UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

UM ANALISADOR DE INTRUSÕES BASEADO EM SÉRIES TEMPORAIS

TRABALHO DE GRADUAÇÃO

Roben Castagna Lunardi

Santa Maria, RS, Brasil

2008

UM ANALISADOR DE INTRUSÕES BASEADO EM SÉRIES TEMPORAIS

Por

Roben Castagna Lunardi

Trabalho de Graduação apresentado ao Curso de Graduação em Ciência da Computação – Bacharelado, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de

Bacharel em Ciência da Computação

Curso de Ciência da Computação

Trabalho de Graduação n° 255

Santa Maria, RS, Brasil

2008

Universidade Federal de Santa Maria

Centro de Tecnologia

Curso de Ciência da Computação

A Comissão Examinadora, abaixo assinada, aprova o

Trabalho de Graduação

UM ANALISADOR DE INTRUSÕES BASEADO EM SÉRIES TEMPORAIS

Elaborado por

Roben Castagna Lunardi

como requisito parcial para obtenção do grau de

Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA

_____

Prof. Dr. Raul Ceretta Nunes
(Orientador)

_____

Profa. Dra. Roseclea Duarte Medina

_____

Prof. Antonio Marcos de Oliveira Candia

Santa Maria, 01 de Fevereiro de 2008.

“Este trabalho é dedicado a tudo e a todos que passaram pela minha vida, pois estes são os responsáveis pelo que sou hoje”

Agradecimentos

Primeiramente agradeço aos meus pais e irmãos, sem os quais não teria a oportunidade de estar concluindo, por diversos motivos, a graduação em Ciência da Computação. Agradeço aos meus avós, que com sua experiência, me mostraram como viver dignamente.

Após estes, mas não menos importantes, agradeço ao apoio dado por todo integrantes e ex-colegas do CRSPE/INPE – MCT, em especial: ao Dr. Nelson Jorge Schuch por toda a orientação a nível pessoal e profissional; ao Koiti Ozaki pela orientação técnica e experiência de vida; ao ex-integrante, atual Doutorando da UNICAMP, Rafael Krummenauer pela amizade e suporte em momentos difíceis; aos ex-colegas de laboratório Érico Marcello Hoff do Amaral, Luciano Guilherme Machado e Sandro Lemos Oliveira pelos ótimos dias de amizade, trabalho e resolução de problemas juntos. Agradeço também aos colegas do Curso de Ciência da Computação, em especial ao Daniel Michelon de Carli, pela amizade, camaradagem, noites de estudos e trabalhos juntos, e pelo apoio sempre que necessário.

Aos irmãos da "The Bozers", que por muitas vezes me ajudaram em momentos difíceis, e também nos bons momentos, como por exemplo nos diversos Shows do CT.

A todos os amigos que me apoiaram na minha trajetória, e sem esquecer de agradecer à Aline Ibaldo Gonçalves, força motivadora no último ano de graduação.

Com carinho especial agradeço ao Orientador deste trabalho Raul Ceretta Nunes, que apesar de pouco tempo trabalhando juntos, foi uma pessoa de extrema importância tanto profissionalmente quanto pessoalmente.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS 9

LISTA DE FIGURAS 10

RESUMO 11

ABSTRACT 12

1. INTRODUÇÃO 13

1.1. RECURSOS UTILIZADOS 14

1.2. ESTADO ATUAL DOS DETECTORES DE INTRUSÕES 14

2. DEFINIÇÕES DE ATAQUES 16

2.1. GRAU DE AUTOMATICIDADE 16
2.1.1. MANUAL 16
2.1.2. SEMI-AUTOMÁTICO 16
2.1.3. AUTOMÁTICO 17

2.2. POSSIBILIDADE DE CARACTERIZAÇÃO 17
2.2.1. CARACTERIZÁVEIS 17
2.2.2. NÃO-CARACTERIZÁVEIS 18

2.3. EXPLORAÇÃO DE FRAGILIDADES 18
2.3.1. EXPLORAÇÃO DO PROTOCOLO IP 18
2.3.2. EXPLORAÇÃO DE SERVIÇOS BÁSICOS 20
2.3.3. NEGAÇÃO DE SERVIÇOS 20

2.4. VÍTIMAS 21
2.4.1. APLICAÇÃO 21
2.4.2. MÁQUINA 22
2.4.3. RECURSO 22
2.4.4. REDE 22
2.4.5. INFRA-ESTRUTURA 22

2.5. FORMAS DE DETECÇÃO 23
2.5.1. ANOMALIAS 23
2.5.2. ASSINATURAS 24

2.6. RESUMO DO CAPÍTULO 24

3. SÉREIS TEMPORAIS 25

3.1. CARACTERÍSTICAS 25

3.2. MODELO ARIMA 25

3.3. RESUMO DO CAPÍTULO 27

4. DETECÇÃO DE ATAQUES 28

4.1. ATAQUES A SEREM ABORDADOS 28
4.1.1. SYN ATTACK 28
4.1.1.1. CARACTERÍTICAS 29
4.1.1.2. ATAQUE 29
4.1.1.3. DETECÇÃO 31
4.1.2. SMURF/FRAGGLE ATACK 32
4.1.2.1. CARACTERÍTICAS 32
4.1.2.2. ATAQUE 32
4.1.2.3. DETECÇÃO 33

4.2. RESUMO DO CAPÍTULO 35

5. IMPLEMENTAÇÃO 36

5.1. PROGRAMAS DE CAPTURA DE DADOS 36
5.1.1. IAS 36
5.1.2. NTOP 38

5.2. SIMULAÇÃO DE ATAQUES 39
5.2.1. PROGRAMAS DE ATAQUE 40
5.2.1.1. HPING 40
5.2.1.2. PACKETH 40
5.2.2. RESULTADO DOS ATAQUES 42

5.3. USO DE SÉRIES TEMPORAIS 43
5.3.1. ARQUITETURA DO DIBSET 44
5.3.2. NÍVEIS DE ALARME 45
5.3.3. RESULTADOS 49

5.4. RESUMO DO CAPÍTULO 50

6. CONCLUSÕES 51

6.1. TRABALHOS FUTUROS 51

REFERÊNCIA BIBLIOGRÁFICA 53

LISTA DE ABREVIATURAS E SIGLAS

ACK	*Acknowledgment*
ARP	*Address Resolution Protocol*
DHCP	*Dynamic Host Configuration Protocol*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
IEEE	*Institute of Electrical and Electronics Engineers*
IP	*Internet Protocol*
MAC	*Media Access Control*
RST	*Reset*
SYN	*Synchronization*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*

LISTA DE FIGURAS

Figura 1: Land Attack.....19
Figura 2: Diversas máquinas enviando pacotes para vítima.....21
Figura 3: Diversas máquinas atacando um roteador.....23
Figura 4: Amostras com comportamento similar no tempo.....26
Figura 5: Início de conexão TCP.....29
Figura 6: SYN Attack.....30
Figura 7: SYN Attack, com redirecionamento de resposta.....30
Figura 8: Smurf/Fraggle Attack.....33
Figura 9: Contadores do IAS 37
Figura 10: Tráfego dos protocolos de transporte.....37
Figura 11: Gráfico gerado pelo NTOP 39
Figura 12: Imagem da interface do packETH.....41
Figura 13: PackETH e opções de quantidade e atraso de pacotes.....41
Figura 14: Comparação de Smurf Attack gerado.....42
Figura 15: Comparação de SYN Attack gerado.....43
Figura 16: Diagrama do DIBSeT.....45
Figura 17: Gráfico de Bytes ICMP.....46
Figura 18: Gráfico de Níveis de Alarme ICMP.....46
Figura 19: Gráfico de Pacotes SYN.....47
Figura 20: Gráfico de Níveis de Alarme SYN.....47
Figura 21: Gráfico de pacotes ICMP.....48
Figura 22: Níveis de Alarme ICMP.....48
Figura 23: Gráfico de pacotes SYN.....49
Figura 24: Níveis de Alarme SYN.....49

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

UM ANALISADOR DE INTRUSÕES BASEADO EM SÉRIES TEMPOREAIS

Aluno: Roben Castagna Lunardi
Orientador: Prof. Dr. Raul Ceretta Nunes

Com a convergência dos mais diversos serviços utilizando a Internet, houve evoluções nos métodos de conseguir, de forma ilegal, benefícios dos computadores em rede. Desta forma, tornaram-se necessários sistemas de barreiras ao acesso indevido. Mesmo configurando métodos para inibir a ação de usuários maliciosos, a todo instante estes acham brechas nos sistemas que permitem de alguma forma prejudicar a segurança dos dados. A segurança é baseada em um tripé: confidencialidade, integridade e disponibilidade, sendo este último muito esquecido pela maioria dos gerentes de redes de computadores.

Desta forma, o trabalho se concentrou basicamente em identificar possíveis ataques a disponibilidade dos sistemas. Para isso foi necessário o estabelecimento de quais ataques foram abrangidos; quais as características principais de cada ataque; como pôde ser detectado um ataque (anomalia) via análise por séries temporais; para enfim ser criado o programa DIBSeT – Detector de Intrusões Baseado em Séries temporais - que identifica possíveis ataques de sistemas em rede.

Palavras Chave: detecção de intrusões, segurança, gerência de redes de computadores.

ABSTRACT

Undergraduate Final Work
Computer Science
Federal University of Santa Maria

A TIME SERIES BASED INTRUSION ANALYZER

Author: Roben Castagna Lunardi
Advisor: Prof. Dr. Raul Ceretta Nunes

In the same time that Internet grows and many services had converged, some algorithms – with objective to capture confidential information in the computers network – had evolved. Therefore, some methods became essential to deny the illegal access. While some methods were configured to interrupt malicious users, many intruders find ways to break the security of systems. The security is based on three things: confidentiality, integrity and disponibility, and many times the network managers forgot the last of this three.

Because that, this work was focused to find disponibility attack attempts. To do that was established: the attacks; the characteristics of witch attack; how the attack could be detected using Temporal Series; and finally, the development of a software called DIBSeT – Temporal Series Based Intrusion Detector – that identify network attack attempts.

Keywords: intrusion detection, security, computer networks management.

1. INTRODUÇÃO

Este trabalho foi motivado pela parceria entre a Universidade Federal de Santa Maria (UFSM) e a Fachhochschule Gelsenkirchen (FHGe). A FHGe produziu um software para coleta de dados e armazenamento em contadores chamado Internet Analysis System - IAS (POHLMANN; PROEST, 2006). Este software armazena em contadores a quantidade de pacotes de conexão trafegados na rede.

O IAS da FHGe é usado nesta mesma Universidade como base para um sistema de detecção de intrusão utilizando redes neurais. Entretanto, a experiência prévia do orientador aponta que pode-se explorar detecção de intrusões baseado em contadores utilizando Séries Temporais (NUNES, 2003). Assunto este explorado neste trabalho.

A idéia de usar Séries Temporais, um tanto quanto inovadora, traz uma nova proposta, pois esta é muito menos custosa a nível computacional. Portanto, utilizando Séries Temporais podem ser feitas curvas de normalidade da rede, pois sabe-se que o tráfego de redes oscila, e picos são muitas vezes considerados normais. Desta forma, estabelecer limites estáticos produziria muitos falsos positivos. Podem ser estabelecidos limites dinâmicos, para que, conforme uma curva passada possa ser estabelecido um limite, e prever como deve ser o comportamento futuro.

Pode se concluir então que qualquer pacote analisado que ultrapasse um *threshold* estabelecido, em relação à curva criada através de análises passadas de normalidade, é um possível ataque.

Com o objetivo de tornar o programa utilizável nas mais diversas plataformas, utilizou-se a linguagem de programação JAVA, principalmente pelas suas características de portabilidade.

1.1. RECURSOS UTILIZADOS

Para realizar este trabalho foram utilizadas as instalações do CRS/INPE – MCT, que possui parceria com UFSM através do LACESM/CT/UFSM e com o grupo de pesquisa GMICRO. O prédio sede do CRS/INPE – MCT possui uma rede estruturada que segue o padrão EIA/TIA 568a (TIA/EIA).

Foram utilizados computadores do GMICRO para realizar a captura e análise dos dados. Para efetuar e simular os ataques foram utilizadas diversas ferramentas disponíveis gratuitamente que serão citadas no decorrer deste trabalho.

1.2. ESTADO ATUAL DOS DETECTORES DE INTRUSÕES

Atualmente existem diversos sistemas de detecção de intrusões, mas que não possuem características que agradem a todos os públicos. Um exemplo destes sistemas é o SNORT, um dos mais conhecidos detectores de intrusões, que apresenta bom funcionamento somente em redes pequenas (SNORT, 2007).

As principais pesquisas atuais concentram-se na busca por assinaturas de ataques, já que a detecção por anomalias de tráfego conhecida atualmente, gera muitos falsos positivos.

Como hoje em dia a diversidade de ataques é muito grande, e cresce o número de diferentes tipos de ataques, principalmente ataques distribuídos (KOMPELLA, et al., 2007), torna-se necessário um estudo constante de cada ataque que está sendo usado (LEVCHENKO, et al., 2004).

Com o objetivo de criar uma detecção por anomalias baseada em um comportamento histórico dos dados obtidos, tornando-a assim mais eficaz, este trabalho tem por objetivo introduzir séries temporais em detecção de intrusões. Para isso veremos, passo a passo, no capítulo 2 as características de um ataque, que brechas são utilizadas, o que é atacado e as formas de detectá-lo; no capítulo

3 a definição de séries temporais e o modelo adotado para realizarmos a detecção; no capítulo 4 quais ataques serão utilizados para testar o detector; e finalmente no capítulo 5 o atual estado de implementação e os programas utilizados.

2. DEFINIÇÃO DE ATAQUES

Para o melhor entendimento do que será abordado, iremos descrever neste capítulo importantes nomenclaturas usadas para definir e classificar os ataques.

2.1. GRAU DE AUTOMATICIDADE

Aquele que pretende atacar um sistema pode fazê-lo manualmente, ou de algumas formas mais automatizadas. Essas diferenças geralmente estão ligadas de alguma forma, com a maneira de como será o ataque.

2.1.1. MANUAL

Em um ataque manual, o invasor procura por máquinas e pelas respectivas fragilidades manualmente. Após achar falhas de segurança, dispara o ataque (MIRKOVIC; REIHER, 2004).

2.1.2. SEMI-AUTOMÁTICO

Em um ataque semi-automático, o atacante escolhe manualmente os tipos de ataques que serão disparados, que máquinas serão atacadas e quanto tempo irão durar os ataques. A parte de varredura, infecção e ataque são feitos automaticamente (MIRKOVIC; REIHER, 2004).

Em ataques distribuídos, o atacante infecta máquinas, chamadas de zumbis, que irão atacar automaticamente outras máquinas. Essa comunicação entre atacantes e máquinas zumbis pode ser:

- Direta: na comunicação direta tanto o atacante quanto a zumbi precisam conhecer uma a outra. A máquina atacante conhece o endereço IP da atacada e armazena durante a infecção seu endereço na máquina atacada, tornando possível assim uma comunicação posterior. Mas usando esse sistema, é possível identificar essa comunicação por programas de captura de dados.

- Indireta: na comunicação indireta, o atacante usa uma comunicação legítima para efetuar seu ataque. Canais IRC (*Internet Chat Program*) são muito utilizados, desde que possibilitem aos atacantes ficar no anonimato, para contaminar máquinas e utilizá-las para disseminar algum ataque, pois a identificação de um tráfego malicioso em um serviço legítimo é de extrema dificuldade. Além disso, possibilita o anonimato por parte do atacante tornando mais difícil sua localização.

2.1.3. AUTOMÁTICO

No ataque automático todos os detalhes de ataque, máquinas que serão atacadas, duração, tipos de ataques são programadas anteriormente ao início do ataque. Normalmente o programador deixa a possibilidade de alteração do código para modificar algum detalhe do ataque.

2.2. POSSIBILIDADE DE CARACTERIZAÇÃO

Algumas formas de ataques podem ser identificadas por características geradas quando este ocorre, formando assim uma assinatura. Mas existem ataques que não são possíveis de se caracterizarem, pois a cada instante que este ocorre, existe a possibilidade de surgir de uma forma diferente. Para isso iremos ver a diferença entre os ataques caracterizáveis e os não caracterizáveis (BEJTLICH, 2004).

2.2.1. CARACTERIZÁVEIS

Ataques caracterizáveis são aqueles que ocorrem sempre de uma mesma forma nas vítimas, ou seja, contém uma assinatura de ataque. Estes ataques podem ser:

- Filtráveis: são ataques que utilizam pacotes com má formação ou pacotes de serviços não permitidos pela vítima, sendo possível sua identificação por Firewalls;
- Não-filtráveis: são ataques que utilizam pacotes corretos e que utilizam serviços disponíveis na máquina alvo. Desta forma é impossível filtrar estes pacotes, pois não é possível diferenciar pacotes legítimos de pacotes suspeitos.

2.2.2. NÃO-CARACTERIZÁVEIS

Ataques não-caracterizáveis são aqueles que utilizam diferentes tipos de pacotes e/ou serviços, podendo mudar suas características a cada novo ataque. Desta forma os ataques não podem ser definidos por uma assinatura de ataque.

2.3. EXPLORAÇÃO DE FRAGILIDADES

Todos os ataques existem porque existe alguma vulnerabilidade que possibilita o ataque. Porém, muitas vezes estas vulnerabilidades são inevitáveis para que os sistemas funcionem corretamente. O atacante pode se aproveitar tanto da forma como é criada a comunicação, ou até mesmo de como são tratados os pacotes (DWYER, 2003).

2.3.1. EXPLORAÇÃO DO PROTOCOLO IPV4

O protocolo IP possui algumas falhas de segurança em sua especificação, fazendo com que seja possível promover um ataque utilizando suas fragilidades. Abaixo as principais definições de como pode ser efetuado um ataque utilizando o protocolo IP como meio.

1 – Ataques explorando insegurança do Protocolo IP:

- Enviar pacotes fabricados através de um endereço IP clonado, este podendo ser um endereço real ou inválido, também conhecido como *Spoofing* (MASELLI, et al., 2003);
- Enviar pacotes de início de conexão para um destinatário, no qual a resposta deve ser enviada para o mesmo destinatário. (Figura 1);

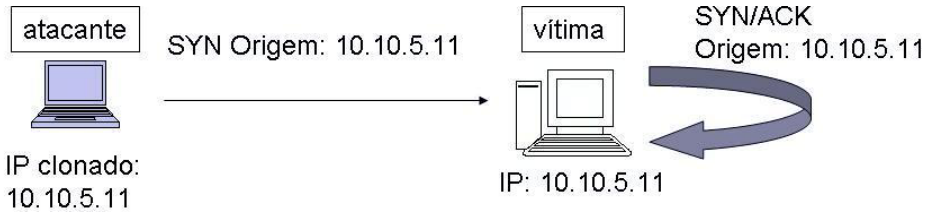

Figura 1 – Land Attack.

- Terminar ou redirecionar conexão através do envio pacotes de início ou fim de conexão TCP fora de ordem, também conhecido como *hijacking* (MASELLI, et al., 2003).

2 – Ataques que fabricam dados errados em pacotes IP válidos:

- Envio de pacotes com tamanho muito grande, fazendo com que máquinas tenham que tratar esse pacote e possam ficar indisponíveis;
- Uso incorreto de combinação de campos do cabeçalho ou fora de seqüência;
- Envio de grande número de pacotes contendo informação incorreta ou contendo campos inválidos.

3 – Ataques que violam as especificações do protocolo IP:

- Envio de pacotes válidos, mas por máquinas que desejam atacar o sistema;

- Exploração do método de conexão de três vias do protocolo TCP;
- Envio de pacotes TCP fora de ordem para detectar detalhes da máquina alvo.

2.3.2. EXPLORAÇÃO DE SERVIÇOS

Alguns ataques exploram falhas dos serviços básicos que acompanham o protocolo IP, com o objetivo de alterar o comportamento da rede. As principais formas de exploração são:

- Alteração do comportamento do serviço após obtenção, de forma ilícita, da permissão de administrador ao acesso do serviço.
- Injeção de informação incorreta, forjando esta.
- Alteração das tabelas de roteamento.

2.3.3. NEGAÇÃO DE SERVIÇOS

Este tipo de ataque tem o objetivo de deixar o tráfego de informações na rede de computadores lento ou fora de funcionamento. Este tipo de ataque pode ser efetuado por uma ou diversas máquinas. Dentro destes ataques estão:

- Envio de pacotes para destinos inválidos, para aumentar o volume de pacotes trafegando;
- Rajadas de pacotes enviados dentro de uma determinada rede, para aumentar o tráfego de pacotes e sobrecarregar a rede;
- Utilização de conexões com sessões sem fim, mantendo o alvo ocupado e indisponível para outros que tentem acessá-lo;
- Rajadas de pacotes de diversas máquinas para uma única máquina com o objetivo de deixá-la indisponível para requisições válidas (Figura 2).

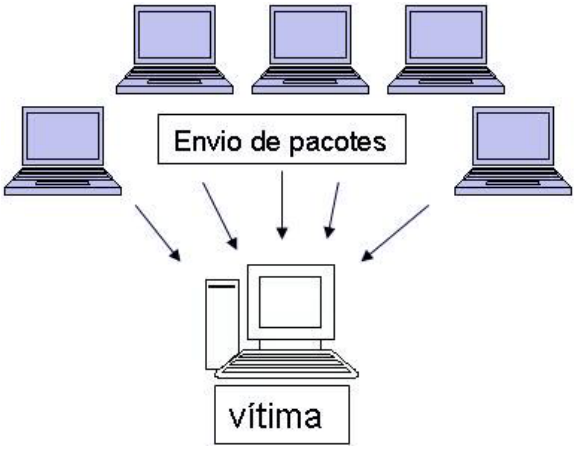

Figura 2 – Diversas máquinas enviando pacotes para vítima.

2.4. VÍTIMAS

Quando um ataque é formulado, ele tem por objetivo prejudicar alguma vítima. Esta vítima pode ser uma aplicação, uma máquina, um recurso, a rede ou infra-estrutura (MIRKOVIC; REIHER, 2004).

2.4.1. APLICAÇÃO

Ataques que tem por objetivo atingir apenas uma aplicação de alguma máquina, fazendo com que os outros serviços desta continuem funcionando normalmente.

Este tipo de ataque costuma ser de difícil identificação, pois os serviços não atacados normalmente continuam em funcionamento, e o tráfego gerado na rede para produzir o ataque é muito pequeno. Os pacotes enviados são de difícil diferenciação de pacotes legítimos, pois costumam ser pacotes bem formados (sem dados incorretos).

2.4.2. MÁQUINA

Quando um ataque a uma máquina ocorre, este tem por objetivo: sobrecarregar o sistema; desabilitar a comunicação com outras máquinas; paralisar o sistema; ou reiniciar o sistema. Caso o ataque não seja identificado, o alvo não consegue se recuperar do ataque sem a interferência de outra máquina ou alguém.

2.4.3. RECURSO

Ataques a recursos costumam objetivar fazer com que um recurso essencial para uma máquina ou várias se torne indisponível, como por exemplo, o servidor de DNS. Normalmente estes serviços essenciais são replicados, para caso de haver problemas o sistema continuar funcionando.

2.4.4. REDE

Ataques a rede tem por objetivo principal sobrecarregar o tráfego da rede, consumindo boa parte da banda. Estes ataques são facilmente identificáveis, devido ao grande número de pacotes trafegando.

2.4.5. INFRA-ESTRUTURA

Ataques a infra-estrutura objetivam atacar recursos distribuídos que são essenciais para um escopo global. Ataques destes tipos costumam ser ataques distribuídos, e são de extrema dificuldade de defesa. Um exemplo é ataque a roteadores, como por Figura 3.

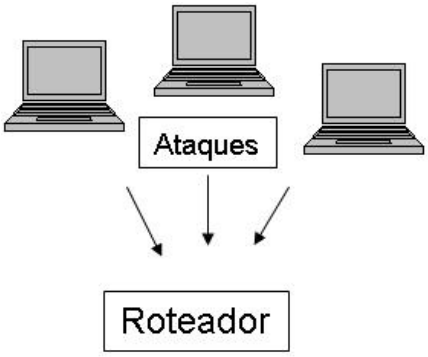

Figura 3 – Diversas máquinas atacando um roteador.

2.5. FORMAS DE DETECÇÃO

Existem duas formas principais de detecção de ataques, uma que se baseia na quantidade de tráfego da rede (anomalias), e outra que se baseia em comportamentos específicos de ataques conhecidos (assinaturas). Para isso veremos detalhadamente como estas duas formas de detecção funcionam.

2.5.1. DETECÇÃO POR ANOMALIAS

A detecção de ataques por anomalias se dá através de atividades que superam limites estabelecidos. O problema deste tipo de detecção é o grande número de falsos positivos, isto é, classificação de comportamentos normais como ataques. Por outro lado, esta forma de detecção consegue captar tentativas de intrusões, mesmo que ainda não sejam conhecidas pela comunidade científica. Portanto, o que basicamente é feito é a análise de picos de protocolos suspeitos que trafegam na rede, e sua distribuição entre os computadores (BARFORD, et al., 2002).

2.5.2. DETECÇÃO POR ASSINATURAS DE ATAQUE

A detecção de ataques por assinaturas utiliza basicamente as principais características de cada ataque. Para poder utilizar essas características é necessário conhecer o ataque que será detectado.

Existem algumas formas de detectar ataques por assinaturas, mas basicamente são estas:

- analisar a semântica dos protocolos, para evitar que algum atacante tente enviar mensagens com combinações erradas de cabeçalhos, que possam de alguma forma afetar o sistema.

- analisar atividades suspeitas, dentre elas: conexões incorretas ou incompletas, pois todas as conexões devem começar e seguir uma ordem, caso o número de pacotes fora de ordem seja elevado é caracterizado um ataque; simulação de conexões, com pacotes de tamanho ou quantidades muito grandes; razão entre requisições e respostas muito menor que 1 (um), por exemplo, sempre que se inicia uma conexão ela deve ser efetuada, se existem muitas conexões em aberto é um sinal de que existem muitas requisições, mas poucas conexões realmente sendo efetuadas (MIRKOVIC; REIHER, 2004).

2.6. RESUMO DO CAPÍTULO

As informações mostradas neste capítulo são de fundamental importância para o desenvolvimento do detector de anomalias, pois as definições de como o ataque pode ser gerado, as características do ataque e o quê explora, os alvos do ataque e a forma de detectá-lo indicam os caminhos para a criação do programa de detecção de intrusões. É importante ressaltar que este trabalho está focado principalmente em ataques de negação de serviço, com detecção por anomalias.

3. SÉRIES TEMPORAIS

Uma Série Temporal é uma análise sobre uma seqüência de dados, capturados no tempo, que possuam uma dependência com o seu passado. Estas análises costumam ser utilizadas em diversas áreas, como bolsa de valores, eletrocardiogramas, previsões do tempo, dentre outros. A idéia de utilização para detecção de intrusões é uma abordagem inovadora, e que, como qualquer experimento científico, pode não gerar os resultados desejados.

3.1. CARACTERÍSTICAS

Por Séries Temporais trabalharem com dados de determinado instante que dependem de dados de instantes anteriores, faz-se necessário o uso de técnicas específicas que levem em conta a ordem temporal dos fatos. A seleção de uma técnica adequada muitas vezes é complicada devido à necessidade de ser efetuada uma análise do que se deseja como resultado e de como os dados capturados se comportam. Desta forma existem diversas técnicas, mas para que se obtenha o resultado esperado é fundamental um estudo prévio de cada caso (EHLERS, 2005).

3.2. MODELO ARIMA

Como modelos autoregressivos (AR), de média móveis (MA), autoregressivo de médias móveis ARMA precisam de dados mais "comportados", e após uma análise dos dados que foram capturados, foi escolhido para trabalhar com detecção de intrusões o modelo Autoregressivo de Médias Móveis Integrado – *Autoregressive Integrated Moving Average* (ARIMA) –para produzir a série temporal dos dados capturados. Este tipo de modelo é feito para trabalhar com dados aleatórios de uma série estacionária ou não estacionária (EHLERS, 2005).

Séries estacionárias representam processos com a variância e co-variância que ficam em torno de uma média, isto é, os dados se comportam de forma mais equilibrada. Já séries não estacionárias representam dados com que não possuem uma aproximação de valores entre as amostras, podendo variar abruptamente. Séries sazonais refere-se a existência de periodicidade dos dados, isto é, tende a possuir repetições de comportamento durante o tempo (TRAN; REED, 2001), como pode ser visto na Figura 4.

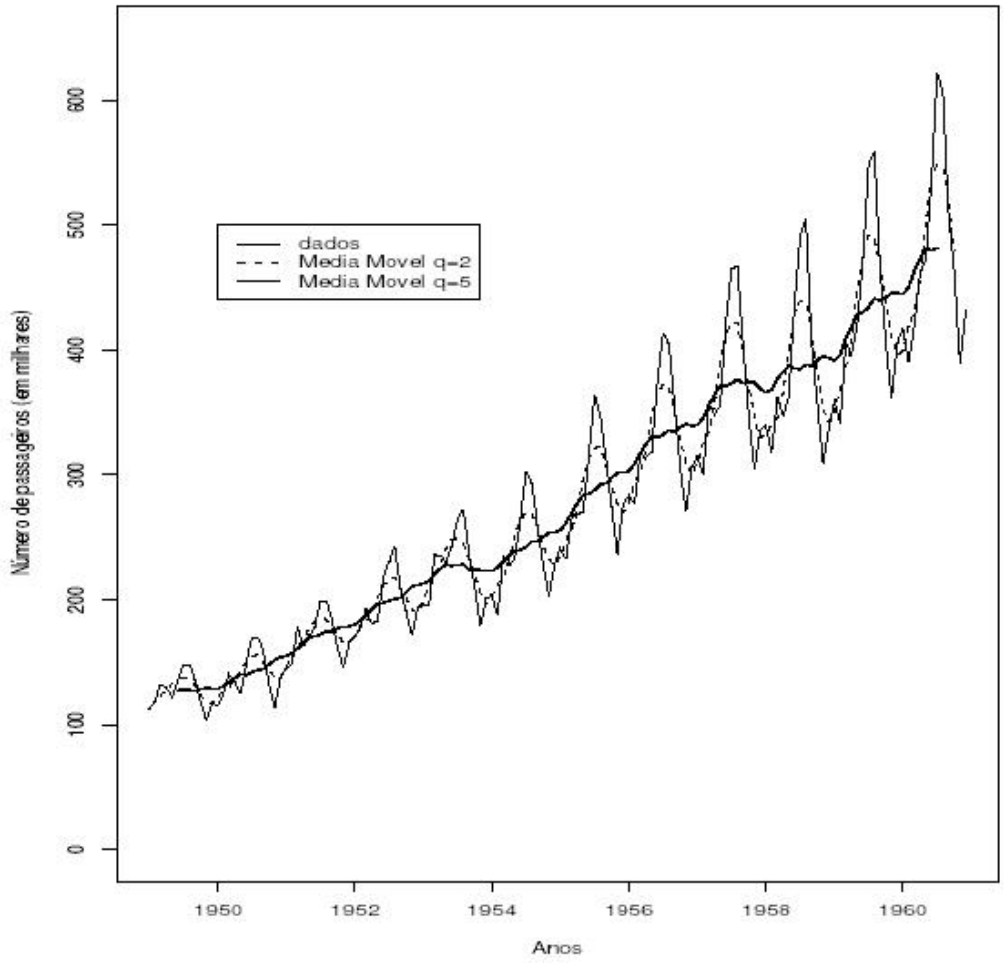

Figura 4 – Amostras com comportamento similar no tempo (TRAN; REED, 2001).

Com o uso das séries temporais, mais especificadamente com o modelo ARIMA, pretende-se estabelecer um padrão de comportamento do tráfego de rede, avaliar a cada amostra se a série está se comportando como esperado. Caso

a série temporal se comporte de forma diferente do esperado, isto pode ser o indício do acontecimento de um ataque. Esta abordagem é implementada neste trabalho, e os testes serão analisados para verificar se este é um bom método.

3.3. RESUMO DO CAPÍTULO

As informações mostradas neste capítulo são a base para o algoritmo que efetua a detecção de uma intrusão. A escolha do modelo ARIMA surgiu depois de um estudo e comparação entre os modelos existentes para criação de séries temporais. Foi visto também o motivo da escolha de séries temporais, já que existem diversos tipos de modelos estatísticos.

4. DETECÇÃO DE ATAQUES

Para a identificação de ataques é necessário decidir qual forma de detecção será adotada no sistema. Este capítulo discute quais os tipos de ataques que serão analisados neste trabalho e como eles são detectados.

4.1. ATAQUES A SEREM ABORDADOS

Para desenvolver este trabalho, partimos de alguns ataques conhecidos, com grande exploração em artigos, que podem nos auxiliar a respeito dos resultados a serem obtidos. Desta forma sabemos quais contadores deveremos analisar e como deve ser o comportamento destes em casos de ataques. Para isso iremos descrever detalhadamente os ataques a serem analisados, buscando identificar os contadores corretos para serem analisados através de séries temporais.

4.1.1. SYN ATTACK

O SYN Attack consiste na inundação de uma máquina por requisições TCP/SYN, fazendo com que ela não possa responder a outras requisições de conexão. Normalmente o atacante altera o seu endereço IP para que outra máquina receba as respostas SYN/ACK. Lembrando que o protocolo TCP, para estabelecimento de conexão, exige um pedido de conexão e uma resposta que pode haver a conexão para finalmente começa-la (PENG, et al., 2007) (Figura 5). Este ataque possui uma vasta literatura na comunidade científica, permitindo comparar os resultados com os já existentes em publicações (MASELLI, et al., 2003) (PENG, et al., 2007) (LEVCHENKO, et al., 2004).

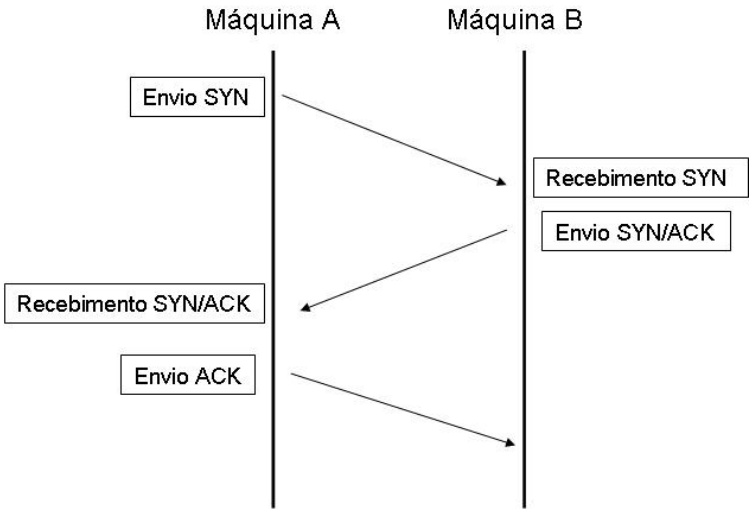

Figura 5 – Início de conexão TCP.

4.1.1.1. CARACTERÍSTICAS

O SYN Attack pode ser disparado manualmente, semi-automaticamente ou automaticamente. Possui característica marcante de negação de serviço, pois o principal objetivo é manter a máquina ocupada enquanto ocorre o ataque, nos casos que utilizam IP clonado também explora fragilidades do protocolo IP. A vítima principal é a máquina.

4.1.1.2. ATAQUE

O ataque SYN Attack possui os seguintes passos:

- O atacante altera seu endereço IP para um endereço conhecido na rede que acabará sendo um dos alvos do ataque (caso utilizem a técnica de clonar o IP);

- O atacante dispara seqüência de SYN (tentativa de conexão) para uma máquina (Figura 6);

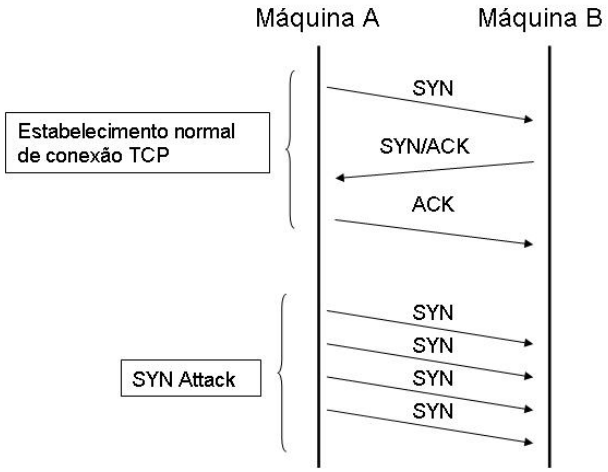

Figura 6 – SYN Attack.

- A máquina que recebeu o SYN, responde com SYN/ACK, que na verdade irá para a máquina que teve seu IP "clonado" pelo atacante, caso o atacante tenha alterado seu IP para um IP existente (Figura 7);

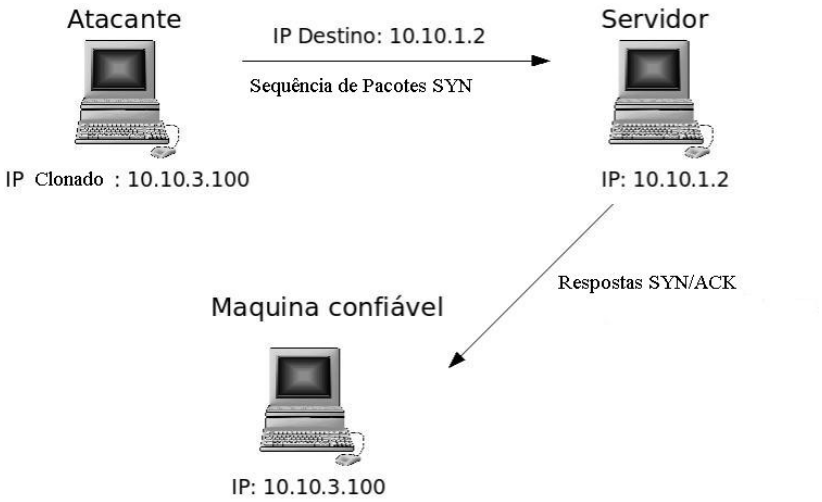

Figura 7 – SYN Attack, com redirecionamento de resposta.

- Desta forma tanto a máquina que recebeu o SYN e a que receber o SYN/ACK ficarão ocupadas e não poderão responder a conexões de outras máquinas (PENG, et al., 2007).

4.1.1.3. DETECÇÃO

O SYN ataque possui uma característica muito marcante, que é o envio massivo de pacotes TCP/SYN, e após isto a conexão não é continuada. Portanto, existem picos de pacotes SYN, com baixo número de pacotes ACK, RST ou FIN.

Desta forma veremos duas formas principais de detectar, usando somente picos de pacotes SYN (visto melhor na página 41) ou pela relação de pacotes SYN com os outros pacotes de conexão.

1 - Somente pacotes SYN:

- Pacotes SYN são comuns, pois são enviados sempre que se estabelecem conexões TCP;
- Em uma comunicação normal TCP, envia-se somente um SYN para estabelecer a conexão, desta forma, a cada início de conexão é enviado um pacote SYN;
- Conforme o tempo passa, espera-se que as utilizações de conexões TCP obedeçam a certo padrão, onde pode ser criada uma série temporal;
- Se a quantidade de pacotes TCP-SYN passar de certo limite em relação à série temporal, temos assim uma possível anomalia;
- Toda anomalia em relação a uma série temporal, é um possível ataque.

2 - Pacotes SYN relacionados com pacotes ACK, RST ou FIN:

- Em toda conexão TCP, após um pacote SYN, é necessário o envio de pacote um pacote ACK. Logo após são enviados pacotes RST, para envio de informação. Finalmente, um pacote FIN é necessário para finalizar a comunicação.

- Esta relação de SYN com o envio de pacotes ACK, RST ou envio de um pacote FIN só é perdida caso haja um problema na comunicação ou um ataque;
- Desta forma fica caracterizado uma assinatura, pois pacotes SYN têm uma relação direta com pacotes ACK, RST e FIN. Lembrando que a relação FIN/SYN deve ser igual a um, ou seja, os contadores SYN e FIN devem possuir o mesmo número.

4.1.2. SMURF/ FRAGGLE ATTACK

O Smurf Attack e o Fraggle Attack ocorrem de forma similar, enquanto o primeiro inunda uma máquina alvo de pacotes ICMP echo, o segundo inunda com UDP echo. Em ambos os ataques os pacotes partem de diversas máquinas para a máquina atacada.

4.1.2.1. CARACTERÍSTICAS

Os dois ataques podem ser produzidos manualmente, semi-automaticamente ou automaticamente. Exploram fragilidades do protocolo IP, pela alteração do IP do atacante pelo da máquina alvo, do protocolo ICMP, pela inundação do serviço ICMP echo em comunicação TCP, ou o serviço UDP echo em comunicação UDP. A vítima principal é uma máquina alvo, mas a rede também é afetada pelo grande número de pacotes (KUMAR, 2007).

4.1.2.2. ATAQUE

O Smurf Attack e o Fraggle attack possuem os seguintes passos:

- O atacante altera seu endereço IP, para um endereço conhecido ativo na rede e que será o alvo efetivo do ataque;

- Dispara-se uma seqüência de pings (verificação de tempo de demora para chegar a um determinado host, se este estiver alcançável) por broadcast;

- Todas as máquinas ativas na rede irão responder o ping para a máquina alvo (máquina que o atacante trocou seu IP).

- A máquina alvo é inundada de respostas de pings (Figura 8).

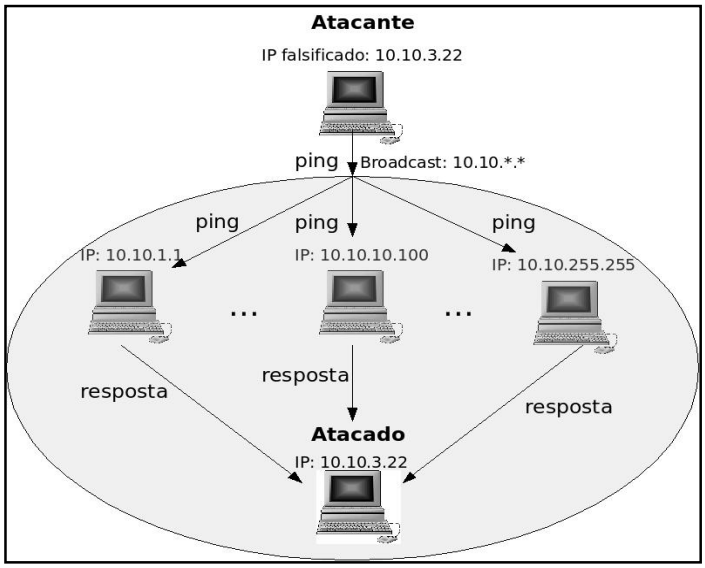

Figura 8 – Smurf/Fraggle Attack.

4.1.2.3. DETECÇÃO

Tanto o Smurf Attack quanto o Fraggle attack possuem características muito marcantes. Enquanto no primeiro é o envio massivo de pacotes ICMP echo, no segundo é baseado no envio de muitos pacotes UDP echo (KUMAR, 2007). Portanto, é possível caracterizar um ataque quando houver um pico muito grande destes pacotes chegando a um host específico. Outra característica é o aumento do número de mensagens broadcast, que também indica que provavelmente um dos dois ataques esteja ocorrendo. Abaixo veremos como detectar os ataques:

1 - ICMP Echo:

- Como o ICMP echo é um pacote que é utilizado pelos equipamentos de rede para verificar o estado dos links, estes são enviados periodicamente;

- Por serem enviados periodicamente, a função discretizada pode ser estabelecida através de uma série temporal;
- Desta forma, o envio de pacotes ICMP echo fora de um certo nível estabelecido em relação à série temporal, será considerado uma anomalia;
- Toda anomalia em relação a uma série temporal, é um possível ataque.

2 - UDP echo:

- Como o UDP echo é um pacote que é utilizado para verificação de tempo de transmissão de pacotes UDP entre duas máquinas, estes são enviados periodicamente;
- Por serem enviados periodicamente, a função discretizada pode ser estabelecida através de uma série temporal;
- Desta forma, o envio de pacotes UDP echo fora de um certo nível estabelecido em relação à série temporal, será considerado uma anomalia;
- Toda anomalia em relação a uma série temporal, é um possível ataque.

3 - Pacotes Broadcast:

- Pacotes enviados para broadcast são utilizados para verificar se dispositivos estão ativos, se existem fragilidades (portas abertas), envio de alertas, dentre outras;
- Pacotes para Broadcast são enviados periodicamente por softwares de gerenciamento para verificar o estado atual de cada máquina;
- Desta forma, o envio de pacotes broadcast fora de certo nível estabelecido em relação à série temporal, será considerado uma anomalia;

- Toda anomalia em relação a uma série temporal, é um possível ataque.

4.2. RESUMO DO CAPÍTULO

Os dados mostrados neste capítulo ressaltam os principais ataques que serão testados durante os testes do detector de intrusões, visto que possuem uma vasta literatura, podendo assim, efetuarem-se comparativos dos resultados obtidos. Foram vistos também quais contadores serão analisados para a criação das séries, essenciais para a verificação da presença de anomalias.

5. IMPLEMENTAÇÃO

Neste capítulo são abordados os passos efetuados para a implementação do Detector de Intrusões Baseados em Séries Temporais - DIBSeT.

5.1. PROGRAMAS DE CAPTURA DE DADOS

Veremos abaixo exemplos de programas que podem ser abordados para captura de dados. Como existe um projeto de cooperação entre a FHGe e a UFSM, a primeira escolha é o IAS – programa desenvolvido na FHGe, mas é importante lembrar que existem outros programas que fazem a mesma função, como por exemplo, o NTOP (MASELLI, et al., 2003) e o Wireshark (WIRESHARK, 2008). Neste texto veremos detalhes do IAS e do NTOP, o Wireshark não será visto por problemas com a sobrecarga do tráfego na rede do Prédio Sede do CRS/INPE - MCT.

5.1.1. IAS

A tarefa principal do sistema de análise de Internet IAS (POHLMANN; PROEST, 2006) é analisar dados de comunicação de subredes locais e, através da conexão entre vários IAS distribuídos em diferentes subredes da Internet, criar uma visão global sobre o comportamento da Internet. As visões local e global possibilitam realizar análises de anomalias considerando a variação no tráfego de rede. A Figura 9 apresenta um exemplo de contagem que pode ser realizado no IAS.

Observe que a partir dos dados coletados é possível deduzir características de um possível ataque. Adicionalmente, a possibilidade de envio das informações para um ou mais sistemas de análise, possibilita o uso paralelo de diferentes estratégias de detecção de intrusão. Outra característica importante do IAS é que as sondas de coleta de dados apenas identificam os tipos de pacotes

transferidos, abstendo-se de extrair qualquer informação relevante ao *payload* do pacote. Este procedimento garante a proteção dos dados e a confidencialidade da informação.

ID	Description		Count
131134	IP (Protocol Number 6)	:	18.854.151
131145	IP (Protocol Number 17)	:	1.123.149
327708	TCP (Flags: SYN)	:	334.435
327723	TCP (Flags: FIN/ACK)	:	480.697
327724	TCP (Flags: SYN/ACK)	:	275.779
545857	HTTP (Request Method POST)	:	2.026
545861	HTTP (Request Method GET)	:	293.616
545863	HTTP (Request Method HEAD)	:	18.992

Figura 9 – Contadores do IAS (POHLMANN; PROEST, 2006)

Para possibilitar a avaliação estatística do comportamento de tráfego, visando a detecção de anomalias, o IAS possibilita a amostragem dos contadores em intervalos eqüidistantes. Desta forma, é possível estimar o comportamento normal do tráfego e realizar análises *off-line* ou em tempo real com intuito de detectar possíveis anomalias. A Figura 10 demonstra o tráfego dos protocolos da camada de transporte em um período de vários dias, o que pode ser utilizado, por exemplo, para determinação de *thresholds* associados a geradores de alarmes.

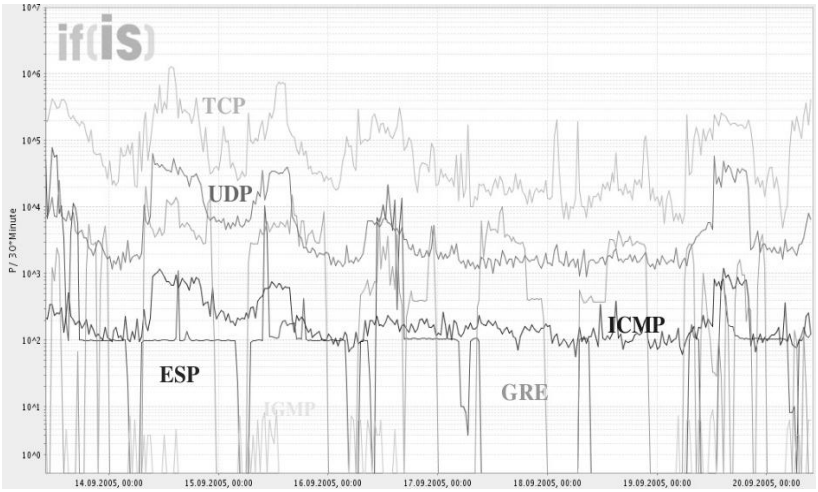

Figura 10 – Tráfego dos protocolos de transporte (POHLMANN; PROEST, 2006).

Apesar de seus benefícios, o IAS apresentou problemas onde foi instalado. Por diversas vezes seus serviços ficaram indisponíveis e seus contadores foram zerados. Além disso, devido a problemas de comunicação com o grupo da FHGe, não foi possível obter os contadores desejados.

5.1.2. NTOP

Assim como IAS, o NTOP (MASELLI, et al., 2003) também é um programa para captura de tráfego de rede e armazenamento dos dados em contadores, porém existem algumas diferenças entre estes.

O NTOP é um software gratuito, disponível no site de seus mantedores, e com fácil acesso aos contadores, ao contrário do IAS, o qual possui uma arquitetura muito "fechada". Porém o NTOP possui menor número de contadores.

Deve ser dito que o NTOP possui alguns problemas em sua instalação e configuração, entre eles: algumas versões possuem *bugs*, os quais não geram arquivos de saídas com os contadores; a assistência no site é muito pobre e precisa de diversas bibliotecas, que não são especificadas por seu fabricante, para o correto funcionamento.

Apesar destes problemas, o NTOP possui fácil instalação, fácil manuseio, as opções podem ser estabelecidas via http ou https, e gera gráficos automaticamente da rede (vide Figura 11).

Os contadores do NTOP, assim como os do IAS, podem ser analisados para a obtenção de séries temporais, para obter anomalias relativas ao histórico do comportamento dos dados capturados. Um dos problemas do manuseio dos dados é que para obter os contadores é necessário utilizar uma consulta via html, visto que os dados são armazenados em formato binário no banco de dados. Porém, através da consulta html é possível obter todos os contadores em

arquivos de diversos formatos, como extensão XML, PHP, TXT, dentre outros menos usuais.

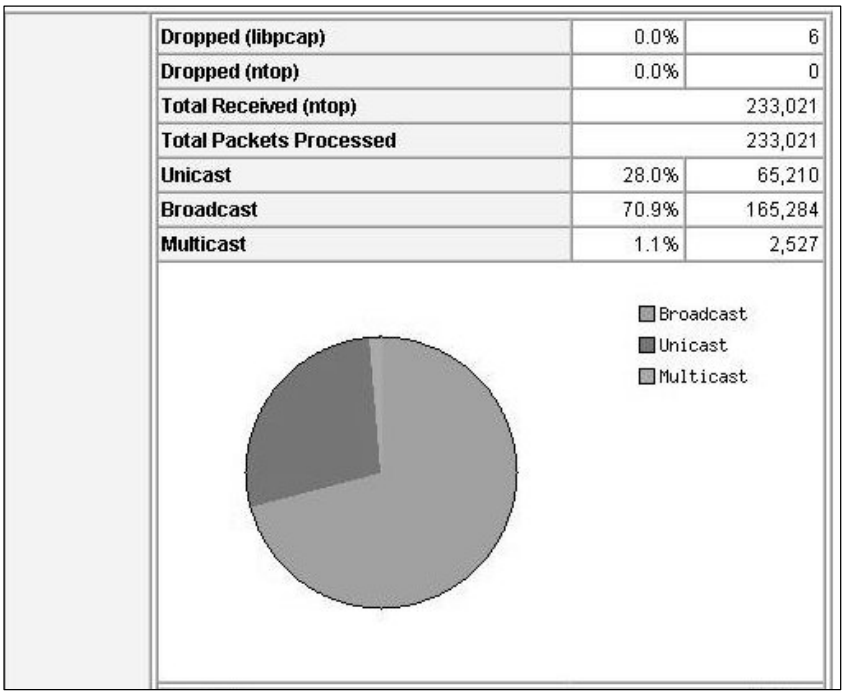

Figura 11 – Gráfico gerado pelo NTOP.

Os contadores do NTOP são sempre incrementados, fazendo com que o resultado de uma consulta sempre retorne um valor igual ou superior ao anterior. Um problema para o monitoramento contínuo é que quando o Sistema Operacional onde está instalando o NTOP é reiniciado, todos os contadores são zerados, forçando um maior controle no algoritmo que monta a série temporal.

5.2. SIMULAÇÃO DE ATAQUES

Com o objetivo de obter dados para a implementação do detector de ataques, foram simulados alguns ataques para verificar o comportamento do tráfego de pacotes. Para a captura de dados foi utilizado o programa NTOP na versão 3 (três). Os resultados dos ataques e os programas serão discutidos no decorrer deste trabalho.

5.2.1. PROGRAMAS DE ATAQUES

Para a construção de pacotes para simular ataques a uma determinada máquina, foram utilizados dois programas: o HPING; e o packETH, ambos disponíveis gratuitamente e com licença GPL versão 2. Nos dois programas é possível encontrar versões para diversos sistemas operacionais como Windows, Linux, FreeBSD.

5.2.1.1. HPING

O HPING é um programa desenvolvido para fabricar pacotes de rede como: ICMP, TCP, UDP, para testar fragilidades de segurança de firewalls, procurar portas abertas (HPING, 2007). O programa, mesmo não possuindo interface gráfica, possui fácil execução e diversas opções de construção e envio. Um ponto interessante deste programa é a possibilidade de utilização de campos errados na construção dos protocolos, o que possibilita uma maior diversidade de ataques.

5.2.1.2. PACKETH

O PackETH foi desenvolvido com a capacidade de gerar pacotes ICMP, TCP, UDP, IGMP, RTP, este último protocolo é usado em comunicações VoIP que vem se tornando um grande alvo de ataques por não existirem padrões de segurança ainda fortemente estabelecidos (SISALEN, et al., 2005). O programa também possibilita a variação do intervalo entre envios, modificação de campos no protocolo, dentre outras várias possibilidades de configurações. Mas o grande diferencial é a interface gráfica de fácil entendimento como visto na Figura 12. Já a possibilidade de escolha de quantidade de pacotes e os atrasos entre envios podem ser vistos na Figura 13.

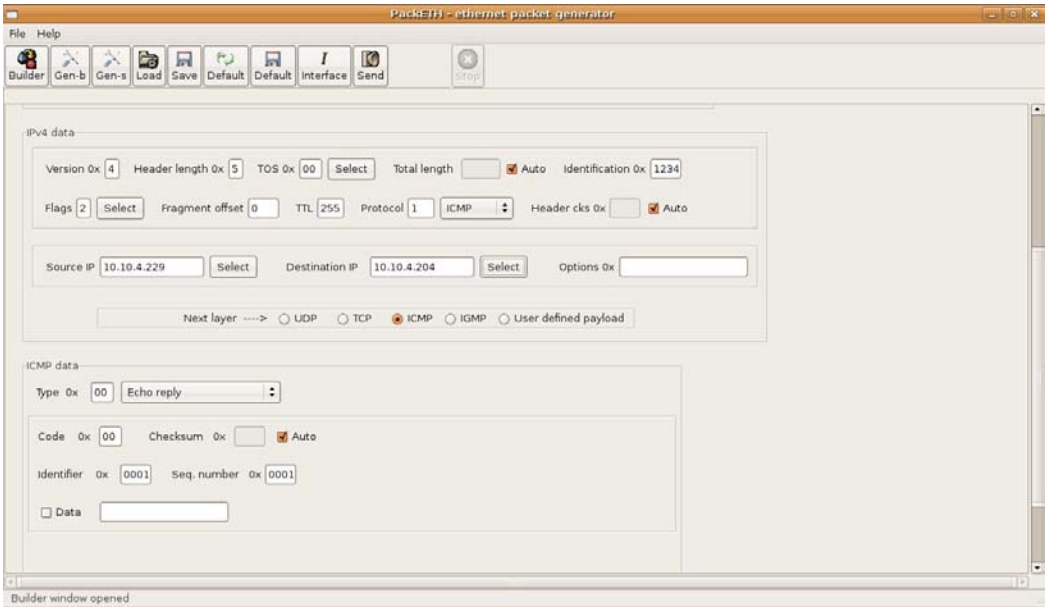

Figura 12 – Imagem da interface do packETH.

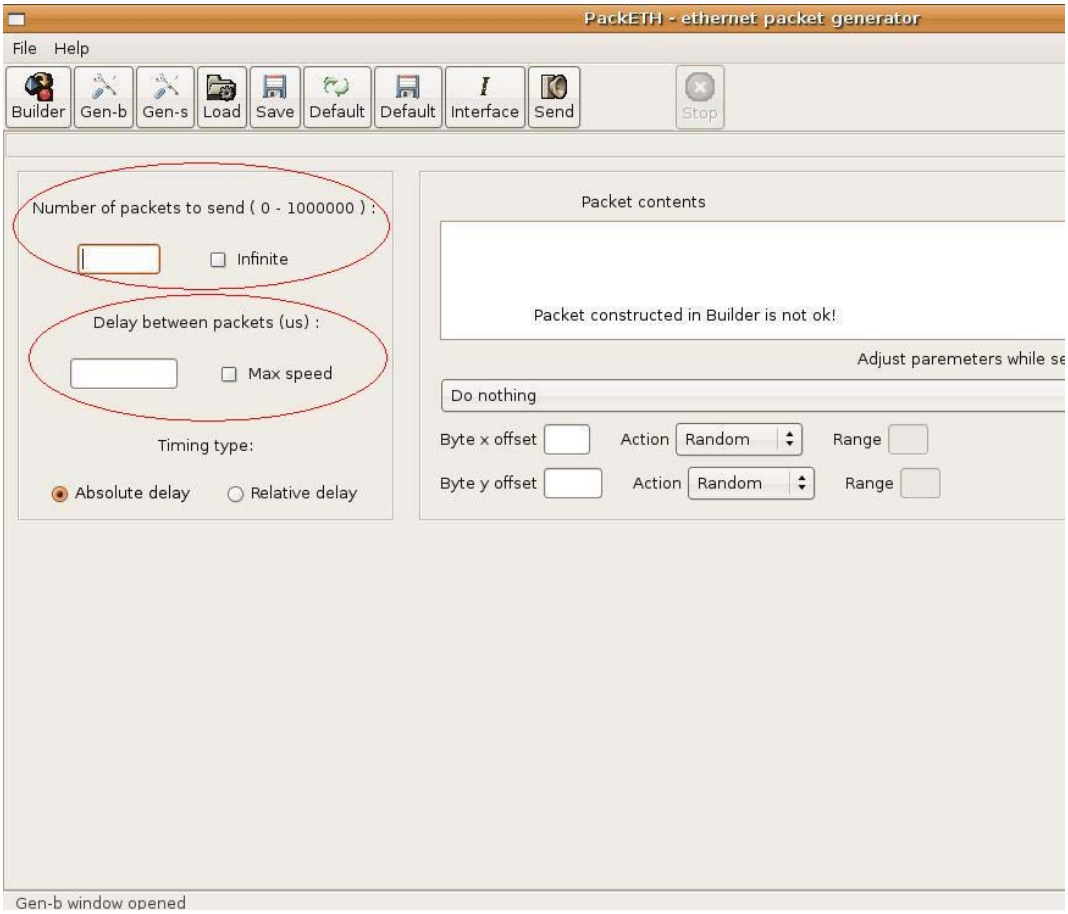

Figura 13 –PackETH e opções de quantidade e atraso de pacotes.

5.2.2. RESULTADO DOS ATAQUES

Foram simulados ataques Smurf Attack e SYN Attack, utilizando os dois programas de geração de pacotes para a geração de ataques, e a captura de dados feita pelo programa NTOP. Os resultados ocorreram como o esperado, gerando anomalias no comportamento do recebimento de pacotes ICMP e SYN/TCP respectivamente. Mas a grande surpresa foi a diferença de anomalias geradas pelos programas. Como podemos ver na Figura 14 e na Figura 15, o programa packETH gerou ataques muito mais efetivos quando comparado com o HPING, mesmo utilizando configurações muito parecidas.

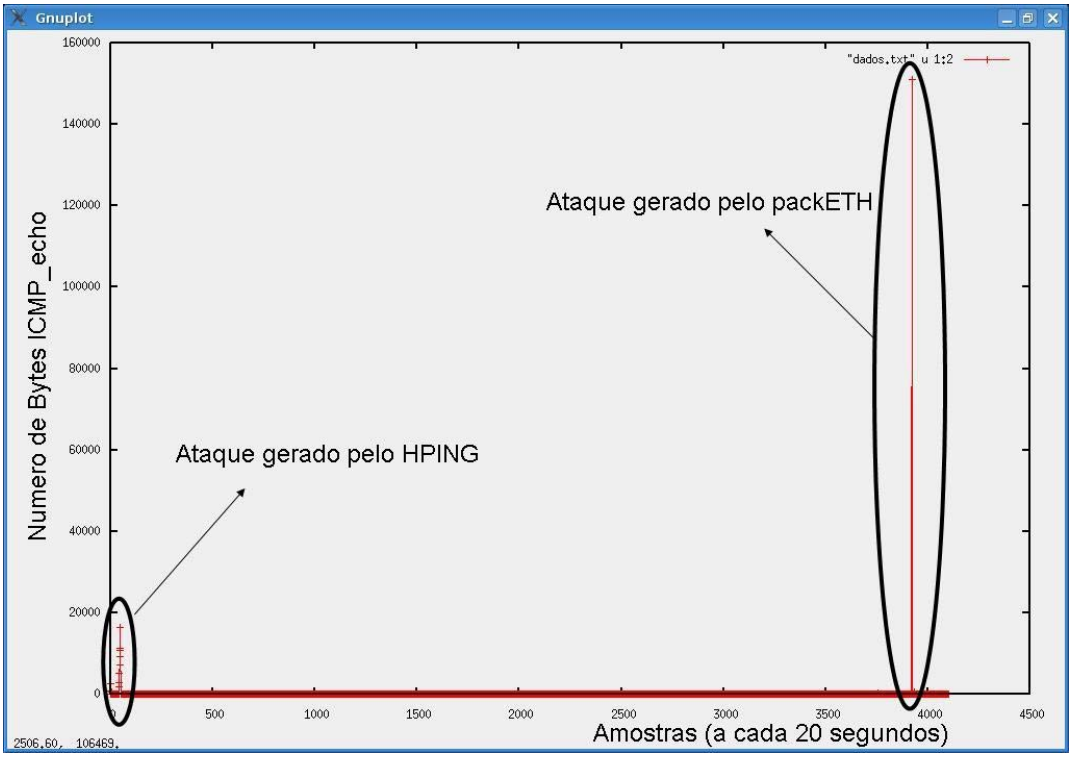

Figura 14 – Comparação de Smurf Attack gerado.

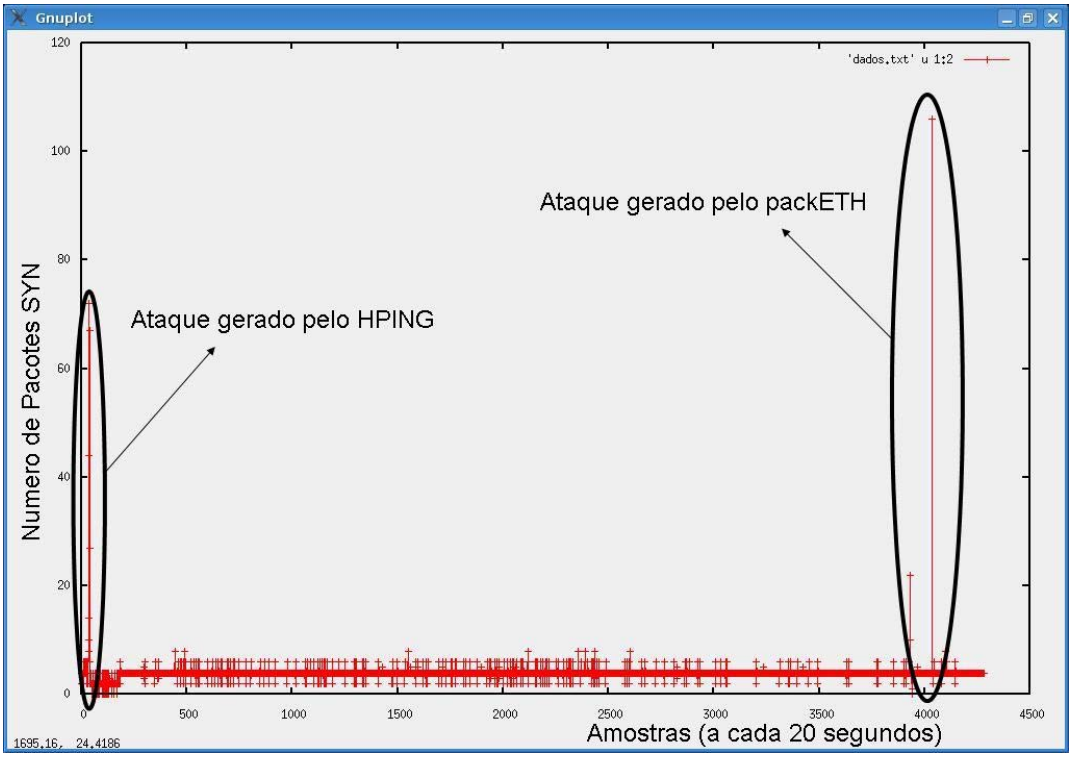

Figura 15 – Comparação de SYN attack gerado.

5.3. USO DE SÉRIES TEMPORAIS

Para construir o DIBSeT foram utilizadas séries temporais, mais especificadamente o modelo ARIMA. Para a criação deste modelo, foi utilizado o preditor de passos em séries temporais desenvolvido pelo orientador deste trabalho em sua tese de doutorado (NUNES, 2003). A partir dos dados processados pelo gerador de séries temporais, somado a uma margem, pôde-se obter previsões de anormalidade, ou seja, os *thresholds* dinâmicos, usando uma curva que se adapta ao histórico dos dados inseridos durante o tempo. Desta forma puderam se obter anomalias em relação aos comportamentos obtidos anteriormente. Quando observado algum comportamento anômalo (GOODALL, 2006), foi definida a série do contador como um possível ataque neste mesmo contador, lembrando que são observados mais de um contador. Para que o DIBSeT não produzisse resultados simplórios, isto é, somente se existe ou não

uma anomalia, foram gerados níveis de alarmes, variando de zero a cinco, sendo o valor zero correspondente a inexistência de anomalias, e os valores de 1 a 5 correspondentes ao valores de menor e maior razão entre a faixa prevista e o valor computado. Para análise posterior e/ou alimentação do DIBSeT com dados antigos quando iniciado, foram gerados *logs* dos contadores e dos níveis de alarme gerados.

5.3.1. ARQUITETURA DO DIBSET

A primeira versão do DIBSeT, foi implementada utilizando os dados do NTOP. Para isso foi necessária a criação de classes para ler os arquivos gerados pelo NTOP e a filtrar a informação desejada. Logo após os contadores obtidos são passados para classes que implementam o modelo ARIMA, desta forma, cada contador é adicionado a sua Série Temporal.

Como as classes implementadas pelo orientador deste trabalho utilizavam funções e bibliotecas do programa RPS (DINDA, 2007), foi necessária a instalação deste *software*. Para a instalação do programa RPS com as funções de Séries Temporais, é necessário a instalação do programa de funções matemáticas não gratuito, Numerical Recipes (NUMERICAL RECIPES, 2007). Ambos programas foram de difícil instalação pois teve-se de utilizar versões antigas de bibliotecas e compiladores, como versões anteriores a 4 da STL (Standard Template Library) (STL, 2008) e versões anteriores a 3 do g++ (GCC, 2008). Para que o RPS (programado em C++) e as classes de séries temporais (feitas em java) pudessem trocar informações foi necessário a criação de um *proxy* usando JNI.

Após a análise por séries temporais e a criação de uma margem, foi necessária a geração de níveis de alarmes. Com níveis de alarmes pode se ter uma melhor idéia da situação da rede, não apenas se houve uma anomalia, mas quão longe a anomalia se distanciou da margem prevista.

Os contadores e os níveis de alarme de cada contador são armazenados para: a realimentação da série, pois a cada vez que é iniciado é necessária a criação da série; e a análise *off-line* dos dados. O diagrama abaixo (Figura 16) ilustra melhor o DIBSeT.

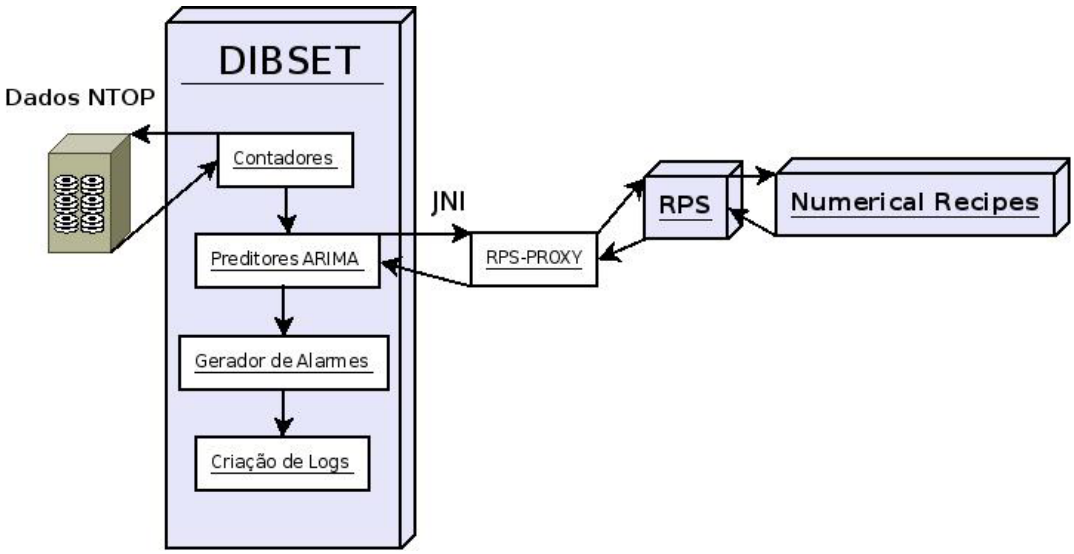

Figura 16 – Diagrama do DIBSeT.

5.3.2. NÍVEIS DE ALARME

Os níveis de alarme geram dados mais detalhados para o software ou gerente que irá tomar a decisão. Os níveis são criados através de uma relação entre o contador atual, e a sua previsão somada com a uma margem. Primeiramente geramos alarmes apenas quando o valor ultrapassava a faixa superior pré-estabelecida, isto é, só quando o valor do contador era maior que a faixa. Quanto maior esta razão, maior o nível de alarme. Abaixo estão gráficos de uma parte das amostras coletadas durante 5 dias. Na Figura 17 pode ser vista a variação do contador ICMP_echo e na Figura 18 a variação dos níveis de alarme. Pode ser visto que as anomalias geradas foram correspondentes aos níveis de alarmes.

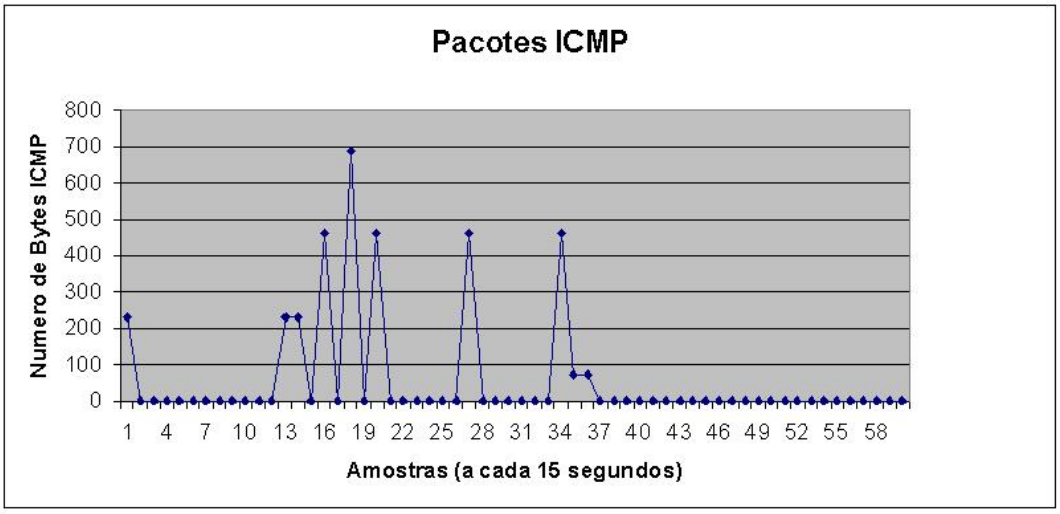

Figura 17 – Gráfico de Bytes ICMP.

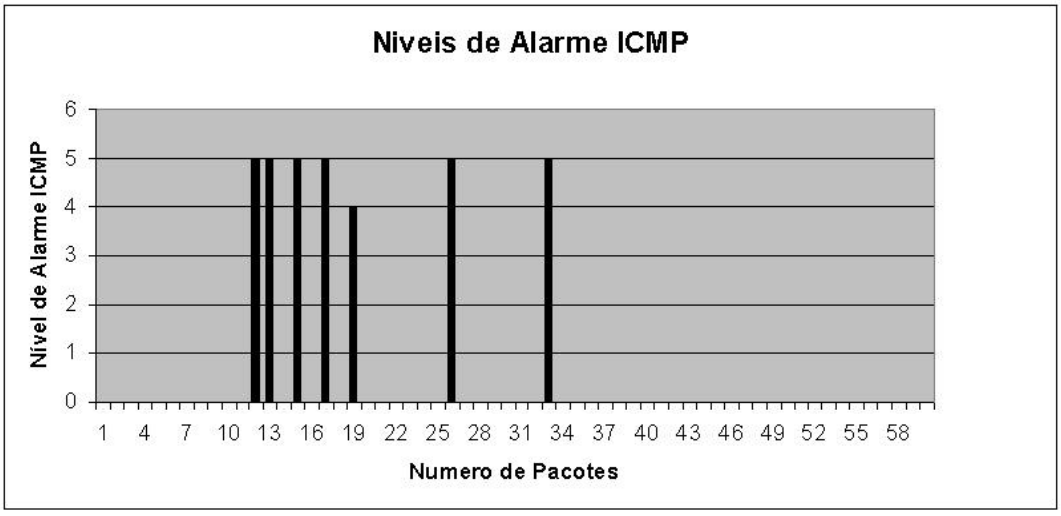

Figura 18 – Gráfico de Níveis de Alarme ICMP.

Na Figura 18, também pode ser observado que os níveis de alarmes muitas vezes são altos com quedas bruscas para zero. Isto se deve ao fato da indisponibilidade, gerada pelo ataque, no servidor onde está instalada a sonda de captura de dados.

Na Figura 20, observa-se que em alguns instantes os níveis de alarmes são relativamente altos, mesmo quando as variações são pequenas, isto se deve ao fato de que a série temporal criada espera valores pequenos. Os valores maiores

para os níveis de alarme ocorrem quando realmente é enviado um número massivo de pacotes SYN entre as amostras 113 e 127, como pode ser visto na Figura 19.

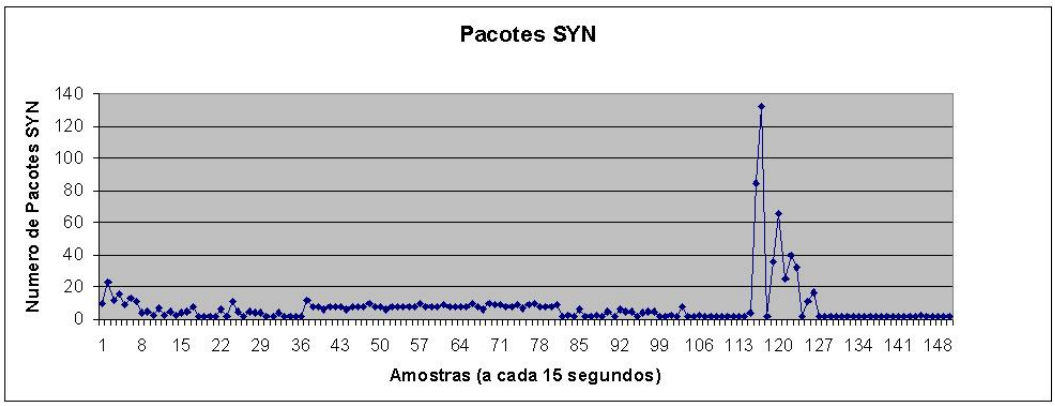

Figura 19 – Gráfico de pacotes SYN.

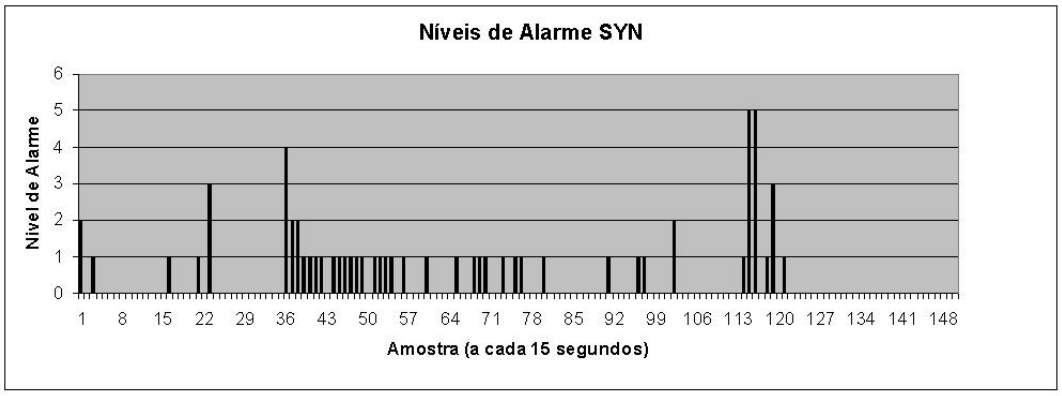

Figura 20 – Níveis de Alarme SYN.

Com o objetivo de gerar dados mais robustos para o gerente ou software que tenha que tomar alguma decisão a partir dos níveis gerados, foram analisados também os dados que ficaram abaixo da faixa inferior, ou seja, além do contador ter de ficar abaixo do *threshold* superior, também precisa ficar acima da margem inferior. Adotaram-se níveis negativos para identificar as amostras que ficaram abaixo do *threshold* inferior. Isto se deve ao fato, visto anteriormente, de que muitas vezes após gerar um ataque, os contadores caem a zero devido a indisponibilidade do sistema, indisponibilidade esta que também

afeta os demais contadores. Desta forma, todo contador que fica abaixo do esperado, pode ser resultado de um ataque.

Na Figura 22 podemos verificar que após ser gerado um *Smurf Attack*, entre as amostras 177 e 188 da Figura 21, os níveis de alarme invertem de sinal, isto é ficam abaixo do esperado pois o servidor que estava sendo analisado caiu.

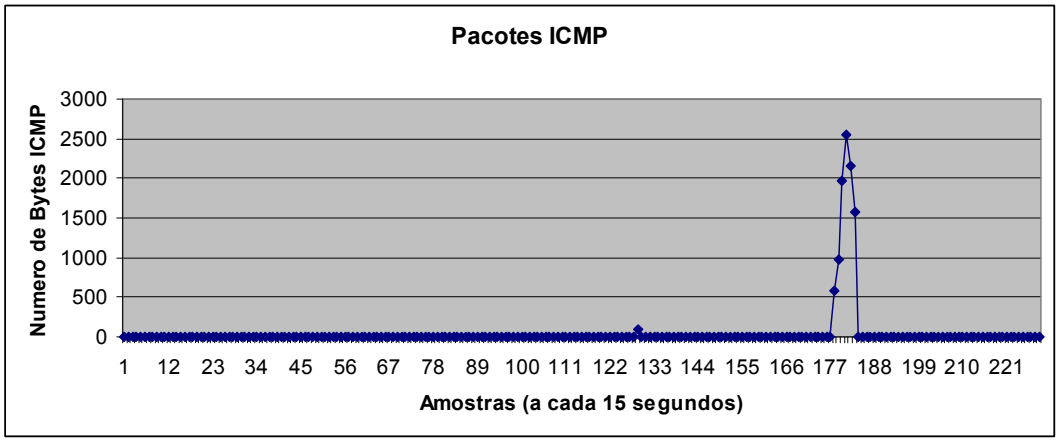

Figura 21 – Gráfico de pacotes ICMP.

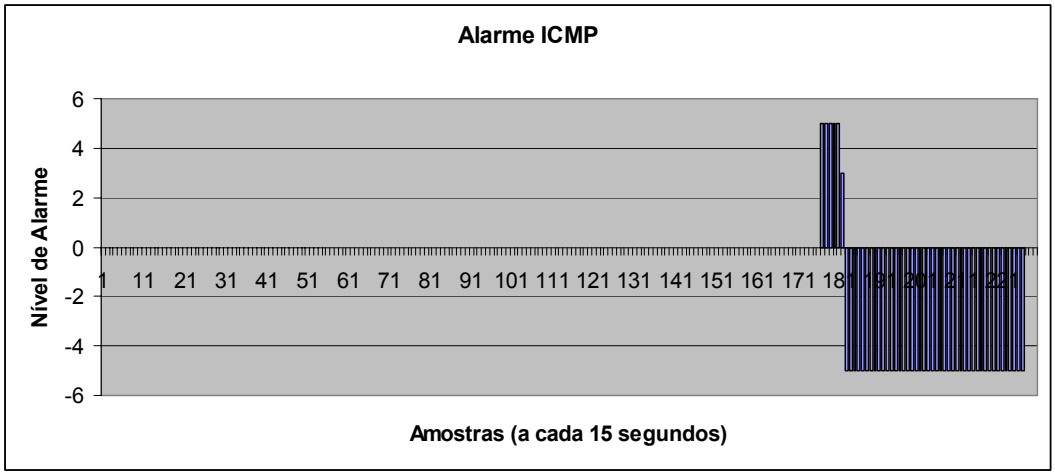

Figura 22 – Níveis de Alarme ICMP.

Na Figura 24 podemos verificar que após ser gerado um *SYN Attack*, entre as amostras 113 e 127 da Figura 23, os níveis de alarme também invertem de sinal, isto é ficam abaixo do esperado pois o servidor que estava sendo analisado, novamente caiu.

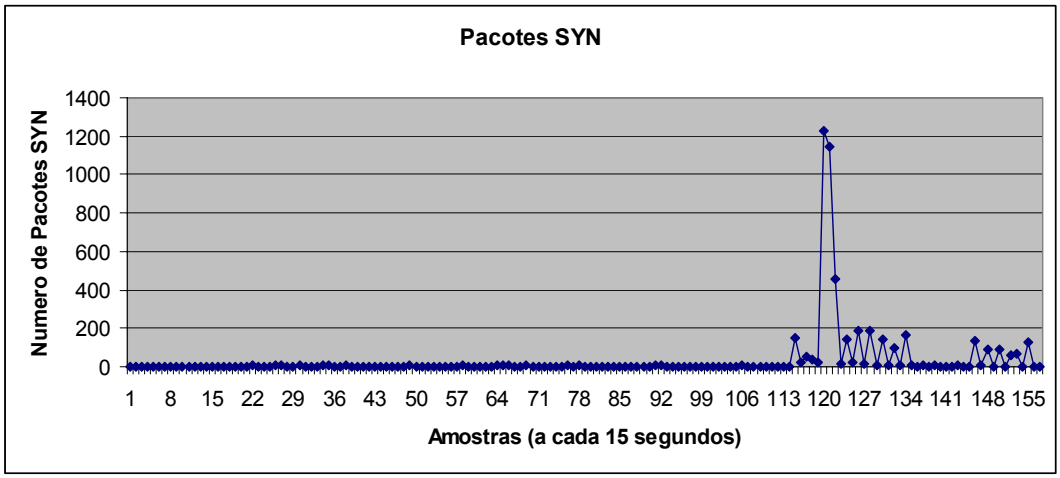

Figura 23 – Gráfico de pacotes SYN.

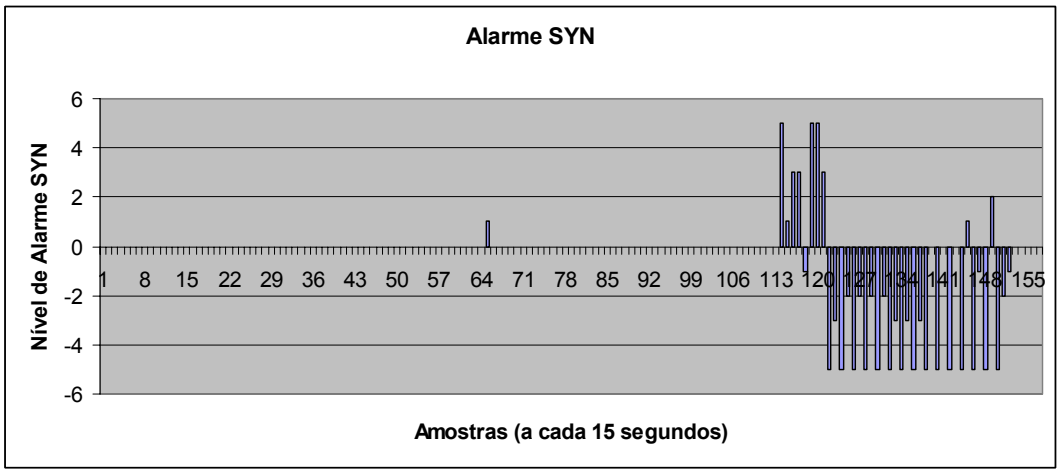

Figura 24 – Níveis de Alarme SYN.

5.3.3. RESULTADOS

Uma observação importante é que nas primeiras horas de análise, a série temporal gerava muitos resultados fora do esperado, mas com a criação da série temporal de mais de três dias, os resultados foram bastante satisfatórios, isto é, baixo número de falsos positivos e grande número de alarmes quando eram gerados ataques. Os parâmetros de níveis de alarmes precisaram ser modificados diversas vezes, pois somente depois de diversos testes de ataques foi possível verificar valores condizentes com o esperado.

5.4. RESUMO DO CAPÍTULO

Neste capítulo foram demonstrados os programas de captura de dados de comunicação, que armazenam em contadores. Esses contadores podem revelar o comportamento da rede ou de uma máquina em específico, quando se é gerado um histórico destes dados. Para poder analisar se um detector de ataques irá funcionar corretamente, é necessário o teste de ataques (BOLZ, 2004), então neste capítulo também foram discutidos programas para gerar pacotes, e produzir determinado ataque a uma máquina. Também foi visto o processo de construção do detector de intrusões DIBSeT, o seu diagrama, os resultados por este produzido e a verificação de que o método alcançou seu objetivo.

6. CONCLUSÕES

Durante a execução deste trabalho obtivemos diversas dificuldades principalmente relacionadas ao uso do IAS por não conseguirmos informações detalhadas sobre seus contadores. O NTOP surgiu como uma alternativa para a obtenção de resultados para testes da abordagem adotada, visto que possui uma estrutura similar ao IAS. Para tornar mais claro como um ataque ocorre, abordou-se nesse texto a caracterização dos ataques, para que fosse possível entender melhor como efetuar a detecção através de contadores. As escolhas do modelo ARIMA e dos ataques a serem analisados pelo detector de intrusões DIBSeT é muito importante para poder se chegar a resultados de testes comparativos com os programas existente. Após os ataques efetuados, foi possível observar que existe a possibilidade de se obter bons resultados aplicando séries temporais para detectar intrusões buscando por anomalias no comportamento dos contadores de pacotes de comunicação.

Com a conclusão da implementação do DIBSeT e os testes efetuados, obtivemos resultados condizentes com o esperado, apesar da necessidade da alteração dos parâmetros de classificação entre os níveis de alarme diversas vezes, até chegar em resultados mais satisfatórios. Também foi visto que é preciso que sejam captados dados de pelo menos três dias para obterem-se resultados satisfatórios.

6.1. TRABALHOS FUTUROS

Como continuação deste trabalho pretende-se utilizar os dados do IAS, motivo da parceria FHGe-UFSM, para a obtenção dos contadores de pacotes de redes. Após o refinamento do código de geração de alarmes, com o objetivo de gerar menos falsos positivos, pretende-se expandir o número de contadores analisados, isto é, capacidade de detectar maior número de ataques; além de

analisar dados de contadores de diferentes máquinas para obter um comparativo entre máquinas servidoras e máquinas de usuários.

REFERÊNCIA BIBLIOGRÁFICA

BARFORD, P., KLINE, J., PLONKA, D., RON, A. **A signal analysis of network traffic anomalies**. Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement, 2002.

BEJTLICH, R. **The Tao of Network Security Monitoring Beyond Intrusion Detection**, Publisher: Addison Wesley, 2004.

BOLZ, C., ROMNEY, G. e ROGERS, B. **Safely Train Security Engineers Regarding the Dangers Presented by Denial of Service Attacks**. In: ACM Conference on Information Technology Education, Session Security II, Pages 62-72, 2004.

DINDA, P. A. RPS – **An Toolkit for Resource Prediction in Distributed System**. http://rps.cs.northwestern.edu, último acesso em novembro de 2007.

DWYER, D. **Network Intrusion Detection**. 3rd Edition, Publisher: New Riders Publishing, 2003.

EHLERS, R. S. **Análise de Séries Temporais**. 3rd edição, Departamento de Estatística, Universidade Federal do Paraná, 2005.

GOODALL, J. **Visualizing Network Traffic For Intrusion Detection**. In: ACM Symposium on Designing Interactive Systems, pages 363-364, 2006.

GCC. GCC, the GNU Compiler Collection. http://www.gcc.gnu.org/, último acesso em Janeiro de 2008.

HPING. http://www.hping.org/, último acesso em novembro de 2007.

JAVA Technology. **Java**. http://java.sun.com/, último acesso Novembro de 2007.

KARGL, F., MAIER, J. e WEBER M. **Protecting Web Servers from Distributed Denial of Service Attacks**. In: ACM Proceedings of the 10th international conference on World Wide Web, 2001.

KOMPELLA, R. R., SINGH, S. E VARGHESE, G. **On Scalable Attack Detection in the Network**. In: IEEE/ACM TRANSACTIONS ON NETWORKING, Vol. 15, No. 1, February 2007.

KUMAR, S. **Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet**. Second International Conference on Internet Monitoring and Protection (ICIMP IEEE 2007), 2007.

LEVCHENKO, K., PATURI, R. e VARGHESE, G. **On the Difficulty of Scalably Detecting Network Attacks**. CCS-ACM, 2004.

MASELLI, G., DERI, L. e SUIN, S.; **Design and Implementation of an Anomaly Detection System: an Empirical Approach**. Proceedings of Terena Networking Conference (TNC 03), Zagreb, Croatia, May 2003.

MIRKOVIC, J. e REIHER, P. **A Taxonomy of DDoS Attack and DDoS Defense Mechanisms**. ACM SIGCOMM Computer Communications Review, Volume 34, Number 2, 2004.

NORTHCUTT, S. **Network Intrusion Detection: An Analyst's Handbook**. Publisher: New Riders Publishing, 1999.

NUMERICAL RECIPES, http://www.nr.com , Cambridge University, último acesso em novembro de 2007.

NUNES, R. C. **Adaptação dinâmica do timeout de detectores de defeitos através do uso de séries temporais**. Tese de Doutorado pela Universidade Federal do Rio Grande do Sul – UFRGS, 2003.

PACKETH. http://packeth.sourceforge.net, último acesso em novembro de 2007.

PENG, T., LECKIE C. e RAMAMOHANARAO K. **Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems**. In: ACM Computing Surveys, Vol. 39, No 1, Article 3, 2007.

POHLMANN, N. e PROEST M.; **Internet Early Warning System: The Global View**. In: Vieweg, Securing Eletronic Business Process, pages 377 – 386, 2006.

SISALEN, D., EHLERT, S., GENEIALTAKIS, D., KAMBOURASKIS, G., DAGIUKLAS, T., MARKL, J., ROKOS, M., BOTRON, O, RODRIGUEZ, J., e LIU, J. **Towards a secure and reliable VoIP infrastructure**. In: SNOCER, D21, 2005.

SNORT. www.snort.com.br, último acesso novembro de 2007.

STL. **Standard Template Library Programmer's Guide**. http://www.sgi.com/tech/stl/, último acesso em Janeiro de 2008.

TANENBAUM, A. S. **Redes de Computadores**, 3. ed. Rio de Janeiro: Editora Campus, 1997.

TIA/EIA, (Telecommunications Industry Association/Eletronics Industry Association). **Comercial Building Telecommunications Cabling – Standard ANSI/TIA/EIA 568 A**.

TRAN, N., REED, D. A. **ARIMA Time Series Modeling and Forecasting for Adaptive I/O Prefetching**. In: ACM 15th International Conference on Supercomputing, Sorrento, Italy, 2001.

WIRESHARK. http://www.wireshark.org/, último acesso em Janeiro de 2008.