boot* – modificam setores de *boot* nos discos flexíveis, e se espalham quando o computador é iniciado através desse disco flexível com o setor modificado. Como esse tipo de vírus não é transmitido pela rede, pode ser combatido com o anti-vírus baseado no cliente;
- Vírus de macro – infectam e se espalham através das linguagens macros existentes nos documentos do tipo *office*. São armazenados como parte de qualquer documento desse tipo, podendo portanto se espalhar rapidamente, devido à sua enorme quantidade (todo mundo troca documentos) e à possibilidade de serem atachados em *e-mails*;
- Vírus de arquivos executáveis – contaminam arquivos executáveis, espalhando-se através desse tipo de arquivos.

Os vírus vêm se tornando uma ameaça constante para as redes das organizações, de modo que uma estratégia adequada com relação aos anti-vírus é importante. Os anti-vírus atuam na detecção de vírus, *worms* e cavalos de tróia. Uma consideração importante é que os vírus atingem basicamente o ambiente Windows. A indústria de anti-vírus está numa eterna briga de "gato e rato" contra os vírus, de modo que, se por um lado a indústria de anti-vírus, que faz a detecção do vírus através de assinaturas como o *checksum*, está cada vez mais ágil na distribuição de atualizações, por outro lado os vírus novos, principalmente os chamados polimórficos, se modificam

a cada equipamento infectado, dificultando a sua detecção. Os anti-vírus então têm que realizar a detecção através da análise do código binário para detectar peças de códigos de vírus. Além dos vírus polimórficos, outros problemas dificultam a ação dos anti-vírus [SEI 00]:

- A compressão dos vírus com algoritmos de compressão pouco utilizados conseguem driblar muitos anti-vírus. Esse problema já foi parcialmente resolvido;
- A compressão dos vírus com operações XOR dos dados driblam muitos anti-vírus;
- O armazenamento de vírus em diretórios que não são verificados pelos anti-vírus, como o *Recycle bin* do Windows, apesar de que o usuário pode configurar o software para que esse diretório seja também verificado;
- A exploração de vários *buffer overflows* em softwares como o Outlook faz com que o vírus rode sem que o usuário possa escolher entre salvar ou não o arquivo atachado. O problema já foi corrigido;
- Utilização de *system calls* e softwares do Windows, como a utilização do Outlook para enviar um vírus atachado em *e-mails* para todos os usuários da lista, que foi utilizado pioneiramente pelo Melissa;
- Adição de algumas características para que o arquivo atachado não seja verificado pelo anti-vírus.

O ciclo de vida de um vírus pode ser observado a seguir [SEI 00]:

- O vírus é escrito e testado em uma rede experimental;
- O vírus é lançado, possivelmente em um alvo selecionado;
- O vírus se espalha para outras redes, se estiver implementado corretamente;
- Alguém percebe atividades estranhas, recolhe arquivos modificados e os envia para a indústria de anti-vírus;
- O vírus é descompilado e analisado, e uma assinatura é criada;
- O criador do anti-vírus irá compartilhar as informações com seus concorrentes, de modo rápido ou atrasado, dependendo da situação;
- A indústria de anti-vírus espalha boletins de segurança, tornando a atualização disponível;

- Alguns clientes com contrato de suporte podem atualizar rapidamente seus anti-vírus, até mesmo de maneira automática, outros não;
- Os administradores de rede e sistemas e usuários ficam sabendo do vírus através de *e-mails*, vêem os boletins de segurança dos anti-vírus ou vêem a notícia do vírus na imprensa, e atualizam seus anti-vírus.

O episódio do vírus Melissa, que infectou com uma impressionante velocidade centenas de milhares de usuários, mostrou que os vírus são uma ameaça real, principalmente devido à grande velocidade e facilidade de contaminação, incrementada em muito pelos *e-mails*. O Melissa se espalhou rapidamente antes que a atualização dos anti-vírus pudesse ser feita. As dificuldades em uma rápida atualização de todos os anti-vírus de todos os clientes também é muito grande, ao contrário do *gateway* anti-vírus, que é hoje uma solução imprescindível dentro de uma organização. Com ele os vírus são bloqueados antes de entrar na rede, que atuam como a primeira linha de defesa contra os vírus. Porém, outros métodos de defesa contra os vírus ainda devem ser utilizados, principalmente devido à existência de drives de discos flexíveis.

Ironicamente, o avanço dos vírus, que estão a cada dia mais sofisticados, tem como um dos fatores a evolução dos *firewalls*. Os *firewalls*, se bem configurados, dificultam muito a efetividade e eficiência dos ataques, de modo que os *hackers* passaram a buscar formas de invadir a rede interna da organização através da utilização dos tráfegos permitidos pelo *firewall*. Por exemplo, um usuário recebe um arquivo atachado contaminado pelo *e-mail*, ou faz a transferência de um arquivo via FTP, que são permitidos pelo *firewall*. Assim, o vírus infecta a rede, busca informações valiosas na rede, e envia essas informações para o *hacker* via HTTP, que é também um tráfego legítimo para o *firewall*. Aliás, o protocolo HTTP é um problema para as organizações, já que praticamente qualquer tipo de tráfego pode passar pelo *firewall* através do tunelamento HTTP. O HTTP é chamado por alguns como "*Universal Firewall Tunneling Protocol*".

O desempenho do *gateway* anti-vírus pode ser melhorado através da utilização de uma arquitetura de *firewall* integrada, onde um *firewall* decide se um arquivo deve ser enviado para um outro equipamento, no caso o *gateway* anti-vírus. Um dos mecanismos para integração de *firewalls* é o *Content Vectoring Protocol* (CVP) da Check Point Software Technologies, que é parte do *Open Platform for Secure Enterprise Connectivy* (OPSEC). O CVP define uma relação cliente/servidor que permite que *firewalls* dividam um servidor de validação de conteúdo em comum. Assim, caso a regra do *firewall* indique que o conteúdo de um arquivo deva ser verifi-

cado, esse arquivo é enviado a um *gateway* anti-vírus, que o analisa e determina o que fazer com ele. O arquivo é devolvido ao *firewall*, que então passa ou proíbe o tráfego desse arquivo, de acordo com a resposta do *gateway* anti-vírus e com a política de segurança da organização.

4.9.5 War Dialing

O *war dialer* é a ferramenta utilizada pelos *hackers* para fazer a varredura dos números de modens, e é também utilizada pelos auditores de segurança para verificar a existência de modens na organização, que na realidade deveriam ser proibidos. O termo surgiu após o filme *"War Games"*, onde a técnica de varredura de números de telefone foi mostrada. Inspirados no filme, diversos *hacker* começaram a desenvolver as suas próprias *"War Games Dialers"*, agora conhecidas apenas como *war dialers* [GAR 98].

[GAR 98] faz a análise dos *war dialers* disponíveis na Internet, e apresenta um software comercial, o *PhoneSweep*.

4.10 Conclusão

Este capítulo apresentou os riscos que as organizações correm quando passam a manter quaisquer tipos de conexões. Os diversos tipos de atacantes e as suas intenções foram apresentadas, bem como as técnicas mais utilizadas por eles. Um ataque se inicia com a obtenção de informações sobre os sistemas alvos, passando por técnicas que incluem negação de serviços (*Denial of Service - DoS*), ataques ativos, ataques coordenados, e ataques às aplicações e aos protocolos. Pode-se considerar que os maiores perigos estão nas vulnerabilidades resultantes de falhas na implementação dos produtos (sistemas operacionais, protocolos, aplicativos), nas configurações equivocadas dos sistemas, e na engenharia social.

Capítulo 5

Política de Segurança

O objetivo deste capítulo é de demonstrar a importância da política de segurança, discutindo pontos como o seu planejamento, os seus elementos, os pontos a serem tratados e os maiores obstáculos a serem vencidos, principalmente em sua implementação. Alguns pontos específicos que devem ser tratados pela política também são exemplificados, como são os casos da política de senhas, do *firewall* e do acesso remoto, chegando até à discussão da política de segurança em ambientes cooperativos, que possuem suas particularidades.

5.1 A Importância

Foi visto que o ambiente tecnológico das organizações está se tornando cada vez mais complexo e difícil de ser entendido, de tal modo que é fácil o administrador se perder quanto às questões relacionadas à segurança. Assim, cresce a importância da utilização de uma metodologia para o planejamento, implementação e gerenciamento da segurança, que passa a ser um ponto crítico, da qual depende o sucesso da organização.

Esse papel é desenvolvido pela política de segurança, que é a base para todas as questões relacionadas à segurança. O seu desenvolvimento é o primeiro e o principal passo de uma estratégia de segurança das organizações. É através dessa política que todos os aspectos envolvidos com a proteção dos recursos são definidos, e portanto grande parte do trabalho é gasto na sua elaboração e planejamento. Será visto, no entanto, na seção 5.6, que as maiores dificuldades estão mais em sua implementação do que em seu planejamento e elaboração.

A política de segurança é importante para se evitar problemas como os que foram enfrentados pela Omega Engineering Corp. A organização demitiu Timothy A. Lloyd, responsável pela

segurança de sua rede e funcionário durante 11 anos. A demissão causou sérias e caras conseqüências para a Omega. A falta de uma política de segurança quanto ao acesso de funcionários demitidos fez com que o ex-funcionário implantasse uma bomba lógica na rede, que explodiu 3 semanas após ele ter deixado a organização. Os prejuízos calculados com essa ação foram de US$ 10 milhões [ULS 98].

Além do seu papel primordial nas questões relacionadas com a segurança, a política de segurança, uma vez fazendo parte da cultura da organização, terá um importante papel ao facilitar e simplificar o gerenciamento de todos os recursos da organização. De fato, gerenciar a segurança é a arte de criar e gerenciar a política de segurança, já que não é possível gerenciar o que não se pode definir [SEC 99-10].

5.2 O Planejamento

Existem dois métodos de planejar a segurança, que em geral são utilizados de maneira combinada [MAN 99]:

- Avaliação de riscos – quando os valores a serem protegidos são identificados e quantificados, e as ameaças identificadas;
- *Baseline Standard* – quando as preocupações com a segurança partem do nível gerencial, que geralmente resultam no apoio financeiro às soluções necessárias, ajudando assim na resolução de um dos grandes problemas dos profissionais de segurança, a falta de verbas.

Normalmente, a abordagem com relação à segurança é reativa, o que pode invariavelmente trazer futuros problemas para a organização. A abordagem proativa é portanto essencial, e depende de uma política de segurança bem definida, onde a definição das responsabilidades individuais devem estar bem claras, de modo a facilitar o gerenciamento da segurança de toda a organização.

A política de segurança, definida de acordo com os objetivos de negócios da organização, deve existir de maneira formal, pois só assim é possível implementar efetivamente a segurança. Caso essa política formal não exista, os administradores de segurança devem documentar todos os aspectos a serem tratados, sendo imprescindível que a aprovação do gerente seja feita de maneira formal. Essa formalidade evitará que no futuro as responsabilidades recaiam totalmente

sobre os administradores, além de evitar situações onde eventos fora do conhecimento da gerência ocorram e tragam conseqüências inesperadas e tensões desnecessárias para a organização. Além do mais, a política de segurança formal é essencial porque as responsabilidades sobre as questões de segurança devem ser dos gerentes, e não dos administradores de segurança [MAN 99].

Sob a perspectiva do usuário, é essencial que exista a sua participação e o seu denvolvimento na hora de definir as práticas, as tecnologias e os serviços a serem adotados. Esse envolvimento é importante porque medidas de segurança que atrapalham o usuário invariavelmente falham, como foi visto na seção 3.9. As medidas devem possuir a máxima transparência possível para o usuário, de modo que as necessidades de segurança da organização estejam em conformidade com as necessidades dos usuários [MAN 99].

5.3 Os Elementos

Os elementos que uma boa política de segurança deve conter dizem respeito àquilo que é essencial quando o objetivo é combater as adversidades, de modo a manter não apenas a proteção contra os ataques de *hackers*, mas também a disponibilidade da infra-estrutura da organização: vigilância, atitude, estratégia e tecnologia [HUR 99]:

- Vigilância – Significa que todos da empresa devem entender a importância da segurança para a organização, fazendo com que todos ajam como guardiões da rede, evitando-se assim abusos sistêmicos e acidentais. No aspecto técnico, a vigilância significa um processo regular e consistente, que inclui o monitoramento dos sistemas e da rede. Alguns desses aspectos são a definição de como responder a alarmes e alertas, como e quando checar a implementação e mudanças nos dispositivos de segurança, e como ser vigilante com relação às senhas dos usuários (seção 5.8);

- Atitude – Significa a postura e a conduta quanto à segurança. Sem a atitude necessária, a segurança proposta não terá valor algum. Como a atitude não é apenas reflexo da capacidade, e sim reflexo inspirado pelo treinamento e habilidades, é essencial que a política definida esteja em um local de fácil acesso, e que o seu conteúdo seja de conhecimento de todos os usuários da organização. Além disso, é também essencial que esses usuários tenham a compreensão e a cumplicidade quanto à política definida, o que pode ser con-

seguido através da educação e do treinamento. Atitude significa também o seu correto planejamento, já que a segurança deve fazer parte de um longo e gradual processo dentro da organização;

- Estratégia – Significa ser criativo nas definições da política e do plano de defesa contra as intrusões, além de possuir a habilidade de ser adaptativo em mudanças no ambiente, tão comuns no ambiente cooperativo. A estratégia leva também em consideração a produtividade dos usuários, de forma que as medidas de segurança a serem adotadas não influenciem negativamente no andamento dos negócios da organização.
- Tecnologia – A solução tecnológica deve ser adaptativa e flexível para suprir as necessidades estratégicas da organização, pois qualquer tecnologia um pouco inferior resulta em um falso e perigoso senso de segurança, colocando em risco toda a organização. Portanto, a solução ideal que uma organização pode adotar não é um produto, e sim uma política de segurança dinâmica, onde múltiplas tecnologias de segurança e práticas de segurança são adotadas. Esse é o ponto que leva ao conceito de *firewall* cooperativo, que será visto no capítulo 12.

5.4 Considerações Sobre a Segurança

Antes de se criar a política de segurança, é necessário que os responsáveis pela sua criação possuam o conhecimento de diversos aspectos de segurança, além da familiarização com os aspectos culturais, sociais e pessoais que envolvem o bom funcionamento da organização. Algumas das considerações a serem tomadas são [CIS 01]:

- Conheça seu inimigo – determine o que eles desejam fazer e os perigos que eles podem causar à organização;
- Contabilize os valores – a implementação e o gerenciamento da política de segurança geram um aumento no trabalho administrativo e educacional, o que pode significar, além da necessidade de maiores recursos pessoais, a necessidade de significativos recursos computacionais e hardwares dedicados. Os custos das medidas de segurança devem ser assim compatíveis e proporcionais com as necessidades da organização e de suas probabilidades de falhas de segurança;
- Identifique, examine e justifique suas hipóteses – qualquer hipótese esquecida ou escondida pode causar sérios problemas de segurança. Uma única variável pode mudar completamente a estratégia de segurança de uma organização;

- Controle seus segredos – muitos aspectos da segurança são baseados nos segredos, que devem portanto ser guardados a sete chaves;
- Avalie os serviços estritamente necessários para o andamento dos negócios da organização – foi visto nas seção 3.8 e 3.9 que a segurança é inversamente proporcional às funcionalidades, e que a segurança pode influir na produtividade dos usuários. Determinar e justificar cada serviço permitido é essencial para se evitar conflitos futuros com os usuários;
- Considere os fatores humanos – muitos procedimentos de segurança falham porque as reações dos usuários a esses procedimentos não são considerados. Senhas difíceis, que para serem utilizadas são "guardadas" debaixo do teclado, por exemplo, podem comprometer a segurança da rede tanto quanto uma senha "normal". Boas medidas de segurança garantem que o trabalho dos usuários não seja afetada, e a idéia de cada medida a ser adotada deve ser vendida para cada usuário. Eles devem entender e aceitar essas necessidades de segurança. Uma boa estratégia é a formalização de um treinamento de segurança para todos os funcionários da organização, antes de liberar o seu acesso à rede, para passar idéias gerais de proteção dos recursos da organização, como nunca passar senhas por *e-mail* ou telefone, e a maneira mais segura de se navegar pela Internet [DID 98];
- Conheça seus pontos fracos – todo sistema possui suas vulnerabilidades. Conhecer e entender esses pontos fracos é o primeiro passo para poder proteger o sistema de uma maneira eficiente;
- Limite o escopo de acesso – barreiras como uma zona desmilitarizada (DMZ) fazem com que se uma parte da rede for atacada, o resto da rede não seja comprometida. A segurança de uma rede é tão forte quanto a parte mais fraca (menos protegida) dessa rede;
- Entenda o ambiente – entender o funcionamento normal da rede é importante para detectar comportamentos estranhos na rede, antes que um invasor cause prejuízos. Os eventos incomuns na rede podem ser detectados com a ajuda de ferramentas específicas, como o *Intrusion Detection System* (IDS), que será discutido no capítulo 7;
- Limite a confiança – principalmente em softwares que possuem muitos *bugs*, que comprometem a segurança do ambiente;
- Nunca se esqueça da segurança física – acessos físicos indevidos a equipamentos ou a roteadores destróem todas as medidas de segurança adotadas;

- A segurança é complexa – qualquer modificação em qualquer peça da rede pode causar efeitos inesperados na segurança, principalmente quando novos serviços são adicionados. Entender as implicações de segurança em cada aspecto envolvido é importante para a manipulação e gerenciamento correto de todas as variáveis envolvidas;
- A segurança deve ser aplicada de acordo com os negócios da organização – entender os objetivos de negócios da organização é importante para a definição da sua estratégia de segurança. Uma organização que partiu para o *e-commerce*, por exemplo, vendendo seus produtos através da Internet, deve possuir uma atenção especial nas estratégias de proteção da infra-estrutura de vendas on-line;
- "Atividades de segurança formam um processo constante, como carpir a grama do jardim. Se isso não é feito regularmente, a grama (ou *hackers*) cobrirá o jardim." – Gembricki da Warrom [DID 98].

5.5 Os Pontos a Serem Tratados

Uma boa política de segurança deve tratar não só de aspectos técnicos, mas também de aspectos relacionados ao trabalho, às pessoas e ao gerenciamento, como os procedimentos e responsabilidades. Uma política mais complexa pode extrapolar os aspectos relacionados à informática e telecomunicações, partindo para aspectos do cotidiano, como por exemplo, a definição dos cuidados necessários com documentos em mesas de trabalhos, e até mesmo com os lixos, já que ele é um dos lugares mais explorados em busca de informações confidenciais (seção 4.4).

Os aspectos culturais e locais também devem ser considerados na elaboração da política de segurança, pois eles influenciam diretamente na sua efetividade. A política de demissão de funcionários por falha na escolha de senhas, por exemplo, poderia ser aplicada nos Estados Unidos, porém na Europa o usuário poderia ganhar um processo na justiça. Um outro exemplo é a proibição da exportação de criptografia forte que existe nos Estados Unidos e no Reino Unido, que não é aplicado em outros países. Essas peculariedades faz com que a ajuda de um profissional local no desenvolvimento da política da organização seja um importante ponto a ser considerado [MAN 99].

A política de segurança deve definir também, do modo mais claro possível, as punições e os procedimentos a serem tomados em caso do não cumprimento da política definida. Esse é um

importante ponto para que abusos sejam evitados e para que os usuários tenham a consciência de que a política de segurança é importante para o sucesso da organização.

Alguns pontos relevantes em uma política de segurança, definidos com base na análise do ambiente da rede e de seus riscos, são [AVO 94]:

- A segurança é mais importante do que os serviços. Caso não haja conciliação, a segurança deve prevalecer;
- A política de segurança deve evoluir constantemente, de acordo com os riscos e mudanças na estrutura da organização;
- O que não é expressamente permitido é proibido. O ideal é bloquear tudo, e os serviços só poderão ser liberados caso a caso, de acordo com a sua análise e dos riscos relacionados;
- Nenhuma conexão direta vinda do exterior para a rede interna deve ser permitida de modo transparente;
- Os serviços devem ser implementados com a maior simplicidade possível, evitando-se a complexidade e a chance de configurações erradas;
- Testes devem ser realizados para garantir que todos os objetivos sejam alcançados;
- Acessos remotos discados devem ser protegidos com a utilização de um método de autenticação forte e criptografia dos dados;
- Nenhuma senha deve ser passada "em claro", ou seja, sem a utilização da criptografia. Caso não seja possível a sua utilização, o ideal é a utilização do *one-time password* (Capítulo 10);
- Informações utilizadas pela computação móvel devem ser cifradas.

A política de segurança pode também ser dividida entre vários níveis, partindo de um nível mais genérico (para que os gerentes possam entender o que está sendo definido), passando pelo nível dos usuários (para que eles passem a ter consciência de seus papéis para a manutenção da segurança na organização), até o nível técnico (que é onde porções específicas como a implementação e regras de filtragem do *firewall* devem residir).

Possuir uma política proativa também é fundamental, pois sem essa abordagem a questão quanto à segurança das informações não é "se", mas sim "quando" o sistema será atacado por

um *hacker* [DID 98]. De fato, de acordo com o *Defense Information Systems Security Agency* (DISS), a maioria das organizações (mais de 70%) nem ao menos percebem que suas redes foram vítimas de ataques [DID 98-2].

5.6 A Implementação

A implementação é a parte mais difícil da política de segurança. A sua criação envolve conhecimentos de segurança, do ambiente de rede, da organização, da cultura, das pessoas e das tecnologias, sendo uma tarefa complexa e trabalhosa. Porém, a dificuldade reside na implementação dessa política criada, onde todos os usuários da organização devem ter o conhecimento da política, todas as mudanças sugeridas devem ser implementadas e aceitas por todos, e toda a tecnologia definida deve ser implantada com sucesso.

Os esforços necessários para a implantação da segurança podem levar anos até que se consiga um resultado esperado, e portanto um planejamento a longo prazo é essencial, bem como a aprovação formal de todos os passos da política. Assim, o ideal é que a segurança possua seu espaço dentro do orçamento das organizações, com seus devidos planejamentos, equipes e dependências. O ideal é que a segurança seja vista como uma área funcional da organização, assim como a área financeira ou a área de *marketing* [WOO 99].

Um ponto importante quanto à política de segurança é que, contrário à percepção inicial, o seu desenvolvimento ajuda a diminuir, e não a aumentar os custos operacionais. Isso porque a especificação dos objetos a serem protegidos, dos controles e das tecnologias necessárias e de seus respectivos valores, resulta em um melhor controle e gerenciamento da segurança a nível organizacional, em oposição à dificuldade de gerenciamento de soluções isoladas de fornecedores aleatórios [WOO 99].

5.7 Os Maiores Obstáculos para a Implementação

Além da dificuldade natural pertinentes à sua implementação, diversos obstáculos aparecem no meio do caminho para complicar a situação. Alguns desses obstáculos para a implementação da política são [WOO 99]:

- "Desculpe, não existem recursos financeiros suficientes, e as prioridades são outras"

A falta de verbas é o obstáculo mais comum, e isso é até mesmo muitas vezes apenas uma desculpa para que as razões verdadeiras não sejam reveladas. Não conseguir os recursos necessários fundamentalmente refletem a falha em vender para a gerência a idéia da importância das informações e dos sistemas de informações da organização, que devem portanto serem protegidos. Uma maneira prática de conscientizar a gerência sobre esse problema é uma simulação de ataque, que deve necessariamente ser realizado somente após uma aprovação prévia escrita.

- "Por que você continua falando sobre a implementação da política?"

Outro obstáculo é a dificuldade da gerência em compreender os reais benefícios da política para a organização. A política é um meio de assegurar que os objetivos de gerenciamento estão sendo seguidos consistentemente dentro da organização, de tal modo que os gerentes devem ter a consciência de que se a política for adotada, seus próprios trabalhos ficarão consideravelmente mais fáceis. Fazendo com que a implementação da política faça parte explicitamente do projeto, existe a possibilidade de descrever os benefícios trazidos com a política de segurança. Por isso há a necessidade de tratar a implementação da política como um assunto específico, que necessita também da aprovação da gerência.

- "Os esforços para o desenvolvimento da política foram gastos, isso é tudo?"

É preciso que a gerência tenha total compreensão de que somente aprovar e publicar os documentos com a política desenvolvida é insuficiente. Essa compreensão é importante para se evitar que uma má impressão de descaso seja passado para os demais funcionários da organização. A implementação da política desenvolvida requer verbas para o suporte, para os programas de conscientização dos usuários e treinamentos, para a substituição de tecnologia e para o estabelecimento de procedimentos adicionais. Por isso é importante que a implementação faça parte do projeto global de segurança.

- "Nós temos que realmente fazer tudo isso?"

A gerência pode aprovar uma política de segurança apenas para satisfazer os auditores, e isso acaba comprometendo a própria organização, que obtém uma política incoerente e sem detalhes essenciais para o sucesso da política desenvolvida. Esse tipo de comportamento faz com que a gerência deva ser convencida de que o melhor a fazer é atuar de modo proativo, em oposição ao comportamento reativo. Sendo reativo, em caso de algum incidente de segurança, a gerência

será obrigada a agir em circunstâncias negativas e de extrema urgência e pressão, que traz como principal conseqüência problemas com a confiança de clientes, de parceiros de negócios e com a opinião pública. O ideal é mostrar os estudos que provam que é mais barato a perspectiva de "prevenir, deter e detectar" do que a de "corrigir e recuperar".

- "O que você quer dizer com – existem dependências?"

As dependências existentes nos diversos tópicos da política devem ser consideradas para que os esforços não sejam gastos em vão. Por exemplo, uma política que torna mandatório o uso de autenticação forte para todos os acessos remotos deve tratar também dos pontos que dependem dele, como a arquitetura da solução e dos produtos padrões a serem utilizados. Sem isso a sua implementação fica comprometida, com os usuários reclamando que não conseguem trabalhar remotamente (comprometendo a sua produtividade), e a gerência reclamando que os usuários não podem trabalhar remotamente porque não existe tecnologia que permite o acesso remoto seguro.

- "O que você quer dizer com – ninguém sabe o que fazer depois?"

Uma visão abrangente dos problemas relacionados à segurança, juntamente com o conhecimento dos processos de negócios da organização, são essenciais para o desenvolvimento da política. Assim, um líder técnico, profundo conhecedor de aspectos de segurança e com uma visão sobre as tendências e tecnologias na área de segurança, é imprescindível para a implementação das instruções definidas na política.

- "Desculpe, isso é muito complexo"

É necessário conhecer a complexidade que envolve a rede e os sistemas de informação, para que os recursos adequados sejam alocados no desenvolvimento da política de segurança. O fato de algum desses aspectos ser complexo não significa que deva ser ignorado. Para tanto, o auxílio de ferramentas para a realização dessa tarefa, tais como um software de planejamento de contingência, deve ser utilizado. Essa mesma complexidade exige que a gerência aloque recursos para sistemas de gerenciamento de redes, sistemas de detecção de intrusões, sistemas de automação de distribuição de software, sistemas de checagem de licenças de software e outros mecanismos de automação, que os humanos não podem realizar sozinhos. Mostrar as novas

ferramentas existentes, e o porquê de suas popularidades para a gerência é importante, para mostrar que a complexidade é gerenciável.

- "A política irá fazer com que eu perca meus poderes?"

Alguns gerentes locais podem resistir à implementação da política geral da organização por achar que isso traz ameaças aos seus poderes e prestígio. Mostrar a esses gerentes a importância da centralização e coordenação da política é essencial para que eles dêem o suporte necessário para o sucesso da implementação. Um caso típico da importância da centralização e padronização é no controle de acesso, onde uma coordenação adequada evita o caos, aborrecimentos e desperdício de esforços para todos.

- "Por que eu tenho que me preocupar com isso? Isso não é o meu trabalho"

Geralmente, a gerência não gosta de compartilhar os detalhes técnicos da segurança. Porém, é importante que todos estejam engajados nesse processo, porque a gerência precisa entender que a segurança da organização não irá a lugar algum se não tiver o seu devido suporte. Além disso, a participação ativa da gerência no desenvolvimento e implementação da política é essencial para o seu sucesso, principalmente porque diversas decisões de negócios incluídas na política não podem ser tomadas pelo pessoal técnico, e sim somente pela gerência. Um exemplo é a política de privacidade de um *site* de comércio eletrônico, que demonstra que a segurança é multi-disciplinar, requerendo participações de todos dentro da organização.

- "Nós não podemos lidar com isso, pois não temos um processo disciplinar"

Um processo disciplinar claro para os casos de não cumprimento da política definida é importante para a organização. Por exemplo, se um usuário comete um erro, a primeira medida é avisá-lo da falta. Se o erro se repetir, o chefe do usuário recebe um comunicado. Pelo terceiro erro, o usuário é suspenso por 2 semanas, e se o erro persistir o usuário é demitido. Essa abordagem em fases deve ser bem definida na política, de modo a evitar situações onde o usuário é sumariamente demitido logo no seu primeiro erro, somente para mostrar para os outros quem possui o poder.

5.8 Política para Senhas

A política de senhas é uma parte específica da política de segurança. As senhas são utilizadas pela grande maioria dos sistemas de autenticação (capítulo 10), e são consideradas um meio fraco de proteção, principalmente porque elas dependem do ponto mais fraco da corrente da segurança, o usuário humano. Por isso, a escolha de uma boa senha é essencial para a garantia de um bom nível de segurança. Porém, a política de senhas é apenas uma pequena parte a ser considerada dentro de um sistema de segurança, pois se ela for passada em texto claro pela rede, ela pode ser capturada através de *sniffing*, como pôde ser analisada na seção 4.5.2. O tipo de autenticação (capítulo 10), e se a passagem das senhas pela rede é realizada através de criptografia ou não, também devem ser considerados na política de segurança.

Um modo de comprometer as senhas é através do *crack*, um software que realiza a cifragem de palavras do dicionário (ataque do dicionário), e as testa com as senhas do arquivo de senhas, até que elas sejam equivalentes. Outros testes envolvem todas as combinações possíveis de caracteres (ataque de força bruta). [SHA 98-C] apresenta uma dessas ferramentas, o L0phtCrack. Existem 3 modos para se obter o arquivo de senhas do Windows NT: através do registro do Windows NT (pode ser prevenido através da proibição de acesso remoto e através do utilitário *SYSKEY*, que cifra o *hash* de senhas), diretamente através do arquivo SAM no disco (diretamente do disco do servidor, do *Emergency Repair Disk* ou de um *backup* qualquer) ou através do *sniffing* na rede.

As senhas são, além de um fator importante dentro de um sistema de segurança, um importante fator também na produtividade dos usuários. O esquecimento da senha é um fato comum, e tem como conseqüência a queda na produtividade do usuário e o aumento dos custos com o *help-desk*. Segundo um estudo da Hurwitz Group [FOO 98], 61% das ligações para o *help-desk* são devidos a esquecimento de senhas. Uma boa política de senhas assim significa uma melhor produtividade dos usuários e menores custos com o *help-desk*.

[KES 96] mostra a fraqueza das senhas e descreve os ataques possíveis contra as senhas, indicando como se deve escolher uma boa senha. Uma avaliação sobre o tamanho apropriado de uma senha também é feita, com a conclusão de que o que o ser humano consegue armazenar em seu cérebro são senhas sem muitos caracteres, o que compromete a sua eficiência, se comparado com uma senha aleatória, que por sua vez é difícil de ser decorada pelo usuário (causa

problemas também porque o usuário certamente vai anotar essa senha em algum lugar). A conscientização dos usuários quanto aos perigos de uma senha mal escolhida devem também fazer parte da política de segurança.

O comportamento dos usuários na escolha das senhas pode ser observado através de uma pesquisa realizada pela Compaq em 1997 [JOH 98]. A pesquisa revelou que as senhas são assim escolhidas:

- Posições sexuais ou nomes abusivos a chefes (82%);
- Nomes ou apelidos de parceiros (16%);
- Nome do local de férias preferido (15%);
- Nome de time ou jogador (13%);
- O que vê primeiro na mesa (8%).

A Shake Communications em [SHA 98] apresenta uma série de medidas que podem ser tomadas para configurar um sistema baseado em senhas de modo seguro e eficiente. Algumas dessas idéias são:

- Entre o próprio usuário e o administrador escolher a senha, é melhor que o administrador escolha a senha, pois o usuário geralmente escolhe palavras comuns, como os que existem em dicionários, nomes de filmes, nomes de animais de estimação ou datas de aniversários, que são facilmente descobertos através de programas de *crack*;
- A senha deve ser redefinida pelo menos a cada dois meses para usuários comuns e a cada mês para usuários com acessos mais restritos;
- Os dados do último acesso, como o tempo, a data e a origem são importantes para que o usuário tenha certeza de que sua conta não foi acessada por pessoas não autorizadas;
- As senhas devem ser travadas a cada 3 ou 5 tentativas erradas, e o administrador do sistema e o usuário devem ser notificados sobre as tentativas;
- A transmissão da senha deve ser feita de modo cifrado;
- Atividades de autenticação devem ser registradas e auditadas, como as tentativas com sucesso, sem sucesso, tentativas de mudança de senha, etc;

- As senhas e as informações relativas à conta devem ser armazenadas de modo extremamente seguro, de preferência em um sistema não conectado à rede da organização;
- As responsabilidades do administrador do sistema incluem o cuidado na geração e alteração da senha dos usuários, além de manter atualizados os dados dos usuários, como números de telefones e endereços, para a sua rápida localização caso isso seja necessário;
- As responsabilidades dos usuários incluem principalmente os cuidados para manter a segurança dos recursos, tais como a confidencialidade da senha e o monitoramento de sua conta contra a utilização indevida. Um treinamento sobre a segurança deve ser dado pela organização para que cada usuário tenha a consciência da importância de atitudes básicas, como a de nunca passar sua senha pelo telefone para alguém que diz ser o administrador do sistema.

O *Request for Comments* (RFC) 1244 oferece um guia de como selecionar e manter senhas. Alguns desses pontos [KES 96], e outras recomendações [SHA 98] são:

- Não utilize palavras que estão em dicionários (nacionais ou estrangeiros);
- Não utilize informações pessoais fáceis de serem obtidas, como o número da rua, bairro, cidade, data de nascimento, nome do time preferido, etc;
- Não utilize senhas somente com dígitos ou letras;
- Utilize senhas com pelo menos 8 caracteres;
- Misture caracteres maiúsculos e minúsculos;
- Misture números, letras e caracteres especiais;
- Inclua pelo menos um caractere especial ou símbolo;
- Utilize um método próprio para lembrar a senha, de modo que ela não deve ser escrita em local algum, em hipótese alguma;
- Não use o nome de usuário;
- Não use o primeiro nome, o nome do meio ou o sobrenome;
- Não use os nomes de pessoas próximas, como os da esposa, filhos, amigos ou animais de estimação;
- Não use senhas com repetição do mesmo dígito ou letra;

- Não passe sua senha para ninguém, por nenhuma razão;
- Use senhas que podem ser digitadas rapidamente, sem a necessidade de olhar para o teclado.

Uma receita recomendada é a de pegar a primeira letra de uma expressão, frase, letra de música ou diálogo que faça parte da vida do usuário, de modo que seja fácil de memorizar. Outra sugestão é alternar entre uma consoante e uma ou duas vogais, ou concatenar duas palavras curtas com um ponto entre elas [KES 96].

5.9 Política Para Firewall

Um dos principais elementos da política de segurança para o *firewall* é a definição das regras de filtragem, que por sua vez é baseada na definição dos serviços a serem providos para os usuários externos e dos serviços que os usuários internos podem acessar.

Uma das partes da política de segurança para *firewall* é a definição da arquitetura do *firewall* (seção 6.4). É com base nessa arquitetura e nos serviços definidos que as regras de filtragem são desenvolvidas. A abordagem a ser utilizada pode ser a de "proibir tudo e liberar somente aqueles serviços que forem explicitamente liberados" ou a de "liberar tudo e proibir somente aqueles serviços que forem explicitamente proibidos". As regras de filtragem e a complexidade de sua definição em um ambiente cooperativo serão discutidas no capítulo 12.

5.10 Política Para Acessos Remotos

Um sério problema nas redes das organizações é que elas estão se preocupando mais com a sua conexão com a Internet, se esquecendo dos riscos que envolvem o acesso remoto. Alguns exemplos de incidentes que ocorreram, relacionados ao problema de modens, podem ser vistos em [GAR 98]. Em um deles, ocorrido em março de 1997, um adolescente fez uma varredura em números telefônicos de sua área, utilizando uma ferramenta disponível na Internet. Através dela ele conseguiu o controle total de um sistema de comunicação de fibra ótica. Ele desligou o sistema, desligando as comunicações da torre de controle do aeroporto local e dos serviços de emergência por diversas horas. Em um outro incidente, a Caterpillar Inc., que possuia um sofisticado *firewall*, teve a sua rede interna atacada através de um modem. Esse incidente mostrou a

importância da segurança em acessos remotos, porque nesses casos um *firewall* sofisticado não faz diferença, pois o ataque não vem da Internet, mas sim através da linha telefônica.

Alguns dos riscos envolvidos incluem o funcionário que instala um modem e configura um software em seu equipamento para permitir acessos irrestritos dentro da rede interna, a fim de facilitar seus trabalhos, ou mesmo para desfrutar de acesso gratuito à Internet, via a rede da organização.

A utilização de *war dialers* (seção 4.9.5) para a detecção desses modens clandestinos deve fazer parte da política de segurança. Para os casos onde os modens são necessários, é essencial que exista um documento escrito que esclareça aos usuários os pontos relacionados à segurança.

5.11 Política de Segurança em Ambientes Cooperativos

Foi visto até agora o significado e os aspectos que devem ser tratados pela política de segurança de uma organização. Mas, e quanto aos ambientes cooperativos? A política de segurança em um ambiente cooperativo torna-se cada vez mais complexo, à medida que esse ambiente vai aumentando através de novas conexões. Como cada organização possui a sua própria política de segurança, cada um idealizado de acordo com a sua própria cultura organizacional, em um ambiente cooperativo essa mesclagem de diversas políticas diferentes pode ser fatal à segurança de todos dentro desse ambiente. As questões que ficam são: onde começa e onde termina a política de segurança de cada usuário em um ambiente cooperativo? O ambiente cooperativo deve ter a sua própria política de segurança?

O exemplo clássico de problemas envolvendo diferentes conexões é o caso de triangulação (figura 5.1), discutido na seção 2.3, onde existem 3 organizações, A, B e C, cada um com a sua própria política de segurança. A política de A permite que usuários de C acessem seu banco de dados, porém usuários de B são proibidos de acessar esses dados. A política de C permite que usuários de B acessem sua rede. Como usuários de C podem acessar os dados de A, e C permite que usuários de B acessem sua rede, então B pode acessar A através de C. Isso demonstra que a política de segurança de A é contrariada, em um caso típico de triangulação para driblar a política de segurança da organização A.

Em ambientes cooperativos esse tipo de confusão passa a ser um fato corriqueiro, a menos que haja uma concordância mútua prévia entre as políticas das organizações do ambiente. Por

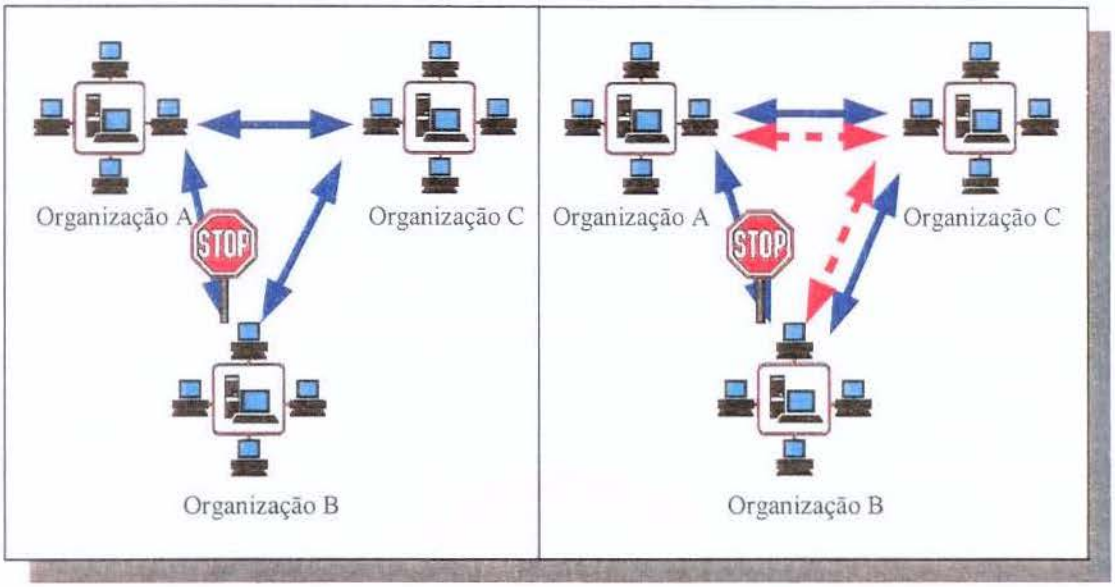


Figura 5.1: A triangulação que dribla a política de segurança de uma organização.

exemplo, no caso onde a organização C funcionou como ponte, usuários de B podem acessar C através de Telnet. A partir daí o usuário de B pode acessar A, que é proibido, através de C. Talvez seja possível convencer A a não permitir mais a utilização do Telnet, porém isso parece ser improvável de acontecer.

Essa grande dificuldade que aparece nos ambientes cooperativos poderia ser minimizada através da criação de uma política de segurança conjunta, que seria adotada pelos integrantes do ambiente cooperativo. Porém, de acordo com o que foi visto, grandes dificuldades irão aparecer, desde a complexidade no desenvolvimento dessa política, até a enorme complicação na sua implementação. Esses contratempos previstos fazem com que essa idéia seja praticamente descartada. Não seria possível, ainda, garantir que todos os integrantes do ambiente cooperativo cumpram o que estaria determinado na política criada.

Desta forma, a idéia que deve ser seguida dentro do ambiente cooperativo é a de que, assim como em um ambiente normal, os usuários de outras organizações devem ser considerados usuários não-confiáveis. Uma vez que os usuários externos entram na rede da organização, eles devem ter todos os seus passos controlados, para que abusos sejam evitados. É necessário que

esse usuário seja controlado como se ele fosse um outro usuário qualquer, ou seja, ele pode ter acesso somente aos recursos para ele permitidos.

Um modelo proposto neste trabalho para sintetizar o que acontece dentro de um ambiente cooperativo é o "Modelo de Bolsões de Segurança". O modelo de segurança convencional tinha como objetivo criar uma parede (representada pelo *firewall*) entre a rede interna da organização e a rede pública. Os usuários externos praticamente não possuíam acesso aos recursos internos da organização (figura 5.2).

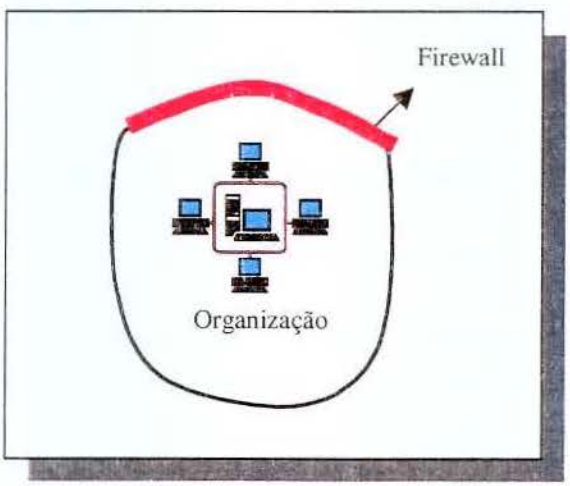


Figura 5.2: Modelo de segurança convencional, representada pelo *firewall*.

Algumas organizações passaram então a disponibilizar serviços para a rede pública, como são os casos típicos do HTTP e do FTP. Nesse modelo, porém, o controle era ainda realizado pelo *firewall*, que liberava os acessos externos para uma rede específica, a rede DMZ. Assim, os acessos externos eram somente para uma área claramente delimitada (DMZ), com a rede da organização ainda continuando a estar isolada contra os ataques externos (figura 5.3).

No ambiente cooperativo porém, tudo muda, pois os níveis de acesso agora variam entre serviços da rede DMZ e serviços internos da organização (através da VPN), e os usuários agora não ficam restritos apenas à área delimitada pela DMZ. Com o modelo proposto, os usuários podem, de acordo com o seu nível de acesso, acessar bolsões de segurança cada vez maiores.

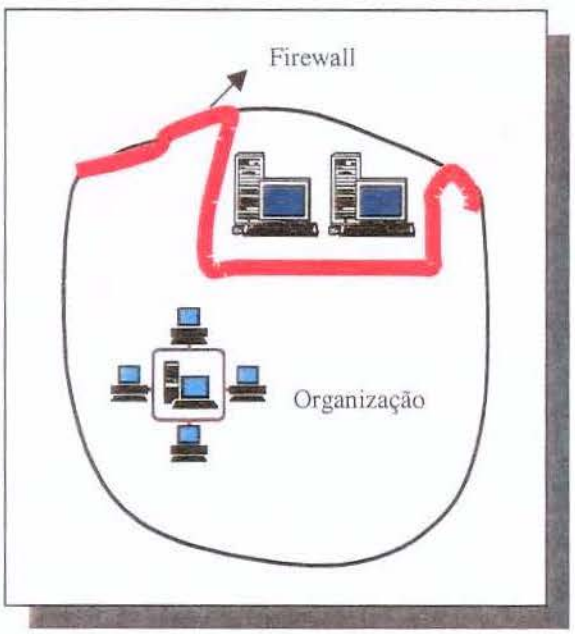


Figura 5.3: Modelo de segurança convencional, representada pelo *firewall* com DMZ.

Se antes as organizações tinham que proteger a DMZ, agora eles têm que proteger esses bolsões de segurança, como pode ser visto na figura 5.4.

Esse modelo pode ser utilizado também para a segurança interna da organização, fazendo com que os próprios usuários internos sejam tratados como usuários externos, tendo que passar pelo controle de acesso para utilizar os recursos desses bolsões. De fato, isso está se tornando cada vez mais necessário, como pode ser visto na seção 12.1.

Assim, cada integrante do ambiente cooperativo deve ser o responsável pela sua própria segurança, reforçando desta maneira a importância de uma política de segurança bem definida. Um integrante de um ambiente cooperativo que não tiver essa política bem definida, passará a ser um alvo fácil de ataques, não só pelos usuários desse ambiente cooperativo, mas também por qualquer outro tipo de usuários externos.

Como resultado, cada organização tem como objetivo criar a sua própria política de segurança para cada bolsão de segurança que ele terá que suportar. Determinados tipos de usuários acessam bolsões de segurança diferentes, que são maiores tanto quanto forem os seus direitos

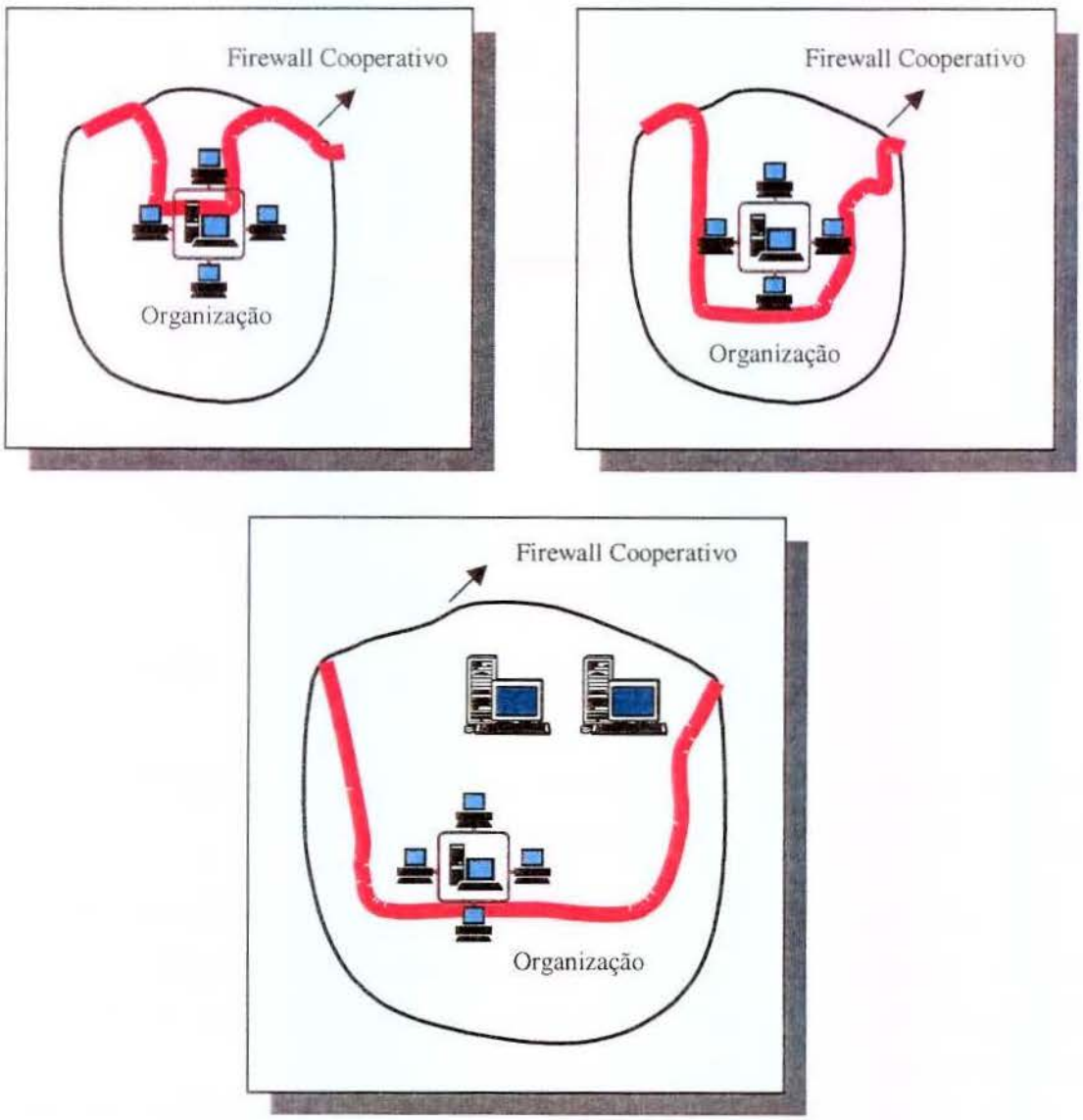


Figura 5.4: Modelo "Bolsões de Segurança", representada pelo *firewall* cooperativo.

de acessos. Representantes comerciais, por exemplo, acessariam bolsões de segurança menores, constituídos pelo banco de dados de estoques e pela lista de preços dos produtos. Já os usuários móveis do setor financeiro, por exemplo, teriam acesso a um bolsão de segurança maior, constituído pelo banco de dados financeiros, acessos a documentos confidenciais e a *e-mails*, praticamente como se ele estivesse trabalhando na própria organização.

5.12 Conclusão

A política de segurança é o principal elemento dentro do esquema de segurança de uma organização. O seu planejamento e a definição dos pontos a serem tratados inclui uma avaliação de todos os aspectos envolvidos, que requer esforços de todos da organização. Diversos obstáculos para a sua implementação são resultantes da visão cega de que a segurança não é um elemento importante para a organização, que pode-se notar que é uma visão equivocada, que invariavelmente traz sérias consequências com a invasão de *hackers*. Alguns pontos específicos requerem uma política específica, como são os casos do acesso remoto, das senhas e do firewall, que foram discutidos neste capítulo. A política de segurança possui uma importância ainda maior em um ambiente cooperativo, onde os bolsões de segurança - definidos neste capítulo - variam de tamanho de acordo com as necessidades de conexões.

Capítulo 6

Firewalls

Este capítulo trata de um dos principais componentes de um sistema de segurança, o *firewall*. O capítulo tem como objetivo discutir a definição do termo *firewall*, que vem sofrendo modificações com o tempo, além de discutir a evolução que vem ocorrendo neste importante componente. As arquiteturas de um *firewall*, que tem como evolução natural o *firewall* cooperativo, também são apresentadas, passando pelo desempenho, mercado, avaliação, testes e problemas encontrados, até a conclusão de que o *firewall* por si só não garante a segurança de uma organização.

6.1 Definição e Função

A necessidade de utilização cada vez maior da Internet pelas organizações leva à uma preocupação cada vez mais crescente quanto à segurança. Como conseqüência, pode-se ver uma grande evolução na área de segurança, principalmente com relação ao *firewall*, que é um dos principais, mais conhecidos e antigos componentes dentro de um sistema de segurança. A sua fama de certa forma acaba contribuindo para a criação de uma falsa expectativa quanto à segurança total da organização, como será discutido na seção 6.10, além de causar uma mudança ou mesmo uma banalização quanto à sua definição. Alguns dos diversos conceitos relacionados ao termo "*firewall*" são:

- Tecnologia do *firewall*, que pode ser filtro de pacotes (*static packet filter*), *proxy* (*application-level gateway* e *circuit-level gateway*), filtro de estados (*dynamic packet filtering, stateful inspection*). Essas e outras tecnologias serão discutidas na seção 6.3, durante a análise da evolução técnica dos *firewalls*;

- Arquitetura do *firewall*, que utilizam componentes como roteadores escrutinadores, *proxies*, zonas desmilitarizadas (*Dismilitarized Zone* – DMZ ou *perimeter network*) e *bastion hosts*, formando as arquiteturas conhecidas como *Dual-Homed Host Architecture, Screened Host Architecture* e *Screened Subnet Architecture*. Essas são as arquiteturas clássicas, cuja abordagem será discutida na seção 6.4. Os componentes que formam uma arquitetura serão discutidos na seção 6.2. Será visto também que novos componentes e funcionalidades foram sendo acrescidas ao *firewall*, sendo que elas não foram incluídas nas arquiteturas clássicas. Assim, uma nova arquitetura, que contempla o que surgiu de novo, também será apresentada na seção 6.4;
- Produtos comerciais como o Check Point Firewall-1, Network Associates Inc's Gauntlet, Cisco Pix Firewall, Watchguard, e outros;
- Produtos componentes da arquitetura do *firewall*, como roteadores (Cisco IOS) ou *proxies* (Microsoft Proxy 2.0);
- Tecnologia responsável pela segurança total da organização.

A mais antiga definição para *firewalls* veio de Bill Cheswick e Steve Bellovin, em *Firewalls and Internet Security: Repelling the Wily Hacker* [CHE 94]. Segundo eles, *firewall* é um ponto entre duas ou mais redes pelo qual passa todo o tráfego. A partir desse único ponto é possível controlar e autenticar o tráfego, além de registrar, através de *logs*, todo o tráfego da rede, facilitando assim a sua auditoria [AVO 99].

Já Chapman [CHA 95] define *firewall* como sendo um componente ou conjunto de componentes que restringe o acesso entre uma rede protegida e a Internet, ou entre outros conjuntos de redes.

Partindo-se dessas duas definições clássicas, pode-se dizer que um *firewall* é um ponto entre duas ou mais redes, ponto este que pode ser um componente ou um conjunto de componentes, por onde passa todo o tráfego, permitindo que o controle e/ou autenticação e registros de todo o tráfego seja realizado. Assim, esse ponto único constitui um mecanismo utilizado para proteger, geralmente, uma rede confiável de uma rede pública, não-confiável. Um *firewall* pode ser utilizado também para separar diferentes sub-redes, grupos de trabalhos ou LANs dentro de uma organização, porém neste trabalho o enfoque será dado para conexões entre organizações diferentes. Os mecanismos utilizados pelo *firewall* para controlar o tráfego serão vistos na seção 6.3.

e o modo de criar a política de segurança para as regras de filtragem, e depois implementá-las, serão vistos no capítulo 12.

Uma outra forma de definir um *firewall* é que ele é um sistema ou um grupo de sistemas que reforça a política de controle de acesso entre duas redes, e portanto pode ser visto como uma implementação da política de segurança. O *firewall* é tão seguro quanto à política de segurança que ele suporta, sendo que não deve-se esquecer que um *firewall* muito restritivo, e portanto mais seguro, não é sempre transparente ao usuário, de modo que alguns usuários podem tentar driblar a política de segurança da organização para poder realizar algumas tarefas a que estavam acostumados, como foi visto na seção 3.9.

6.2 Funcionalidades

O *firewall* é composto por uma série de componentes, onde cada um deles possui uma funcionalidade diferente, e realiza um papel que influi diretamente no nível de segurança do sistema. Algumas dessas funcionalidades formam o chamado componentes clássicos de um *firewall*, definidos por Chapman [CHA 95]. As 4 primeiras funcionalidades fazem parte desse grupo, e as 3 funcionalidades restantes foram inseridas no contexto, devido à evolução natural das necessidades de segurança. Serão analisadas a seguir quais são essas funcionalidades do *firewall*. Algumas delas serão explicadas com mais detalhes na seção 6.3, dentro do contexto da evolução que ocorreu e está ocorrendo nos *firewalls*.

6.2.1 Filtros

Os filtros realizam o roteamento de pacotes de maneira seletiva, ou seja, aceitam ou descartam pacotes através da análise das informações de seus cabeçalhos. Essa decisão é tomada de acordo com as regras de filtragem definidas na política de segurança da organização. Os filtros podem, além de analisar os pacotes, tomar decisões baseadas em estados das conexões, como serão vistos, respectivamente, nas seções 6.3.1 e 6.3.2.

6.2.2 Proxies

Os *proxies* são softwares que atuam como um *gateway* entre duas redes, permitindo as requisições dos usuários internos e as respostas dessas requisições, de acordo com a política de segu-

rança definida. Eles podem atuar simplesmente como um *relay*, podendo também realizar uma filtragem mais apurada dos pacotes, por atuar na camada de aplicação do modelo ISO/OSI. Os *proxies* serão vistos com mais detalhes na seção 6.3.3.

6.2.3 Bastion Hosts

Os *bastion hosts* são equipamentos que devem ser protegidos o máximo que for possível, pois são eles que estarão em contato direto com as conexões externas, estando portanto passíveis de ataques. Essa máxima proteção possível significa que o *bastion host* deve rodar apenas os serviços e aplicativos essenciais, bem como possuir sempre a última versão desses serviços e aplicativos, sempre com os *patches* de segurança instalados imediatamente após a sua criação. Uma grande interação ocorre entre os *bastion hosts* e a zona desmilitarizada (DMZ), já que os serviços que serão oferecidos pela DMZ devem ser inequivocamente instaladas em *bastion hosts*.

6.2.4 Zona Desmilitarizada

A zona desmilitarizada (DMZ) ou o *perimeter network* é uma rede que fica entre a rede interna, que deve ser protegida, e a rede externa, de modo que caso algum equipamento dessa rede desmilitarizada (um *bastion host*) seja comprometido, a rede interna continuará intacta. A DMZ será melhor discutida no capítulo 11, onde será possível entender a sua importância e necessidade.

6.2.5 Network Address Translation (NAT)

O NAT não foi criado com a intenção de ser um componente de segurança, mas sim para tratar de problemas em redes de grande porte, onde a escassez de endereços IPs poderia ser um problema. A rede interna assim pode utilizar endereços IPs inválidos (RFC 1918), sendo o NAT responsável pela conversão desses endereços inválidos para endereços válidos, quando a rede externa é acessada. O NAT pode assim esconder os endereços dos equipamentos da rede interna, e conseqüentemente a sua topologia de rede, tornando eventuais ataques externos mais trabalhosos.

6.2.6 Virtual Private Network (VPN)

A VPN foi criada inicialmente para que protocolos não IP pudessem trafegar pela rede IP. Como não era aceitável que as informações trafegassem em claro pela Internet, a VPN passou a utilizar conceitos de criptografia para manter a confidencialidade dos dados. Mais do que isso, o IPSec, protocolo padrão de fato das VPNs, garante, além da confidencialidade, a integridade e a autenticação desses dados. A VPN será discutida com maiores detalhes no capítulo 9.

6.2.7 Autenticação/Certificação

A autenticação e/ou certificação dos usuários podem ser baseados em endereços IPs, senhas, certificados digitais, *tokens*, *smartcards* ou biometria. Tecnologias auxiliares são a *Public Key Infrastructure* (PKI) e o *Sign Sign-On* (SSO). Os aspectos da autenticação dos usuários e o SSO serão vistos no capítulo 10, e a PKI será visto na seção 8.6.

6.3 A Evolução Técnica

O *firewall* é considerado uma tecnologia "antiga" na indústria de segurança, porém não pode ser considerado ainda uma tecnologia estável, já que ele continua em constante processo de evolução, principalmente devido ao aumento da complexidade das redes das organizações, que adicionam cada vez mais características e funcionalidades, que precisam ser protegidas. Algumas das funcionalidades adicionadas ao *firewall* são importantes, como são os casos do NAT ou da VPN, enquanto outras são respostas à demanda do mercado, como a inserção de serviços como servidor Web, destinadas à organizações pequenas, que podem no entanto acabar tendo um resultado inverso, ou seja, podem ser perigosos para a segurança da rede da organização (seção 3.8).

A utilização crescente da Internet para os negócios, combinada com incidentes como o de Morris Worm, mostraram que a Internet é um mundo de novas oportunidades, porém um terreno pantanoso para a realização desses negócios. Assim, a necessidade de uma segurança melhor e mais granular fez com que empresas como a DEC e a AT&T desenvolvessem soluções para o acesso seguro à Internet. Algumas dessas soluções tornaram-se produtos comerciais (DEC, Raptor, ANS e TIS), que se concentraram na segurança dos serviços básicos como o Telnet, FTP, *e-mail* e news Usenet [AVO 99].

Os primeiros *firewalls* foram implementados em roteadores a cerca de 10 anos atrás, por serem eles o ponto de ligação entre duas redes. As regras de filtragem dos roteadores eram baseados em decisões do tipo "permitir" ou "descartar", que eram tomadas de acordo com a origem, o destino e o tipo do pacote IP [AVO 99].

A partir disso, tudo mudou rapidamente, de modo que a própria definição de que o *firewall* deve separar "nós" "deles" mudou. O mundo tornou-se mais integrado, e os serviços básicos hoje são o acesso à Web, acesso a banco de dados via Internet, acesso a serviços internos da organização via Internet, serviços de áudio, vídeo, vídeo conferência, voz sobre IP, entre tantos outros. Com isso, as organizações vêm tendo cada vez mais usuários utilizando seus serviços, muitas vezes utilizando serviços críticos como se estivessem fisicamente dentro da organização.

Assim, os novos requerimentos de segurança fizeram com que os *firewalls* se tornassem mais complexos, resultando nos avanços verificados nas tecnologias de filtro de pacotes, *proxies*, inspeção de estados, híbridos e os adaptativos, sendo que os dois últimos surgiram em 1999, mas que são na realidade uma mistura das tecnologias já existentes, como será visto a seguir. Além disso, novos nomes a supostas tecnologias de *firewalls* surgiram, tais como o *firewall* reativo e o *firewall* individual, mas que, como será visto a seguir, são na realidade apenas *firewalls* com novas funcionalidades ou para fins específicos.

Além dos avanços na tecnologia e nas funcionalidades inseridas nos *firewalls*, eles se tornaram ainda o sistema de base para os outros serviços da rede e de segurança, como a autenticação (controle de acesso), criptografia (VPN), qualidade de serviço e filtragem de conteúdo, como anti-vírus, filtragem de URL e filtragem de palavras-chaves para *e-mails* [AVO 99].

Pode-se considerar assim que atualmente existe uma tendência para adicionar cada vez mais funcionalidades aos *firewalls*, que podem não estar relacionadas necessariamente à segurança, como servidores Web, servidores FTP, servidores de *e-mail* ou servidores *proxy* (não relacionados à segurança, como *proxy* de *stream* áudio e vídeo) integrados. Isso vêm contra o dogma da segurança, de que a segurança e a complexidade são inversamente proporcionais (seção 3.8), e portanto podem comprometer a segurança ao invés de incrementá-la. Uma boa prática é separar as funções (gerenciamento Web e gerenciamento de segurança), a não ser que a organização seja pequena e o administrador do *firewall* seja também o webmaster e o administrador de todos os sistemas da organização. Quanto mais funções possuir o *firewall*, maiores as chances de alguma coisa sair errado. Além disso, quanto maior o número de serviços, maiores os *logs*, e

quanto maior o número de usuários, maiores os trabalhos com a administração, o que leva à maior possibilidade de erros, que por si próprio já compromete a segurança da organização [AVO 99].

As principais tecnologias de *firewalls* e as suas variações serão discutidas nas seções a seguir.

6.3.1 Filtro de pacotes

Esse tipo de filtro trabalha na camada de rede e de transporte da pilha TCP, de modo que realiza as decisões de filtragem baseando-se nas informações do cabeçalho dos pacotes, como endereço de origem, endereço de destino, porta de origem, porta de destino e direção das conexões. Normalmente, essas regras são definidas de acordo com endereços IPs ou serviços (de acordo com as portas TCP/UDP relacionadas) permitidos ou proibidos, e são estáticas, de modo que esse tipo de *firewall* é conhecido também como *static packet filtering*.

O fato de trabalhar na camada de rede e de transporte faz com que ele seja simples, fácil, barato e flexível de se implementar, de modo que a maioria dos roteadores, que já atuam como *gateway*, possuem também essa capacidade. Isso torna o filtro de pacotes transparente ao usuário, garantindo também um maior desempenho, se comparados com os *proxies*. Em contrapartida, o filtro de pacotes garante um menor grau de segurança, já que os pacotes podem ser facilmente falsificados ou criados especificamente para que passem pelas regras de filtragem definidas. Além disso, um filtro de pacotes não é capaz de distinguir entre pacotes verdadeiros e pacotes falsificados. A capacidade de verificação do sentido dos pacotes para determinar se um pacote vem da rede externa ou interna, e a sua apropriada configuração, é essencial para se evitar ataques como o *IP spoofing* (seção 4.5.6). Na realidade, o que pode ser evitado é a exploração de endereços de equipamentos internos por um *host* externo, sendo impossível um filtro de pacotes evitar *IP spoofing* de endereços públicos reais falsificados.

Uma outra conseqüência da simplicidade dos filtros de pacotes é a sua limitação com relação a *logs* e aos alarmes. Além disso, o suporte a serviços como FTP, X11 e RPC não são simples de se implementar apenas com base no cabeçalho desses pacotes, porque esses serviços utilizam dois canais de comunicação ou portas dinâmicas. Um outro problema que ocorre com esse tipo de filtro é com relação à fragmentação de pacotes (seção 4.6.3), que podem passar pelo *firewall* através da validação apenas do primeiro pacote fragmentado, com os pacotes posteriores pas-

sando pelo filtro sem a devida verificação, resultando em possíveis vazamentos de informações e ataques que tiram proveito dessa fragmentação (seção 4.6.5).

As vantagens do filtro de pacotes são [AVO 98]:

- Baixo *overhead* / alto *throughput*;
- Barato, simples e flexível;
- Bom para o gerenciamento de tráfego;
- Transparente para o usuário;

As desvantagens do filtro de pacotes são [AVO 98]:

- Permite a conexão direta para *hosts* internos de clientes externos;
- Difícil de gerenciar em ambientes complexos;
- Vulnerável a ataques como *spoofing* de endereços, a menos que seja configurada para que seja evitado (apenas *spoofing* de endereços internos);
- Não oferece autenticação do usuário;
- Dificuldade de filtrar conexões que utilizam portas dinâmicas, como o RPC;
- Deixa brechas permanentes abertas no perímetro da rede;

6.3.2 Filtros de Estados

As brechas permanentes nos filtros de pacotes ocorrem quando uma política ou a sua implementação possui erros, o que não é difícil de acontecer em um ambiente tão complexo como o ambiente cooperativo. Para resolver esse problema, foram desenvolvidos os filtros de pacotes dinâmicos (*dynamic packet filtering*), também conhecidos como filtros de inspeção de estados (*stateful inspection*), que tomam as decisões de filtragem baseadas nas informações dos pacotes de dados e da sua tabela de estados. O *firewall* trabalha abrindo as portas para uma série de pacotes de uma sessão, e depois que todos esses pacotes passam pela porta, ela é fechada [AVO 98]. Assim como o filtro de pacotes, o filtro de estados também trabalha na camada de rede da pilha TCP, possuindo portanto um bom desempenho. A diferença quanto ao filtro de pacotes é que a filtragem pode ser baseada em todos os dados do pacote, e não apenas no cabeçalho. O estado das conexões é monitorado a todo instante, permitindo que a ação do *firewall* seja defi-

nida de acordo com o estado de conexões anteriores mantidas em sua tabela de estados. Isso permite também a segurança de conexões UDP, enquanto o bom desempenho permanece [SEC 99-7].

Um *firewall* de inspeção de estados funciona da seguinte maneira [SPI 99]: quando um pacote SYN inicia uma conexão TCP, ele é comparado com as regras do *firewall*, na ordem seqüencial da tabela de regras, como em um filtro de pacotes. Se o pacote passa por todas as regras sem ser aceito, então o pacote é descartado. A conexão é assim rejeitada (RST é enviado de volta ao *host*). Caso o pacote seja aceito, a sessão entra na tabela de estados do *firewall*, que está na memória do *kernel*. Isso pode ser verificado na figura 6.1. Para os demais pacotes, se a sessão está na tabela, e o pacote é parte dessa sessão, então o pacote é aceito. Se os próximos pacotes não fazem parte de nenhuma sessão presente na tabela de estados, então eles são descartados. Isso pode ser verificado na figura 6.2. O desempenho do sistema melhora, pois apenas os pacotes SYN são comparados com a tabela de regras do filtro de pacotes, e todos os outros pacotes restantes são comparados com a tabela de estados, que fica no *kernel*, tornando o processo mais rápido.

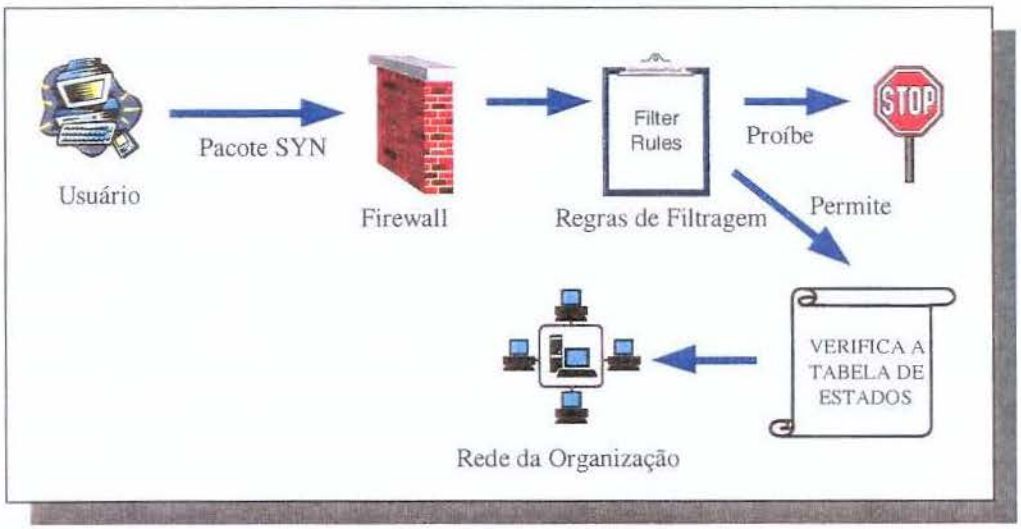


Figura 6.1: Filtro de estados trabalhando na chegada de pacotes SYN.

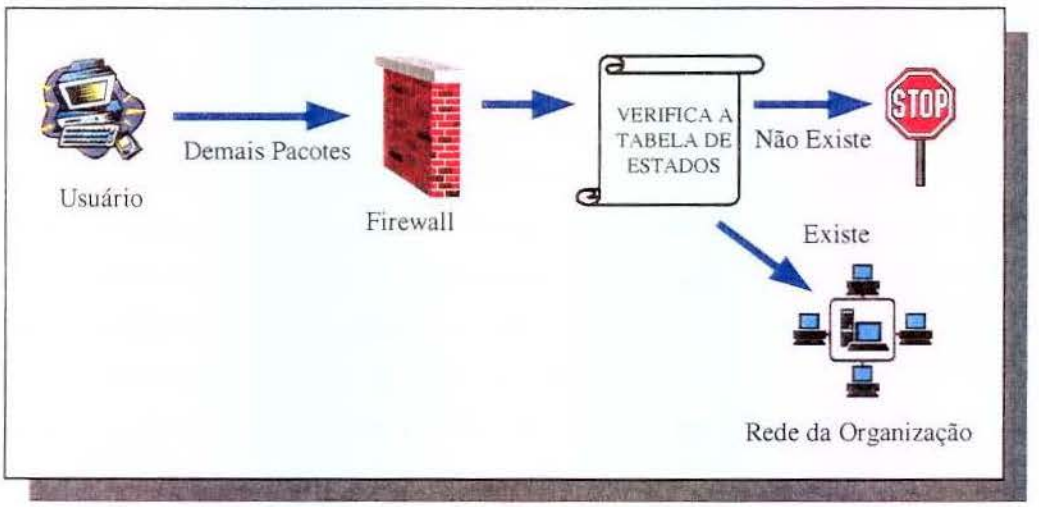


Figura 6.2: Filtro de estados trabalhando na chegada dos demais pacotes.

Um processo diferente de verificação ocorre com pacotes ACK, como pode ser visto na figura 6.3. Quando um desses pacotes chega ao *firewall*, ele primeiramente é comparado com a tabela de estados. Caso não exista nenhuma sessão aberta para esse pacote, então ele passa a ser analisado de acordo com a tabela de regras do *firewall*. Caso o pacote seja aceito de acordo com a tabela de regras, então ele passa pelo *firewall*, e passa a ter uma sessão aberta na tabela de estados. Assim os outros pacotes são verificados de acordo com essa entrada na tabela de estados, e passam pelo *firewall* sem a necessidade de comparação com a tabela de regras do *firewall* [SPI 99].

Um *time-out* de 60 segundos é utilizado quando o pacote SYN é aceito pelo *firewall*. Após a primeira resposta, o *time-out* passa para 3600 segundos, de forma a dificultar ataques de *SYN flooding* (seção 4.6.2). Porém, o *firewall* não se preocupa com o tipo de pacote da resposta, principalmente com relação ao número de seqüência dos pacotes. O Firewall-1, por exemplo, se preocupa apenas com o endereço IP e a porta. Pacotes FIN, RST ou Xmas não iniciam conexões, portanto não são inseridos na tabela de estados. Ataques DoS são prevenidos através da mudança do *time-out* de 3600 para 50 segundos quando o módulo de inspeção recebe um pacote FIN ou RST [SPI 99].

Quanto à inspeção de pacotes UDP, que não utilizam o conceito de conexão, e portanto não possuem distinção entre uma requisição e uma resposta, ou à inspeção de pacotes RPC, que uti-

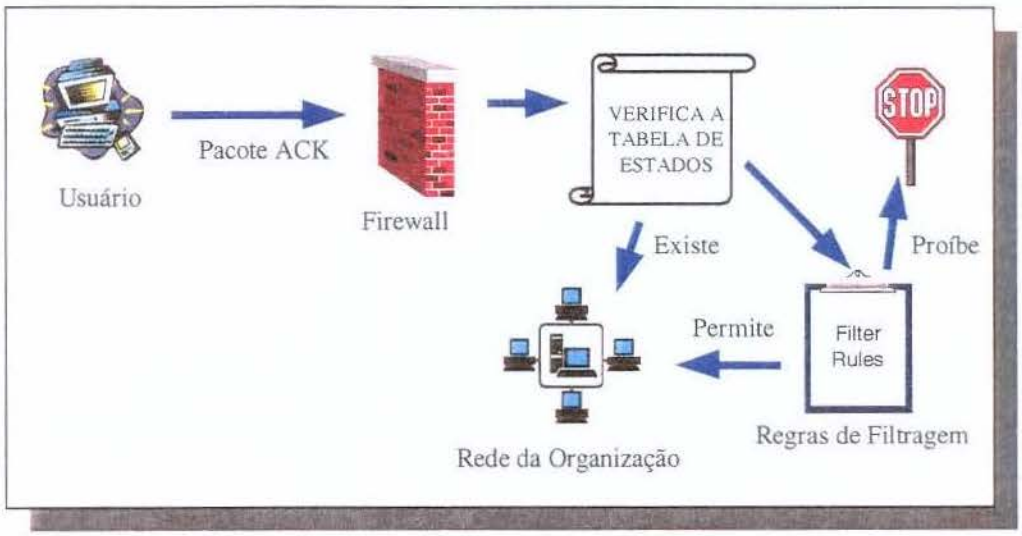


Figura 6.3: Filtro de estados trabalhando na chegada de pacotes ACK.

liza alocação dinâmica de portas, que muda freqüentemente, o filtro de estados armazena dados de contexto, mantendo assim uma conexão virtual das comunicações UDP ou RPC. Assim, quando um pacote tenta entrar na rede, ele é verificado de acordo com a tabela de estados. Caso haja uma entrada na tabela dizendo que a conexão está pendente, o pacote é autorizado.

Teoricamente, o filtro de estados é capaz de examinar através da camada de aplicação, mas na prática é muito difícil executar essa função, já que o *inspection engine*, módulo de verificação dos pacotes do Firewall-1, é inserido entre as camadas 2 e 3 da pilha TCP. Para controlar comandos FTP, por exemplo, seria necessário saber a quantidade de pacotes, guardar esses pacotes, saber por quanto tempo guardá-los até que o último pacote do comando seja recebido. Seria necessário se preocupar também com a seqüência de pacotes, fragmentação e fragmentos *overlapping*, que não é uma tarefa simples e trivial. Porém, algumas implementações adicionam essa funcionalidade.

Exemplos de filtros de estados são o *Context-Based Access Control* (CBAC) da *Cisco IOS Firewall* [CIS 98-2], e o *Inspect Engine* da Checkpoint (Firewall-1) [CHE 98].

As vantagens do filtro de estados são [AVO 98]:

- Aberturas apenas temporárias no perímetro da rede;
- Baixo *overhead* / alto *throughput*;

- Suporta quase todos os tipos de serviços.

As desvantagens do filtro de estados são [AVO 98]:

- Permite a conexão direta para *hosts* internos de clientes externos;
- Não oferece autenticação do usuário, a não ser via *gateway* de aplicação (*application gateway*);

6.3.3 Proxy

O *proxy* funciona através de *relays* de conexões TCP, ou seja, o usuário se conecta a uma porta TCP no *firewall*, que então faz a conexão com o mundo exterior. O *proxy* pode trabalhar tanto na camada de sessão ou de transporte (*circuit level gateway*) quanto na camada de aplicação (*application level gateway*), o que dá um maior controle sobre a interação entre o cliente e o servidor externo. A conexão direta entre um usuário interno e o servidor externo não é permitida através dessa tecnologia, e o re-endereçamento do tráfego, ao fazer com que tráfego pareça que tenha origem no *proxy*, mascara o endereço do *host* interno, garantindo assim uma maior segurança da rede interna da organização.

O servidor *proxy* funciona como um *daemon*, e não utiliza um mecanismo geral de controle de tráfego, mas sim um código especial para cada serviço a ser suportado. O código de um *proxy* não é considerado complexo o suficiente para que possa conter erros que possam ser explorados em ataques [RAE 97]. Uma das grandes vantagens dos *proxies* é a possibilidade de se registrar todo o tráfego, seja ele com origem interna ou externa, podendo assim ativar um sistema de alarme quando um tráfego não apropriado estiver em andamento.

As diferenças entre o *circuit-level gateway* e o *application-level gateway* estão, além da camada TCP em que atuam, também no mecanismo de segurança utilizado. O primeiro funciona apenas como *relay* entre o cliente e o servidor externo, porém sem realizar a verificação dos serviços. Isso pode causar um problema de segurança: se um outro serviço utiliza a porta 80, que é o padrão HTTP, o *circuit-level gateway* não saberá diferenciar esses pacotes, permitindo que eles passem pelo *proxy*. Já o *application-level gateway*, ao trabalhar na camada de aplicação, permite que o *payload* dos pacotes sejam filtrados, como é o caso das filtragens que ocorrem em *tags* HTML feitas pelo *proxy* HTTP.

O esquema de funcionamento típico dos *proxies* é que o cliente deve primeiro se conectar ao servidor *proxy*, e em seguida ser autenticado pelo *firewall*. Após a autenticação o cliente envia a sua requisição ao *proxy*, que a retransmite ao servidor. A resposta do servidor externo passa também pelo *proxy*, com o *proxy* funcionando assim como um *gateway* entre o cliente e o servidor. Isso protege o cliente e o servidor através do controle de requisição de serviços, proibindo certos eventos no nível de aplicação (no *application-level gateway*), tais como o *FTP PUT* e o *FTP GET*.

A necessidade de modificar as aplicações clientes para a interação com o *proxy* vem diminuindo com o avanço da tecnologia [SKO 98], como pode ser observado no *proxy* transparente (seção 6.3.3.1). Porém, a escalabilidade ainda constitui um problema, devido à necessidade de um *proxy* diferente para cada aplicação.

As vantagens do *proxy* são [AVO 98]:

- Não permite conexões diretas entre *hosts* internos e *hosts* externos;
- Suporta autenticação a nível de usuário;
- Analisa comandos da aplicação no *payload* dos pacotes de dados, ao contrário do filtro de pacotes;
- Permite criar *logs* do tráfego e de atividades específicas.

As desvantagens do *proxy* são [AVO 98]:

- É mais lento do que os filtros de pacotes (somente o *application-level gateway*);
- Requer um *proxy* específico para cada aplicação;
- Não suporta todas os tipos de conexões possíveis;
- Requer que os clientes internos saibam sobre ele (vem mudando com o *proxy* transparente (seção 6.3.3.1)).

6.3.3.1 Proxy Transparente

O *proxy* transparente é um servidor *proxy* modificado, que exige mudanças na camada de aplicação e também no *kernel* do *firewall*. Esse tipo de *proxy* redireciona as sessões que passam pelo *firewall* para um servidor *proxy* local de modo transparente, eliminando assim a necessidade de modificações no lado cliente ou na interface do usuário [TRA 98]. Os clientes (software

e usuário) não necessitam saber que as suas sessões são manipuladas por um *proxy*, de modo que as suas conexões são transparentes, como se elas fossem diretas para o servidor.

Como os *proxies* transparentes trabalham baseando-se em portas, eles funcionam apenas para tráfegos TCP e UDP [TRA 98].

Um exemplo pode ser visto no Linux, onde o redirecionamento dos pacotes para o *proxy* transparente é manipulado pelo *firewall*:

```
ipfwadm -I -a accept -r 2323 -P tcp -S 192.168.37.0/24 -D any/0 telnet
```

Esse comando redireciona (opção -r) todas as sessões Telnet originando da rede 192.168.37.0 para o servidor Telnet local, que é o *proxy*, que está escutando a porta 2323. Se nenhuma porta ou a porta 0 é especificada com a opção -r, a porta a ser utilizada é a mesma do destino original.

No exemplo a seguir, todas as sessões referentes aos protocolos especificados serão redirecionadas para um servidor no *host* local, utilizando a porta original.

```
ipfwadm -I -a accept -r -P tcp -S 192.168.37.0/24 -D any/0 smtp www
gopher z3950
```

Os detalhes dos modo de funcionamento, com a especificação de chamadas a sistemas, para sessões TCP e UDP, que são tratadas de maneiras diferentes, podem ser vistas em [TRA 98]. Através desses detalhes é possível verificar que é relativamente simples modificar um servidor *proxy* existente para que ele seja transparente, com o servidor local atuando como um simples redirecionador de pacotes.

6.3.4 Firewalls Híbridos

Os *firewalls* híbridos misturam os elementos das 3 tecnologias apresentadas anteriormente, de modo a garantir para os serviços que exigem auto grau de segurança a proteção dos *proxies*, e para os serviços onde o desempenho é o mais importante, a segurança do filtro de pacotes ou de estados. Assim, os serviços melhores manipulados pelos filtros de pacotes, como o Telnet, utilizam o filtro de pacotes, enquanto os serviços que necessitam de filtragem mais a nível de aplicação, como o FTP, utilizam o *proxy* [SKO 98]. Atualmente, a maioria dos *firewalls* são híbridos, aproveitando as melhores características dos filtros de pacotes, filtros de estados e *proxies*, para cada um dos serviços específicos.

6.3.5 Proxies Adaptativos

A diferença entre o *firewall* híbrido e o *proxy* adaptativo é que o primeiro utiliza os mecanismos de segurança em paralelo, que não representa um aumento no nível de segurança, apenas traz maior flexibilidade, ao permitir a utilização de filtros de pacotes, filtros de estados e *proxies* para serviços específicos. Já o *proxy* adaptativo (*adaptative proxy*) utiliza mecanismos de segurança em série, que traz benefícios para o nível de segurança da rede da organização [AVO 99].

A arquitetura de *proxy* adaptativo possui duas características que não são encontradas em outros tipos de *firewalls* [WES 98]:

- Monitoramento bi-direcional e API de controle entre o *proxy* adaptativo e o *Dynamic Packet Filter* (DPF) ou *Stateful Inspection*;
- Controle dos pacotes que passam pelo *proxy*, com a habilidade de dividir o processamento do controle e dos dados entre a camada de aplicação (*application-level gateway*) e a camada de rede (filtro de pacotes ou de estados).

O *proxy* adaptativo direciona o controle dos pacotes de acordo com as regras por ele definidas. Caso determinados pacotes necessitem de maior segurança, esse fluxo de pacotes é direcionado para o *proxy* de aplicação, que realiza um controle no nível de aplicação. Caso determinados pacotes necessitem de maior desempenho, o *proxy* adaptativo direciona esse fluxo para o filtro de pacotes, que é bem mais rápido do que os *proxies*. Um exemplo dos benefícios trazidos pelo *proxy* adaptativo pode ser visto no FTP.

O FTP utiliza duas conexões, uma para o tráfego de controle e outra para a transferência dos dados. A conexão de controle, que envia comandos FTP, é processada na camada de aplicação pelo *proxy* adaptativo, de modo que ele pode decidir quais comandos são permitidos ou proibidos. Já quando o *proxy* encontra pacotes da conexão de dados, as informações da política de segurança são utilizadas para decidir se esses pacotes devem ser filtrados pelo filtro de pacotes. A API do DPF ainda pode ser utilizada para aplicar regras de filtragem no filtro de pacotes, sem a necessidade de enviar esse fluxo de dados para a camada de aplicação, quando isso não é necessário.

Assim, a arquitetura do *proxy* adaptativo combina eficientemente o auto grau de segurança do controle na camada de aplicação (*proxies*) e o desempenho do processamento na camada de rede (filtro de pacotes), para realizar a filtragem em um mesmo protocolo, como o FTP. O filtro

de estados não permite o controle de comandos FTP individuais, pois o controle é feito no nível de pacotes.

A API bi-direcional permite que o *proxy* gerencie as duas conexões de modo que se a conexão de controle é encerrada, a DPF API é utilizada para encerrar também a conexão de dados. Se a conexão de dados termina antes, o *proxy* é notificado via DPF API para que o tráfego da conexão de controle seja reiniciado [WES 98].

Um exemplo de um *firewall* adaptativo é o Gauntlet, da Network Associates Inc.

6.3.6 Firewalls Reativos

Um próximo passo da evolução dos *firewalls* envolve o seu papel dentro do esquema de segurança. Os *firewalls* são primariamente designados para a prevenção, porém alguns fabricantes já chamam seus *firewalls* que possuem integração com sistema de detecção de intrusões (*Intrusion Detection System* (IDS), que serão vistos no próximo capítulo) e sistemas de respostas, de *firewalls* reativos.

Os *firewalls* reativos incluem funções de detecção de intrusões e alarmes, de modo que a segurança seja mais ativa do que passiva. Com a adição dessas funções, o *firewall* pode policiar acessos e serviços, além de ser possível mudar a sua configuração de modo dinâmico, enviar mensagens aos usuários e ativar alarmes [AVO 99].

De fato, um sistema de detecção de intrusões é um componente importante dentro de um sistema de segurança, como será discutido no próximo capítulo.

6.3.7 Firewalls Individuais

Uma tendência que pode ser observada é que cada vez mais as organizações precisarão, além do controle da rede, também do controle dos *hosts*. Um problema que pode ser observado na discussão dos aspectos de segurança do cliente VPN [NAK 00] é que, com os usuários passando a serem remotos, um *firewall* atuando na borda da rede passa a não ser mais suficiente para garantir a segurança da organização. Um *hacker* poderia, por exemplo, atacar um *host* cliente, e utilizá-la como uma ponte entre a Internet e a rede interna da organização, como pode ser vista na figura 6.4.

Assim, o advento da computação móvel e remota, além da computação sem fio, resultou na possibilidade de conexão à rede interna da organização a partir de qualquer lugar a qualquer

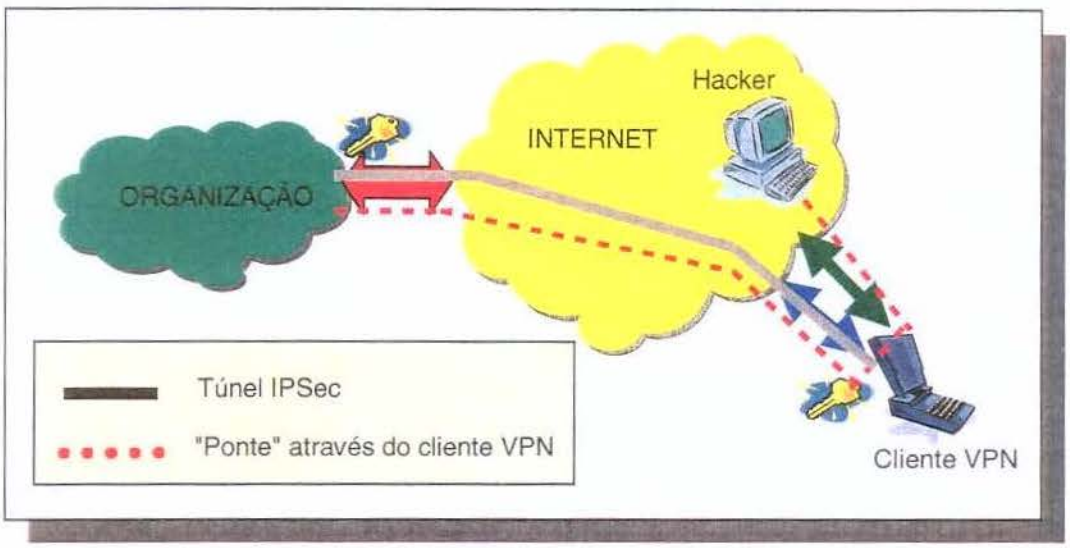


Figura 6.4: Um *hacker* pode acessar a rede da organização através do cliente VPN.

momento, através da VPN. Isso porém trouxe as implicações de segurança também para o *host* do cliente, que portanto deve ser protegido de maneira adequada. Com isso, criou-se um contexto onde uma política de segurança pode não ser suficiente ou praticamente impossível de ser implementada – um equipamento que está dentro da organização pode ser controlada, porém controlar um *notebook* ou um equipamento na casa de um funcionário passa a ser mais complicado, de modo que esses próprios equipamentos agora já necessitam de uma proteção para que a rede da organização não seja comprometida.

Um *firewall* individual ou *firewall* pessoal é uma das alternativas para a proteção das conexões de *hosts* individuais, e ele é um *firewall* que atua não na borda da rede da organização, mas no próprio equipamento do usuário. Alguns produtos já estão no mercado, como o ConSeal, desenvolvido pela Signal 9 [SIG 99], que atua na camada de enlace de dados e filtra pacotes IP (TCP, UDP, ICMP), NetBEUI, IPX, ARP, etc.

Uma análise no ConSeal mostra que através dele é possível controlar acessos aos recursos, monitorar todo o tráfego gerado ou que chega ao sistema, gerar regras de acordo com uma aplicação específica que está funcionando e criar *logs* de todos os acessos do sistema [SIG 99].

É possível criar regras de acordo com as seguintes características [SIG 99]:

- Quando um aplicativo específico está funcionando;

- De acordo com determinado dispositivo Ethernet ou serial;
- Quando um número de telefone específico é utilizado;
- Para serviços, arquivos ou compartilhamentos específicos;
- Para endereços IPs específicos;
- Para a direção de fluxo dos pacotes;
- Para um usuário específico;
- Para conexões VPN ou conexões discadas.

Assim, através de um *firewall* individual é possível obter uma proteção das conexões do cliente, de modo que uma política que poderia ser seguida em um ambiente cooperativo seria a de permitir somente as conexões com a rede da organização dos clientes que já estivessem protegidos por um *firewall* individual.

Porém, não se deve esquecer que um vírus sempre pode reescrever essas regras, mesmo que isso exija um trabalho extra para o atacante. Além disso, basta que a solução fique conhecida para que ela passe a se tornar alvos dos atacantes. É importante, portanto, considerar o *firewall* individual apenas como um incremento no nível de segurança de uma organização, não sendo ele suficiente para a garantia da segurança da rede.

6.3.8 A Melhor Tecnologia de Firewall

A evolução natural dos *firewalls* pôde ser observada através das seções anteriores. Hoje já não é possível utilizar somente uma tecnologia de *firewall*, tais como somente *proxies* ou somente filtros de estados. Foi visto que os filtros de estados possuem desempenho semelhante ao filtro de pacotes, com o aumento do nível de segurança, de modo que um filtro de pacotes, puro e simples, praticamente já não é mais utilizado, a não ser nos roteadores. O *proxy* é importante para garantir a segurança em serviços na camada de aplicação, como é o caso do HTTP, onde a filtragem de algumas *tags* HTML são importantes para a manutenção da segurança exigida pela organização. Esse mesmo *proxy* HTTP pode realizar filtragem de URLs, enquanto um *proxy* FTP pode realizar filtragem de comandos do protocolo, como o PORT. Assim, ainda existe a necessidade de utilizar filtros de estados para as conexões que exigem maior desempenho, enquanto *proxies* são necessários para as conexões mais complexas e que exigem maior grau de segurança

e controle. Isso explica o fato da grande maioria dos *firewalls* hoje se encaixarem no perfil de *firewalls* híbridos.

Os *proxies* adaptativos fazem um controle ainda maior, ao separar o tipo de filtragem dentro de um mesmo protocolo, como é o caso do FTP, que utiliza o *proxy* para o seu canal de controle, e o filtro de estados para o seu canal de dados. Já os *firewalls* reativos fazem parte de uma evolução natural, onde um sistema de detecção e resposta a eventos de segurança são importantes, mas que cujo os mesmos resultados podem ser obtidos através de um sistema de detecção de intrusões.

Assim, a melhor tecnologia a ser utilizada por uma organização é uma questão relativa, já que tudo deve ser analisado de acordo com o ambiente onde o *firewall* deverá funcionar. Caso uma organização tenha como objetivo apenas liberar o acesso Web para seus usuários internos, um *proxy* seria mais do que suficiente, além de um filtro de pacotes para bloquear os pacotes que não fossem relativos ao HTTP (porta 80). De nada adiantaria, por exemplo, a instalação de um *proxy* adaptativo. Tudo deve ser analisado de acordo com as necessidades da organização. A melhor tecnologia é, sem dúvida, aquela que melhor se adequa às necessidades da organização, levando-se em consideração o grau de segurança requerido e a disponibilidade de recursos (técnicos e financeiros) para a sua implantação.

6.4 As Arquiteturas

A arquitetura de um *firewall* deve ser desenvolvida de acordo com as necessidades da organização, utilizando-se os componentes e as funcionalidades descritos na seção 6.2 e as tecnologias discutidas na seção 6.3.

O estabelecimento de uma rede desmilitarizada (DMZ) é essencial, já que permite que serviços sejam providos para os usuários externos (através de *bastion hosts*), ao mesmo tempo em que protegem a rede interna dos acessos externos, ou seja, os acessos externos ficam confinados nessa rede desmilitarizada. Como os servidores localizados na DMZ possuem acesso direto externo, eles devem ser configurados de modo a funcionar com o mínimo suficiente de recursos possíveis para que o serviço determinado seja provido. Esse servidor, com todas as funcionalidades desnecessárias eliminadas, é conhecido como *bastion host*, e é normalmente o alvo dos ataques externos, já que os usuários possuem acesso somente para os recursos localizados na DMZ. Caso um desses servidores da DMZ seja comprometido em um ataque, a rede interna da

organização ainda estará protegida. Porém, para que isso seja verdade, a política de segurança e a sua implementação devem estar totalmente de acordo com o estabelecido, principalmente porque em um ambiente cooperativo essa política é complexa o suficiente para que erros sejam possíveis de serem cometidos. Para facilitar o entendimento, o desenvolvimento, o gerenciamento, a implementação e a atualização dessa política foi sugerida um modelo onde a segurança da rede da organização é dividida em 5 níveis hierárquicos de defesa, que serão discutidos no capítulo 12.

As clássicas arquiteturas do *firewall* apresentadas por Chapman [CHA 95] são as três descritas a seguir, sendo que a quarta arquitetura é a do *firewall* cooperativo, que é proposto neste trabalho.

6.4.1 Dual-Homed Host Architecture

É a arquitetura formada por um equipamento que possui duas interfaces de rede (figura 6.5), que funciona como um separador entre as duas redes. Os sistemas internos têm que se conectar ao *firewall* para que possam se comunicar com os servidores externos, e vice-versa, mas nunca diretamente. Assim, as comunicações são realizadas através de *proxies* ou quando o usuário se conecta anteriormente no *host dual-homed*, para depois se conectar ao servidor externo. Essa última abordagem causa problemas, principalmente para o usuário, já que o processo de acesso externo não é transparente, além de ser mais improdutivo.

6.4.2 Screened Host Architecture

Essa arquitetura (figura 6.6) é formada por um filtro de pacotes e um *bastion host*. O filtro deve possuir regras que permitam o tráfego para a rede interna somente através do *bastion host*, de modo que os usuários externos que desejem acessar um *host* da rede interna devem primeiramente se conectar ao *bastion host*. O *bastion host* pode funcionar também como um *proxy*, exigindo assim que os usuários internos acessem a Internet através dele. Outra possibilidade de usuários internos acessarem serviços externos é através de regras no filtro de pacotes. Essas duas possibilidades podem também ser mescladas, resultando no *firewall* híbrido.

Os problemas que podem ocorrer nessa arquitetura é que, se o *bastion host* é comprometido, então o atacante já está dentro da rede interna da organização. Outro problema é que o filtro

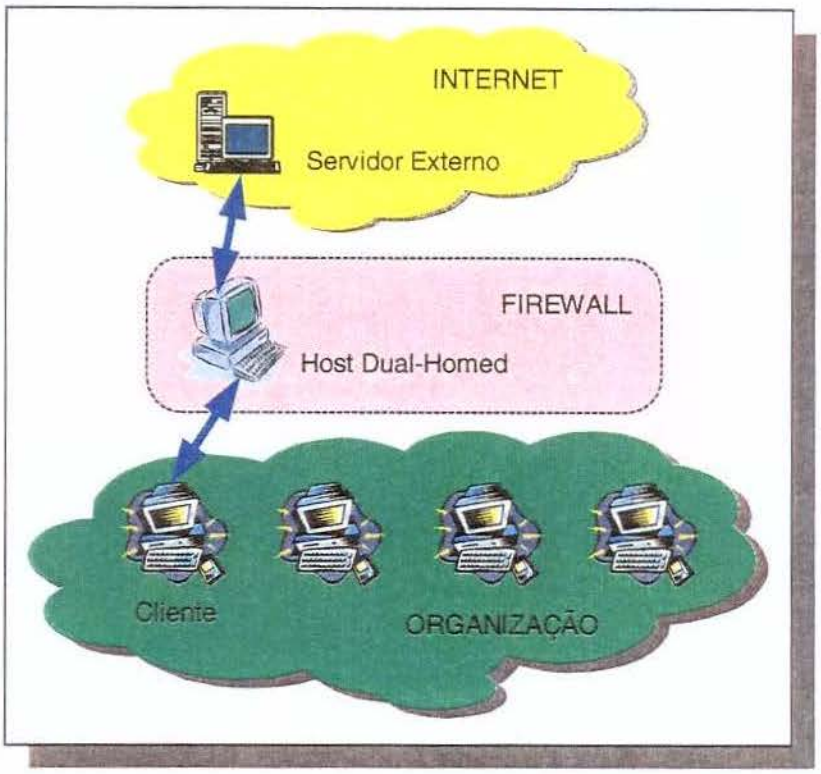


Figura 6.5: A arquitetura *host dual-homed*.

forma um único ponto de falha, de modo que se for atacado, a comunicação da rede da organização com a Internet fica comprometida.

6.4.3 Screened Subnet Architecture

Essa arquitetura (figura 6.7) aumenta o nível de segurança com relação à arquitetura *screened host* ao adicionar a rede DMZ. Se antes um ataque ao *bastion host* significava que o atacante já estaria com a rede interna disponível para ele, isso não ocorre na arquitetura *screened subnet*. O *bastion host* fica na DMZ, que é uma zona de confinamento entre a rede externa e a rede interna, que fica entre dois filtros. A DMZ evita que um ataque ao *bastion host* resulte, por exemplo, na utilização de um *sniffer* para a captura de pacotes de usuários internos.

Um ponto importante da arquitetura é a definição dos filtros internos e externos. Qualquer falha em sua definição ou implementação pode resultar em uma falsa sensação de segurança.

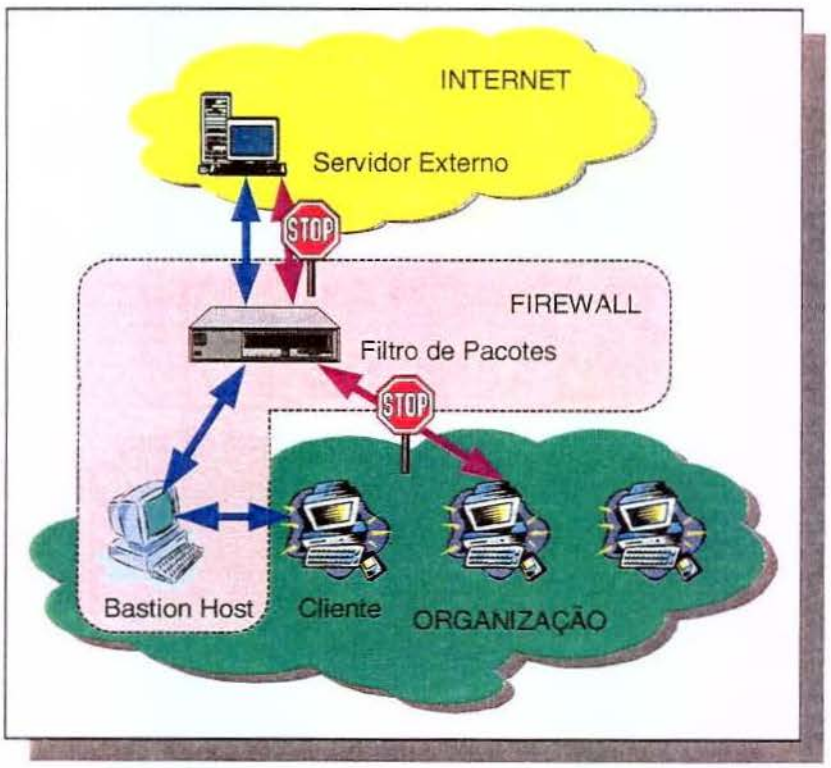


Figura 6.6: A arquitetura *screened host*.

O filtro externo deve permitir o tráfego dos serviços disponíveis na DMZ, bem como o tráfego das requisições dos usuários internos. Já o filtro interno deve permitir somente a passagem das requisições e respostas do serviços permitidos para os usuários internos. Permitir o tráfego do *bastion host* para a rede interna poderia comprometer a segurança da rede interna caso ele seja comprometido.

Uma variação bastante comum dessa arquitetura é a utilização de um equipamento com três interfaces de rede, um para a rede externa, outra para a rede interna, e a terceira para a rede DMZ (figura 6.8). Os filtros funcionariam em cada interface, sendo portanto conceitualmente uma arquitetura *screened subnet*.

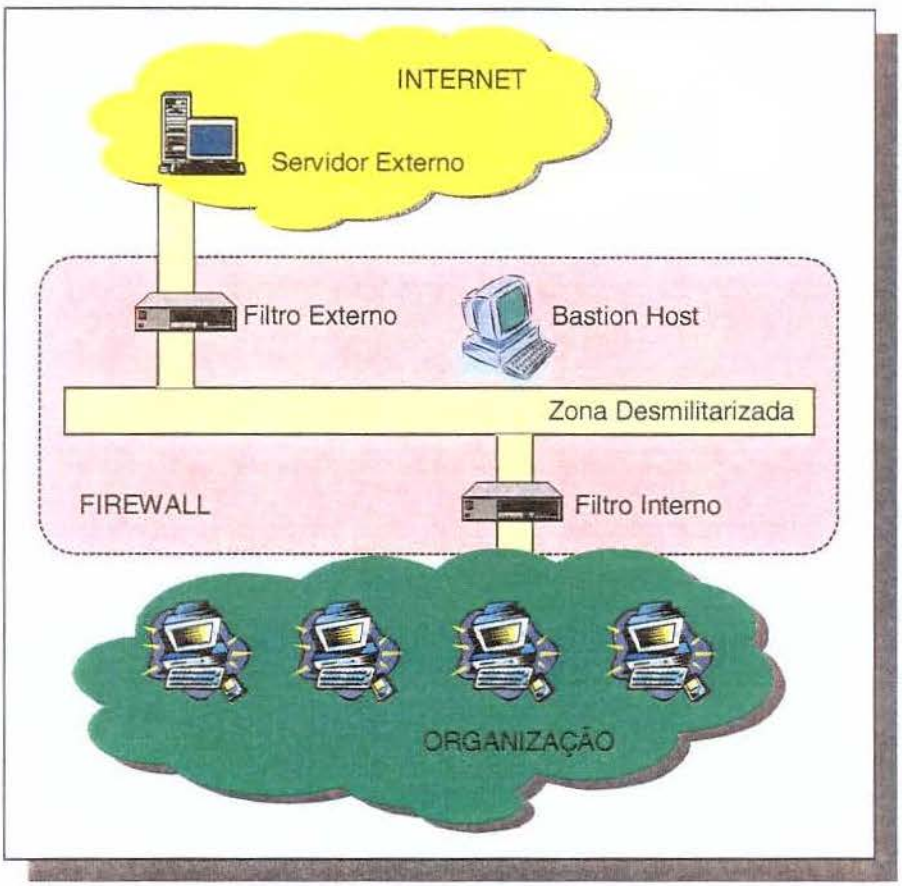


Figura 6.7: A arquitetura *screened subnet*.

6.4.4 Firewall Cooperativo

O *firewall* cooperativo é a arquitetura proposta neste trabalho, onde os novos componentes como a VPN, o NAT, o IDS e a PKI são inseridos. Essa arquitetura será descrita e discutida no capítulo 12.

6.5 O Desempenho

Como o *firewall* é o responsável pela análise de todos os pacotes que passam pelas conexões da rede, é imprescindível que ele possua um desempenho satisfatório, para que ele não se torne um gargalo para a rede. Testes realizados em [NEW 99] mostram que em 1999 os *firewalls* melho-

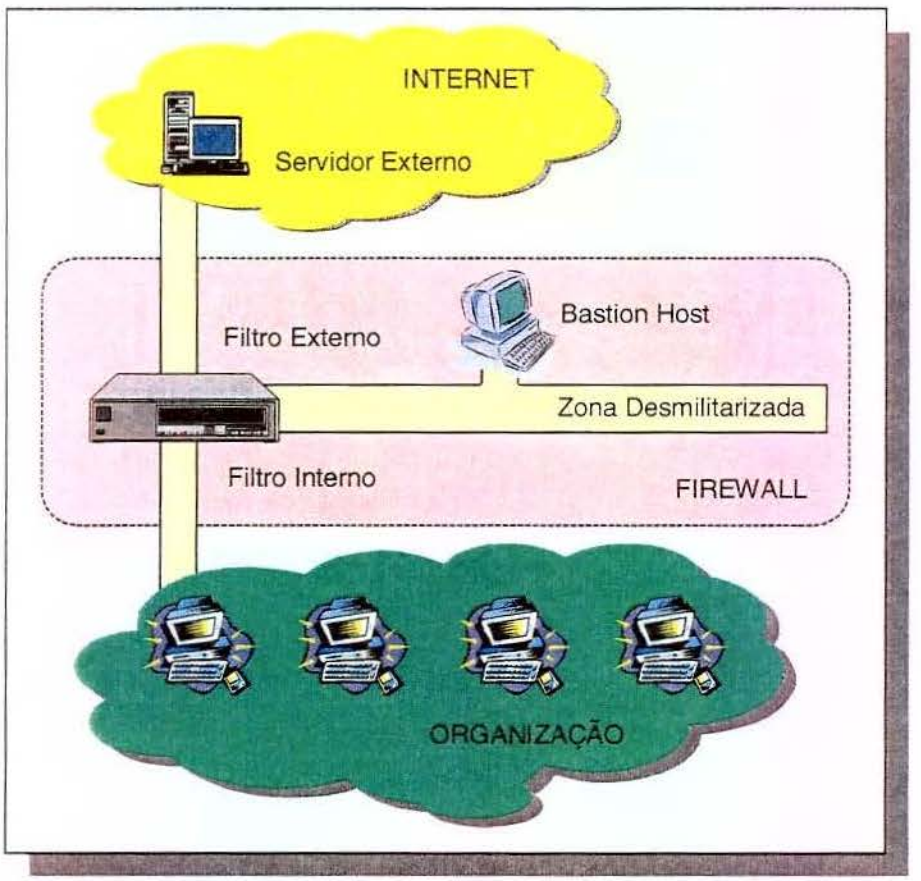


Figura 6.8: Uma variação da arquitetura *screened subnet*.

raram o seu desempenho em 30% se comparado com os testes de 1998, e 300% se comparado com os testes de 1997, mostrando uma evolução natural.

O desempenho é essencial em um ambiente cooperativo, pois a complexidade das conexões, com o grande conjunto de regras e o grande número de conexões concorrentes, exige um grande poder de processamento para a análise rápida de todos os pacotes das conexões.

O desempenho de um *firewall* depende de alguns fatores:

- Hardware:

- Velocidade da placa de rede;
- Número de placas de rede;
- Tipo de barramento (PCI, etc);

- Velocidade da CPU;
- Quantidade de memória.
  - Sofware:
- Código do *firewall*;
- Sistema operacional;
- Pilha TCP/IP;
- Quantidade de processos que estiver sendo utilizado pela máquina;
- Configuração, como a complexidade das regras de filtragem;
- Tipo de *firewall*: *proxy* ou filtro de estados? No *proxy* a CPU é o mais importante, pois cada pacote tem que ser desmontado, analisado e remontado. No filtro de estados a RAM é o mais importante, pois as informações sobre os estados precisam estar disponíveis na RAM para uma maior rapidez nas respostas.

Um ponto importante a ser considerado é que poucos *firewalls* possuem *throughput* de 100 Mbps, o que pode causar gargalos quando utilizados como *firewalls* internos. O gerenciamento de banda vem se tornando um dos fatores principais para assegurar o sucesso da utilização da rede como uma ferramenta de negócios das organizações, e alguns dos *firewalls* também oferecem essa capacidade [SEC 99-7].

A necessidade crescente de poder de processamento cada vez maior nos *firewalls* fez surgir uma tendência de utilização de equipamentos conhecidos como *firewall appliances*, que serão discutidos na seção a seguir.

6.6 O Mercado

Skoudis analisa em [SKO 98] o amadurecimento do mercado de *firewalls*, que no início era formado por simples filtros nos *gateways*, sendo que até recentemente existiam diversas pequenas empresas que comercializavam seus produtos sem a tecnologia e sem os testes adequados e necessários. A demanda crescente fez com que os grandes fabricantes também entrassem no mercado, de modo que isso resultou em custos cada vez menores. Alguns pequenos fabricantes também entraram no mercado com produtos considerados eficientes.

Uma tendência de mercado, que segue a necessidade de maior desempenho e facilidade de gerenciamento, é a utilização de *firewall appliances*, que são produtos que já vêm pré-instalados com o hardware. [AVO 99] divide esses produtos em 3 categorias, que são:

- *Large Enterprises* (mais de 1000 usuários) – Secure Computing's Sidewinder, Lucent's Managed Firewall; Cisco System's PIX. São os *firewalls* "clássicos", que se tornaram fáceis de gerenciar, porém necessitam de um profissional de segurança dedicado para a sua manutenção;

- *Small Enterprises* (entre 50 e 1000 usuários) – Technologic's Interceptor, Watchguard's Firebox, Internet Devices' Ft. Knox Policy Router, NetScreen's NetScreen-10, Sonic System's SonicWall. Tipicamente *plug'n play*, esses produtos possuem poucas opções de configuração, e não permitem que o sistema operacional seja modificado. A filosofia é a de que poucas escolhas resultam em uma melhor segurança para aqueles com pouco conhecimento. As características dos produtos podem variar bastante, como a adição de *Web caching*, filtragem de conteúdo, gerenciamento de tráfego, *scanning* de vírus e até mesmo uma função onde os *patches* e avisos sobre segurança são enviados automaticamente para o administrador de segurança. Essa categoria de *firewall appliances* são indicados para aqueles que possuem pouco conhecimento técnico, devido à sua facilidade de gerenciamento;

- *Small Office Home Office* (SOHO) (entre 5 e 50 usuários) – eSoft's IPAD, Freegate's OneGate 150, Whistle Communication's InterJet. Múltiplos serviços integrados, como *firewall*, servidor Web e servidor de *e-mail*, facilitam o gerenciamento, e são destinados para as organizações com poucos recursos técnicos para a administração da segurança. Como esses produtos combinam diversas funcionalidades, é importante saber qual a definição de "*firewall*" dos fabricantes, como qual tecnologia é empregada pelo *firewall*. Essa integração entre diversos serviços pode trazer problemas, devido ao aumento das funcionalidades, como foi visto na seção 3.8. Além disso, existem os problemas com a robustez dos *logs* de segurança e com os relatórios.

É preciso tomar cuidado com relação aos diversos produtos que estão no mercado, já que a impressão que se tem é de que os fabricantes estão aproveitando a força do termo "*firewall*" e vendendo produtos como se eles fossem a solução para todos os problemas de segurança das organizações.

A grande afirmação que fica é que essa complexidade que vem sendo adicionada aos *firewalls* traz consigo uma dificuldade com relação à confiança na real segurança desses produ-

tos. Permitir um novo serviço através do *firewall* é fácil, o difícil é fazer isso mantendo o mesmo nível de segurança. Como foi visto na seção 3.8, a segurança é inversamente proporcional às suas funcionalidades, e portanto essas adições devem ser evitadas ao máximo, já que aumentam a probabilidade de vulnerabilidades, que causam a diminuição do grau de segurança da rede. O mais recomendado é que o *firewall* seja o responsável apenas pela segurança de borda da organização, com os demais serviços sendo oferecidos na rede DMZ.

6.7 A Avaliação do Firewall

A realidade é que não é o produto que irá garantir a segurança necessária, mas sim a política de segurança, e a sua implementação. Assim, o melhor produto para uma organização é aquele que melhor permite a implementação da política de segurança definida, e melhor se ajusta à experiência e capacidade do profissional. A escolha do produto deve ser uma parte efetiva da estratégia de segurança da organização.

Diversos aspectos devem ser analisados e discutidos na escolha do *firewall* mais adequado para a organização. Alguns desses aspectos são:

- Fabricante/fornecedor – Alguns programas de certificações de *firewalls* podem ser consultados para a escolha do produto. Porém, essas certificações ainda são novos e de difícil avaliação e confiabilidade;
- Suporte técnico – Serviços que podem auxiliar na utilização e atualizações do produto;
- Análise do *design* – É importante levar em consideração aspectos de implementação do *firewall*, como por exemplo, como ele manipula *buffers* e previne o *buffer overflow*. *Firewalls* com código aberto permitem uma melhor análise e discussão sobre problemas de implementação. As interações do *firewall* com o sistema operacional e com o hardware também merecem ser considerados;
- Análise de *logs* – Através da sua análise é possível detectar erros e problemas no sistema, além de tornar possível a detecção de tentativas de ataques. A capacidade dos *logs* deve portanto ser suficiente para que essas detecções sejam possíveis. Porém, o fato do *firewall* registrar os eventos mais importantes não assegura a sua efetividade, sendo imprescindível uma ferramenta eficiente de análise dos *logs*. Um outro problema que pode acontecer é que, quando um ataque é detectado através da análise dos *logs*, esse

ataque geralmente já foi realizado, não sendo mais possível impedi-lo. Um sistema de detecção de intrusões, que será visto no próximo capítulo, é assim um ponto essencial para a proteção de uma rede;

- Desempenho – Como foi visto na seção 6.5, o desempenho é importante em um ambiente cooperativo, porém não tem significado algum se a segurança do *firewall* não é garantido. A segurança sempre deve vir em primeiro lugar;
- Gerenciamento – A configuração remota, a criptografia, os avisos em caso de incidentes de segurança, a capacidade de análise de *logs*, a localização de *logs* em outras máquinas, e o que o *firewall* faz quando o espaço para o *log* acaba, são alguns dos pontos que devem ser observados;
- Teste do *firewall* – Os testes no *firewall* são essenciais para determinar a efetividade do que foi implementado na organização. Os aspectos que devem ser analisados, e a maneira e por quem devem ser realizados, serão discutidos na próxima seção.

6.8 Teste do Firewall

Testar um *firewall* significa verificar se uma política de segurança foi bem desenvolvida, foi implementada de modo correto, e se o *firewall* realiza aquilo que ele declara realizar. Tentar passar pelo *firewall* ou driblá-lo é um instrumento valioso para a análise de um *firewall*, além de ser valioso para a própria política de segurança, já que o conjunto de regras implementado pode ser validado, falhas podem ser encontradas, e evoluções podem ser realizadas.

Um teste de *firewall* pode ser estruturado em 4 etapas [RAE 97]:

- Coleta de informações indiretas – informações que podem ser obtidas sem que o *firewall* faça registros ou ative alarmes. São as informações públicas, como por exemplo, de servidores Web, FTP, *whois* ou *nslookup*. A busca por mensagens em *newsgroup* enviadas por funcionários da organização, por exemplo, podem revelar endereços de *e-mail* específicos, como é o caso de joao.teixeira@mars.planet.solar.com, ao invés de joao.teixeira@solar.com;
- Coleta de informações diretas – são as informações que são vigiadas, e que portanto podem ter seus acessos detectados, como por exemplo, a procura por informações adicionais em servidores de nomes. Outro exemplo poderia ser enviar um *e-mail* para um usuário inexistente de uma rede, o que pode revelar a topologia dessa rede através da análise do cabeçalho. Através do *scanning* é possível visualizar a topologia da rede da

organização. Um *scanning* no *firewall* revela as portas dos serviços abertos e os respectivos pontos de ataques. Modos seguros de *scanning* (*stealth*) também podem ser utilizados, como foram discutidos na seção 4.5.4;

- Ataques externos – Através de *hosts* confiáveis, ou através de *IP Spoofing*, que tentam burlar as regras do *firewall*, ou através de um dos métodos descritos no capítulo 4;
- Ataques internos – Esse teste pode ser visto de duas formas: ataques que usuários internos realizam em *hosts* externos, e ataques entre os usuários internos.

Para o auxílio no desenvolvimento dos testes descritos acima, diversas ferramentas para testar vulnerabilidades em *firewalls* podem ser encontrados no mercado e na própria Internet. São ferramentas que podem ser utilizadas para realizar a auditoria do próprio *firewall*, além de ser possível utilizá-las também em serviços como Web, FTP, SMTP.

Outro ponto importante é definir quem irá realizar os testes: o revendedor, *hackers*, os próprios funcionários, ou uma empresa especializada. Cada um possui suas vantagens e desvantagens. Os revendedores possuem o conhecimento sobre o seu próprio produto, e detalhes do funcionamento podem ser esclarecidos, porém os testes realizados podem ser imparciais. Um *hacker* pode analisar de forma produtiva as vulnerabilidades da política de segurança implementada, porém o risco é que, se não houver ética, esse *hacker* pode esconder algumas vulnerabilidades encontradas, e dividi-las com seus colegas *hackers*. Os próprios funcionários parecem ser a melhor opção, porém o que falta é o *know-how* de como realizar esses testes, o que pode acabar comprometendo os testes. A empresa especializada pode ter o *know-how* necessário, porém pode sair caro para a organização, já que a segurança é um processo constante e dinâmico, e diversas análises serão necessárias.

6.9 Problemas Relacionados

Os *firewalls* são essenciais dentro do sistema de segurança de uma organização, porém a falta de alguns cuidados podem tornar todos os esforços inválidos, e instaurar um perigoso falso senso de segurança. Os problemas mais comuns encontrados que resultam no perigo são:

- Instalações de *firewalls* mal configurados;
- Implementação incorreta da política de segurança;

- Gerenciamento pobre;
- Falta de atualizações.

Mesmo tomando-se medidas contra os problemas relacionados acima a vigilância é importante, já que um *firewall* geralmente leva à falsa sensação de segurança. Diversos problemas podem ocorrer com falhas no desenvolvimento da própria política de segurança ou com falhas na própria implementação dos *firewalls*.

Essas falhas nas implementações dos *firewalls* já estiveram presentes em diversos *firewalls* comerciais, como é o caso do Cisco PIX Firewall. O *scanner* de segurança ISS 5.6.2 era capaz de "derrubar" o PIX, segundo a lista *firewall-wizards*. Isso é um sério problema, já que pode ser utilizado para ataques DoS e isolar a rede da organização do acesso externo, comprometendo assim os negócios.

Outro caso de possíveis vulnerabilidades podem ser encontradas no Firewall-1 da Check Point, que possuem permissões padrão para *TCP Source Porting* e tráfego de pacotes UDP, o que pode causar problemas em organizações onde o *firewall* é mal administrado e funciona com essas configurações padrões [NEW 99].

Com relação à configuração e gerenciamento do *firewall*, diversos equívocos podem ser cometidos, o que pode comprometer a segurança da organização, tais como [AVO 99]:

- Adicionar novos serviços porque os usuários dizem que "precisam" deles – é importante separar o que eles "precisam" do que eles "querem". Os novos serviços devem ser claras necessidades para os negócios da organização. Mesmo um serviço aparentemente benigno aumenta o trabalho de administração do *firewall*, além de adicionar potenciais possibilidades de ataques;
- Concentrar os esforços no *firewall* enquanto outras medidas de segurança são ignoradas – os *firewalls* não são suficientes, são apenas uma parte do arsenal de segurança necessários;
- Ignorar os arquivos de *logs* – se os arquivos de *logs* não são nunca avaliados, então não possuem nenhum valor;
- Desligar as mensagens de alertas – ao desligar os alarmes e alertas, a segurança do perímetro da rede está em perigo;

- Permitir usuários no *firewall* – os *firewalls* devem ser tão simples quanto possíveis. Como usuários adicionam complexidade, as contas dos usuários são potenciais pontos de ataques, e os próprios usuários são potenciais atacantes. Além disso, qualquer usuário pode abrir brechas de segurança no *firewall* "sem querer", através de seus próprios erros;
- Permitir que diversas pessoas administrem o *firewall* – todo administrador é um atacante em potencial, e um administrador pode causar danos mais sérios do que qualquer outro usuário, devido aos seus direitos no sistema;
- Presença de modens – qualquer modem atrás do *firewall* pode driblar o perímetro de segurança, de modo que forma uma entrada em potencial para a rede da organização;
- Driblar a segurança do *firewall* e usar uma política própria – o *firewall* deve ser configurado de acordo com a política de segurança da organização. Fugir disso, como criar um *backdoor* para facilitar a sua administração, certamente trará muitos problemas futuros;
- Ignorar a existência da política de segurança da rede e dos computadores – se existir a política que trata desses aspectos, ela deve ser utilizada. E se esses aspectos não estiverem sendo tratados, a política deve ser modificada com a revisão para os *firewalls*;
- Não possuir uma política de segurança – sem um conjunto de regras, não há como tomar decisões relativos à segurança. O melhor é que uma política seja criada, mesmo que gradativamente. Essa abordagem deve ser utilizada em oposição à idéia de se criar a política apenas quando ela for efetivamente necessária, depois da organização sofrer um ataque. Já que a política deve ser criada (se não tiver, o *firewall* será mal configurado, e um ataque será inevitável e certo), então o melhor é criá-la antes da implantação do *firewall*. Maiores detalhes da política de segurança podem ser vistos no capítulo 5.

6.10 O Firewall Não é a Solução Total de Segurança

É importante lembrar que o *firewall* é apenas uma parte de um conjunto de componentes de um sistema de segurança necessário para a proteção das organizações. Assim, a idéia de que um *firewall* é a solução para todos os problemas de segurança, disseminada por alguns fabricantes, e que infelizmente ainda faz a cabeça de muitos profissionais de informática, é um conceito equivocado, que acaba colocando em risco toda a organização.

Firewalls podem ser uma "faca de dois gumes": eles representam uma boa primeira linha de defesa e são essencialmente necessários em uma infra-estrutura que envolve a segurança. Porém,

eles tendem a acalmar as organizações com uma falsa e perigosa sensação de segurança e satisfação [DID 98]. Segundo Mark Fabro, da Secure Computing, "*firewalls* são como cercas em volta de uma propriedade: eles guardam o perímetro. Uma falha no sistema operacional, nos serviços ou na aplicação pode fazer com que o *hacker* simplesmente passe pelo *firewall* e tenha acesso aos dados sensitivos".

O maior problema relacionado ao *firewall* pode ser considerado justamente a falsa idéia de que o *firewall* é a solução dos problemas de segurança. A própria definição de *firewall*, visto na seção 6.1, parece não estar mais de acordo com os dias de hoje. A pouco tempo atrás, era fácil definir um *firewall* e suas funções. Ele atuava na borda de uma rede, evitando que os intrusos entrassem na rede da organização. Esse perímetro era facilmente definido, e o *firewall* cuidava desse perímetro. Hoje esse perímetro é intangível, com as extranets e VPNs estendendo as redes para a comunicação com os parceiros, a Web e os banco de dados sendo acessados pelo público em geral, e a computação móvel e a utilização indiscriminada de modens criando pontos de acesso à rede da organização que não passam efetivamente pelo *firewall*. O perímetro hoje está mudando, fluindo e ativo, como pode ser visto no modelo de bolsões de segurança definidos na seção 5.11. Assim, o *firewall* hoje não pode ser considerado um muro, e sim uma defesa ativa, que é a idéia principal do *firewall* cooperativo.

Mesmo a definição desse "muro" criado pelo *firewall* torna-se mais complicado, já que mais de 60% dos ataques vêm da própria rede interna, e com a computação móvel e o ambiente cooperativo, o foco acaba mudando de "muros altos" para "controle dos usuários". O foco de segurança agora está em selecionar os usuários que podem entrar na rede e selecionar os direitos que esses usuários possuem na rede. Definir os recursos que um usuário particular pode acessar, e os níveis de acesso de cada usuário na rede, e a certeza de que eles estão fazendo aquilo que lhes são explicitamente permitidos passa a ser essencial. Assim, o controle de acesso é também um importante aspecto de segurança. Basicamente, não basta apenas controlar, é necessário também monitorar o que o usuário está realizando dentro da rede [SEC 99]. Além disso, a rede não deve ser protegida apenas das invasões intencionais, mas também contra inúmeros erros comuns de usuários autorizados [CIS 98].

Levando-se isso em consideração, onde se pode ver que a definição original do *firewall* não se aplica mais no contexto atual, este trabalho propõe uma arquitetura de *firewall* que visa atender às necessidades das organizações, o *firewall* cooperativo, que será visto no capítulo 12.

6.11 Conclusão

Este capítulo discutiu diversos aspectos do *firewall*, entre eles a sua definição, que parece que vem sendo modificado com o tempo, em grande parte devido ao mercado e à errada percepção de que ele é a solução de todos os problemas de segurança de uma organização. As funcionalidades do *firewall* foram apresentadas, e a evolução que vem ocorrendo nesse importante componente de segurança foi discutida. A arquitetura influi diretamente no nível de segurança, e as diferentes possibilidades foram analisadas, culminando com a proposta do *firewall* cooperativo, que será apresentada no capítulo 12. Foram analisados ainda aspectos como o seu desempenho, o seu mercado, os seus testes, e os problemas relacionados. Apesar de não ser a solução de todos os problemas de segurança, o *firewall* é um componente essencial em uma organização, ao atuar na borda de sua rede, protegendo-a contra ataques e acessos indevidos.

Capítulo 7

Sistema de Detecção de Intrusões

Foi visto no capítulo anterior que os *firewalls* são apenas um dos componentes de um sistema de segurança destinado a proteger a rede da organização. Neste capítulo será discutido qual o objetivo dos sistemas de detecção de intrusões (IDS), outro componente importante para a segurança. Os tipos de IDS e as metodologias de detecção utilizadas serão vistas sem muitos detalhes, bem como a sua melhor localização na rede da organização.

7.1 Objetivos

Foi visto no capítulo anterior que o *firewall* é apenas um dos componentes dentro da estratégia de segurança de uma organização. Foi visto também que o foco está mudando, de uma abordagem baseada na segurança de borda, para a necessidade de maior acompanhamento e monitoramento das atividades dos usuários, já que o grande nível de interconectividade entre as organizações intensificam essa necessidade. Além da ameaça de *hackers*, que se movem entre diversos pontos de acesso para realizar a invasão, os ataques provenientes a partir da própria organização também são uma ameaça constante. Como muitos desses ataques internos podem ser realizados também através de contas comprometidas, um sistema de detecção de intrusões (*Intrusion Detection System* - IDS), que têm como objetivo detectar atividades inapropriadas, incorretas ou anômalas, é um elemento importante dentro do arsenal de defesa da organização. Além de ser importante para a segurança interna, o IDS pode detectar ataques que são realizados através de portas legítimas que são permitidas pelo *firewall*.

O IDS trabalha como uma câmera ou um alarme contra as intrusões, podendo realizar a detecção com base em assinaturas conhecidas ou em desvios de comportamento, como serão

vistos na seção 7.4. Algumas das ações que podem ser tomadas após a detecção de um ataque são [GRA 99]:

- Reconfiguração do *firewall*;
- Alarme (som);
- Aviso SNMP para sistemas de gerenciamento de redes, como o OpenView ou o Spectrum;
- Evento do NT;
- *Syslog*;
- Envio de *e-mail*;
- Envio de mensagem para o *pager*;
- Gravação das informações do ataque;
- Gravação das evidências do ataque para análise posterior (computação forense);
- Execução um programa capaz de manipular o evento;
- Finalização da conexão.

7.2 Características

Serão vistos nas próximas seções os diversos tipos e metodologias empregados pelos sistemas de detecção. Eles ainda estão em processo de amadurecimento, e não existe mesmo uma classificação mais clara para o que será apresentado, porém algumas das características chaves que devem ser observadas em um IDS são [HAL 98][SAN 99-2]:

- Detecção em tempo real;
- Provimento de informações valiosas sobre atividades maliciosas na rede;
- Análise baseada em cada caso, com resposta apropriada para cada um deles;
- Ajuda na identificação do local onde o ataque está ocorrendo;
- Flexibilidade e inteligência, já que cada organização é diferente. O IDS deve ser capaz de se adaptar a mudanças no ambiente da organização;

- Gerenciamento central, garantindo que todos os casos sejam analisados e respondidos de maneira consistente;
- Transparência, com o sistema não indicando quais pontos ou segmentos da rede estão sendo monitorados;
- Capacidade de reportagem gerencial efetiva, ou seja, habilidade de recriar invasões para prevenir novos ataques do mesmo tipo;
- Flexibilidade de resposta, com a capacidade de reação para a prevenção de danos;
- Configuração, tomando cuidado para respostas "falso positivo", que pode ser tão perigoso quanto a um ataque real.

7.3 Tipos

A classificação a seguir foi feita baseada em diversas fontes, não existindo ainda uma classificação clássica para o tema. Os sistemas de detecção de intrusões podem ser divididos entre os seguintes tipos, que serão vistos nas seções a seguir [HAL 98][GRA 99]:

- *Host-Based Intrusion Detection System*;
- *Network-Based Intrusion Detection System*;
- *Real-Time Activity Monitoring*;
- *System Integrity Verifiers*;
- *Log File Monitors*;
- *Deception Systems*.

7.3.1 Host-Based Intrusion Detection System

Também conhecido como *Audit Trail Analysis*, esse tipo de IDS (HIDS) faz o monitoramento do sistema baseado em informações de arquivos de *logs* ou de agentes de auditoria. O HIDS pode ser capaz de monitorar acessos e mudanças em arquivos críticos do sistema, além de mudanças nos privilégios dos usuários. A análise dos *logs* faz com que ataques de força bruta possam ser detectados, porém ataques mais sofisticados podem não ser detectados. As informações históricas indicam que tipo de ataque ocorreu, porém como o processo não é feito em tempo real, o

invasor pode explorar as vulnerabilidades, roubar informações ou introduzir códigos maliciosos antes que uma ação seja tomada com relação ao ataque [HAL 98]. Essa metodologia utilizada pelo HIDS é conhecida como *Behavior-Based Intrusion Detection*, que será vista na seção 7.4.2.

O HIDS é o responsável não apenas pelo monitoramento do tráfego, mas também pela checagem da integridade dos arquivos do sistema (*System Integrity Verifiers* – SIV) e pelo monitoramento de processos suspeitos (*Log File Monitors* – LFM).

O SIV monitora sistemas de arquivos em busca de arquivos modificados, que podem ser *backdoors*. Na maioria das vezes, eles são considerados ferramentas ao invés de sistemas, já que não são capazes de emitir alertas em tempo real. Um exemplo de SIV é o Tripwire [GRA 99];

O LFM monitora arquivos de *logs* gerados por serviços da rede. De maneira similar ao *Network-Based Intrusion Detection System* (seção 7.3.2), que utiliza a metodologia *Knowledge-Based Intrusion Detection* (seção 7.4.1), o LFM procura por padrões nos arquivos de *logs* que sugerem um ataque. Um exemplo de LFM é o Swatch [GRA 99].

Os *host wrappers* ou os *firewalls* individuais podem ser utilizados e configurados para monitorar todos os pacotes, tentativas de conexões ou tentativas de *logins* no *host* que está sendo monitorado, que pode incluir também tentativas de conexões *dial-in* ou em outras portas de comunicação diferentes das habituais [SAN 99-2]. Assim, eles também podem ser considerados um HIDS.

O próprio Unix possui ferramentas que realizam a detecção de intrusões [SAN 99-2]. Alguns exemplos são:

- *syslog* – arquivos de *log* do sistema e dos usuários;
- *TCPwrappers*, *lastlog* – monitoramento da conectividade;
- *lsof* – monitoramento de processos;
- *quotas* – monitoramento de utilização do disco;
- *audit* – auditoria do sistema;
- monitoramento de sessões, como os que podem ser realizados com o FTP.

O HIDS do UNIX é tão eficiente quanto os registros que forem realizados. Melhorias podem ser feitas pelos administradores, como por exemplo, escrever programas para realizar a análise de arquivos de *log* e alertá-los via *e-mail* ou *pager*.

7.3.2 Network-Based Intrusion Detection System

Também conhecido como *Packet Analysis*, o *Network-Based Intrusion Detection System* (NIDS) monitora o tráfego do segmento da rede, geralmente com a interface de rede atuando em modo promíscuo e em tempo real, como se os pacotes passassem por um sensor. A análise é feita através da captura e examinação dos cabeçalhos e conteúdos dos pacotes. A metodologia utilizada é o *Knowledge-Based Intrusion Detection*, que será vista na seção 7.4.1, onde as informações do tráfego são comparadas com assinaturas conhecidas. As ações são tomadas de acordo com essa avaliação.

Eficiente em ataques como o *IP spoofing* ou o *SYN flooding*, o NIDS não é eficiente contra o *buffer overflow* e contra ataques através de conexões discadas, além de não conseguir analisar pacotes cifrados. Outro problema é que esse tipo de IDS reside em equipamentos diferentes, e devido a características físicas e/ou drivers de redes diferentes, não conseguem prever se um equipamento pode aceitar um determinado pacote. Isso faz com que o sistema fique vulnerável a dois tipos de ataques, que envolvem o *stream* de dados [HAL 98]:

- Se o IDS rejeita um pacote que um equipamento deve aceitar, o atacante pode mudar a seqüência do *stream*, fazendo com que o IDS rejeite os pacotes válidos subseqüentes;
- Se o IDS aceita um pacote que o equipamento deve rejeitar, o atacante pode facilmente passar porções do *stream* de dados pelo monitor de detecção, que é então rearranjado na máquina, resultando em um ataque.

7.3.3 Real-Time Activity Monitoring

O *Real-Time Activity Monitoring* funciona através da instalação de agentes inteligentes nos sistemas e dispositivos de rede da organização. Esses agentes inteligentes possuem vantagem como [HAL 98]:

- centralização das atividades suspeitas que ocorrem em múltiplas localizações da rede;
- rápidas atualizações dos agentes com novas assinaturas digitais;
- detecção de intrusões mesmo se a conexão for cifrada ou se o invasor utiliza conexão discada.

7.3.4 Deception Systems

São também conhecidos como *sacrificial lamb*, *decoy*, *booby trap*, *lures*, *fly-traps* ou *honeypots*, que funcionam como armadilhas para a captura de *hackers* [GRA 99][SAN 99-2]. Eles não contém dados ou aplicações críticas para a organização, e o seu único propósito é de se passar por um legítimo equipamento da organização, que é configurado para interagir com um potencial *hacker*, de modo que detalhes do ataque sejam capturados. O *honeypot* é capaz de registrar novos tipos de ataques e a técnica utilizada pelo *hacker* que conseguiu passar pelos primeiros sistemas de defesa, geralmente o *firewall*. Um método para a configuração de um *honeypot* pode ser visto em [SPI 99-2].

7.4 Metodologias de Detecção

As metodologias utilizadas pelos IDS para a decisão de detecção de um ataque são o *Knowledge-Based Intrusion Detection*, o *Behavior-Based Intrusion Detection* e o *Computer Misuse Detection System*, que serão apresentadas a seguir.

7.4.1 Knowledge-Based Intrusion Detection

Também conhecida como *Misuse Detection Intrusion Detection System*, é a abordagem mais utilizada pelos IDS. Ele atua como um anti-vírus, onde um conjunto de assinaturas representam tipos de conexões e tráfegos, que podem indicar um ataque particular em progresso. Todas as ações que não são reconhecidas pelo conjunto de assinaturas são consideradas aceitáveis. A taxa de acertos desse tipo de IDS é considerada boa, porém depende de atualização constante dessa base de conhecimento, que é dependente de sistema operacional, versão, plataforma e aplicação [SAN 99-2].

O *Burglar Alarm* é um modelo que utiliza o *Knowledge-Based Intrusion Detection* e faz uma analogia a um alarme residencial, onde o alarme dispara de acordo com alguns eventos definidos. Assim como o alarme residencial pode ser armado de acordo com uma política (por exemplo, de que ela irá disparar se alguém entrar pela porta dos fundos ou pela janela e mexer num quadro), o *burglar alarm* também funciona de acordo com uma política definida, onde a detecção baseia-se no conhecimento da rede e no que não pode ocorrer na rede. A idéia é de que o

administrador possui o conhecimento da rede e o *hacker* não, de modo que assim ele pode definir o momento em que um alarme deve ser disparado [RAN 99].

Esse tipo de metodologia é mais rápido e não gera tanto falsos positivos se comparado com o *Behavior-Based Intrusion Detection* (seção 7.4.2), já que ele "entende" o ataque que está em andamento. O seu ponto fraco é que, assim como os anti-vírus com relação aos vírus, ele não consegue detectar ataques não conhecidos, novos ou que não foram atualizados pelo fabricante do sistema. Além disso, ele pode ser enganado através de técnicas como a inserção de espaços em branco no *stream* de dados do ataque [RAN 99]. Outro ponto negativo é o recurso computacional exigido, que é dificultado quando um ataque distribuído coordenado é realizado, onde a análise em tempo real de todos os pacotes (em grande número) pode ficar comprometida. Soluções como realizar análises em dados já capturados previamente, como pacotes da rede ou *logs*, reduz a necessidade de recursos computacionais, porém a detecção não é feita em tempo real [BRE 98].

As assinaturas, como as que detectam um grande número de falhas em conexões TCP em diversas portas, indicando que alguém esta realizando um *scanning* na rede, são divididos em 3 tipos [SAN 99-2]:

- *Strings* - olham por *strings* que indicam um possível ataque. Um exemplo de assinatura de *string* para Unix pode ser *"cat "+ +" > /.rhosts"*. Para minimizar o número de falsos positivos, é necessário refinar as assinaturas de *strings* utilizando assinaturas compostas, como por exemplo, os de ataques Web, que misturam "*cgi-bin*", "*aglimpse*" e "*IFS*";
- Portas - monitoram tentativas de conexões nas portas;
- Cabeçalho - procuram por combinações perigosas ou sem lógica nos cabeçalhos dos pacotes. Um exemplo é o *WinNuke*, que envia um pacote para a porta NetBIOS (139) e liga os *bits Urgent* e *Out of Band*, o que resulta no "*blue screen of death*" em sistemas Windows. Outro exemplo é a assinatura que identifica pacotes TCP que possuem ligados os *flags* SYN e o FIN, que significa que o cliente deseja iniciar e finalizar a conexão ao mesmo tempo, o que não pode existir em uma situação normal, sendo portanto um claro indício de tentativa de ataque.

7.4.2 Behavior-Based Intrusion Detection

O *Behavior-Based Intrusion Detection*, também conhecido como *Anomaly Detection Intrusion Detection Systems*, assume que as intrusões podem ser detectadas através de desvios de comportamento dos usuários ou dos sistemas. O modelo de normalidade é definido através de diversas maneiras (deve ser tomado cuidado para que o padrão de normalidade não seja definido quando o recurso está sendo atacado) e é comparado com a atividade corrente. Qualquer comportamento que não corresponda ao comportamento padrão é considerado intrusivo [SAN 99-2].

A decisão é tomada através de uma análise estatística para encontrar mudanças de comportamentos, tais como o aumento súbito de tráfego, de utilização da CPU, de atividade de disco, de *logon* de usuários, de acessos a discos, etc. A abordagem utilizada é a de que tudo o que não foi visto anteriormente é perigoso, e portanto deve ser evitado. Assim, todos os ataques podem ser capturados, mesmo os que não possuem assinaturas definidas, incluindo os ataques novos. Além disso, essa metodologia é independente de sistema operacional ou plataforma. O lado negativo dessa abordagem é que o IDS pode gerar falsos negativos (quando o ataque não causa mudanças significativas na metragem do tráfego) e falsos positivos (*bug* no sistema de monitoramento ou erro no modo de análise da metragem) [BRE 98]. Para minimizar esses problemas, diversas pesquisas estão em andamento, principalmente com a utilização de redes neurais, lógica *fuzzy*, e inteligência artificial [GRA 99]. Uma lista dos projetos e pesquisas em andamento com IDS pode ser visto em [COA 00].

7.4.3 Computer Misuse Detection System

O *Computer Misuse Detection System* (CMDS), da ODS Networks, é um IDS que inclui as características do *Knowledge-Based Intrusion Detection* e do *Behavior-Based Intrusion Detection* [ODS 99].

O CMDS utiliza uma arquitetura agente/gerenciamento/console que oferece diversas ferramentas para facilitar a coleta e a análise de *logs* e de dados de aplicações para a auditoria. Os agentes são pequenos programas que coletam, comprimem e cifram os dados. Os gerenciadores processam os dados à procura de assinaturas que representem eventos de segurança. O console oferece um GUI com uma série de ferramentas para a análise e geração de relatórios [SAN 99-2].

O CMDS possui um método universal de análise de *logs*, que permite que o administrador de segurança importe *logs* e dados particulares, de modo que a auditoria, o reconhecimento de assinaturas e a análise estatística possam ser realizadas.

As características do *Knowledge-Based Intrusion Detection* do CMDS podem ser vistas através de um sistema baseado em regras que avalia todos os eventos registrados em busca de assinaturas de ataques, que incluem falhas de *logins*, execução indevida de software, acessos a arquivos e diretórios ou a utilização de privilégios de administrador. Essas assinaturas podem ser adicionadas pelos usuários para a customização dos processos.

As características do *Behavior-Based Intrusion Detection* do CMDS podem ser vistas através do *Statistical Profiler*, que aprende automaticamente padrões de comportamento e os compara com a atividade corrente do sistema. Uma mudança de padrão de comportamento pode indicar roubo de dados ou comprometimento do nome de usuário e senha. Perfis para aplicações também podem ser definidos, como a de uma aplicação que gera determinada quantidade de informações e que passa a gerá-las em quantidades maiores. Perfis estatísticos também podem gerar alertas com relação a desvios de padrões.

7.5 Padrões

A padronização do IDS é um processo que ainda está em andamento, e tem como objetivo criar formatos e procedimentos para o compartilhamento de informações entre os sistemas. Alguns dos trabalhos que estão sendo realizados são o *Intrusion Detection Exchange Format* e o *Common Intrusion Detection Framework*, que serão vistos rapidamente a seguir:

7.5.1 Intrusion Detection Exchange Format

O *Intrusion Detection Exchange Format* faz parte do *Internet Engineering Task Force* (IETF) (http://www.ietf.org/html.charters/idwg-charter.html) e tem como objetivos:

- Definir formatos de dados e procedimentos para troca de formatos de respostas;
- Definir formatos de dados e procedimentos para o compartilhamento de informações de interesse a diversos sistemas de detecção de intrusões;
- Definir métodos de gerenciamento dos sistemas que necessitam interagir entre eles.

Os resultados esperados são:

- Criação de um documento que descreve os requerimentos funcionais de alto nível para a comunicação entre IDS e requerimentos para comunicação entre IDS e sistemas de gerenciamento;
- Especificação de uma linguagem comum que descreve os formatos de dados que satisfazem os requerimentos;
- Um *framework* que identifica os melhores protocolos utilizados para a comunicação entre IDS, descrevendo como os formatos de dados se relacionam com eles.

Alguns *Internet drafts* já criados são:

- *Intrusion Alert Protocols* (IAP) - protocolo no nível de aplicação para a troca de dados de alerta entre elementos IDS, como o sensor/analisador e gerentes em redes IP. O protocolo é designado para ser independente do tipo da representação de dados. O modelo de definição e formatação dos dados para alertas é descrito em documentos do grupo de trabalho;
- Requerimentos para o *Intrusion Detection Exchange Format*;
- Modelo de dados do *Intrusion Detection Exchange Format*.

7.5.2 Common Intrusion Detection Framework

O *Common Intrusion Detection Framework* (CIDF) (http://gost.isi.edu/cidf) é um projeto de pesquisa realizado pelo *Defense Advanced Research Projects Agency* (DARPA) para o desenvolvimento de um formato intercambiável a ser utilizado pelos pesquisadores do DARPA.

7.6 Localização do IDS na Rede

Um ponto importante a ser considerado na utilização de um IDS é quanto ao tráfego da rede. De acordo com [SAN 99-2], e levando-se em consideração o IDS RealSecure, quanto maior o tráfego de uma rede, menores as porcentagens de eventos registrados pelo sistema. Assim, uma das abordagem que podem ser utilizadas para o monitoramento adequado é a utilização de múltiplos sensores em diferentes pontos da rede, de modo que a análise de ataques possam ser divididos

entre esses pontos. Desta forma, cada sensor poderia realizar a inspeção dos pacotes de acordo com os diferentes riscos de ataques existentes.

Quanto à localização do IDS dentro da rede da organização, o HIDS não pode ser utilizado em sistemas Windows 9x, que não possui capacidade de registros suficientes para uma análise mais apurada dos eventos de segurança. A instalação de um *sniffer* por um *hacker* em um sistema Windows 9x, por exemplo, passaria despercebido aos olhos do administrador de segurança. Para redes com o Windows 9x, portanto, a melhor estratégia é a utilização do NIDS.

O IDS pode ser instalado em diversas localidades da rede da organização, onde cada posição significa um tipo de proteção específico. Algumas das posições em que o IDS pode ser utilizado podem ser vistos na figura 7.1.

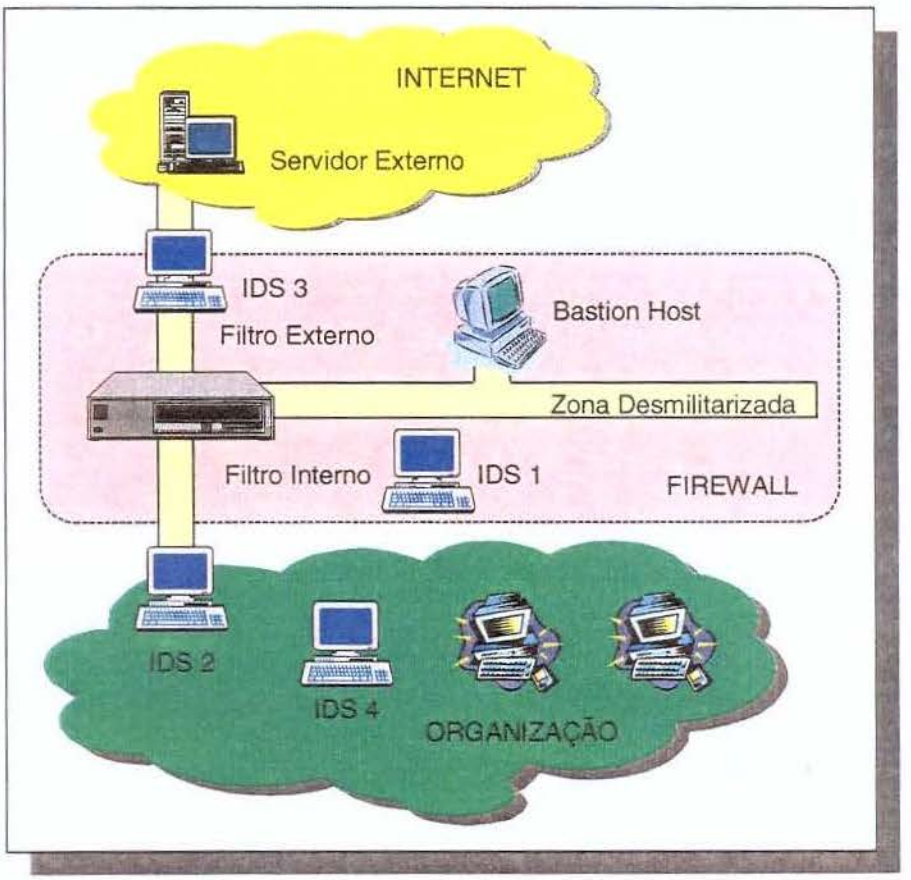


Figura 7.1: A localização do IDS na rede da organização.

As funções de cada posição podem ser vistas a seguir:

- IDS 1 – É pouco utilizada, pois o *firewall* produz poucas informações que podem ser analisadas;
- IDS 2 – Detecta ataques que passaram com sucesso pelo *firewall*;
- IDS 3 – Detecta ataques contra o *firewall*;
- IDS 4 – Detecta ataques internos na organização.

O IDS é mais eficiente se localizado no perímetro da rede, perto de servidores *dial-up* e nas conexões com outras organizações. O IDS pode funcionar bem em WANs, que geralmente possuem largura de banda relativamente baixas, que podem ser monitoradas pelo IDS, ao contrário de LANs, onde a alta largura de banda faz com que o IDS tenha alguns problemas.

Utilizar IDS em *switches* também é uma boa idéia, mas isso pode causar problemas devido ao alto tráfego nos segmentos. Uma possibilidade poderia ser utilizar a porta monitor do *switch*, que porém possui a desvantagem de possuir baixa taxa de velocidade. Outra possibilidade seria utilizar o NIDS entre *switches*, mas que possui o mesmo problema que ocorre com o alto tráfego;

7.7 Problemas do IDS

Alguns problemas a serem enfrentados pelos IDS são [GRA 99]:

- *Slow scans* - o NIDS pode ter dificuldade de manter registros de ataques com intervalos de tempos consideráveis, como onde o *hacker* realiza o *scanning* de uma porta a cada hora;
- Segmentos de rede com alto tráfego, como os de 100 Mbps são difíceis de serem monitorados, pois a quantidade de pacotes a serem analisados é muito grande;
- Ataques coordenados ou de baixa largura de banda;
- *IP Spoofing / Proxying* - impossibilita a descoberta da origem dos ataques;
- Limitação de recursos - como foi visto anteriormente, faltam melhores recursos para processar a fragmentação de pacotes, armazenar *logs* para evitar ataques coordenados ou lentos, e também a capacidade de analisar uma rede de 100 Mbps ou mais;
- Falso positivo - principalmente através da tecnologia de detecção de anomalias;

- Novos tipos de ataques - difícil monitoramento de ataques que utilizam a fragmentação de pacotes, ou os que mudam as configurações padrões de protocolos, como por exemplo, as que modificam a porta de conexão do Back Orifice.

7.8 Conclusão

Este capítulo apresentou rapidamente as características, os objetivos e os principais tipos de sistemas de detecção de intrusões (IDS). Esses sistemas estão em um processo constante de evolução, onde alguns padrões estão sendo desenvolvidos, para que a troca de informações entre os diversos sistemas seja facilitado. O capítulo apresentou também uma discussão sobre a melhor localização de um IDS na rede da organização, e os principais problemas a serem enfrentados por esses sistemas.

Capítulo 8

A Criptografia e a PKI

A criptografia é uma ciência que possui uma importância fundamental para a segurança, ao servir de base para diversas tecnologias e protocolos, tais como a *Public Key Infrastructure* (PKI) e o *IP Security* (IPSec). As suas propriedades – confidencialidade, integridade, autenticação e não-repúdio – garantem o armazenamento, as comunicações e as transações seguras, essenciais no mundo atual. Este capítulo discute o papel da criptografia e os aspectos relacionados à sua segurança, e também a PKI, componente importante em um ambiente baseado em chaves públicas.

8.1 O Papel da Criptografia

A criptografia possui cada vez mais uma função e importância fundamental dentro das soluções de segurança das organizações. Além de sua função primária, que é a de garantir a confidencialidade dos dados, a criptografia é a responsável também pela integridade, autenticação, certificação e não-repúdio, propriedades fundamentais em protocolos como o *Secure Shell* (SSH) e *IP Security* (IPSec), além de ser o ponto chave do *Virtual Private Network* (VPN) e da *Public Key Infrastructure* (PKI).

Apesar de fundamental, principalmente devido à necessidade crescente de sua utilização na Internet, Bellovin mostra em [BEL 98] que as soluções existentes são poucas, além de não serem completas. Alguns exemplos citados são o *Pretty Good Privacy* (PGP) e o *Secure Multi-Purpose Internet Mail Extensions* (S/MIME), utilizados para a segurança em *e-mails* (falta de uma certificação mais geral), o *Secure Socket Layer* (SSL) para a *Web* (autenticação apenas em uma via, onde apenas o servidor é autenticado e o usuário permanece sem autenticação nenhuma), o IPSec para a criptografia na camada de rede (conflito com *firewalls*, já que os pacotes IPSec possuem

cabeçalhos e conteúdos cifrados, que os *firewalls* não podem processar, e portanto filtrar) e o *Secure Electronic Transaction* (SET) para o comércio eletrônico (as lojas virtuais gostariam de ter acesso ao número do cartão de crédito para aproveitar a base de dados de seus clientes). Os problemas envolvendo a integração da VPN, que utiliza o IPSec, com os *firewalls* são analisados na seção 11.2, de modo que uma configuração correta faz com que a solução funcione da maneira adequada, ou seja, protegendo de fato a organização.

Tudo isso, aliado ao fato do poder de processamento estar seguindo a Lei de Moore, facilitando a quebra de chaves de alguns algoritmos criptográficos, mostra que a criptografia é uma área onde grandes evoluções irão acontecer. Um dos principais fatos está na escolha do *Advanced Encryption Standard* (AES) pelo *National Institute for Standards and Technology* (NIST), que irá substituir o DES, que é o algoritmo padrão atual. 5 finalistas foram escolhidos no *7th Fast Software Encryption Workshop* (FSE 2000), e após a escolha, as discussões e o processo formal de aprovação do governo, o novo padrão tem previsão para entrar em operação em meados de 2001. Um ponto importante é que o novo padrão será escolhido pela comunidade ligada à criptografia, e não pelo NSA [SCH 99-2]. Um outro fato é o avanço da criptografia de curvas elípticas, cada vez mais utilizada em componentes como *smart cards* e na computação móvel.

8.2 A Segurança dos Algoritmos Criptográficos

A segurança dos algoritmos criptográficos é baseada nos seguintes fatores [HER 98]:

- Tamanho da chave: são diferentes para a criptografia de chave privada e para a criptografia de chave pública;
- Mecanismo de troca das chaves: por exemplo, Diffie-Hellman para criptografia e RSA para assinaturas. O método preferido hoje é o *Internet Key Exchange* (IKE), se comparado com o *Simple Key Management for Internet Protocol* (SKIP). A vantagem primária do IKE sobre o SKIP é a sua habilidade de negociar com um número diferente de chaves criptográficas;
- Taxa de troca das chaves: como regra, quanto maior a freqüência da troca automática das chaves, maior a confidencialidade dos dados. A troca de chaves manual é considerada insegura, além de ser trabalhoso realizar todo o processo manualmente, o que pode influir na produtividade do usuário;

- Geração das chaves: com a utilização de um número aleatório real como base para a criação das chaves, é impossível saber ou adivinhar a estrutura das chaves futuras, o que garante uma maior segurança. A geração das chaves através de *hardwares* possuem a vantagem de utilizarem componentes dedicados na criação aleatória desses números, além de não utilizarem os algoritmos conhecidos utilizados pelos *softwares*, que podem ser quebrados mais facilmente.

Além dos fatores verificados, deve ser levada em consideração também a qualidade do sistema criptográfico e a sua correta implementação, seja ela em software ou hardware.

Em termos matemáticos, o algoritmo criptográfico, que tem origem a partir de um problema matemático difícil, é considerado seguro se 50.000 computadores não podem resolver esse problema em um milhão de anos. Diversos tipos de problemas matemáticos difíceis existem, tais como o problema do logaritmo discreto (Diffie-Hellman e *Digital Signature Algorithm* (DSA), fatoração de números grandes (RSA) e curvas elípticas [ROT 98-3]. Atualmente, os 3 mais utilizados são o *Integer Factorization Problem* (IFP), o *Discrete Logarithm Problem* (DLP) e o *Elliptic Curve Discrete Logarithm Problem* (ECDLP) [ROT 98-2].

Funções *one-way hash* são consideradas fáceis de serem executadas em uma direção, porém são extremamente difíceis de serem executadas na direção contrária. Fazendo-se uma analogia, esse tipo de função seria como um ovo, que pode ser facilmente quebrado, mexido e frito, porém quase impossível de recuperar a sua forma original. Funções *trap-door one-way hash* utilizam uma parte de informação (o *trap-door*) para realizar a função nas duas direções. O tamanho da chave determina o grau de dificuldade do problema matemático. Uma discussão teórica que envolve as funções *one-way hash* está relacionada com a sua própria existência, pois matematicamente não existem modos de provar essa afirmação [ROT 98-3].

Quanto ao RSA e a outros algortimos de chaves públicas, a sua segurança é baseada na dificuldade que envolve a fatoração de números primos grandes. Enquanto é fácil multiplicar dois números primos grandes, fatorar o produto desses dois números é muito mais difícil. As chaves pública e privada do RSA são funções de pares de números primos muito grandes, com centenas de dígitos. Uma característica do RSA e de outros algoritmos de chave pública é que ela pode ser utilizada para a cifragem de dados, e também para a autenticação via assinaturas digitais. [ROT 98-3].

8.2.1 A Segurança pelo Tamanho das Chaves

Foi visto que a segurança de um algoritmo criptográfico não pode ser medida apenas pelo tamanho da chave. Além disso, não se pode esquecer que a criptografia de chave privada (simétrica), de chave pública (assimétrica) possuem tamanho de chaves diferentes equivalentes, ou seja, o fato de um algoritmo de chave pública utilizar chaves de 512 bits não significa que ele seja mais seguro do que um algoritmo de chave privada que utiliza 128 bits. A tabela 8.1 apresenta as resistências comparativas quanto ao custo de processamento entre os algoritmos de chave simétrica e de chave assimétrica [GEU 00].

Chave Simétrica	Chave Pública
56 bits	384 bits
64 bits	512 bits
80 bits	768 bits
112 bits	1792 bits
128 bits	2304 bits

Tabela 8.1: Resistências comparativas entre os algoritmos de chave simétrica e assimétrica.

A questão do tamanho das chaves em algoritmos simétricos é avaliada em [BLA 96]. Um algoritmo criptográfico é considerado forte se não existem facilidades que permitam que se recupere as informações sem a utilização de ataques de força bruta (teste de todas as combinações de chaves), e também se o número de chaves possíveis é suficientemente grande para fazer com que ataques de força bruta sejam impraticáveis. A tabela 8.2 mostra o número de chaves possíveis no espaço de chaves e o tempo de processamento (1 milhão de tentativas/seg) [GEU 00].

Combinações Permitidas (Byte)	7 Bytes	7 Bytes	8 Bytes	8 Bytes
Letras minúsculas (26)	$8{,}0 \times 10^{9}$	2,2 horas	$2{,}1 \times 10^{11}$	2,4 dias
Minúsculas e dígitos (36)	$7{,}8 \times 10^{10}$	22 horas	$2{,}8 \times 10^{12}$	33 dias
Alfanuméricos (62)	$3{,}5 \times 10^{12}$	41 dias	$2{,}2 \times 10^{14}$	6,9 anos
Caracteres imprimíveis (95)	$7{,}0 \times 10^{13}$	2,2 anos	$6{,}6 \times 10^{15}$	210 anos
Caracteres ASCII (128)	$5{,}6 \times 10^{14}$	18 anos	$7{,}2 \times 10^{16}$	2.300 anos
Caracteres ASCII de 8 bits (256)	$7{,}2 \times 10^{16}$	2.300 anos	$1{,}8 \times 10^{19}$	580.000 anos

Tabela 8.2: O espaço de chaves e o tempo de processamento necessário.

As chaves desses algoritmos podem ser quebradas através de ataques de força bruta, que testam cada combinação possível de chave até que se descubra a combinação correta. Esse ataque de força bruta pode ser realizada desde por equipamentos convencionais (PCs), passando pela tecnologia *Field Programmable Gate Array* (FPGA) - um chip especial para a realização de cálculos, até o *Application-Specific Integrated Circuits* (ASICs), que é cerca de 7 vezes mais rápido que um chip FPGA, porém necessita de um grande investimento em engenharia, o que aumenta os seus custos. A tabela 8.3 mostra que basta ter o recurso financeiro para que as chaves sejam quebradas através de força bruta. A tabela 8.4 mostra o tempo de fatoração para quebra de chaves de algoritmos assimétricos.

A Lei de Moore parece estar de acordo com a realidade. Quando o DES foi proposto em 1975, a chave de 56 bits era considerada segura. Aplicando-se a Lei de Moore, o tamanho da chave considerada segura para 20 anos depois (1995, época do artigo), era 70 bits. O artigo recomenda a utilização de 75 bits, correspondente a 61 bits em 1975. Seguindo a mesma linha de raciocínio, para garantir a segurança de uma chave em um prazo de 20 anos a partir de 1995, o tamanho ideal é de 90 bits [BLA 96]. A grande diferença está agora na computação distribuída, principalmente através da Internet, onde um grande número de equipamentos pode trabalhar em paralelo para que o objetivo da quebra da chave seja alcançado mais rapidamente. Além disso, diversos

equipamentos dedicados para a quebra, como o Deep Crack e o Twinkle, que serão vistos na seção 8.4, contribuem para a evolução da criptografia.

Custo	56 bits	64 bits	112 bits	128 bits
$100 K	3,5 horas	37 dias	10^{13} anos	10^{18} anos
$1 M	21 minutos	4 dias	10^{12} anos	10^{17} anos
$10 M	2 minutos	9 horas	10^{11} anos	10^{16} anos
$100 M	13 segundos	1 hora	10^{10} anos	10^{15} anos
$1 G	1 segundo	5,4 minutos	10^{9} anos	10^{14} anos
$10 G	0,1 segundos	32 segundos	10^{8} anos	10^{13} anos
$100 G	0,01 segundos	3 segundos	10^{7} anos	10^{12} anos
$1 T	1 milissegundo	0,3 segundos	10^{6} anos	10^{11} anos

Tabela 8.3: Estimativas para ataques de força bruta.

Nº de bits	MIPS/Ano Necessários	Tempo /p Pentium II - 300 MHz
512	< 200	8 meses
768	100.000	300 anos
1024	3×10^{7}	1×10^{5} anos
1280	3×10^{9}	1×10^{7} anos
1536	2×10^{11}	7×10^{8} anos
2048	4×10^{14}	$1,3 \times 10^{12}$ anos

Tabela 8.4: Fatoração de chaves do algoritmo assimétrico.

Assim, é essencial considerar o tempo que a informação deverá ficar protegida pela criptografia, para que o tamanho ideal da chave seja utilizada para proteger efetivamente as informações. Os diversos tipos de informações necessitam de diferentes períodos de proteção, e portanto de diferentes tamanhos de chaves:

- Transferências eletrônicas de fundos, seja de milhões ou bilhões de dólares, necessitam de segurança, porém o tempo de exposição é extremamente curto;

- Planos estratégicos corporativos necessitam de confidencialidade durante alguns anos;
- Informações proprietárias de produtos, como a fórmula da Coca-Cola, necessitam ser protegidos por um longo período, talvez décadas ou séculos;
- Informações privadas pessoais, como condições médicas ou avaliações profissionais, devem ser protegidas durante a vida do indivíduo.

8.3 As Maiores Falhas nos Sistemas Criptográficos

Schneier analisa em [SCH 98] os fatores que podem causar falhas em sistemas criptográficos:

- falha na checagem do tamanho dos valores;
- reutilização de parâmetros aleatórios que nunca deveriam ser reutilizados;
- alguns sistemas não destróem a mensagem em texto claro depois da cifragem;
- alguns sistemas utilizam arquivos temporários para proteger os dados que podem ser perdidos durante uma pane no sistema, ou utilizam a memória virtual para aumentar a disponibilidade da memória;
- em casos extremos o sistema operacional pode deixar as chaves no disco rígido. Existem sistemas que permitem que a senha fique armazenada na memória de vídeo;
- falhas também na utilização da base de dados para a recuperação de chaves em emergências;
- em um sistema que utiliza geração de números aleatórios, se são geradas chaves fracas, o sistema é totalmente comprometido, não importando a efetividade do algoritmo criptográfico.

Essas falhas podem ser exploradas através de ataques por hardware, que podem introduzir deliberadamente falhas no processamento da criptografia para determinar as chaves secretas [SCH 98].

Bellovin [BEL 98] mostra também que os algoritmos hoje implementados possuem falhas devido à sua complexidade, e que os erros em seu desenvolvimento são comuns. Além disso, revendedores ainda comercializam produtos com algoritmos já considerados inseguros e até proprietários (apostando na segurança via obscuridade), além das interfaces com o usuário ainda serem difíceis de serem utilizadas.

8.4 Os Ataques aos Sistemas Criptográficos

Além dos ataques tradicionais às chaves, descritas em [SCH 96] – *chypertext-only attack, known-plaintext attack, chosen-plaintext attack, adaptive-chosen-plaintext attack, chosen-ciphertext attack, chosen-key attack e rubber-hose cryptanalysis (purchase-key attack)*, diversos outros tipos de ataques podem ser utilizados, contra os próprios sistemas criptográficos. O próprio Schneier diz em [SCH 98] que os ataques aos sistemas não são através da tentativa de testar todas as chaves possíveis (força bruta) ou explorar falhas nos algoritmos, mas sim na exploração de erros no *design*, na implementação e na instalação.

Em [SCH 99-1], Schneier fala da evolução dos métodos de *hacking* contra os sistemas criptográficos (*crypto-hacking*) e de seu futuro, que não é destinado à grande massa, como vem ocorrendo com a segurança em redes, já que a criptografia exige conhecimentos profundos de matemática avançada, o que não está ao alcance de todos [SCH 99-2]. Segundo ele, os métodos de *hacking* vêm evoluindo, o que pode ser visto em ataques do tipo *side-channel attacks*, onde a segurança dos *smart cards* e dos *tokens* são testados através de informações sobre tempo, consumo de energia e radiação do dispositivo. Outro tipo de ataque é o *failure analysis*, onde diversos tipos de falhas são forçadas durante a operação, de modo a quebrar a segurança de *smart cards*. Outro ataque é analisar não o algoritmo de criptografia em si, mas o gerador de números aleatórios; o algoritmo pode ser seguro, mas se o método de se produzir as chaves para o algoritmo for fraco, o número de chaves não é tão suficiente quanto deveria ser [SCH 99-1].

Os sistemas criptográficos podem ser quebrados através da análise dos modos com que as diferentes chaves se relacionam entre si. Cada chave pode ser seguro, porém a combinação de diversas chaves relacionadas pode ser suficiente para a criptanálise do sistema. Ainda é possível quebrar a segurança do RSA através da análise dos padrões de processamento, porém sem quebrar o algoritmo [SCH 99-1].

Um dos ataques é o *timing attack*, que faz a análise e mistura dos tempos relativos das operações da criptografia. Esse tipo de ataque é utilizado na recuperação de chaves privadas do RSA, e também contra *smart cards* e *tokens* de segurança, além de servidores de comércio eletrônico [SCH 98].

Muitas chaves são armazenadas no próprio equipamento do usuário, escondidas no meio de *strings* ou no próprio sistema. Shamir descreve em [SHA 98-4] ataques algébricos e estatísticos

utilizados para localizar chaves escondidas em uma grande *string* ou em grandes programas. Segundo Shamir, essas técnicas podem ser utilizadas para aplicar *lunchtime attacks* em chaves de assinatura utilizadas por instituições financeiras, ou para driblar o mecanismo de *authenticode* existente em alguns pacotes de software. O *lunchtime attack* é realizado por alguém que se aproveita da hora do lanche de algum funcionário de alguma instituição financeira, por exemplo, para procurar por chaves de assinatura, que pode estar em um arquivo dentro do seu equipamento, ou incorporada à própria aplicação. Uma importante consideração é que as chaves podem ser armazenadas no equipamento sem o conhecimento do usuário, como por exemplo, em arquivos *swap* do Windows (contém o estado intermediário da sessão de assinatura anterior), ou em arquivos de *backup* criados automaticamente pelo sistema operacional, em intervalos fixos. Ainda pode aparecer em setores danificados que não são considerados como sendo parte do sistema de arquivos [SHA 98-4].

Com relação a ataques de força bruta, em 1998 a *Electronic Frontier Foundation* (EFF) criou o Deep Crack, um computador com processamento paralelo de US$ 220.000, com o objetivo de demonstrar que o DES de 56 bits não oferece uma real segurança. O Deep Crack quebrou o DES no *RSA DES Challenge II* em 56 horas [MOS 99], sendo que a média das quebras foi de 4 dias e meio [SCH 99-2].

Após essa demonstração, o DES já não é mais recomendado para proteger informações por mais de 20 horas, sendo que algumas das opções são o 3DES, o *Carlisle Adams and Stafford Tavares* (CAST), *International Data Encryption Algorithm* (IDEA) e o *Rivest Cipher #5* (RC5), que utilizam conceitos parecidos com os do 3DES, ou seja, utilizam chaves de 128-bits com *cyper blocks* de 256-bits. Suas fraquezas, assim, são os mesmos do 3DES, sendo que o CAST, o IDEA e o RC5 possuem vantagens com relação ao 3DES no seu desempenho. O CAST, da Entrust, não necessita de pagamento de *royalties*. O CAST com 64-bits leva 235 dias para ser quebrado pelo Deep Crack, enquanto o CAST com 80-bits leva 43.000 dias. O CAST com 128-bits leva 3 milhões de vezes mais tempo do que o CAST de 80-bits. O DES-X é o DES com uma chave extra de 56-bits que faz operações XOR, aumentando significativamente a segurança do algoritmo, porém o 3DES ainda é mais seguro.

Um outro equipamento para a quebra das chaves, agora de chaves públicas, é o que Adi Shamir descreveu, o Twinkle [TWI 99]. O Twinkle é um computador elétrico-ótico destinado a fatorar números com velocidade com fator 1000. O computador ainda não foi construído, porém

Shamir mostra que isso é possível, mostrando ainda que chaves públicas de 512 bits não são mais seguras para a utilização operacional.

Apesar do Twinkle ainda não ser construído, um grupo de holandeses fatorou um número de 512 bits utilizando 300 *workstations* da Silicon Graphics Inc. e Pentiums, durante mais de 7 meses. O algoritmo utilizado foi o *General Number Field Sieve*. Schneider analisa que, se os esforços cooperativos utilizando-se recursos da Internet, como foram utilizados para a quebra do DES, fossem utilizados, a chave pública poderia ser quebrada em uma semana. A chave mínima recomendada pela RSA hoje é de 768 bits.

Ao mesmo tempo em que equipamentos dedicados à quebra de chaves são desenvolvidos, equipamentos dedicados também estão sendo desenvolvidos. O *Department of Energy's (DOE) Sandia National Laboratories* desenvolveu um cifrador cerca de 10 vezes mais rápido do que os similares, que pode cifrar mais de 6,7 bilhões de bits por segundo. Isso pode ser útil para a proteção de diversos tipos de dados digitais – vozes, áudio, vídeo, telefone celular, rádio e televisão. O chip utilizado é o *SNL Data Encryption Standard (DES) Application Specific Integrated Circuit* (ASIC), que consiste de 16 conjuntos de 16000 transistores integrados em um chip do tamanho de uma moeda. Além de suportar o DES, o DES ISIC suportará também novos algoritmos, como o *Advanced Encryption Standard* (AES), que será adotado como novo padrão para a criptografia simétrica [SAN 99].

8.5 Certificados Digitais

Diversos protocolos de segurança, como o *Secure Multipurpose Internet Mail Extensions* (S/MIME), o *Transport Layer Security* (TLS) e o *Internet Protocol Security* (IPSec) utilizam a criptografia de chaves públicas para prover a confidencialidade, a integridade, a autenticação, e o não-repúdio das comunicações. Os certificados digitais são um dos elementos baseados na criptografia de chave pública utilizados por esses protocolos, e as propriedades vistas são essenciais em um modelo de segurança como o do ambiente cooperativo, onde diversos níveis de acessos devem ser controlados e protegidos.

Os certificados digitais podem incluir diversas informações sobre o seu dono, e a quantidade dessas informações determina o nível de confiabilidade do certificado. A complexidade da estrutura e de determinados tipos de informações culminou na definição do *Attribute Certificate* (AC), que foi incorporado na definição do X.509 pelo *PKIX Working Group*. O formato do AC permite

que informações adicionais sejam ligadas ao certificado digital através de estruturas de dados assinadas digitalmente, e podem ter referências a múltiplos certificados [ARS 99].

Os certificados digitais são normalmente criados pelas autoridades certificadoras (*Certification Authorities* – CAs), que têm a função de criar, manter e controlar todos os certificados por ele emitidos, de modo que os certificados comprometidos ou expirados sejam invalidados. A manutenção envolve a segurança de sua própria chave privada, que se descoberta ou roubada compromete todo o sistema, sendo necessário assim invalidar os certificados anteriormente emitidos, e substituí-los com a nova chave do CA. Rothke [ROT 98] mostra a complexidade envolvida com todas essas funções de gerenciamento dos certificados digitais, o que justificou a definição de uma infra-estrutura de chave pública (*Public Key Infrastructure* – PKI), que possui componentes responsáveis por funções específicas, como poderão ser vistos nas seções a seguir.

8.6 Public Key Infrastructure

Em um ambiente heterogêneo como o ambiente cooperativo, o gerenciamento dos certificados digitais e de todas as suas funções torna-se extremamente complexo, fazendo com que uma infra-estrutura de chave pública (*Public-Key Infrastructure* - PKI) seja uma tecnologia importante dentro de uma arquitetura de segurança. A PKI é importante principalmente para a segurança interna da organização, ao tornar possível uma autenticação consistente, baseada nos certificados digitais, eliminando assim a necessidade de armazenamento de um grande número de senhas, e também de múltiplos processos de autenticação. Nesse ponto, pode-se considerar que a PKI pode funcionar como um *Single Sign-On* (seção 10.3), porém isso vem de encontro com a real interoperabilidade entre os diferentes sistemas que utilizam os certificados digitais como métodos de autenticação, entrando assim no mérito da certificação cruzada. Apesar dessa abordagem, será visto na seção 12.1.1 que mesmo essa diferenciação entre usuários internos, usuários externos e usuários remotos sofre uma alteração em um ambiente cooperativo, onde o acesso aos recursos internos torna-se cada vez maior, chegando-se aos níveis dos próprios usuários internos, através do *Virtual Private Network* (VPN) (capítulo 9). A característica principal do ambiente cooperativo é a formação de bolsões de segurança (seção 5.11).

Além do SSO, a PKI pode ser aplicado para outros objetivos: eliminar os documentos em papel e assinar de modo on-line os pacotes que trafegam pela rede [BRA 97].

Na sua forma mais simples, a PKI é um sistema utilizado para a publicação de chaves públicas utilizadas na criptografia de chaves públicas, que possui duas operações básicas [BRA 97]:

- Certificação – processo de ligar uma chave pública a um indivíduo, organização ou qualquer outra entidade, ou mesmo a uma peça de informação, como uma permissão ou uma credencial;
- Validação – processo de verificação da validade do certificado.

Já o *PKIX Working Group* (seção 8.6.4) define a PKI como sendo o conjunto de hardware, software, pessoas, políticas e procedimentos necessários para criar, gerenciar, armazenar, distribuir e revogar certificados digitais baseados na criptografia de chave pública [ARS 99].

Outras definições incluem o de McClure [MCC 98], que considera a PKI como sendo um *backbone* de uma corporação segura, enquanto a Netscape e a Verisign tratam a PKI como sendo uma combinação de software, criptografia e serviços que permitem que as organizações realizem transações pela Internet de modo seguro [NET 99].

Já do ponto de vista organizacional, a PKI pode ser considerado uma coleção de políticas, regras, responsabilidades, decisões, serviços e controles para a utilização da criptografia entre as aplicações da organização, além de ser também um conjunto de idéias, entendimentos, convenções, concordâncias, contratos, leis, regulamentos, instituições, pessoas e confiança que permite que os certificados e assinaturas digitais possam ser utilizados do mesmo modo que documentos são utilizados e documentos no papel são assinados [MUR 99].

8.6.1 Funções da PKI

O PKI possui uma série de funções, que são executadas por componentes específicos da infraestrutura, como poderá ser visto na próxima seção. Um ponto importante é que muitas das funções a seguir foram especificadas, porém não completamente implementadas, de modo que muitos problemas ainda têm que ser resolvidos.

As funções especificadas em uma PKI são [RSA 99][ARS 99][NET 99][SEC 99-4][BRE 99]:

- Registro – processo na qual uma entidade se registra a uma autoridade certificadora (*Certificate Authority* – CA), geralmente através de um *Registration Authority* (RA). O CA, que pode contar com a ajuda do RA, verifica se o nome e outros atributos estão corretos, de acordo com a política da organização definida no *Certification Practice Statement* (CPS) [BHI 98];

- Inicialização – é o processo na qual a entidade pega os valores necessários para o início das comunicações com a PKI. Por exemplo, a inicialização pode envolver o fornecimento para o cliente da chave pública e/ou certificado digital do CA, ou a geração do par de chaves privada/pública da própria entidade (cliente);

- Certificação – é o processo em que o CA envia um certificado digital para a entidade que a solicitou, e coloca esse certificado em um repositório;

- Recuperação do par de chaves – em algumas situações, uma organização quer ser capaz de ter acesso a informações que estão protegidas, como *e-mails* ou projetos, quando um funcionário não está disponível, seja porque ele está doente, porque ele não trabalha mais para a organização, ou mesmo para uma investigação sobre a sua conduta. Nesses casos, o *backup* da chave privada do usuário pode ser feita por um CA ou por um sistema separado de *backup*. A PKI deve prover um sistema que permita a recuperação, sem prover riscos inaceitáveis de comprometimento da chave privada.

- Geração de chaves – dependendo da política do CA, o par de chaves pode ser gerado pelo próprio usuário em seu ambiente local, ou ser gerado pelo CA. Nesse último caso, a chave pode ser distribuída para o usuário em um arquivo cifrado ou em um *token* (*smart card* ou cartão PCMCIA);

- Atualização das chaves – todo par de chaves precisa ser atualizado regularmente, isto é, substituído por um novo par de chaves. Isso deve acontecer em dois casos: normalmente, quando a chave ultrapassa o seu tempo de vida, e, excepcionalmente, quando a chave é comprometida.

- Certificação cruzada – a certificação cruzada é necessária quando um certificado de um CA é enviado para um outro CA, de modo que uma entidade de um domínio administrativo possa se comunicar de modo seguro com uma entidade de um outro domínio administrativo;

- Revogação – várias circunstâncias podem fazer com que um certificado tenha a sua validade revogada antes da expiração do período de validade. Essas circunstâncias incluem mudanças no nome, na associação entre a entidade e o CA (funcionário que sai da organização), e quebra da confidencialidade da chave privada correspondente. O X.509 define um método de revogação de certificados, que inclui uma estrutura de dados assinada digitalmente, chamada *Certificate Revocation List* (CRL). O CRL é uma lista com *time-stamp* que identifica os certificados revogados (através do seu número serial) que permanece disponível livremente em um repositório público. Algumas considerações com relação ao CRL incluem a freqüência de sua atualização e a remoção do certificado da lista, por exemplo, quando o prazo de validade do certificado expira;

- Distribuição e publicação da notificação de revogação e dos certificados – a distribuição dos certificados inclui a sua transmissão para o seu proprietário, e a sua publicação envolve o repositório dos certificados. A distribuição da notificação da revogação envolve postar CRLs em um repositório.

8.6.2 Componentes da PKI

Os componentes da PKI, definidos pelo *PKIX Working Group*, são [ARS 99][RSA 99][SEC 99-6][BHI 98][ENT 99]:

- Autoridade certificadora (*Certificate Authority* – CA), que é a entidade que cria os certificados digitais. Pode ser interno a uma organização, ou ser uma terceira parte confiável, como pode ser visto na seção 8.6.2.1;

- *Organizational Registration Authorities* (ORAs), ou simplesmente RA, que é uma entidade dedicada a realizar o registro dos usuários (processo de coletar informações do usuário e verificar a sua identidade) e aceitar as requisições de certificados. O departamento de recursos humanos pode gerenciar o RA, enquanto o departamento de informática pode gerenciar o CA. O RA pode ser uma função do CA [RSA 99];

- *Certificate Holders*, que podem assinar digitalmente e cifrar documentos;

- Clientes, que validam as assinaturas digitais e os caminhos de certificação a partir de uma chave pública conhecida de um CA confiável;

- Serviço de diretório, como o *Lightweight Directory Access Protocol* (LDAP), que funciona como repositório para as chaves, certificados e *Certificate Revocation Lists* (CRLs), que são as listas com certificados inválidos;

A figura 8.1 simplifica o arquitetura do modelo assumido pelo *PKIX Working Group* [ARS 99]:

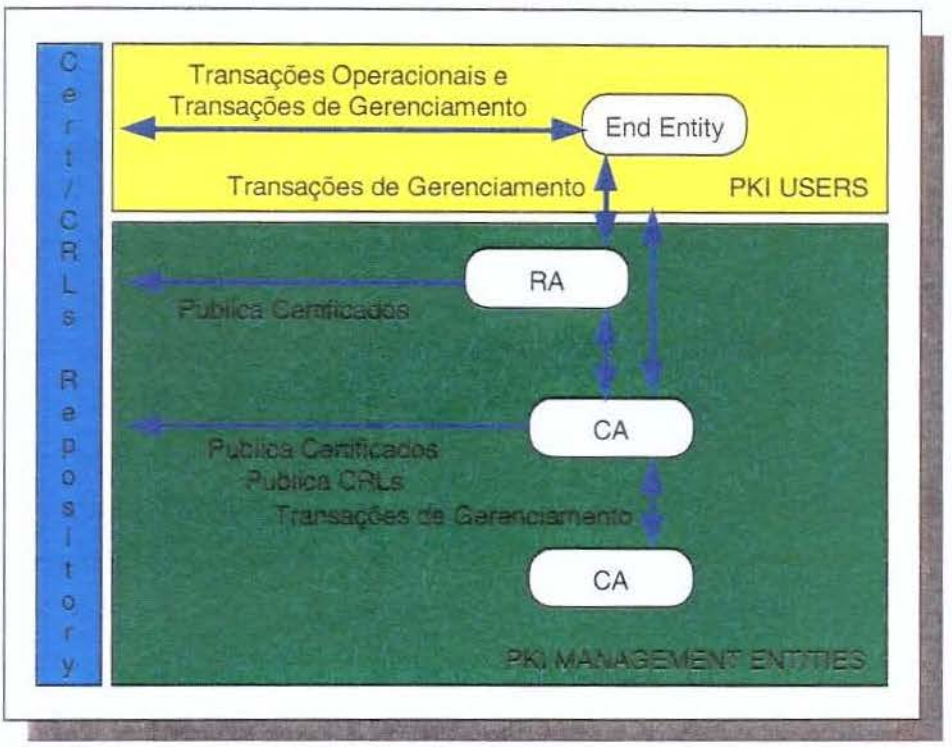


Figura 8.1: A arquitetura do modelo PKIX.

8.6.2.1 A Autoridade Certificadora

O modelo de confiança das autoridades certificadoras (CAs) pode ser considerado sendo de 3 tipos [ROT 98][RSA 99]:

- Modelo de autoridade central, onde a autoridade certificadora é única;
- Modelo de autoridade hierárquica, onde uma cadeia de autoridades emite os certificados para outras autoridades no nível mais baixo da cadeia, e assim por diante. A autoridade certificadora principal (*root*) certifica autoridades primárias (*Primary Certification Authorities* (PCAs), que podem criar, suspender e revogar certificados dentro da hierarquia. Os PCAs, por sua vez, podem certificar CAs. Exemplificando, o S/MIME utiliza uma hierarquia de confiança (*chain of trust*), onde um certificado de um CA deve ser aceito por um CA confiável. Isso é utilizado, por exemplo, quando um certificado A1 é validado por

CA1, porém a organização não confia em CA1. Então CA1 deve ter seu certificado validado por CA2, na qual a organização confia plenamente. O CA mais conhecido é o Verisign [ZDN 98].

- *Web of Trust*, onde a responsabilidade da confiança está no próprio usuário, ou seja, se Tom confia em Anne, e Anne confia em Beth, então Tom confia em Beth. Este é o modo em que funciona o *Pretty Good Privacy* (PGP).

8.6.3 Desafios da PKI

Foi visto até agora que a especificação da PKI define uma série de funções, componentes e protocolos, mas que possui ainda alguns aspectos a serem melhor discutidos, principalmente quando a sua utilização é analisada em um ambiente produtivo. Alguns desses desafios incluem a certificação cruzada, as listas de revogação (CRLs) e as convenção de nomes. A certificação cruzada é um ponto importante, pois, se os CAs de duas companhias, A e B, são diferentes, como a companhia A pode confiar em um usuário que possui um certificado da companhia B? A tecnologia atual não permite essa interoperabilidade sem que exista uma combinação prévia, porém diversos padrões estão sendo propostos atualmente (seção 8.6.4). Uma solução, adotada pelo *Automotive Network eXchange* (ANX), é o uso de um CA terceirizado (modelo de autoridades hierárquica), que certifica todos os certificados digitais dos diversos outros CAs (por exemplo, a Verisign ou a Entrust) [SEC 99-4].

A interoperabilidade é de fato um dos principais aspectos a serem resolvidos pelas PKIs, seja com relação às operações (geração, distribuição e gerenciamento dos certificados) ou com relação aos formatos dos componentes da PKI. O que pode-se verificar é que os padrões estão sendo definidos, porém eles não são sempre implementados ou demoram a serem desenvolvidos. Isso faz com que as organizações tenham que escolher um único fabricante para a sua solução PKI, o que traz uma limitação quanto à escalabilidade, além de fazer com que organização fique dependente da evolução desse fabricante [BRE 99].

A interoperabilidade entre as PKIs é uma questão séria que trouxe algumas sugestões, como as da LockStar Inc. ou da SHYM Technology. A solução da SHYM, por exemplo, cria uma infraestrutura para onde os pedidos são encaminhados, de onde o servidor SHYM determina em qual PKI a aplicação está ligada, e verifica se o certificado é válido para essa aplicação. Assim o pedido

é encaminhado para a PKI apropriada. O lado negativo dessa solução é a necessidade do software cliente, o que aumenta a complexidade e os custos [BRE 99].

Além desses aspectos, a implementação da PKI traz uma série de pontos que ainda precisam ser analisados, como as aplicações que manipulam os certificados, a aceitação dos usuários, a legislação, o planejamento e a escalabilidade. Esses pontos podem ser vistos em [BRE 99] e [BHI 98]. [MUR 99] enfatiza a idéia de que uma infra-estrutura não se compra, se constrói, mostrando a importância de uma visão estratégica para a implementação da PKI na organização. Deve-se pensar na PKI como um meio ou uma "estrada" para que uma conexão segura seja possível. Assim, como se pode comprar um automóvel, mas não se pode comprar a estrada, não se pode comprar a PKI. Porém se pode comprar os serviços de diretórios e os certificados digitais que são utilizados pela infra-estrutura. Alguns riscos a serem considerados na implantação de uma infra-estrutura de chaves públicas podem ser vistos no artigo escrito por Schneier em [SCH 00].

8.6.4 Padrões PKI

Como visto anteriormente, a interoperabilidade entre as PKIs forma um grande obstáculo para a sua real implementação dentro das organizações. Os padrões PKI, que serão discutidos nessa seção, têm como objetivo permitir a interoperabilidade entre diferentes PKIs, através das definições de [RSA 99]:

- Procedimentos de registros;
- Formatos de certificados;
- Formatos de CRLs;
- Formatos para as mensagens de registro (requisição, certificados, certificados do servidor);
- Formatos para as assinaturas digitais;
- Protocolos de desafio/resposta.

Um dos padrões é definido pelo *Internet Engineering Task Force* (IETF), conhecido como *PKIX Group* (*PKI for X.509 certificates*).

As 5 diferentes áreas em que o *PKIX Working Group* desenvolve documentos são [ARS 99]:

- *Profiles* do X.509 v3 e padrões CRL do X.509 v2 para a Internet;

- Protocolos operacionais, onde as partes envolvidas podem obter informações dos certificados, bem como de seus *status*;
- Protocolos de gerenciamento, onde diferentes entidades do sistema trocam informações necessárias para o gerenciamento apropriado da PKI;
- Informações sobre políticas de certificados e declarações de práticas de certificações, cobrindo áreas da segurança da PKI que não são diretamente endereçadas no resto do PKIX;
- *Time stamping* e certificação de dados, que podem ser utilizados para a construção de serviços como o não-repúdio.

A especificação do PKIX é baseada em 2 padrões: o x.509, da *International Telecommunication Union* (ITU) e o *Public Key Cryptography Standards* (PKCS), da RSA Security. O X.509 tem o objetivo de especificar serviços de autenticação para o serviço de diretórios X.500, porém não tem a intenção de definir uma PKI completa e interoperável. O formato padrão dos certificados digitais é definido pelo padrão X.509 [RSA 99].

O PKCS é uma série de padrões que cobrem a PKI em áreas como o processo de registro e renovação dos certificados, e a distribuição de CRLs. Para a interoperabilidade de PKIs, os padrões mais importantes são o PKCS #7 (*Cryptographic Message Syntax Standard*), o PKCS #10 (*Certificate Request Syntax Standard*) e o PKCS #12 (*Personal Information Exchange Syntax Standard*) [RSA 99].

Outras definições são o *Certificate Management Protocol* (PKIX CMP) e o *Certificate Management Message Format* (CMMF).

8.7 Conclusão

Foi visto que a criptografia possui uma importância fundamental para as organizações, ao prover segurança através da confidencialidade, integridade, autenticação e não-repúdio. Diversos aspectos devem ser considerados para que um algoritmo criptográfico seja seguro, que foram vistos neste capítulo. Os certificados digitais, provenientes da criptografia de chaves públicas, possui um papel importante em um ambiente cooperativo, ao facilitar, principalmente, a autenticação entre usuários de organizações diferentes de modo mais seguro do que o tradicional. Isso faz com que uma infra-estrutura de chave pública (*Public Key Infrastructure* – PKI) seja considerada.

Capítulo 9

Virtual Private Network

As redes privadas virtuais (*Virtual Private Network* – VPN) possuem uma importância fundamental para as organizações, principalmente no seu aspecto econômico, ao permitirem que as conexões dedicadas sejam substituídas pelas conexões públicas. Além do que ocorre com as conexões privadas, economias também podem ser geradas com a substituição das estruturas de conexões remotas, que podem ser eliminadas em função da utilização dos clientes VPN e dos provedores VPN. Porém, essas vantagens requerem uma série de considerações com relação à segurança, já que as informações das organizações passam a trafegar através de uma rede pública. Esse capítulo visa discutir a VPN e as implicações de segurança envolvidas, além dos principais protocolos disponíveis para a comunicação entre as organizações através de túneis virtuais.

9.1 Objetivos e Configurações

As Redes Privadas Virtuais (*Virtual Private Network* – VPN) são um componente importante dentro do ambiente cooperativo, e têm como objetivo utilizar uma rede pública para a comunicação, em substituição às conexões privadas e às estruturas de acesso remoto, que possuem custos bastante elevados. Para os usuários que se comunicam através de suas organizações, é como se essas duas redes diferentes fossem na realidade uma única rede, constituindo assim uma rede virtual privada que passa fisicamente por uma rede pública. Esse tipo de VPN, que é transparente ao usuário, pode ser chamado de *gateway-to-gateway VPN* (Figura 9.1), e o túnel VPN (seção 9.3) é iniciado e finalizado nos *gateways* das organizações. O *gateway-to-gateway VPN* pode ser visto também em acessos remotos, quando o usuário se conecta ao provedor VPN, de

onde o túnel VPN é iniciado (Figura 9.2). Outro tipo de VPN é o *client-to-gateway VPN*, onde o túnel é iniciado no próprio equipamento do usuário, através de um software cliente (Figuras 9.3 e 9.4).

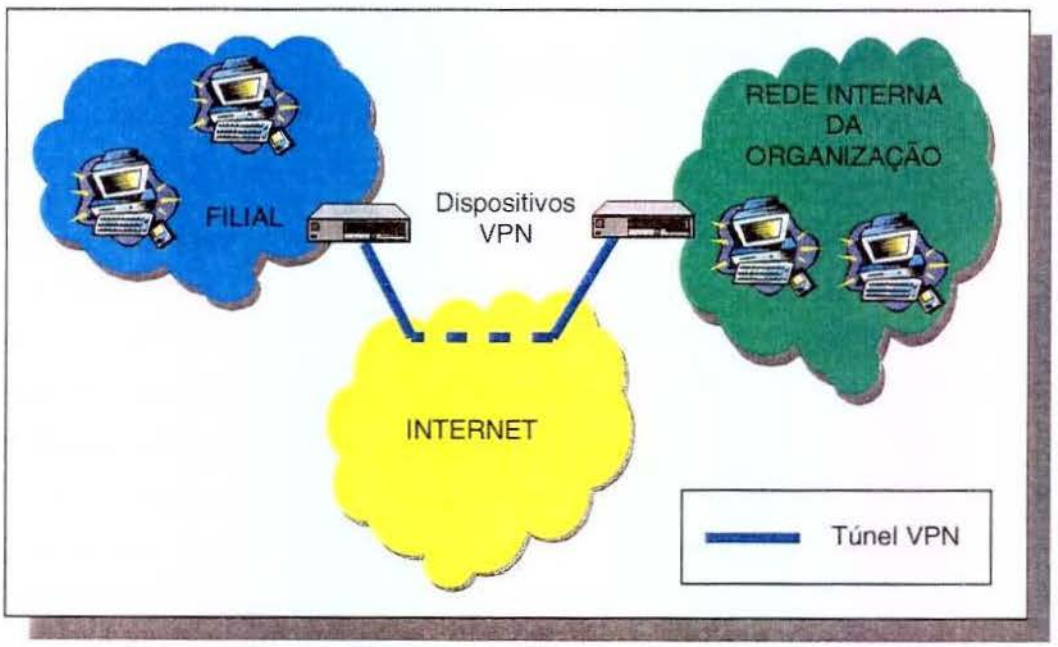


Figura 9.1: *Gateway-to-gateway VPN*, onde o túnel VPN é criado entre 2 redes.

Esses dois tipos de VPNs podem ser utilizados para caracterizar uma *Intranet VPN*, que conecta departamentos e filiais dentro de uma organização, ou uma *Extranet VPN*, que conecta a organização a parceiros estratégicos, clientes e fornecedores. A *Intranet VPN* exige uma tecnologia de ponta para as conexões de alta velocidade presentes em LANs, além de alta confiabilidade, para assegurar a prioridade em aplicações de missão crítica. A facilidade de gerenciamento para acomodar mudanças com novos usuários, novas filiais e novas aplicações também é importante. Já a *Extranet VPN* requer uma solução padrão para assegurar a interoperabilidade entre as várias soluções dos parceiros, sendo que o controle de tráfego é importante para se evitar os gargalos e garantir a rápida resposta aos dados críticos.

Além da economia com as linhas dedicadas, a VPN pode ser utilizada também como substituto dos acessos remotos tradicionais. A manutenção com os componentes do acesso remoto,

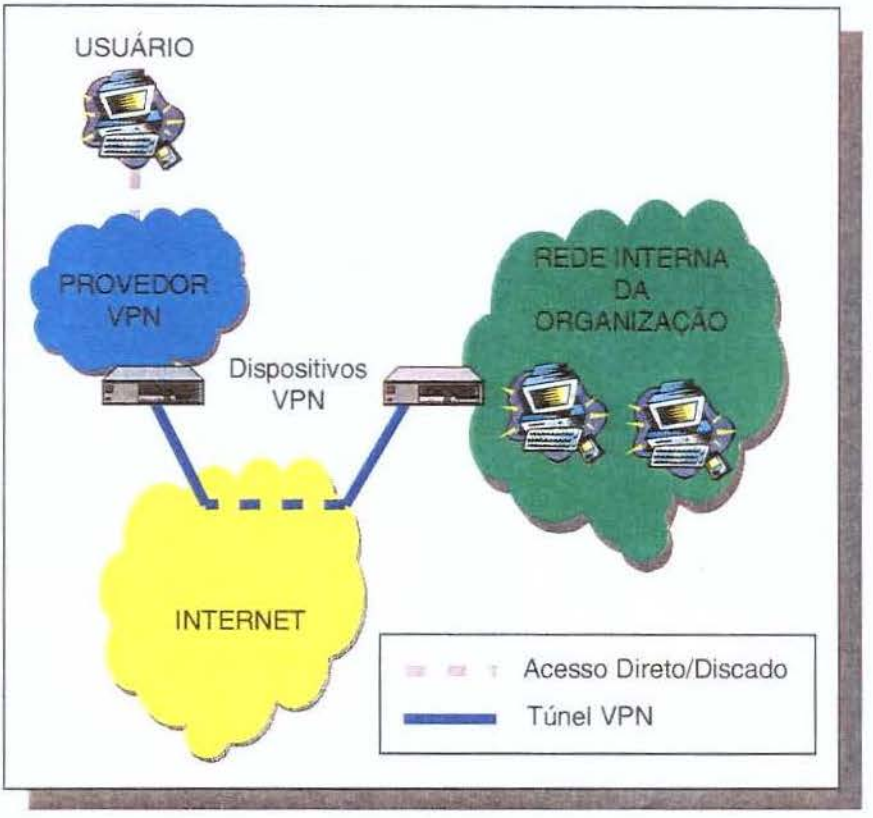


Figura 9.2: *Gateway-to-gateway VPN*, onde o usuário utiliza um provedor VPN.

que incluem o *pool* de modens e as linhas telefônicas, pode ser considerada bem mais cara do que uma solução VPN. Com isso, a VPN permite uma economia significativa, também com a administração, que estaria a cargo do provedor de acesso Internet ou de acesso VPN. Essa solução, onde o túnel VPN é iniciado no cliente, que se conecta a um provedor de acesso, substituindo os acesso remotos diretos, é conhecida como *remote-access VPN*, e pode ser vista na Figura 9.5. O *remote-access VPN* possui uma grande aplicabilidade em um ambiente cooperativo, onde os usuários remotos podem deixar de realizar ligações interurbanas, acessando os recursos da organização através de um túnel virtual criado através da Internet. Uma autenticação forte é um requerimento para o *remote-access VPN*, já que os recursos da organização são acessados diretamente pelos usuários, e a segurança física é difícil de ser implementada em soluções remotas.

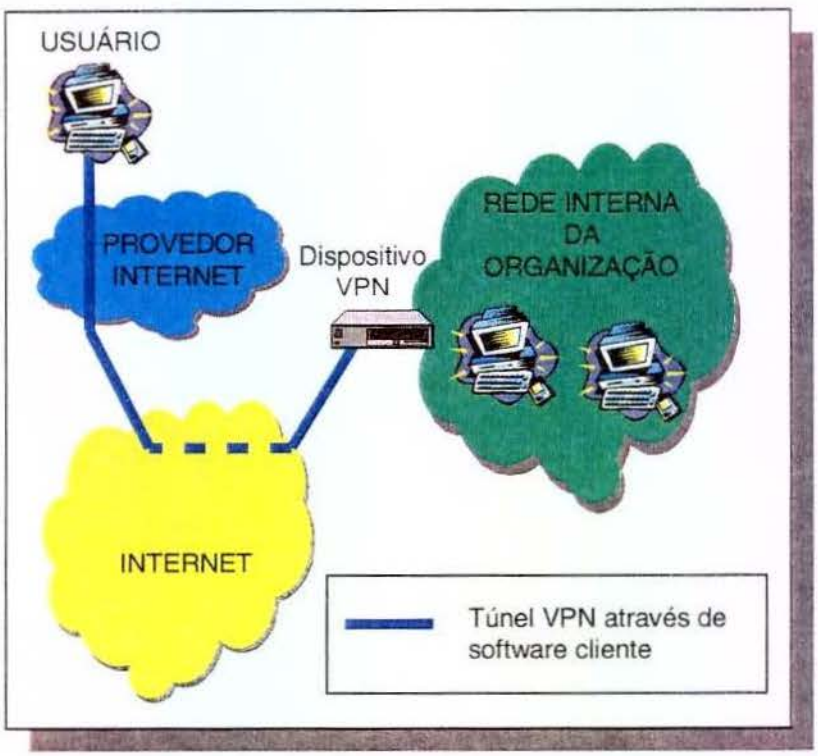


Figura 9.3: *Client-to-gateway VPN*, com provedor de acesso e software VPN.

Uma outra forma de *remote-access VPN*, menos comum no Brasil, é onde o túnel VPN é iniciado no provedor de acesso, que se torna assim o provedor VPN. O cliente assim utilizaria uma conexão discada PPP para o provedor VPN, de onde o túnel seria iniciado para a rede da organização (Figura 9.6). Um filtro de pacotes no provedor poderia permitir apenas o tráfego VPN para a rede da organização, eliminando assim a possibilidade do cliente ser utilizado como uma ponte entre a Internet e a rede da organização, um risco bastante comum na primeira forma de *remote-access VPN*, onde o túnel é iniciado no próprio cliente, através de um software VPN [NAK 00].

Assim, quando a VPN é utilizada, o serviço aparece para o usuário como se ele estivesse conectado diretamente à rede privada, quando na realidade ele utiliza uma infra-estrutura pública [BAY 98]. A utilização da rede pública para a comunicação entre matriz, filiais e parceiros comerciais significa custos mais baixos (A Forrester Research estima que a redução dos custos é maior que 60%) e maior flexibilidade e escalabilidade com relação a usuários móveis e a mudan-

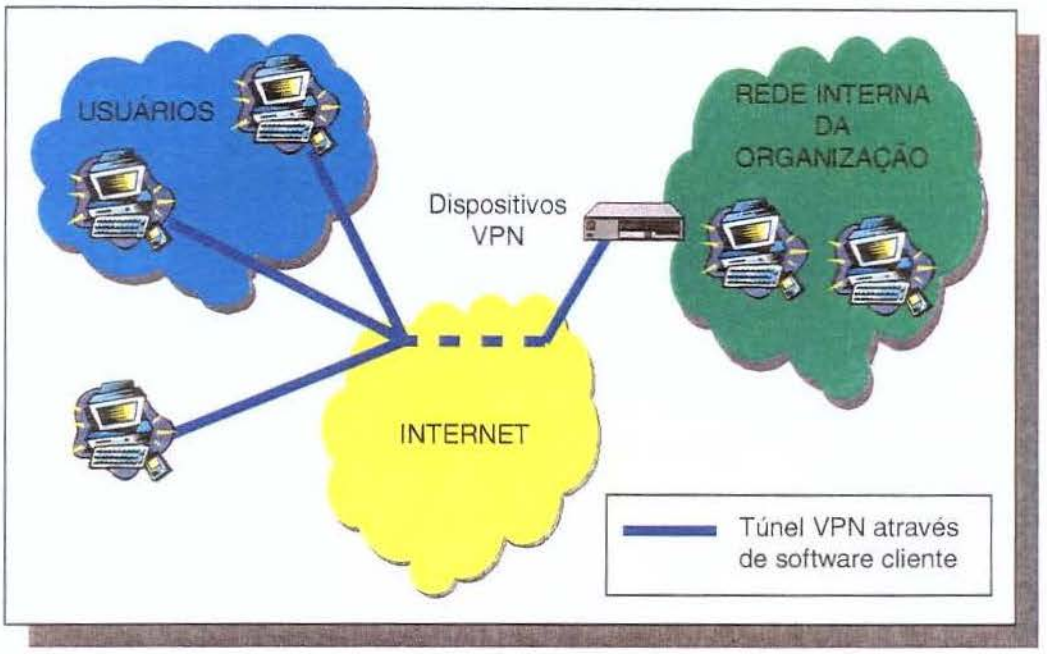


Figura 9.4: *Client-to-gateway VPN*, onde os usuários utilizam um software VPN.

ças nas conexões (comparando com as conexões privadas, que possuem custos altos para mudanças dessas conexões) [SEC 98]. Pesquisas da Forrester Research, Inc [MIN 97] indicam que as VPNs estão sendo cada vez mais utilizadas, principalmente devido ao melhor custo/benefício, à utilização do IP, por ser mais escalável e por ser uma alternativa à expansão da WAN. De fato, a utilização de uma conexão Internet facilita o gerenciamento das conexões (não é mais necessário criar um ponto de conexão privado para cada uma das conexões, apenas uma, para a Internet), tirando-se vantagem ainda da conectividade global, que é mais difícil de ser alcançada através de conexões dedicadas. Esse conjunto de fatores facilita a conexão entre as organizações, que podem assim buscar a evolução natural em seus processos de negócios.

9.2 Implicações

A proposta da VPN, de substituir as caras conexões dedicadas e estruturas de acesso remoto pela utilização da rede pública, trouxe uma série de implicações, principalmente quanto à segurança das informações, que passam a correr riscos com relação à sua confidencialidade e à sua integridade.

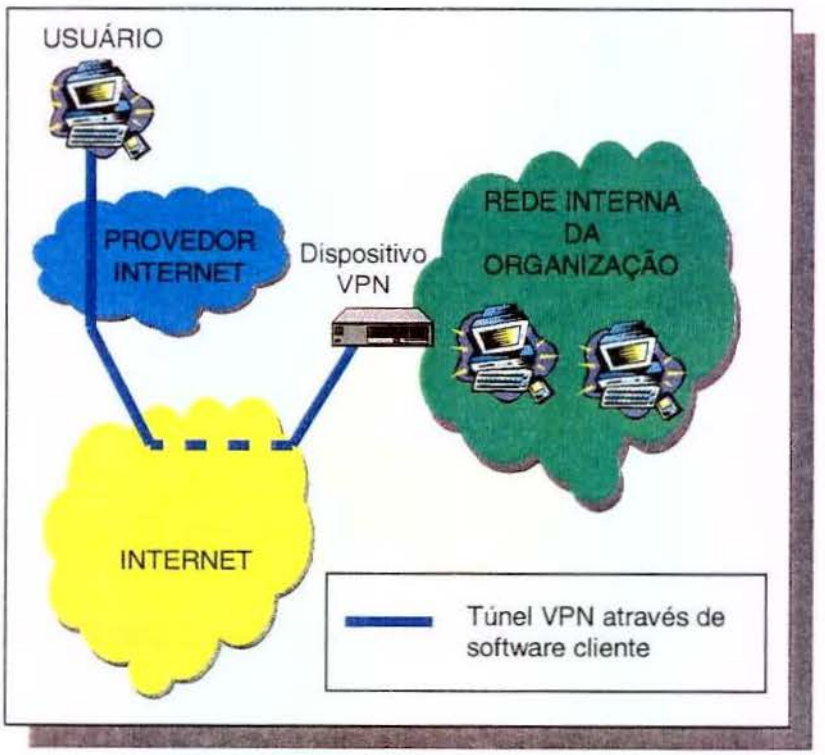


Figura 9.5: *Remote-access VPN*, através de provedor Internet e software VPN.

O primeiro problema que pode ocorrer com a utilização da rede pública é a possibilidade de *packet sniffing* (seção 4.5.2), onde qualquer indivíduo tem a possibilidade de capturar pacotes contendo informações das organizações, quebrando assim a sua confidencialidade. Outro problema é a possibilidade de um ataque ativo à conexão TCP (seção 4.7), de modo que a sua integridade pode ser comprometida. Problemas de *IP Spoofing* (seção 4.5.6) também podem ocorrer, com um usuário podendo se passar como sendo outro, causando assim problemas de autenticação e autorização.

9.3 Os Fundamentos da VPN

Os conceitos que fundamentam a VPN são a criptografia e o tunelamento. A criptografia é utilizada para garantir a autenticidade, a confidencialidade e a integridade das conexões, e é a base para a segurança dos túneis VPN, como poderá ser observado na seção 9.5.2, que discute o IPSec, um dos protocolos mais difundidos em VPNs. Trabalhando na camada 3 do modelo ISO/

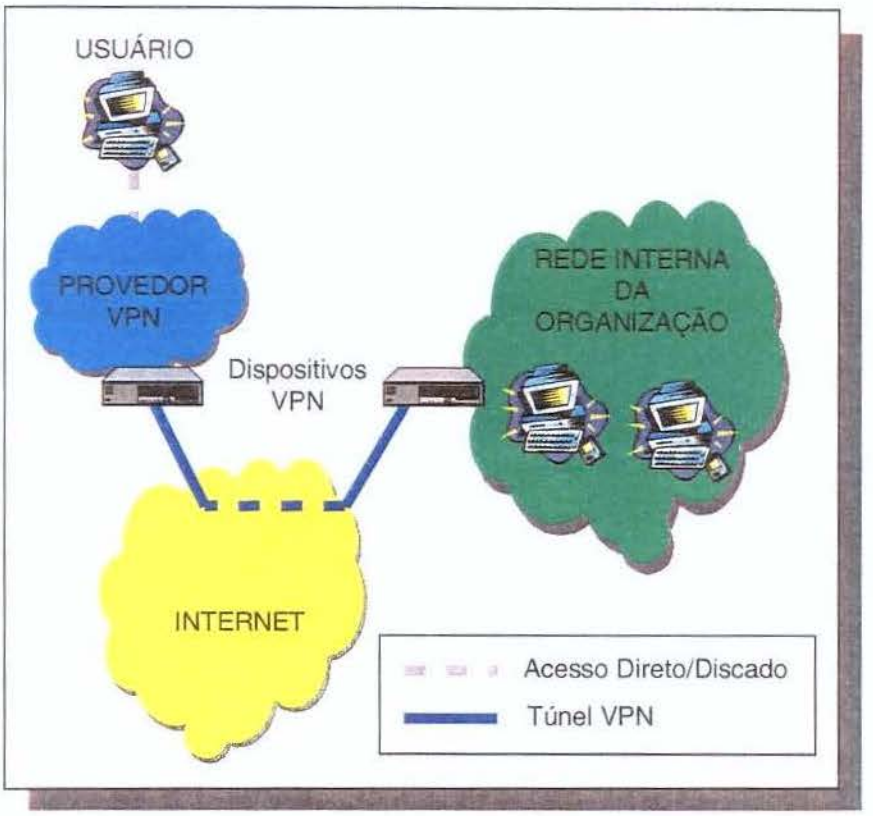


Figura 9.6: *Remote-access VPN*, através de provedor VPN, de onde o túnel é criado.

OSI, a criptografia é independente da rede e da aplicação, podendo ser aplicada em qualquer forma de comunicação roteável, como voz, vídeo e dados [HER 98].

O túnel VPN é formado pelo tunelamento, que permite a utilização de uma rede pública para o tráfego das informações, inclusive de protocolos não IP, através da criação de um túnel virtual que é formado entre as duas partes da conexão.

Pode-se considerar, portanto, que uma VPN é formada pelo conjunto do tunelamento, que permite o tráfego em uma rede pública, e da criptografia, que visa garantir a segurança dessa conexão. Porém, os diversos protocolos existentes diferem entre si na camada do modelo ISO/OSI onde atuam, ou no modo em que a criptografia é utilizada: o *Layer 2 Tunneling Protocol* (L2TP) e o *Point-to-Point Tunneling Protocol* (PPTP) fazem uso apenas da autenticação, enquanto o *IP Security* (IPSec) pode fazer uso da autenticação, da integridade e da confidencialidade dos pacotes.

Além do tunelamento e da criptografia, outras características chaves que devem ser consideradas na implementação de uma VPN são o gerenciamento e o controle de tráfego, que serão analisados na seção 9.6.

9.4 O Tunelamento

O tunelamento consiste do encapsulamento dos dados do usuário (*payload*) em pacotes IP, de modo que eles podem ser roteados entre múltiplos protocolos. Esses pacotes são então desempacotados no outro lado do túnel, de modo a permitir assim que os usuários autorizados, sejam eles móveis ou não, utilizem a rede da organização a qualquer momento e a partir de qualquer localidade. O tunelamento é importante porque um túnel IP pode acomodar qualquer tipo de *payload*, e o usuário móvel pode utilizar a VPN para acessar transparentemente a rede da organização, seja ela baseada em IP, *Internet Packet Exchange* (IPX), AppleTalk ou outros.

9.5 Os Protocolos de Tunelamento

O tunelamento pode ser realizado nas camadas 2 e 3, sendo que ambas possuem suas vantagens e desvantagens. Alguns dos protocolos propostos para a camada 2 são o *Point-to-Point Tunneling Protocol* (PPTP), *Layer 2 Forwarding* (L2F), *Layer 2 Tunneling Protocol* (L2TP) e o *Virtual Tunneling Protocol* (VTP). O *Mobile IP* e o *Internet Security* (IPSec) são alguns protocolos utilizados na camada 3 [BAT 98].

O tunelamento no nível 2, por atuar em um nível mais baixo do modelo ISO/OSI, possui algumas vantagens com relação ao tunelamento no nível 3, tais como a simplicidade, a compressão e cifragem fim-a-fim, e a inicialização bidirecional do túnel. As suas características fazem com que ele seja indicado principalmente para os acessos discados ou para os que têm os seus custos relacionados à sua utilização, ou seja, quando os custos são definidos de acordo com a quantidade de *bytes* que trafegam por essa VPN. Já as suas desvantagens são a padronização ainda em desenvolvimento e as questões relativas à escalabilidade, à confiabilidade, e à sua segurança [BAY 98].

Um problema de escalabilidade pode ser visto quando a segurança é provida no L2TP, geralmente pelo IPSec, onde o cabeçalho sofre um *overhead* considerável, como pode ser visto na Figura 9.7.

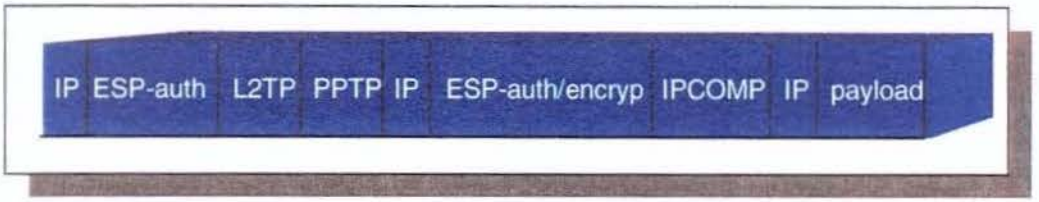


Figura 9.7: *Overhead* que pode ocorrer no cabeçalho de um pacote L2TP.

O *IP Payload Compression Protocol* (IPCOMP) provê a compressão dos dados. Esse *overhead* influi diretamente na fragmentação e na perda de pacotes, prejudicando assim o desempenho do acesso VPN.

Por sua vez, o tunelamento no nível 3 possui as vantagens da escalabilidade, da segurança e da confiabilidade, enquanto suas principais desvantagens são a limitação do número de fabricantes e a complexidade em seu desenvolvimento [BAT 98]. Porém, essas desvantagens estão sendo minimizadas rapidamente, como pode ser observado pelo grande número de fabricantes que implementam o IPSec em seus produtos, tornando-o padrão *de facto* das VPNs.

9.5.1 PPTP e L2TP

O *Layer 2 Tunneling Protocol* (L2TP) é definido pelo *Internet Engineering Task Force* (IETF), e é baseado no *Layer 2 Forwarding* (L2F) da Cisco Systems e no *Point-to-Point Tunneling Protocol* (PPTP) da Microsoft. Ele suporta o tunelamento e a autenticação do usuário (por exemplo, pelo CHAP ou pelo PAP), sendo bastante utilizado para o encapsulamento de pacotes PPP, utilizado em conexões discadas. Um ponto a ser considerado nos dois protocolos é que a confidencialidade, integridade e autenticidade dos pontos que se comunicam devem ser providos por um outro protocolo, normalmente o IPSec.

Uma diferença entre o L2TP e o PPTP é que o L2TP pode ser transparente para o usuário, no sentido de que esse tipo de tunelamento pode ser iniciado no *gateway* VPN de um provedor VPN (Figura 9.8). Quando o PPTP é utilizado, essa abordagem é diferente, e o tunelamento é sempre iniciado no próprio cliente (Figura 9.9). Com isso, o PPTP é mais indicado para a utilização em *laptops*, por exemplo, onde o usuário poderá se conectar à rede da organização via VPN, através do PPTP, a partir de qualquer lugar. Apesar disso, um cliente L2TP também pode

ser instalado no equipamento do usuário, dispensando assim o provedor VPN para o protocolo, como pode ser visto na Figura 9.10.

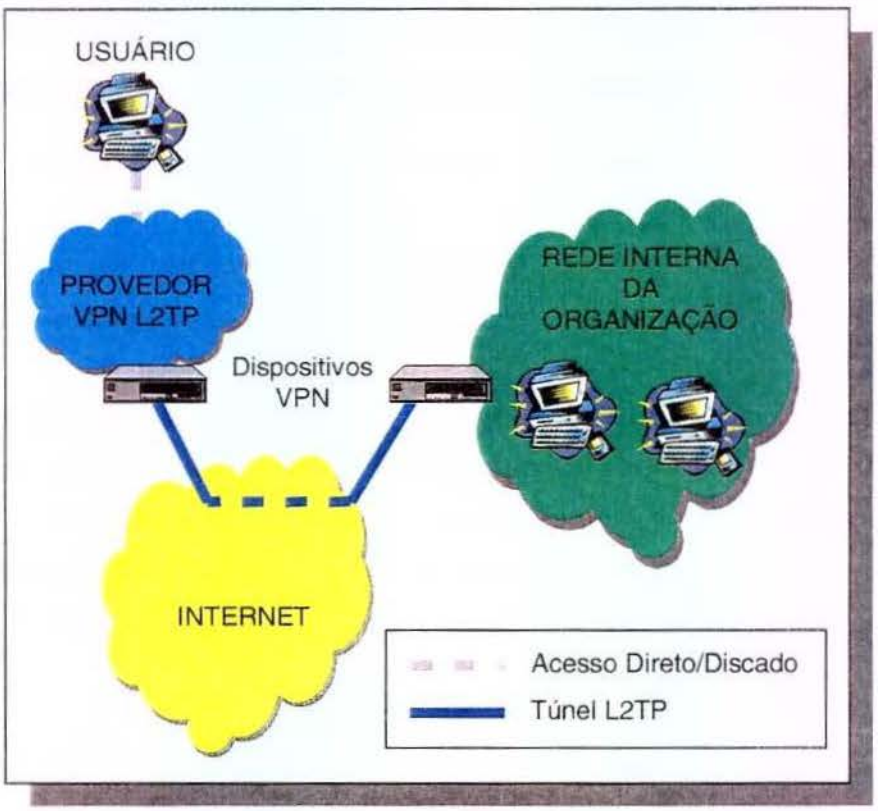


Figura 9.8: O protocolo L2TP sendo utilizado através de um provedor VPN.

O L2TP é utilizado principalmente para o tráfego de protocolos não IP sobre uma rede pública baseada em IP.

9.5.2 IPSec

Stallings [STA 98] mostra o surgimento do *Internet Protocol Security* (IPSec) em 1995, como sendo uma resposta à necessidade de segurança contra o monitoramento e o controle do tráfego da rede não autorizados. A autenticação e a cifragem definidas pelo IPSec são independentes das

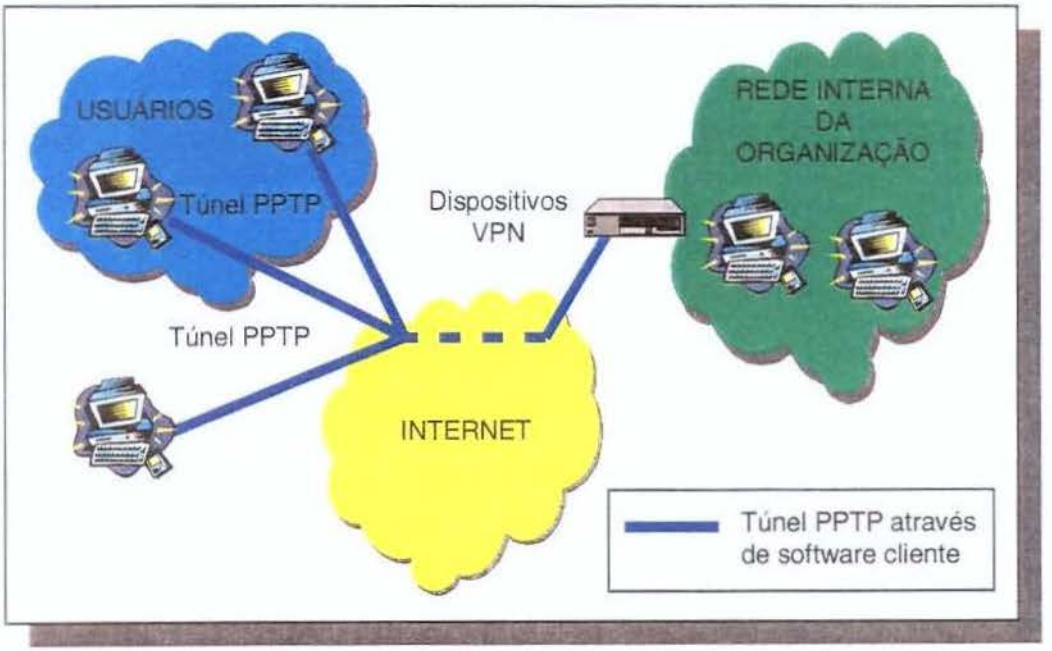


Figura 9.9: O protocolo PPTP sendo utilizado através de um software cliente.

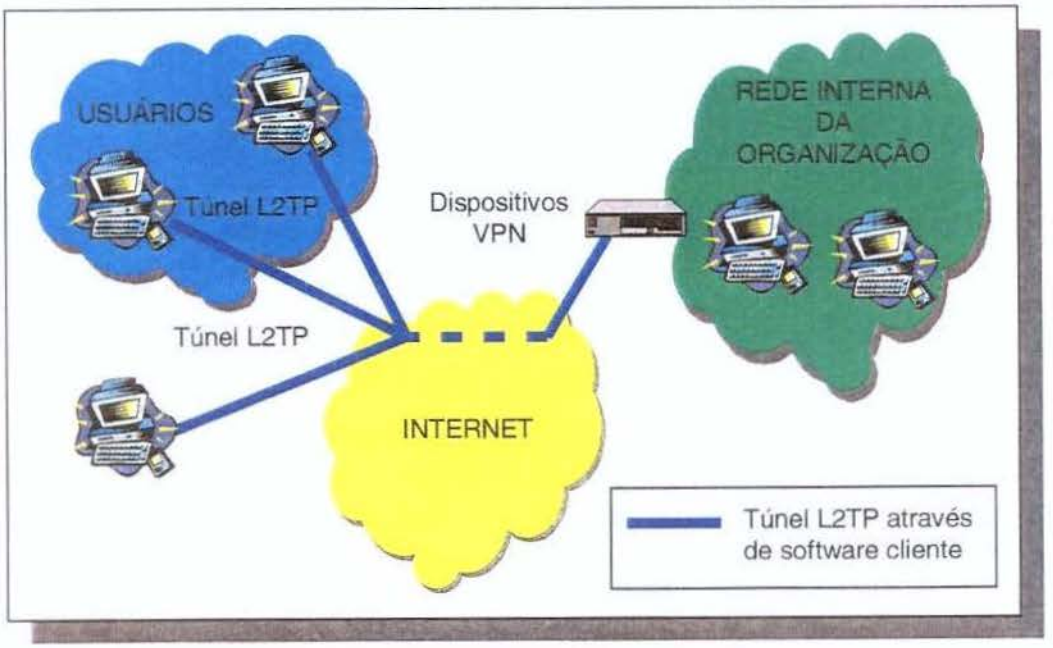


Figura 9.10: O protocolo L2TP sendo utilizado através de um software cliente.

versões do protocolo IP (versões 4 ou 6), e o protocolo vem se tornando o verdadeiro padrão utilizado pelos túneis VPN. Alguns ataques teóricos foram discutidos por Bellovin em [BEL 97], principalmente a possibilidade de se adquirir informações dos cabeçalhos IPSec.

O IPSec é composto por 3 funcionalidades principais [TIM 98]:

- Cabeçalho de autenticação (*Authentication Header* – AH), que provê a integridade dos pacotes e a garantia de sua origem;
- Cabeçalho de encapsulamento do *payload* (*Encapsulation Security Payload* – ESP), que provê a confidencialidade dos pacotes que trafegam através da rede pública;
- Protocolo de negociação e troca de chaves (*Internet Key Exchange* – IKE), que permite a negociação da comunicação entre as organizações de modo seguro.

A autenticação pode ser provida pelo AH e também pelo ESP, sendo que a diferença entre eles é que a autenticação provida pelo ESP não protege os cabeçalhos IP que antecedem o encapsulamento ESP. Já o AH faz a autenticação desse cabeçalho IP e também do encapsulamento ESP.

9.5.2.1 Os Dois Modos do IPSec

O IPSec trabalha de duas maneiras [STA 98]:

- *Transport Mode* – modo nativo, onde há a transmissão direta dos dados protegidos pelo IPSec entre os *hosts*. A cifragem e a autenticação são realizadas no *payload* do pacote IP, e não no cabeçalho IP (Figura 9.11). É utilizado em dispositivos que incorporam o IPSec na implementação do TCP/IP (Figura 9.12), como nos *softwares* clientes IPSec. Algumas modalidades que utilizam o modo de transporte são o *gateway-to-gateway VPN* (Figura 9.2), *client-to-gateway VPN* (Figura 9.3 e 9.4) e o *remote-access VPN* (Figura 9.5);
- *Tunnel Mode* – é geralmente utilizado pelos *gateways* IPSec, que manipulam o tráfego IP gerado por *hosts* que não suportam o IPSec, como nas modalidades que podem ser observados nas Figuras 9.1 e 9.6. O *gateway* encapsula o pacote IP com a criptografia do IPSec, incluindo o cabeçalho IP original. Ele então adiciona um novo cabeçalho IP no pacote de dados (Figura 9.13), e o envia através da rede pública para o segundo *gateway*, onde a informação é decifrada e enviada para o *host* destinatário, em sua forma original (Figura 9.14).

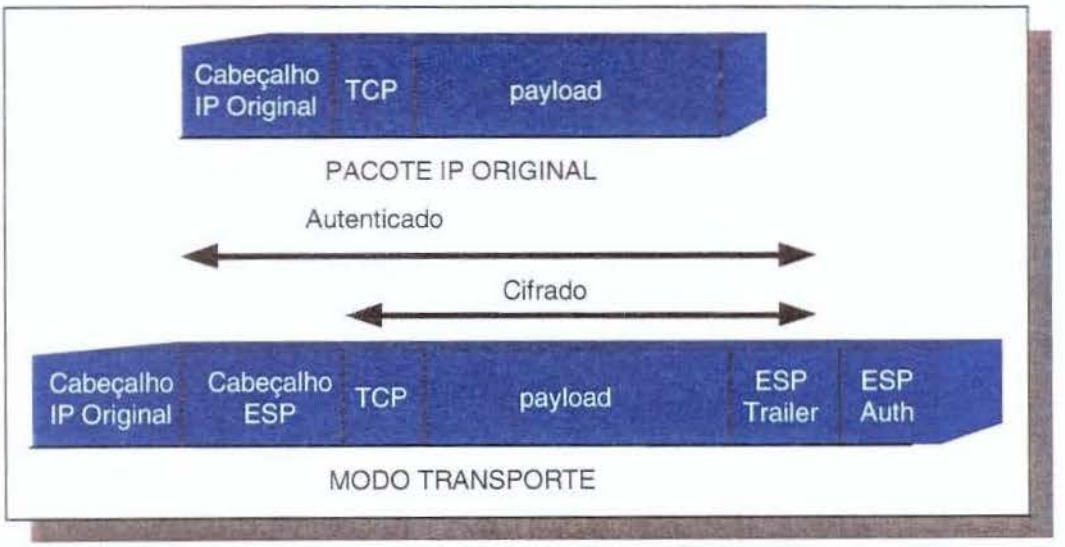


Figura 9.11: A cifragem e a autenticação no modo transporte do IPSec.

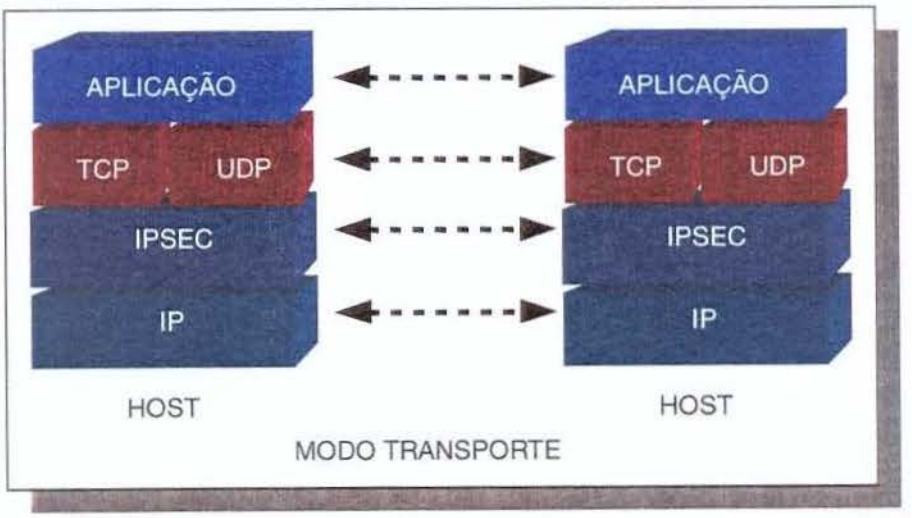


Figura 9.12: No modo transporte o IPsec é incorporado na pilha TCP/IP.

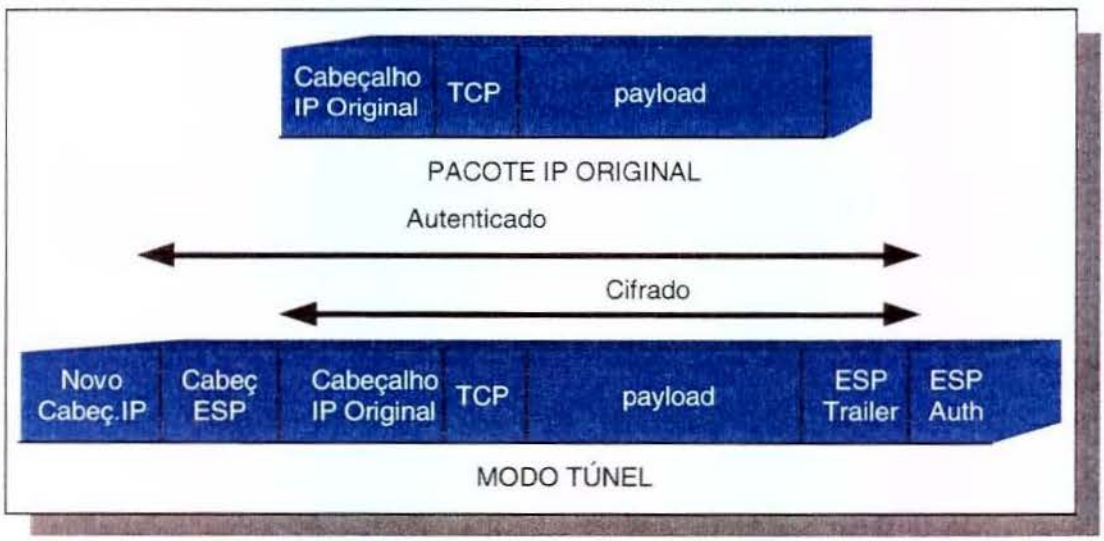


Figura 9.13: A cifragem e a autenticação no modo túnel do IPSec.

9.5.2.2 A Negociação dos Parâmetros do IPSec

O início de uma conexão segura é através do *Security Association* (SA). Ele permite que os usuários negociem um conjunto comum de atributos de segurança de um modo seguro, e contém uma série de informações que devem ser compartilhadas e aceitas por ambas as partes, como se fosse um contrato. O SA define como os sistemas que estão se comunicando utilizam os serviços de segurança, incluindo informações sobre o protocolo de segurança, o algoritmo de autenticação e o algoritmo de cifragem, incluindo ainda informações sobre o fluxo de dados, tempo de vida e o número de seqüência, que visa inibir o *replay-attack* [ENT 00].

O SA é unidirecional, ou seja, para cada par de sistemas se comunicando, existem pelo menos duas conexões seguras. Um SA pode utilizar o ESP ou o AH, porém não os dois. Caso

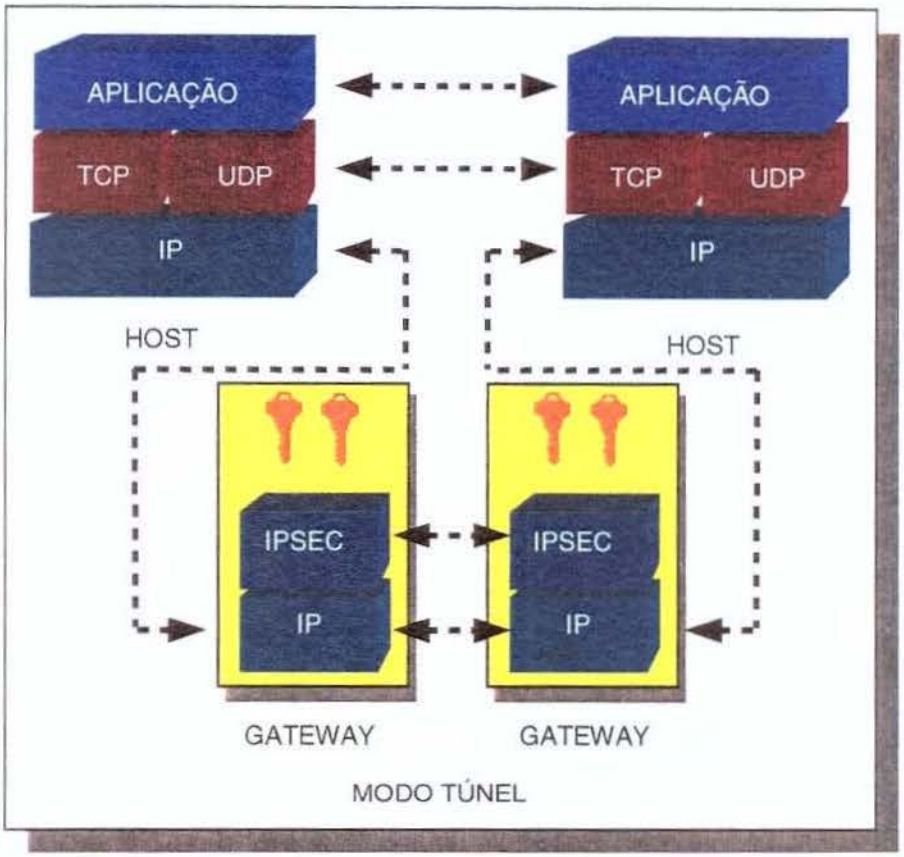


Figura 9.14: No modo túnel o IPsec é implementado no *gateway*.

seja necessário utilizar ambos, serão necessários dois SAs para cada um, somando no total quatro SAs para as conexões bidirecionais [ENT 00].

O SA é identificado pela combinação de [ENT 00]:

- *Security Parameter Index* (SPI), um número único aleatório;
- O endereço IP de destino do pacote;
- O protocolo de segurança a ser utilizado (AH ou ESP).

9.5.2.3 O Gerenciamento das Chaves

O gerenciamento de chaves é um dos processos mais importantes do IPSec, e grande parte da segurança da comunicação reside nele, principalmente nas trocas iniciais das chaves. Um

esquema bem definido de trocas deve ser adotado para se evitar ataques do tipo *man-in-the-middle*, onde o *hacker* pode capturar as trocas de informações dos dois lados da comunicação, alterando-as de acordo com os seus objetivos.

O gerenciamento de chaves definido pelo IPSec é realizado pelo *Internet Key Exchange* (IKE), que é baseado no *Internet Security Association and Key Management Protocol* (ISAKMP) e no Oakley, que é o responsável pela troca de chaves.

O IKE está relacionado diretamente com a negociação dos *Security Associations* (SAs) e com a troca de chaves. O seu funcionamento possui 2 fases, onde na primeira fase o par estabelece um canal seguro para a criação do IKE SA, que é um SA utilizado para a negociação dos SAs (segunda fase) (Figura 9.15). A idéia de se dividir o processo em 2 fases consiste na eliminação da redundância em alguns pontos da negociação do SA, e no conseqüente ganho de tempo e processamento, já que um canal seguro já está estabelecido pela primeira fase da negociação.

O IKE provê 3 modos de troca de informações e estabelecimento de SAs [TIM 98]:

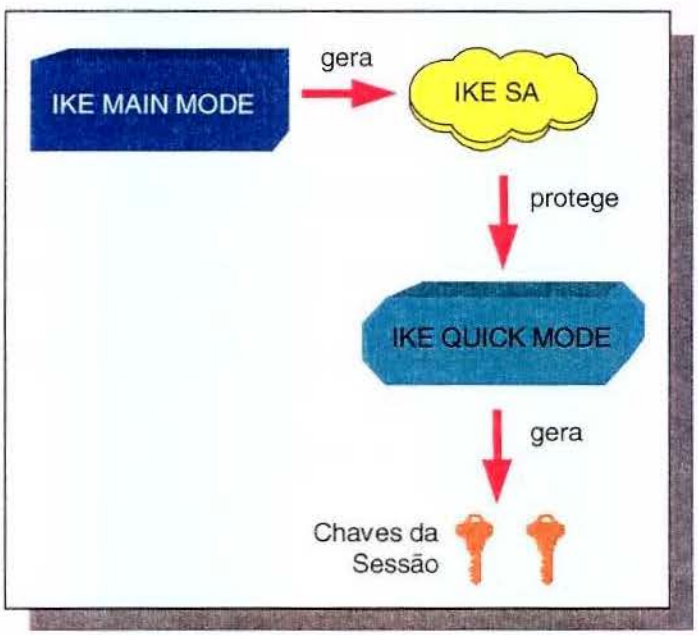


Figura 9.15: As fases até a negociação do SA.

- *Main Mode* – corresponde à fase 1 do IKE, e estabelece o canal seguro para a fase seguinte, gerando o IKE SA;

- *Aggressive Mode* – corresponde também à fase 1 do IKE, porém é mais simples e mai rápido do que o *main mode*, já que não provê a proteção das identidades das entidade que estão se comunicando. Isso ocorre porque as identidades são transmitidas juntament com as solicitações de negociação, sem que um canal seguro seja criado antes, estand assim susceptíveis a ataques *man-in-the-middle*;
- *Quick Mode* – corresponde à fase 2 do IKE, e é a comunicação estabelecida para a nego ciação do SA.

O *main mode* é composto por 3 fases, cada um com 2 mensagens. Na primeira fase, as duas partes envolvidas trocam informações sobre os algoritmos e *hashes* básicos a serem utilizados; na segunda fase, eles trocam as chaves públicas para uma negociação *Diffie-Helman* e passam os números aleatórios que a outra parte deve assinar e devolver para provar a sua identidade; e na terceira fase, eles verificam as identidades.

O *aggressive mode* provê os mesmos serviços do *main mode*, com a diferença de ser bem mais rápido, com apenas 2 fases, com 1 mensagem cada. Assim, no total são 3 trocas de mensagens, ao invés dos 6 requeridos pelo *main mode*. Isso ocorre porque este modo não provê a proteção da identidade das entidades participantes.

O *quick mode* utiliza o canal seguro estabelecido pela utilização do IKE SA, gerado pelo *main mode* ou pelo *aggressive mode*, para negociar os parâmetros da comunicação IPSec e trocar as chaves a serem utilizadas nessa comunicação. Como este modo já trabalha sob um canal seguro já estabelecido, todo o processo de negociação fica mais flexível e rápido, sendo composto por 3 trocas de mensagens, como no *aggressive mode*.

Após o SA ser negociado, as entidades estão aptas a trocarem informações por uma rede pública de modo seguro, formando assim o túnel VPN. A Figura 9.16 mostra os passos simplificados no estabelecimento de uma conexão VPN baseada em IPSec. Na primeira parte, o *gateway* verifica, através da política de segurança implementada, se o *host* pode criar um túnel virtual. Caso essa verificação seja positiva, o *gateway* inicia a negociação do *Security Association* da sessão, que pode ser vista na segunda parte da figura. Finalmente, na terceira, o *host* se comunica através do canal seguro criado.

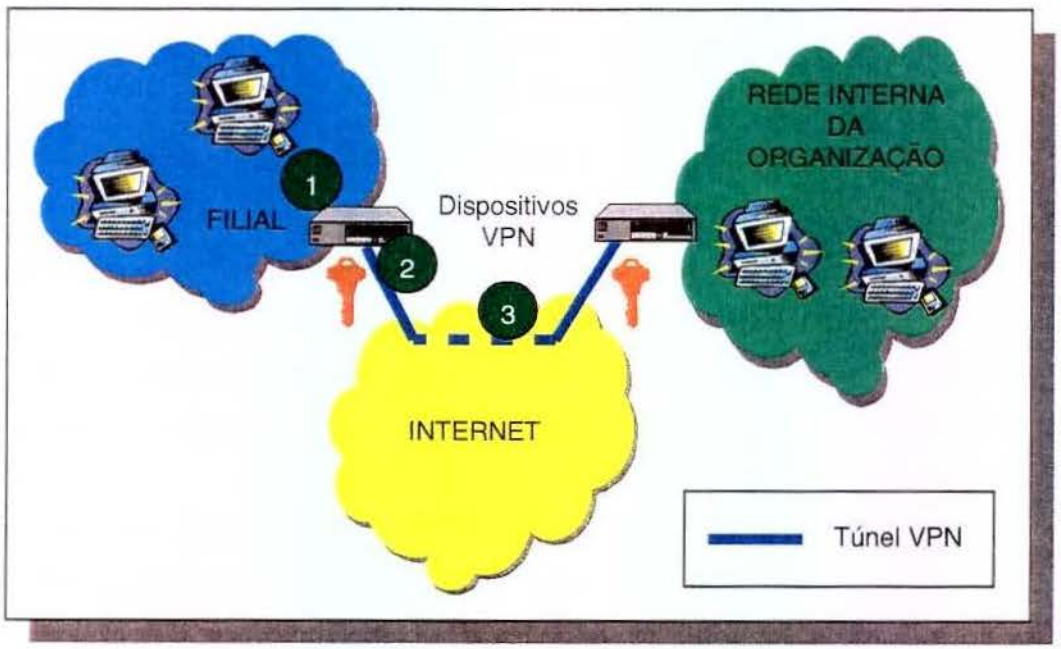


Figura 9.16: O estabelecimento de uma conexão VPN baseada em IPSec.

9.6 Gerenciamento e Controle de Tráfego

Além da segurança, que tem a função de realizar o controle de acesso e utiliza a criptografia para garantir a autenticidade dos usuários e das conexões e a privacidade e a integridade da comunicação, outros elementos são essenciais em uma VPN, como as que são citadas pela Check Point [CHE 98-3][CHE 98-4][CHE 98-5] - o controle de tráfego e o gerenciamento.

O controle de tráfego é essencial para que um dos principais problemas relacionados com a VPN, que é a qualidade de serviço, seja tratada. Esse controle é realizado fundamentalmente pelo gerenciamento de banda, que determina a largura de banda que cada protocolo pode utilizar, em busca do bom desempenho.

Já o gerenciamento tem como objetivo facilitar a integração da VPN com a política de segurança da organização, através do gerenciamento centralizado local ou remoto, além de facilitar também a escalabilidade da solução. Algumas ferramentas auxiliam nesse processo, como as utilizadas para o provisionamento dos serviços, para o monitoramento, para a detecção e solução dos problemas, para a contabilidade, e para a cobrança pela utilização da VPN. A contabilidade

pela utilização passa a ser importante, a partir do momento em que os serviços VPN passam a ser providos por empresas especializadas (provedores VPN), sendo que a cobrança pode ser baseada em alta confiabilidade, alto desempenho ou em níveis especiais de serviços. A garantia de qualidade de serviço também deve possuir o seu custo. A contabilidade também poderá ser baseada na importância dos pacotes, sendo que pacotes com maior garantia e desempenho no tráfego podem possuir valor maior do que pacotes considerados "normais" [BAY 98]. Alguns aspectos a serem considerados na contratação de serviços VPN são: área de cobertura, acesso, desempenho, segurança, gerenciamento, largura de banda e garantia de qualidade de serviço [HIF 98].

9.7 Obstáculos

Além das considerações relacionadas com a segurança, a VPN possui alguns obstáculos que devem ser analisados antes de sua implantação. Algumas dessas barreiras, que impedem a VPN de oferecer o mesmo nível de disponibilidade, desempenho e segurança das redes privadas, foram analisadas em [KIN 99]:

- Autenticação/Gerenciamento de Chaves – as diversas soluções utilizam variados mecanismos de autenticação, como os segredos compatilhados, os *tokens*, o *Radius* ou os certificados digitais, de modo que a compatibilidade fica comprometida. A *Public Key Infrastructure* (PKI) (seção 8.6) está ainda em fase de amadurecimento, onde o *Public Key Infrastructure Working Group* (PKIX) da IETF vêm trabalhando em busca da padronização das requisições, validações e formatos dos certificados digitais;

- Tolerância a Falhas – a necessidade de alta disponibilidade faz com que mecanismos de tolerância a falhas precisem ser desenvolvidos. O IPSec, por exemplo, não possui suporte a esse tipo de mecanismo;

- Desempenho – os algoritmos de chave pública utilizam altos recursos de processamento. Os computadores normais não possuem capacidade de I/O para realizar essa tarefa, e a única solução viável é a combinação do *Network Interface Card* (NIC) com a função de criptografia, que forma o equivalente à caixa-preta. A desfragmentação de pacotes também influi no desempenho da VPN, devido às sucessivas adições de cabeçalhos nos pacotes originais [SAL 99]. Por exemplo, um pacote IPX de uma LAN pode ser inserido em um pacote IP para trafegar sobre a Internet. Esse pacote IP, por sua vez, pode receber

um cabeçalho PPP, que por sua vez pode receber um cabeçalho PPTP para o tunelamento e outro cabeçalho IPSec para a cifragem desse pacote. Esse processo todo pode fazer com que o limite do tamanho do pacote seja ultrapassado, e quando isso ocorre o pacote é separado em dois novos pacotes. Como resultado, a fragmentação ocorre, e a quantidade de pacotes que trafegam entre os *sites* quando a VPN é utilizada é maior do que quando um *backbone* IP é utilizado, causando assim o aumento do tráfego. Alguns testes indicam que o aumento dos pacotes devido à fragmentação é de cerca de 30% [SAL99];

- Transporte Confiável – empresas como a Bay Networks e a Internet Devices estão tentando prover serviços de transporte confiáveis através do protocolo *ReSerVation Protocol* (RSVP), que oferece qualidade de serviço e reserva de banda pela alocação de recursos da rede. Outro protocolo em desenvolvimento é o *MultiProtocol Label Switching* (MPLS), que envolve o uso de diferentes *labels* ou *tags*, que permitem múltiplos caminhos;

- Posicionamento na Rede – envolve a análise do posicionamento do *gateway* VPN na arquitetura da organização. Isso se relaciona com uma série de fatores, como a filtragem de pacotes cifrados e o drible da política de segurança, e será discutido com maiores detalhes na seção 11.2. De fato, o seu posicionamento dentro da arquitetura de segurança influi diretamente no nível de segurança da organização, de forma que as várias alternativas devem ser analisadas, o que pode ser verificado no *firewall* cooperativo;

- Endereçamento/Roteamento – endereçamentos, que podem ser endereços IPs válidos ou não, precisam ser considerados, e alterações nas tabelas de roteamento são essenciais, como no caso do *gateway-to-gateway VPN*, onde um *gateway* deve se comunicar apenas com o outro *gateway*. O *Network Address Translation* (NAT) também deve ser considerado, pois ele influi diretamente no roteamento da solução. De fato, o NAT é incompatível com o *Authentication Header* (AH) do IPSec, seja ele utilizado no modo de transporte ou no modo túnel. Isso ocorre porque o AH realiza a autenticação do cabeçalho IP, que é assinado digitalmente para a verificação de sua integridade, e ele é modificado pelo NAT. Isso não ocorre quando o *Encapsulation Security Payload* (ESP) é utilizado, já que o cabeçalho IP não é autenticado, não ocasionando assim problemas de integridade;

- Administração/Gerenciamento – alguns produtos utilizam interfaces GUI cliente/servidor, e outros utilizam uma interface baseada em Web. Um canal seguro é essencial para a administração e gerenciamento da VPN. Um problema é que não é possível administrar

diversas VPNs a partir de uma mesma interface, a não ser que elas sejam do mesmo fabricante. O gerenciamento do cliente VPN também é complicado, desde a sua instalação até a sua distribuição, configuração e administração [NAK 00];

- Interoperabilidade – o IPSec veio para prover a interoperabilidade que está faltando nos produtos VPN. Porém, o que pode ser observado é que nem mesmo os produtos baseados no IPSec possuem uma comunicação compatível entre si, sendo que a compatibilidade entre eles é apenas parcial.

9.8 Conclusão

Este capítulo teve como objetivo apresentar as redes privadas virtuais (*Virtual Private Network* - VPN), mostrando os seus objetivos e as suas configurações. A utilização da rede pública traz consigo uma série de considerações de segurança, que são tratadas principalmente pelo protocolo IPSec, padrão *de facto* das VPNs. Os protocolos L2TP, PPTP e IPSec foram rapidamente discutidos, e o capítulo mostrou ainda a importância do gerenciamento e do controle de tráfego, além dos obstáculos a serem enfrentados na implementação de uma VPN.

Capítulo 10

Autenticação

A autenticação possui um papel fundamental para a segurança dos sistemas, ao validar a identificação dos usuários, concedendo-lhes a autorização para o acesso aos recursos. A autenticação pode ser realizada baseada em alguma coisa que o usuário sabe, em alguma coisa que o usuário tem, ou em alguma característica do usuário, como será visto neste capítulo. O capítulo mostra também os pontos importantes a serem considerados no controle de acesso, que tem como base a autenticação dos usuários, e discute também as vantagens e desvantagens do *Single Sign-On* (SSO), que tenta resolver um dos maiores problemas relacionados à autenticação.

10.1 A Identificação e a Autorização

O acesso às diversas tecnologias discutidas nos capítulos anteriores e aos recursos das organizações depende fundamentalmente de um processo de verificação do usuário, que deve ser realizado de uma maneira na qual apenas o legítimo usuário tenha acesso a esses sistemas e recursos. As funções responsáveis por essa verificação são a identificação e a autenticação, que formam a primeira linha de defesa em muitos sistemas. A identificação é a função na qual o usuário declara uma determinada identidade para um sistema. A autenticação é a função responsável pela validação dessa declaração de identidade do usuário. A segurança desse processo de validação depende de uma série de considerações, tais como a forma de coleta dos dados de autenticação, o método de transmissão desses dados, e a garantia de que o usuário que já obteve a autorização seja o verdadeiro usuário [NIS 00].

A autenticação, ou a validação da identificação do usuário, pode ser realizada baseando-se em 3 métodos:

- Naquilo que o usuário sabe – senha, chave criptográfica ou *Personal Identification Number* (PIN);
- Naquilo que o usuário possui – *token*, cartão ou *smart card*;
- Naquilo que o usuário é – biometria (seção 10.1.3), ou seja, reconhecimento de voz, impressão digital, geometria das mãos, reconhecimento de retina, reconhecimento de íris, reconhecimento digital de assinaturas.

Todos esses métodos possuem seus pontos negativos, de forma que uma autenticação baseada em dois deles passa a ser recomendada em determinados tipos de acessos que exigem um maior grau de segurança – a chamada autenticação em dois fatores. Por exemplo, o usuário poderia utilizar o reconhecimento de retina adicionado ao uso de um *smart card* para o acesso à informações críticas da organização.

Será visto a seguir os pontos positivos e os pontos negativos que podem ser encontrados em cada um dos métodos de autenticação. Deve-se considerar no entanto que cada um deles possui seus custos, e o método ideal é sempre aquele que cumpre com os objetivos de segurança definidos pela organização. Por exemplo, a biometria é indicada para a autenticação em sistemas críticos, como as que estão sendo utilizadas pelo Departamento de Defesa dos Estados Unidos. Já esse método não seria eficaz nem justificável para o acesso à caixa postal de um usuário, por exemplo.

O artigo de Lober [LOB 97] trata da necessidade de uma autenticação forte para a proteção de bens e valores. Cita o caso de um *hacker* russo que fez transferências de $10 milhões no Citibank em 1994. O *hacker* conseguiu achar ou "adivinhar" diversas senhas. A conclusão foi a de que as senhas proviam um certo controle de acesso às transferências de fundos, porém a solução não provia a autenticação forte do usuário.

10.1.1 Autenticação Baseada Naquilo que o Usuário Sabe

É a autenticação baseada em algum conhecimento do usuário, onde os mais utilizados são as senhas. As chaves criptográficas podem ser inseridas nesta categoria, porém necessitam que o usuário tenha acesso ao dispositivo que realize as funções criptográficas. Como esses dispositivos devem pertencer ao usuário, eles serão discutidos na seção 10.1.2.

10.1.1.1 Senhas

As senhas constituem atualmente o método de autenticação mais utilizado, porém possuem alguns problemas, como poderão ser vistos a seguir. Isso está fazendo com que elas sejam cada vez mais substituídas por métodos mais eficientes, como a biometria (seção 10.1.3) ou os certificados digitais (seção 8.5).

Um ponto a favor de sua utilização é que os usuários e os administradores já estão familiarizados com a sua utilização, e a sua simples integração com os diversos tipos de sistemas faz com que ela seja fácil de ser implementada.

A fraqueza deste método está no fato de que a segurança depende da manutenção da confidencialidade da senha, que pode ser quebrada de diversas formas [LOB 97][KES 96]:

- Adivinhação ou "pesca" de senhas, como observar o que o usuário está digitando ou procurar pedaços de papéis que podem conter senhas;
- Quebra da confidencialidade, seja por intenção do próprio usuário, que pode compartilhar a sua senha com um colega, ou através de técnicas de engenharia social (seção 4.5.1);
- Monitoramento de senhas, através de *sniffers* de rede (seção 4.5.2);
- Acesso ao arquivo de senhas do usuário, que mesmo estando cifrado, pode ser quebrado com relativa facilidade, caso o algoritmo criptográfico esteja mal implementado;
- Ataques de força bruta contra o sistema, seja através de combinações de códigos ou através de ataque do dicionário;
- Utilização de técnicas mais avançadas, como o armazenamento em um arquivo de tudo o que o usuário digita através do teclado, que depois é enviado ao *hacker* (*logging spoofing*). Essa técnica pode ser facilmente utilizada caso o equipamento do usuário esteja contaminado com cavalos-de-tróia, como o Netbus;
- Monitoramento das emissões eletromagnéticas do monitor (efeito *tempest*), que não requer a presença física do *hacker*.

As senhas representam uma questão de extrema relevância dentro das organizações, sendo também um dos principais causadores de problemas relacionados ao suporte técnico. Uma política bem definida e eficiente de senhas pode minimizar profundamente as implicações de segu-

rança envolvidas, e alguns dos pontos a serem considerados nessa política podem ser vistos na seção 5.8.

Os problemas vistos anteriormente podem ser minimizados através de sistemas *one-time password*, onde as senhas são válidas apenas por uma única vez. Esses sistemas podem ser considerados mais seguros do que as senhas comuns, porém ainda podem ser explorados em alguns ataques [LOB 97]:

- *Man-in-the-middle* – o *hacker* se coloca entre o usuário e o servidor, de modo que ele pode capturar os pacotes, modificá-los e reenviá-los a ambos os lados da conexão;
- *Race* – ataque que requer sorte, tempo e conhecimento. O atacante monitora o número de bytes que passam na rede, e antes que o usuário envie o último byte, o atacante envia uma série de combinações para tentar se conectar no lugar do usuário. Funciona somente em protocolos que passam os dados byte a byte.

10.1.2 Autenticação Baseada Naquilo que o Usuário Possui

O segundo método de autenticação é aquele baseado em alguns dispositivos que pertencem ao usuário. Eles podem ser divididos entre dispositivos de memória (*memory token*) e dispositivos inteligentes (*smart tokens*), que serão vistos a seguir [NIS 00].

10.1.2.1 Dispositivos de Memória

Também conhecidos como *memory tokens*, os dispositivos de memória apenas armazenam, e não processam informações. São quase sempre utilizados em conjunto com senhas (combinação de alguma coisa que o usuário sabe e alguma coisa que o usuário tem, formando portanto uma autenticação de dois fatores). Um exemplo desse tipo de dispositivo são os cartões de bancos.

Alguns dispositivos que se baseiam apenas naquilo que o usuário possui podem ser exemplificados através dos cartões que fornecem acessos físicos a locais como salas e edifícios.

As desvantagens desse tipo de dispositivo estão relacionadas com o seu alto custo, devido à necessidade de um hardware específico, à dificuldade de administração, à possibilidade de perda e à insatisfação dos usuários com a sua manipulação.

10.1.2.2 Dispositivos Inteligentes

Os *smart tokens* são os *memory tokens* com circuitos integrados, que atuam no processamento de algumas informações. Eles podem ser divididos em categorias que levam em consideração as características físicas, a *interface* e o protocolo.

Quanto às características físicas, os dispositivos inteligentes podem ser divididos entre os *smart cards* e os outros tipos de dispositivos (parecidos com chaves, calculadoras ou outros objetos portáteis).

Quanto à *interface*, os dispositivos inteligentes podem possuir *interfaces* manuais (existe um dispositivo como teclas ou visores para a interação entre o usuário e o *token*) ou *interfaces* eletrônicas, que requerem um dispositivo de leitura, como é o caso dos *smart cards*.

Os protocolos que podem ser utilizados pelos *smart tokens* para a autenticação podem ser divididas em:

- Troca de senhas estáticas – o usuário se autentica no *token*, e o *token* autentica o usuário no sistema;
- Geração dinâmica de senhas – as senhas são alteradas temporariamente, de modo que, em *smart tokens* com *interface* estática, os usuários devem ler as informações do dispositivo e digitá-las no sistema para a autenticação. Em *smart tokens* com *interface* eletrônica, esse processo é feito automaticamente;
- Desafio-resposta – com esse protocolo, que se baseia na criptografia, o sistema envia um desafio ao usuário. A resposta é enviada ao sistema, que a avalia de acordo com o desafio corrente.

Os dispositivos inteligentes resolvem os problemas presentes nas senhas comuns, seja através da criptografia (que evita o monitoramento das senhas que passam pela rede), ou através da geração dinâmica das senhas (quem capturar a senha não pode reutilizá-la).

Quanto aos seus pontos negativos, tanto os dispositivos de memória quanto os dispositivos inteligentes são equivalentes. Além desses pontos, apresentados na seção anterior, eles podem também serem perdidos, roubados ou quebrados facilmente. Porém, os dispositivos inteligentes possuem um custo relativamente superior aos dispositivos de memória.

10.1.2.3 Autenticação Baseada Naquilo que o Usuário é

Foi visto que os problemas com os métodos de autenticação baseadas naquilo que o usuário sabe ou naquilo que o usuário possui variam entre o esquecimento ou a adivinhação da senha, e a perda do dispositivo responsável pela identificação. A proteção de informações críticas requer um método de autenticação onde as possibilidades de acessos indevidos sejam mínimas, de modo que a autenticação garanta a identificação do usuário em seu nível de acerto máximo. Esse método de autenticação, considerado mais seguro que os anteriores, é aquele baseado naquilo que o usuário é – a biometria.

A biometria é um método de autenticação que mede características físicas (o que nós somos) ou comportamentais (o que nós fazemos) de um indivíduo, comparando-o com os dados armazenados em um sistema, para confirmar a sua identidade [WOO 98]. Algumas dessas características podem ser: composição química do cheiro corporal, características faciais e emissões térmicas, características do olho, impressões digitais, geometria da mão, poros da pele, análise de assinaturas, padrões de escrita ou vozes. Dessas características, apenas 3 podem ser consideradas como sendo únicas, ou seja, não existem dois indivíduos com uma dessas características exatamente iguais: a retina, a íris do olho e a impressão digital.

A grande vantagem da biometria é que o reconhecimento é feito unicamente por aspectos humanos intrínsicos. Chaves, *tokens* ou cartões podem ser perdidos, falsificados, duplicados, roubados ou esquecidos. Senhas, códigos secretos e número de identificação pessoal (*Personal Identification Number – PIN*) podem ser esquecidos, divididos, comprometidos ou observados. Já a biometria não possui esses problemas, ao tratar de características individuais dos humanos.

A tecnologia biométrica mais comum que pode ser vista no mercado é o reconhecimento de impressão digital, porém uma das tecnologias com maior aceitação é o reconhecimento de face [SEC 99-5]. Essa tecnologia é facilmente integrada aos sistemas, e uma de suas vantagens é que, para o usuário, ela é higiênica, não é intrusiva e não justifica qualquer resistência, pois não é necessário que o usuário fique em uma determinada posição, nem que ele faça alguma coisa em particular. O processo de autenticação é assim realizada de um modo natural para o usuário. Com relação à segurança, os dados das imagens são cifrados e enviados através da rede, de modo seguro, até o servidor, onde as imagens são comparadas e a autorização é concedida. Isso evita que uma imagem roubada possa ser utilizada para acessos não autorizados.

A geometria das mãos é uma das tecnologias de biometria que possui uma das menores taxas de erro de autenticação, com uma taxa de 0,1% de falsas rejeições. A leitura da geometria é rápida, geralmente em um segundo, e as características que são medidas incluem a largura, o comprimento, a espessura e a área de superfície. Essas características são armazenadas através de uma representação matemática, que é atualizada a cada leitura, para garantir a leitura de mãos de crianças ou de pessoas que sofrem alterações no peso [SEC 99-5].

No entanto, alguns problemas podem ocorrer com a biometria. Algumas organizações relataram que um dos principais problemas estão relacionados à higiene e ao medo dos dispositivos biométricos causarem problemas de saúde. Por exemplo, o dispositivo de impressão digital deve ser limpado constantemente, pois, além dos problemas com a higiene, o acúmulo de sujeiras influi diretamente no nível de acerto da autenticação, que pode diminuir consideravelmente. O problema relacionado com o medo dos usuários pode ser observado em dispositivos de leitura de retina ou de íris do olho, onde alguns usuários podem acreditar que o laser ou a luz pode fazer mal à sua saúde.

10.2 Controle de Acesso

Dentro de uma organização, as informações possuem diversos níveis de acesso, ou seja, uma informação relevante para o trabalho de um funcionário pode não ser relevante para o trabalho de um outro funcionário. Uma informação confidencial que pode ser acessada somente pelos gerentes, por exemplo, não pode chegar aos demais funcionários. O controle de acesso lógico, designado ao controle realizado sobre recursos computacionais, cuida do acesso aos diversos níveis existentes. Os elementos a serem considerados no controle, que podem ser utilizados individualmente ou em conjunto, são: identificação, função, localização, tempo, transação, serviços (controle de licenças de software, limites em transações em caixas eletrônicos, tipo de acesso em computadores (como a permissão para enviar *e-mails* mas proibição para conexão com outras máquinas)), direitos (leitura, gravação, criação, eliminação, busca, execução). Mais do que isso, o controle de acesso controla também o acesso aos diversos tipos de recursos da organização, tais como serviços, programas e acessos via modens, por exemplo.

Pode-se considerar que o acesso é a habilidade de realizar algo com recursos computacionais, e a autorização é a permissão, dada direta ou indiretamente, pelo sistema ou pelo dono do

recurso, para a utilização desses recursos. A autenticação é a responsável pela garantia de que o usuário é realmente quem ele declara ser.

O controle de acesso lógico é o responsável pela:

- Proteção contra modificações ou manipulações não-autorizadas de sistemas operacionais e outros sistemas (software), garantindo assim a sua integridade e a sua disponibilidade;
- Garantia da integridade e disponibilidade das informações, ao restringir o número de usuários e processos que acessam determinados tipos de informações;
- Confidencialidade das informações, que não podem chegar a usuários não-autorizados.

Além dos métodos de controle de acesso baseados na autenticação, vistos nas seções anteriores, outros podem ser utilizados:

- Listas de controle de acesso (*Access Control Lists* – ACLs);
- Interfaces com usuários – comandos somente através de menus ou através de um *shell* restritivo que aceita somente comandos específicos; uso de banco de dados, onde os usuários só podem acessar os dados desse banco de dados; restrição física, como os utilizados pelos caixas eletrônicos, que restringem a entrada de dados através de um teclado numérico único;
- *Labels*, como por exemplo, dados de propriedade da organização não podem ser acessadas por terceiros, e os dados públicos podem ser acessados por todos.

O controle de acesso externo é realizado entre os recursos a serem protegidos e as pessoas, sistemas ou serviços externos. Um dos principais métodos é a utilização de um dispositivo físico, como um computador, para separar os recursos internos dos externos. Alguns exemplos são o *dial-back modem*, que realiza a checagem do número de telefone para acessos discados, evitando a utilização do acesso remoto por usuários não autorizados. Outros exemplos são os *gateways* seguros ou os *firewalls*, vistos no capítulo 6.

Uma consideração importante é que o controle de acesso pode ser enfraquecido por um simples modem em um dos *workstations* da organização. Os usuários móveis estão aumentando a cada dia, e um controle de acesso bem definido para esses usuários é necessário para que problemas não sejam causados à organização. Outros problemas podem ocorrer quando um equipamento está conectado a uma rede, principalmente quando os usuários armazenam dados sensitivos em seu disco rígido. O próprio ambiente Windows utiliza *caches* locais, que podem

ser lidos através de um editor de discos. O controle de acesso deve levar em consideração esses pontos, geralmente através do uso de criptografia no disco rígido [SEC 98-2].

10.3 Single Sign-On (SSO)

Um dos principais pontos em um ambiente cooperativo é que a complexidade das conexões aumenta a cada novo integrante. Com isso, os usuários passam a acessar diversas aplicações e recursos de múltiplas plataformas, aumentando ainda mais a complexidade envolvida. Tudo isso, aliado ao aumento da utilização da Internet/Intranet, traz como conseqüência um número bastante grande de senhas que cada usuário deve utilizar para o acesso a esses recursos, aumentando também a necessidade de um método seguro de autenticação e autorização de serviços, principalmente para o perímetro externo da rede.

Uma conseqüência dessa necessidade do usuário, de lembrar de vários nomes de acesso e senhas, é o aumento dos riscos dele guardar a senha sob sua mesa ou confundir as senhas e necessitar de ajuda do *help-desk*, causando perda de produtividade e desperdício de recursos. Além disso, a administração de todas as senhas de cada usuário torna-se um processo complicado, e uma modificação nos dados de um usuário pode resultar na necessidade de atualização de múltiplas base de dados dos diversos aplicativos. Para combater toda essa complexidade e esse processo trabalhoso, que influi diretamente na segurança e na produtividade dos usuários e dos administradores de sistemas, uma alternativa é o *Single Sign-On* (SSO).

O SSO surgiu como um método de identificação e autorização que permite uma administração consistente, de modo que os usuários podem acessar os diversos sistemas de modo transparente e unificado, através de uma única autenticação. A definição do SSO, por sua vez, traz suas próprias implicações de segurança, já que através de uma única senha o usuário pode acessar diversos sistemas, significando que, caso essa senha seja comprometida, todos os sistemas podem sofrer com isso [TRI 98]. Uma solução SSO pode utilizar diversas formas de autenticação, como certificados digitais, *smart cards*, *tokens* e biometria. Algumas soluções SSO tratam somente da autenticação, ficando a autorização a cargo dos próprios serviços ou aplicação [CAR 99].

As principais características de um SSO são [TRI 98]:

- Combinação de nome de usuário e senha únicos;

- Único método de administração, centralizado ou descentralizado, onde as mudanças são propagadas através dos diversos sistemas da organização;
- Segurança robusta nas sessões de *logon* e no armazenamento das informações do usuário e da sua senha;
- Integração das regras de autorização nas múltiplas aplicações.

As primeiras soluções que utilizaram as características de um SSO foram o *Kerberos*, e mesmo os *scripts* escritos para os *workstations*. O *Kerberos* é baseado em *tickets* e credenciais, e a conexão inicial é feita em um servidor central de autenticação. A senha nunca trafega pela rede, eliminando assim as chances de ataques *replay attack* e *man-in-the-middle*. A desvantagem do *Kerberos* é que os clientes tem que ser "kerberizados", ou seja, devem possuir a implementação do protocolo, para que ele possa iniciar uma requisição de autenticação. Além disso, todos os sistemas e aplicações devem estar habilitados a aceitar *tickets* em vez do sistema tradicional baseado em senhas [TRI 98].

A outra solução, que são os *scripts* nos *workstations*, é baseada na definição da política de acesso no próprio equipamento do usuário. A vantagem dessa solução é a desnecessidade de alteração das aplicações existentes. As desvantagens incluem a necessidade do usuário utilizar somente aquele *workstation*, o fato da segurança se basear no próprio equipamento (segurança física), e a vulnerabilidade dos *scripts*, que podem ser modificados pelos próprios usuários. Com isso, a política de segurança pode ser driblada, e acessos a recursos inicialmente proibidos podem ser obtidos. Além disso, uma outra desvantagem é com relação à administração desses *scripts*, que é evidente quando o protocolo de autenticação sofre uma alteração. Os custos com a administração são altos, pois todos os *workstations* são afetados e necessitam de uma atualização [TRI 98].

Uma das soluções SSO utilizam serviços de autenticação baseados na rede, onde o usuário inicialmente se conecta ao servidor de autenticação, e logo após requisita os acessos a sistemas individuais ou aplicações. Porém, isso resolve somente os problemas do usuário, aumentando a importância de uma administração eficiente, já que o administrador deve cuidar de múltiplas tabelas e bases de dados dos sistemas e serviços, além do servidor de autenticação central [TRI 98].

Algumas considerações de segurança relacionadas com o SSO são [TRI 98]:

- A identificação e a senha única faz com que, caso uma senha seja descoberta, o acesso a todos os serviços seja permitido;
- O repositório central dos dados do usuário, entre eles o nome de acesso e a senha, era o objetivo dos administradores, mas passou a ser também o objetivo dos *hackers*, que agora possuem um único ponto de invasão;
- O serviço de autenticação forma um único ponto de falha, onde um ataque ou uma falha faz com que todos os serviços sejam comprometidos ou se tornem indisponíveis. A replicação do serviço central de autenticação é portanto importante para a manutenção da disponibilidade dos serviços.

Algumas questões que devem ser analisadas na implementação de um SSO são [TRI 98]:

- Existe uma política para garantir senhas fortes e que sejam regularmente modificados?
- Re-autenticação é necessário em certas funções, como nas transações que ultrapassam um certo valor?
- Existe controle de *time-out* que pede a re-autenticação em caso de um determinado tempo de inatividade?
- Existem *logs*, alarmes e travas?
- Como tentativas de conexão inválidas são detectadas, reportadas e manipuladas?
- A política é consistente entre as plataformas e aplicações?

Os resultados do sucesso na implementação do SSO são o aumento na produtividade dos usuários e dos administradores de sistemas. Os usuários ganham acesso mais fácil aos recursos, e os problemas com senhas, que resultam em utilização do *help-desk* (60 a 70% das chamadas, segundo estudos), são minimizados. Os administradores também ganham em produtividade, com a possibilidade de padronização da política de nomes de acesso/senhas e a aplicação consistente da política de segurança [TRI 98].

Uma alternativa interessante ao SSO, aplicável principalmente quando o problema da organização é apenas quanto aos custos com suporte técnico devido aos problemas com as senhas, é a sincronização de senhas. Essa solução é menos complexa do que o SSO, sendo que a principal diferença é que o usuário tem que se autenticar em cada serviço através de uma única

senha. Quando uma senha é alterada, essa mudança é propagada para todos os servidores, através de um agente de servidor [CAR 99].

A *Public Key Infrastructure* (PKI), visto na seção 8.6, também pode ser considerado um SSO, pois a autenticação dos usuários pode ser feita através do certificado digital. O usuário poderia acessar os recursos através desse certificado digital, porém, para que isso seja possível, é necessário que esses recursos sejam compatíveis com a PKI. Através dos certificados, os sistemas de autenticação podem ser integrados em uma infra-estrutura única. De fato, devido ao alto nível de segurança proporcionado pela criptografia de chaves públicas, a PKI possui uma importância bastante grande dentro da estratégia de segurança de qualquer organização, e pode ser considerado uma solução ideal dentro de um ambiente cooperativo, principalmente por facilitar a autenticação e o não-repúdio, além de ser capaz de prover a confidencialidade das informações.

10.4 Conclusão

O controle de acesso, baseado na autenticação e na autorização dos usuários, é um componente essencial para a segurança das organizações. Este capítulo analisou os diversos aspectos envolvidos, apresentando os principais métodos utilizados no controle de acesso, além dos métodos de autenticação existentes. A autenticação pode ser baseada em alguma coisa que o usuário sabe, em alguma coisa que o usuário possui, ou em alguma característica do usuário. A utilização de dois desses métodos aumenta o nível de segurança da autenticação, sendo chamada de autenticação de dois fatores. As senhas, o método de autenticação mais utilizado atualmente, traz uma série de problemas, sejam elas de segurança ou de produtividade, tanto do usuário quanto dos administradores de sistemas. O *Single Sign-On* (SSO) é um sistema que visa minimizar esses problemas, não só de senhas, mas de qualquer outro método de autenticação, ao eliminar a necessidade de múltiplas autenticações. A *Public Key Infrastructure* (PKI), ao utilizar a criptografia assimétrica, que garante um alto grau de segurança (se corretamente implementado), também resolve muitos dos problemas que envolvem a autenticação dos usuários, constituindo assim um importante elemento dentro da estratégia de segurança de uma organização.

Capítulo 11

As Configurações de um Ambiente Cooperativo

Este capítulo tem como objetivo apresentar os diversos cenários que representam as redes das organizações, que através de sua evolução (aumento dos números de conexões) chegam até a formação do ambiente cooperativo. Será visto que a complexidade aumenta a cada nova conexão, o que exige uma análise profunda das tecnologias necessárias que serão utilizadas na arquitetura de segurança da organização. Este capítulo analisa as diversas configurações de componentes importantes para a segurança da organização, como o *firewall*, o *Virtual Private Network* (VPN), o *Intrusion Detection System* (IDS) e a *Public Key Infrastructure* (PKI), de acordo com as necessidades que vão surgindo com a evolução das conexões. As discussões desse capítulo culminam com a arquitetura do *firewall* cooperativo, que é conceituado no próximo capítulo.

11.1 Os Cenários até o Ambiente Cooperativo

A arquitetura do *firewall* cooperativo será apresentada de acordo com um exemplo de ambiente cooperativo, através da análise das necessidades, dos problemas e das respectivas soluções propostas. Essa apresentação será realizada de modo gradual, ou seja, será apresentada a evolução de uma rede e das necessidades de proteção, até a formação do ambiente cooperativo.

Uma organização típica tem no início uma rede com o objetivo de conectar seus recursos internamente (figura 11.1), em busca de facilitar as tarefas básicas da organização. Nesse primeiro

passo evolutivo da rede, a organização ainda não é conectada a uma rede pública, ou seja, ainda não existiam acessos externos, apenas acessos internos.

Figura 11.1: A rede interna de uma organização.

Isso muda com a necessidade de comunicação entre a organização e as suas filiais. Para isso foram utilizadas as conexões dedicadas (figura 11.2), que possuem um custo bastante alto.

Nesse ponto, a preocupação com a segurança ainda é pouca (segurança interna), pois ainda

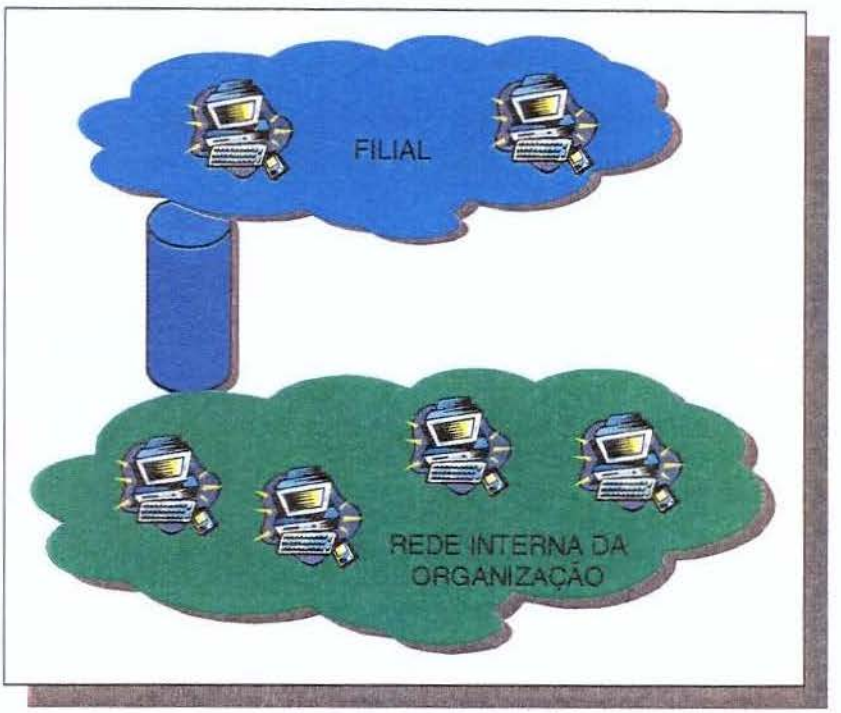


Figura 11.2: A comunicação entre organizações através de conexão dedicada.

não existem acessos externos. No entanto, é preciso considerar que os acessos remotos suportados através de modens podem passar a serem pontos de ataques, como o que resultou no ataque realizado por Kevin Mitnick. Os modens ainda constituem pontos ativos de ataques que podem comprometer a organização, principalmente através da formação de um atalho que pode driblar o *firewall*, se ele for mal implementado. Os modens não serão considerados nesta arquitetura, por justamente aumentarem a complexidade da segurança, ao exigir proteções específicas, tanto para a estrutura de acesso remoto quanto para usuários internos que utilizam modens. Somado a isso, ainda existem os problemas de segurança envolvidos com os novos tipos de acessos remotos, como as conexões via cabo ou xDSL, que devem ser tratados com extremo cuidado, já que também podem ser utilizados como um desvio em torno do *firewall*. Tudo isso, porém, serve para mostrar a importância de uma política de segurança bem definida, que trata de todos os aspectos envolvidos (capítulo 5).

Os problemas de segurança passaram a ser mais preocupantes com o advento da Internet. Logo que o acesso à Internet passa a fazer parte da rede da organização, o inverso também passa a ser verdadeiro, ou seja, qualquer um da Internet passa a poder acessar a rede da organização. O *firewall* (capítulo 6) passa assim a ser um componente essencial para as organizações que contam com o acesso dedicado à Internet (figura 11.3).

Desta forma, o *firewall* passa a isolar a rede da organização contra os acessos externos

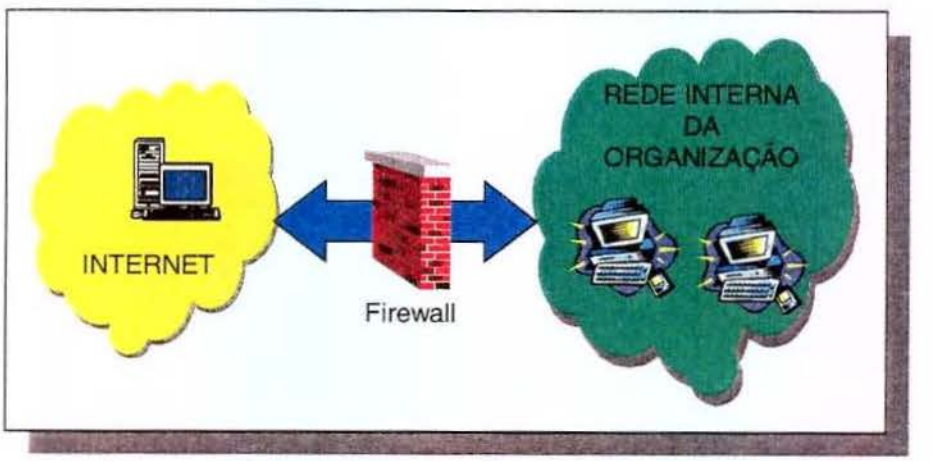


Figura 11.3: A necessidade do *firewall* nas conexões com a Internet.

vindos da Internet, que de fato não são necessários neste cenário. O que são necessários são apenas os acessos dos usuários internos para a Internet, e não o inverso. As regras de filtragem assim são bastante simples, bastando bloquear tudo o que vem da rede pública, permitindo apenas as conexões com origem na rede interna dos serviços permitidos pela política de segurança.

Isso porém passa a mudar quando a organização passa a prover serviços para toda a comunidade da Internet, de modo que assim os usuários externos passam a acessar recursos da organização, principalmente servidores Web, servidores FTP e servidores de *e-mail* (figura 11.4).

A questão de onde localizar os servidores culminou no conceito de DMZ (capítulo 6), onde o sucesso de um ataque contra esses servidores não significa o comprometimento da rede interna da organização. Na figura 11.4 pode-se observar que o comprometimento de um dos serviços providos para a Internet resulta no acesso automático do *hacker* aos recur-

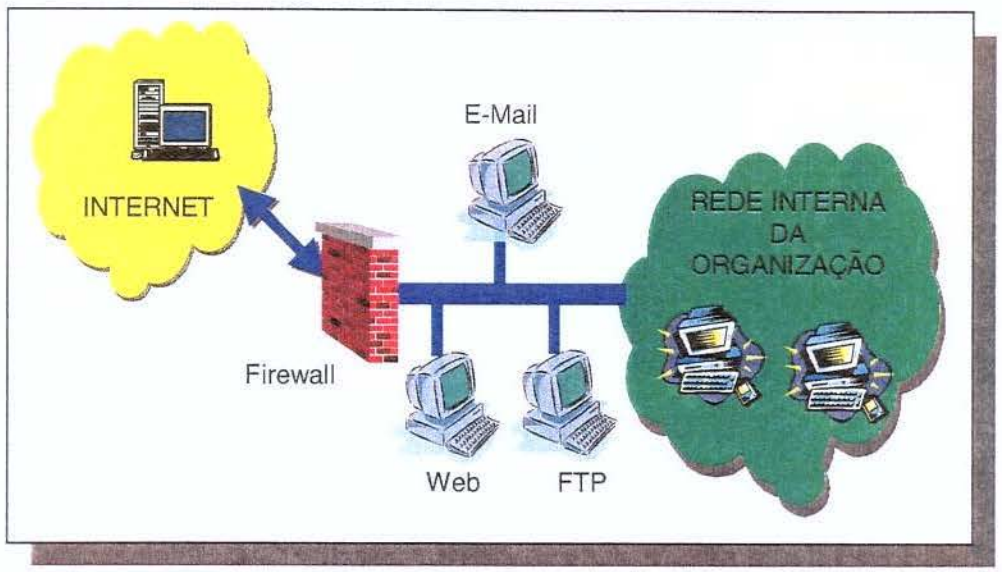


Figura 11.4: A organização provendo serviços para os usuários externos.

sos internos da organização, ou seja, sem a DMZ, o sucesso de um ataque fará com que o *hacker* esteja dentro da rede da organização. A DMZ evita esse tipo de risco ao criar uma sub-rede formada por duas barreiras (figura 11.5). Caso o *hacker* passe pela primeira barreira e ataque um dos serviços providos, ainda existe a segunda barreira a ser vencida pelo *hacker*, para que ele tenha acesso aos recursos internos da organização. Esse *firewall* (figura 11.5) poderia ser composto, como foi visto no capítulo 6, de um filtro de estados na barreira 1, e de *proxies* na barreira 2. Este foi o esquema utilizado na configuração do LAS-IC-Unicamp, que será vista no capítulo 12.

As duas barreiras que formam a DMZ podem ser colocadas nas interfaces de um *firewall*, como pode ser observado na figura 11.6. A questão que se tem aqui é com relação à melhor configuração. O melhor é utilizar o esquema da figura 11.5 ou o esquema da figura 11.6?

Pode-se verificar que nos dois esquemas os serviços são providos através da DMZ. A diferença é que na figura 11.5 o *firewall* é composto por dois componentes (barreira 1 e barreira 2), fora a DMZ, enquanto na figura 11.6 o *firewall* é formado por um único componente, com 3 interfaces de rede.

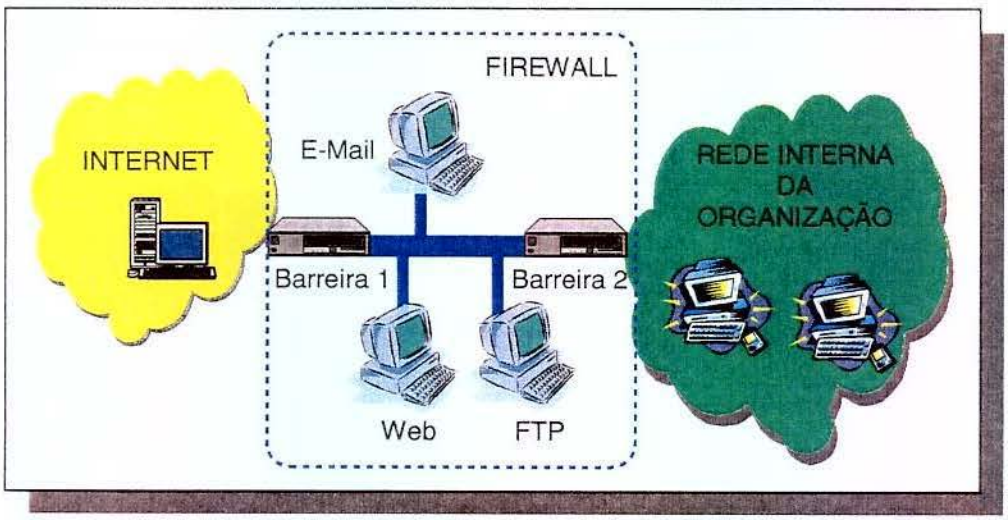


Figura 11.5: As duas barreiras que formam a DMZ do *firewall*.

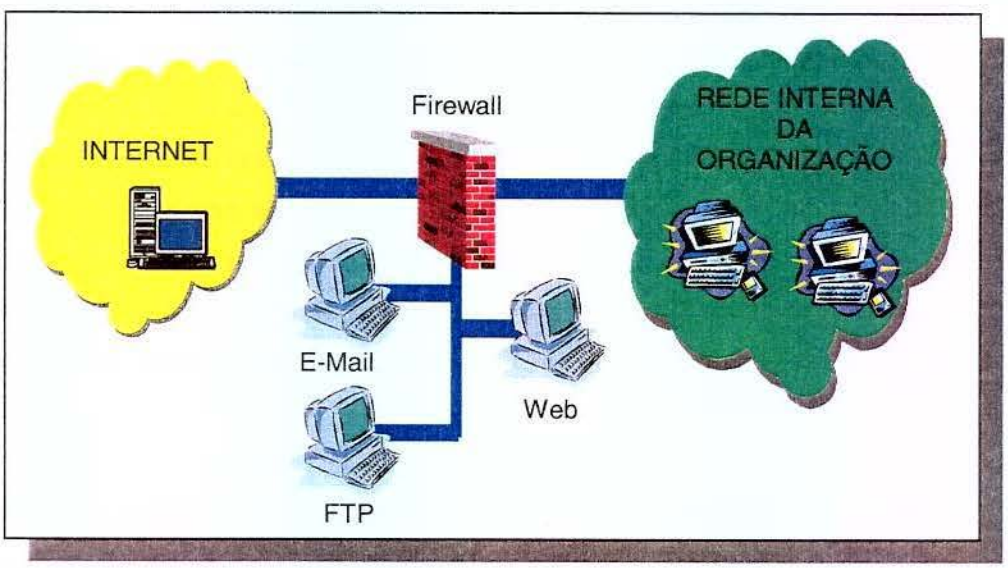


Figura 11.6: O *firewall* composto por 3 interfaces de rede.

Mas será que existem diferenças quanto ao nível de segurança entre os dois esquemas? Foi visto no capítulo 4 que *bugs* podem resultar em acessos não autorizados através da exploração de *buffer overflow*, condições inesperadas, entradas não manipuladas ou de *race conditions*. Foi visto também no capítulo 3 que a complexidade é inversamente proporcional ao nível de segurança dos sistemas. No capítulo 6 foi visto ainda que a complexidade dos *firewall* vêm aumentando através da combinação de diversas funcionalidades em um

único equipamento. Essa observação é coerente, uma vez que a complexidade traz maiores possibilidades de erros em sua implementação, que resultam em *bugs* que podem ser explorados, diminuindo assim o nível de segurança do sistema. E o que vem acrescentando a complexidade dos *firewalls* é a adição de novas funcionalidades.

Assim, o melhor para um *firewall* é que ele seja o mais simples possível, o que de fato é verdade, se forem consideradas as tecnologias básicas que funcionam como barreira na rede da organização (filtros de pacotes, filtros de estados, *proxies*). Os filtros de pacotes e de estados atuam no *kernel* do sistema operacional, sendo extremamente simples, com a mínima possibilidade de *bugs* que possam ser explorados. *Race conditions*, que podem resultar em inconsistências de informações, não aparecem nos filtros, e o *buffer overflow* não pode ser explorado, pois os pacotes IP são regidos pelo MTU, ou seja, pacotes com tamanho grande não podem ser utilizados sem que antes exista a fragmentação desses pacotes em unidades menores. Os *proxies*, que atuam na camada de aplicação, também possuem poucas chances de conterem erros, pois a maioria deles realizam apenas a função de *relay*, no nível de circuitos, entre o cliente e o servidor. Os *proxies* de nível de aplicação podem realizar algumas filtragens no conteúdo dos pacotes, porém, como esses pacotes não ultrapassam o tamanho determinado pelo MTU, não podem sofrer com o *buffer overflow*, além do *race conditions* também não existir.

Desta maneira, pode-se afirmar que o esquema 2 (figura 11.6) é tão seguro quanto ao esquema 1 (figura 11.5), possuindo a vantagem de facilitar a administração, devido ao menor número de equipamentos a serem gerenciados. O que deve ser lembrado é que nenhum outro serviço deve estar sendo executado no equipamento. O desempenho pode sofrer algumas alterações, sobretudo em um ambiente complexo como o ambiente cooperativo, porém a importância deve ser dada em torno da segurança, sendo o desempenho um fator secundário. Se necessário, mais equipamentos podem ser utilizados para que a carga seja distribuída entre eles.

Implementando-se uma das configurações acima, a organização está apta a acessar serviços da Internet e também de prover serviços para usuários externos. Seguindo os passos da evolução, a organização então passa a ter a necessidade de prover informações mais específicas a seus usuários, como informações sobre compras on-line. Esse tipo de informação, que é normalmente específica e confidencial, e portanto deve ser protegida contra acessos indevidos, fez surgir a necessidade de maiores cuidados com relação à localização do banco de dados. A sua localização na DMZ, como um *bastion host*, poderia ser uma opção (figura 11.7). Um ponto importante

é a escolha do método de autenticação utilizado para que o acesso seja provido (capítulo 10). Porém, sabe-se que os recursos residentes na DMZ possuem acesso externo permanente, sendo portanto alvos de tentativas de ataques.

Essa configuração coloca em risco as informações do banco de dados, resultando então

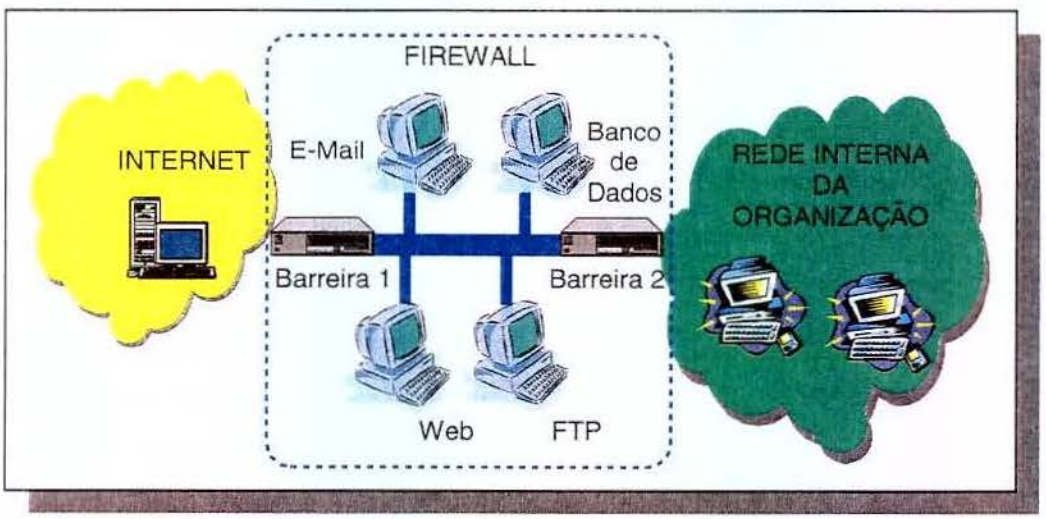


Figura 11.7: O servidor de banco de dados na DMZ.

na idéia de colocar o banco de dados não na DMZ, mas sim na rede interna da organização (figura 11.8).

Essa configuração, porém, dá a impressão de que um ataque ao servidor de banco de dados resulta no acesso à rede interna da organização, o que de fato é verdade, mas que pode ter o risco minimizado com o correto desenvolvimento e correta implementação da política de segurança no *firewall*. A idéia aqui é fazer com que a segunda barreira permita passar apenas o tráfego referente à conexão entre o servidor Web e o servidor de banco de dados, não sendo possível o acesso direto ao banco de dados. Assim, para que o *hacker* tenha acessos não-autorizados à base de dados, seria necessário primeiro comprometer o servidor Web, e depois o servidor de banco de dados. É importante lembrar ainda que a autenticação deve fazer parte desse esquema de acesso aos dados (capítulo 10).

Porém, existe ainda uma configuração que traz maior nível de segurança à organização, que é a utilização de uma segunda rede DMZ (figura 11.9):

Esse esquema possui o mesmo grau de segurança do esquema da figura 11.8, com relação à base de dados da organização. A vantagem é que esse novo esquema evita o problema

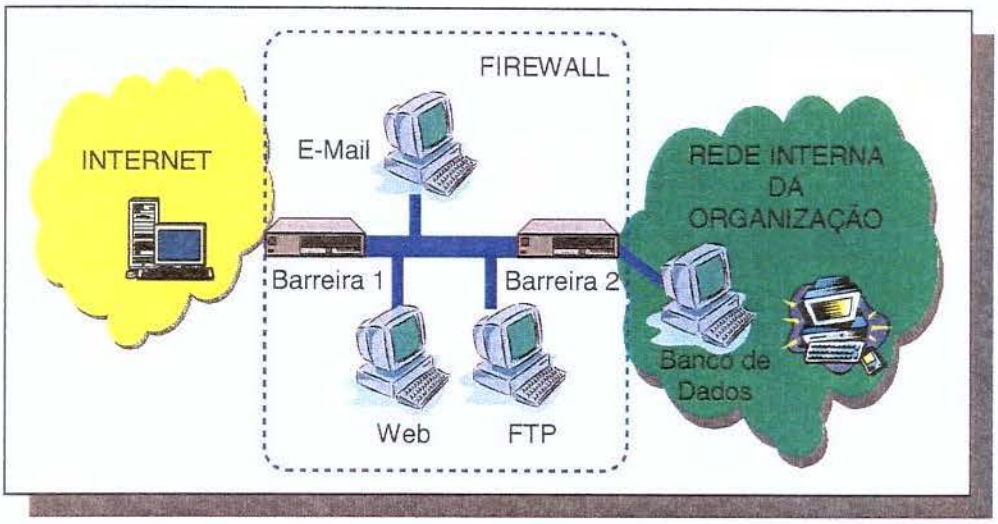


Figura 11.8: O servidor de banco de dados na rede interna da organização.

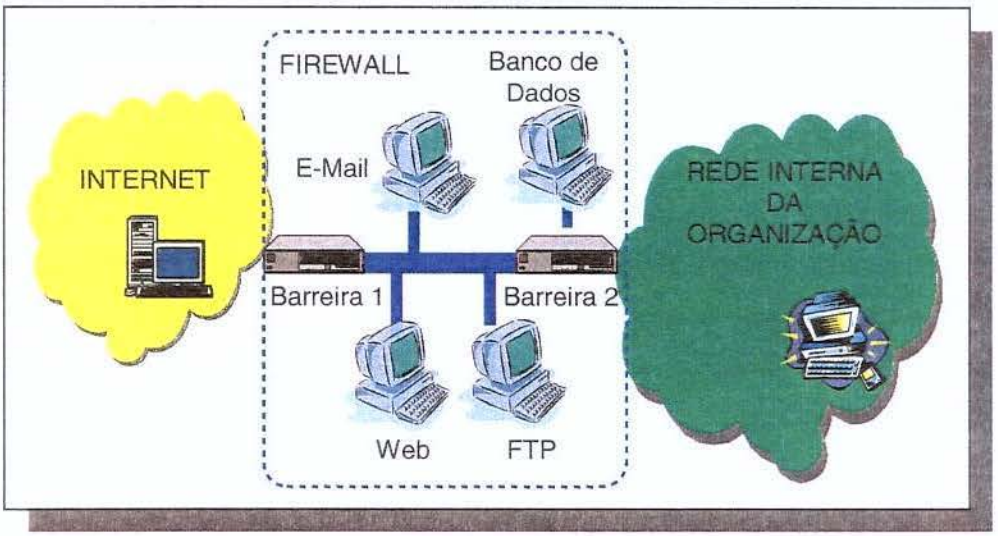


Figura 11.9: O utilização de uma segunda DMZ para o servidor de banco de dados.

do comprometimento da rede interna da organização caso um ataque ao servidor de dados tenha sucesso. A mesma arquitetura, utilizando-se um único componente de *firewall*, com quatro *interfaces* de rede, pode ser vista na figura 11.10.

Neste ponto, a organização possui o acesso à Internet, provê serviços para os usuários,

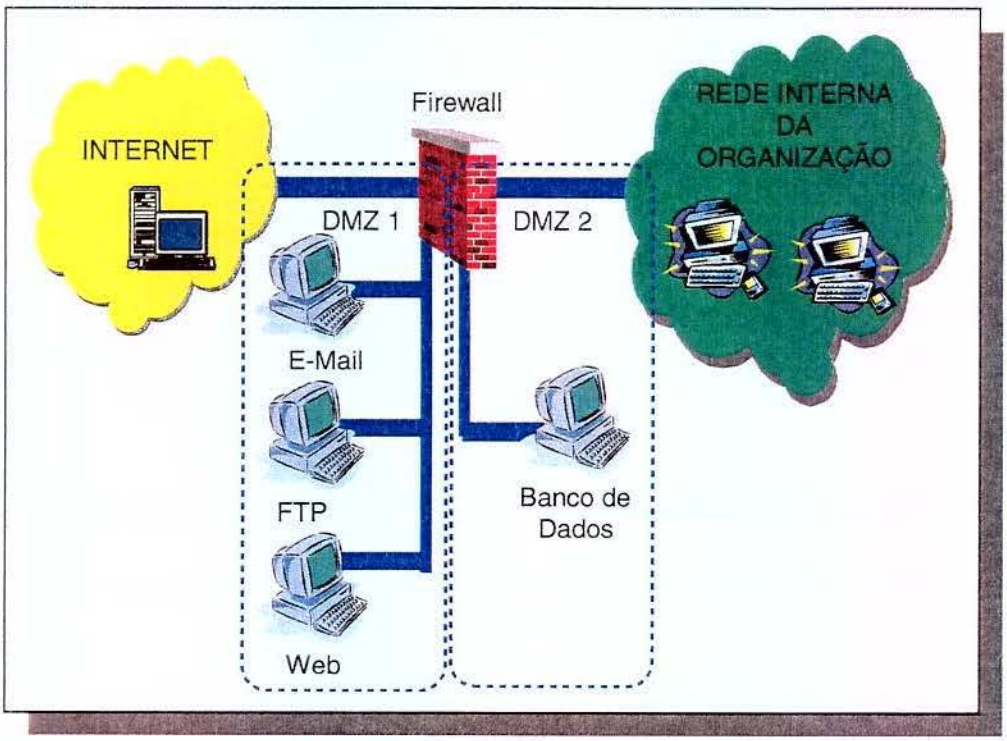


Figura 11.10: Duas DMZs em um único componente de *firewall*.

sendo que um deles é o acesso à informações consideradas confidenciais. Pode-se considerar também que a organização possui ainda uma linha dedicada com a sua filial, e deseja reduzir os custos referentes a essa linha, através da utilização da VPN (capítulo 9).

De acordo com o esquema visto até o momento, a arquitetura da organização seria a que pode ser vista na figura 11.11.

Aqui, o que entra em discussão são as conexões existentes na filial. No esquema da figura 11.11 pode-se ver que a filial não possui nenhum outro tipo de conexão, de modo que tudo está de acordo, ou seja, ataques vindos do exterior são improváveis, podendo-se considerar a segurança como estando no n[ivel interno da organização. Porém, a existência de outras conexões na filial pode colocar em risco a rede interna da organização, como pode ser visto na figura 11.12.

No esquema da figura 11.12, a rede interna da organização corre o risco de acessos não-autorizados dos usuários da Rede A, que podem chegar à rede interna através da rede da filial. Essa situação pode se tornar ainda mais crítica se a Rede A possui acesso à Internet sem a proteção necessária (figura 11.13):

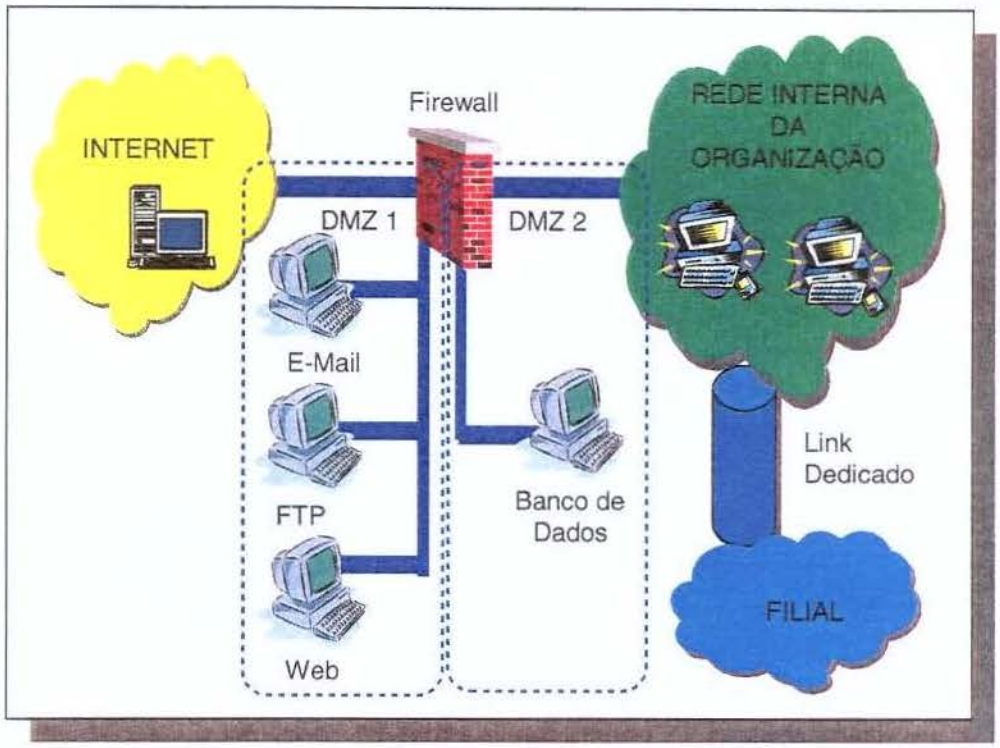


Figura 11.11: A arquitetura da organização com os acessos à Internet e à filial.

Nesse esquema (figura 11.13), qualquer usuário da Internet pode chegar à rede interna da organização passando antes pela rede A, depois pela filial, até chegar à rede interna. Pode-se observar que o *firewall*, implementado para proteger a rede interna contra acessos não autorizados, passa a não ter função alguma, ao ser driblado através da passagem pela rede A e pela rede da filial.

Na realidade, esses dois passos (rede A e rede da filial) não é ao menos necessário, caso a própria filial possua o acesso à Internet. Como pode ser visto na figura 11.14, essa é uma configuração reconhecidamente perigosa, já que a filial não possui os mesmos mecanismos de segurança da rede interna da organização, ou seja, a filial não possui o *firewall*.

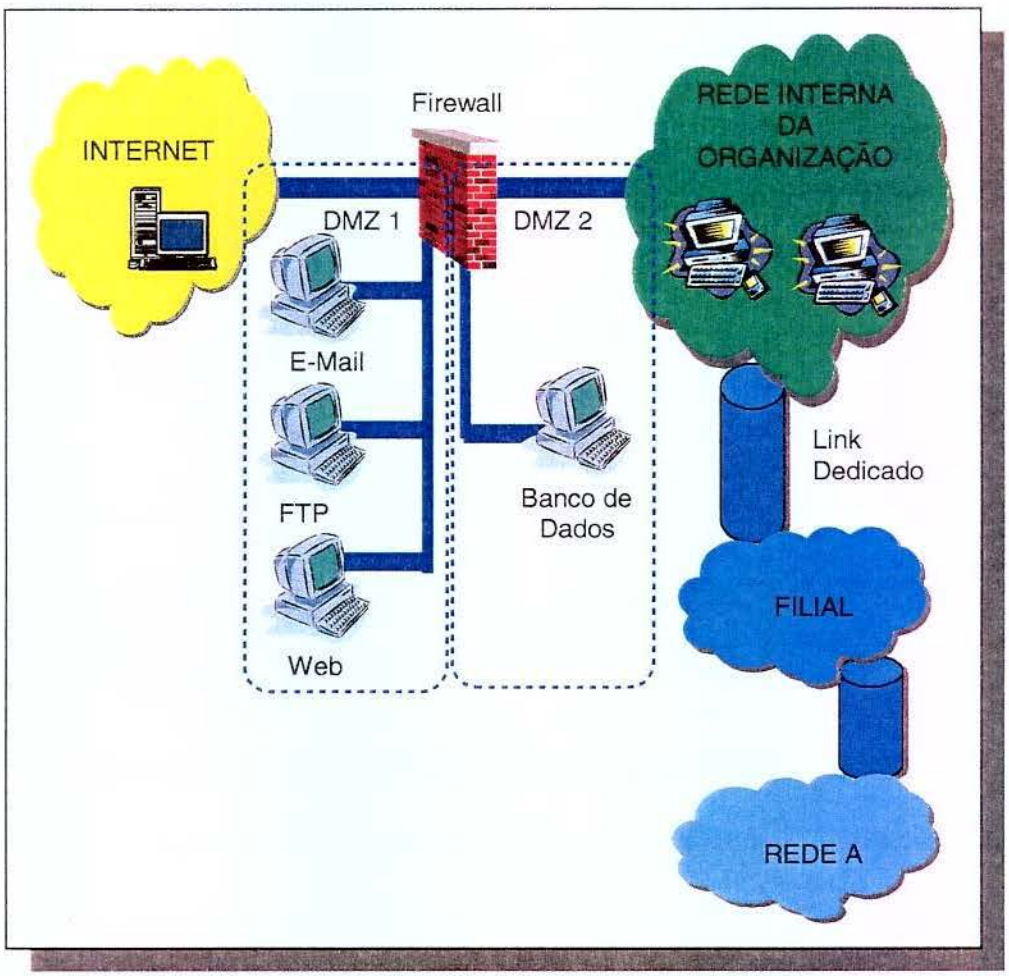


Figura 11.12: Os riscos envolvidos em múltiplas conexões.

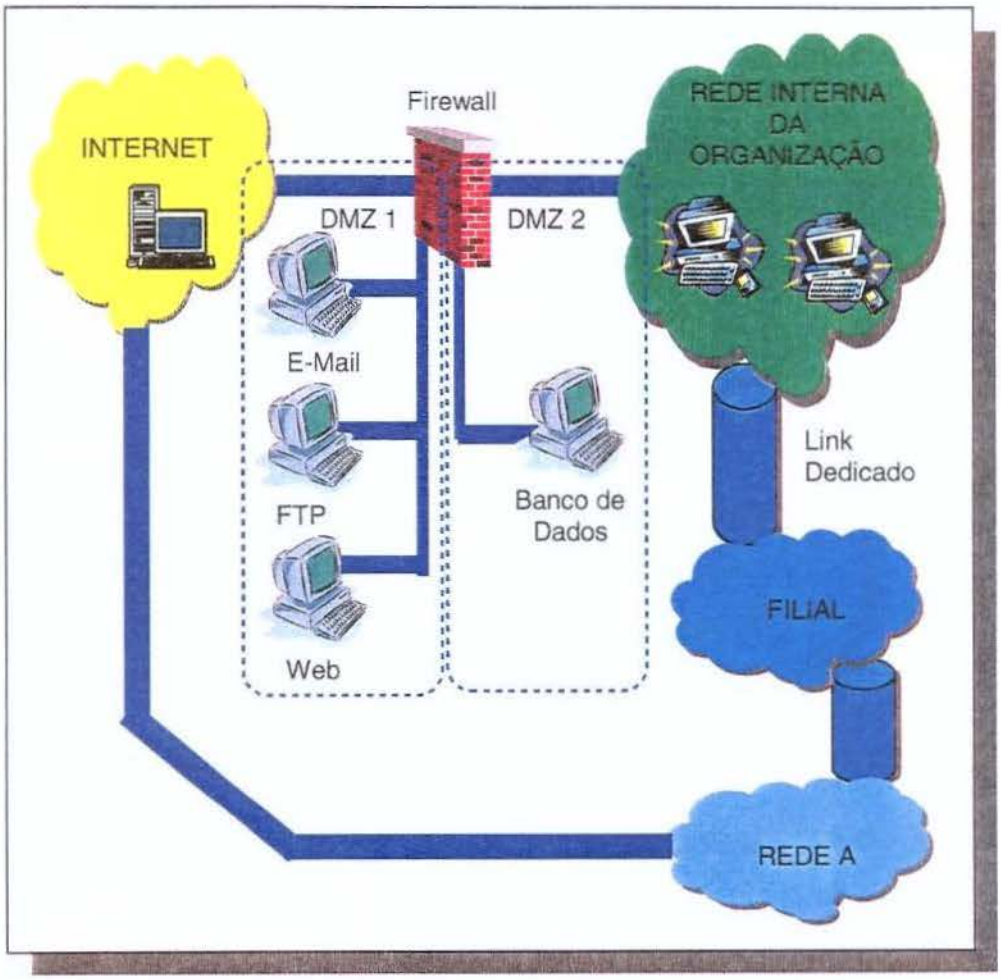


Figura 11.13: Múltiplas conexões envolvendo a Internet.

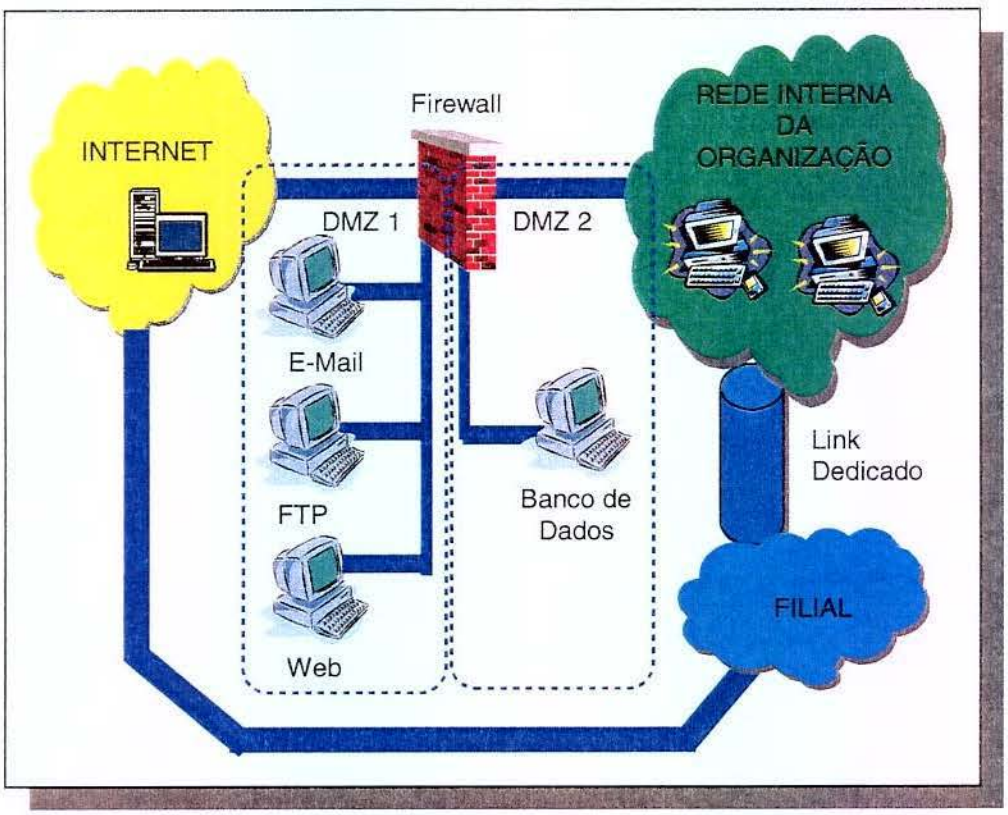


Figura 11.14: Mecanismos de segurança não equivalentes entre matriz e filial.

Deve-se considerar, no entanto, que uma avaliação dos perigos existentes na filial, resultantes da conexão à Internet, deve ser realizada. Neste ponto, já começa-se a enxergar o início de um ambiente cooperativo, e verifica-se que o que foi discutido no capítulo 5, a política de segurança em ambientes cooperativos, passa a ser aplicável.

Se for levada em consideração que cada organização deve cuidar da sua própria segurança, o que de fato foi a conclusão obtida no capítulo 5, então a abordagem a ser seguida é a de implementar um *firewall* entre a filial e a rede interna da organização. Assim, mesmo que a filial sofra um ataque, os riscos quanto à rede interna podem ser minimizados. De fato, essa é uma abordagem que deve mesmo ser seguida (*firewall* interno), porém, em se tratando de uma mesma organização, geralmente uma outra abordagem é seguida. A mais utilizada é a duplicação da configuração da borda de rede da matriz na borda de rede da filial.

Porém, como a duplicação de esforços para que a rede da filial possua o mesmo nível de segurança da rede interna significa altos custos de implementação e gerenciamento, ela

não é justificada para casos de acessos aos serviços básicos da Internet, como são a Web, FTP e *e-mail*. Assim, a configuração mais utilizada a princípio é a que se segue (figura 11.15), onde o acesso à Internet é realizado através da linha dedicada até a rede interna da organização, onde a partir daí o acesso à Internet é permitido, passando-se pelo *firewall*. Esse esquema não resulta em nenhuma implicação de segurança, pois a filial não possui outros tipos de conexões, sendo que todas as comunicações são realizadas através da rede interna da organização, que está protegida dos acessos externos indevidos pelo *firewall*.

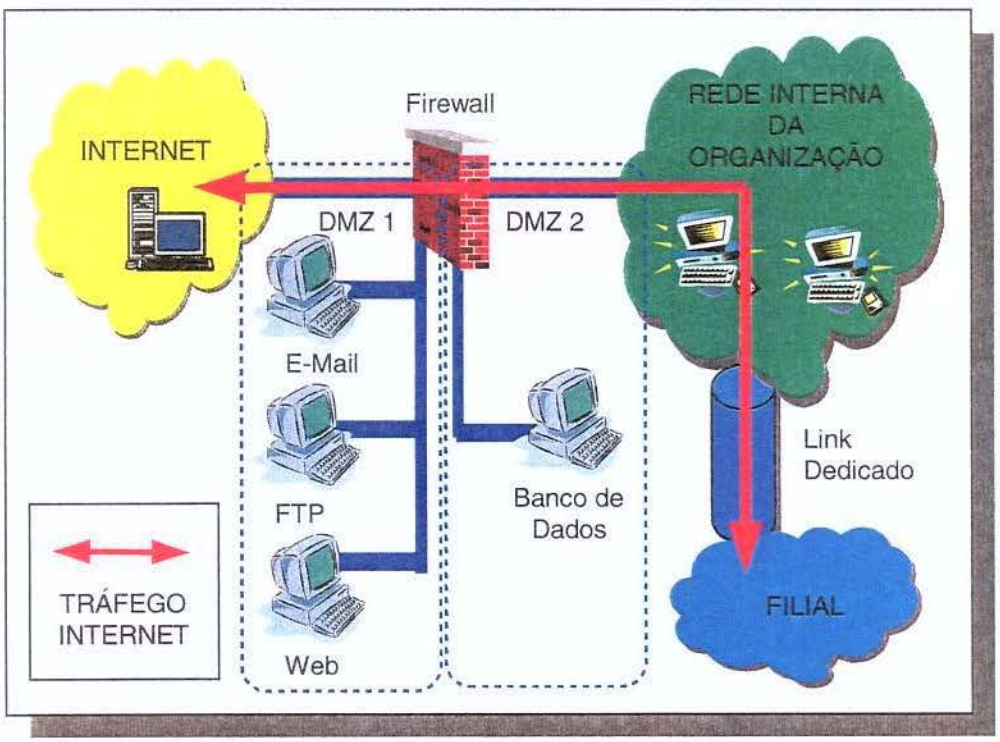


Figura 11.15: Acesso à Internet da filial através de linha dedicada.

Porém, a utilização de uma linha dedicada para o tráfego Internet resulta em custos bastante altos, sobretudo em uma organização onde a filial possui um grande número de usuários. Assim, a organização deve considerar a idéia de utilizar uma conexão direta com a Internet na filial, para que a linha dedicada possa ser economizada. Uma das idéias é a utilização da VPN para realizar essa função (figura 11.16). Porém, pode-se verificar que essa alternativa é totalmente desneces-

sária para o caso do acesso apenas aos serviços da Internet. O primeiro passo para a utilização da VPN é a necessidade de uma conexão com a Internet, por onde o túnel virtual será criado. Deste modo, o túnel é criado no *gateway* da rede da filial e finalizado no *gateway* da rede da matriz.

Esse tipo de conexão, que é feito entre duas redes organizacionais, é conhecida como *gateway-to-gateway VPN*. Não será abordada neste cenário os outros tipos de conexões VPN, que podem ser feitos entre o cliente e a rede da organização (*client-to-gateway VPN*) e entre um cliente e outro (*client-to-client* VPN).

Nesse esquema (figura 11.16) surgem duas questões essenciais para a segurança da orga-

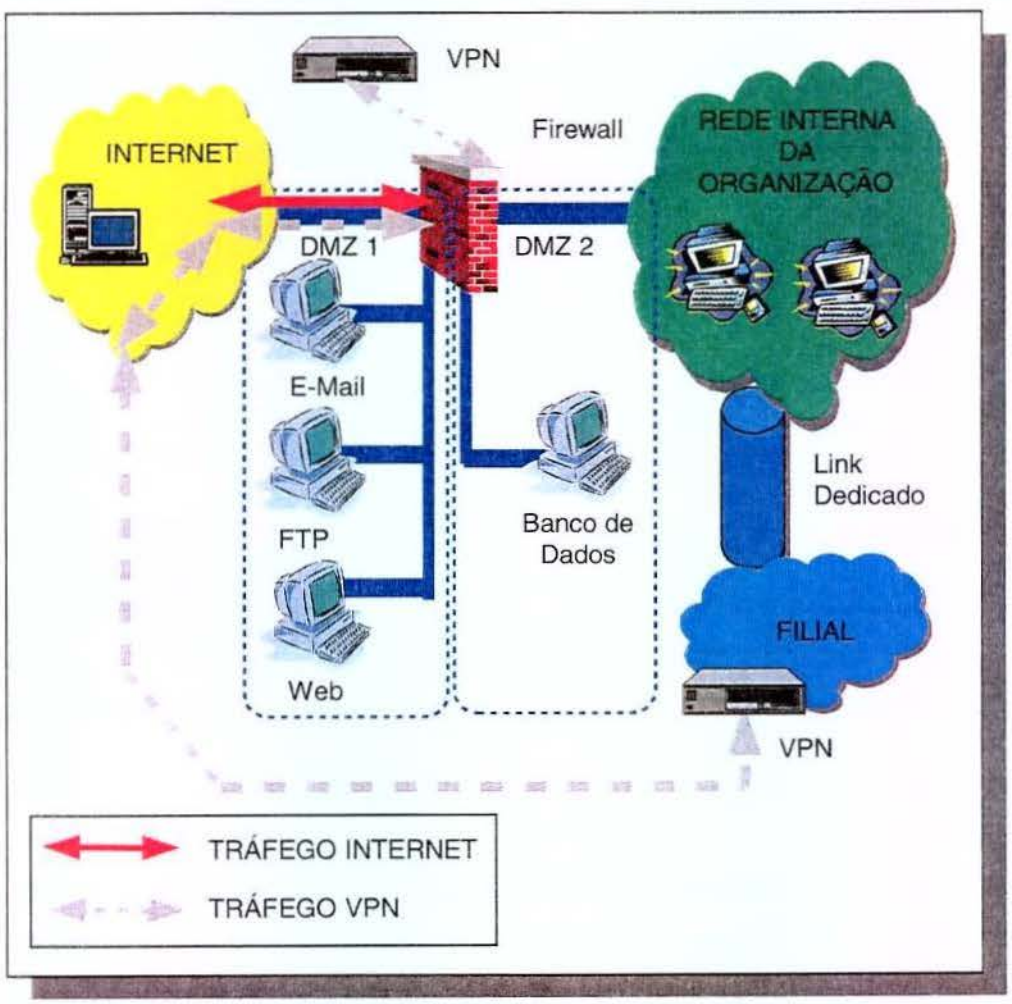


Figura 11.16: Acesso à Internet da filial através de VPN.

nização:

1. Do que consiste a VPN que funciona no *gateway* da filial?

2. Como deve ser a configuração da VPN no *firewall*?

Pode-se observar que o tráfego Internet nessa conexão é realizado de modo a desperdiçar recursos, já que a requisição do usuário da filial passa pelo túnel virtual, vai até a Internet, chega até o *firewall* da matriz, e vai até o dispositivo VPN, onde o túnel é desfeito, e a requisição chega novamente à Internet, dessa vez até o seu destino. O caminho inverso, da resposta, é feito da mesma maneira, ou seja, a resposta retorna ao *firewall*, onde então o encapsulamento da resposta é feito pela VPN, e a resposta é enviada via túnel para a Internet, até que chegue à origem, ou seja, o usuário da filial. Pode-se observar que o mesmo pacote de requisição e resposta passa 3 vezes pela Internet, sendo que em circunstâncias normais existe uma única passagem pela Internet, bidirecional, que é a requisição e a resposta direta ao cliente.

A utilização da VPN somente para o tráfego Internet passa assim a ser injustificável, a não ser que a confidencialidade dos dados seja um requerimento essencial, o que pode ocorrer no caso de transferência de *e-mails* entre a matriz e a filial através da Internet, em oposição à utilização da linha dedicada.

Porém, independente da justificativa com relação à utilização, o ponto primordial a ser considerado nesse esquema surge a partir da questão 1 referida a pouco: do que consiste a VPN que funciona no *gateway* da filial? Essa questão surge porque o ponto fundamental a ser tratado quando uma organização passa a ter uma conexão direta com a Internet é: se possui acesso à Internet, então o controle de borda, realizado pelo *firewall*, deve existir, para que os acessos indevidos sejam evitados. Levando-se isso em consideração, será que no esquema visto anteriormente (figura 11.16), as funcionalidades VPN são acompanhadas pela proteção de borda, ou seja, será que o dispositivo VPN está fazendo também o papel de *firewall*? Não é o que está representado na figura 11.16, e essa questão é relevante, uma vez que, como foi visto no capítulo 6, o *firewall* muitas vezes é visto erroneamente como sendo a solução de todos os problemas de segurança. E o esquema visto parece estar se aproveitando dessa afirmação, ao fazer com que o tráfego passe obrigatoriamente pelo *firewall* da matriz, como se assim o nível de segurança fosse assegurado. Mas, como quando a conexão com a Internet existe, o *firewall* também tem que existir, utilizar a VPN para que o *firewall* da matriz seja utilizado não faz sentido, pois o *firewall* tem

que existir na rede da filial de qualquer modo, devido à necessidade de proteção contra os ataques vindos a partir dessa conexão com a Internet.

Esse *firewall* na filial pode ser implementado de maneira extremamente simples, pois nenhum serviço será provido a partir da filial. O *firewall* apresentado na figura 11.3 pode resolver o problema do acesso à Internet da filial, sem comprometer a segurança da matriz. O *firewall* da filial deve permitir apenas que somente os pacotes dos serviços básicos permitidos para os usuários internos passem pelo filtro.

A VPN pode ser utilizada para o tráfego de *e-mails* entre a matriz e a filial, por exemplo, além de ser possível também utilizá-la como canal de troca de documentos com informações confidenciais. A figura 11.17 mostra a configuração ideal para essa situação.

O que foi abordado responde à questão 1, faltando ainda abordar a questão 2 surgida com

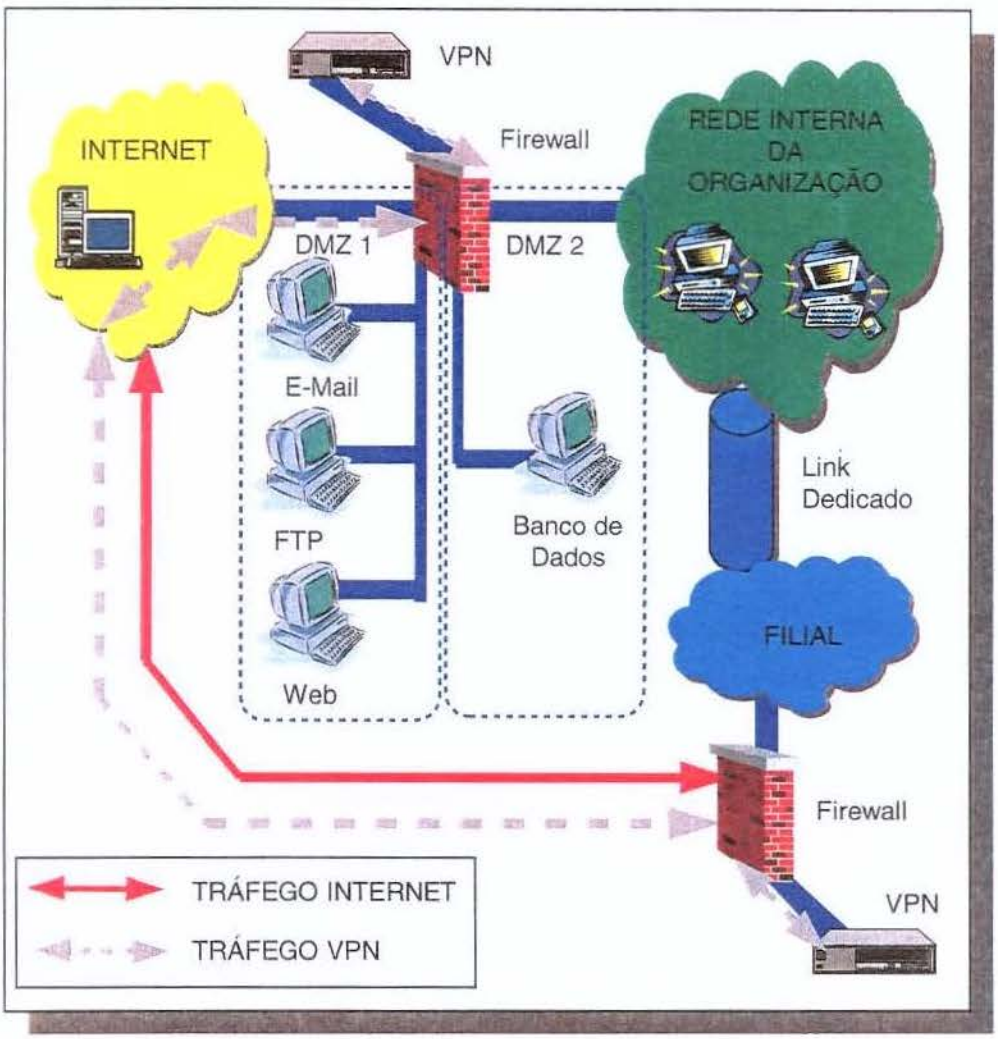


Figura 11.17: Acesso à Internet em conjunto com VPN.

a utilização da VPN: como deve ser a configuração da VPN no *firewall*? As possibilidades de configurações do VPN com relação ao *firewall* serão exploradas na seção 11.2.

Pode ser visto claramente que a complexidade da arquitetura de segurança vem aumentando (figura 11.18), de acordo com as novas necessidades de conexões. Essa complexidade passa a ser maior e mais séria quando acessos à rede interna devem ser providos. Um caso típico é quando um fornecedor deseja acessar informações internas da organização, referente a estoques.

Esse tipo de acesso pode ser realizado através de uma aplicação específica, e o requerimento básico é que as informações não possam trafegar em claro pela rede, além de ser necessário também garantir a integridade desses dados para que eles não sejam alterados no meio do caminho entre o servidor e o cliente (ataque *man-in-the-middle*). Outro requerimento básico é garantir que apenas os usuários legítimos tenham o acesso no servidor. O primeiro requerimento pode ser obtido através da utilização da VPN, e o segundo através de um esquema forte de autenticação e controle de acesso (capítulo 10).

Pode-se observar a partir da figura 11.18 que a complexidade é bastante alta, que aumenta ainda mais em um ambiente cooperativo, onde um número maior de níveis de conexões diferentes é necessário (*telecommuters*, revendas, clientes, parceiros comerciais, etc). Quando existem, por exemplo, 10 níveis de conexões diferentes, a política de segurança torna-se difícil de ser desenvolvida, e principalmente, de ser implementada. Uma discussão sobre a complexidade envolvida nos ambientes cooperativos será feita no capítulo 12, que inclui também dificuldades nas regras de filtragem, e uma maneira de simplificar essas regras.

Ao mesmo tempo em que o número de diferentes conexões vai aumentando, a autenticação dos usuários torna-se mais complicada, requerendo também um método mais forte de autenticação. Como foi visto no capítulo 10, a autenticação baseada em certificados digitais pode ser considerada uma solução ideal para o ambiente cooperativo. A autoridade certificadora (CA) da infra-estrutura de chaves públicas (seção 8.6) pode atuar em conjunto com a VPN. A VPN, e apenas ela, deve se comunicar com a CA para que as autenticações sejam validadas. Os certificados digitais, como vistos na seção 8.5, dão a garantia de confidencialidade, integridade e não-repúdio, necessárias em conexões críticas.

A posição da CA dentro da arquitetura de segurança é um ponto a ser discutido. A localização da CA na DMZ, como pode ser vista na figura 11.19, pode ser válido, porém essa localização faz com que ela esteja diretamente exposta à acessos externos, o que inviabiliza essa posição. O sucesso de um ataque à CA pode resultar no comprometimento dos certificados digitais dos usuários, culminando na falha total da estratégia de segurança da organização.

A sua localização na segunda DMZ (DMZ 2), do mesmo modo que o banco de dados foi localizado (somente o servidor Web se comunica com o banco de dados), pode ser utilizado (somente a VPN pode se comunicar com a CA). Essa arquitetura pode ser vista na figura 11.20, e ela elimina a possibilidade da CA sofrer acessos externos diretos. O usuário

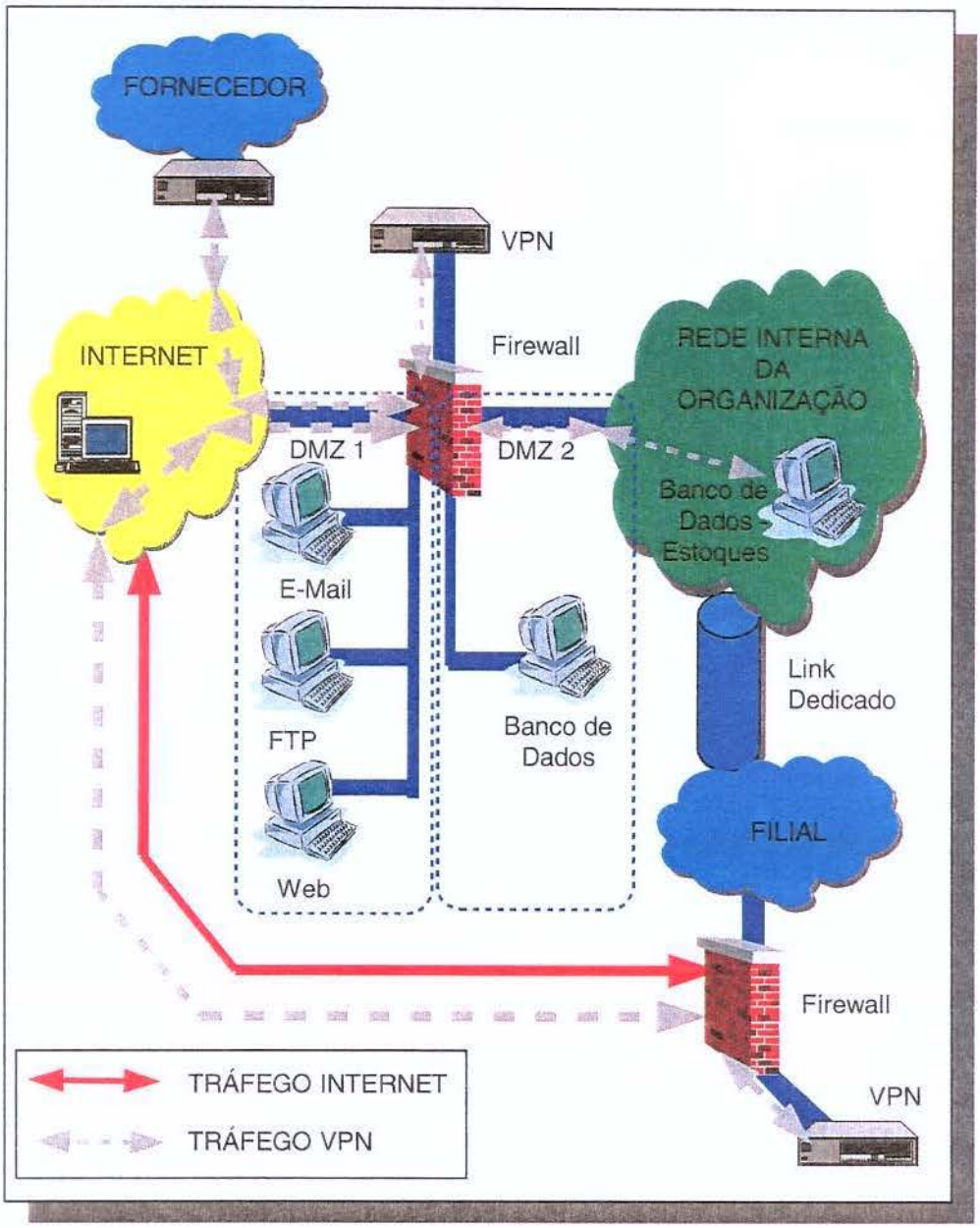


Figura 11.18: Aumento da complexidade das conexões.

somente teria a permissão de acessar os recursos da rede interna da organização após a CA confirmar a sua identidade.

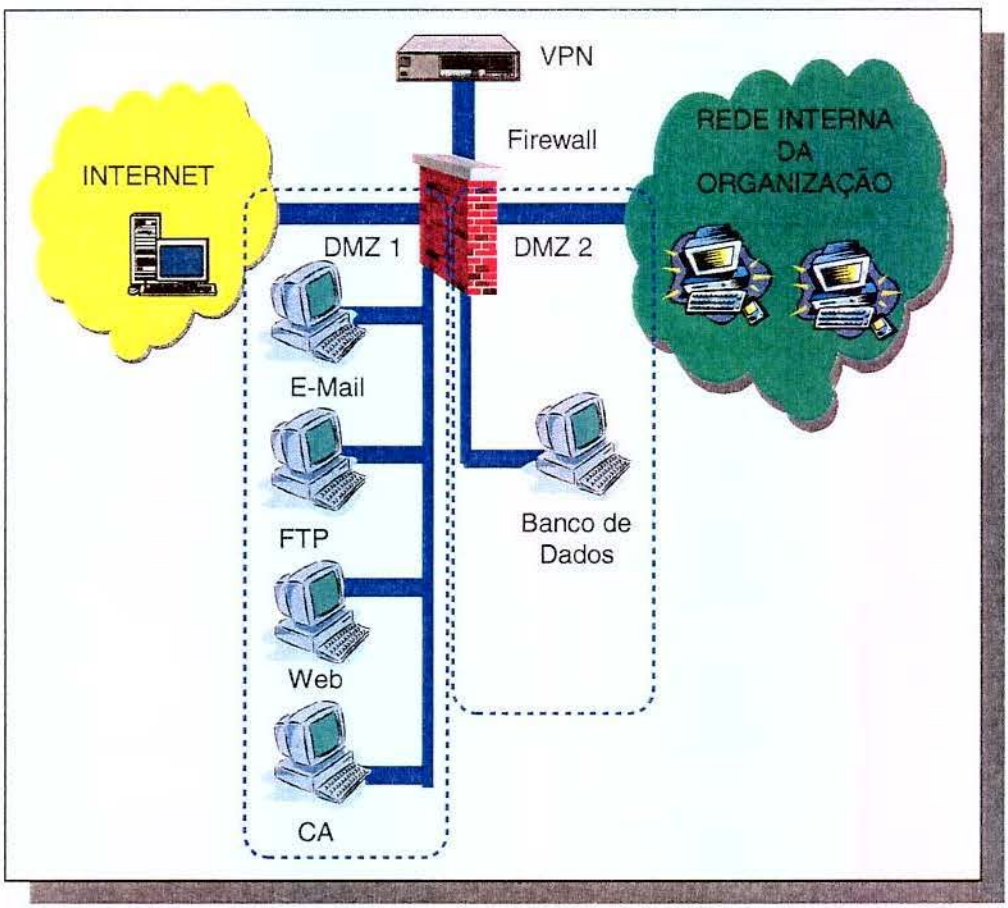


Figura 11.19: Localização do CA na DMZ.

Apesar de ainda estar em processo de padronização, como foi vista na seção 8.6, as certificações cruzadas fazem parte de uma funcionalidade fundamental em um ambiente cooperativo, ao permitir que, por exemplo, os usuários de uma fornecedora acessem recursos da matriz e das filiais, sem a necessidade de que certificados específicos para cada um deles sejam criados em cada uma das partes envolvidas nessa comunicação. Essa certificação cruzada faz com que uma infra-estrutura de chaves públicas (PKI) seja um componente importante na estratégia de segurança da organização, como foi visto na seção 8.6.

Um outro componente de segurança importante, principalmente no nível interno das organizações, são os sistemas de detecção de intrusões (IDS), que foram vistos no capítulo 7. Esses sistemas monitoram todas as atividades dos usuários dentro da rede da organização, sendo assim possível detectar anormalidades que possam ser prenúncios de ataques. A arquitetura de segurança com o IDS pode ser implementada de acordo com o esquema que pode ser visto na figura 11.21:

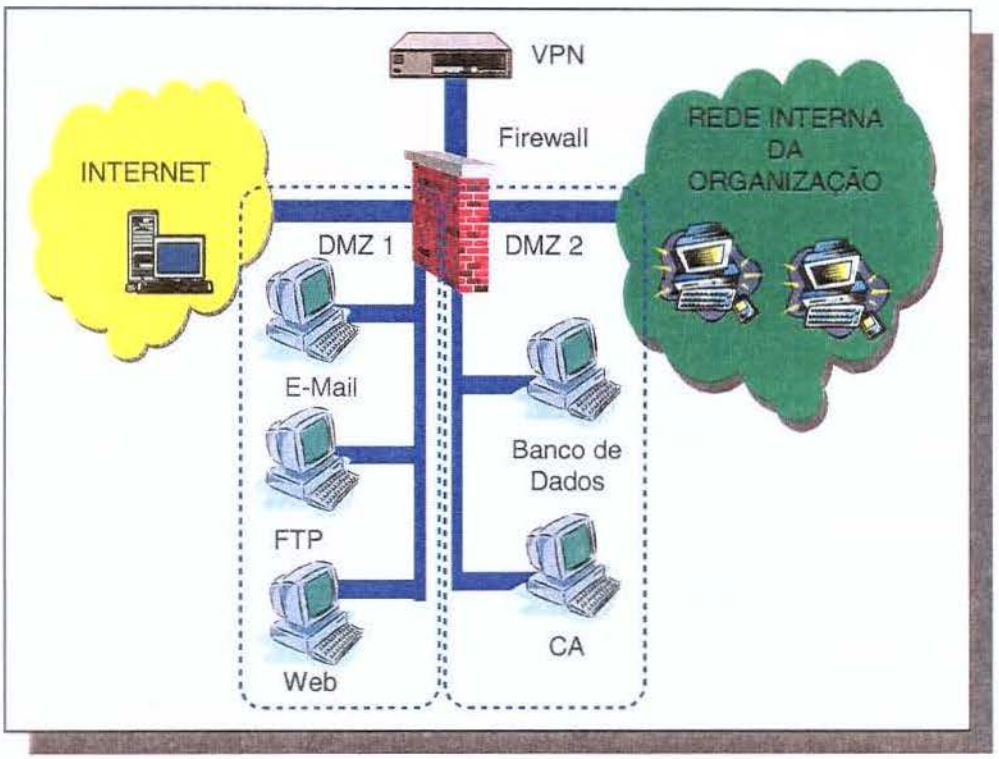


Figura 11.20: Localização do CA na segunda DMZ.

- IDS 1 – É pouco utilizada, pois o *firewall* produz poucas informações que podem ser analisadas;
- IDS 2 – Detecta ataques que passaram com sucesso pelo *firewall*;
- IDS 3 – Detecta ataques contra o *firewall*;
- IDS 4 – Detecta ataques internos na organização.

11.2 Configuração VPN/Firewall

A localização da VPN na arquitetura de segurança é um ponto que deixa margem a diversas interpretações, e nesta seção as diferentes possibilidades dessas localizações serão analisadas e discutidas.

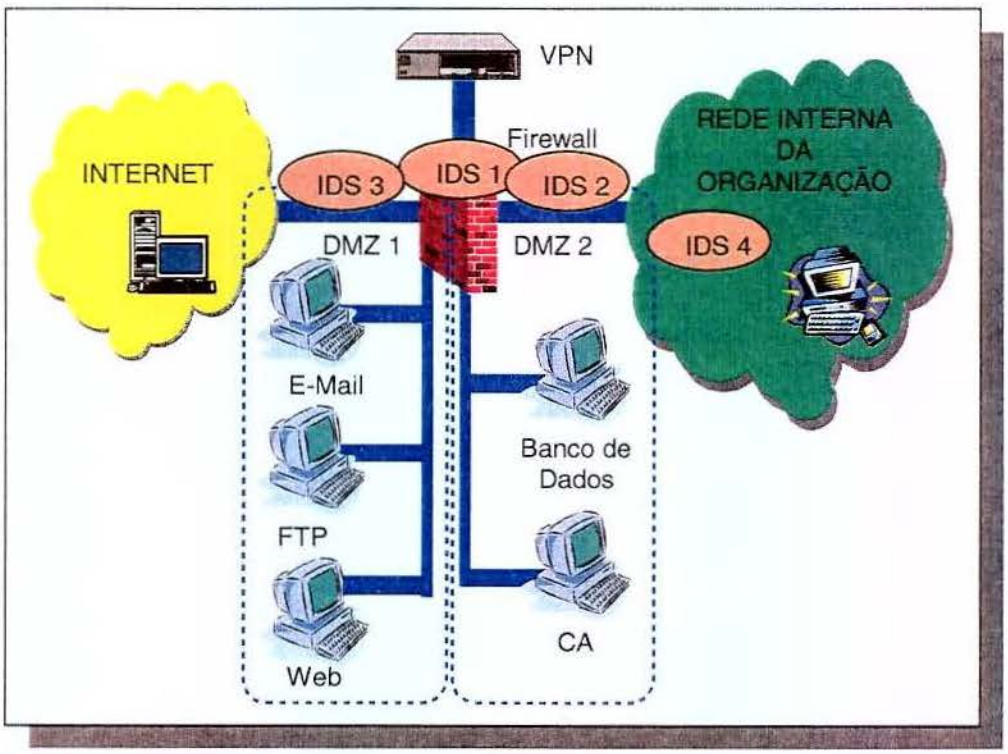


Figura 11.21: A arquitetura de segurança com o IDS.

As diferentes configurações já foram analisadas por King em [KIN 99]. King analisa as 5 possíveis localizações da VPN com relação ao *firewall*:

- Em frente ao *firewall*;
- Atrás do *firewall*;
- No *firewall*;
- Paralelo ao *firewall*;
- Na interface dedicada do *firewall*.

11.2.1 Em Frente ao Firewall

A configuração onde a VPN é colocada em frente ao *firewall*, como pode ser vista na figura 11.22, pode funcionar corretamente, porém apresenta um único ponto de falha que pode ser explorado pelos *hackers*. Isso deve ser considerado seriamente, pois a implementação da VPN pode conter erros que podem ser explorados, especialmente pelos ataques do tipo DoS. Foi visto que um dos pontos a ser considerado em segurança é que o que não é conhe-

cido deve ser considerado de risco. Além do mais, através da utilização dessa configuração não é possível verificar se um *gateway* VPN foi ou não comprometido, caracterizando assim um risco para a organização.

Além disso, essa configuração requer que a VPN seja capaz de aceitar todo tipo de tráfego,

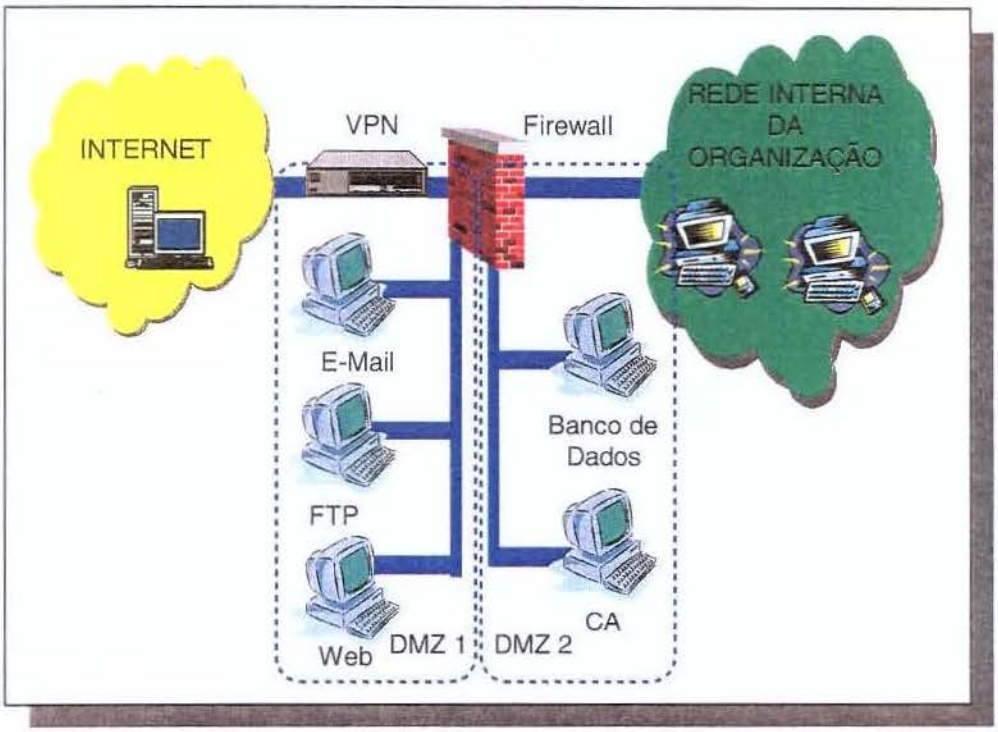


Figura 11.22: A VPN na frente do *firewall*.

seja ele cifrado ou não, já que todos os pacotes devem passar por esse ponto.

11.2.2 Atrás do Firewall

A configuração onde a VPN é localizada atrás do *firewall*, como pode ser vista na figura 11.23, apresenta os mesmos problemas identificados quando ela é colocada na frente do *firewall*, como foi visto na seção anterior.

O maior agravante que pode ser encontrado nessa configuração é que o *firewall* deve deixar passar todo tipo de tráfego cifrado para a VPN, de forma que a política de segurança implemen-

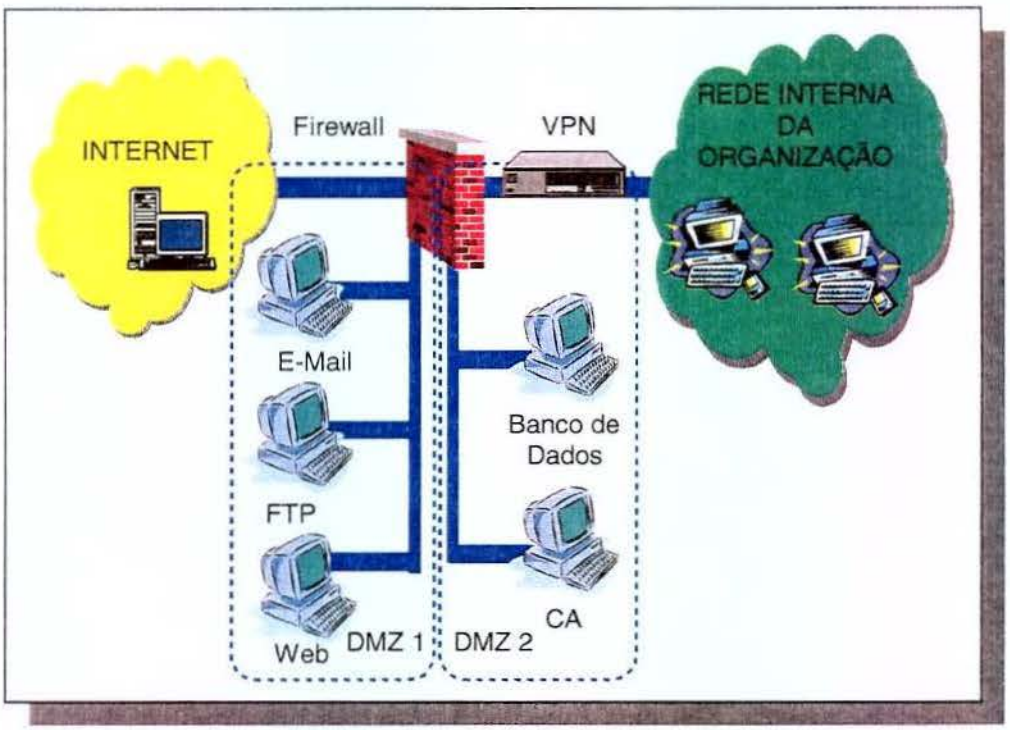


Figura 11.23: A VPN atrás do *firewall*.

tada no *firewall* não é de fato executada nesses pacotes cifrados. Assim, um *hacker* poderia enviar pacotes maliciosos cifrados, que passariam sem problemas pelas regras de filtragem do *firewall*, chegando ao dispositivo VPN. A partir desse dispositivo, os pacotes seriam decifrados e enviados diretamente ao seu destino, que é geralmente um *host* na rede interna da organização. Para que essa configuração funcione, o *firewall* deve ser configurado de modo a deixar passar os pacotes IPs dos tipos 50 e 51 (AH e ESP), além de deixar aberta a porta 500 para o IKE (*Internet Key Exchange*).

11.2.3 No Firewall

A localização da VPN no *firewall* (figura 11.24) faz com que a administração e o gerenciamento sejam simplificados, porém ainda traz o risco de se tornar um único ponto de falha na rede. Além disso, essa configuração não é a mais eficiente em um ambiente cooperativo, por exigir que todo o processo de cifragem/decifragem das informações, além do gerenciamento de todas as sessões IPSec, sejam realizadas nesse único ponto. Essa ineficiência se torna presente porque esse mesmo ponto deve ainda executar a função do *firewall*, que é de controlar o acesso e registrar todos as tentativas de conexões.

Um outro problema é que a existência de falhas na implementação da VPN pode resultar em ataques que podem dar o controle do equipamento ao *hacker*, que poderia assim alterar as regras do *firewall* sem maiores dificuldades.

11.2.4 Paralelo ao Firewall

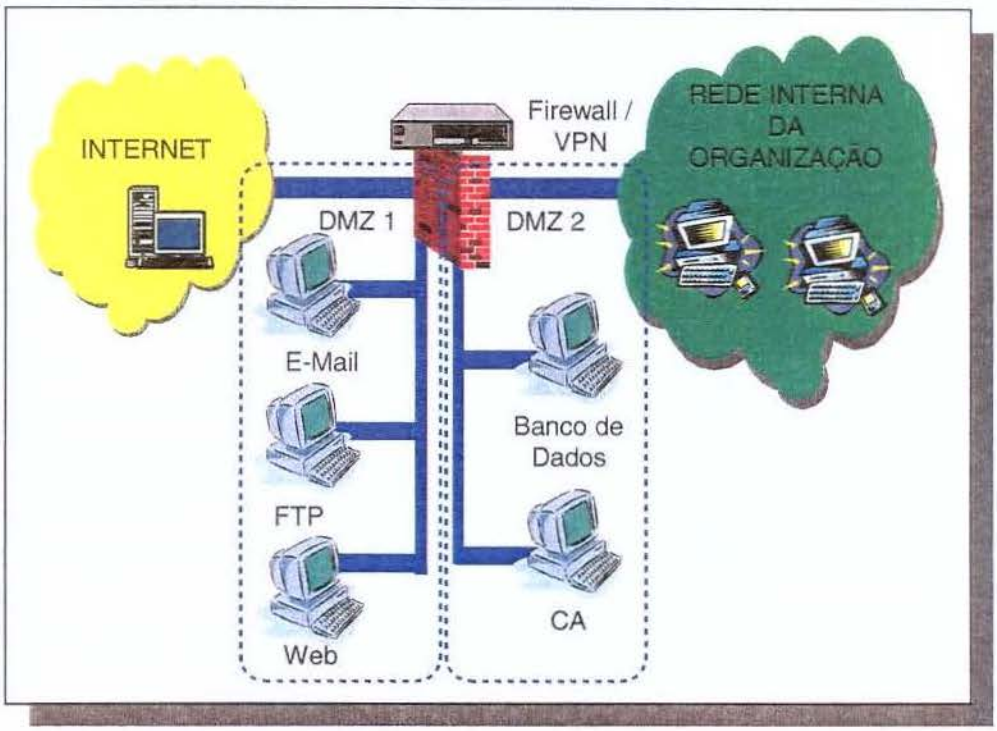


Figura 11.24: A VPN no *firewall*.

Essa configuração paralela (figura 11.25) elimina o problema da VPN constituir um único ponto de falha, porém faz com que a VPN esteja à mercê de possíveis ataques vindos da Internet. Esse esquema oferece ao *hacker* um outro caminho até a rede interna que pode ser explorado, sem que para isso ele tenha que passar necessariamente pelo *firewall*. O *hacker* assim não precisaria comprometer o *firewall* para chegar à rede interna da organização, mas apenas explorar a VPN.

Além disso, o mais importante a ser considerado nessa configuração é que a política de segurança implementada no *firewall* não será aplicada para as conexões VPN. Isso abre a possibili-

dade de que pacotes que normalmente seriam barrados pelo *firewall* cheguem até a rede interna através do túnel VPN, podendo causar assim sérios danos à organização.

11.2.5 Na Interface Dedicada do Firewall

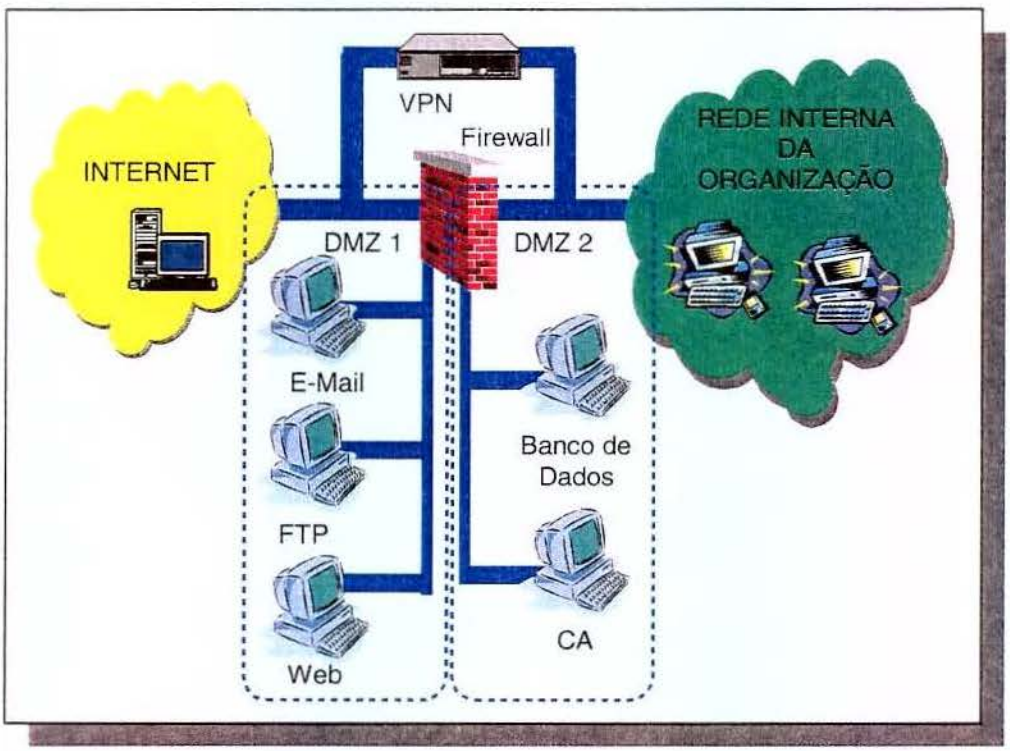


Figura 11.25: A VPN paralela ao *firewall*.

A utilização da VPN em uma interface dedicada do *firewall* (figura 11.26) é a mais indicada, pois o dispositivo VPN é protegido pelo *firewall* contra possíveis ataques vindos a partir da rede pública. O único ponto de falha desaparece, e o mais importante, todos os pacotes são filtrados de acordo com a política de segurança implementada no *firewall*. O funcionamento seria da seguinte maneira:

- Os pacotes IPSec seriam enviados somente para o dispositivo VPN, que realizaria a decifragem dos pacotes, e as entregaria de volta ao *firewall*, onde eles seriam então filtrados de acordo com a política de segurança implementada;
- Os pacotes não-IPSec seriam filtrados de acordo com a política de segurança implementada no *firewall*.

11.3 Conclusão

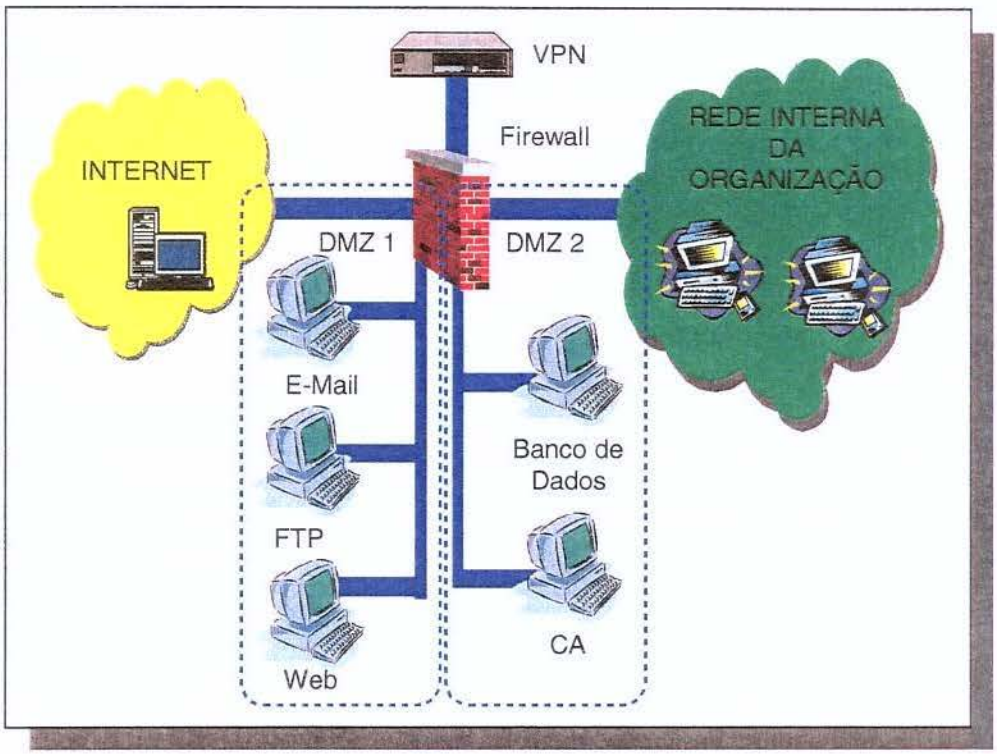


Figura 11.26: A VPN na interface dedicada do *firewall*.

Este capítulo teve como objetivo demonstrar a formação de um ambiente cooperativo e toda a complexidade envolvida, discutindo e analisando as possibilidades de configurações possíveis quanto aos diversos sistemas de segurança disponíveis - *firewall*, redes privadas virtuais (*Virtual Private Network* - VPN, sistemas de detecção de intrusão (*Intrusion Detection System* - IDS) e a infra-estrutura de chave pública (*Public Key Infrastructure* - PKI). O que pode ser observado através da análise da evolução que pode ocorrer na rede de uma organização (formação do ambiente cooperativo) é que, além do aumento da complexidade dos níveis de conexões, os usuários estão com necessidades cada vez maiores de acessar recursos internos da organização, como foi de fato visto no capítulo 5, quando foi proposto o modelo de "bolsões de segurança". A própria diferenciação entre usuários internos, usuários externos e usuários remotos parece estar desaparecendo, como pode ser visto no capítulo 12. Isso faz com que a proteção da rede

interna torne-se mais difícil de ser esquematizada e implementada, e para isso uma divisão em níveis de defesa torna a compreensão das necessidades de segurança mais objetiva e mais simplificada. Esses níveis de defesa serão discutidos no capítulo 12. Com isso, o *firewall* cooperativo ficou caracterizado, como será discutido no próximo capítulo. Este capítulo discutiu também o melhor posicionamento da VPN dentro da rede da organização.

Capítulo 12

O Modelo Proposto

Este capítulo tem como objetivo apresentar o modelo de segurança proposto para o ambiente cooperativo. As primeiras seções mostram os aspectos envolvidos com o ambiente cooperativo, e a seguir as dificuldades na definição e implementação das regras de filtragem são demonstradas. A seguir uma abordagem para a manipulação da complexidade das regras de filtragem é discutida (*iptables*). A arquitetura do *firewall* cooperativo é demonstrada na seção a seguir, culminando na definição de 5 níveis hierárquicos de defesa, que visa minimizar a complexidade e tornar mais claro a administração da segurança em um ambiente cooperativo.

12.1 Os Aspectos Envolvidos no Ambiente Cooperativo

Um ambiente cooperativo apresenta uma enorme complexidade, de tal modo que a administração de sua segurança torna-se difícil e passível de erros, que acabam por comprometer a segurança da organização como um todo. Alguns dos aspectos envolvidos são a diferenciação entre os diversos usuários existentes, os desafios a serem enfrentados em um ambiente cooperativo, e a complexidade das conexões desse ambiente, que serão analisados a seguir.

12.1.1 Usuários Internos X Usuários Externos X Usuários Remotos

Nesse ambiente cooperativo resultante da globalização, dos novos negócios, das fusões, das aquisição, das parcerias e das reestruturações, será que faz sentido se pensar em diferenciar usuários internos dos usuários externos? Se sim, como diferenciar esses usuários? Através do nome de acesso e de senha? Através de endereços IPs? E os usuários móveis? Como controlá-los? [SEC

98] sustenta que não existe mais a distinção entre usuários internos e usuários externos, o que de fato pode ser considerado uma realidade.

Foi visto que, em um ambiente cooperativo, os diferentes tipos de usuários passam a acessar diferentes bolsões de segurança (capítulo 5), e através da VPN (capítulo 9) os funcionários podem acessar os recursos da organização como se estivessem de fato nela. Com os bolsões de segurança e os acessos externos para recursos internos, a diferenciação entre usuários internos, usuários externos e usuários remotos (via cliente VPN ou mesmo acessos *dial-up*) passa a sofrer algumas indagações. Uma vez que esses diferentes tipos de usuários passam a acessar cada vez mais recursos internos da organização, o mais prudente é não fazer essa diferenciação, e tratar a segurança no seu nível interno. A segurança interna passa dessa forma a ser essencial nos ambientes cooperativos, para garantir que os recursos sejam acessados somente por usuários autorizados. De fato, isso está de acordo com o que foi discutido na seção 5.11, onde foi verificado que em um ambiente cooperativo cada organização deve tomar as devidas medidas de segurança, de acordo com a sua própria política de segurança. E, como os bolsões de segurança são formados em parte pela rede interna da organização, então a rede interna deve ser protegida.

Assim, a evolução mostra claramente que a segurança interna da organização passa a ser imprescindível em ambientes cooperativos. O controle do universo de usuários e a definição dos recursos que cada usuário pode acessar deve ser realizado com extremo cuidado, e para isso foi visto que a PKI (seção 8.6) e o IDS (capítulo 7) são importantes. As seções a seguir mostram a complexidade envolvida com o ambiente cooperativo, e também as propostas para facilitar o gerenciamento dessa complexidade.

12.1.2 O Desafio no Ambiente Cooperativo

Pôde-se observar, através dos cenários apresentados no capítulo 11, a formação de um ambiente cooperativo. O maior desafio a ser enfrentado nesse tipo de ambiente é o modo de lidar com a complexidade resultante dos diferentes níveis de acessos existentes, sem comprometer a segurança do ambiente cooperativo, e também dos integrantes desse ambiente.

Quando diversas conexões passam a se misturar, o risco de interferências e possibilidades de acessos a conexões de outros usuários torna-se maior, se não houverem regras e proteções específicas. Esse tipo de risco pode ser observado, por exemplo, em provedores de acesso, principalmente nos provedores de acesso via cabo, onde um usuário pode acessar sem restrições o

computador de um outro usuário do mesmo provedor. Se for considerado que uma organização é o provedor de serviços e acessos em um ambiente cooperativo, cuidados devem ser tomados com relação a esse risco.

Além dos riscos de acessos não autorizados entre usuários da rede, é preciso que uma idéia a princípio paradoxal seja tratada: a necessidade de abrir a rede para acessos externos, sendo que antes o objetivo era não permitir que nenhum acesso externo atingisse a rede da organização. Em outras palavras, se antes o objetivo era isolar a rede interna da rede pública, o ambiente cooperativo agora requer que os acessos via rede pública tornem-se mais constantes, sendo portanto essencial o controle sobre todas essas conexões.

Assim, o foco agora muda de "muros altos" para "controle dos usuários que acessam a rede". Dessa forma, ter o controle dos recursos que cada usuário pode acessar passa a ser essencial, e ter a certeza de que eles estão fazendo exatamente aquilo que são explicitamente permitidos é uma questão vital para o sucesso do sistema de segurança. Para isso, não basta apenas controlar, sendo necessário também monitorar as atividades dos usuários. O próprio conceito de diferenciar usuários internos de usuários externos, e também de usuários remotos, sofre algumas alterações e questionamentos, como pôde ser visto na seção 12.1.1.

12.1.3 A Complexidade das Conexões

Os níveis de acesso e, conseqüentemente, seus métodos de controle, atingem rapidamente um grau de complexidade bastante alto em um ambiente cooperativo, tornando o seu gerenciamento bastante trabalhoso e passível de erros. Isso pode ser observado, por exemplo, em um ambiente onde a organização tem que controlar acessos de 30 conexões diferentes, que acessam serviços diferentes entre si. Pode-se considerar um ambiente onde a filial A tem acesso a serviços como Intranet, banco de dados financeiro, sistema de logística de peças e ao servidor de *e-mails*. O fornecedor A tem acesso a serviços como o sistema de logística de peças, bem como ao servidor FTP. Já o fornecedor B tem acesso somente ao sistema de controle de estoques, para poder agilizar o processo de reposição de peças. Um representante comercial tem acesso ao sistema de estoques, ao sistema de logística e ao sistema de preços. Os clientes da organização têm acesso aos sistemas de estoques e de preços, para poder verificar a disponibilidade e os preço dos produtos.

Neste exemplo foram vistos somente 5 diferentes tipos de conexões, que já mostram a necessidade de se criar um modelo de segurança para melhor gerenciamento das conexões. Se for levado em consideração que em cada elemento do ambiente cooperativo ainda existem diferentes níveis de usuários, a complexidade do ambiente passa a ser ainda maior.

Assim, pode-se verificar que a conjuntura dentro de um ambiente cooperativo, com sua enorme complexidade, faz com que o modelo de segurança desse ambiente deva ser bem planejado. Dois tópicos principais merecem destaque dentro desse modelo de segurança: as regras de filtragem e a arquitetura de segurança, que resulta no *firewall* cooperativo. As próximas seções abordam esses dois elementos principais, e através dessa análise um modelo de segurança dividido em níveis hierárquicos de defesa é proposto.

12.2 As Regras de Filtragem

Diferentemente da abordagem recente, onde os *firewalls* eram utilizados para isolar a rede interna do mundo externo, no ambiente cooperativo essa idéia sofre alterações. A abordagem recente permitia que as regras de filtragem fossem extremamente simples, onde a organização geralmente utilizava uma das abordagens "o padrão é liberar todos os serviços e negar todos os serviços explicitamente proibidos", ou "o padrão é proibir todos os serviços e liberar somente aqueles que são explicitamente permitidos". A segunda abordagem é que permitia um maior grau de segurança, pois serviços novos e desconhecidos sempre trazem dúvidas tanto quanto à sua importância e também quanto à sua segurança, sendo assim recomendável evitar a sua utilização. Alguns exemplos das implicações que a liberação impensada de quaisquer serviços podem trazer podem ser vistas principalmente em serviços Web, como o ICQ (diversas vulnerabilidades) e o Real Audio (torna o ambiente improdutivo, além de ser grande devoradora de largura de banda).

Essa abordagem recente possui uma característica, que é a simplicidade de suas regras de filtragem, resultante justamente da simplicidade das conexões, como pôde ser visto no capítulo 11. Como poucos serviços eram oferecidos, poucas regras eram necessárias. As regras praticamente se resumiam em liberar os serviços oferecidos para os usuários externos, tais como o HTTP, FTP e SMTP, e criar as regras para os serviços que os usuários internos poderiam acessar.

Já a complexidade em um ambiente cooperativo pode trazer grandes problemas de segurança e desempenho à organização. Os problemas de segurança podem aparecer devido a dois fatores: erro na definição e criação dessas regras ou erro na implementação dessas regras. De

fato, em um ambiente com 30 diferentes níveis de conexões, erros humanos tornam-se bastante prováveis, além de serem praticamente impossíveis de se gerenciar.

Os problemas de desempenho tornam-se também evidentes quando as regras de filtragem atingem um tamanho estratosférico. Em geral, o *firewall* analisa e toma decisões de acordo com uma análise contínua e seqüencial das regras de filtragem. O Cisco, por exemplo, faz a análise de todos os pacotes, um por um, em busca de uma regra de filtragem compatível. Essa análise das regras é feita seqüencialmente, ou seja, o Cisco pega as informações do pacote a ser analisado, e passa a compará-la com a primeira regra de filtragem. Caso ela não esteja de acordo, analisa-se as informações do pacote de acordo com a segunda regra de filtragem, e assim por diante, até que uma regra em que o pacote se encaixa seja encontrada. E quando uma regra é encontrada, a decisão de liberar, descartar ou bloquear o pacote é tomada de acordo com a política de segurança definida. Se for considerado que cada pacote tem que passar por essa análise através de um imenso conjunto de regras, é fácil observar a perda de desempenho que irá ocorrer no *firewall*, que irá rapidamente se tornar o gargalo de toda a comunicação da organização.

O poder de processamento dos equipamentos tecnológicos vêm acompanhando a Lei de Moore, dobrando a sua capacidade de processamento a cada 18 meses. Porém a demanda por velocidade na análise das regras de um ambiente cooperativo mostra-se ainda maior, de modo que uma nova abordagem deve ser tomada. Depender do aumento do poder de processamento tornou-se inadequado, e uma nova maneira de se criar e analisar filtros traz grandes benefícios à organização. Uma das abordagens úteis pode ser vista no modo de funcionamento do NetFilter, que será visto na seção 12.3. Um exemplo da criação de regras de filtragem pode ser visto na seção a seguir, onde o Laboratório de Administração e Segurança (LAS) serviu de cenário para o seu desenvolvimento.

12.2.1 Exemplos de Filtragem - Laboratório de Administração e Segurança (LAS)

Um exemplo das dificuldades na criação das regras de filtragem pôde ser observado na implementação das regras do Laboratório de Administração e Segurança do Instituto de Computação da Unicamp (LAS-IC-Unicamp).

A arquitetura do LAS-IC-Unicamp foi planejada de acordo com o esquema que pode ser visto na figura 12.

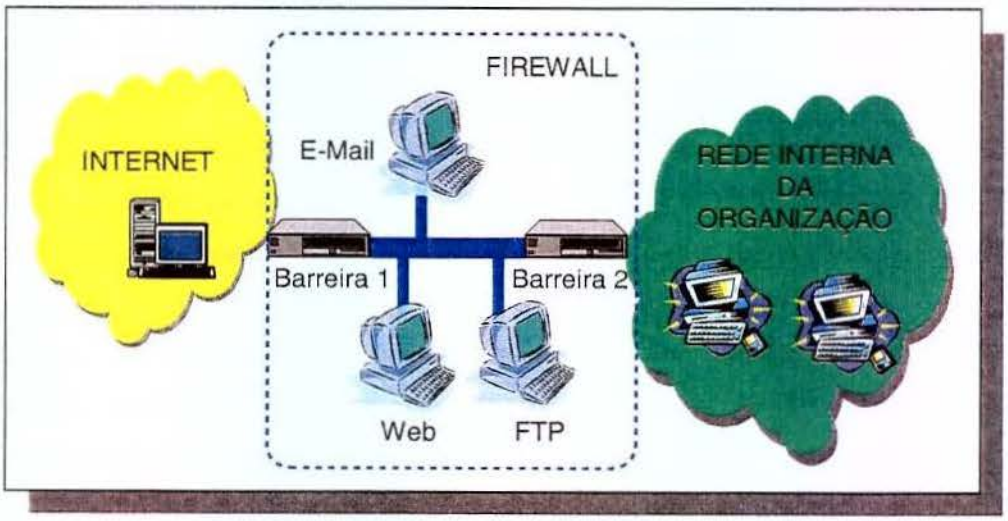


Figura 12.1: O esquema utilizado pelo LAS.

Optou-se pela utilização de dois componentes, onde a barreira 1 funciona como filtro de pacotes, enquanto a barreira 2 trabalha como *proxy* dos serviços a serem acessados pelos usuários da rede interna. O *proxy* protege a rede interna contra as tentativas de conexões indevidas, e as regras de filtragem aqui definidas foram aplicadas na barreira 1.

O primeiro passo foi a definição dos serviços da rede, em dois aspectos:

- Os serviços providos pelo LAS-IC-Unicamp para a Internet – HTTP, FTP, SSL, DNS, SSH, SMTP;
- Os serviços utilizados pelos usuários do LAS – HTTP, FTP, SSL, DNS, SSH, SMTP.

Os serviços externos seriam acessados pelos usuários do LAS através de *proxies*. A rede DMZ, que aloca os serviços providos pelo LAS, foi definida como sendo entre o filtro de pacotes (barreira 1) e os *proxies* (barreira 2).

A política de segurança geral das regras de filtragem definida foi a de permitir somente os serviços explicitamente permitidos e negar todos os outros serviços. Essa abordagem é o que garante o maior nível de segurança, ainda mais atualmente, que com a complexidade dos serviços e do seu número cada vez maior, torna-se extremamente difícil e inviável negar todos os serviços que não são permitidos. De fato, essa é a abordagem padrão da maioria dos *firewalls*

para a filtragem dos pacotes, inclusive utilizada pelos roteadores da Cisco, que foi utilizado para realizar a filtragem dos pacotes do LAS.

Diversas complicações apareceram durante o desenvolvimento das regras de filtragem, que serão relatadas a seguir. Algumas dessas dificuldades foram com relação à criação das regras de filtragem permitindo o acesso de equipamentos específicos a serviços específicos, principalmente com relação à uma máquina de ataques que seria utilizada para a auditoria de segurança na Internet. Uma outra dificuldade foi a de encontrar uma maneira de criar as regras de modo que, caso um novo equipamento fosse adicionado na rede, não fosse necessário modificar as regras ou criar regras específicas para esse equipamento em especial.

Como a tecnologia utilizada foi o filtro de pacotes, foi necessário definir regras para 4 canais de conexões, que podem ser vistas nas figuras 12.2 e 12.3. Essas figuras foram divididas em duas para uma melhor visualização, porém poderiam ser mescladas em uma só:

- Canal de requisição a partir dos usuários internos;
- Canal de resposta das requisições dos usuários internos;
- Canal de requisição de serviços providos pelo LAS a partir da Internet;
- Canal de resposta dos serviços requisitados pelos usuários da Internet.

Caso um filtro de estados fosse utilizado, não seria necessário definir esses 4 canais, pois as respostas seriam aceitas de acordo com as conexões já abertas, consultadas na tabela de estados, sendo que essas conexões só seriam abertas de acordo com as regras definidas (seção 6.3.2).

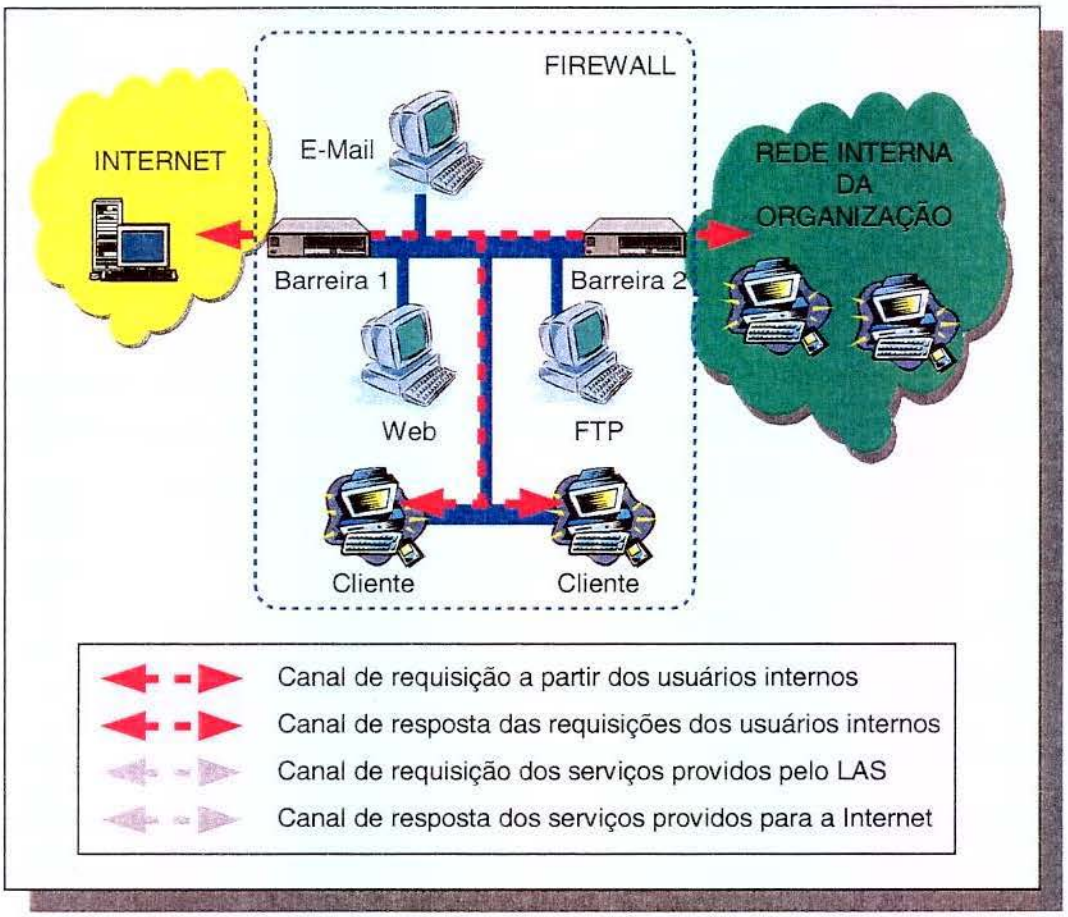


Figura 12.2: Os canais utilizados pelos usuários internos.

Pode-se verificar que os canais de comunicação são bem definidos, ou seja, o caminho que cada canal tem que tomar, bem como a sua direção e os serviços que passam por eles, estão bem definidos. As regras assim têm que ser definidas de acordo com esses canais.

Foi visto na seção 12.2 que a ordem das regras de filtragem é importante, pois a sua verificação se dá em uma ordem seqüencial. Assim, deve-se lembrar que as regras mais específicas devem sempre ser criadas antes das regras mais gerais, pois se uma das regras estiver no contexto do pacote, então será essa a regra que será utilizada.

As regras mais específicas são as que eliminam as possibilidades de ataques comuns ao protocolo TCP/IP, tais como o *IP Spoofing* ou o *Smurf*. As regras a seguir evitam o *IP Spoofing* e a utilização de endereços de *broadcast* e *multicast*, que de fato não devem entrar na rede da orga-

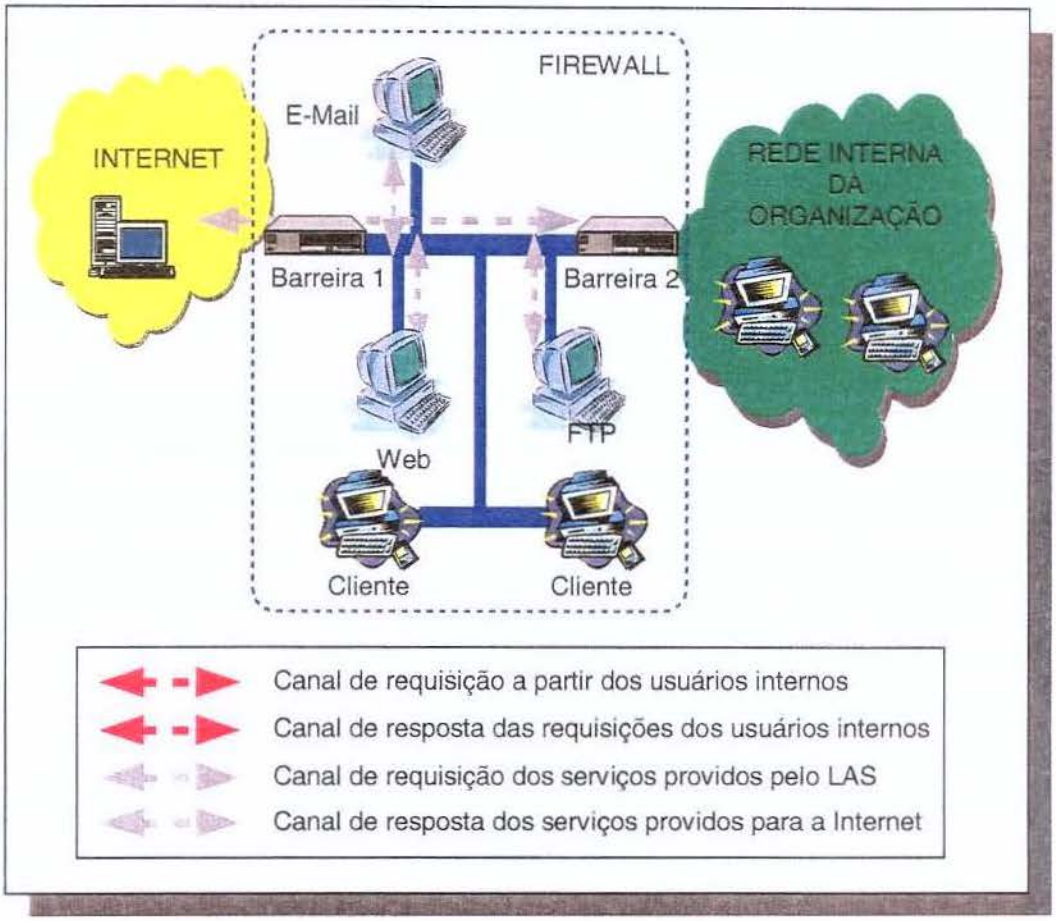


Figura 12.3: Os canais utilizados pelos usuários vindos da Internet.

nização se vindos da Internet. O canal que tem a direção da Internet para a rede da organização é chamada de 112.

```
!! Descarta os endereços do rfc 1918 (10.0.0.0 – 10.255.255.255; 172.16.0.0 –
172.31.255.255); 192.168.0.0 – 192.168.255.255)
access-list   112   deny   ip   10.0.0.0       0.255.255.255   any
access-list   112   deny   ip   172.16.0.0     0.15.255.255    any
access-list   112   deny   ip   192.168.0      0.0.255.255     any
!
!! Impede IP Spoofing de endereços da rede DMZ (143.106.60.0 – 143.106.60.255)
access-list   112   deny   ip   143.106.60.0   0.0.0.255       any
!
```

```
!! Impede IP Spoofing de endereços loopback (127.0.0.0 – 127.255.255.255)
access-list   112   deny   ip   127.0.0.0   0.255.255.255   any
!
!! Impede Smurf, Teardrop, que usam endereço de broadcast (255.0.0.0-
255.255.255.255)
access-list   112   deny   ip   255.0.0.0   0.255.255.255   any
!
!! Impede endereço de multicast (224.0.0.0-231.255.255.255)
access-list   112   deny   ip   224.0.0.0   7.255.255.255   any
```

Essas regras devem estar no início do conjunto de regras, para que caso algum deles estiver sendo utilizado, o pacote seja prontamente descartado. As regras a seguir dizem respeito aos serviços providos para a Internet pelo LAS. O canal 112 é utilizado para as requisições vindos da Internet, e o canal 111 é utilizado para as respostas dessas requisições, ou seja, possui a direção da rede da organização para a Internet.

4 canais foram definidos, porém o canal 112 é utilizado também para as respostas das requisições dos usuários internos, e o canal 111 é utilizado também para as requisições dos usuários internos.

Assim, a estratégia utilizada foi a de definir primeiramente os serviços que os usuários externos podem acessar (canal 112 para as requisições e o canal 111 para as respostas), e depois os serviços que os usuários internos podem acessar (canal 111 para as requisições e o canal 112 para as respostas).

Assim, os canais utilizados pelos usuários externos foram definidos, sendo primeiramente o canal de requisição:

```
!! Permite tráfego HTTP (porta 80) para o Bastion (143.106.60.15)
access-list   112   permit tcp   any gt 1023   host 143.106.60.15 eq 80
!
!! Permite o tráfego FTP (porta 21) para o Bastion (143.106.60.15)
access-list   112   permit tcp   any gt 1023   host 143.106.60.15 eq 21
access-list   112   permit tcp   any eq 20     host 143.106.60.15 gt 1023
access-list   112   permit tcp   any gt 1023   host 143.106.60.15 gt 1023
!
!! Permite o tráfego SSL (porta TCP 443) para Bastion (143.106.60.2)
access-list   112   permit tcp   any gt 1023   host 143.106.60.15 eq 443
```

```
!
!! Permite o tráfego SMTP (porta 25) para o proxy (143.106.60.2)
access-list    112    permit tcp    any gt 1023    host 143.106.60.2 eq 25
!
!! Permite o tráfego SSH (porta 22) para maquinas da DMZ (143.106.60.2-60.254)
access-list    112    permit tcp    any gt 1023    143.106.60.2 0.0.0.252 eq 22
!
!! Permite o tráfego DNS (porta UDP 53) para o proxy (143.106.60.2)
access-list    112    permit udp    any gt 1023    host 143.106.60.2 eq 53
!
!! Bastion (143.106.60.15) nega o resto
access-list    112    deny ip any    host 143.106.60.15
!
```

O canal de resposta para as requisições vindas da Internet foram assim definidas:

```
!! Permite tráfego HTTP (porta 80) para o Bastion (143.106.60.15)
access-list    111    permit tcp host 143.106.60.15 eq 80 any gt 1023
!
!! Permite o tráfego FTP (porta 21) para o Bastion (143.106.60.15)
access-list    111 permit tcp host 143.106.60.15 eq 21 any gt 1023
access-list    111 permit tcp host 143.106.60.15 gt 1023 any eq 20
access-list    111 permit tcp host 143.106.60.15 gt 1023 any gt 1023
!
!! Permite o tráfego SSL (porta TCP 443) para Bastion (143.106.60.2)
access-list    111 permit tcp host 143.106.60.15 eq 443 any gt 1023
!
!! Permite o tráfego SMTP (porta 25) para o proxy (143.106.60.2)
access-list    111 permit tcp host 143.106.60.2 eq 25 any gt 1023
!
!! Permite o tráfego SSH (porta 22) para maquinas da DMZ (143.106.60.2-60.254)
access-list    111 permit tcp 143.106.60.2 0.0.0.252 eq 22 any gt 1023
!
!! Permite o tráfego DNS (porta UDP 53) para o proxy (143.106.60.2)
access-list    111 permit udp host 143.106.60.2 eq 53 any gt 1023
!
```

A segunda parte da estratégia foi a de definir os canais dos serviços a serem acessados pelos usuários internos. Primeiro foi definido o canal de requisição, lembrando que as regras mais específicas têm que vir antes das regras mais gerais. Nesse caso as regras mais específicas significam que as regras para determinados equipamentos devem vir antes:

```
!! BlackHole (143.106.60.14) (portas 6000-6010) pode requisitar serviços X
access-list    111 permit tcp host 143.106.60.14 gt 1022 any
access-list    111 permit tcp host 143.106.60.14 range 6000 6010 any gt 1022
!
! Permite a requisiçao de serviços HTTP (porta 80)
access-list    111 permit tcp 143.106.60.2 0.0.0.29 gt 1023 any eq 80
!
!! Permite a requisiçao de serviços SMTP (porta 25)
access-list    111 permit tcp 143.106.60.2 0.0.0.29 gt 1023 any eq 25
!
!! Permite a requisiçao de serviços SSH (porta 22)
access-list    111 permit tcp 143.106.60.2 0.0.0.29 gt 1023 any eq 22
!
!! Permite a requisiçao de serviços FTP (porta 21)
!! SOMENTE MODO PASSIVO - Canal de requisicoes
access-list    111 permit tcp 143.106.60.2 0.0.0.29 gt 1023 any eq 21
!
!! Permite a requisiçao de serviços FTP (porta > 1023)
!! Canal de dados
access-list    111 permit tcp 143.106.60.2 0.0.0.29 gt 1023 any gt 1023
!
!! Liberando FTP modo ativo somente para o proxy (143.106.60.2)
access-list    111 permit tcp host 143.106.60.2 gt 1023 any eq 20
!
!! Permite a requisiçao de serviços DNS (porta UDP 53)
access-list    111 permit udp 143.106.60.2 0.0.0.29 gt 1023 any eq 53
!
!! Permite a requisiçao de serviços SSL (porta TCP 443)
access-list    111 permit tcp 143.106.60.2 0.0.0.29 gt 1023 any eq 443
!
```

O canal de resposta das requisições dos usuários da rede interna foi assim definida:

```
!! Canal de resposta X para o BlackHole (143.106.60.14) (portas 6000-6010)
access-list     112 permit tcp any host 143.106.60.14 gt 1022
access-list     112 permit tcp any gt 1022 host 143.106.60.14 range 6000 6010

!
!! Permite clientes utilizarem o HTTP (porta 80)
access-list     112 permit tcp any eq 80 143.106.60.2 0.0.0.29 gt 1023
!
!! Permite clientes utilizarem o SMTP (porta 25)
access-list     112 permit tcp any eq 25 143.106.60.2 0.0.0.29 gt 1023
!
!! Permite clientes utilizarem o SSH (porta 22)
access-list     112 permit tcp any eq 22 143.106.60.2 0.0.0.29 gt 1023
!
!! Permite clientes utilizarem o FTP (porta 21)
!! SOMENTE MODO PASSIVO - Canal de requisicao
access-list     112 permit tcp any eq 21 143.106.60.2 0.0.0.29 gt 1023
!
!! Permite clientes utilizarem o FTP (porta 21)
!! SOMENTE MODO PASSIVO - Canal de dados
access-list     112 permit tcp any gt 1023 143.106.60.2 0.0.0.29 gt 1023
!
!! Liberando FTP modo ativo somente para o proxy (143.106.60.2)
access-list     112 permit tcp any eq 20 host 143.106.60.2 gt 1023
!
!! Permite clientes utilizarem o DNS (porta UDP 53)
access-list     112 permit udp any eq 53 143.106.60.2 0.0.0.29 gt 1023
!
!! Permite clientes utilizarem o SSL (porta TCP 443)
access-list     112 permit tcp any eq 443 143.106.60.2 0.0.0.29 gt 1023
!
```

Para finalizar a implementação das regras de filtragem, todo o resto é negado. Apesar dessa abordagem ser padrão, colocar explicitamente essas regras não deixa margem a dúvidas sobre o que acontecerá com os pacotes que passam pelo processo de filtragem.

```
!! ****************************************
```

```
!! NEGA TODO O RESTO
!! ****************************************
access-list    111 deny ip any any
access-list    112 deny ip any any
!
```

Uma das dificuldades encontradas foi o estabelecimento de regras de filtragem para o equipamento de auditoria que se localiza na rede DMZ. Esse equipamento é o responsável pela realização de simulações de ataques em redes que devem ter a sua segurança avaliada. Embora não esteja diretamente mapeada para o caso de ambiente cooperativo, essa situação oferece um bom estudo de caso. Como poderá ser observado, esse caso demonstra o antagonismo de tráfegos diferentes e a dificuldade que se tem em definir regras de filtragem que satisfaçam tais disparidades.

Esse equipamento deve ter um conjunto específico de regras, já que todos os tipos de pacotes devem ser permitidos. As regras podem ser:

```
access-list   111   permit ip   host   143.106.60.30   any
access-list   112   permit ip   any                    host   143.106.60.30
```

Essas regras permitem que qualquer tipo de pacote possa trafegar entre a Internet e o *host* de ataque. Porém, isso limita o escopo dos ataques, já que a maioria deles utilizam o *IP Spoofing* para mascarar a sua origem. Essas regras não permitem que essa técnica seja utilizada, o que traz como conseqüência a necessidade de mudanças nas regras de filtragem. Porém, uma outra questão surge com a possibilidade de utilização do *IP Spoofing*. Um ponto a ser considerado é que geralmente os pacotes de resposta não são requeridos. Receber respostas quando o *IP Spoofing* é utilizado é uma tarefa complicada, já que a rota para o endereço falsificado não aponta para o equipamento que fez o ataque, mas sim para o equipamento real, que teve o seu endereço utilizado indevidamente. Então, para que respostas sejam recebidas, é necessário que outras técnicas sejam utilizadas, como ataques a roteadores para a alteração da tabela de rotas ou a utilização de *source routing*. Assim, existem 2 possibilidades quando se utiliza o *IP Spoofing*:

- Existe a necessidade de receber os pacotes de resposta da vítima;
- Não existe a necessidade de receber as respostas da vítima.

De acordo com um dos casos acima, as regras de filtragem podem influenciar diretamente na segurança dos outros *hosts* da organização.

Caso a resposta não seja necessária, a regra de filtragem, única, pode ser:

```
access-list   111   permit      ip     any          any
```

As regras para o canal de resposta vindos da Internet não seriam necessárias, já que não existiriam respostas. O que estaria valendo seriam as regras já definidas anteriormente, de acordo com a política de segurança da organização. A conseqüência prática da utilização dessa nova regra é que qualquer pacote saindo a partir da rede da organização é permitido, tornando possível que qualquer *host* da organização se torne a base para a realização de ataques. Além disso, agora os usuários podem fazer a requisição que desejarem, de quaisquer serviços. Porém, a segurança da rede da organização ainda está no mesmo nível da anterior, pois somente os pacotes dos serviços permitidos passam pelo filtro. Em outras palavras, pacotes estranhos e respostas de requisições inválidas vindos da Internet são bloqueados.

No outro caso, onde as respostas são necessárias, as únicas regras que permitem isso são:

```
access-list   111   permit      ip     any          any
access-list   112   permit      ip     any          any
```

Com essas regras, o *host* de ataque pode utilizar *IP Spoofing* para realizar um ataque, e tem condições de receber uma resposta. A conseqüência dessa regras é que a rede da organização também passa a poder fazer requisições e receber respostas e requisições de quaisquer tipos, ou seja, é a mesma situação onde o acesso é totalmente transparente, sem nenhum filtro.

A conclusão que se obtém com essa experiência, é a de que um ambiente de produção não combina com um ambiente de ataques, ou seja, ambos são opostos. Caso seja possível, a utilização de uma outra interface de rede para a criação de uma sub-rede somente para o *host* de ataque (figura 12.4) passa a ser imprescindível. Caso isso não seja possível, ataques mais sofisticados, que requerem respostas, ficam comprometidos. O ponto mais importante é que a segurança da organização deve receber a máxima prioridade.

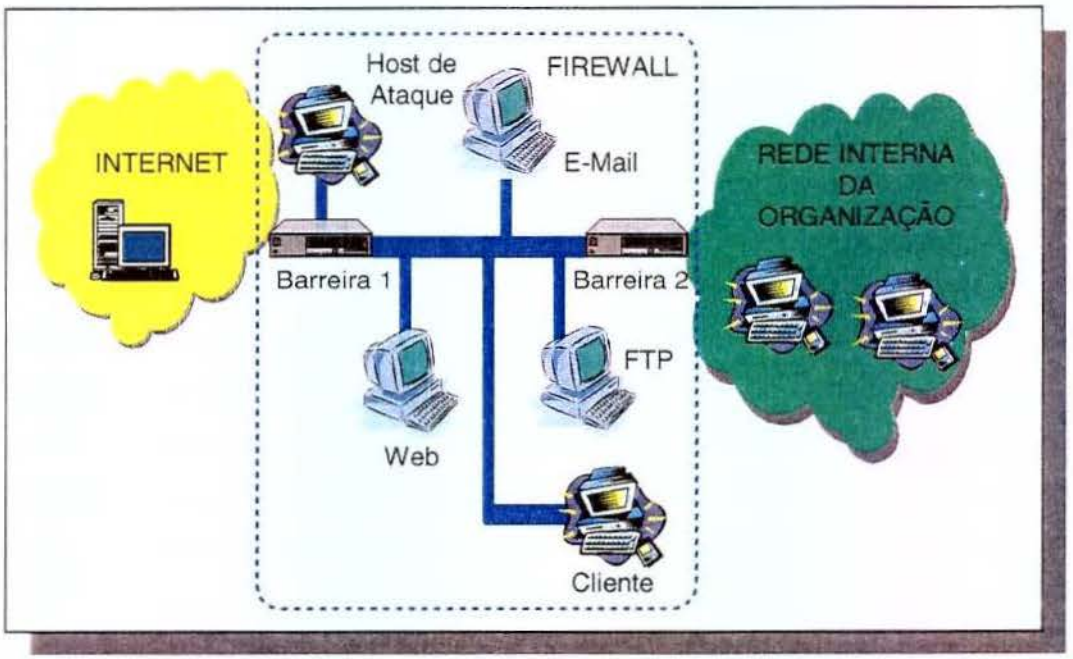


Figura 12.4: A arquitetura e equipamentos utilizados pelo LAS.

O ambiente do LAS é um caso típico de uma rede, como já pôde ser visto no capítulo 11, e mesmo assim algumas variáveis faz com que as regras de filtragem atinjam um alto nível de complexidade. O exemplo do LAS ilustra a dificuldade que pode aparecer no desenvolvimento da política de segurança para ambientes cooperativos, que envolve inúmeras conexões diferentes. Foi visto que as dificuldades na definição do conjunto de regras são grandes quando os objetivos são antagônicos. E essa situação é bastante provável de ser encontrada em um ambiente cooperativo. Provavelmente a dificuldade surge da planaridade dos mecanismos comuns de filtragem, ou seja, da ausência de mecanismos hierárquicos de filtragem ou de blocos, como as utilizadas em linguagens de programação estruturada. Uma característica do *netfilter* pode ser utilizado para facilitar o desenvolvimento das regras de filtragem, como pode ser visto na seção a seguir. Somado a isso, a arquitetura do *firewall cooperativo* e a divisão em níveis hierárquicos de defesa também auxiliam na definição de um conjunto de regras complexo, como o que é encontrado em um ambiente cooperativo. Os níveis hierárquicos serão vistos na seção 12.5.

12.3 Manipulação da Complexidade das Regras de Filtragem

De acordo com as seções anteriores, a complexidade das regras de filtragem cresce cada vez mais em ambientes cooperativos, e uma forma de facilitar o seu gerenciamento torna-se um fator importante para que erros em sua criação e implementação sejam minimizados. Além disso, existe ainda o fator desempenho, que é prejudicado quando o número de regras de filtragem que deve ser verificado por cada pacote é muito grande, o que é de fato uma característica dos ambientes cooperativos. Um dos sistemas de filtragem que tenta resolver os problemas levantados até agora é o *iptables*, que faz parte do Linux. O *kernel* do Linux possui o filtro de pacotes desde a versão 1.1. A primeira geração foi baseada no *ipfw* do BSD, e foi portado por Alan Cox em 1994. Ele foi melhorado por Jos Vos no Linux 2.0, quando passou a se chamar *ipfwadm*. Em meados de 1998, o *ipchains* apareceu no Linux 2.2. A quarta geração é o *iptables*, que começou a ser desenvolvido em meados de 1999 para a inclusão na versão 2.4 (versão de produção), previsto para liberação em meados de agosto de 2000. O *netfilter* é um *framework* para o manipulamento de pacotes, onde diversos ganchos (*hooks*) são criados na pilha do protocolo IPv4. O *iptables* utiliza os *hooks* definidos no *netfilter* para a realização da filtragem de pacotes. As próximas seções mostram o funcionamento do *iptables*, e também do *netfilter*.

12.3.1 IPTables

Em sistemas que utilizam o *iptables*, o *kernel* é inicializado com 3 listas de regras, que são também chamadas de *firewalls chains* ou apenas *chains* (cadeias), que são o *INPUT*, *OUTPUT* e o *FORWARD*.

O funcionamento do *iptables* é de acordo com o diagrama da figura 12.5.

Quando o pacote atinge uma das cadeias, o pacote é examinado. Se a cadeia diz para descartar o pacote, ele é descartado nesse ponto; se a cadeia diz para aceitar o pacote, então ele continua a percorrer o diagrama da figura 12.5.

Cada uma dessas cadeias são um conjunto de regras que são examinadas uma a uma, seqüencialmente. Caso nenhuma regra da cadeia seja verificada, a próxima cadeia é verificada. Caso não haja regras em nenhuma cadeia, então a política padrão da cadeia é utilizada, o que geralmente é descartar o pacote.

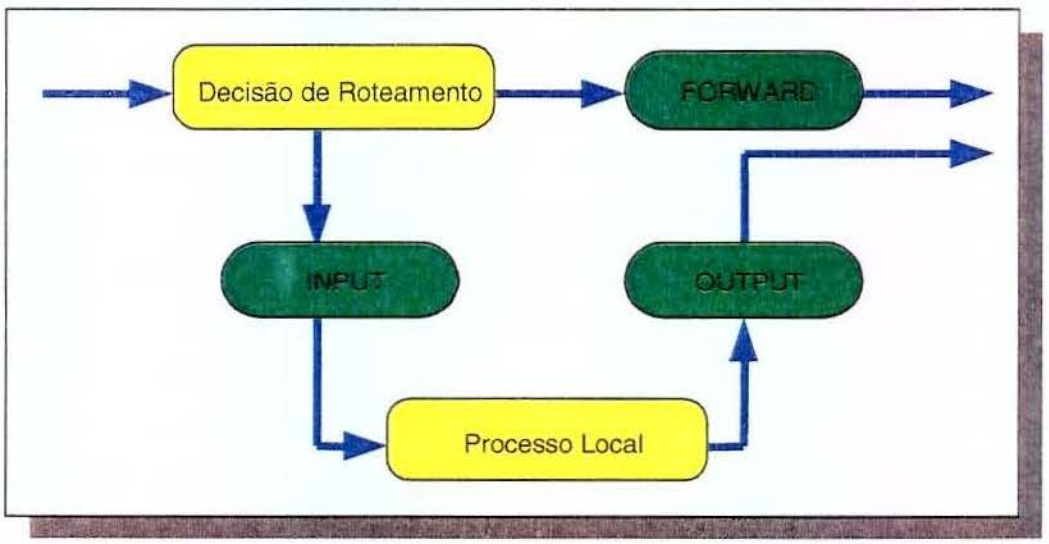


Figura 12.5: O funcionamento do *iptables*.

O modo de funcionamento do *iptables* pode ser resumido da seguinte forma:

- Quando um pacote é recebido através da placa de rede, o *kernel* primeiro verifica qual o seu destino (decisão de roteamento);
- Se o destino é o próprio equipamento, então o pacote é passado para a cadeia INPUT. Se ele passa pelas regras dessa cadeia, então ele é repassado para o processo destino local, que está esperando pelo pacote;
- Se o *kernel* não possui o *forwarding* habilitado, ou se ele não sabe como encaminhar esse pacote, ele é descartado. Se o *forwarding* estiver habilitado em uma outra interface de rede, então ele irá para a cadeia FORWARD. Se ele passa pelas regras dessa cadeia, então o pacote é aceito e repassado para frente;
- Um programa rodando no equipamento pode enviar pacotes para a rede, que são enviadas para a cadeia OUTPUT. Caso esses pacotes sejam aceitos pelas regras existentes nessa cadeia, eles são enviados através da interface.

Essas 3 cadeias não podem ser removidas do *kernel*, sendo que o *iptables* possui as seguintes opções: criar uma nova cadeia, remover uma cadeia vazia, alterar a política da cadeia, listar as regras da cadeia, eliminar as regras de uma cadeia, zerar o contador de pacote e byte das regras da cadeia, adicionar uma nova regra na cadeia, inserir uma nova regra em alguma posição na

cadeia, remover uma regra de alguma posição na cadeia, e remover a primeira regra encontrada na cadeia.

As filtragens do *iptables* podem ser baseadas em endereços IP de origem e destino, protocolos e interface. O *iptables* é extensível, sendo que novas características podem ser adicionadas às regras. Algumas das extensões são o do TCP, quando o protocolo TCP está selecionado. Pode-se através deles criar regras utilizando-se *flags* TCP como o *syn*, *ack*, *fin*, *rst*, *urg*, *psh*, portas de origem ou portas de destino. Extensões UDP são a porta de origem ou destino, e as extensões ICMP permitem criar regras com tipos específicos ICMP. Outras extensões são referentes a endereços MAC, e limites de pacotes a serem registrados. Outro módulo é o de estados, que permite o controle das conexões: novas, estabelecidas, relatadas ou inválidas.

A fragmentação também pode ser controlada para que a filtragem seja realizada, não apenas no primeiro pacote da conexão, mas também nas subseqüentes. Quando o NAT é utilizado, os fragmentos são desfragmentados antes de chegarem ao código de filtragem do *kernel*.

O *iptables* insere e exclui regras na tabela de filtragem de pacotes do *kernel*, e é perdido quando o sistema operacional é reinicializado, sendo portanto necessário o seu armazenamento em algum local seguro.

Uma das principais diferenças entre o *ipchains* e o *iptables* é que as referências ao *redirect* e ao *masquerade*, existentes no *ipchains*, foram removidos do *iptables*, de modo que o *iptables* nunca deve alterar pacotes, apenas filtrá-las, simplificando assim o seu funcionamento. Como conseqüência dessa simplificação, pode-se observar um melhor desempenho e segurança.

12.3.2 Netfilter

O *netfilter* é um *framework* (figura 12.6) para o manipulamento de pacotes, onde diversos ganchos (*hooks*) são criados na pilha do protocolo IPv4. O *iptables* se baseia no *netfilter*, como foi visto na seção anterior.

Antes dos pacotes entrarem no *netfilter*, eles passam por algumas checagens simples, como descartar pacotes truncados ou com o *checksum* do IP errados, e também evitar recebimentos promíscuos. Os pacotes que passam por essa checagem entram no *netfilter* (figura 12.6) e vão para o *hook* NF_IP_PRE_ROUTING (1), e depois vão para o código de roteamento, onde é decidido se o pacote é destinado para uma outra interface, ou para um processo local. O código de roteamento pode descartar pacotes que não possuem rotas. Se o destino é o próprio equipa-

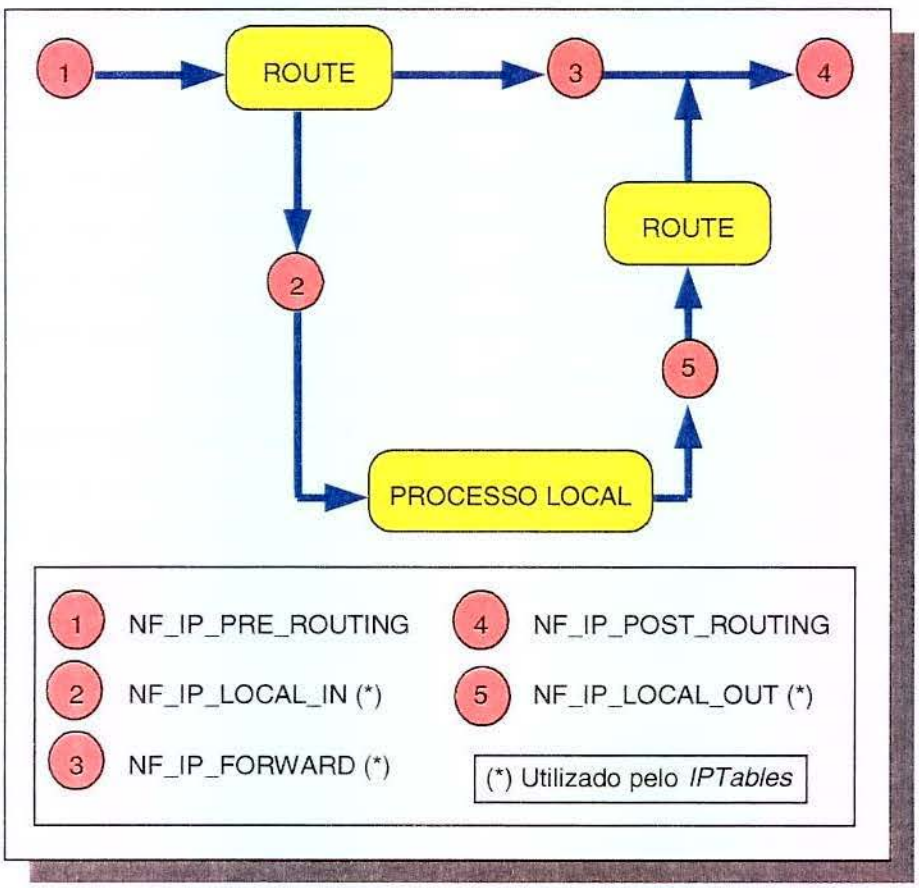


Figura 12.6: O funcionamento do *netfilter*.

mento, o *netfilter* envia os pacotes para o *hook* NF_IP_LOCAL_IN (2) antes deles chegarem ao processo local. Se o pacote tem como destino outra interface, o *netfilter* o envia para o *hook* NF_IP_FORWARD (3), e depois para o *hook* NF_IP_POST_ROUTING (4), antes de chegar ao cabo novamente. O *hook* NF_IP_LOCAL_OUT (5) é utilizado para os pacotes criados localmente. O roteamento é realizado após a chamada do *hook* (5), de modo que o código de roteamento é chamado primeiramente para descobrir o endereço IP de origem e algumas opções IP, sendo chamado novamente caso o pacote seja alterado.

Módulos do *kernel* podem ser registrados para atuar nos *hooks*, de modo que, quando os *hooks* são chamados, os módulos registrados estão livres para manipular os pacotes. O módulo pode então fazer com que o *netfilter* realize uma das funções:

- NF_ACCEPT – continua normalmente;
- NF_DROP – descarta o pacote;
- NF_STOLEN – não continua a manipulação do pacote;
- NF_QUEUE – coloca o pacote na fila.

O *iptables* utiliza os *hooks* do *netfilter* NF_IP_LOCAL_IN, NF_IP_FORWARD e o NF_IP_LOCAL_OUT, de modo que qualquer pacote passa por um único *hook* para a filtragem.

12.3.3 O IPTables no Ambiente Cooperativo

Foi visto na seção anterior que o *netfilter* oferece o *framework* para o *iptables*, que através da sua abordagem baseada em cadeias faz com que a filtragem dos pacotes seja realizada de uma maneira mais controlada e mais fácil de ser gerenciada (porém não mais fácil de ser desenvolvida), resultando também em maior desempenho (não é necessário que todas as regras sejam examinadas seqüencialmente, apenas as regras de cada cadeia).

Essa abordagem pode ser aproveitada em ambientes cooperativos, onde uma cadeia pode ser criada para cada elemento do ambiente, de modo que as regras de filtragem podem ser desenvolvidas separadamente para cada elemento, ao invés de se criar um único conjunto imenso de regras para o ambiente em sua totalidade. Essa abordagem resulta assim em um melhor gerenciamento e maior desempenho do *firewall*, diminuindo desta forma as chances de que erros sejam cometidos no desenvolvimento e implementação das regras. A figura 12.7 mostra o *iptables* sendo utilizado no ambiente cooperativo, com as diferentes cadeias para cada organização que faz parte do ambiente.

No ambiente cooperativo, a cadeia FORWARD trabalha para realizar as filtragens mais genéricas, aplicáveis a todos os elementos do ambiente cooperativo, e principalmente, para direcionar a filtragem para a cadeia correspondente a cada organização. A cadeia INPUT deve conter regras para a defesa do próprio *firewall*. A grande questão aqui é a definição do evento que servirá de base para o direcionamento para a cadeia correspondente.

Pode-se supor que utilizar o endereço IP do usuário como base de decisão pode ser perigoso, pois ele pode ser facilmente falsificado, de modo a driblar um conjunto de regras mais restritivo. Porém, essa suposição pode ser considerada incorreta, pois as regras existentes no *firewall* são as mesmas que existiriam no conjunto único de regras (também baseadas em endereços IPs),

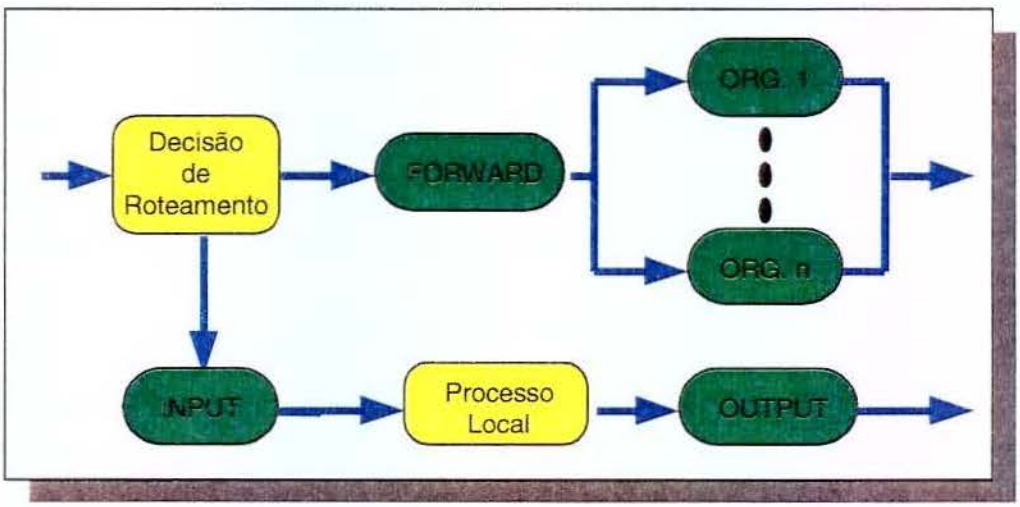


Figura 12.7: O *iptables* no ambiente cooperativo.

apenas com a diferença deles estarem divididas por organização, não sendo necessário que todas as regras de todas as organizações sejam verificadas. Como o objetivo é facilitar o gerenciamento e maximizar o desempenho do *firewall*, o *iptables* parece atender plenamente às necessidades.

Um ponto importante a ser considerado é que, em um ambiente cooperativo, as regras de filtragem são apenas um dos elementos do *firewall* cooperativo, ou seja, a segurança da organização não pode depender apenas dessas regras de filtragem, mas também de um bom mecanismo de autenticação (capítulo 10), preferencialmente baseado em certificados digitais (seção 8.5) e sistemas de detecção de intrusões (capítulo 7) . O *firewall* cooperativo, que será visto na seção a seguir, e o modo em que o *iptables* trata as regras de filtragem (seção 12.3), servem de base para o modelo hierárquico a ser apresentado na seção 12.5.

12.4 Integrando Tecnologias - Firewall Cooperativo

Como foi visto nos capítulo 6 e 11, a evolução das necessidades de segurança indica que diversos outros elementos, além do *firewall*, estão se tornando imprescindíveis para a proteção dos recursos da organização. Foi visto também que diversas arquiteturas de *firewall* já foram propostas, mas que ainda não contemplam as outras tecnologias de segurança, como por exemplo, a VPN,

o IDS e a PKI. O *firewall* cooperativo tem como objetivo apresentar uma arquitetura que inclui essas tecnologias de segurança que não foram inseridas nas arquiteturas tradicionais conhecidas.

Assim, como seria constituído e como ficaria o "muro" das organizações em um ambiente cooperativo (figura 12.8)?

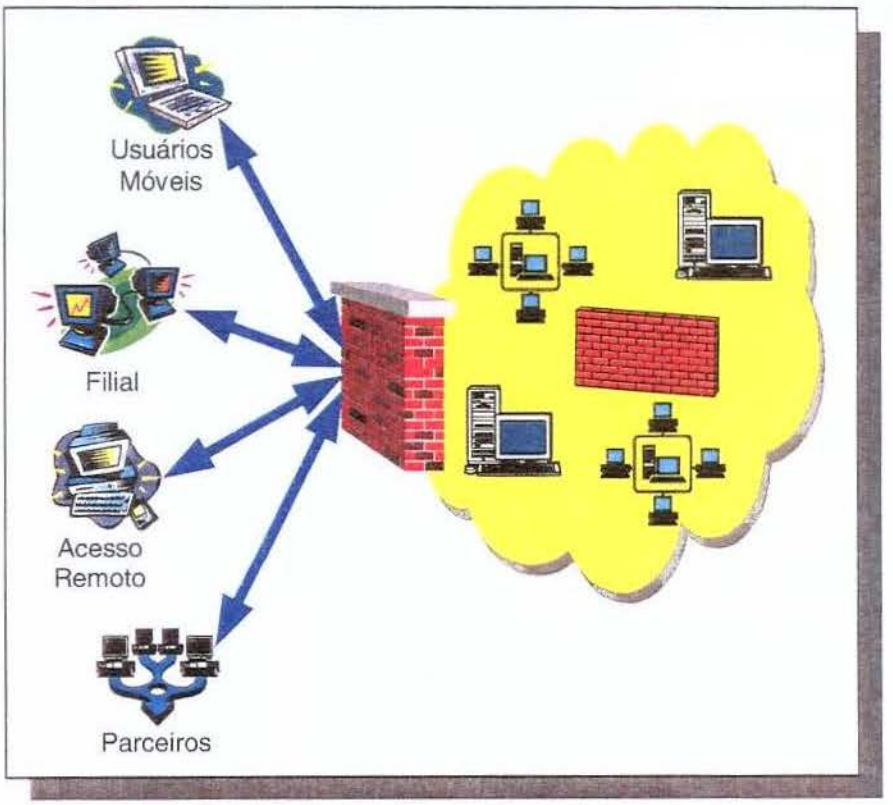


Figura 12.8: O "muro" em um ambiente cooperativo.

Integrar conceitos e tecnologias como *firewall*, DMZ, VPN, PKI, SSO, IPSec, IDS, NAT e criptografia de dados, cada um com a sua função específica, é uma tarefa que requer um planejamento profundo, de acordo com as necessidades e os recursos financeiros da organização. A função de cada um desses componentes e o seu posicionamento dentro da arquitetura influi diretamente no resultado final esperado.

Já pôde ser observado no capítulo 11 a evolução que pode ocorrer numa rede, que através da necessidade de prover recursos para os diversos tipos de usuários, integra conceitos, técnicas e tecnologias, até chegar à arquitetura do *firewall* cooperativo (figura 12.9).

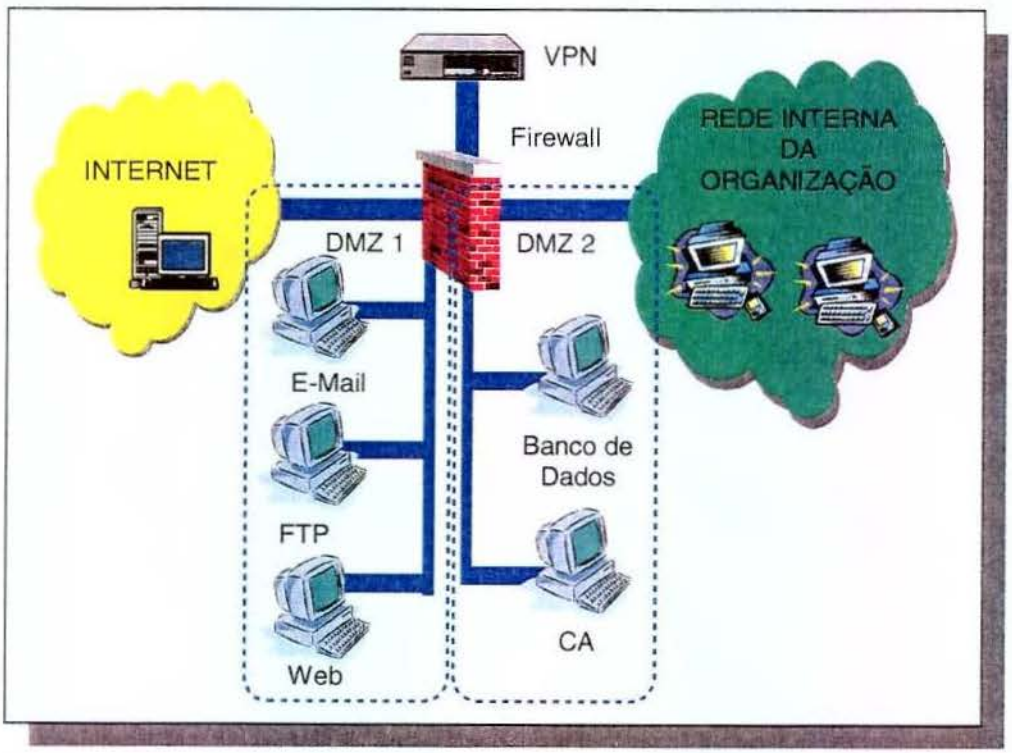


Figura 12.9: A arquitetura do *firewall* cooperativo.

O *firewall* cooperativo tem como objetivo tornar mais claro a administração da segurança do ambiente cooperativo, ao integrar tecnologias específicas para a proteção do ambiente. A grande dificuldade existente é a de inserir cada um dos conceitos e tecnologias dentro do contexto da organização, e o *firewall* cooperativo ajuda nessa tarefa. Porém, se por um lado a arquitetura do *firewall* cooperativo auxilia na definição da estratégia de defesa da organização, os administradores não devem deixar de lado a importância da compreensão das funções de cada um desses elementos. A VPN deve atuar em conjunto com a CA, para que trabalhem juntos na autenticação dos usuários que podem acessar recursos da rede interna, bem como para garantir a confidencialidade das informações que são trocadas com outros elementos do ambiente cooperativo.

Em um paralelo com os bolsões de segurança definidos no capítulo 5, o *firewall* cooperativo sugere que a VPN e a PKI sejam utilizados para a autenticação dos usuários que acessam bolsões maiores da organização (mais recursos internos). O *firewall* cooperativo mostra também que recursos disponibilizados para os acessos exclusivamente através da Internet devem ser disponibilizados na DMZ. Uma segunda DMZ deve ser utilizada para os recursos que necessitam de um maior grau de segurança, e que não possuem acessos diretos dos usuários vindos da Internet. Esse é o caso do banco de dados, que são acessados pelos usuários através do servidor Web que está localizado na primeira DMZ, ou seja, o usuáio acessa o servidor Web, que acessa o banco de dados.

O *firewall* cooperativo trata também dos recursos localizados na rede interna da organização que devem estar sob monitoramento constante, o que é realizado pelos IDS. Esse monitoramento deve coibir ações maliciosas, tanto dos usuários que acessam os recursos via VPN, quanto dos usuários que estão fisicamente dentro da rede interna da organização.

Assim, o *firewall* cooperativo faz a integração de diversos conceitos e tecnologias de segurança, e acaba fazendo uma divisão em 3 partes das localizações dos recursos:

- Recursos públicos disponibilizados para o acesso via Internet – Localização na DMZ;
- Recursos privados disponibilizados para o acesso via Internet – Localização na DMZ 2;
- Recursos internos acessados via VPN – Localização na rede interna.

As proteções referentes a cada um dos tipos de recursos são melhores compreendidas quando uma divisão em diversos níveis de defesa é realizada, como será vista na seção a seguir.

12.5 Níveis Hierárquicos de Defesa

Além do *firewall* cooperativo e do modo em que o *iptables* trabalha com as regras de filtragem, um modelo baseado em diferentes níveis hierárquicos de defesa também é proposto para facilitar a compreensão do esquema de segurança, de modo a facilitar o desenvolvimento, a implementação e o gerenciamento de todas as conexões em um ambiente cooperativo. Essa divisão auxilia também na definição das proteções necessárias para os 3 tipos de recursos identificados na seção anterior (12.4).

Cinco níveis hierárquicos são propostos, de acordo com os conceitos e as tecnologias apresentados durante o trabalho, e que fazem parte de um ambiente cooperativo. O *firewall* cooperativo, apresentado na seção anterior, é uma arquitetura de segurança que, em conjunto com os 5 níveis hierárquicos propostos, ajuda a facilitar a implementação das medidas de segurança necessárias.

Os níveis hierárquicos de defesa compreendem as regras de filtragem, e também as autenticações que devem ser realizadas para o acesso aos recursos da organização. Os 5 níveis formam uma barreira gradual, onde o nível inferior representa uma barreira contra ataques mais genéricos. A filtragem atinge uma maior granularidade de acordo com o aumento hierárquico do nível de defesa, que vai cada vez mais se posicionando em direção à rede interna da organização. Isso pode ser observado na figura 12.10.

Figura 12.10: A granularidade dos níveis hierárquicos de defesa.

Na figura 12.11 pode-se observar as ações que são tomadas em cada um desses níveis hierárquicos. As filtragens são realizadas nos níveis 1, 3 e 5, sendo que os níveis 2 e 4 são referentes à autenticações dos usuários.

Pode-se notar aqui que o 2° nível hierárquico é onde os recursos externos considerados públicos são acessados, ou seja, acessos públicos passam apenas por esse nível hierárquico, além de passar também pelo primeiro nível hierárquico de defesa, que é inerente a todos os tipos de

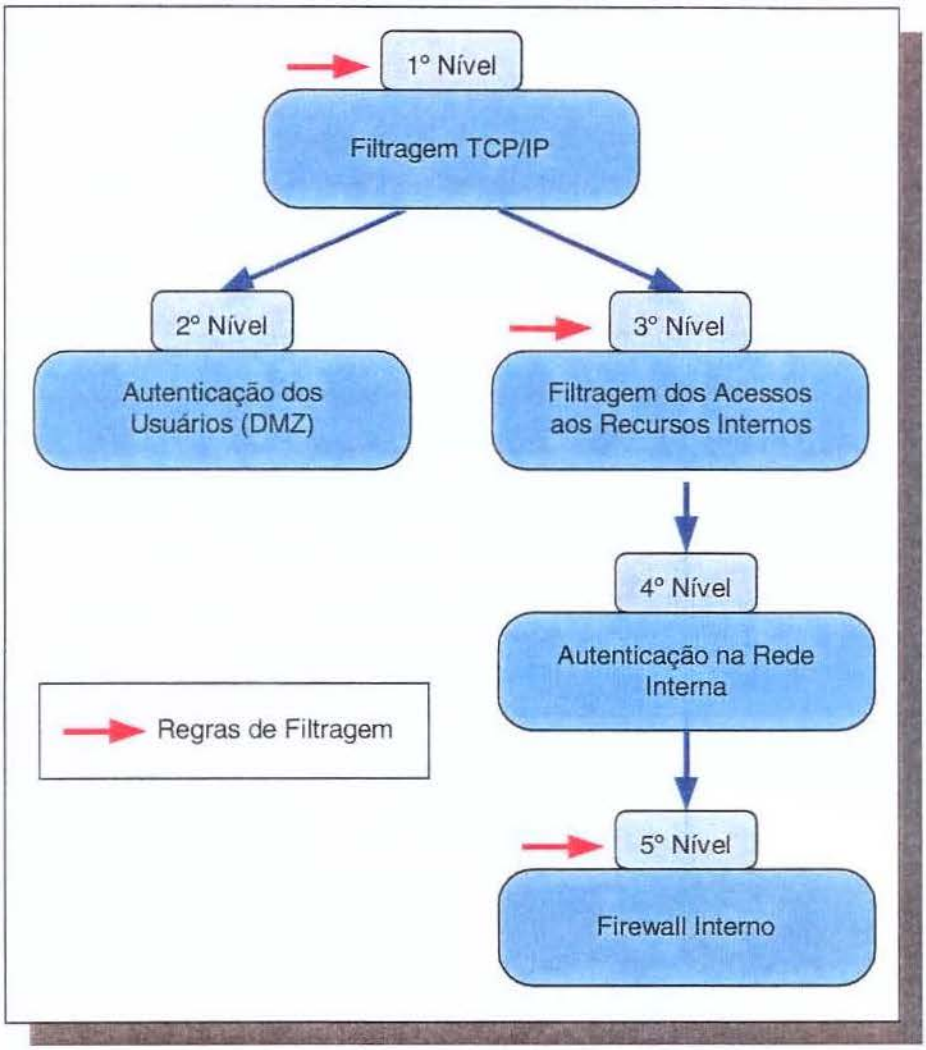


Figura 12.11: As ações em cada nível hierárquico de defesa.

conexões. Os acessos aos recursos internos têm que passar necessariamente pelos níveis hierárquicos 1, 3, 4 e 5. Essa diferenciação faz com que confusões sejam minimizadas, como a dúvida sobre onde e em que ordem definir e implementar as regras de filtragem. Nas seções a seguir estão descritas as ações realizadas em cada um dos níveis de defesa.

12.5.1 1° Nível Hierárquico de Defesa

Essa primeira linha de defesa é a responsável pela filtragem dos pacotes TCP/IP antes deles serem encaminhados para os níveis hierárquicos 2 e 3. Todos os tipos de conexões passam necessariamente por essa linha de defesa, sendo permitidos somente os pacotes referentes a serviços públicos disponíveis na DMZ e a canais de respostas dos serviços disponíveis para os usuários internos, bem como pacotes IPSec dos túneis VPN. Um ponto a ser considerado é que o "muro" das figuras que serão vistas a seguir podem ser consituidas por filtros de pacotes/estados e também pelos *proxies*.

Esse primeiro nível hierárquico é destinado a descartar pacotes de serviços que não são permitidos e não são providos, minimizando assim as implicações de segurança e otimizando a utilização da largura de banda da rede da organização. É o nível responsável também pela proteção contra ataques ao protocolo TCP/IP, tais como *o IP Spoofing* e *SYN Flooding*. A proteção contra o *Smurf* também é realizada neste nível.

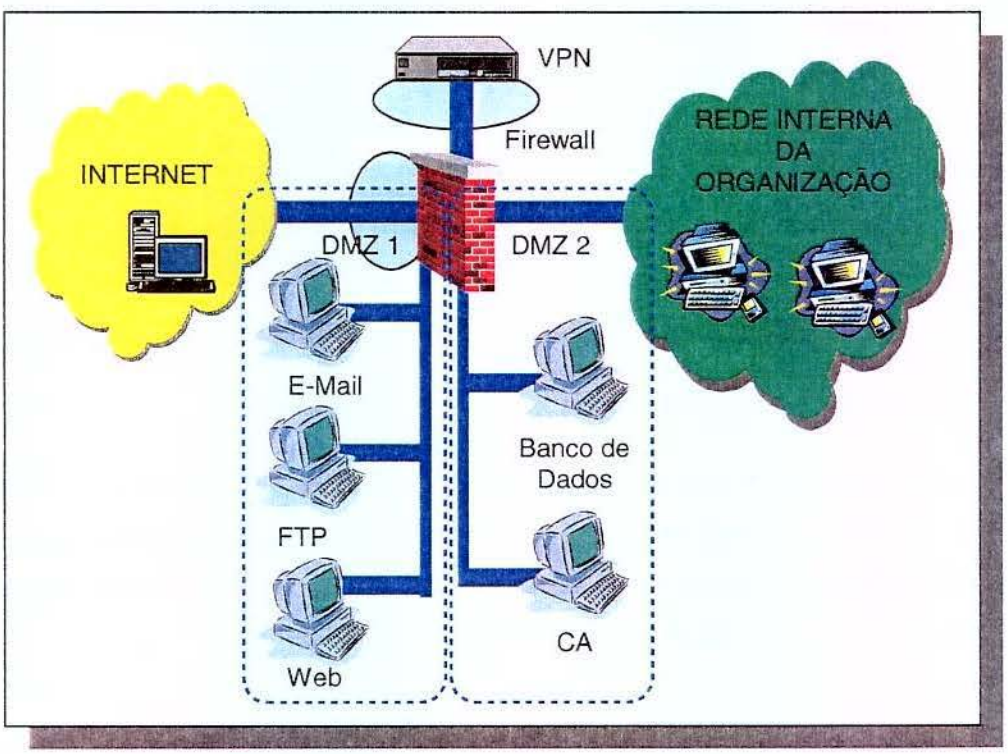


Figura 12.12: O 1° nível hierárquico de defesa.

12.5.2 2° Nível Hierárquico de Defesa

O 2° nível hierárquico de defesa é o responsável pela autenticação dos usuários que acessam os serviços públicos localizados na DMZ. Não existem regras de filtragem nesse nível. A utilização da autenticação não é sempre mandatória, como são os casos de serviços públicos como a Web e o FTP anônimo. Alguns dos serviços que necessitam desse segundo nível de defesa são o FTP não anônimo e o acesso Web a informações particulares, que requerem uma autenticação do usuário para que o acesso possa ser liberado. Esse servidor Web pode se comunicar com o banco de dados residente na segunda DMZ, onde as informações estão localizadas efetivamente. As regras de filtragem que controlam esse acesso entre o servidor Web e o banco de dados são definidas pelo 3° nível de defesa.

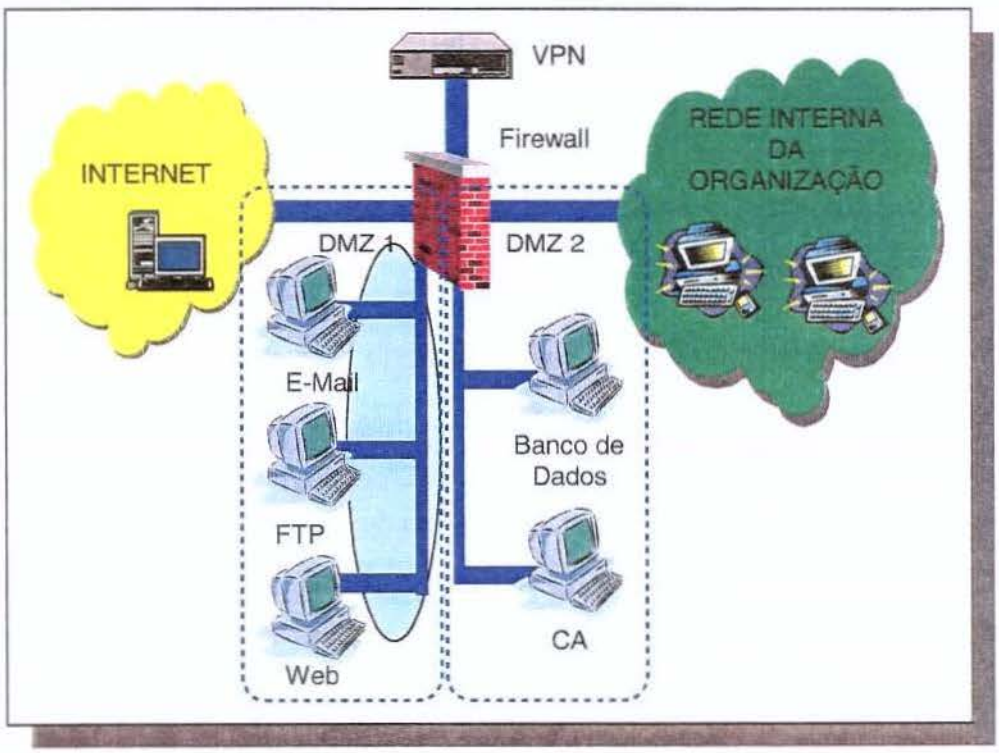


Figura 12.13: O 2° nível hierárquico de defesa.

12.5.3 3° Nível Hierárquico de Defesa

A complexidade das regras de filtragem reside no 3° nível hierárquico de defesa. É através do conjunto de regras desse nível que os usuários terão acessos apenas às informações e serviços pertinentes a ele. Esse nível cuida da porta de entrada da rede interna da organização, e pode ser dividido em 2 partes: as regras de filtragem para a segunda DMZ, e as regras para o acesso à rede interna da organização.

As regras de filtragem para a segunda DMZ são definidas de modo que, nesse exemplo, o banco de dados possa ser acessado somente pelo servidor Web da primeira DMZ, e a entidade certificadora (CA) possa ser acessada apenas pelo dispositivo VPN. Isso faz com que as informações e os recursos importantes localizados nessa segunda DMZ sejam bem protegidos, e ao mesmo tempo, acessíveis para os usuários externos.

As regras para o acesso à rede interna devem ser definidas de modo que somente os usuários autenticados passem por esse nível de defesa, bem como garantir que esses usuários autenticados acessem somente os recursos a que são explicitamente permitidos. Através disso, pode-se perceber que grande parte da complexidade da segurança reside neste nível, e uma abordagem como a que é utilizada pelo *iptables* (seção 12.3.3), é importante. Relembrando, a abordagem se refere à divisão das regras de filtragem em diversas cadeias, cada uma correspondente a uma organização integrante do ambiente cooperativo.

12.5.4 4° Nível Hierárquico de Defesa

A 4ª linha de defesa é a referente à autenticação dos usuários para o acesso aos serviços e informações internas da organização. É a autenticação realizada como se os usuários estivessem fisicamente na organização. A partir desse nível hierárquico, portanto, a segurança se baseia na proteção existente internamente na organização, como as senhas para acesso a sistemas internos. Nenhuma filtragem é realizada nesse nível, e toda a autenticação pode ser realizada em conjunto com a entidade certificadora, de modo que os serviços sejam acessados de acordo com os certificados digitais de cada usuário.

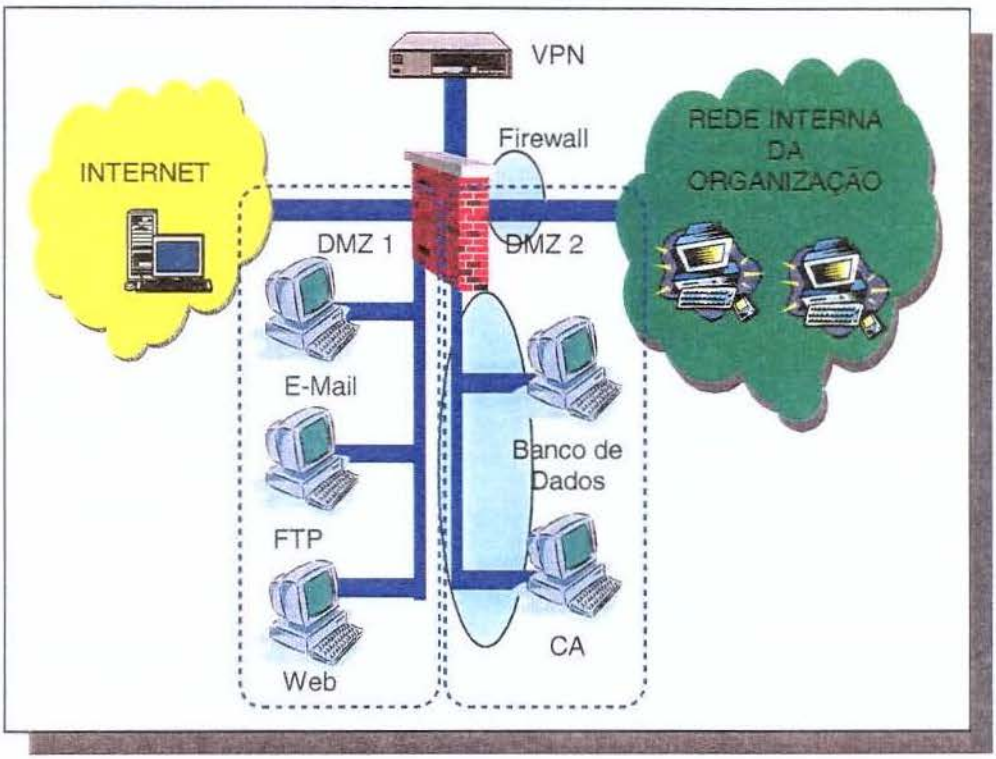


Figura 12.14: O 3° nível hierárquico de defesa.

12.5.5 5° Nível Hierárquico de Defesa

O 5° nível hierárquico de defesa se refere aos acessos e ao controle dos usuários que estão acessando a rede interna da organização. Esse nível pode ser considerado como sendo um *firewall* interno, com a adição de sistemas de detecção de instrusões (IDS) para o monitoramento das atividades dos usuários. Um funcionário da seção de manufatura, por exemplo, não pode ter acesso às informações do setor financeiro. Assim o controle é realizado internamente, e pode ser realizado também em conjunto com a entidade certificadora.

12.5.6 Os Níveis Hierárquicos de Defesa na Proteção dos Recursos

De acordo com a divisão dos tipos de recursos vista na seção 12.4, os seguintes níveis hierárquicos de defesa podem ser utilizados para a proteção dos 3 tipos de recursos definidos (públicos, privados e internos):

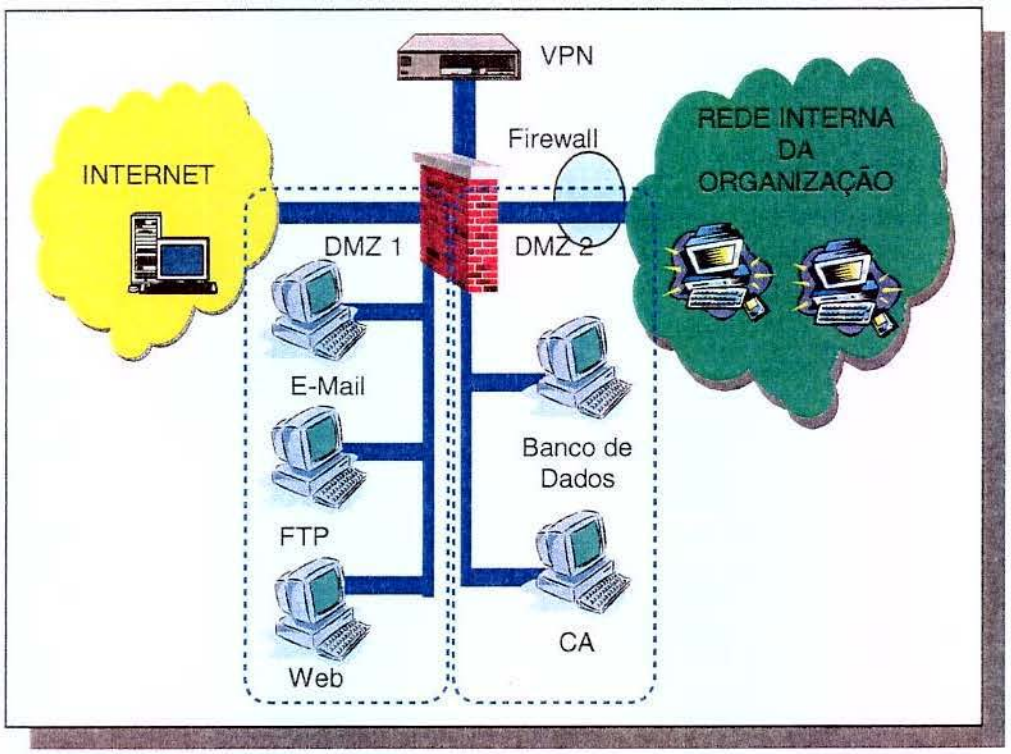


Figura 12.15: O 4° nível hierárquico de defesa.

- Recursos públicos: 1° nível hierárquico de defesa;
- Recursos privados: 1° e 2° níveis hierárquicos de defesa;
- Recursos internos: 1°, 3°, 4° e 5° níveis hierárquicos de defesa.

Com essa relação entre os recursos a serem protegidos e os níveis hierárquicos de defesa responsáveis pela proteção, vê-se que a definição das medidas de segurança e das regras de filtragem torna-se mais fácil de ser compreendida.

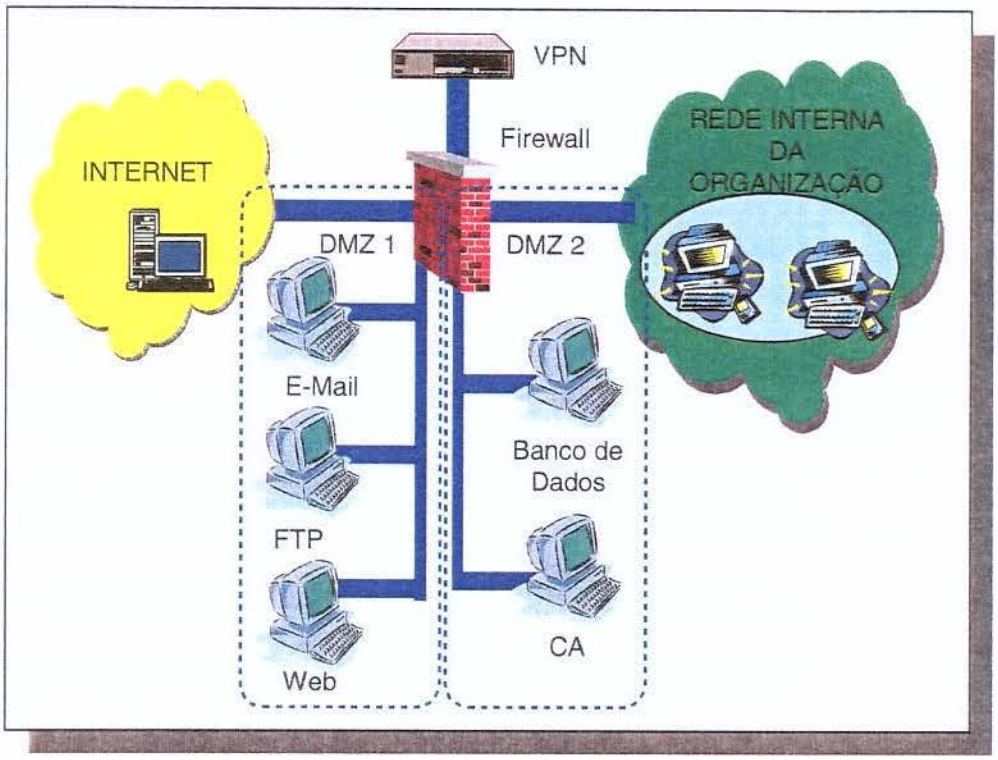


Figura 12.16: O 5° nível hierárquico de defesa.

12.6 Conclusão

Este capítulo apresentou a proposta do modelo de segurança para ambientes cooperativos. Através do exemplo do conjunto de regras de filtragem definido para o LAS-IC-Unicamp, pôde-se verificar que a complexidade que envolve as conexões é bastante alta, tornando a sua definição e implementação passíveis de erros. Três elementos principais formam o modelo de segurança proposto: o *firewall* cooperativo, que é a arquitetura que integra os conceitos e tecnologias de segurança necessários; o modo de manipulação das regras de filtragens utilizado pelo *iptables*, baseado em cadeias; e os 5 níveis hierárquicos de defesa, que facilitam a compreensão das medidas de segurança a serem adotadas.

Através da combinação desses 3 elementos, a definição da estratégia de segurança em um ambiente cooperativo torna-se mais clara e mais compreensível, minimizando-se assim a possibilidade de erros de análise, definição e implementação das medidas de segurança necessárias.

Capítulo 13

Conclusão

A globalização e a evolução no modo em que os negócios são realizados, principalmente com o advento da Internet, trazem uma série de conseqüências para as organizações. As conexões entre elas atingem um alto grau de complexidade, onde diversos problemas têm que ser resolvidos. O ambiente cooperativo, que é formado pelas conexões entre organizações e filiais, parceiros comerciais e usuários móveis, serviu de cenário para demonstrar as dificuldades existentes.

Grande parte da dificuldade reside no fato dos acessos estarem cada vez profundos, ou seja, entrando cada vez mais nos domínios da organização. Isso faz com que a abordagem utilizada pelo *firewall*, de criar um grande muro para separar a rede interna da rede pública, não seja totalmente válida em ambientes cooperativos. Um modelo baseado em bolsões de segurança, fazendo um diferir do outro de acordo com o nível de acesso, faz com que os níveis de acesso possam ser visualizados de uma maneira mais clara. Assim, uma filial, por exemplo, possui acesso a bolsões de segurança maiores do que os de fornecedores. Esse modelo demonstra claramente que a segurança de borda é necessária, porém não é mais suficiente, sendo imprescindível uma segurança no nível interno da organização. De fato, os bolsões de segurança entram cada vez mais nos domínios internos da organização, de acordo com os níveis de acesso necessários.

Com isso, um modelo de segurança que contempla a segurança interna torna-se necessária. A política de segurança, ponto primordial na definição das proteções efetivas da organização, ganha uma importância ainda maior, ao ter que tratar de pontos bastante complexos. Além do *firewall*, responsável pela filtragem e controle de acesso, um sistema de detecção de intrusões

(IDS) para a detecção de indícios de ataques, um bom sistema de autenticação para a identificação dos usuários, e redes privadas virtuais (VPN) para a integridade e confidencialidade das conexões, torna-se importante. Incluindo ainda a infra-estrutura de chaves públicas (PKI), pode-se verificar que a integração entre as diferentes tecnologias torna-se uma tarefa difícil para o administrador de segurança. O *firewall* cooperativo é uma arquitetura proposta para auxiliar essa integração entre as diversas tecnologias.

As regras de filtragem do *firewall* têm o seu nível de complexidade multiplicado pelo número de conexões, exigindo uma metodologia que minimize as chances de erros na sua definição e implementação. Um dos sistemas de filtragem que tentam resolver esses problemas é o *Netfilter*. Através da utilização de cadeias e ganchos, ele acaba abordando também o problema de desempenho no processamento de um conjunto longo de regras.

A arquitetura do *firewall* cooperativo, somado ao modo de funcionamento do *Netfilter*, tem como resultado o auxílio na definição, na implementação e no gerenciamento da segurança de uma organização, principalmente em um ambiente cooperativo. Somado a isso, um modelo hierárquico de defesa, aplicado sobre o *firewall* cooperativo, visa facilitar a compreensão da complexidade envolvida. 5 níveis hierárquicos de defesa tratam de questões relativas às regras de filtragem e à autenticação, e diferem entre si quanto aos seus objetivos.

Assim, foi visto que o surgimento de um ambiente cooperativo traz junto novas necessidades de segurança. Diversos conceitos e tecnologias de segurança devem ser utilizados em conjunto, para a proteção desse ambiente. Eles foram discutidos, e o *firewall* cooperativo foi apresentado para que eles possam ser utilizados de um modo integrado. Foi visto também que os diferentes níveis de acesso resultam em uma grande complexidade na criação do conjunto de regras de filtragem. Para minimizar os possíveis erros na definição e na implementação dessas regras, o modo de funcionamento do *Netfilter* foi discutido. Somado a isso, níveis hierárquicos de defesa foram definidos para auxiliar na definição das regras e na autenticação, quando o *firewall* cooperativo é implementado.

É necessário contudo salientar que tipicamente a complexidade do ambiente é bastante grande, de modo que o modelo apresentado pode se tornar difícil de ser seguido. Isso pode ocorrer principalmente quando novos serviços, que não foram considerados neste trabalho, são inseridos na arquitetura. Ainda, uma limitação pode ser vista no modelo, no momento da implementação das regras de filtragem, de acordo com a abordagem adotada pelo *Netfilter*. Foi visto

que o *Netfilter* faz parte do *kernel* do Linux, porém uma solução de segurança não deve se ater somente a um sistema operacional específico. O ideal é seguir o modo como o *Netfilter* interpreta as regras de filtragem. Porém, o que pode ser visto é que somente ele utiliza essa abordagem[1], que facilita a compreensão das regras e melhora o desempenho da filtragem. Assim, uma sugestão para trabalhos futuros é o desenvolvimento de um método de filtragem que permita a criação de regras de um modo estruturado. Essa estruturação pode ser vista no *Netfilter* e também nas linguagens de programação estruturadas, como o C ou o Pascal. Isto sugere a adoção de técnicas avançadas de administração de segurança, que fica como uma outra sugestão para um trabalho futuro.

Outra importante sugestão para pesquisa futura é um melhor estudo do acesso remoto. Neste trabalho deixou-se de focar o acesso remoto por ele abrir um amplo campo de complexidade da segurança envolvida, necessitando de medidas específicas. Aspectos como a segurança física, e principalmente, a política de segurança, devem ser consideradas com extremo cuidado no acesso remoto. A segurança física é importante, por exemplo, em *notebooks*, e a política de segurança é difícil de ser seguida em equipamentos que ficam fora dos domínios da organização. A sugestão para o futuro é ainda mais válida quando pode ser verificado que o acesso remoto é cada vez mais utilizado dentro das organizações, tendo um papel ainda maior dentro de um ambiente cooperativo.

Diversos outros aspectos necessitam ser trabalhados no futuro, como as conexões sem fio (*wireless*), cada vez mais utilizadas dentro das organizações. Um outro aspecto é a adaptação do modelo apresentado em redes baseadas em IPv6.

A segurança é complexa, e a definição de sua estratégia, a sua implementação e o seu gerenciamento formam um processo constante. Conceitos, modelos e metodologias que auxiliam nessa tarefa são importantes para que o objetivo de proteção seja alcançado. Este trabalho permitiu apresentar este processo de maneira global, auxiliando o administrador de segurança a implementar defesas eficazes e eficientes.

---

1. IPFilter (filtro mais usado em plataformas BSD) também pode indiretamente prover uma estruturação nas regras.

Bibliografia

[ABE 97] ABELSON, Hal; ANDERSON, Ross; BELLOVIN, Steven M.; BENALOH, Josh; BLAZE, Matt; DIFFIE, Whitfield; GILMORE, John; NEUMANN, Peter G.; RIVEST, Ronald L.; SHILLER, Jeffrey I.; SCHNEIER, Bruce. *The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption.* Final Report: May 27, 1997.

[ALE 98] ALEXANDER, Steve. *Going Above and Beyond the Firewall.* July 1998. 14/01/99. http://www.computerworld.com/home/features.nsf/All/980727intra_proj

[ALE 99] ALEXANDER, Steve. Star Tribune. *Modem Security Breach Lets Neighbor learn a Little too Much.* July 10, 1999. 13/07/99. http://www.startribune.com/stOnLine/cgi-bin/article?thisSlug=cabl10

[ARK 99] ARKIN, Ofir. *Network Scanning Techniques – Understanding How It Is Done.* Publicom Communications Solutions. November, 1999.

[ARS 99] ARSENAULT A.; TURNER, S. PKIX Working Group. Internet Draft. *Internet X.509 Public Key Infrastructure – PKIX Roadmap.* October 22, 1999. http://www.pca.dfn.de/eng/team/kelm/drafts/draft-ietf-pkix-roadmap-04.txt

[AVO 94] AVOLIO, F. M.; RANUM, M. J. *A Network Perimeter With Secure External Access.* Glenwood, MD: Trusted Information Systems, Incorporated, January, 1994.

[AVO 98] AVOLIO; BLASK. *Application Gateways and Stateful Inspection: A Brief Note Comparing and Contrasting.* January 22, 1998. 16/04/99. http://www.avolio.com/apgw+spf.html.

[AVO 99] AVOLIO, Frederick M. Information Security. Cover Story. *Firewalls: Are We Asking Too Much?* May 1999. 16/06/99. http://www.infosecuritymag.com/may99/cover.htm

[BAR 99] BARRETT, Randy. ZDNet Tech News. *Major Unix Flaw Emerges.* March 1, 1999. 10/03/99. http://www.zdnet.com/zdnn/stories/news/0,4586,2217922,00.html

[BAY 98] Bay Networks. *Understanding and Implementing Virtual Private Networking (VPN) Services.* http://www.baynetworks.com/products/Papers/2746.html. 29/01/99.

[BEL 97] BELLOVIN, Steven M. *Probable Plaintext Cryptanalysis of the IP Security Protocols.* AT&T Labs Research. Florham Park, NJ, USA: 1997.

[BEL 97-2] BELLOVIN, Steven M. *Probable Plaintext Cryptanalysis of the IP Security Protocols.* AT&T Labs Research. Florham Park, NJ, USA: 1997.

[BEL 98] BELLOVIN, Steven M. *Cryptography and the Internet.* CRYPTO '98. AT&T Labs Research. Florham Park, NJ, USA: August 1998.

[BHI 98] BHIMANI, Anish. Information Security. *All Eyes on PKI.* 25/01/99. October 1998. http://www.infosecuritymag.com/oct/pki.htm

[BIT 98] BITAN, Sarah. Chief Technology Officer. RADGUARD. *Hardware Implementation of IPSec: Performance Implications.* 18/09/98. http://www.radguard.com/VPN_hardware_IPSec.html.

[BLA 96] BLAZE, Matt; DIFFIE, Whitfield; RIVEST, Ronald L.; SCHNEIER, Bruce; SHIMOMURA, Tsutomu; THOMPSON, Eric; WIENER, Michael. *Minimal Key Lengths For Symmetric Ciphers To Provide Adequate Commercial Security.* A Report By An Ad Hoc Group Of Cryptographers And Computer Scientists. January 1996. 24/02/99. http://www.bsa.org/policy/encryption/cryptographers_c.html.

[BRA 97] BRANCHAUD Marc. *A Survey of Public-Key Infrastructures.* Master of Science in Computer Science Thesis. McGill University, Montreal: 1997, Department of Computer Science.

[BRE 98] BREZINSKI, Dominique; KAPLAN, Ray. Information Security. *(R)evolutionary IDS.* November 1998. 25/01/99. http://www.inforsecuritymag.com/nov/ids.htm

[BRE 99] BREED, Charles. Infosecurity Magazine. *PKI: The Myth, the Magic and the Reality.* June, 1999. 16/06/99. http://www.infosecuritymag.com/jun99/PKI.htm

[BRI 98] BRINEY, Andy. Information Security. *1998 Annual Industry Survey.* June 1998. 25/01/99. http://www.inforsecuritymag.com/industry.htm

[BRI 99] BRINEY, Andy. Information Security. *Inforsecurity: A View From the Frontlines.* Febryary 1999. 25/02/99. http://www.inforsecuritymag.com/feb99/rndtable.htm

[BRI 99B] BRINEY, Andy. Information Security. *'99 Survey.* July 1999. 07/07/99. http://www.inforsecuritymag.com/july99/cover.htm

[BRI 99C] BRINEY, Andy. Information Security. *Secure Remote Access. Remote Security: Sink or Swim?*. July 1999. 07/07/99. http://www.inforsecuritymag.com/july99/secure_remote.htm

[BRU 98] BRUSSIN, David. Information Security. *All for One, and One for All.* June 1998. 25/01/99. http://www.inforsecuritymag.com/all-for-one.htm

[CAM 98] CAMPBELL, Robert P. Information Security. *Planning for Success; Preparing for Failure.* July 1998. 25/01/99. http://www.inforsecuritymag.com/campb.htm

[CAR 99] CARDEN, Phillip. Network Computing. *The New Face of Single Sign-On.* March 22, 1999. 24/03/99. http://www.techweb.com/se/directlink.cgi?NWC19990322S0013

[CAR 99-2] CARDEN, Phillip. Network Computing. *Border Control: Na Antivirus Gateway Guide.* May 31, 1999. 01/06/99. http://www.techweb.com/se/directlink.cgi?NWC19990531S0026.

[CER 99-1] CERT Coordination Center. CERT Incident Note IN-99-07. *Distributed Denial of Service Tools.* November 18, 1999. 30/01/2000. http://www.cert.org/incident_notes/IN-99-07.html.

[CER 99-2] CERT Coordination Center. *CERT Advisory CA-99-17 Denial-of-Service Tools.* December 28, 1999. 30/01/2000. http://www.cert.org/incident_notes/IN-99-07.html.

[CER 99-3] CERT Coordination Center. *Results of the Distributed-Systems Intruder Tools Workshop.* Pittsburgh, Pennsylvania USA. November 2-4, 1999.

[CHA 95] CHAPMAN, D. Brent. AWICHY, Elizabeth D. *Building Internet Firewalls.* O'Reilly & Associates, Inc. 1995.

[CHE 94] CHESWICK, William R.; BELLOVIN, Steven M. *Repelling the Wily Hacker. Addison-Wesley. 1994, April.*

[CHE 97] Check Point Software Technologies Ltd. *Privacy in Public Networks Using Check Point FireWall-1.* January 1997.

[CHE 98] Check Point Software Technologies Ltd. *Stateful Inspection Firewall Technology.* 08/03/99. http://www.checkpoint.com/products/technology/stateful1.html

[CHE 98-2] Check Point Software Technologies Ltd. *Virtual Private Network – Security Components – A Technical White Paper.* March 23, 1998.

[CHE 98-3] Check Point Software Technologies Ltd. http://www.checkpoint.com. 04/01/99.

[CHE 98-4] Check Point Software Technologies Ltd. *Redefining the Virtual Private Network.* March 4, 1998.

[CHE 98-5] Check Point Software Technologies Ltd. *Virtual Private Network – Security Components – A Technical White Paper.* March 23, 1998.

[CHR 99] CHRISTENSEN, John. CNN Interactive. *Bracing for Guerrila Warfare in Cyberspace.* March 29, 1999. 06/04/99. http://www.cnn.com/TECH/specials/hackers/cyberterror

[CIA 98-19] CIAC. Computer Incident Advisory Capability. U.S. Department of Energy. *I-031a: Malformed UDP Packets in Denial of Service Attacks.* March 6, 1998. http://ciac.llnl.gov/ciac/bulletins/i-031a.shtml. 29/12/99.

[CIA 98-31] CIAC. Computer Incident Advisory Capability. U.S. Department of Energy. *I-019: Tools Generating IP Denial-of-Service Attacks.* December 19, 1997. http://ciac.llnl.gov/ciac/bulletins/i-019.shtml. 29/12/99.

[CIS 96] Cysco Systems Inc. *Defining Strategies to Protect Against TCP SYN Denial of Service Attacks.* 1996. 08/04/99. http://cio.cisco.com/warp/public/707/4.html

[CIS 98] Cisco Systems, Inc. *Field Notice: Cisco PIX and CBAC Fragmentation Attack.* September 11, 1998. http://www.cisco.com/warp/public/770/nifrag.shtml. 29/12/99.

[CIS 98-2] Cisco Systems, Inc. *Building a Perimeter Security Solution with the Cisco IOS Firewall Feature Set.* 1998.

[CIS 99] Cysco Systems Inc. *Defining Strategies to Protect Against UDP Diagnostic Port Denial of Service Attacks.* 1999. 08/04/99. http://cio.cisco.com/warp/public/707/3.html

[CIS 01] Cisco Systems, Inc. *Increasing Security on IP Networks.*

[COA 00] COAST. *Intrusion Detection Systems.* http://www.cerias.purdue.edu/coast/intrusion-detection/ids.html. 02/02/00.

[COH 99] COHEN, Frederick B. *TCP Packet Fragment Attacks Against Firewalls and Filters.* http://packetstorm.securify.com/docs/hack/frag.txt. 29/12/99;

[COM 95] COMER, Douglas E. Department of Computer Sciences, Purdue University, West Lafayette, IN 47907. *Internetworking With TCP/IP Vol I: Principles, Protocols, and Architecture.* 3. Edition. 1995: Prentice-Hall, Inc.

[COP 99] COPELAND John A. *Macintosh DoS Flood Attack.* December 29, 1999. Georgia Institute of Technology. http://www.csc.gatech.edu/~copeland/macattack/index.htm.

[DAV 97] DAVIS, Don. *Compliance Defects in Public-Key Cryptography.* March 10, 1997.

[DEJ 99] DEJESUS, Edmund. Infosecurity Magazine. *No Anxiety at the ANX.* April, 1999. 16/04/99. http://www.infosecuritymag.com/apr99/ANX%20SIDEBAR.htm

[DEN 99] DENNING, Dorothy E. Infosecurity Magazine. *Who's Stealing Your Information?.* April, 1999. 16/04/99. http://www.infosecuritymag.com/apr99/cover.htm

[DID 98] DIDIO, Laura. ComputerWorld. *Halt, Hackers !.* July 1998. 14/01/99. http://www.computerworld.com/home/features.nsf/all/980727intra_main

[DID 98-2] DIDIO, Laura. ComputerWorld. *From Bad to Worse.* July 1998. 14/01/99. http://www.computerworld.com/home/features.nsf/All/980727intra_side2

[DIT 99-01]DITTRICH, David. *The DoS Project's "trinoo" Distributed Denial of Service Attack Tool.* University of Washington. October 21, 1999. 02/02/2000. http://staff.washington.edu/dittrich/misc/trinoo.analysis.

[DIT 99-02]DITTRICH, David. *The "Tribe Flood Network" Distributed Denial of Service Attack Tool.* University of Washington. October 21, 1999. 02/02/2000. http://staff.washington.edu/dittrich/misc/tfn.analysis.

[DIT 99-03]DITTRICH, David. *The "Stacheldraht" Distributed Denial of Service Attack Tool.* University of Washington. December 31, 1999. 02/02/2000. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.

[DUV 98] DUVAL, Mel. Interactive Week Online. *Group Working On VPN Product Standard.* October 5, 1998. http://www.zdnet.com/intweek/stories/news/0,4164,2145301,00.html. 29/01/99.

[DYK 98] DYKE, Gary Van. Information Security. *Expect Thunderstorms.* September 1998. 25/01/99. http://www.infosecuritymag.com/sept/edgewise.htm

[ENT 99] Entrust Technologies. *Summary of Protocols for PKI Interoperability.* 1999. 11/03/99. http://www.entrust.com/products/library/protocols_pki.htm

[ENT 00] Enterasys Networks. *IP Security (IPSec).* 11/07/00. http://www.enterasys.com/vpn/VPNipsec.htm

[FEL 97] FELTEN, Edward W.; BALFANZ, Dirk; DEAN, Drew; WALLACH, Dan S. *Web Spoofing: An Internet Con Game.* February 1998. Departiment of Computer Science, Princeton University.

[FER 98] FERGUSON, P. Network Working Group. Request for Comments 2267. *Network Ingress Filtering: Defeating Denial of Service Attacks wich Employ IP Source Address Spoofing.* January 1998. 08/04/99. ftp://ftp.isi.edu/in-notes/rfc2267.txt

[FIST 98] Front-line Information Security Team (FIST). Network Security Solutions Ltd. *Techniques Adopted By ' System Crackers' When Attempting to Break Into Corporate or Sensitive Private Networks'.* December 1998. 15/01/99. http://www.ns2.co.uk/archive/FIST/papers/NSS-cracker.txt

[FIST 99] Front-line Information Security Team (FIST). Network Security Solutions Ltd. *Understanding Concepts In Enterprise Network Security And Risks In Networked Systems - Part 1 of 3: Understanding Riscks In Networked Systems.* January 1999. 15/01/99. http://www.ns2.co.uk/archive/FIST/papers/NSS-risk-pt1.txt

[FOO 98] FOOTE, Steven. Information Security. *19 Infosecurity Predictions For '99.* November 1998. 25/01/99. http://www.inforsecuritymag.com/nov/cover.htm

[FOR 98] Forrester Research Inc. http://www.forrester.com. 04/01/99.

[FYO 97] FYODOR. *The Art of Port Scanning.* Phrack Magazine. Volume 7, Issue 51. September 01, 1997. 29/01/00. http://www.insecure.org/nmap/p51-11.txt

[FYO 98] FYODOR. *Remote OS Detection via TCP/IP Stack FingerPrinting.* October 18, 1998. Last Modified: April 10, 1999. 29/01/00. http://www.insecure.org/nmap/nmap-fingerprinting-article.html

[FYO 99] FYODOR. *Nmap Network Security Scanner Man Page.* 29/01/00. http://www.insecure.org/nmap/nmap_manpage.html.

[GAR 96] GARFINKEL Simson L; SPAFFORD, Gene. *Practical UNIX and Internet Security, Second Edition.* O'Reilly & Associates, Inc. 1996.

[GAR 98] GARFINKEL Simson L. *Advanced Telephone Auditing with PhoneSweep: A Better Alternative to Underground "War Dialers".* 1999. 27/08/99. http://www.mids.org/mn/812/sim.html

[GEU 00] GEUS, Paulo Lício. *Curso de Segurança de Redes.* 2000: Unicamp.

[GOL 98] GOLDSMITH, David; SCHIFFMAN, Michael. Cambridge Technology Partners, Enter-

prise Security Services. *Firewalking – A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists.* 1998.

[GRA 99] GRAHAM, Robert. *Network Intrusion Detection System FAQ.* Version 0.6.1, August 5, 1999. 01/02/00. http://www.shake.net/network-intrusion-detection.htm

[HAL 98] HAL, Ron. Information Security. *Intrusion Crack Down.* August 1998. 25/01/99. http://www.infosecuritymag.com/august/cover.htm

[HER 98] HERSCOVITZ, Eli. President and CEO of RADGUARD Ltd. *Secure Virtual Private Networks: The Future of Data Communications.* 18/09/98. http://www.radguard.com/VPNmrkt.html.

[HIG 98] HIGGINS, Kelly Jackson. Network Computing. *A Moving VPN Target.* June 15, 1998. http://www.techweb.com/se/directlink.cgi?NWC19980615S0013. 29/01/99.

[HTTP 01] http://www.ncs.gov/n5_hp/n5_ia_hp/html/eitr/thra2_1.htm

[HTTP 02] http://www.wired.com/news/news/technology/story/15673.html

[HUE 98] HUEGEN, Craig A. *The Latest in Denial of Service Attacks: "Smurfing" - Description and Information to Minimize Effects.* Dec 30, 1998. 08/04/99. http://users.quadrunner.com/chuegen/smurf.cgi

[HUR 99] HURLEY, Jim. Information Security. *Survival of the Fittest.* January 1999. 25/02/99. http://www.infosecuritymag.com/jan99/cover2.htm

[ICS 98] ICSA Releases. *ICSA Announces First IPSec Certified Products.* 23/09/98. http://www.ncsa.com/news_alerts/press_room/IPSEC2.html.

[INF 98] Infonetics Research Inc. http://www.infonetics.com. 04/01/99.

[INF 99] Information Security. *Down With Hacktivism !.* February 1999. 25/02/99. http://www.inforsecuritymag.com/feb99/underground.htm

[INF 99-2] Info World. Test Center. *Sniffing out Network Holes.* February 8, 1999. 14/04/99. http://www.infoworld.com/cgi-bin/display/TC.pl?/990208comp.htm

[JOH 98] JOHNSON, Anna. Shake Security Journal. *Companies Losing Millions over Rising Computer Crime.* March 1998. 15/01/99. http://www.shake.net/solutions/ssj/march98/crime_march98.htm

[JON 95] JONCHERAY, Laurent. *A Simple Active Attack Against TCP.* Merit Network, Inc.

April 24, 1995.

[KEN 97] KENNEY, Malachi. *Ping of Death*. http://www.insecure.org/sploits/ping-o-death.html. 29/12/99.

[KES 96] KESSLER, Gary C. *Passwords – Strengths and Weaknesses*. January 1996. 11/03/99. http://www.www.hill.com/library/password.html

[KIM 99] KIMBER, Lee. CMP Net. *New Attacks Point Up Web Pages' Vulnerability*. www.techweb.com/wire/story/TWB19991209S0007. December 9, 1999. 02/01/2000.

[KIN 98] KING, Christopher M. Information Security. *Keys to the Kingdom*. 25/01/99. April 1998. http://www.infosecuritymag.com/keys.htm

[KIN 99] KING, Christopher M. Information Security. *The 8 Hurdles to VPN Deployment*. March, 1999. http://www.infosecuritymag.com/mar99/cover.htm.

[KRA 99] KRANE, Jim. Crime Tech. *Computer Crime Tops $100 Million – Hackers Steal Secrets, Wreak Havoc, Report Says*. March, 15 1999. 17/03/99. http://www.apbonline.com/safestreets/1999/03/15/hacker0315_01.html

[KRO 99] KROCHMAL, Mo. TechWeb. *Report Emphasizes Managing Security To Minimize Damage*. April 16, 1999. 20/04/99. http://www.techweb.com/wire/story/TWB19990416S0003

[L0P 99] L0pht Security Advisory. *Any Local User Can Gain Administrator Privileges and/or Take Full Control over the System*. February 18, 1999. 26/02/99. http://www.l0pht.com/advisories/dll_advisory.txt

[LOB 97] LOBEL, Mark. PricewaterhouseCoopers. Security Dymanics Technologies Inc. *The Case for Strong User Authentication*. 1997. 11/03/99. http://www.securid.com/products/whitepapers/casestrong-wp.html

[MAC 99] McGARVEY, Joe. Inter@tive Week. ZDNet. *Protocol Promoted to Beef Up IP Security*. September 24, 1999. 24/09/99. http://www.zdnet.com/intweek/stories/news/0,4164,2340243,00.html

[MAI 99] MAIER, Timothy W. Insight Magazine On-Line. *Is U.S. Ready for Cyberwarfare?*. Vo. 15, N. 13, April 5-12, 1999. 17/03/99. http://www.insightmag.com/articles/story4.html

[MAN 99] MANSFIELD, Nick. Secure Computing. *A Practical Look at Information Security Management*. June, 1999. 16/06/99. http://www.wetcoast.com/securecomputing/

1999_06/feature/article.html.

[MCC 98] McCLURE, Stuart. Info World. *PKI Tames Network Security*. 11/03/99. September 14, 1998. http://www.infowold.com/cgi-bin/displayArchive.pl?/98/37/pkia.dat.htm

[MCC 99] McCLURE, Stuart; SCAMBRAY, Joel. *Beware of the obvious: Ubiquitous SNMP provides a back door to your network secrets*. February 1, 1999. 01/01/99. http://www.infoworld.com/cgi-bin/displayNew.pl?/security/990201sw.htm

[MCC 00] McCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. *Hacking Exposded: Network Security Secrets & Solutions*. Osborne. 2000.

[MIN 97] MINES, Christopher; GOODTREE, David; GOLDBERG, Mark L.; MacDONALD, Megan K. Forrester Research, Inc. Volume Two, Number Six, November 1997. http://www.forrester.com

[MIO 98] MIORA, Michael; COBB, Stephen. Information Security. *Springing Into Action*. May, 1998. 25/01/99. http://www.infosecuritymag.com/cirtified.htm

[MOD 99] Módulo Security Solutions. Mailing Lists. March, 12 1999. http://www.modulo.com.br

[MOS 99] MOSKOWITZ, Robert. Network Computing. *DES Is Dead. Long Live ... Well, Um, What ?* 24/03/99. March 22, 1999. http://www.techweb.com/se/directlink.cgi?NWC19990322S0017

[MUR 99] MURRAY, William. Infosecurity Magazine. *You Can't Buy PKI*. June, 1999. 16/06/99. http://www.infosecuritymag.com/jun99/buy_pki.htm

[NAK 00] NAKAMURA, Emilio T. Artigo submetido ao SSI-2000 – Simpósio sobre Segurança em Informática 2000. *Segurança no Acesso Remoto VPN*. Agosto, 2000.

[NEW 98] NEWMAN, David; GIORGIS, Tadesse; YAVARI-ISSALOU, Farhad. Data Communications. *VPNs: Safety First, But What About Speed?*. July 1.998. Acesso Web: 23/09/98. http://www.data.com/lab_tests/first.html.

[NEW 99] NEWMAN, David. Data Communications. *Super Firewalls!*. May 21, 1999. http://www.data.com/issue/990521/firewalls.html.

[NET 99] Netscape. Netscape Netcenter. *Understanding PKI* 13/11/99. http://verisign.netscape.com/security/pki/understanding.html

[NIS 00] National Institute of Standards and Technology. *An Introdution to Computer Secu-*

rity: The NIST Handbook. Technology Administration. U.S. Department of Commerce. Special Publication 800-12.

[NUT 99] NUTTALL, Chris. BBC News. *Sci/Tech Virtual 'Nuked' on Net.* January 26, 1999. 24/03/99. http://www.news.bbc.co.uk/hi/english/sci/tech/newsid_263000/263169.stm

[ODS 99] ODS Networks. *Advancing the Art of Intrusion Detection.* 04/02/2000. http://www.ods.com/security/info/behavorial.shtml.

[ODW 97] O'DWYER, Frank. Rainbow Diamond Limited. *Hiperlink Spoofing: An Attack on SSL Server Authentication.* January 3, 1997. http://www.iol.ie/~fod/sslpaper/sslpaper.htm

[PIN 98] PINCINCE, Tom. Network World. *Are VPNs ready for prime time? Yes, for remote access.* May 25, 1998. http://www.nwfusion.com/forum/0525vpnyes.html. 29/01/99.

[POR 99] Security Portal. *ISS Helps to Deliver Practical, Integrated Security Management Solutions With Availability of First ANSA Software Development Kit.* Mar, 1999. 10/03/99. http://www.securityportal.com/topnews/19990301isspr.html

[PHR 99] PHRACK MAGAZINE. *Perl CGI Problems.* Vol. 9, Issue 55, Article 07. September 9, 1999. http://www.phrack.com/search.phtml?view&article=p55-7. 02/01/1999.

[RAD 99] RADCLIFF, Deborah. *Hacker For Hire.* January 14, 1999. 27/08/99. http://www.infowar.com/hacker/99/hack_011999b_j.shtml

[RAE 97] RAENY, Reto E. *Firewall Penetration Testing.* The George Washington University. Cyberspace Policy Institute. January, 1997.

[RAN 01] RANUM, M. J. *Thinking About Firewalls.* Glenwood, Maryland: Trusted Information Systems, Inc.

[RAN 95] RANUM, Marcus J. *On the Topic of Firewall Testing.* 1995. http://www.clark.net/pub/mjr/pubs/fwtest/index.htm. 12/01/2000.

[RAN 99] RANUM, Marcus J. Network Flight Recorder, Inc. *Intrusion Detection: Challenges and Myths.* 20/04/99. http://www.nfr.net/forum/publications/id-myths.html

[ROT 98] ROTHKE, Ben. Information Security. *New Authority Figures.* January 1998. 25/01/99. http://www.infosecuritymag.com/rothke_art.htm

[ROT 98-2] ROTHKE, Ben. Information Security. *Crypto: RSA vc. ECC.* 25/01/99. August 1998. http://www.infosecuritymag.com/august/crypto.htm

[ROT 98-3] ROTHKE, Ben. Information Security. *Crypto: Plain & Elegant.* 25/01/99. July 1998. http://www.infosecuritymag.com/crypto.htm

[ROT 99] ROTHSTEIN, Philip Jan. Information Security. *Now What?.* May 1999. 15/06/99. http://www.infosecuritymag.com/may99/feature.htm

[ROT 99-B] ROTHKE, Ben. Information Security. *The .38 Special of Cracking.* May 1999. 15/06/99. http://www.infosecuritymag.com/may99/news.htm

[RSA 99] RSA Security. *Understanding Public Key Infrastructure (PKI).* Technology White Paper. 1999.

[SAL 99] SALAMONE, Salvatore. Internet Week. *IT Managers Seek Answers To VPN Performance Queries.* January 25, 1999. http://www.techweb.com/se/directlink.cgi?INW19990125S0006. 29/01/99.

[SAN 99] Sandia National Laboratories. *Sandia Researchers Develop World's Fastest Encryptor.* 07/07/99. http://www.sandia.gov/media/NewsRel/NR1999/encrypt.htm

[SAN 99-2] Sans Institute. *Intrusion Detection FAQ [Version 1.03].* 01/02/00. http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.htm

[SCH 96] SCHNEIER, Bruce. *Applied Cryptography.* Second Edition. John Wiley & Sons, Inc, 1996.

[SCH 98] SCHNEIER, Bruce. *Security Pitfalls in Cryptography.* Counterpane Systems, 1998.

[SCH 98-2] SCHNEIER, Bruce. Information Security. *Scrambled Message.* October, 1998. 25/01/99. http://www.infosecuritymag.com/oct/edgewise.htm

[SCH 99] SCHWARTAU, Winn. Information Security. *Infrastructure Is Us.* June 1999. 06/06/99. http://www.infosecuritymag.com/lun99/Infrastruc.htm

[SCH 95] SCHNEIER, Bruce. *Applied Cryptography, Second Edition.* John Wiley & Sons, Inc. 1995.

[SCH 99-1] SCHNEIER, Bruce. Counterpane Systems. *The Future of Crypto-Hacking.* Crypto-Gram. July 15, 1999. http://www.counterpane.com.

[SCH 99-2] SCHNEIER, Bruce. Information Security. *The 1998 Crypto Year-in-Review.* January

1999. 25/02/99. http://www.infosecuritymag.com/jan99/crypto.htm

[SCH 00] SCHNEIER, Bruce; ELLISON, Carl M. *Tem Risks of PKI: What You're Not Being Told About Public Key Infrastructure.* Computer Security Journal. Volume XVI, Number 1, 2000.

[SEC 97] Secure Computing. *Cyber Terrorism.* July, 1997.

[SEC 97-2] Secure Computing. *Security in the Internet Age.* September, 1997. 29/09/98. http://www.wetcoast.com/securecomputing/september/article/article.html.

[SEC 98] Secure Computing. *Corporate Security: The Way Ahead.* November 1998. 01/03/99. http://www.wetcoast.com/securecomputing/1998_11/cover/cover.html

[SEC 98-1] Secure Computing. *The New Cold War – Industrial Espionage.* April, 1998. 29/09/98. http://www.wetcoast.com/securecomputing/1998_04/cover/cover.html.

[SEC 98-2] Secure Computing. *Access Control.* Dezembro 1998. 01/03/99. http://www.wetcoast.com/securecomputing/1998_11/buyers/buyers.html

[SEC 98-3] SecureXpert Labs Advisory SX-98.12.23-01. *Widespread DoS Vulnerabilityh can Crash Systems or Disable Critical Services.* 03/01/2000. http://packetstorm.securify.com/new-exploits/nmap-DoS-2.txt.

[SEC 98-4] Security Magazine. June 1.998. *Virtual Private Network.* 23/09/98. http://www.westcoast.com/securecomputing/1998_06/buyers/buyers.html.

[SEC 99] Secure Computing. *Border Control.* January 1999. 02/03/99. http://www.wetcoast.com/securecomputing/1999_01/cover/cover.html

[SEC 99-2] Secure Computing. *Secure Computing.* February 1999. 02/03/99. http://www.wetcoast.com/securecomputing/1999_02/editor/editor.html

[SEC 99-3] Secure Computing. *Security Suites.* February 1999. 02/03/99. http://www.wetcoast.com/securecomputing/1999_02/testc/products.html

[SEC 99-4] Secure Computing. *Digital Certificates: Proven Technology, Upcoming Challenges.* February 1999. 04/03/99. http://www.wetcoast.com/securecomputing/1999_02/feature/feature.html

[SEC 99-5] Secure Computing. *Biometrics.* March 1999. 04/03/99. http://www.wetcoast.com/securecomputing/1999_03/cover/cover.html

[SEC 99-6] Secure Computing. *PKI – Public Key Infrastructure*. March 1999. 04/03/99. http://www.wetcoast.com/securecomputing/1999_03/survey/survey.html

[SEC 99-7] Secure Computing. *Firewalls*. April 1999. 16/04/99. http://www.wetcoast.com/securecomputing/1999_04/survey/survey.html

[SEC 99-10]Secure Computing. *Policy Management*. April, 1999. 16/04/99. http://www.wetcoast.com/securecomputing/1999_04/cover/cover.html.

[SEI 00] SEIFRIED, Kurt. *Network Intrusion Detection Systems and Virus Scanners – Are They the Answer?*. January 5, 2000. http://www.securityportal.com/direct.cgi?/closet/closet20000105.html.

[SHA 98] Shake Communications. *How to Develop a Simple yet Secure Password System*. March 1998. 15/01/99. http://www.shake.net/solutions/ssj/march98/password_march98.html

[SHA 98-B]Shake Communications. *Security News in Brief 1997 to 1998*. March 1998. 15/01/99. http://www.shake.net/solutions/ssj/march98/briefs_march98.html

[SHA 98-C]Shake Communications. *The Password Cracker that Eats Windows NT for Breakfast*. March 1998. 15/01/99. http://www.shake.net/solutions/ssj/march98/l0pht_march98.html

[SHA 98-4] SHAMIR, Adi; SOMEREN, Nicko van. *Playing Hide and Seek With Stored Keys*. September 22, 1998.

[SHO 98] SHOK, Glen. LAN Times. May, 1998. *Secure your WAN with ease*. 23/09/98. http://www.lantimes.com/testing/98may/805b050a.html.

[SIG 99] SIGNAL 9 SOLUTIONS. *ConSeal PC Firewall*. 10/08/99. http://www.signal9.com.

[SKO 98] SKOUDIS, Edward. Information Security. *Fire in the Hole*. July 1998. 25/01/99. http://www.infosecuritymag.com/fire.htm

[SMI 99] SMITH, Richard. Information Security. *Encryption Across the Enterprise*. January 1999. 25/02/99. http://www.infosecuritymag.com/jan99/feature.htm

[SON 97] SOMMER, Peter. Secure Computing. *Cyber Extortion. Part 3: Preventative Measures and Managing the Crisis*. April, 1997.

[SPI 99] SPITZNER, Lance. *How Stateful is Stateful Inspection? Understanding the FW-1 State Table*. June 11, 1999. 14/07/99. http://www.enteract.com/~lspitz/fwtable.html

[SPI 99-2] SPITZNER, Lance. *To Build a Honeypot.* October 25, 1999. 02/01/00. http://www.enteract.com/~lspitz/honeypot.html

[SPO 99] Security Portal. *United States Encryption Export Policies.* 19/07/99. http://www.securityportal.com/coverstory19990719.html

[STA 98] STALLINGS, William. Information Security. *A Secure Foundation for VPNs.* March 1998. 25/01/99. http://www.infosecuritymag.com/vpn.htm

[STA 99] STALLINGS, William. Information Security. *SNMPv3: Simple & Secure.* January 1999. 25/02/99. http://www.infosecuritymag.com/jan99/feature2.htm

[TIM 98] TimeStep Corporation. *Understanding the IPSec Protocol Suite.* December 1998. http://www.timestep.com

[TRA 98] *Transparent Proxying.* http://packetstorm.securify.com/UNIX/firewall/ipfwadm/ipfwadm-paper/node5.html#SECTION00050000000000000000. 25/01/00.

[TRI 98] TRICKEY, Fred L. Information Security. *Secure SSO: Dream On?.* September 1998. 25/01/99. http://www.infosecuritymag.com/sept/feature.htm

[TRU 99] TRUSTe. Proposed by Ernest & Young LLP. *Building a Web You Can Believe In.* 1999. 12/03/99. http://www.truste.org/webpublishers/pub_verification.html

[TWI 99] TWINKLE. http://jya.com/twinkle.eps

[ULS 98] ULSCH, MacDonnell. Information Security. *Hold Your Fire.* July 1998. 25/01/99. http://www.infosecuritymag.com/hold.htm

[ULS 99] ULSCH, MacDonnell; JUDGE, Joseph. Information Security. *Bitter-Suite Security.* January 1999. 25/02/99. http://www.infosecuritymag.com/jan99/cover.htm

[USH 99] United States House of Representatives. *Statement of The Honorable John J. Hamre. Deputy Secretary of Defense.* Washington, D.C.: February 1999. 15/03/99. http://www.house.gov/hasc/testimony/106thcongress/99-02-23hamre.htm

[YAH 99] Yahoo! Finance. Business Wire. *GTE Joins Security Research Alliance.* March 15, 1999. 19/03/99. http://biz.yahoo.com/bw/990315/ma_gte_1.html

[WES 98] West Coast Publishing Ltd. Secure Computing. *Active Security Solution – The Network Associates Active Security.* 1998. 24/09/99. http://www.wetcoast.com/review/net_assoc/index.htm.

[WOO 98] WOODWARD, John D. Information Security. *Believing in Biometrics*. February 1998. 25/01/99. http://www.infosecuritymag.com/biometrics.htm

[WOO 99] WOOD, Charles Cresson. Infosecurity Magazine. *Policies: The Path to Less Pain ... & More Gain*. August, 1999. 24/09/99. http://www.infosecuritymag.com/aug99/cover.htm.

[WU 98] WU, David; WONG, Frederick. *Remote Sniffer Detection*. Computer Science Division. University of California, Berkeley. December 14, 1998.

[ZDN 98] ZDNet. *Signing and Trus*. September, 30 1998. 11/03/99. http://www.zdnet.com/devhead/filters/homepage