Curso de Ciência da Computação

Universidade Estadual de Mato Grosso do Sul

Um protótipo para detecção de ataques de negação de serviço

Frances Albert Santos

Prof. Dr. Fabrício Sérgio de Paula (Orientador)

22 de Novembro de 2013

Um protótipo para detecção de ataques de negação de serviço

Frances Albert Santos

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Frances Albert Santos e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 22 de Novembro de 2013.

Prof. Dr. Fabrício Sérgio de Paula (Orientador)

RESUMO

Os ataques de negação de serviço têm causado grande prejuízo aos seus alvos, degradando a qualidade dos serviços oferecidos pelas vítimas, podendo torná-los totalmente indisponíveis a clientes legítimos. Esses ataques vêm evoluindo e se tornando mais sofisticados e difíceis de detectar. O objetivo deste estudo é investigar ataques de negação de serviço, em busca de compreendê-los, analisando suas diferentes formas de funcionamento. Uma vez compreendido o funcionamento de alguns ataques de negação de serviço, um protótipo de detecção baseado em anomalia do trafego é proposto. Por fim, testes são realizados para analisar a eficiência do protótipo em um ambiente com ataques de negação de serviço.

Palavras-chave: Segurança. Detecção. Anomalia.

LISTA DE SIGLAS

DARPA	-	Defense Advanced Research Projects Agency
DDoS	-	Distributed Denial of Service
DNS	-	Domain Name System
DoS	-	Denial of Service
Gbps	-	Gigabit per second
GCC	-	Gnu Compiler Collection
ICMP	-	Internet Control Message Protocol
IDS	-	Intrusion Detection System
IP	-	Internet Protocol
IRC	-	Internet Relay Chat
SMTP	-	Simple Mail Transfer Protocol
Pod	-	Ping of death
TCP	-	Transmission Control Protocol
UDP	-	User Datagram Protocol
URL	-	Uniform Resource Locator

LISTA DE TABELAS

LISTA DE FIGURAS

Capítulo 1

INTRODUÇÃO

Embora existam diferentes estimativas, pode-se afirmar que *softwares* maliciosos causaram danos da ordem de bilhões de dólares nas últimas décadas. Esses danos englobam destruição de dados, danos em *hardware* e, em uma grande parcela, tempo dispensado para lidar com o problema causado (LEHTINEN, 2006).

Um tipo bastante comum de ataque realizado por *softwares* maliciosos é a negação de serviço (DoS, do inglês *Denial of Service*). Esse tipo de ataque produz uma enorme quantidade de requisições de recursos – aparentemente normais – na máquina atacada, o que a sobrecarrega e a torna indisponível para atender requisições legítimas (LEHTINEN, 2006; WANG, 2009; BISHOP, 2004).

Em uma versão distribuída, os ataques são lançados para a máquina atacada a partir de diversos computadores distintos. Esses computadores, chamados de zumbis, são previamente atacados para tal propósito. Esse ataque é chamado de negação de serviço distribuído (DDoS, do inglês, *Distributed Denial of Service)* (LEHTINEN, 2006; WANG, 2009).

A título de exemplo, no ano de 2000 um estudante de 15 anos de Montreal, Canadá, lançou ataques de DDoS contra diversos alvos paralisando servidores Web por uma semana. Empresas como Amazon, eBay, Dell e Yahoo! tiveram substanciais perdas financeiras por causa desse ataque (WANG, 2009).

No Brasil, no final de janeiro e início de fevereiro de 2012, um grupo de *hackers* lançaram ataques de DDoS contra instituições financeiras como Banco Central do Brasil, Itaú/Unibanco, Bradesco, Banco do Brasil, HSBC, Santander, Caixa Econômica Federal, Cielo e Redecard, provocando em diversas situações instabilidades e indisponibilidade do *site* para clientes. No caso do ataque ao Banco do Brasil a carga no sistema de comunicação chegou a 8,7 Gbps (LOBO, 2012).

Uma pesquisa no Exploit Database (OFFENSIVE SECURITY, 2006), *site* que abriga um conjunto de *exploits* – programas maliciosos que exploram falhas conhecidas ou desconhecidas (LEHTINEN, 2006) –, revela quase uma centena desses programas construídos para explorar falhas que levam à negação de serviço.

1.1 Objetivo

O presente estudo teve o objetivo de analisar ataques de negação de serviço, distribuídos ou não, de forma a entender o funcionamento/implementação e propor um protótipo de detecção que pode ser aplicado em diferentes situações.

1.1.1 Objetivos específicos

Os objetivos específicos são:

- Configurar ambiente confinado para simulação de ataques;
- Pesquisar diferentes formas de negar serviço a usuários legítimos;
- Analisar métodos existentes que propõem mitigar os ataques;
- Implementação de um protótipo detector de ataque;
- Testar a aplicação desenvolvida com a finalidade de avaliar a eficiência.

1.2 Metodologia

O estudo sobre ataques de negação de serviço com o propósito de desenvolver um protótipo de detecção, reuniu informações da evolução dos ataques, o funcionamento e as características de alguns ataques, meios de efetuar a detecção, análise e discussão dos resultados e documentação sobre o desenvolvimento do estudo. Livros e páginas *web* foram os instrumentos de pesquisa do estudo.

Para o desenvolvimento do protótipo, foram utilizados os seguintes *softwares*:

- Sistema operacional GNU/Linux;
- Editor de texto Gedit;
- Biblioteca pcap.

A base de dados DARPA referente ao ano de 1999 (DARPA, 2013) é utilizada para o protótipo realizar os testes e os resultados serem comparados com a documentação fornecida pela DARPA.

1.3 Organização do Texto

Este texto está organizado em um único volume, além da seção de introdução, contém outras cinco seções, cujos conteúdos são sumarizados:

Capítulo 2

Histórico de Ataques de Negação de Serviço

Nesta seção é apresentado o histórico destes ataques até os dias de hoje, citando alguns exemplos reais de ataques e o prejuízo causado por eles.

Capítulo 3

Funcionamento de Ataques de Negação de Serviço

Nesta seção desenvolveu-se um estudo sobre o funcionamento de alguns ataques, os fatores que dificultam detectar um tráfego de ataque, as características do ataque e exemplos de ataques.

Capítulo 4

Técnicas de Detecção de Ataques de Negação de Serviço

Nesta seção foram abordados as técnicas existentes para efetuar a detecção de ataques, ressaltando a vantagem e desvantagem de cada uma. Algumas ferramentas de detecção são citadas como exemplo.

Capítulo 5

PDADoS

Nesta seção é apresentado o protótipo desenvolvido, que engloba uma aplicação de análise do comportamento do tráfego, gerando alerta de detecção quando há anomalias no tráfego. Além de testes e avaliação da eficiência do protótipo proposto.

Capítulo 6

Considerações Finais

Nesta seção o estudo é concluído e trabalhos futuros são propostos.

Capítulo 2

HISTÓRICO DE ATAQUES DE NEGAÇÃO DE SERVIÇO

Ataques do tipo DoS e DDoS iniciaram em meados dos anos 90, causando graves problemas aos seus alvos (PIVOTTO, 2006). Nesta seção é apresentado o histórico dos ataques de negação de serviço, desde o surgimento dos primeiros ataques até os ataques mais recentes.

Inicialmente, no ano de 1996, foram detectadas falhas no protocolo TCP/IP e então, em 1997 ataques em larga escala em redes IRC se iniciaram. No mesmo ano, uma técnica conhecida por *Smurf* teve início. Ela consistia em enviar pacotes na rede a um determinado alvo, então o atacante era capaz de multiplicá-los por centenas e milhares, de acordo com o tamanho da rede. Apesar de eficaz esses ataques podiam ser facilmente evitados, desligando a capacidade de recepção *broadcast* nos roteadores. Outra forma de ataque foi adotada pelos atacantes, o envio de centenas de pacotes ao alvo, sendo o início dos ataques DoS (PIVOTTO, 2006).

Em 1998, as velocidades das conexões se tornaram mais homogêneas e esses ataques se tornaram menos frequentes. Então teve início os ataques do tipo DDoS, que semelhantes aos DoS, eles inundam um sistema alvo, porém, como o próprio nome diz, este tipo de ataque não se origina de uma única máquina, os atacantes têm ao seu dispor uma rede de máquinas programadas para realizar o ataque ao seu comando, sendo um ataque eficaz e mais agressivo aos anteriores (PIVOTTO, 2006). Na próxima seção o funcionamento desses ataques são explorados detalhadamente.

Desde então, a técnica dos ataques DDoS tiveram melhorias, como combiná-lo com *worms*, e em 2000 teve início uma série de ataques a grandes *sites*, como eBay, Yahoo, Amazon e CNN, gerando um prejuízo de milhões de dólares. Empresas renomadas no mercado (como a Microsoft) e órgãos públicos (como o FBI), sofreram com ataques desse tipo (PIVOTTO, 2006).

Atualmente, ataques DDoS ainda são muito prejudiciais, tendo como alvos desde *sites* governamentais até domínios de grandes bancos, que são redes consideravelmente seguras. Sistemas de monitoramento de Internet apontam que esses ataques estão cada vez maiores. Em Setembro de 2012 teve em média 1.67 Gbps de requisição contra um *site*, isso representa um aumento de 72% em relação ao mesmo período do ano anterior. Tendo aumentado também a intensidade de ataques de médio alcance (2 a 10 Gbps) em torno de 14.35%, enquanto ataques grandes (10 Gbps ou mais) subiram até 90% no mesmo ano, sendo o maior de 100.84 Gbps (SOCKRIDER, 2013).

Como exemplo de ataques DDoS no Brasil em 2012, instituições financeiras tiveram seus *sites* atacados por um grupo de *hackers* conhecidos por *Anonymous*. Tais como: Itaú/Unibanco, Bradesco, Banco do Brasil e HSBC. Ainda tentaram tirar do ar o portal do Banco Central, que apesar de ter apresentado instabilidade, resistiu ao ataque. Outros bancos (Santander e a Caixa Econômica Federal) também foram alvos dos *hackers*, no entanto, resistiram ao ataque. Posteriormente ao ataque aos bancos, os alvos foram as administradoras de cartão de crédito. Os *sites* da Cielo e da Redecard tiveram problemas de conexão (LOBO, 2012).

Capítulo 3

FUNCIONAMENTO DE ATAQUES DE NEGAÇÃO DE SERVIÇO

Ao contrário dos conhecidos ataques da Internet, ataques DoS e DDoS não possuem o intuito de invadir um computador para roubar informações confidenciais da vítima, como senhas bancárias, dados de cartões, nem mesmo para causar danos a arquivos armazenados na máquina alvo. Estes ataques buscam bloquear os serviços fornecidos pela vítima aos demais usuários legítimos, sendo difícil de detectar quando um ataque está em curso, uma vez que ataques DoS e DDoS se assemelham a requisições legítimas, porém, eles aumentam o tráfego de pacotes na rede a ponto de inundar a vítima de requisições não legítimas (LAUFER et al., 2005).

Arquitetados por atacantes que não necessitam de grande experiência, já que existem diversas ferramentas disponíveis na Internet para realizar ataques DoS e DDoS (PIVOTTO, 2006). Os ataques DDoS consistem em 3 etapas, primeiramente o atacante utiliza ferramentas automáticas para comprometer a segurança das máquinas e obter acesso privilegiado, posteriormente é instalado um software DDoS nas máquinas invadidas, então com a rede de ataque pronta, o atacante orquestrará um ataque distribuído a uma ou mais vítimas (SOLHA, 2013). Como consequência do ataque, serviços providos pela vítima podem ser congelados ou reinicializados, ou ainda, o esgotamento total dos recursos necessários para prover o seu serviço (LAUFER et al., 2005).

3.1 Funcionamento de Ataques

Nesta seção são detalhadas algumas técnicas utilizadas para promover um ataque de negação de serviço. Estes ataques podem explorar falhas nos protocolos de rede existentes, ou utilizar da força bruta (*flooding*) para inundar a vítima. Dentre os ataques *flooding*, eles podem ser considerados diretos, com *spoofing* ou em *loop*. Os ataques diretos ocorre quando o atacante realiza o ataque sem alterar seu endereço IP. Com *spoofing*, onde o atacante utiliza técnicas para falsificar o endereço IP de origem, dificultando o processo de identificação da origem dos ataques. Em *loop*, quando o atacante utiliza *spoofing*, com a característica de o endereço IP falsificado ser o mesmo que o endereço IP da vítima (SILVA, 2012).

Os ataques que exploram falhas em protocolos de rede, fazem uso de características que garantiram à Internet o sucesso atual, como a comutação por pacotes e dos protocolos TCP/IP, que possuem alguns princípios que facilitam o avanço de ataques de negação de serviço.

Segundo Laufer (2005), na Internet não há reserva de recursos em uma comunicação entre dois nós, adotando-se o "melhor esforço" como procedimento básico. Aproveitando-se do procedimento adotado, um ataque de negação de serviço consumirá o máximo dos recursos da rede e os usuários legítimos serão prejudicados com a ausência de recursos, que estarão sendo empregados em ações maliciosas.

Outro fator utilizado nos ataques de negação de serviço é o roteamento. O roteador tem a função de encaminhar pacotes entre a fonte e o destino, para realizar esta tarefa, diferentes rotas podem ser utilizadas. A transparência na seleção de rotas que são utilizadas e a ausência de um mecanismo mais seguro para garantir o correto roteamento até o destino, pode gerar falhas na comprovação de autenticidade do endereço IP de origem de cada pacote, desta forma, os atacantes ficam ocultos, porque não necessitam de comunicações bidirecionais. Eles podem forjar seus verdadeiros endereços IP, sem deixar vestígios para serem rastreados (LAUFER et al., 2005).

A topologia da Internet é outro fator facilitador dos ataques de negação de serviço, pois o núcleo da rede é constituído por enlaces de alta capacidade, enquanto as bordas normalmente possuem enlaces de baixa capacidade conectados ao núcleo. Com essa assimetria, nós presentes no núcleo podem encaminhar pacotes de várias origens distintas para diferentes destinos, podendo ocasionar uma sobrecarga nos nós presentes nas bordas pelo tráfego agregado de outros nós (LAUFER et al., 2005).

Quando um ataque se inicia, cada nó da rede é responsável por uma pequena parcela do tráfego agregado, então dificilmente alguma anomalia será identificada em determinadas redes e passam sem problemas pelos roteadores (LAUFER et al., 2005).

3.2 Classificação de Ataques

Esta seção aborda alguns tipos de ataque de negação de serviço e como podem ser classificados de acordo com seu funcionamento.

3.2.1 Ataque por inundação

Este é um ataque de negação de serviço muito comum, o ataque inunda segmentos TCP SYN, tirando vantagem do procedimento de abertura de conexão do protocolo de transporte TCP (LAUFER et al., 2005).

Conhecido por atender serviços que necessitam que seus dados sejam entregues ao destino de forma confiável, o protocolo TCP segue um procedimento conhecido como o acordo de três vias (*three-way handshake*) para estabelecer uma nova conexão entre dois nós (LAUFER et al., 2005).

O acordo de três vias se inicia com o pedido do cliente que envia um segmento TCP SYN para o servidor, solicitando uma nova conexão. Com o segmento, um parâmetro denominado número de sequência inicial permite reconhecer dados repetidos, fora de ordem ou perdidos. Após o segmento TCP SYN chegar ao servidor, leva-se um tempo para processar a solicitação e alocar memória para reter informações do cliente. Terminado o processamento e alocação, o servidor notifica o cliente que sua solicitação foi aceita, enviando-lhe um segmento TCP SYN/ACK, que reconhece o número de sequência do cliente e envia o número de sequência inicial do servidor. Finalmente, o cliente conclui a abertura da conexão enviando um segmento TCP ACK para reconhecer o número de sequência do servidor (LAUFER et al., 2005).

Desta forma, o ataque por inundação consegue explorar a fragilidade da vítima, gerando segmentos a uma taxa maior que a vítima possa processar, ocasionando a falta de processamento de novos pedidos em um tempo hábil, como mostra a **Figura 3.1**. Além do processamento, a memória é outro recurso que pode ser explorado durante um ataque. Forjando o endereço IP de origem dos pacotes, substituindo-o por algum endereço não utilizado. Com o envio destes pacotes forjados, iniciam-se inúmeras conexões semiabertas que consomem a memória da vítima. Este ataque só obtém sucesso se for gerado segmentos a uma taxa maior que a liberação dos recursos. Portanto, quando um usuário legítimo acessar o serviço alvo dos ataques, terá seu acesso descartado, juntamente com o tráfego de ataque (LAUFER et al., 2005).

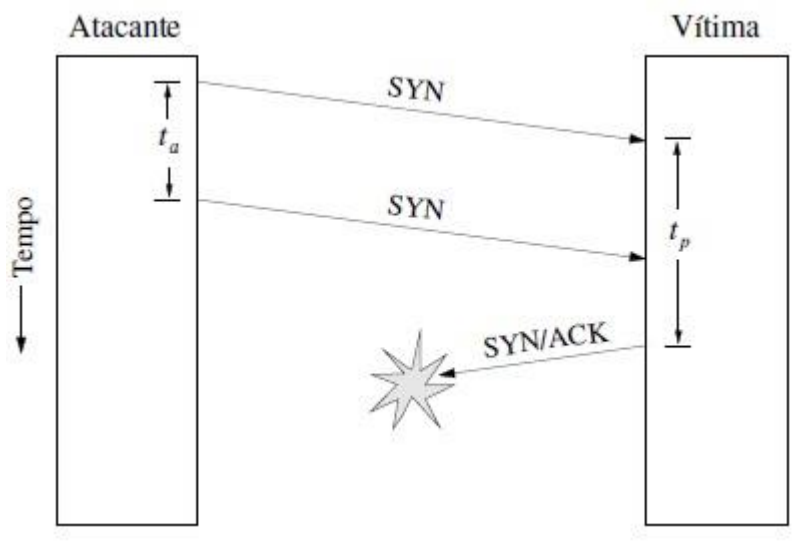

Figura 3.1: Ataque de negação de serviço por inundação, onde a taxa de ataque (t_a) deve ser menor que a taxa de processamento (t_p) da vítima (LAUFER et al., 2005).

3.2.2 Ataque por refletor

O ataque por refletor é outro tipo de ataque por negação de serviço. Semelhante ao ataque por inundação, este ataque também busca consumir recursos da vítima. O que difere os ataques é que no ataque por refletor existe uma estação intermediária entre a vítima e o atacante (LAUFER et al., 2005).

Esse ataque utiliza a estação intermediária para refletir o tráfego de ataque em direção à vítima, como mostra a **Figura 3.2**. A identidade do atacante permanece ainda mais sigilosa com essa manobra, já que o tráfego que chega à vítima é oriundo do refletor e não do atacante. Deve-se levar em consideração que o refletor utilizado no ataque leve um tempo menor que a vítima para processar as requisições, senão o refletor irá inundar e seu tráfego excedente será descartado sem causar efeitos a vítima (LAUFER et al., 2005).

O ataque por refletor não se restringe a um determinado protocolo de transporte. Existindo um protocolo que envie pacote de resposta ao atender a algum tipo de requisição o seu funcionamento transcorrerá normalmente (LAUFER et al., 2005).

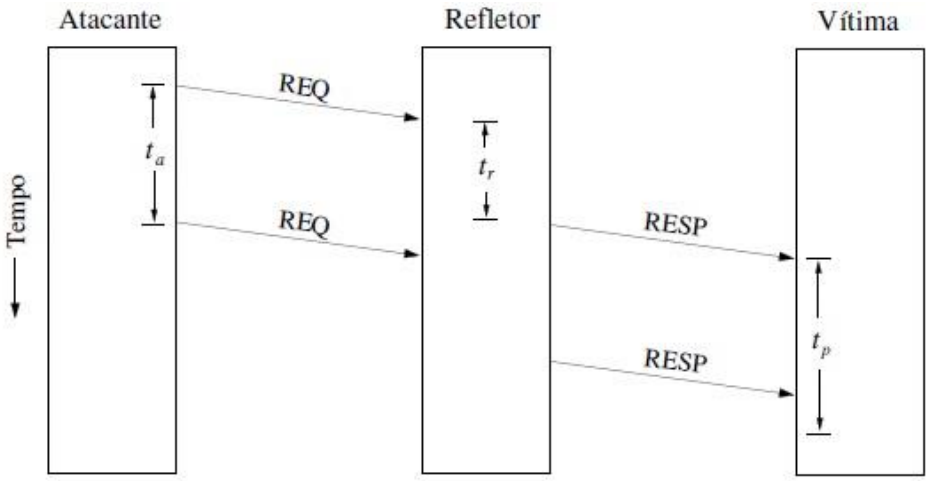

Figura 3.2: Ataque de negação de serviço por refletor, onde a taxa de ataque tem que ser menor que a taxa de processamento da vítima ($t_a < t_p$) e a taxa do refletor também deve ser menor ($t_r < t_p$) para o ataque obter sucesso (LAUFER et al., 2005).

3.2.3 Ataque à infraestrutura de rede

Ataques de negação de serviço vistos anteriormente tomavam como alvo suas vítimas diretamente, entretanto quando se busca negar serviços de *sites* globalmente conhecidos, ataques distribuídos de pequena escala não são capazes de consumir os recursos como um todo, afinal a vítima é superdimensionada, com recursos de processamento e memória em larga escala. Então para ter êxito, o atacante pode direcionar seus esforços em algum ponto fundamental para o funcionamento do serviço independente da vítima (LAUFER et al., 2005).

Atingindo a infraestrutura de rede, seja consumindo toda a banda passante ou inundando os servidores DNS da vítima, por mais que a vítima possua recursos suficientes para atender as requisições que lhe chegam, inúmeras requisições se perdem na infraestrutura de rede (LAUFER et al., 2005).

Alvo comum em ataques à infraestrutura de rede são os roteadores responsáveis por encaminhar os pacotes até a vítima. Ataques por inundação ou outros tipos de ataques podem ser direcionados aos roteadores. Onde o atacante pode preencher a tabela de encaminhamento do roteador responsável por encaminhar pacotes à vítima, dificultando a busca de endereços na tabela e conseguindo atrasar significativamente o encaminhamento dos pacotes ou enganar o roteador para desviar o tráfego legítimo para locais diferentes (LAUFER et al., 2005).

3.2.4 Ataque por vulnerabilidade

O ataque por vulnerabilidade é outra maneira de se negar serviços providos pela vítima e deixá-la inativa de algum modo. Este tipo de ataque explora vulnerabilidades de suas vítimas, como na implementação da pilha de protocolos ou aplicações próprias (LAUFER et al., 2005).

Alguns ataques de negação de serviço que aproveitam vulnerabilidades presentes em suas vítimas já exploraram tais vulnerabilidades no protocolo IP. O protocolo IP permite a quebra de pacotes grandes para serem transmitidos em pacotes menores e enviam-nos separadamente. Para que o receptor possa agregá-los em um único pacote novamente, esses fragmentos possuem identificação, o ataque se aproveita dessa vulnerabilidade, envia inúmeros fragmentos IP pertencentes ao mesmo pacote com números de sequência que se sobrepõem (CERT, 1997 Apud LAUFER et al., 2005). Assim, quando os fragmentos chegam a vítima, há alguns intencionalmente mal formados, ocasionando no congelamento ou reinicialização (LAUFER et al., 2005). Além de sobrepor os números de sequência, o atacante também pode inundar a vítima com diversos fragmentos, com identificadores distintos, e como a vítima armazena os fragmentos até que todos cheguem ou o temporizador estoure, e que cada fragmento deve ser processado para determinar sua posição correta no pacote original, o ataque esgotaria a memória da vítima e elevaria seu processamento (LAUFER et al., 2005).

3.2.5 Ataque distribuído

Os ataques distribuídos são comumente utilizados para potencializar ataques por inundação que necessitam elevar a capacidade de consumir algum recurso da vítima (LAUFER et al., 2005).

Para realizar esse tipo de ataque, é necessário que diversas estações gerem o tráfego de ataque em direção à vítima. No geral, as estações utilizadas para realizar ataques distribuídos não pertencem ao atacante, ele as controla aproveitando de alguma falha de segurança destas estações e instala um programa que lhe permite controlá-las remotamente e faz tudo removendo seus rastros, para que não seja identificado, tais máquinas são chamadas de agentes, zumbis ou escravos (LAUFER et al., 2005).

A invasão e controle de máquinas zumbis são tarefas árduas enfrentadas pelos atacantes que realizam estes ataques, considerando que os ataques de negação de serviço distribuídos são compostos por centenas ou milhares de zumbis (MIRKOVIC et al., 2004 Apud LAUFER et al., 2005). Então os atacantes desenvolveram ferramentas para automação dessas tarefas, eles

encontram estações vulneráveis e as invadem automaticamente. Com ferramentas que possibilitam a criação de redes hierárquicas, eles controlam um grande número de zumbis. Um ataque a uma determinada vítima, tem o início simultâneo, com a participação de diferentes estações zumbis, através de um simples comando (LAUFER et al., 2005).

Nos ataques de negação de serviço distribuídos, os autores utilizam estações intermediárias entre si e os zumbis, para proteger sua identidade. Estas estações são chamadas de mestres, onde cada uma é responsável pelo controle de um conjunto de zumbis, como podemos ver na **Figura 3.3**. O atacante envia suas ordens de ataque as estações mestres que por sua vez repassam aos zumbis. Para rastrear o verdadeiro atacante, é preciso descobrir quem são os zumbis, posteriormente quem são os mestres e enfim chegar ao autor do ataque, portanto, quanto maior for a hierarquia dessas camadas, mais protegidos estarão os atacantes (LAUFER et al., 2005).

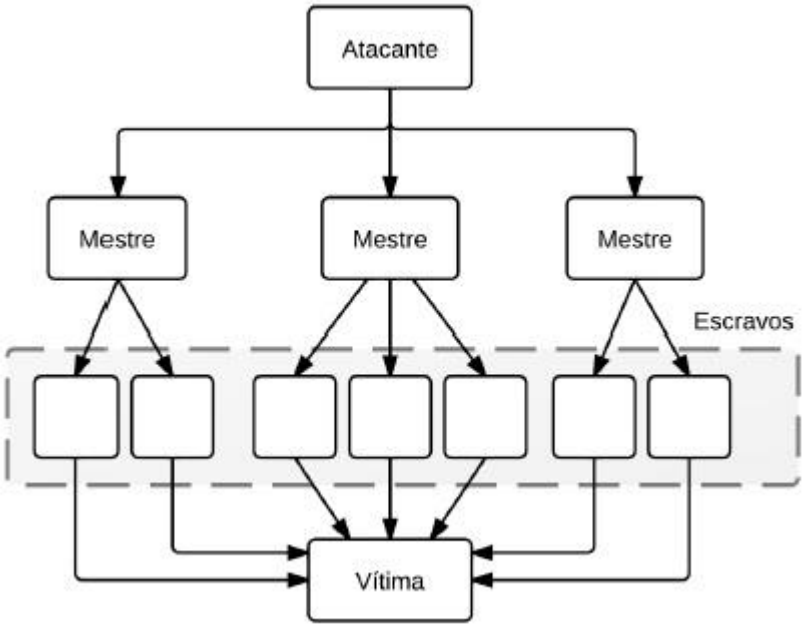

Figura 3.3: Ataque de negação de serviço distribuído (LAUFER et al., 2005).

3.3 Exemplos de Ataques

O *modus operandi* de alguns ataques conhecidos de negação de serviço são:

- **Apache2:** ataques direcionados a servidores *Web Apache*, que acontece quando o servidor não é capaz de resolver uma quantidade elevada de solicitações de serviços (SANTOS, 2009);

- **Back:** são enviadas requisições a um servidor *Web Apache* com diversas barras no endereço URL, atrasando o tempo de resposta, gerando posteriormente a negação de serviço do servidor Web (AZEVEDO, 2012);
- **Mailbomb**: são enviadas milhares de requisições a um servidor de *e-mail* através da porta 25, referente ao protocolo SMTP, comprometendo os recursos do servidor de *e-mail*, tornando-o indisponível para usuários legítimos, ou ainda, comprometendo a conta de algum usuário com elevado número de requisições (AZEVEDO, 2012);
- **Neptune:** são enviadas inúmeras requisições do tipo TCP/SYN com endereços de retorno inválidos, preenchendo a tabela de conexões do *host* destino. Com a tabela de conexões totalmente preenchida, a vítima passa a negar novas conexões. Este ataque explora a vulnerabilidade do protocolo TCP/IP para abertura de novas conexões (AZEVEDO, 2012);
- **Ping Of Death (pod)**: são enviadas requisições ICMP do tipo *ping* com pacotes maiores que 65.535 *bytes* para um *host* a ser atacado (AZEVEDO, 2012). Como o protocolo TCP/IP permite um tamanho máximo de um pacote até 65.536 *bytes*, com envio de pacotes *ping* maiores que os estabelecidos, o sistema operacional do *host* gastaria muitos recursos de armazenamento e processamento para responder a estes pacotes mal formados, causando a negação de serviços a requisições legítimas;
- **Process Table:** ataque criado especificamente para avaliação, onde para cada conexão estabelecida com o servidor, um novo processo é criado (SANTOS, 2009);
- **Smurf:** são enviadas requisições ICMP do tipo *ping* para um endereço de *broadcast* de uma rede intermediária. O atacante utiliza a técnica de *spoofing* para alterar o endereço IP de origem para o endereço IP da vítima. Assim, todos os *hosts* da rede intermediária responderam o *ping* enviando pacotes ICMP *echo reply* para a vítima, causando a inundação (SANTOS, 2012);
- **Teardrop:** são enviados pacotes IP fragmentados com o valor de sequência (*offset*) adulterados. Quando estes fragmentos chegam ao destino, não podem ser reagrupados, já que estão adulterados. Com o excesso de fragmentos dispersos, ocorre uma reinicialização ou congelamento do sistema da vítima (SANTOS, 2009);

- **Udpstorm:** explora a vulnerabilidade do protocolo UDP (SANTOS, 2009). Este ataque é capaz de atacar dois servidores de uma só vez. Enviando pacotes de dados com endereço IP de origem forjado, para um servidor alvo, este tentará se comunicar com o servidor de origem (que foi previamente forjado). Após receber os pacotes de dados, o segundo servidor alvo repetirá o processo, gerando um *loop* entre os servidores (GARBER, 2000);
- **Ipsweep e portsweep:** o ataque *ipsweep* determina quais alvos serão escutados através de uma varredura de inspeção. O *portsweep* é utilizado para saber quais portas estão abertas, de um computador específico da rede, ou sub rede (DARPA, 1999).

Capítulo 4

TÉCNICAS DE DETECÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO

É extremamente importante identificar quando um ataque de negação de serviço está em curso, uma vez que estes ataques simulam tráfegos reais. Somente após a detecção do ataque de negação de serviço, é possível iniciar ações para combatê-lo.

Atualmente os sistemas de detecção de intrusão, também conhecidos por IDS (*Intrusion Detection System*), podem ser agrupados em três classes (SILVA, 2012); (SANTOS, 2012).

1. Baseado nas informações contidas nos pacotes da rede, rastreando assinaturas de pacotes não legítimos;
2. Baseado em anomalias presente no tráfego de rede, onde o comportamento normal do tráfego é modelado e alterações significativas são consideradas ataque;
3. Híbridos, uma combinação dos métodos anteriores.

4.1 Baseado em Assinaturas

A detecção baseada em assinaturas identifica ataques de negação de serviço conhecidos. São estudados padrões únicos que diferenciam os ataques de requisições legítimas, construindo um banco de dados com estes padrões. Com esse banco de dados é realizado o monitoramento da rede buscando esses padrões. Este sistema de detecção é altamente eficiente na identificação de ataques e vulnerabilidades conhecidos, com baixíssimas taxas de falso negativo. A grande desvantagem deste método é que novas formas de ataque e variações dos ataques já conhecidos, permanecem invisíveis a detecção (ABLIZ, 2011).

A técnica de IDS baseado em assinaturas, consiste de uma lista contendo assinaturas de ataques e os respectivos alertas a serem enviados. Considerando as informações contidas nos pacotes que contém o ataque, como as listadas abaixo, são construídas as assinaturas (MUZZI, 2010 Apud BARBOSA, 2000):

- Portas de origem e destino;
- Número de sequência;
- *Flags* dos protocolos TCP, por exemplo, SYN e FIN;

- Outros campos do protocolo e suas opções;
- E principalmente uma pequena parte do conteúdo do protocolo da camada de aplicação.

Exemplos de ferramentas que realizam esta detecção são: Bro (PAXSON, 1999 Apud ABLIZ, 2011), MIB (CABRERA, 2001 Apud ABLIZ, 2011) e Spectral analysis (CHENG, 2002 Apud ABLIZ, 2011).

4.2 Baseado em Anomalias

Os mecanismos que utilizam esta técnica são capazes de detectar ataques com uma relativa eficiência, mas caso aumente o comportamento malicioso e novos ataques surjam, o método já não será eficiente (ABLIZ, 2011). Ao contrário da detecção baseada em assinaturas, a detecção baseada em anomalias não necessita conhecer assinaturas para constatar um ataque. Ela tem como premissa o comportamento do tráfego da rede em condições normais, e identifica possíveis ataques quando há desvios significativos do comportamento normal (SILVA, 2012). Através desse método é possível detectar ataques inéditos.

O grande desafio para utilizar essa técnica é determinar até quando pode ser considerado normal o tráfego de rede. Há muitos estudos que buscam estabelecer um parâmetro, mas ainda não existe, porque, para cada situação o comportamento do tráfego é distinto. Um limite para comportamento anômalo estrito pode rotular de forma equivocada um tráfego legítimo como anômalo (falso positivo). Por outro lado, um limite muito amplo poderia permitir a existência de muitos ataques sem ter a percepção de estar sendo atacado (falso negativo) (ABLIZ, 2011). Para utilização da técnica de detecção de anomalias no tráfego de rede com baixo índice de falso positivo e falso negativo, tem sido utilizadas técnicas de mineração de dados (*data mining*), tais como aprendizagem de máquina e detecção de *outliers* (SILVA, 2007).

Exemplos de ferramentas que realizam esta detecção são: MULTOPS (GIL; POLLETO, 2001 Apud ABLIZ, 2011), SYN *flood detection* (WANG, 2002 Apud ABLIZ, 2011) e NOMAD (TALPADE, 1999 Apud ABLIZ, 2011).

4.3 Híbridos

A detecção de ataques de negação de serviço em sistemas de detecção híbridos, combinam as duas técnicas anteriores. Estes sistemas, rastreiam assinaturas conhecidas ao

mesmo tempo que monitoram o tráfego de rede. As decisões são tomadas, considerando o tráfego de rede normal, como o comportamento intrusivo dos atacantes.

Exemplos de ferramentas que realizam esta detecção são: *Prelude*-IDS e *Tripwire* (MARTINS, 2012).

Capítulo 5

PDADoS

O presente estudo propõe um protótipo de detecção de ataques de negação de serviço baseado em anomalias. O protótipo desenvolvido batizado de PDADoS (Protótipo Detector de Ataque DoS), monitora o tráfego de rede em busca de ataques de negação por serviço por inundação que utilizam pacotes ICMP *echo reply*.

Ataques de negação de serviço com pacotes ICMP *echo reply*, podem ser realizados de duas formas: distribuída ou não. Na forma não distribuída, o funcionamento deste ataque é similar ao do *Smurf*, abordado nas **seções 2 e 3.3**, onde um único atacante envia grande quantidade de pacotes ICMP *echo request* em um curto espaço de tempo, para um endereço IP de *broadcast*. O atacante também utiliza a técnica de *spoofing* para alterar o endereço IP de origem dos pacotes enviados por ele para o endereço IP do alvo do ataque. Como consequência, é encaminhada uma enorme quantidade de pacotes ICMP *echo reply* para o alvo, causando a inundação de sua rede e consequentemente, negando serviço aos usuários legítimos. O protocolo ICMP será brevemente abordado na **seção 5.1**. Na versão distribuída, o ataque funciona de forma semelhante, porém, ao contrário do ataque no formato não distribuído, o atacante utiliza uma arquitetura de ataque semelhante a abordada na **seção 3.2.5**, onde ele enviará pacotes ICMP *echo request* para as máquinas zumbis, com o endereço IP de origem dos pacotes adulterados, assim, as máquinas zumbis responderão estes pacotes com pacotes ICMP *echo reply* com destino a vítima, gerando um imenso tráfego de rede de pacotes ICMP *echo reply* com o destino em comum. A vítima não conseguirá processar a todos e então passará a descartar não só estes pacotes, como todos os demais pacotes que estão chegando para ela, negando serviço aos usuários legítimos.

Nas seções subsequentes são apresentadas uma breve abordagem do protocolo ICMP para melhor entendimento de seu funcionamento, a análise do funcionamento do protótipo proposto, a base de dados DARPA que foi utilizado para simular o tráfego de rede para treinamento e avaliação de eficiência do protótipo, e por fim, os resultados e discussões.

5.1 Protocolo ICMP

O protocolo ICMP (*Internet Control Message Protocol*) é responsável por reportar eventos inesperados no tráfego de rede, como por exemplo, a falha de um roteador, ele também

pode ser utilizado para testar a Internet. O ICMP está presente na camada de rede, e utiliza mensagens para reportar os erros. Existem algumas dezenas de mensagens ICMP definidas, as mais importantes estão listadas na **Tabela 5.1** (TANENBAUM, 2003). Cada tipo de mensagem ICMP é encapsulada em um pacote IP. O cabeçalho do protocolo ICMP é representado pela **Figura 5.1**.

Tabela 5.1: Principais mensagens do protocolo ICMP (TANENBAUM, 2003, pg. 346).

Tipo	Nome	Descrição
0	Echo reply	Responde se está ativa
3	Destination unreachable	Não foi possível entregar o pacote
8	Echo request	Pergunta a uma máquina se ela está ativa
11	Time exceeded	Tempo de vida do pacote (TTL) acabou
12	Parameter problem	Campo de cabeçalho inválido
13	Timestamp request	Similar a Echo request, mas com timbre de hora
14	Timestamp reply	Similar a Echo reply, mas com timbre de hora

De acordo com Tanenbaum (2003):

- As mensagens *echo request* e *echo reply* são utilizadas em conjunto para verificar se um destino está ativo e acessível;
- A mensagem *destination unreachable* normalmente é utilizada quando a sub rede ou um roteador não consegue localizar o destino;
- A mensagem t*ime exceeded* é enviada quando um pacote é descartado porque seu contador foi decrementado até zero;
- A mensagem *parameter problem* indica que um valor inválido foi detectado em um campo de cabeçalho;
- As mensagens *timestamp request* e *timestamp reply* são semelhantes as mensagens *echo request* e *echo reply*, respectivamente, exceto pelo fato do tempo de chegada da mensagem e de saída da resposta serem registrados na mensagem de resposta.

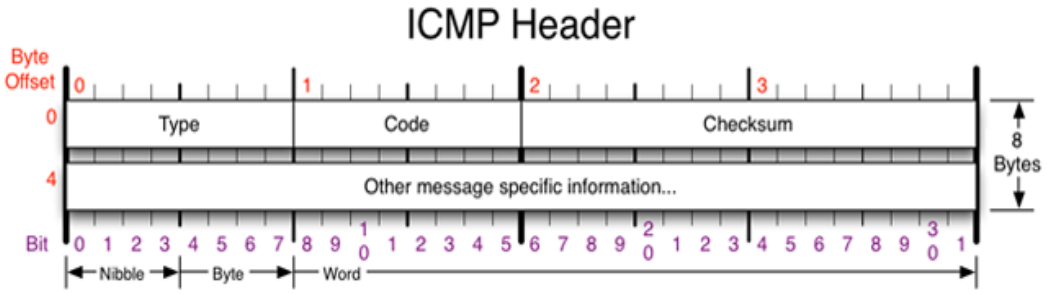

Figura 5.1: Cabeçalho do protocolo ICMP (POSTEL, 2013).

O campo *Type* contém o tipo da mensagem ICMP, para o protótipo interessará somente as mensagens com o tipo 0, referindo que se trata de uma mensagem *echo reply*. Os demais campos do cabeçalho não são utilizados no protótipo neste instante, porém, podem ser úteis em futuras extensões do protótipo, auxiliando na detecção de outros tipos de ataques.

5.2 O Funcionamento do PDADoS

O PDADoS tem o objetivo de realizar a detecção de um caso específico de ataque de negação de serviço. Apesar da detecção do ataque ser apenas uma parte do problema causado por estes ataques, espera-se contribuir com o avanço na metodologia de novas ferramentas que venham a mitigar os ataques de negação de serviço. Atualmente, muitas ferramentas para detecção de ataques de negação de serviço estão disponíveis aos usuários, porém, ainda não é um assunto encerrado, já que novas formas de ataques ainda conseguem violar a segurança desses sistemas. A grande dificuldade em detectar ataques de negação de serviço está no fato deles serem semelhantes a pacotes legítimos, tornando difícil distinguir um tráfego de ataque de um legítimo. A abordagem com base em anomalias no tráfego de rede para detecção de ataques de negação de serviço, foi a adotada no PDADoS.

Nesta seção é explicado o funcionamento do protótipo desenvolvido, apresentando pseudocódigos dos procedimentos responsáveis pela detecção do ataque de negação de serviço em tempo real, especificando as entradas necessárias para execução do protótipo e quais são as saídas geradas por ele.

5.2.1 Principais procedimentos

Um ataque de negação de serviço no tráfego de rede, consiste no envio intencional de centenas, ou milhares de pacotes simultaneamente com o destino em comum. Essa prática tem

o interesse de causar um acumulo no tráfego de rede do alvo, que em determinado instante seus recursos de processamento, ou armazenamento esgotam, causando o congelamento dos serviços oferecidos por ele, ocasionando a negação de serviço.

O PDADoS é dividido em dois programas, o primeiro para realizar a modelagem do tráfego de rede normal de pacotes ICMP *echo reply*, que definirá o limite máximo destes pacotes em um tráfego de rede que pode ser considerado normal. Para realizar a modelagem, a primeira semana de dados da base de dados DARPA é monitorada (fase de treinamento). O segundo programa, é responsável pela detecção de ataques de negação de serviço por inundação com pacotes ICMP *echo reply* (fase de detecção). A **Figura 5.2** mostra o funcionamento do protótipo, onde os dois programas têm o funcionamento parecido.

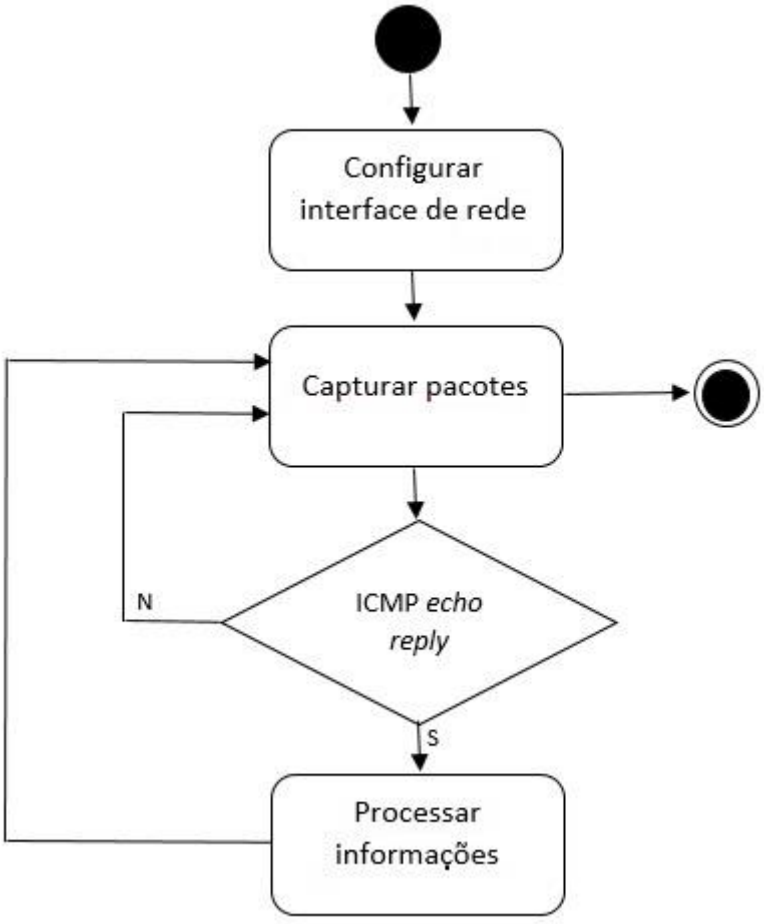

Figura 5.2: Diagrama de execução do PDADoS.

Os programas iniciam configurando o dispositivo de captura de pacotes na interface de rede informada, cujo funcionamento é similar ao *sniffer* (TCPDUMP, 2013). Algumas funções da biblioteca **pcap** são utilizadas para realizar a configuração, são elas:

- Atribuir ao dispositivo de captura o nome informado: **dev = argv[1];**
- Pegar o número de rede e a máscara associada: **pcap_lookupnet(dev, &net, &mask, errbuf);**
- Certificar que está sendo capturado em um dispositivo *Ethernet*: **pcap_datalink(handle);**
- Abertura do dispositivo de captura: **handle = pcap_open_offline(dev, errbuf);**
- Compilar a expressão de filtragem: **pcap_compile(handle, &fp, filter_exp, 0, net);**
- Aplicar o filtro: **pcap_setfilter(handle, &fp);**
- Iniciar o *loop* de captura: **pcap_loop(handle, num_packets, got_packet, NULL).**

Os pacotes são capturados e para cada pacote capturado o procedimento got_packet() é invocado. Sua função é verificar se o pacote capturado utiliza o protocolo ICMP, caso for verdade, o procedimento analysisPacketICMP**()** é invocado, senão retorna ao *loop* de captura de pacotes à espera do próximo. A **Figura 5.3** mostra como os pacotes capturados são dispostos.

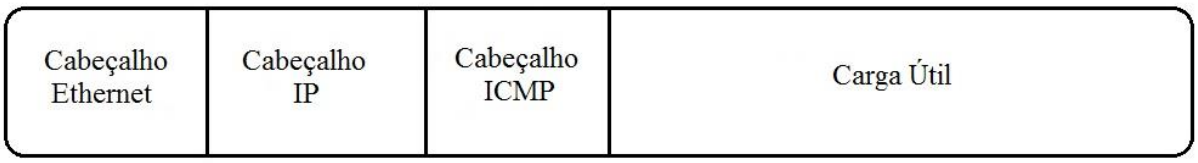

Figura 5.3: Formato dos pacotes capturados pelo protótipo.

O protocolo ICMP tem dezenas de mensagens definidas, para o protótipo interessa somente as mensagens do tipo *echo reply*. O procedimento analysisPacketICMP() realiza essa verificação, e se a mensagem for do tipo *echo reply* (campo *Type* do cabeçalho ICMP for igual a zero), então o procedimento SMURF() é invocado, senão retorna ao *loop* de captura de pacotes. O procedimento SMURF() tem esse nome, por causa do famoso ataque de negação de serviço já citado nas seções anteriores que utilizam mensagens do tipo *echo reply*, o mesmo tipo que está sendo monitorada, para inundar seu alvo. Nesse procedimento, há diferença de código entre os programas e cada caso é analisado separadamente.

O protótipo armazena as informações em uma lista simplesmente encadeada, onde cada nó contém as seguintes informações: endereço IP de destino, o horário do primeiro pacote capturado, o horário do último pacote capturado e a quantidade de pacotes.

É importante lembrar que um número elevado de pacotes encaminhados para um mesmo endereço IP de destino não se caracteriza, por si só, como um ataque. Se milhões de pacotes forem enviados para um mesmo destino dentro de um grande intervalo de tempo, o destino processará todos eles e responderá sem problemas. Agora, o envio de muitos pacotes em um pequeno intervalo de tempo poderá causar danos ao destino. Por esse motivo, foi escolhida uma estratégia de detecção que utiliza janelas de tempo fixo para observar o tráfego recebido por cada máquina na rede. Portanto, para cada pacote capturado, é verificado se o endereço IP de destino do pacote é o mesmo de outros pacotes já capturados. Caso seja, o tempo de captura do pacote é comparado com o tempo de captura do último pacote capturado para o mesmo endereço IP de destino, se a diferença de tempo entre eles for menor que o tamanho da janela, significa que o pacote chego dentro de um intervalo de tempo pequeno, o que pode caracterizar um ataque de negação de serviço. Se a diferença de tempo de captura entre os pacotes for superior ao da janela, então é inicializada uma nova janela para este pacote. Caso o endereço IP de destino não seja o mesmo para os demais pacotes capturados, então uma nova janela é inicializada para este pacote. A **Figura 5.4** apresenta o pseudocódigo para fase de treinamento do PDADoS e a **Figura 5.5** traz o pseudocódigo para a fase de detecção do PDADoS.

```
1   Procedimento SMURF()
2   Início
3       Se for primeiro pacote capturado Então
4           Armazene as informações na lista e no arquivo de saída
5       Senão
6           Ponteiro := buscaTabSMURF()
7           Se Ponteiro ≠ NULL Então
8               delta_t := tempo_atual – ponteiro->tempo_ultimo
9               Se delta_t < janela Então
10                  Incremente quantidade de pacotes
11                  Atualize tempo do último
12                  Armazene informações no arquivo de saída
13              Senão
14                  Reinicie quantidade de pacotes
15                  Atualize tempo primeiro e último
16                  Armazene informações no arquivo de saída
17              Fim-se
18          Senão
19              Armazene as informações na lista e no arquivo de saída
20          Fim-se
21      Fim-se
22  Fim
```

Figura 5.4: Procedimento SMURF para treinamento do protótipo.

Analisando o pseudocódigo, as linhas 3 e 4 verificam se o pacote capturado é o primeiro, caso seja, a lista é inicializada com as informações e depois escritas no arquivo de saída. Caso não seja o primeiro, a linha 6 chama a função buscaTabSMURF(), que busca na lista se o endereço IP de destino do pacote atual já está cadastrado na lista. Se encontrar, a busca retorna o ponteiro que aponta para o nó com o mesmo endereço IP de destino e continua a execução nas linhas 8 à 17. Senão, significa que é um novo endereço IP de destino, então a busca retorna NULL e a execução salta para a linha 19. As linhas 8 à 17, comparam a janela com a diferença do horário de captura do pacote atual em relação ao horário do último pacote capturado para um mesmo endereço IP de destino. Se a diferença for inferior a janela, então a quantidade de pacotes é acrescida, o horário do último pacote é atualizado e as informações são escritas no arquivo de saída. Senão, a quantidade de pacotes é reiniciada com o valor 1, o horário do primeiro e do último recebem o mesmo valor e as informações são escritas no arquivo de saída. A linha 19 aloca um novo nó na lista para salvar as informações do pacote, depois as escreve no arquivo de saída.

Depois do procedimento SMURF(), o protótipo retorna ao *loop* de captura de pacotes e a execução do protótipo termina quando todo o tráfego do arquivo de entrada foi analisado.

```
1   Procedimento SMURF()
2   Início
3       Se for primeiro pacote capturado Então
4           Armazene as informações na lista e no arquivo de saída
5       Senão
6           Ponteiro := buscaTabSMURF()
7           Se Ponteiro ≠ NULL Então
8               delta_t := tempo_atual – ponteiro->tempo_ultimo
9               Se delta_t < janela Então
10                  Incremente quantidade de pacotes
11                  Atualize tempo do último
12                  Armazene informações no arquivo de saída
13                  Se Ponteiro->quantidade > MAX Então
14                      Alerta()
15                  Fim-se
16              Senão
17                  Reinicie quantidade de pacotes
18                  Atualize tempo primeiro e último
19                  Armazene informações no arquivo de saída
20              Fim-se
21          Senão
22              Armazene as informações na lista e no arquivo de saída
23          Fim-se
24      Fim-se
25  Fim
```

Figura 5.5: Procedimento SMURF para detecção de ataques.

A execução do procedimento SMURF() no programa que realiza a detecção é similar ao mesmo procedimento utilizado na fase de treinamento, com exceção das linhas 13 e 14. São detectados os ataques contendo a quantidade de pacotes em uma mesma janela de observação com o mesmo endereço IP de destino superior à média aritmética calculada com base na amostra com o maior índice de pacotes ICMP *echo reply* gerada na fase de treinamento acrescida de 3 vezes o valor do desvio padrão (DUCLOS, 1997). Após a detecção, o método alarmeDeteccaoSMURF() é invocado e escreve na tela informações relativas ao ataque alertando o usuário, além de criar um arquivo de saída com essas informações, para serem analisadas posteriormente.

5.2.2 Ambiente de detecção

O PDADoS foi compilado e executado no sistema operacional Linux Ubuntu 11.04. A linguagem de programação utilizada é C e compilador GCC.

A maioria das bibliotecas necessárias para o funcionamento do protótipo já estão instaladas por padrão no Linux, mas a biblioteca **pcap**, que é essencial para captura dos pacotes no tráfego de rede, normalmente não está instalada no sistema operacional. Portanto, para instalar a biblioteca **pcap**, existem duas formas: Através do comando *apt-get install*, ou encontrar o repositório da biblioteca na Internet e baixar os arquivos manualmente. Em ambos os casos, existem tutoriais em *sites* de pesquisa que orientam a instalação (TCPDUMP, 2013).

Com as bibliotecas instaladas no sistema operacional, para o sistema executar, basta abrir o terminal no diretório que se encontra o protótipo e compilar o código com a linha de comando (1) e depois executá-lo como administrador, informando a interface de rede que pretende monitorar, no nosso caso, os arquivos DARPA (2):

(1) `gcc -Wall -o PDADoS PDADos.c –lpcap`

(2) `sudo ./PDADoS arquivo_no_formato_tcpdump.tcpdump`

5.3 Base de Dados DARPA

Os experimentos realizados para avaliar o protótipo de detecção proposto, utilizou a base de dados DARPA (1999) para simular um tráfego de rede (DARPA, 1999). O *data set* (conjunto de dados) DARPA foi gerado pelo laboratório Lincoln do MIT (*Massachussetts*

Institute of Technology) junto com a DARPA (*Defense Advanced Research Projects Agency*) e contém 5 semanas de tráfego de rede. A primeira e a terceira semana não constam nenhum tipo de ataque, são úteis para o treinamento de sistemas de detecção com base nas anomalias do tráfego. A segunda semana contém um seleto subconjunto de ataques de negação de serviço conhecidos, alguns deles citados na **seção 3.3**. As duas últimas semanas contém inúmeros ataques não identificados e não documentados, o que dificulta a comparação e avaliação da eficácia do protótipo proposto. Todas as semanas consideram 5 dias de tráfego registrados, considerando cada dia com início às 8 horas da manhã e término às 6 horas da manhã do outro dia, totalizando 22 horas de tráfego por dia (DARPA, 1999).

O estudo utilizou a primeira semana de dados para realizar o treinamento do PDADoS, gerando uma amostra considerada como o tráfego normal de pacotes ICMP *echo reply*, para cada dia da semana. Para análise de eficácia do PDADoS, os cinco dias da segunda semana foram monitorados pelo protótipo.

5.4 Resultados e Discussões

Nesta seção, as amostras do tráfego de rede de pacotes ICMP *echo reply* geradas durante a semana de treinamento são estudadas para compreender o tráfego de rede que pode ser considerado como tráfego normal destes pacotes. As janelas de observação que não contiveram pacotes durante o treinamento, não foram contabilizadas. As que tiveram apenas 1 pacote, foram desconsideradas, já que interessa ao protótipo quando há maiores incidência desses pacotes. Após análise do protótipo durante a semana de treinamento, dados do tráfego de pacotes ICMP *echo reply* durante a semana com incidência de ataques, são apresentados e comparados com os dados da documentação DARPA, para julgar a eficiência do protótipo em detectar ataques de negação de serviço com o tráfego anormal de pacotes ICMP *echo reply*.

Semana de treinamento

A janela de observação está definida com 5 segundos, ou seja, caso o pacote atual seja capturado com o *timestamp* com mais de 5 segundos em relação ao último pacote capturado referente ao mesmo endereço IP de destino, então uma nova janela é estabelecida. Em futuros testes, outros valores para janela podem ser testados, afim de alcançar um valor ideal para a janela de observação. Para os testes realizados, o valor estabelecido mostrou ser um bom intervalo de tempo entre pacotes com o destino em comum.

A base de dados DARPA abordada na **seção 5.3**, é utilizada para simular o tráfego de rede a ser monitorado pelo PDADoS. Existem outras bases de dados que podem ser utilizadas para simular o tráfego de rede, porém, a maioria não são disponibilizadas. A forma mais eficaz de realizar o teste do protótipo, seria configurar um ataque de negação de serviço distribuído, e direcionar o tráfego de ataque para a rede que o protótipo esteja monitorando. No entanto, apenas a base de dados DARPA foi utilizada.

Para calcular a média a fórmula (1) foi utilizada e para o cálculo do desvio padrão a fórmula (2).

$$(1)\ \overline{x} = \left(\sum_{i=1}^{t} x_i\right) \div t$$

$$(2)\ \sigma = \sqrt[2]{\sum_{i=1}^{t} (x_i - \overline{x})^2 \div (t-1)}$$

Onde i é o índice da janela; t é o total de janelas; x é o número de pacotes por janela.

Segunda-feira

A **Tabela 5.2**, apresenta o tráfego de rede de pacotes ICMP *echo reply* do primeiro dia da primeira semana da base de dados DARPA, com o mesmo endereço IP de destino. O tráfego de rede é dividido por janelas de observação, as janelas com a quantidade de pacotes superior a um estão presentes nesta amostra.

Tabela 5.2: Amostra do tráfego de pacotes ICMP *echo reply* detectado pelo protótipo no primeiro dia de treinamento.

Índice	**Janela**	**Nº Pacotes**	**$(x_i - x)^2$**
1	J_1	7	2,621315193
2	J_3	7	2,621315193
3	J_4	4	1,907029478
4	J_{11}	10	21,33560091
5	J_{16}	5	0,145124717
6	J_{17}	9	13,09750567
7	J_{19}	5	0,145124717
8	J_{25}	2	11,430839
9	J_{31}	7	2,621315193
10	J_{32}	8	6,859410431
11	J_{41}	7	2,621315193
12	J_{46}	2	11,430839
13	J_{49}	5	0,145124717
14	J_{50}	2	11,430839
15	J_{53}	4	1,907029478
16	J_{54}	5	0,145124717
17	J_{56}	8	6,859410431
18	J_{64}	4	1,907029478
19	J_{84}	4	1,907029478
20	J_{88}	5	0,145124717
21	J_{100}	3	5,66893424
	$\overline{x}$ =	**5,380952381**	
	σ =	**2,312491956**	

Terça-feira

A **Tabela 5.3**, apresenta o tráfego de rede de pacotes ICMP *echo reply* do segundo dia da primeira semana da base de dados DARPA, com o mesmo endereço IP de destino. O tráfego de rede é dividido por janelas de observação, as janelas com a quantidade de pacotes superior a um estão presentes nesta amostra.

Tabela 5.3: Amostra do tráfego de pacotes ICMP *echo reply* detectado pelo protótipo no segundo dia de treinamento.

Índice	**Janela**	**Nº Pacotes**	$(x_i - x)^2$
1	J_1	4	1
2	J_7	5	0
3	J_9	5	0
4	J_{14}	9	16
5	J_{20}	5	0
6	J_{32}	6	1
7	J_{35}	4	1
8	J_{37}	2	9
9	J_{44}	6	1
10	J_{60}	4	1
	$\overline{x}$ =	**5**	
	σ =	**1,825741858**	

Quarta-feira

A **Tabela 5.4**, apresenta o tráfego de rede de pacotes ICMP *echo reply* do terceiro dia da primeira semana da base de dados DARPA, com o mesmo endereço IP de destino. O tráfego de rede é dividido por janelas de observação, as janelas com a quantidade de pacotes superior a um estão presentes nesta amostra.

Tabela 5.4: Amostra do tráfego de pacotes ICMP *echo reply* detectado pelo protótipo no terceiro dia de treinamento.

Índice	**Janela**	**Nº Pacotes**	**$(x_i - x)^2$**
1	J_5	5	4
2	J_{10}	6	1
3	J_{25}	9	4
4	J_{38}	5	4
5	J_{41}	7	0
6	J_{42}	4	9
7	J_{48}	10	9
8	J_{56}	13	36
9	J_{69}	10	9
10	J_{82}	10	9
11	J_{92}	10	9
12	J_{101}	9	4
13	J_{104}	3	16
14	J_{107}	3	16
15	J_{113}	6	1
16	J_{115}	3	16
17	J_{196}	6	1
	$\overline{x}$ =	**7**	
	σ =	**3,041381265**	

Quinta-feira

A **Tabela 5.5**, apresenta o tráfego de rede de pacotes ICMP *echo reply* do quarto dia da primeira semana da base de dados DARPA, com o mesmo endereço IP de destino. O tráfego de rede é dividido por janelas de observação, as janelas com a quantidade de pacotes superior a um estão presentes nesta amostra.

Tabela 5.5: Amostra do tráfego de pacotes ICMP *echo reply* detectado pelo protótipo no quarto dia de treinamento.

Índice	**Janela**	**Nº Pacotes**	**$(x_i - x)^2$**
1	J_1	4	2,777777778
2	J_7	7	1,777777778
3	J_{11}	6	0,111111111
4	J_{14}	6	0,111111111
5	J_{28}	10	18,77777778
6	J_{31}	6	0,111111111
7	J_{38}	9	11,11111111
8	J_{40}	2	13,44444444
9	J_{42}	4	2,777777778
10	J_{43}	2	13,44444444
11	J_{45}	9	11,11111111
12	J_{54}	2	13,44444444
13	J_{74}	9	11,11111111
14	J_{75}	8	5,444444444
15	J_{76}	4	2,777777778
16	J_{88}	9	11,11111111
17	J_{89}	2	13,44444444
18	J_{91}	4	2,777777778
19	J_{92}	5	0,444444444
20	J_{94}	5	0,444444444
21	J_{96}	4	2,777777778
22	J_{98}	5	0,444444444
23	J_{102}	8	5,444444444
24	J_{119}	6	0,111111111
	$\overline{x}$ =	**5,666666667**	
	σ =	**2,513730411**	

Sexta-feira

A **Tabela 5.6**, apresenta o tráfego de rede de pacotes ICMP *echo reply* do quinto dia da primeira semana da base de dados DARPA, com o mesmo endereço IP de destino. O tráfego de rede é dividido por janelas de observação, as janelas com a quantidade de pacotes superior a um estão presentes nesta amostra.

Tabela 5.6: Amostra do tráfego de pacotes ICMP *echo reply* detectado pelo protótipo no quinto dia de treinamento.

Índice	**Janela**	**Nº Pacotes**	**$(x_i - x)^2$**
1	J_3	2	14,44
2	J_9	6	0,04
3	J_{16}	8	4,84
4	J_{18}	7	1,44
5	J_{19}	2	14,44
6	J_{28}	6	0,04
7	J_{31}	10	17,64
8	J_{41}	5	0,64
9	J_{42}	4	3,24
10	J_{51}	2	14,44
11	J_{53}	4	3,24
12	J_{99}	10	17,64
13	J_{100}	7	1,44
14	J_{104}	9	10,24
15	J_{111}	5	0,64
	$\overline{x}$ =	**5,8**	
	σ =	**2,730776969**	

Nota-se que quarta-feira teve a maior média de pacotes ICMP *echo reply* em um tráfego de rede normal, portanto, a média e o desvio padrão deste dia são os utilizados para calcular o número máximo destes pacotes por janela de observação. Assim, o máximo será:

$$MAX = \overline{x} + (3 \times \sigma)$$

$$MAX \cong 16$$

Semana de ataque

Durante o monitoramento da segunda semana da base de dados DARPA, o PDADoS detectou 5 ataques de negação de serviço que utilizam pacotes ICMP *echo reply* para inundar o tráfego de rede do alvo, sendo que nenhum ataque deste tipo foi detectado na quarta-feira e onde dois ataques foram detectados na sexta-feira. A **Tabela 5.7** lista os 5 ataques detectados, informando o dia, o horário que foi capturado o primeiro pacote do tráfego de ataque (*timestamp*) e o número de pacotes ICMP *echo reply* presentes no tráfego de ataque de cada janela de observação. A **Tabela 5.8** lista todos os ataques presentes na segunda semana de tráfego da base de dados DARPA. Em negrito destacam-se os ataques de negação de serviço

que utilizam pacotes ICMP *echo reply* para realizar o ataque. Por fim, uma análise de eficiência do protótipo é realizada, com base nessas duas tabelas.

Tabela 5.7. Ataques detectados.

Índice	Data	Tempo (HH:MM:SS)	Nº Pacotes
1	08/03/1999	08:50:15	235
2	09/03/1999	08:44:17	1003
3	11/03/1999	10:50:11	1001
4	12/03/1999	09:18:15	423
5	12/03/1999	17:13:10	5001

Tabela 5.8. Listagem dos ataques presentes na segunda semana da base de dados DARPA (DARPA, 2013).

Índice	Data	Tempo (HH:MM:SS)	Nome
1	08/03/1999	08:01:01	Ntinfoscan
2	**08/03/1999**	**08:50:15**	**pod**
3	08/03/1999	09:39:16	back
4	08/03/1999	12:09:18	httptunnel
5	08/03/1999	15:57:15	land
6	08/03/1999	17:27:13	secret
7	08/03/1999	19:09:17	ps attack
8	**09/03/1999**	**08:44:17**	**portsweep**
9	09/03/1999	09:43:51	eject
10	09/03/1999	10:06:43	back
11	09/03/1999	10:54:19	loadmodule
12	09/03/1999	11:49:13	secret
13	09/03/1999	14:25:16	mailbomb
14	**09/03/1999**	**13:05:10**	**ipsweep**
15	09/03/1999	16:11:15	phf
16	09/03/1999	18:06:17	httptunnel
17	10/03/1999	12:02:13	satan
18	10/03/1999	13:44:18	mailbomb
19	10/03/1999	15:25:18	perl (*Falied*)
20	**10/03/1999**	**20:17:10**	**ipsweep**
21	10/03/1999	23:23:00	eject (*console*)
22	10/03/1999	23:56:14	crashiis
23	11/03/1999	08:04:17	crashiis
24	11/03/1999	09:33:17	satan
25	**11/03/1999**	**10:50:11**	**portsweep**
26	11/03/1999	11:04:16	neptune

27	11/03/1999	12:57:13	secret
28	11/03/1999	14:25:17	perl
29	11/03/1999	15:47:15	land
30	**11/03/1999**	**16:36:10**	**ipsweep**
31	11/03/1999	19:16:18	ftp-write
32	12/03/1999	08:07:17	phf
33	12/03/1999	08:10:40	perl (*console*)
34	12/03/1999	08:16:46	ps (*console*)
35	**12/03/1999**	**09:18:15**	**pod**
36	12/03/1999	11:20:15	neptune
37	12/03/1999	12:40:12	crashiis
38	12/03/1999	13:12:17	loadmodule
39	12/03/1999	14:06:17	perl (*Falied*)
40	12/03/1999	14:24:18	ps
41	12/03/1999	15:24:16	eject
42	**12/03/1999**	**17:13:10**	**portsweep**
43	12/03/1999	17:43:18	ftp-write

A segunda semana de tráfego contém 3 ataques de negação de serviço que exploram a vulnerabilidade do protocolo ICMP para atingir o alvo com mensagens *echo reply*, são eles: pod, *ipsweep* e *portsweep*.

Os ataques aconteceram mais de uma vez, e para os ataques pod e *portsweep* o PDADoS efetuou a detecção com eficiência no exato momento que tiveram início os ataques. Já o ataque *ipsweep* não foi detectado em nenhum momento pelo PDADoS, sendo um caso de falso negativo.

Após análise do arquivo com o histórico do tráfego de pacotes ICMP *echo reply* gerado pelo PDADoS, pode-se concluir que o ataque *ipsweep* não foi detectado, porque, o tráfego de pacotes ICMP *echo reply* gerado pelo ataque é inferior ao limite superior ao estabelecido para cada janela de observação, sendo que o objetivo deste ataque é apenas localizar os *hosts* que estão ativos na rede, ou sub rede, e não efetuar um ataque de negação de serviço por inundação. Portanto, provavelmente este ataque isoladamente dificilmente obteria êxito em inundar sua vítima somente com pacotes ICMP *echo reply*. A **Tabela 5.9** representa o comportamento do PDADoS após monitorar a segunda semana de dados DARPA.

Tabela 5.9: Análise de eficiência do protótipo

Nome	Quantidade	Detecção	Falso Negativo
pod	2	2	0
ipsweep	3	0	3
portsweep	3	3	0

Portanto, como é apresentado na **Tabela 5.9**, para os ataques que causam um comportamento anômalo no tráfego de rede, o PDADoS teve um resultado eficiente, detectando 100% dos ataques que enviaram muitos pacotes ICMP *echo reply* dentro de um pequeno intervalo de tempo e com o endereço de destino em comum, caracterizando um ataque de negação de serviço por inundação. A base de dados DARPA, proporcionou ao estudo, a realização de testes com ataques de negação de serviço não distribuídos, porém, para ataques distribuídos, o protótipo também tem a capacidade de realizar a detecção, apesar de não terem sido realizados testes para estes ataques neste estudo.

Capítulo 6

CONSIDERAÇÕES FINAIS

O projeto desenvolvido alcançou seu objetivo proposto, de desenvolver um protótipo para detecção de ataques de negação de serviço, revelando os desafios para lidar com esses ataques. O estudo histórico destes ataques, forneceu conhecimento de como surgiram e o rumo tomado por eles nos dias atuais, e a partir disto, foi possível analisar os principais procedimentos adotados pelos autores destes ataques, averiguar as suas diferentes formas de atacar e descobrir os meios dos quais eles tiram vantagem para obterem sucesso. As técnicas de detecção foram abordadas, e a estratégia de detecção escolhida para o protótipo foi a detecção baseada em anomalias no tráfego. Com o desenvolvimento do protótipo, foi possível analisar o comportamento normal e anormal do tráfego, tendo êxito na detecção em 62,5 % dos ataques alvos deste estudo. Considerando que o ataque que não foi capturado não é capaz de inundar a vítima apenas com pacotes ICMP *echo reply*, é possível dizer que o protótipo teve nível de detecção satisfatório.

Portanto, o projeto proporcionou o conhecimento do funcionamento dos ataques de negação de serviço e possibilitou o desenvolvimento de um protótipo adequado para detecção de ataques de negação de serviço por inundação com pacotes ICMP *echo reply*.

6.1 Dificuldades Encontradas

Durante a execução deste estudo, apenas a base de dados DARPA é utilizada para simular o tráfego de rede, a dificuldade de utilizar outras base de dados e de simular com máquinas reais o tráfego de rede dos ataques de negação de serviço, são os fatores que dificultaram o estudo.

6.2 Trabalhos Futuros

Os ataques de negação de serviço ainda são um problema sem uma solução eficiente. Diversos estudos acadêmicos são voltados a este tema, que apesar de complexo se mostra muito interessante e desafiador. O protótipo desenvolvido conta com um padrão de detecção, para trabalhos futuros, novos padrões podem ser incrementados e testados, com o objetivo de detectar outros ataques que não foram alvo neste estudo. O estudo adotou que o tráfego de rede

analisado, tem um comportamento não normal, e por esse motivo adotou um limite superior com base em cálculos presentes em outros estudos. Uma contribuição para o presente estudo, é realizar uma análise estatística do comportamento do tráfego de rede, para estabelecer um limite superior restrito.

REFERÊNCIAS

ABLIZ, Mehmud. **Internet Denial of Service Attacks and Defense Mechanisms**. Department of Computer Science, University of Pittsburgh. 2011.

AZEVEDO, Renato P. **Detecção de ataques de negação de serviço em redes de computadores através da transformada Wavelet 2D**. Universidade Federal de Santa Maria. Santa Maria. 2012.

BISHOP, Matt. **Introduction to Computer Security**. 1st ed., Prentice Hall PTR, 2004.

DARPA. **Cyber Systems and Technology**. MIT Lincoln Laboratory. Disponível em: http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/. Data de acesso: 10/10/2013.

DUCLOS, Emmanuel; PILLET, Maurice. **Contribution à la Maîtrise Statistique des Precédés, Cas des Procédés Non Normaux**. PhD, Université de Savoie. 1997.

SILVA, Nícolas R. **Avaliação da sensibilidade de métricas para a detecção de ataques de inundação**. Instituto Militar De Engenharia. Rio de Janeiro. 2012.

GARBER, Lee. **Denial-of-Service Attacks Rip the Internet**. IEEE Computer. 2000.

LAUFER, Rafael P.; MORAES, Igor M.; VELLOSO, Pedro B.; BICUDO, Marco D. D.; CAMPISTA, Miguel Elias M.; CUNHA, Daniel de O.; COSTA, Luís Henrique M. K.; DUARTE, Otto Carlos M. B. **Negação de Serviço: Ataques e Contramedidas**. 2005.

LEHTINEN, Rick. **Computer Security Basics**. 2nd ed., O'Reilly, 2006.

LOBO, Ana Paula. **Convergência Digital: Grupo Anonymous Ataca Banco Central**. Disponível em: http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=29125&sid=18. Data de acesso: 19/10/2012.

MARTINS, Daniel M. **Uma estratégia para sistemas de detecção e prevenção de intrusão baseada em software livre**. Universidade Federal do Ceará. 2012.

MUZZI, Fernando A. G. **Análise de botnet utilizando plataforma de simulação com máquinas virtuais visando detecção e contenção**. Escola Politécnica da Universidade de São Paulo. 2010.

Offensive Security. **Exploit database**. Disponível em: http://www.exploit-db.com. Data de acesso: 10/09/2012.

PIVOTTO, Carlos V. C.; PIMENTA, Luís Carlos de S. **Denial of Service - Negação de Serviço**. 2006. Disponível em: http://www.gta.ufrj.br/grad/06_1/dos/index.html. Data de acesso: 21/02/2013.

POSTEL, Jonathan B. **Internet control message protocol – RFC 792**. Disponível em: http://tools.ietf.org/html/rfc792. Data de acesso: 30/09/2013.

SANTOS, Anderson F. P. **Identificação e Análise de Comportamentos Anômalos**. Laboratório Nacional de Computação Científica. 2009.

SANTOS, Wellington H. dos. **Método de detecção de ataques ddos compostos baseado em filtragem sequencial**. Instituto Militar De Engenharia. 2012.

SILVA, Lília de S. **Uma metodologia para a detecção de ataques no tráfego de redes baseada em redes neurais**. Instituto Nacional de Pesquisas Espaciais. 2007.

SOLHA, Liliana E. V. A.; TEIXEIRA, Renata C.; PICCOLINI, Jacomo D. B. **Tudo que você precisa saber sobre os ataques DDoS**. Disponível em: http://www.rnp.br/newsgen/0003/ddos.html. Data de acesso: 26/04/2013.

SOCKRIDER, Gary. **Crescimento dos ataques DDoS exigem nova estratégia de segurança**. Disponível em: http://cio.uol.com.br/tecnologia/2012/12/10/crescimento-dos-ataques-ddos-exigem-nova-estrategia-de-seguranca/. Data de acesso: 22/02/2013.

TANENBAUM, A. S. **Redes de Computadores**. 4ª Edição. Campus (Elsevier). 2003.

TCPDUMP. Disponível em: http://www.tcpdump.org. Data de acesso: 03/10/2013.

WANG, Jie. **Computer Network Security: Theory and Practice**. 1st ed., Higher Education Press, 2009.