UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO

DETECÇÃO DE INTRUSÃO ATRAVÉS DA ANÁLISE DE SÉRIES TEMPORAIS E CORRELAÇÃO DO TRÁFEGO DE REDE

DISSERTAÇÃO DE MESTRADO

Francisco Carlos Vogt

Santa Maria, RS, Brasil.

2011

DETECÇÃO DE INTRUSÃO ATRAVÉS DA ANÁLISE DE SÉRIES TEMPORAIS E CORRELAÇÃO DO TRÁFEGO DE REDE

por

Francisco Carlos Vogt

Dissertação apresentada ao Curso de Mestrado do Programa de
Pós-Graduação em Engenharia de Produção, Área de Concentração
em Qualidade e Produtividade, da Universidade Federal de Santa Maria (UFSM, RS),
como requisito parcial para a obtenção do grau de

Mestre em Engenharia de Produção

Orientador: Prof. Dr. Raul Ceretta Nunes

Santa Maria, RS, Brasil.

2011

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Vogt, Francisco Carlos
DETECÇÃO DE INTRUSÃO ATRAVÉS DA ANÁLISE DE SÉRIES TEMPORAIS E CORRELAÇÃO DO TRÁFEGO DE REDE / Francisco Carlos Vogt.-2011.
88 f.; 30cm

Orientador: Raul Ceretta Nunes
Dissertação (mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Produção, RS, 2011

1. Detecção de Anomalias; 2. Detecção de Ataques; 3. Séries Temporais. I. Nunes, Raul Ceretta II. Título.

Endereço: Rua Castro Alves, 486 Carazinho, RS. CEP: 99500-000
Fone (0xx)54 33313110; E-mail: fcvogt@gmail.com

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Comissão Examinadora, abaixo assinada,

Aprova a Dissertação de Mestrado

DETECÇÃO DE INTRUSÃO ATRAVÉS DA ANÁLISE DE SÉRIES TEMPORAIS E CORRELAÇÃO DO TRÁFEGO DE REDE

elaborada por
Francisco Carlos Vogt

como requisito parcial para a obtenção do grau de

Mestre em Engenharia de Produção

COMISSÃO EXAMINADORA

Raul Ceretta Nunes, Dr. (Orientador)

Luis Felipe Dias Lopes, Dr. (UFSM)

Roseclea Duarte Medina, Dra. (UFSM)

Santa Maria, 2011.

AGRADECIMENTOS

Agradeço a minha família pelo apoio irrestrito, ao meu orientador que sempre fez encontrar o melhor caminho para resolver os problemas encontrados no decorrer do desenvolvimento do projeto e aos meus colegas de grupo de pesquisa que trouxeram força nos momentos de dificuldade.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	– *Application Programming Interface*
AR	– Modelo Auto-Regressivo
ARMA	– Modelo Auto-Regressivo e Média Móveis
ARIMA	– Modelo Auto-Regressivo Integrado de Médias Móveis
DDoS	– *Distributed Denial of Services*
DoS	– *Denial of Services*
GLR	– *Generalized Likelihood Ratio*
HTTP	– *Hypertext Transfer Protocol*
HIDS	– *Host Intrusion Detection System*
IDS	– *Intrusion Detection System*
ICMP	– *Internet Control Message Protocol*
IP	– *Internet Protocol*
IPSec	– *Security Internet Protocol*
LAN	– *Local Area Network*
OSI	– *Open Systems Interconnection*
MA	– Modelo de Médias Móveis
NCP	– *Network Control Protocol*
NIDS	– *Network Intrusion Detection System*
RFC	– *Request for Comments*
SYN	– Pacote de sincronização do TCP
SNMP	– *Simple Network Management Protocol*
TCP	– *Transmission Control Protocol*
UDP	– *User Datagram Protocol*

RESUMO

Dissertação de Mestrado

Programa de Pós-Graduação em Engenharia de Produção

Universidade Federal de Santa Maria

DETECÇÃO DE INTRUSÃO ATRAVÉS DA ANÁLISE DE SÉRIES TEMPORAIS E CORRELAÇÃO DO TRÁFEGO DE REDE

AUTOR: Francisco Carlos Vogt
ORIENTADOR: Prof. Raul Ceretta Nunes

Data e Local da Defesa: Santa Maria, 09 de Dezembro de 2011.

Este trabalho apresenta um modelo para identificação de anomalias no comportamento da rede de computadores, aplicado ao problema de gestão do tráfego de redes e segurança da informação. Devido à característica de crescimento de tráfego, alguns modelos não diferenciam anomalias de um ataque, gerando falsos positivos prejudiciais a segurança da rede e conseqüentemente a sua qualidade serviço. Com fim de apresentar uma alternativa, este trabalho explora o modelo ARIMA, que permite tornar estacionária a série temporal, e o algoritmo CUSUM, que permite detectar anomalias. Esta abordagem possibilita avaliar com melhor qualidade o comportamento e a identificação de uma anomalia a partir de variáveis descritoras de tráfego e suas correlações. Os resultados demonstram que a abordagem exige uma etapa criteriosa de seleção de variáveis que podem ser influenciadas pelos ataques de interesse.

Palavras-chave: Detecção de Anomalias**;** Detecção de Ataques; Séries Temporais.

ABSTRACT

Master Dissertation

Graduate Program in Production Engineering

Federal University of Santa Maria

INTRUSION DETECTION THROUGH TIME SERIES ANALYSIS AND NETWORK TRAFFIC CORRELATION

AUTHOR: Francisco Carlos Vogt
ADVISOR: Prof. Raul Ceretta Nunes

Date and Local: December 9th, 2011, Santa Maria/Brazil.

This work presents a model to identify anomalies in the computer network behavior applied to the problem of traffic management and security information. Due to the feature of the traffic growth, some models do not differ an anomaly from an attack, generating false positives that damage the security and quality service of the network. In order to present an alternative, this work explores ARIMA model that allows turning stationary the time series and the CUSUM algorithm that allows to detect anomalies. This approach provides a way to evaluate the behavior and identification of an anomaly with better quality through the traffic variables and its correlations. The results demonstrate the approach demands a careful step of variables selection that can have influence by interest's attacks.

Key-words: Anomaly Detection**;** Attack Detection; Time Series.

Sumário

1 INTRODUÇÃO

A internet é uma ferramenta de comunicação muito ágil e dinâmica que permite a realização de diferentes tarefas para seus usuários. Seu uso possibilita inovar a forma de comunicação das empresas e o tratamento dado a informação, possibilitando a conquista de novos mercados e a simplificação das tarefas e permite agilizar os processos empresariais (BORNELY, 2010). Porém, nos negócios "virtuais", como e-comerce, que usam a internet, a agilidade e dinâmica obtidas pelas empresas e seus usuários podem gerar cobiça, principalmente por parte de concorrentes ou pessoas mal intencionadas, o que fomenta a exploração de vulnerabilidades das aplicações e da rede de computadores.

Dentre as facilidades prometidas, tem-se a possibilidade de encontrar acesso a comunicações, informações, dados oficiais, efetuar compras em lojas virtuais, transações bancárias. Algumas das facilidades encontradas são seguras por si só, não geram risco para as empresas, porém algumas delas, quando não protegidas adequadamente podem trazer prejuízos significativos a empresas e seus clientes. Deste modo, o uso da internet nas organizações deve ser cuidadosamente planejado de forma que os riscos sejam minimizados.

Atualmente, um número significativo de empresas opera no segmento de vendas pela internet. Neste segmento, um ataque pela internet pode deixar o site de vendas indisponível, bem como os demais canais de comunicação com os clientes. Um site de vendas paralisado por alguns minutos pode gerar milhares de reais de prejuízo em vendas que deixam de ser realizadas, em potenciais vendas resultantes de pesquisas para futuras compras ou ainda em clientes que deixam de apostar em sua marca por não conseguirem ter uma experiência agradável de consumo (KAHNEY, 2008). Enfim, a interrupção dos serviços web prestados pelos servidores de aplicação pode inviabilizar todo o comércio realizado por uma loja virtual.

De acordo com a agência de notícias IGNOW![1], o tipo de ataque mais popular em 2010 foi o de negação de serviço (*Denail of Service* - DoS), o qual consome os recursos disponíveis na internet, causando a indisponibilidade do serviço web (PENG, LECKIE e RAMAMOHANARAO, 2007). Um DoS normalmente resulta do uso abusivo de uma canal ou porta de comunicação de rede através do envio massivo de mensagens originadas de um emissor de alta capacidade de envio de mensagens. Para que seus efeitos tornem-se mais eficientes e capazes de desativar servidores preparados para receber grande quantidade requisições, uma versão distribuída do ataque (*Distributed Denail of Service* - DDoS), com vários emissores, pode ser utilizada (CABRERA et al., 2002).

A grande quantidade de requisições de conexão recebidas por um servidor pode representar um ataque DoS ou DDoS (tráfego anômalo), mas também há situações onde a rede apresenta-se somente congestionada por requisições legais (tráfego normal[2]). A distinção de um tráfego normal de um anômalo, gerado por um ataque, é então essencial para evitar indisponibilidade dos serviços oferecidos pela internet e pode ser feita modelando o tráfego da rede com posterior realização de análise de variabilidade entre um tráfego de referência e um tráfego recém amostrado. O desafio é obter uma boa taxa de acerto de indicadores de anormalidade (verdadeiros positivos).

A modelagem do tráfego de rede depende de suas propriedades. A propriedade de auto-similaridade do tráfego de rede indica que a rede assume padrões comportamentais que se mantém em várias escalas de tempo (na faixa de *milisegundos* a *minutos* ou *horas*). A propriedade de memória longa (*Long Range Dependence* – LRD), por outro lado, indica que o tráfego apresenta auto-correlação em passos grandes. Estas propriedades sugerem a modelagem utilizando séries temporais (MORETTIN e TOLOI, 2004). Com séries temporais, o comportamento normal da rede pode ser comparado com uma curva de tráfego atual e indicar a possibilidade de estar havendo alterações que sinalizem um possível ataque a rede. Esta dissertação explora esta abordagem para detectar ataques de negação de serviço, a fim de evitar quebras na produtividade de empresas que operam com base na internet.

---

[1] http://idgnow.uol.com.br/seguranca/2011/03/15/ataques-de-negacao-de-servico-foram-os-mais-populares-de-2010/

[2] O termo "normal", neste contexto, não refere-se ao tráfego com distribuição de probabilidades Normal, mas sim ao tráfego legítimo, sem perturbações estatísticas, não anômalo.

1.1 Definição do Problema

Sistemas de Detecção de Intrusão (*Intrusion Detection Systems* - IDSs) são ferramentas bem conhecidas para detectar ataques e atividades maliciosas em sistemas de computação e redes de computadores. Segundo Debar (2000) há basicamente duas classes de IDSs, os baseados na detecção de assinaturas e os baseados na detecção de anomalias. Cada uma destas classes apresenta vantagens e deficiências. Na baseada em assinaturas a principal vantagem é a precisão na detecção de uma assinatura conhecida, porém a deficiência esta associada a ataques que fogem do padrão definido pela sua assinatura, fato que impossibilita detectar variações de ataques, pois a base de assinaturas não apresenta informações que identifiquem o dado ataque. Na baseada em anomalias a principal vantagem é a possibilidade de detecção de ataques desconhecidos (HOLANDA, MAIA e CARMO, 2007). Porém limitar o número de falsos positivos maximizando os verdadeiros positivos é o principal desafio. Adicionalmente, cada classe pode ainda olhar para dados manipulados num dado computador (nomeado *host-based*_IDS) ou trafegados na rede (nomeado *network-based* IDS). Este trabalho foca na abordagem de detecção de anomalias de tráfego de pacotes, onde algoritmos de detecção analisam o comportamento habitual da rede como base para identificar alterações no tráfego de rede. Comportamentos que superem limiares de aceitação podem corresponder a tráfego anômalo gerado por aplicações maliciosas.

O modelo de detecção por comparação de tráfego gera um número grande de detecções e muitas destas detecções não correspondem a um ataque a rede. O desafio é então encontrar algoritmos que apresentem melhor qualidade de detecção sem comprometer a segurança da rede empresarial. Em outras palavras, para que seja melhorada a qualidade de descoberta de incidentes de rede, especialmente os ataques do tipo negação de serviço, é necessário reduzir a quantidade de falsos positivos gerados pelos algoritmos de detecção de intrusão.

1.2 Objetivos e Contribuições

O tráfego de rede é formado pela soma de todos os pacotes que trafegam pelas interfaces de rede, sendo que os pacotes são originados das comunicações estabelecidas pelos protocolos de comunicação implementados pelo sistema. Cada protocolo apresenta características referentes ao comportamento do fluxo de dados e serviços por ele gerenciados, e um ataque de negação de serviço normalmente gera tráfego anormal nos contadores. Deste modo, este trabalho tem por objetivo, explorar o comportamento do fluxo de dados para verificar se houve um incremento significativo na quantidade de pacotes que trafegaram por um determinado protocolo.

Para avaliar a presença de anomalias e ataques, explora-se análise de séries temporais aplicadas ao tráfego rede (contadores de pacotes dos protocolos). Busca-se através do emprego das séries temporais identificar o modelo que descreve o comportamento dos dados que trafegam na rede. O trabalho de Thottan e Ji (1998) emprega o modelo auto-regressivo de primeira ordem (AR 1) para modelar a rede, porém o modelo auto-regressivo não apresenta diferenciações que permitem manipular crescimento exponencial do tráfego de rede. Este trabalho trata esta limitação empregando modelos de séries temporais ARIMA. O modelo ARIMA emprega termos auto-regressivos, de médias móveis e de diferenciações. Desta forma são considerados valores do passado (recente e/ou remoto) do tráfego de rede, valores atuais e diferenciações responsáveis por remover as características de não estacionaridade encontradas nas séries e com isso melhorar a qualidade de detecção (MOROTIN e TOLOI, 2004).

Modelada a série temporal, este trabalho identifica mudanças abruptas do tráfego por meio do emprego do algoritmo (CUSUM), de somas acumuladas, (BASSEVILLE e NIKIFOROV, 1993).

O seu emprego pode identificar saltos que podem descrever dois cenários distintos de anomalia: aplicações legítimas consumindo a banda de rede ou a presença de um ataque em andamento.

O emprego do CUSUM possibilita encontrar comportamentos anômalos individualmente nas variáveis descritora de tráfego. Porém para determinar um

comportamento anômalo que esta sendo causado por um ataque, propõe-se também que se verifique a correlação entre os descritores de tráfego distintos.

A correlação tem por objetivo determinar se o grau de anomalia encontrado numa variável descritora de tráfego é verificado também em outras variáveis, o que amplifica a possibilidade de um comportamento de ataque. Em alguns ataques o comportamento de outras variáveis descritoras de tráfego pode sofrer "contaminação" de outros serviços com o comportamento oriundo do ataque.

1.3 Metodologia

Nesta seção é apresentada a metodologia para a exploração de detecção de intrusão. Para entender o desafio proposto para detecção de intrusão realizaremos os seguintes passos: revisão bibliográfica sobre os tipos de ataques que alteram o comportamento de tráfego de redes, as características das ferramentas de detecção de intrusão e as séries temporais, técnica utilizada neste trabalho para a identificação de anomalias de tráfego.

Dentre os sistemas de detecção de intrusão existem algoritmos baseados em diversas técnicas. A revisão bibliográfica apresenta algumas abordagens empregadas, entre elas temos bayesianas, redes neurais, métodos baseado em estatística como séries temporais e séries temporais com detecção de alterações abruptas.

Baseado em séries temporais e sistemas de detecção abrupta é proposto um algoritmo que detecta anomalias de rede com ARIMA(p,d,q) que se ajusta as necessidades dadas pelo tráfego de rede de forma dinâmica. Devido às características do ARIMA de melhor adaptação ao comportamento dos dados e formar um modelo adequado ao tráfego.

O modelo com maior precisão fornece dados de melhor qualidade para o detector de abruto de anomalias. Quando uma anomalia é detectada o algoritmo entra em sua segunda fase, que é formada pela tentativa de confirmação da contaminação do tráfego de rede de outros protocolos. A confirmação do ataque é dada pela exploração da correlação que identifica se outros indicadores de anormalidade apresentaram dados indicativos de comportamento de anomalia traduzindo-se em ataque.

Para validar o algoritmo foi desenvolvida uma aplicação responsável por realizar a tarefa de detecção das duas fases do algoritmo. No capítulo 6 serão encontradas informações e detalhes para o desenvolvimento da aplicação responsável pela detecção das anomalias e da correlação dos indicadores de tráfego de cada protocolo avaliado.

No capítulo 7 são descritos os testes da aplicação proposta para o algoritmo proposto. Os testes visam validar a eficiência das duas fases encontradas no algoritmo. Os resultados obtidos serão contrastado com os obtidos pelo algoritmo baseado em séries temporais AR(1), com o fim de evidenciar a melhoria de qualidade dado pelo emprego de um modelo mais robusto.

1.4 Organização do Texto

O capítulo 2 apresenta a revisão bibliográfica, o cenário de ataques e os tipos de ferramentas empregadas para identificar os ataques em sistemas computacionais.

No capítulo 3 é feita uma revisão sobre séries temporais e seus modelos e suas características para o emprego no problema de detecção de intrusão.

No capítulo 4 é apresentado alguns trabalhos relacionados que sintetizam as abordagens adotadas para a detecção de intrusão através da análise de anomalias.

O capítulo 5 descreve o de detecção de ataque s proposto, o qual emprega series temporais ARIMA e análise de correlação.

No capítulo 6, é descrita a infra-estrutura de software utilizada para implementar o algoritmo proposto, bem como o sistema para aquisição de dados.

No capítulo 7 os resultados obtidos com o emprego do algoritmo de identificação de anomalias e correlação de variáveis são analisados e discutidos

Finalmente o trabalho apresenta suas conclusões gerais no capitulo 8.

2 AMEAÇAS E ATAQUES

Em ações criminosas realizadas por *crackers, que* disparam ataques que podem ser dos mais diversos tipos e com diversos objetivos. Em tentativas de burlar as medidas de segurança podem ter objetivo de reduzir a disponibilidade do sistema, produzindo prejuízos ou mesmo invadir o sistema para aquisição de algum conhecimento tido como estratégico e de alto valor para empresas concorrentes.

Nesta seção será dado um *overview* dos tipos ataques, para ter–se conhecimento de como é realizado, e em especial os ataques responsáveis pela variação no volume de trafego, técnicas e protocolos afetados são necessários para a compreensão do ambiente em que o trabalho inserido.

2.1 Técnicas de Preparação de Ataques

Um ataque efetivo deve ser preparado, utilizando algumas ferramentas e técnicas para descobrir as vulnerabilidades do sistema alvo. Estas ferramentas precisam descobrir informações que descrevem as características da rede e quais são as portas de entrada que estão abertas. A descoberta das portas evidencia de serviços que são prestados, destes é possível conhecer as ferramentas que são utilizadas e desta forma conhecer a forma de exploração de algumas vulnerabilidades não corrigidas de aplicações. Por esses motivos temos de conhecer algumas técnicas básicas utilizadas. Estas técnicas são *IP Spoofing* (WANG; ZHANG e SHIN, 2002) e varredura de portas.

2.1.1 *IP Spoofing*

Um *IP Spoofing* não pode ser caracterizado como um ataque direto, pois seu uso é dado como uma técnica para a realização de ataques como, por exemplo, um ataque

SYN, *Smurfs*, dos e DDoS, é necessário conhecer a prática. O *IP Spoofing* (WANG; ZHANG e SHIN, 2002) é realizado quando um computador passa a enviar pacotes IP com o endereço falsificado, passando-se por outra máquina da rede. Ocorre por conta da vulnerabilidade do protocolo IP que permite a manipulação do cabeçalho do datagrama, onde é alterado o campo que contém o endereço de origem. A falsificação ocorre porque quando o pacote passa por um roteador e é verificado somente o destino do pacote, não ocorrendo validação da origem do pacote. Essa técnica é conhecida por desvio de sessão TCP, ou *TCP session hijacking*.

Como o IPv4 está sendo substituído pelo IPv6, foram incorporados mecanismos de segurança que garantem um melhor funcionamento. Entretanto, os mecanismos de segurança ainda não são suficientes para evitar todos os ataques. Esta colocação é devido ao fato que o IPSec, apesar de possuir criptografia que permite criar "túneis seguros", pode vir a apresentar falhas e sua utilização não é obrigatória.

A exploração das falhas do IPv6, especialmente no AH (*Authentication Header*), acontece quando são suprimidas a utilização de algumas das ferramentas, como autenticação e confiabilidade, por conta da não obrigatoriedade de uso. Este fato permite que um pacote seja alterado modificando aos dados do cabeçalho. A modificação no cabeçalho requer que o atacante calcule o *hash* do AH para o novo pacote falsificado (SENA; GEUS e AUGUSTO, 2002), caso o pacote não possuir o mesmo *hash* ele será identificado como forjado e descartado.

A forma mais simples de resolução do problema de confidencialidade na transmissão de pacotes é a utilização do cabeçalho ESP, que realiza a cifragem dos dados. Este fato não descarta o fato da possibilidade de ataque, mesmo que não ter acesso às informações, que é feita através injeção de dados falsos, o que pode impossibilitar a comunicação pela perda de pacotes (SENA, GEUS, AUGUSTO, 2002).

2.1.2 Varredura de Porta

A varredura de porta ou (*port scanning*) (HARRIS e HUNT, 1999), consiste no fato de que um atacante pode tentar conectar a portas TCP e UDP procurando por serviços remotos disponibilizados pela rede. Este processo faz parte de um

planejamento de ataques e é essencial se conhecer o sistema operacional e aplicações utilizadas. Com estas informações o atacante pode procurar falhas nas aplicações utilizadas ou simplesmente aproveitar-se de erros da configuração dos sistemas. Algumas técnicas utilizadas por ferramentas de varredura de portas são:

- **Varredura de conexão TCP:** Utiliza-se de do envio de envia três pacotes (SYN, SYN/ACK, ACK) a uma porta TCP. O pacote SYN mostra a intenção do invasor em se conectar, o pacote SYN/ACK, enviado pelo alvo determina que o sistema ouviu a requisição, o pacote ACK do invasor demonstra que a mensagem foi recebida.
- **Varredura TCP/SYN:** A técnica é chamada de varredura semi-aberta, devido ao fato que não há uma conexão TCP completa. Em vez disso, um pacote SYN é enviado à porta-alvo. Se um SYN/ACK for recebido da porta-alvo, é possível deduzir que ela está ouvindo, se um SYN/RST for recebido, normalmente isso indica que ela não está *escutando*. Um RST/ACK será enviado ao alvo, de forma que uma conexão completa nunca seja estabelecida. Esta técnica dificulta identificar o invasor.

2.2 Ataques

Os ataques que os sistemas podem sofrer são implementados em diversos protocolos que prestam serviços uteis a manutenção e descoberta de eventuais problemas na rede.

2.2.1 *Ping of Death*

Os pacotes IP são definidos para possuírem um tamanho fixo de 65535 *bytes*, porém o envio de dados é permitido através da fragmentação em mais de um pacote, o valor que excedeu o limite, o que permite um ataque.

O ataque, conhecido como *Ping of Death* (PENG; LECKIE e RAMAMOHANARAO, 2007), é o mesmo que um simples comando ***ping: ping* -t -l 65535**. Este comando envia pacotes que necessitam utilizar a fragmentação ICMP *echo,*

que é dada em pacotes que estão acima dos 65535 *bytes*. A pilha de rede do sistema vítima do ataque necessita remontar o pacote armazenado em um *buffer* que pode não estar disponível. Isto pode causar a falha do sistema, sua queda e a retomada da operação do sistema.

2.2.2 *Ping Flood Attack*

Ping Flood Attack (PENG; LECKIE e RAMAMOHANARAO, 2007) é uma técnica de ataque que tenta saturar uma conexão de Internet através do envio contínuo de uma serie de *pings* originados tipicamente em redes de alta velocidade para redes de baixa velocidade. Com o recebimento de um grande número de ICMP *Echo Request* (*ping*) por parte do servidor que está sendo atacado, faz com que o mesmo perca tempo e gaste seus recursos tentando responder todos os pacotes recebidos com ICMP *Echo Reply* (*ping*). Dependendo da rapidez e quantidade dos pacotes recebidos pelo servidor, a conexão de acesso a internet poderá ser comprometida o que, às vezes, também pode ocasionar à re-inicialização do servidor.

2.2.3 *Smurfs*

O ataque *Smurfs* (CERTc, 1996) utiliza um IP *spoofing* e *broadcast* (LUNARDI, 2008),(ver Figura 2.1) para realizar o ataque. O ataque faz uso do protocolo ICMP, que normalmente é utilizado para verificação do estado da rede. Para a realização do ataque um único pacote ICMP *Echo Request* é enviado como um direcionado broadcast para uma sub-rede na Internet. Todos os computadores naquela sub-rede responderão para o endereço que o atacante falsificou que normalmente é o utilizado pela vitima escolhida pelo atacante (ver figura). Dessa maneira, quando os computadores que receberem o *broadcast*, responderão com ICMP *Echo Reply* para o endereço IP falsificado contido naquele broadcast.

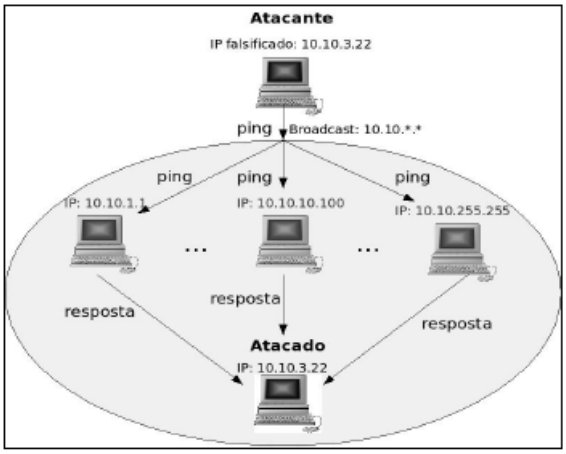

Figura 2.1 - *Smurfs Attack* Lunardi, (2008).

Dependendo do número de computadores naquela sub-rede dezenas, centenas ou ate milhares de pacotes ICMP *Echo Reply* serão enviado para o endereço IP da vítima fazendo com que a conexão torne-se indisponível ou lenta. Essa técnica pode ser aplicada em conjunto com vários outros atacantes para que o efeito tenha maior repercussão.

2.2.4 ARP *Spoofing*

A tarefa realizada pelo protocolo ARP é a de encontrar os endereços físicos dos nós da rede, muitas das vezes esta requisição (*ARP Request*) é realizada em *broadcast* (na versão IPv4) e recebe a resposta (*ARP Reply*) (GONDIM e CARNUT, 2003) através de *unicast*. O protocolo não realiza controle sobre as requisições e respostas recebidas, permitindo o recebimento de respostas não desejadas, assim o ataque de ARP *spoofing* se utiliza dessa vulnerabilidade para comprometer o sistema.

Esse ataque também ficou conhecido como homem do meio (*Man-in-the-middle)* por escutar o tráfego que flui entre os comunicantes, vítimas, ou ainda por ser utilizado para realizar um ataque de negação de serviço (DoS) (GONDIM e CARNUT, 2003).

O ARP é realizado através do envio de uma *ARP Reply* forjado para outra máquina, informando o endereço IP de outra vítima, associado ao endereço MAC do

atacante. Depois de realizado o envenenamento do cachê, os dados enviados pela vítima serão transmitidos para o endereço IP informado no *ARP Reply***,** forjado pelo atacante.

O ataque pode ser realizado de duas formas diferentes, unidirecional e bidirecional. O ataque unidirecional é realizado quando o ARP Reply é enviado para somente uma das vítimas e o tráfego é escutado de somente de um dos lados, dado pelo envenenamento de somente uma ARP Cachê. O ataque bidirecional (GONDIM e CARNUT, 2003) ocorre quando ambas as vítimas tem a ARP Cachê envenenada, tornando-o capaz de ouvir o tráfego das duas vítimas.

As informações no ARP Cachê são alteradas após um intervalo de tempo, podendo variar de acordo com a implementação do sistema operacional. Para que a conexão seja mantida, de tempos em tempos o atacante envia pacotes para a vítima, com o objetivo de manter a cachê envenenado.

Esse ataque faz uso de brechas de implementações do ARP cachês nos sistemas operacionais, para mandar os pacotes no momento correto, o que torna difícil a detecção por IDS. Mesmo em implementações mais recentes (TRABELSI e EL-HAJJ, 2007), encontram-se sob a noção de estado, ou seja, aceitando apenas respostas para requisições feitas, o ataque ainda ocorre, utilizando a combinação de ICMP e ARP.

2.2.5 NDP *Spoofing*

Apesar de que o protocolo NDP ser adotado com o uso do IPv6, sua tarefa é semelhante ao do protocolo ARP. Por este motivo, o NDP herdou sua vulnerabilidades, que é dada através da falsificação de cabeçalhos do protocolo IP e IPv6 permitindo este tipo de ataque.

Os ataques ARP *Spoofing* com IPv6 passaram a explorar o NDP quase da mesma forma como era ARP, a única alteração no ataque fica por conta do modo operacional como o NDP realiza a sua tarefa. Com a um ataque NDP *Spoofing* bem sucedido é possível conhecer e alterar rotas, o que pode propiciar e facilitar um ataque DoS.

Também existe a possibilidade de realizar a escuta de todo o tráfego, quando uma máquina torna-se roteador da rede, o que permite que outras formas de ataques possam ser geradas por uso do *NDP Spoofing*. Para realizar esse ataque o *host* do

atacante insere-se na rede do alvo como um *host* externo. O equipamento do atacante passa ser um nó da rede, armazenado na tabela de vizinhos de conhecidos de um *host* da rede local. Esse ataque permite que sejam lançados outros tipos de ataques como ataques DoS e Homem do Meio.

2.2.6 *SYN Attack*

No capitulo anterior foi apresentados os protocolos e o pontos que podem apresentar vulnerabilidades, dentre os protocolos o TCP é um dos protocolos mais utilizado na realização de conexões. Ao conectar via TCP é necessária que siga os "rituais" estabelecidos pela RFC 793 (RFC 793, 1981).

No procedimento utilizado para estabelecer uma conexão são empregadas *flags*, vista no capitulo anterior, sinalizando o tipo de pacote, que pode ser uma requisição de uma conexão ou o seu fim. Entretanto, a implementação das regras propicia brechas de segurança no protocolo.

A brecha de segurança junto ao SYN (CERTB, 1996) ocorre quando um atacante, utilizando *IP spoofing*, envia pacotes sinalizados com SNY de forma que "inunda" o servidor com requisições de conexão (ver figura 2.2). O servidor recebe o pacote e realiza o passo seguinte, que é alocação da conexão em uma tabela de *half connection* e o servidor envia um pacote contendo SYN/ACK ao cliente (ver Figura 2.2). Como o cliente não pode ser alcançado devido ao endereço IP falsificado o servidor não receberá um SYN/ACK do cliente. Durante um período de 75 segundos o servidor ficará esperando pela resposta do cliente (WANG; ZHANG e SHIN, 2002).

Existem estudos que dizem que *firewalls* especializados para a tarefa de conter ataques SYN não são capazes de suportar um número elevado de requisições SYN. Se o número de requisições superarem o número de 14.000 pacotes por segundo (DARMOHRAY e OLIVER, 2000) o *firewall* torna-se inoperante e permitindo que o sistema alvo seja atacado e derrubado. Apesar de que (DARMOHRAY e OLIVER, 2000) diz que um *firewall* não ser capaz de resistir a mais de 14.000 pacotes por segundo, essa informação reporta ao uso de equipamentos obsoletos comparando com a capacidade dos atuais equipamentos, onde os *firewalls* estão instalados atualmente.

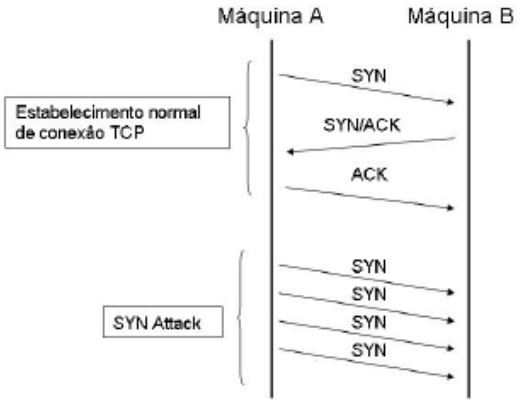

Figura 2.2 - Ataque SYN Lunardi (2008).

2.2.7 Ataques usando UDP

Pacotes UDP originalmente não necessitam que uma conexão seja gerada efetivamente, não ocorrendo o estabelecimento de uma conexão, além de que o protocolo não possuiu o seqüenciamento dos números utilizados pelo protocolo TCP. Essas características são determinantes quando se fala de segurança de pacotes trafegando, pois é possível que uma máquina atacante falsifique um endereço IP realizando um IP *Spoofing*. No protocolo UDP é possível gerar tráfego capaz de superlotar o *buffer* do receptor efetivamente inviabilizando o tráfego de dados para este *host.* O ataque usando UDP é muito conhecido como UDP *Flood attack.*

O UDP *Flood attack* (CERTA, 1998), (SPECHT e LEE, 2004) é diferente do TCP SYN attack visto que o protocolo UDP não faz o processo de *Three Way Handshake* para efetuar a conexão. Então já que o protocolo UDP não oferece nenhum meio de garantia de entrega do pacote, o atacante simplesmente envia pacotes de UDP randomicamente para todas as portas do servidor da vítima. Quando o servidor da vítima recebe os pacotes de UDP enviado pelo atacante, ele tenta determinar qual aplicação está aguardando pelo pacote naquela determinada porta em que foi recebido o pacote. Porém, quando o servidor verifica que não existe nenhuma aplicação aguardando tal pacote recebido, ele então emite um pacote ICMP para o destinatário (que obviamente forjou o endereço de IP da fonte com um IP *Spoofing*) dizendo que o

pacote não encontrou seu destino. Se uma grande quantidade de pacotes de UDP enviados para as portas do servidor da vítima, o sistema poderá ser comprometido com a queda do sistema alvo.

2.2.8 DoS

Os ataques DoS ou *Denial of Services* são classificados como ataques que visam retirar o serviço de um site ou aplicação, mantida em um endereço, do ar (PENG; LECKIE e RAMAMOHANARAO, 2007). Esse tipo de ataque utiliza-se de vulnerabilidades apresentadas nos protocolos para que seja efetuado. Por exemplo, um ataque do tipo SYN pode ser utilizado por um atacante para que seja atingida a meta, ou mesmo outros tipos ataques.

Definir um ataque DoS como um envio maciço de mensagens e requisições a um servidor efetuado por um atacante, que por sua vez domina um determinado host que possui um endereço IP falsificado pelo atacante. Desse endereço falsificado partem inúmeras mensagens que superlotaram a vítima, que na tentativa de responder ao host inexistente, se tornará inoperante.

2.2.9 DDoS

O ataque DDoS ou *Distributed Denial of Services*, como o DoS tem por objetivo derrubar uma aplicação ou serviço distribuídos em uma empresa. Porém a diferença entre as versões desse ataque é a forma de como ele é realizado. O DoS utiliza uma única máquina zumbi controlada pelo atacante, em contrapartida, na versão distribuída, centenas ou milhares de máquinas zumbis são empregadas para a realização do ataque PENG; LECKIE e RAMAMOHANARAO, 2007).

Como em sistemas distribuídos legítimos os ataques DDoS utilizam recursos de diversos computadores para disponibilizar serviços e realizar conexões, para que um sistema distribuído seja criado, para o uso em um ataque, os computadores são “recrutados” pelo atacante no momento que é instalado um software de controle remoto,

o que torna o equipamento um zumbi. Uma rede formada por zumbis é conhecida como uma *botnet* (SPECHT e LEE, 2004), com recursos computacionais distribuídos (PENG; LECKIE e RAMAMOHANARAO, 2007). Na *botnet,* cada zumbi torna-se responsável pelo envio de mensagens e requisições de conexões, de acordo com o tipo de ataque empregado (Figura 2.3), a um *host* escolhido pelo atacante. A quantidade de mensagens geradas pelo sistema distribuído visa superar os recursos disponíveis na rede de servidores atacados.

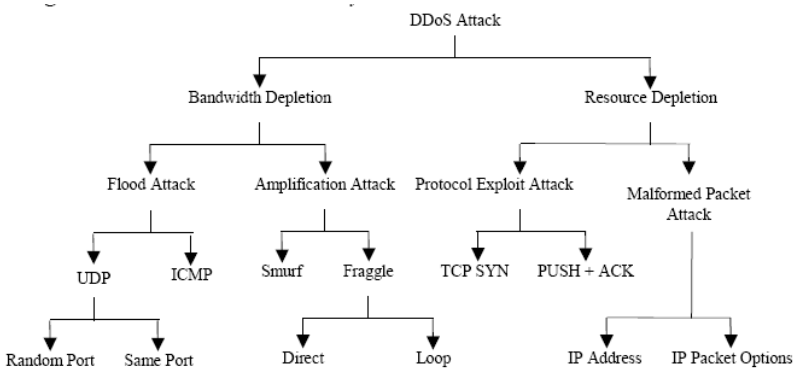

Figura 2.3 - Taxonomia do ataques DDoS Specht e Lee (2004).

Em 1999 foram registrados os primeiros ataques que utilizavam o formato distribuído como ferramenta para realizar o ataque a sites como Yahoo (SPECHT, LEE, 2004), Ebay, Amazon e CNN, que quando atacados ficam por horas sem poder fornecer seus serviços. Na semana seguinte, sites brasileiros como UOL, Globo On e IG foram alvos desses ataques (SOLHA: TEIXEIRA e PICCOLINI, 2000).

Ataques vêm sendo registrados desde 1999 pelo CERT/CC, que registra ataques diários a máquinas de usuários comuns distribuídas pela internet. Os equipamentos atacados não possuem sistemas de segurança em suas conexões e são contaminados por vermes que, uma vez instalados, constroem *botnets* capazes de realizar um ataque de negação de serviço distribuído.

As *botnets* compostas pelas ferramentas de ataque DDoS são formadas por diferentes atores responsáveis por diferentes ações e atividades na rede. As *botnets* possuem diversos atacantes (Figura 2.4), mestres, agentes e o alvo do ataque (SPECHT e LEE, 2004). A *botnet* é composta por centenas ou milhares máquinas zumbis, que são

infectadas por uma aplicação que abre portas de comunicação de um equipamento de um usuário na internet.

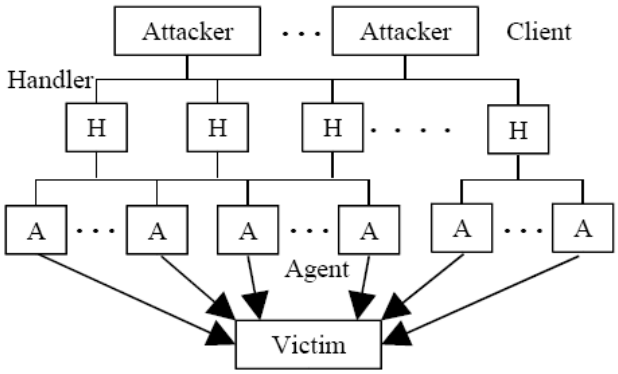

Figura 2.4 Arquitetura de um ataque DDoS Specht, Lee (2004).

Cada zumbi utiliza ferramentas de *spoofing*, responsáveis por aumentar a eficiência do ataque. Os zumbis recebem comando que inicializam um determinado tipo de ataque a um alvo estabelecido pelo atacante. O grupo de zumbis é controlado por um *host* específico que possui um *software* zumbi que permite controlar outros zumbis, esse também utiliza *IP spoofing*, porém o acesso é feito diretamente pelo atacante, que dispara comandos de inicialização de um tipo de ataque a determinado site repassando a tarefa aos zumbis pertencentes à rede da *botnet*.

2.3 Sistemas de Detecção de Intrusão

Em redes corporativas e de pesquisa são geradas informações estratégicas sensíveis ao funcionamento da corporação e, por muitas vezes, a rede não possui mecanismos de segurança adequados, o que torna a rede alvo de intrusão. Com o objetivo de reconhecer padrões de ataques através da comparação com predefinições ou diferenciações do padrão normal da utilização de uma rede, os detectores de intrusão são importantes ferramentas para realizar a tarefa. Devido à importância de conhecer os detectores de intrusão este capítulo apresenta os tipos características particulares de cada um deles.

2.3.1 HIDS - *Host Intrusion Detection System*

Um sistema de detecção de intrusos baseados em *host* (DEBAR, 2000) é construído para realizar o monitoramento de uma máquina, detectando a possibilidade da ocorrência de um ataque pelas características encontradas nas chamadas de sistema (BECKER e PETERMANN, 2005) e acesso aos recursos disponíveis; através deste sistema de detecção de intrusão o sistema pode responder ao usuário sobre as atividades e definir se ocorrerá algum ataque nesse determinado *host*. Basicamente, é possível identificar a detecção de intrusos baseados em estações, que se preocupam com o que acontece dentro do servidor; e em rede, que se preocupam com os dados transmitidos entre os computadores.

Os IDS baseados em *host* operam sobre informações coletadas de uma única estação, permitindo que ocorra precisão na hora de analisar as ocorrências neste nó. Além disso, é capaz de determinar exatamente que processos e usuários estão envolvidos em um ataque, pois é possível acessar e monitorar diretamente os dados e processos do sistema que são possíveis alvos de ataques. O IDS poderá detectar também que programa acessa determinado tipo de informação de forma que venha a garantir que um processador de texto, por exemplo, não irá inesperadamente começar a modificar os dados de um banco de dados de senhas do sistema.

Algumas características de um HIDS são detectadas e evidenciam seus pontos fortes e vulneráveis. Dentre as características positivas de um HIDS, encontrado em todos os nós da rede, encontra-se a que permite um monitoramento de todos os equipamentos pertencentes à rede. A seguir algumas das vantagens e desvantagens dos IDS baseados em host.

V**antagens:**

- Apresentam melhor confiabilidade, pois o monitoramento de eventos locais de uma estação permite detectar ataques que não são detectados por um IDS baseado em rede;
- Podem operar em ambientes onde o tráfego de rede seja criptografado (ANT, 2002);

- IDS baseados em *host* não são afetados por *switches.*

Desvantagens:

- São difíceis de gerenciar, pois para cada *host* monitorado deve possuir um IDS (DEBAR, 2000).
- Fraco desempenho para conteúdos dinâmicos.
- Podem ser desativados por certos tipos de ataques de negação de serviço (ANT, 2002).
- Consomem recursos computacionais dos *hosts* (DEBAR, 2000).

2.3.2 *Network Intrusion Detection System* - NIDS

Sistemas de detecção de intrusos baseados na rede (DIAS, 2006; DEBAR, 2000; SNORT, 2008) usam os pacotes da rede como fonte de dados. Um IDS baseado na rede utiliza tipicamente um adaptador de rede rodando no modo promíscuo, podendo ser colocado em vários pontos, para monitorar e analisar todo o tráfego em tempo real. Desta forma, os IDS são mais facilmente protegidos contra ataques. Muitas das vezes, estes sensores ficam rodando em uma ponte (*bridge*), o que tornar mais difícil a localização do IDS por um *cracker*.

Quando um ataque é detectado, o sistema provém uma variedade de opções de notificação, alerta e tomada de ação em resposta ao ataque. Estas respostas dos IDS envolvem a notificação do administrador, encerramento da conexão e/ou gravação da seção para análise forense e coleta de evidências. Quando se está em um ambiente de difusão, deverá estar conectada ao roteador no modo promíscuo, o que permitirá capturar e analisar qualquer pacote que trafegue pelo meio compartilhado. Em uma rede densa, pode ser custosa a análise de todos os pacotes circulantes.

Quando, ao invés da figura do roteador, tem-se o *switch* (comutador), perde-se a idéia da conexão no modo promíscuo, pois os pacotes que saem do comutador vão diretamente ao destino através de um meio não compartilhado. Neste ambiente, a responsabilidade pelo IDS deverá estar conectada ao comutador através de um sistema

de espelhamento. Ou seja, o comutador irá refletir pacotes que saírem de qualquer porta para a porta onde a unidade do IDS está conectada.

Vantagens:

- Utiliza menos pontos (mas bem posicionados) de rede rodando um IDS, mas para monitorar uma grande rede, o que reflete em um bom desempenho da rede.
- A instalação provoca um pequeno impacto na rede, pois eles são dispositivos passivos que escutam a rede.
- Podem ser bem seguros contra ataques e ainda podem ser invisíveis a muitos atacantes.

Desvantagens:

- Podem ter dificuldades em processar todos os pacotes em uma rede grande e sobrecarregada.
- Muitas das vantagens dos IDS baseados em rede não se aplicam as redes mais modernas baseadas em switches, pois estes subdividem a rede em muitos segmentos.
- IDS baseados em rede não podem analisar informações criptografadas.
- A maioria não pode informar quando um ataque foi bem sucedido. Eles só podem alertar que um ataque foi iniciado, e cabe ao administrador investigar cada host atacado para determinar de fato quais destes foram violados.

2.3.2.1 Detectores baseados em Assinaturas

Quando se fala em detectores de intrusão de rede, têm-se duas possibilidades distintas de execução. A detecção é realizada através da comparação do comportamento da rede com o comportamento estabelecido como de um ataque. Como o comportamento de ataques e sondagens, como *buffer overflow*, *port scans*, ataques CGI (*Common Gateway Interface*), verificação de SMB (*Server Message Block*).

Pode-se dizer que, nesses casos, configura-se uma estrutura composta por diversos elementos combinados, capaz de identificar ataques. A estrutura, na sua grande

maioria, é composta por um banco de dados responsável por armazenar dados coletados e certificados como padrões encontrados em alguns ataques. A este padrão coletado dá-se o nome de assinatura.

Muitas ferramentas comerciais e de software livre são implementadas utilizando o método de assinatura. Um dos motivos da indústria de software adotar assinaturas como ferramenta de detecção se deve ao fato de que muitos ataques podem ser facilmente identificados pelo padrão de distribuição de pacotes, sua assinatura, que trafegam na rede. Tornando um sistema com grande precisão, ou *acurracy*, (DEBAR, 2000) na geração de alarmes de intrusão.

Porém a facilidade de localizar uma tentativa de intrusão necessita que sempre o responsável pelo sistema esteja revisando-o, com objetivo de encontrar novos padrões de ataque dados pelo desvio de tráfego, disponibilizados pelos fabricantes, que evidenciam um ataque e que podem gerar uma nova assinatura devido ao fato que pequenas distorções e modificações do padrão assinado de um ataque podem impedir que, efetivamente, seja detectado como sendo um ataque.

2.3.2.2. Detectores baseados em Anomalias de Rede

As ferramentas para detecção de intrusão baseado em anomalias de rede são diferentes das ferramentas baseadas em assinaturas, pois o modelo utilizado é o do comportamento normal da rede (HOLANDA FILHO; MAIAe CARMO, 2007). O comportamento normal da rede é obtido após períodos de observação da rede em momentos que sua carga não apresenta alterações motivadas por ataques. Com o comportamento normal modelado, variações ocorridas no tráfego normal podem ser analisadas e, em caso de apresentarem distorções, pode-se evidenciar um ataque, o que gera um alerta.

O comportamento normal da rede é modelado como se fosse uma assinatura de ataque, porém essa assinatura é a carga que a rede sofre em determinados horários, dado pelo número de requisições aos serviços do tipo: HTTP, FTP, SNMP, entre outros protocolos de aplicações disponíveis na camada de aplicação. O modelo normal irá adicionar a um contador de pacotes os tipos de requisições para cada tipo de aplicação

disponível, somando e totalizando o número de requisições realizadas para o dado instante de tempo (POHLMANN, 2006).

A utilização do modelo de comportamento de tráfego normal de rede é uma linha base para realizar a comparação com os dados dos contadores de pacotes e, em caso de alterações acima de um limite de tolerância, é possível diagnosticar um ataque, caso ocorra alguma brusca alteração nos contadores de pacotes (POHLMANN, 2006), em um curto período de tempo o IDS pode responder com um alerta ao administrador, para que seja tomada alguma providência em relação à ação suspeita no sistema.

Características encontradas num IDS baseado em anomalias de rede contribuem para a detecção de um ataque com estruturas de pacotes desconhecidos. Diferentemente do modelo baseado em assinatura, vulnerável aos dados correntes, não estão modelados em alguma assinatura.

Entretanto, este tipo de IDS pode gerar um grande número de falsos positivos devido ao fato de que algumas alterações nas taxas de tráfego podem ativar o estado de alerta, além das alterações na quantidade de pacotes, geradas por aplicação e usuários legítimos que não caracterizam uma situação de ataque real. Desta forma, esse tipo de IDS precisa de mecanismos que agreguem mais parâmetros para descrever um ataque de um excesso de carga.

Os alertas nos IDS do tipo de anomalia ou assinatura podem ser enviados ao administrador do sistema de diversas formas: através de mensagens SMS, *e-mail* e chamadas de sistema. O que melhora a qualidade dos dados para a tomada de decisão por parte do responsável.

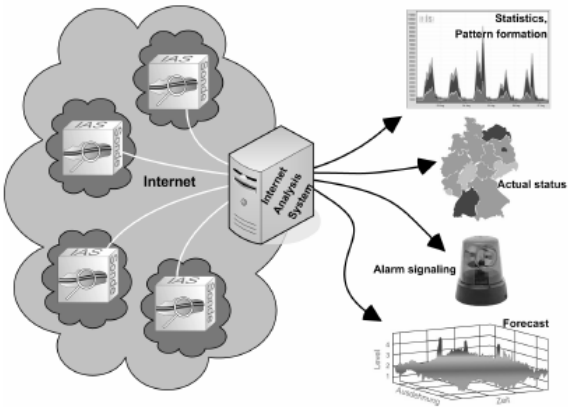

Figura 2.5 – IDS baseado em anomalia de rede Pohlmann (2006).

Na Figura 2.5 é apresentado o desmembramento dos componentes do IDS baseado em anomalia de rede. Os componentes, como já mencionados, é um padrão probabilístico gerado, os dados que são capturados no dado momento, um gerador de alertas e um módulo de previsão.

2.4 Conclusões Parciais

Neste capítulo foi realizado um estudo sobre sistemas de detecção de intrusão, suas características e abordagens de projeto para identificação de ataques, bem realizado um estudo sobre os tipos de ataques no nível de rede utilizados para explorar as vulnerabilidades do sistema.

A classificação dos IDS possibilita detectar a qual grupo pertence o detector proposto neste trabalho e também contribuiu para entender os pontos fortes e fracos de cada abordagem.

O capítulo mostra que ataques de negação de serviço costumam explorar as vulnerabilidades de protocolos como ICMP, UDP e TCP e são muitas vezes construídos com o emprego de ataques mais simples, principalmente em sua versão distribuída. A característica de paralisar os sistemas atacados e de haver diversidade de vulnerabilidades e ataques, o que aponta exploração de diferentes protocolos num mesmo ataque, sugere a busca por ferramentas capazes de identificar ataques que afetam simultaneamente mais de um protocolo.

3 SÉRIES TEMPORAIS

Este capítulo é dedicado à exploração das características da ferramenta estatística utilizada neste trabalho, as Séries Temporais. Uma série temporal nada mais é do que um grupo de dados seqüenciais organizados no tempo. A modelagem utilizada pelas séries temporais visa trazer características que descrevam, expliquem, investiguem, prevejam ou controlem as situações verificadas nos dados. O método de previsão utilizado traz estimativas que apresentam um bom grau de confiabilidade

Neste capítulo, apresentam-se os modelos utilizados pelas séries temporais para explicar o comportamento dos dados. Na seção 3.1 são apresentadas as características gerais das séries temporais; na seção 3.2 são apresentados os modelos que tratam das séries temporais, e na seção 3.3 conclusões parciais do capítulo.

3.1 Características Fundamentais

Uma série temporal é definida como um conjunto de dados de uma variável observada e organizada no tempo (NUNES, 2003). A característica fundamental das séries temporais é a dependência entre os dados, sendo esta dependência o alvo dos modelos de séries temporais.

Os dados obtidos por um contador de pacotes formam uma série temporal na qual se pode verificar a dependência dos dados pelo simples fato que é possível verificar qualquer alteração no trafego de rede e se essa alteração está correlacionada com a progressão do tráfego dos pacotes na rede. Em condições usuais, o tráfego de rede é auto-similar, sendo possível prever o seu comportamento para os mesmos períodos.

Através desta previsão, realizada com base nos dados série temporal, pode-se explicar a variação em outra série, o que se denomina auto-similaridade, e explica o comportamento dos usuários na rede. Entretanto, o comportamento anômalo da rede por um período de tempo pode revelar que a rede está sobre ataque.

3.2 Modelos das Séries Temporais

Neste tópico serão analisados os modelos utilizados para tratar as séries temporais, dentre os quais se podem citar o modelo Auto-regressivo, Médias Móveis, ARMA e ARIMA.

3.2.1 Modelo Auto-Regressivo – (AR(p))

O modelo auto-regressivo representa séries que possuem características auto-regressivas, isto é, os dados anteriores explicam o valor corrente da série. O valor da variável (*p)* é a ordem do modelo, o que explica quantos valores estará sendo empregado no modelo para dizer qual é o valor corrente **(z_t).** A função é formada pelo valor corrente somado a um ruído aleatório $\boldsymbol{a_t}$ (NUNES, 2003).

$$z_t = \phi_1 . z_{t-1.} + \ldots + \phi_p . Z_{t-p} + a_t . \qquad (1)$$

A ordem *p* do processo auto-regressivo indica o número do coeficiente utilizado no modelo.

$$z_t = \frac{1}{\phi(B)} . a_t . \qquad (2)$$

3.2.2 Modelo de Médias Móveis – (MA(q))

Este modelo permite representar processos que empregam médias móveis. O modelo de médias móveis utiliza o parâmetro (***q)*** para determinar quantos termos são utilizados para determinar o valor corrente **(z_t)** da média móvel e para determinar este valor é utilizado um valor de ruído aleatório $\boldsymbol{a_t}$**.** A média móvel é determinada pela expressão:

$$z_t = a_t - \theta_1 . a_t - \ldots - \theta_q . a_{t-q} . \qquad (3)$$

$$z_t = \theta(B).a_t. \tag{4}$$

3.2.3 Modelo Auto-Regressivo e Média Móveis – (ARMA(p,q))

O modelo ARMA permite utilizar termos auto-regressivos $\phi(B)$e termos de médias moveis $\theta(B)$. Esse modelo necessita de dois parâmetros, *p* e *q* que descrevem a ordem dos termos empregados. O modelo ARMA é dado por:

$$z_t = \frac{\theta(B)}{\phi(B)}.a_t. \tag{5}$$

Este modelo é a combinação do modelo AR(*p*) e das MA(*q*), no modelo são necessários poucos coeficientes responsáveis para descrever a série.

3.2.4 Modelo Auto-Regressivo Integrado de Média Móveis – (ARIMA(p,d,q))

O modelo ARIMA, como no modelo ARMA, integra termo AR e termos MA, porém é utilizada uma unidade de integração que permite utilizar séries não estacionárias homogêneas no modelo. São utilizados três valores para determinar a ordem utilizada no modelo *p*, *d* e *q* o modelo é representado por:

$$z_t = \frac{\theta(B)}{\phi(B).\nabla^d}.a_t. \tag{6}$$

O parâmetro (*d)* representa quantas diferenças são necessárias para transformar uma série não estacionária em uma estacionária, é usual utilizar até duas integrações. Se

todos os parâmetros estiverem presentes, será usado o modelo ARIMA, e se houver a omissão de algum parâmetro, o modelo será reduzido ao modelo anterior.

3.3 Conclusões Parciais

As Séries Temporais são utilizadas para descrever as amostras obtidas no decorrer do tempo, obtidas em intervalos de tempo iguais, eqüidistantes, o que permite que seja realizada uma previsão de eventos em um dado momento futuro.

Como já mencionado, a rede apresenta auto-similaridade onde o tráfego apresenta a mesma característica para uma dada hora em outro dia. Então, para uma dada rede, é possível estabelecer uma função que representa o comportamento desta rede. Esta função é obtida através de uma modelagem com base no número de pacotes recebidos durante o intervalo utilizado para estabelecer o comportamento da rede. Através desta função, a característica auto-similar da rede permite que se possa observar a rede e identificar alguma anomalia de tráfego, por conta do desvio da função estabelecida de trafego normal de rede.

O modelo ARIMA é capaz de tratar dados estacionários e dados não estacionários homogêneos e tratar processos que se apresentam, em qualquer ordem dada pelos seus parâmetros.

O presente trabalho visa utilizar as séries temporais para que seja possível identificar anomalias de tráfego e se esta variação é indícios de ataques de negação de serviço.

4 TRABALHOS RELACIONADOS

Neste capítulo apresenta-se uma revisão no que diz respeito a algoritmos de detecção de intrusão que consideram a detecção de anomalia. Serão apresentados algoritmos baseado em redes *Bayesianas* (seção 4.1), em redes neurais (seção 4.2) e em técnicas estatísticas (seção 4.3), mas especificamente o algoritmo de detecção de mudanças abruptas, proposto por Thottan e Ji (2003) (seção 4.4), o qual é está fortemente relacionado a este trabalho.

4.1 Rede Bayesiana

O tráfego da rede de computadores apresenta padrões de comportamento que podem ser modelados por variáveis descritoras de tráfego, sendo reconhecida a característica de incerteza destas variáveis descritoras. Deste modo, modelos probabilísticos podem ser utilizados para detectar anomalias no comportamento do tráfego rede.

Uma técnica que busca modelar o comportamento probabilístico da rede é a de redes bayesianas, pois modelam o comportamento baseado na sua incerteza ou que possuam conhecimento incompleto (CHEBROLU, 2004) (JEMILI, 2007) (KRUEGEL et al., 2003). Assim através de modelos probabilístico redes bayesianas identificam padrões que caracterizam anomalias e ataques.

As redes bayesianas são empregadas como classificadores que avaliam se o comportamento do tráfego é “normal” ou “anormal”. Seu emprego requer que as informações sejam atualizadas constantemente aprimorando o funcionamento da rede de detecção (LERMEN e BRUNO, 2008). O objetivo é classificar as informações desencontradas (ataque) no comportamento de rede como ataques (ZHAI et al., 2004). A cada tentativa de ataque, o padrão de tráfego é alterado, pois assume-se que o ataque modifica o funcionamento da rede. Com base em informações passadas dois grupos de

informação são avaliados, um deles composto por informações de evidências que podem indicar um ataque e outro composto por dados que representam o estado normal do sistema. Os novos comportamentos são então classificados.

4.2 Redes Neurais

Outra técnica para detectar anomalias é usando redes neurais artificiais (RNA) (MAFRA et al., 2008). Uma RNA responde a estímulos através do aprendizado do comportamento do analisado. O aprendizado da rede ocorre através do armazenamento do comportamento do tráfego de rede.

O treinamento da rede é dado pela atribuição de valores para os pesos sinápticos aos eventos do tráfego, especialmente para aqueles que influenciam o fluxo de dados nesta rede. Isto significa para que os pesos sinápticos sejam determinados requer-se um período adequado as suas características de tráfego.

Estabelecido os pesos da rede neural que definem o grau de influência de uma variável para a composição da saída, e assim o conhecimento sobre o tráfego usual de rede. A RNA terá as informações necessárias para a tomada de decisão quando detectar um comportamento anômalo e determinar se este conjunto de dados é oriundo de um ataque.

No trabalho de MAFRA visou melhorar a taxa de acertos em detecções, empregando uma técnica de classificação de eventos em duas categorias, normal e anômalo, e outra para a determinação se o evento é um ataque. A técnica de classificação é dada por uma rede neural construída com pelo método de Kohonen (MAFRA et al., 2008). Em se trabalho a rede foi escolhida por permitir o aprendizado de forma automática e pela facilidade de em separar padrões conhecidos e pela generalização das detecções, o que permite a detecção de variações de tráfego.

O tráfego de rede classificado em uma categoria, a saída desta rede neural é analisada por quatro redes neurais do tipo *Support Vector Machines* (SVM) (MAFRA et al., 2008), que recebem o tráfego do classificador. Essas redes neurais tratam de quatro categorias distintas de ataques.

Outro trabalho aborda a detecção de ataques através de um modelo híbrido de redes neurais RBF/Elman proposto (TONG; WANG e YU, 2009). Este modelo de redes neurais avalia a reação da função exponencial para aproximar uma entrada e a saída não linear (ACOSTA, 1995) e (KALINLI e SAGIROGLU, 2006). As redes neurais do tipo RFB necessitam de menos tempo para serem treinadas (TONG; WANG e YU, 2009) melhorando o tempo de reação para detecção de ataques.

A rede neural realiza a detecção de ataques comparando cada um dos comportamentos individuais e a rede neural hibrida realiza o papel de identificar os comportamentos anormais usando um arquivo conhecido como "*software profile*" (TONG; WANG e YU, 2009) usado no treinamento da rede.

4.3 Modelos Estatísticos empregado para detecção de intrusão

A detecção de intrusões pode ser realizada através do emprego de métodos estatísticos. Os métodos estatísticos avaliam o comportamento da rede por meio de padrões de comportamento com o uso de modelos que o descrevem. Cada modelo do sistema necessita ser construído com atribuição de aprendizado e ser atualizado para que amostras antigas não interfiram na qualidade de detecção de intrusões. Para resolver o problema de detecção através de análise estatística pode-se empregar análise de séries temporais.

Com uso de séries temporais pode-se construir modelos que descrevem o comportamento do tráfego de rede e avaliar se o comportamento de rede extrapola um limiar de tolerância estabelecido, *threshold* (HAJJI, 2003) e (THOTTAN e JI, 2003). O emprego do *threshold* contribui para detecção quando estabelece um limite de quanto o tráfego de rede pode variar, sem que a rede seja classificada como estando sob ataque. Os modelos gerados são responsáveis por apresentar uma previsão de como se dará o fluxo de pacotes na rede que é contrastado com o tráfego efetivo de rede, desta forma apontando anormalidade que são geradas por ataques ou por aplicações que possuem alto consumo de banda.

O trabalho de Thottan e Ji (2003) é uma abordagem de detecção de ataques baseado na identificação de anomalias através da ênfase de séries temporais que considera o modelo auto-regressivo (AR(p)). Identificado o comportamento anômalo da

rede emprega-se um algoritmo que faz o reconhecimento do ataque através da identificação de correlação entre fluxos de dados.

4.4 Séries Temporais e Algoritmos de Detecção Abrupta

No trabalho de Thottan e Ji (2003) a identificação de uma anomalia no comportamento de rede faz uso de uma aplicação de detecção abrupta em cada variável descritora de tráfego analisada, combinado com uma identificação por meio de correlação entre as variáveis. As variáveis são inicialmente processadas por um algoritmo de detecção abrupta, indicando um alarme. Após a identificação de um alarme é realizado a combinação dos alarmes gerados para diferenciar um alarme de um ataque.

A Figura 4.1, apresenta a estrutura do algoritmo de Thottan e Ji (2003). A figura é formada pela a entrada de dados dada pelas variáveis descritoras de tráfego, no seu trabalho representado pela MIB, a unidade que processa a série temporal e envia os dados para o módulo de análise de mudança abrupta responsável por gerar alarmes e a unidade de combinação de alarmes individuais para a determinação de alarmes do sistema.

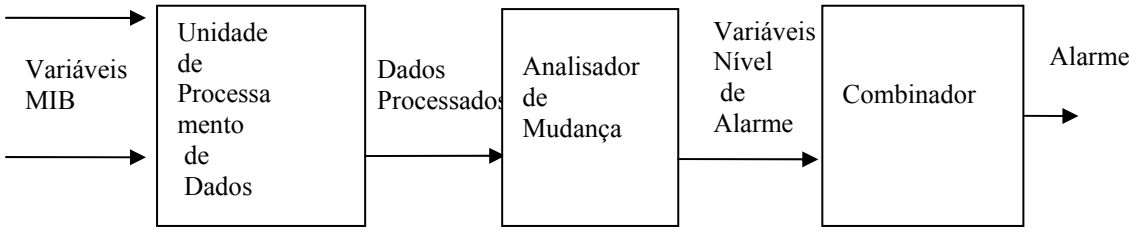

Figura 4.1 Estrutura do Detector de Anomalias Thottan e Ji (2003).

A seção 4.4.1 detalhada o processo de detecção de anomalias e a seção 4.4.2 detalha como é feito a análise de correlação anomalias detectadas em fluxos distintos.

4.4.1 Detecção de Anomalias

Nesta seção são avaliados os processos usados para detecção de anomalias em tráfego de rede pelo uso do algoritmo CUSUM. A seção 4.4.1.1 trás mais detalhes sobre a coleta de dados e a seção 4.4.1.2 sobre o processo de detecção de anomalias.

4.4.1.1 Coleta de Informações

O modelo de identificação de anomalias de rede proposto por Thottan e Ji (2003) faz uso de informações coletadas por ferramentas de gerenciamento de rede através do protocolo SNMP. O protocolo SNMP utiliza variáveis que armazenam o resultado da contagem de pacotes realizada pelo agente presente em um host da rede. Através destas informações permite-se aplicar algoritmos que possibilitam caracterizar o comportamento da rede e também identificar anomalias (THOTTAN e JI, 1998, 2003) (WU e SHAO, 2005).

Após definido o comportamento da rede (tráfego usual sem ataque), o algoritmo de Thottan e Ji (2003) necessita que sejam definidas regras de atuação do algoritmo. Uma regra pelo grupo escolhido de variáveis, que representam o comportamento do tráfego no sistema. No sistema são analisados os grupos de variáveis obtidas no monitoramento do roteador da rede, através das informações armazenadas no grupo (*ip*) da MIB ou em um *bridge*.

No trabalho, a coleta de informações é feita no grupo *ip* da MIB SNMP. Deste grupo são utilizadas três variáveis: *ipIR, ipIDE e ipOR*. Na variável *ipIR* é armazenado o tráfego geral do roteador; a variável *ipIDE* representa o tráfego de entrado no roteador; e a variável *ipOR* descreve o tráfego de saída da interface de rede. Quando é realizada uma comunicação que utiliza a tabela de roteamento, o padrão de comunicação da rede pode ser descrito por estas variáveis permitindo a identificação de anomalias (THOTTAN e JI, 2003).

4.4.1.2 Detecção de Mudança Abrupta

O método proposto por Thottan e Ji (2003), verifica a qualidade do funcionamento da rede através do emprego de um indicador de anormalidade para cada variável da MIB coletada. O parâmetro de anormalidade obtido define a "saúde" da rede, e é obtido por meio de dados estatísticos equacionados para gerar o valor do indicador.

As anormalidades são percebidas através do uso de teste estatístico para a detecção de mudanças abruptas. O algoritmo usado para detectar as mudanças abruptas é baseado por um teste de hipóteses que considera a taxa de verossimilhança generalizada (BASSEVILLE e NIKIFOROV, 1993), ou teste GLR. O resultado de um teste GLR é apresentado como indicador de anormalidade graduado entre 0 e 1, onde "zero" é a ausência de anomalias e "um" é um estado anômalo da rede.

O processo de detecção que verifica a ocorrência de uma mudança abrupta é realizado através da comparação dos resíduos obtidos de duas janelas adjacentes de dados que são admitidas como estacionárias através de pequenos segmentos estacionários, como (THOTTAN e JI, 2003), sendo uma com tamanho N_r, para a janela de aprendizado, e outra com tamanho N_s para a janela de testes. As fórmulas (7) e (8) representam, respectivamente, as janelas de dados conhecidas como de aprendizado *R(t)* e testes (*S(t)*.

$$R(t) = \{r_1(t), r_2(t), ...rN_r(t)\}. \quad (7)$$

$$S(t) = \{s_1(t), s_2(t), ...sN_s(t)\}. \quad (8)$$

A janela de aprendizado representa o estado normal de funcionamento da rede, para cada período do dia. Para a construção de *R(t)* utilizam-se amostras que não contenham anomalias. A janela de testes *S(t)* corresponde ao tráfego atual da rede no momento de análise.

Para a identificação de anomalias é avaliado os resíduos das janelas obtidos aplicando o modelo AR (Auto-Regressivo) de ordem 1, isto é, utilizam o valor de uma regressão para estabelecer qual será o próximo valor da série. Ao aplicar a equação que retorna os resíduos do modelo AR é formado um conjunto de parâmetros.

A taxa da verossimilhança é obtida considerando a variância dos resíduos da janela de aprendizado (σ_L) e a variância dos resíduos da janela de testes (σ_s) e a variância associação das janelas de aprendizado e teste, dado por σ_P .

O algoritmo baseia-se na avaliação de hipóteses. Uma hipótese é formada pela avaliação da alteração nos parâmetros estatísticos, variância dos resíduos (σ_L), (σ_s) e (σ_P), avaliação dos parâmetros que descrevem o modelo AR atribuído para a série de dados e a taxa de verossimilhança comparada com *threshold* do sistema.

Desta forma o teste permite que seja reduzida a probabilidade de cometer falhas na identificação das anomalias. Pois antes de avaliar as alterações que estão ocorrendo sobre o modelo gerado pela serie temporal e não por alterações percebidas na transposição do *threshold*, evitando assim incidência de alarmes falsos, prejudiciais ao sistema.

Quando identificado um instante de tempo anômalo o mesmo é colocado em um vetor de anormalidade ($\psi(t)$). Neste vetor é encontrada a medida das mudanças abruptas do comportamento da rede. Cada vetor é a representação de uma variável de comunicação analisada pelo algoritmo de mudança abrupta. O algoritmo aponta que ocorreu alteração, mas não realiza a geração do alarme que fica a critério do filtro de duração.

O filtro de duração responsável pela avaliação das ocorrências de anomalia nas variáveis descritoras de tráfego, componente do sistema que avalia a correlação de linhas de comunicação com mudanças abruptas, que será visto na próxima seção.

4.4.2 Correlação de Mudanças Abruptas

Identificado anomalias nas variáveis descritoras de tráfego é empregado um vetor, conhecido como vetor de anormalidade, dado por ($\varphi(t)$). Vetor que descreve as variações que ocorreram na rede. Cada vetor de anormalidade é acompanhado de um operador linear dado por A. Este operador é calculado baseado nas correlações encontradas entre as variáveis da MIB (THOTTAN e JI, 2003), usando a função quadrática (9) com um operador escalar, responsável por indicar a saúde da rede e

também é o indicador de Anormalidade da rede que estará entre 0 e 1. Onde 0 representa a saúde da rede e 1 o grau máximo de anormalidade.

$$\int(\vec{\psi}(t)) = \vec{\psi}(t) A \vec{\psi}(t). \quad (9)$$

O operador A é um sistema composto por uma matriz de M x M, onde M é o número de variáveis consideradas. A composição do sistema é formada autovetores ortogonais de base real R^M e apresentam simetria. É simétrico porque possui M vetores ortogonais com M elementos, que forma um subconjunto baseado nos vetores de anormalidades. Cada auto valor contido no sistema corresponde a um estado anomalo do sistema, sendo detectado pela equação (10).

$$t_a = \inf\{t : \int(\vec{\psi}(t)) \geq \lambda_N\}. \quad (10)$$

Como o processo estabelece dois limites, 0 e 1, é necessário verificar o limite superior que corresponde a totalidade de anomalia, dada pela função (5).

$$\int(\vec{\psi}(t)) \leq \lambda_M = 1. \quad (11)$$

Cada elemento do vetor é uma anomalia percebida pelas variáveis da MIB, obtidos pelo teste GLR. Para que o sistema matricial seja completado é necessário que seja incorporado ao sistema uma previsão do comportamento da variavel descritora de tráfego. Adicionando o componente de previsão torna-se necessária normalizar o dado vetor. O emprego da normalização, garante que a expectativa do operador ***A*** seja entre 0 e 1.

A matriz formada pelo operador ***A*** é dada por (M+1)x(M+1) este operador Matriz foi projetado com base em autovetor. Estados normais de funcionamento da rede não são associados com o estado anômalo da rede (THOTTAN e JI, 2003). A operação da matriz emprega bloco diagonal da matriz superior com um bloco inferior. Os elementos do bloco superior seguem duas regras $I = J$ (12) e $I \neq J$. Para $I \neq J$ (13) é a média do conjunto dos pontos de correlação que possuem cruzamento espacial no vetor de anomalias estimadas no tempo.

$$A_{upper}(i,j) = \frac{1}{T}\sum_{t=1}^{T}\psi_i(t)\psi_j(t). \quad (12)$$

Regra I = J.

$$A_{upper}(i,j) = 1 - \sum_{i \neq j} A(i,j). \quad (13)$$

Regra I ≠ J

4.5 Conclusões Parciais

Os trabalhos estudados e descritos neste capítulo demonstram a existência de diferentes algoritmos para detectar anomalias em redes de computadores, dentre os quais pode-se citar algoritmos baseados em redes bayesianas, redes neurais, métodos estatísticos e detecção de mudanças abruptas (baseados no teste CUSUM). Para cada um foram constatadas peculiaridades que demonstram características de cada algoritmo, bem como pontos que descrevem suas vantagens e deficiências.

Os sistemas baseados em redes bayesianas requerem uma construção criteriosa que envolve um conhecimento sobre o problema e também uma constante atualização. O método baseado em redes neurais requer um treinamento, e consequentemente dados para isto, para que seja possível identificar os padrões. O emprego de métodos estatísticos avalia o comportamento do tráfego criando modelos que permitem descrever e prever a variabilidade do tráfego. Já o trabalho de Thottan e Ji (2003) explora séries temporais e detecção abrupta, porém por usar um modelo AR(p) não é capaz de trabalhar com séries não estacionárias.

5 UM MODELO DE IDENTIFICAÇÃO ATAQUES USANDO SÉRIES TEMPORAIS ARIMA

Este capítulo propõe uma nova abordagem de detecção de intrusão baseada em anomalias, a identificação de anomalias nos descritores de tráfego com base na análise de séries temporais do modelo auto-regressivo integrado de médias móveis (ARIMA) combinado com análise de correlação entre as anomalias em diferentes descritores. A apresentação do algoritmo proposto nesta seção faz parte da metodologia para detecção de intrusões. A seção 5.1 apresenta a estrutura do mecanismo de detecção, bem como o Algoritmo Geral para detecção de anomalias baseado em *ARIMA*, e a seção 5.2 detalha cada passo da identificação de anomalias e indicação de pré-alarme de ataque. A seção 5.3 trata da *Combinação de alarmes* para detectar se a ocorrência de anomalias dá-se em mais de uma variável analisada, a seção 5.4 compara o modelo proposto com o modelo de detecção proposto em (THOTTAN e JI, 2003) e a seção 5.5 apresenta as conclusões parciais.

5.1 Algoritmo Geral para detecção de Ataques baseado em ARIMA

Considerando que o tráfego de rede apresenta comportamento desconhecido (não pode ser descrito por uma única distribuição de probabilidade) e que apresenta momentos de não estacionariedade (DIAS et al., 2003), esta seção apresenta um mecanismo de detecção de ataques (anomalias) baseado em séries temporais ARIMA. O detector, nomeado Detector de Intrusão baseado em Séries Temporais implementado no R (DIBSETiR), é similar ao modelo adotado por Thottan e Ji (2003) (descrito na seção 4.4), porém trabalha com modelos ARIMA ao invés de apenas adotar o AR() de primeira ordem. Como resultado, possibilita modelar mais adequadamente o tráfego de rede, potencializando a melhor efetividade da detecção de intrusão.

A arquitetura do mecanismo de detecção do DIBSETiR está ilustrada na Figura 5.1. A entrada de dados, composta por descritores do tráfego de rede, é submetida a um módulo de **pré-processamento** dos dados que, para cada variável de entrada, organiza as estruturas de dados do detector, determina a estrutura do modelo ARIMA (parâmetros *p*, *d*, *q*) e computa os coeficientes do modelo (processo de *fit* do modelo). Os dados pré-processados são então passados ao **analisador de mudanças abruptas**, que inicialmente filtra a série aplicando o modelo ARIMA ajustado no pré-processamento e depois aplica o algoritmo CUSUM, responsável pela avaliação de mudanças e geração de um primeiro pré-alarme com base na análise dos resíduos, da variância e de testes de hipóteses. Cada pré-alarme é conseqüência da observação de anormalidades em um descritor de tráfego da entrada, ou seja, o alarme é sinalizado por fluxo de tráfego analisado. Os pré-alarmes são armazenados em um vetor de anormalidade que é combinado para verificar a presença de um ataque e gerar um alarme de ataque.

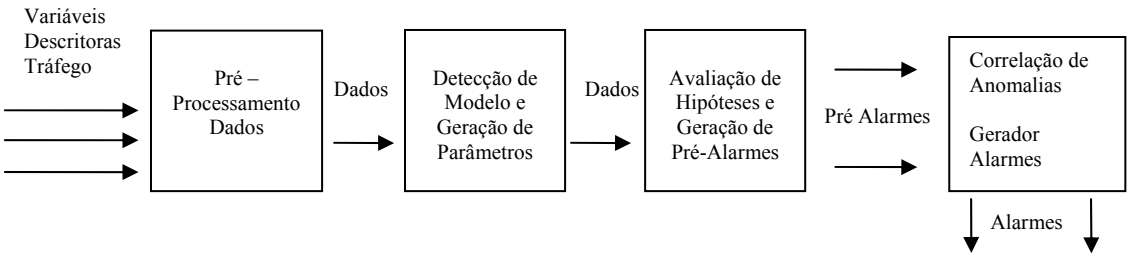

Figura 5.1 - Estrutura do Mecanismo de Detecção DIBSETiR.

A Figura 5.2 apresenta o algoritmo geral que implementa o funcionamento do mecanismo de detecção descrito na Figura 5.1. O primeiro passo do algoritmo corresponde à coleta de informações (variáveis descritoras de tráfego), que pode ser implementada de diversas maneiras, podendo ser ajustada de acordo com os descritores adotados. O segundo passo calcula os parâmetros e coeficientes das séries temporais de aprendizado e teste e identifica a ocorrência de anomalias realizando uma comparação estatística entre as séries. A comparação é realizada através de um teste de hipóteses baseado na taxa de verossimilhança (*likelihood ratio - LLH*). A comparação visa identificar o quão bem a janela de teste é representada pela janela de aprendizado,

considerando um dado limiar de variabilidade admitida (*threshold*). Como resultado pode se então identificar ou não um pré-alarme, que será inserido no vetor de anormalidade. Os dados inseridos no vetor de anormalidade são utilizados pela ***Correlação***, onde se confirma a ocorrência de um ataque nas variáveis descritoras de tráfego.

```
Inicio
Sempre que coletado um novo dado faça
{
        Processamento das séries "aprendizado" e "teste" da variável descritora;
        Testa-se o desvio de comportamento de aprendizado e testes para parâmetros e variância
        {
                Se desvio significativo, aplica teste de força com LLH e threshold para
                confirmação de uma anomalia encontrada, com conseqüente inserção no vetor de
                anormalidade
        }
}
Testa-se correlação entre as 3 variáveis descritoras de tráfego analisadas
{
        Testa-se o valor obtido na correlação para determinar o grau de ataque// entre 0 a 1
}
Fim
```

Figura 5.2 – Algoritmo Geral de Detecção de Anomalias.

5.2 Mecanismo de Identificação das Anomalias

5.2.1 Coleta de Informações

A coleta de dados/informações corresponde a quantificar a comunicação de rede pela contagem de pacotes transmitidos e recebidos por unidade de tempo. As séries temporais geradas a partir dos números de pacotes trafegados permitem expressar as características volumétricas do tráfego coletado, discretizado por protocolo ou fluxo, possibilitando a aplicação de técnicas de detecção de mudanças abruptas.

5.2.2 Pré-Processamento dos dados coletados.

Para cada variável de interesse tem-se uma série de aprendizado *R(t)* de tamanho N_r, expressa pela Equação (14), e uma série de teste *S(t)* de tamanho N_s, expressa pela Equação (15). A série de teste representa os dados que trafegam num determinado momento sob análise. Ambas as séries são modeladas como série temporal do tipo ARIMA(*p*,*d*,*q*). O modelo ARIMA é descrito pela Equação (16) (BOX; JENKINS; REINSEL, 1994), onde $\theta(B)$ é o coeficiente auto-regressivo, $\phi(B)$ é o coeficiente de média móvel, ∇^d é o coeficiente de diferenciação e a_t é o elemento de ruído branco.

$$R(t) = \{r_1(t), r_2(t), ...rN_r(t)\} \quad (14)$$

$$S(t) = \{s\ (t), s_2(t), ...sN_r(t)\} \quad (15)$$

$$z_t = \frac{\theta(B)}{\phi(B).\nabla^d}.a_t \quad (16)$$

Na fase de pré-processamento, os dados são organizados em séries temporais e as séries resultantes são modeladas. A modelagem consiste no cômputo dos parâmetros *p*, *d* e *q* do modelo ARIMA e seus respectivos coeficientes (*fit* do modelo). Este cômputo é realizado periodicamente por observação do erro quadrático médio da função de previsão. Cada vez que o erro ultrapassa 5% a estrutura do modelo é recalculada e é realizado novo *fit*.

O emprego do ARIMA permite obter modelos que apresentem uma melhor adaptação ao comportamento dos dados. Séries ARIMA podem ser mais fieis ao comportamento do tráfego e permitem tratar séries de dados não estacionárias. Como resultado, permitem reduzir os erros e melhorar a qualidade da detecção anomalias e ataques.

5.2.3 Detecção de Modelo e Geração de Parâmetros

Segundo a metodologia de Box, Jenkins e Reinsel, os coeficientes *p*, *d* e *q* do modelo ARIMA podem ser determinados através da avaliação das funções de autocorrelação e autocorrelação parcial da série. Bowerman e O'Connel (1993) apontam que a determinação pode ser realizada automaticamente usando regras experimentais.

Este trabalho realiza a determinação automática no qual o modelo ARIMA é recalculado periodicamente para manter-se ajustado aos dados de teste e sempre que a janela de aprendizado for alterada. Sempre que é realizado o ajuste do modelo, os parâmetros são recalculados. O mesmo é feito se o erro quadrático médio de previsão ultrapassa o limiar de 5%.

5.2.4 Avaliação de Hipóteses e Geração de Pré-Alarmes

Neste trabalho um desvio da normalidade do tráfego é percebido pelo método de detecção de mudanças abruptas (*abrupt detection*) proposto por Basseville e Nikiforov (1993). O algoritmo usado para detectar as mudanças abruptas realiza testes de hipóteses, o qual avalia os resultados obtidos pela modelagem da série temporal que é utilizada para obter características que descrevem o tráfego. O algoritmo responsável pela realização desta tarefa é o LLH generalizado, ou GLR (*Generalized Likelihood Ratio*). Nesta seção será visto como são formadas as hipóteses usadas no algoritmo para detectar mudanças abruptas e gerar pré-alarmes.

5.2.4.1 Teste de Hipótese

No teste de hipóteses é avaliada a possibilidade do comportamento do fluxo dos dados nas variáveis descritoras de tráfego analisadas. Para isso são avaliados dados como variância, parâmetros da série temporal e o valor de LLH que descrevem o comportamento do fluxo. Através deste teste são encontradas anomalias no comportamento da variável analisada, indicando assim um possível ataque (pré-alarme).

Os parâmetros que são avaliados no testes de hipóteses são obtidos por funções estatísticas. É obtida a variância dos resíduos das janelas analisadas (aprendizado e teste) e também de uma janela que representa a soma das duas janelas, conhecida como *polled*. Ficando disponíveis para análise os valores de: variância de aprendizado $\sigma(R)$, variância de testes $\sigma(S)$ e a variância da janela *polled* $\sigma(P)$.

O teste de hipótese avalia um traço de comunicação, que em um dado instante de tempo pode ou não apresentar uma alteração no perfil da comunicação. Para que seja determinada esta alteração, avaliam-se os parâmetros. Estes parâmetros são organizados em duas possibilidades de hipóteses diferentes, descritas por H0 e H1, apresentado nas equações (17), (18), (19) e (20).

O teste de H0 avalia se o comportamento do traço de comunicação sob teste (S) corresponde ao observado no traço de aprendizado (R), avaliando se o modelo ARIMA e a variância dos resíduos permanecem inalterados. De maneira similar, o teste de H1 corresponde a verificação de variabilidade estatística entre R e S. A Figura 5.3 apresenta o algoritmo que implementa o teste de hipótese.

H0

$$ARIMA(R) = ARIMA(S); \quad (17)$$

$$\sigma(R) = \sigma(S). \quad (18)$$

H1

$$ARIMA(R) \neq ARIMA(S); \quad (19)$$

$$\sigma(R) \neq \sigma(S). \quad (20)$$

```
TestaParametros(ParamentrosAprendizado, ParametrosTeste)
Inicio
        Se (ARIMA(R)) = (ARIMA(S)) e (^σ(R)) = (^σ(S))
                Hipótese=H0;
        Senão Hipótese =H1
Fim
```

Figura 5.3 – Algoritmo de Teste de Hipóteses

No caso de H1, um segundo teste é aplicado, o Teste de Força. Neste teste é calculado o *Log LikeHood Ratio* dado pela equação (21):

$$-\ln\lambda = N_R(\ln\sigma_P - \ln\sigma_R) + N_S(\ln\sigma_P - \ln\sigma_s) \quad (21)$$

Então, o valor obtido na equação (-ln λ) é comparado com valor de um *threshold* estático *h* que atribui 7% de tolerância sobre a expectativa de tráfego para o canal de comunicação, este valor foi empregado por cobrir a necesidades. O teste de força é demonstrado nas equações (22) e (23) e na figura 5.4.

H0

$$-\ln \lambda \leq h; \quad (22)$$

H1

$$-\ln \lambda > h. \quad (23)$$

```
TesteForça
Inicio
        Calcula(LLH);
        Se (LLH ≤ Threshold)
                Hipótese=H0;
        Senão  Hipótese=H1
Fim
```

Figura 5.4 – Algoritmo de t0este de Força das Hipóteses

5.2.4.2 Pré-Alarme

O resultado do teste de força (-ln λ) provê um indicador de anormalidade que é escalado entre 0 e 1, onde 0 indica que a rede apresenta funcionamento normal e 1 indica que a rede apresenta-se no grau máximo de anomalia do tráfego. Deste modo, um vetor de anormalidade $\psi(t)$ é composto pelos indicadores de comportamento dos fluxos analisados em cada teste. Este vetor de teste corresponde ao vetor de pré-alarmes, por fluxo, o qual indica o instante de tempo em que foi verificada uma alteração abrupta no comportamento de um dado fluxo de dados. Ou seja, para cada canal de comunicação

analisado tem-se um vetor de anormalidade e este vetor é a medida utilizada para representar as mudanças abruptas.

5.2.5 Combinação de alarmes

Para diferenciar uma anomalia de um ataque, o método proposto realiza uma análise combinada dos vetores de anormalidade de diferentes descritores de tráfego. Em cada vetor de anormalidade $\Psi_{(t)}$ encontra-se anomalias nos serviços analisados por meio da coleta de pacotes. Cada elemento do vetor de anormalidade é um indicador de "saúde" para um canal em particular. Entretanto, a "saúde" da rede só pode ser obtida quando vetores de anormalidade de diferentes descritores foram analisados de maneira combinada. A análise combinada dos vetores visa então reduzir a incidência de falsos positivos, um dos principais problemas dos mecanismos de detecção de intrusão baseados em anomalias.

Os valores contidos nos vetores de anormalidade apresentam-se entre 0 e 1. Para manter a escala, na análise combinada cada vetor de anormalidade é acompanhado de um operador linear dado por **A**. Este operador é desenvolvido baseado nas correlações encontradas entre os fluxos de dados coletados das interfaces de rede, usando uma função quadrática (Equação 24) com um operador escalar.

$$\int(\vec{\psi}(t)) = \vec{\psi}(t) A \vec{\psi}(t) \quad . \tag{24}$$

O operador **A** é uma matriz de *M x M*, onde cada *M* é o número de variáveis avaliadas, ou número de protocolos. A composição do sistema é formada por autovetores ortogonais de base real R^M e que apresentam simetria.

É simétrico porque possui *M* vetores ortogonais com *M* elementos, que juntos formam um subconjunto baseado nos vetores de anormalidades encontrados nos protocolos. Cada autovalor contido no sistema corresponde a um possível estado anômalo do sistema. Como o processo estabelece dois limites, 0 e 1, é necessário verificar o limite superior, o qual corresponde a totalidade de anomalia. Considerando que λN e λM correspondem, respectivamente, aos limites inferior e superior de anormalidade, o primeiro instante onde ocorre anomalia é dado por (24) e o *upper*

bound da função é dado por (25). Toda vez que (26) for satisfeita, tem-se a indicação de uma anormalidade na rede (alarme).

$$t_a = \inf\{t : \int(\vec{\psi}(t)) \geq \lambda_N\}. \tag{25}$$

$$\int(\vec{\psi}(t)) \leq \lambda_M = 1. \tag{26}$$

Para completar a matriz e garantir que o operador ***A*** tenha seu valor entre 0 e 1 é incorporado o elemento de previsão do comportamento usual da rede. O elemento de previsao da rede de ser normalizado para garantir os limites.

A matriz formada pelo operador tem ordem (M)x(M) e este operador matriz foi projetado com base em autovetor, para que o comportamento usual da rede não esteja relacionado com o comportamento anômalo da rede. A operação da matriz emprega bloco diagonal da matriz superior com um bloco inferior. Os elementos do bloco superior seguem duas regras $I = J$ e $I \neq J$. Para $I \neq J$ (27) é a média do conjunto dos pontos de correlacão que possuem cruzamento espacial no vetor de anomalias estimadas no tempo, em (28) a regra $I=J$.

$$A_{upper}(i,j) = 1 - \sum_{i \neq j} A(i,j). \tag{27}$$

Regra I ≠ J

$$A_{upper}(i,j) = \frac{1}{T}\sum_{t=1}^{T} \psi_i(t)\psi_j(t). \tag{28}$$

Regra I = J.

Os resultados gerados garantem que o autovalor gerado pelo bloco superior seja no máximo igual a 1. O valor obtido é o indicador de contribuição na anomalia da rede. Apesar de que o estado de saúde não deve contribuir para o indicador de anormalidade. Por este motivo, para detectar ataques somente é considerado o bloco superior da matriz.

5.3 Comparação dos Modelos e Discussão

Os modelos anteriormente apresentados, proposto neste capítulo e em Thottan e Ji (2003), são empregados na identificação de anomalias no tráfego de rede. Apesar das semelhanças dadas pelo uso do mesmo algoritmo (CUSUM) na identificação de mudanças abruptas, é possível apontar algumas diferenças que podem ser relevantes na obtenção de qualidade na identificação de anomalias.

Em Thottan e Ji (2003) as séries temporais são pré-processadas considerando o modelo AR(1). Porém o modelo AR(1) considera que os fluxos de dados coletados apresentam um comportamento estatístico estacionário, isto é, que não apresentam crescimento exponencial, e considera que apenas um elemento da série explique o próximo. Este fato cria uma limitação, pois o tráfego de rede é complexo e o modelo pode não representar adequadamente o tráfego, principalmente se na janela observada o tráfego apresentar não estacionariedade.

Com o emprego do ARIMA a série temporal não necessita de outra ferramenta para estacionarizar a série, permitindo que seja reduzido o custo computacional de tratamento e melhorando a qualidade de entrega de dados para os algoritmos de detecção. Outro fator a ser considerado é o emprego do cômputo dinâmico do modelo ARIMA e seus parâmetros de acordo com as necessidades verificadas.

5.4 Conclusões Parciais

Neste capítulo foi proposto um modelo de detecção de intrusão baseado em séries temporais ARIMA. O modelo ARIMA é empregado para melhorar a qualidade de representação do modelo de conhecimento do tráfego de redes aplicada na filtragem dos dados coletados. A qualidade do modelo gerado para descrever os dados e filtrá-los, incide diretamente na qualidade de detecção e geração de estimativas de alarmes.

Com base no modelo é definido qual o instante que o sistema apresenta alterações abruptas no tráfego de rede e a confirmação da sua ocorrência por meio da aplicação da correlação de variáveis de tráfego de rede.

6 IMPLEMENTAÇÃO DO DIBSETiR

Neste capítulo é apresentado detalhes do software que implementa o sistema de detecção de intrusão baseado em anomalia de rede, o DIBSETiR (Detector de Intrusão baseado em Séries Temporais implementado no R (R, 2011), proposto no capítulo 5. Para isso o capítulo foi dividido em três seções: a seção 6.1 descreve a arquitetura do software desenvolvido; a seção 6.2 descreve o módulo de coleta de informações de rede; a seção 6.3 descreve a implementação do analisador de anomalias; a seção 6.4 apresenta o módulo de análise combinada de anomalias; finalmente a seção 6.5 apresenta as conclusões parciais do capítulo.

6.1 Arquitetura de Software do DIBSETiR

Um IDS é formado basicamente por três módulos: coletor de dados, analisador e visualizador de respostas. O protótipo do DIBSETiR implementa apenas os módulos de coleta e análise de dados, dado que a visualização foge ao escopo deste trabalho.

O módulo de coleta de dados foi implementado baseado na Libpcap (TCPDUMP, 2010), uma biblioteca amplamente conhecida por fornecer funções para a coleta de informações do tráfego de rede. No módulo, os dados coletados são separados por protocolo e armazenados em arquivo. O módulo de análise de dados implementa o algoritmo proposto no capítulo 5 e utiliza o pacote estatístico R (R, 2011), para realizar suas funções estatísticas. Ambos os módulos foram implementados na linguagem de programação Java[3].

A comunicação do detector de intrusão com o pacote R foi implementada usando a API Rserve[4]. O Rserve é uma interface que permite a comunicação de uma aplicação escrita em Java ou C com uma aplicação R, viabilizando requisições de

[3] http://www.java.com
[4] http://www.rforge.net/Rserve/

processamento estatístico do detector para o R e vice-versa. O Rserve também permite que requisições sejam enviadas através da rede, visto que a interface é desenvolvida por meio de *sockets*, possibilitando assim o recebimento de instruções de linha de comando do R de vários detectores espalhados em uma rede. Com o R, as informações são agrupadas em séries temporais e processadas para obter dados que representam o comportamento da rede e suas anomalias. Os pacotes R que são utilizados no DIBSETiR o **Tseries** e o **Stats**, estes pacotes são responsáveis pela organização dos dados em séries temporais, modelar os dados e obter informações estatísticas Tseries e Stats.

6.2 Módulo Coletor de Dados

O módulo coletor de dados foi projetado para atender a demanda de validação do DIBSETiR, ou seja, foi projetado para ler dados da base de dados coletada na UFSM e para ler dados da base de dados DARPA. Porém o emprego da biblioteca Libpcap (TCPDUMP, 2010) também permite que o módulo colete descritores de tráfego (pacotes de protocolos pré-selecionados) diretamente da rede.

Como a implementação do DIBSETiR tem como foco a análise de protocolos com vista a detecção de intrusão, a implementação do protótipo possibilita a escolha dos protocolos TCP, ICMP e UDP, os mais explorados nos ataques, bem como a seleção da fonte, se da base de dados DARPA 99, da base UFSM 2010, dados coletados do tráfego total de entrada e saída da rede da UFSM, no mês de agosto e setembro de 2010.

Como os testes foram realizados no modo offline, os dados passam a estar armazenados em arquivos, que são carregados das bases de dados citadas. Para o processo de coleta de informações de tráfego utilizamos uma classe nomeada de Sniffer, que é aplicada para a coleta de informações presente nos arquivos, coletados anteriormente modo offline. Nesta classe encontramos alguns métodos manipulam os arquivos e leiam os dados de tráfego armazenados.

Para a chamada que manipulam os dados utilizamos a main. Na classe principal encontramos métodos que contém classes que avaliam o tráfego de acordo com o protocolo escolhido, dados por protocolos como TCP, ICMP e UDP.

6.3 Módulo Analisador de Tráfego

O projeto do analisador de mudanças abruptas do DIBSETiR, ou módulo analisador de tráfego, implementa o método de detecção de anomalias baseado na análise de séries temporais. O analisador de tráfego implementa uma classe de identificação do modelo ARIMA chamada ***IdentifyModel***. Esta classe oferece métodos que permitem analisar uma dada janela de tráfego com objetivo de reconhecer quais são os parâmetros ARIMA que melhor modelam a janela (série temporal). Periodicamente o modelo é recalculado a fim de uma melhor representação das estimativas de comportamento da rede.

O teste de verossimilhança generalizado foi implementado numa classe **TestGLR**. Este teste avalia hipóteses formadas pela comparação do modelo ARIMA da janela de aprendizado e da janela de teste, bem como a variância dos resíduos para estas janelas. A variância dos resíduos foi implementada no R. A interface com o R pode ser visualizada na Figura 6.1. Os métodos que manipulam os dados recebidos do coletor de informações de rede são: **loadRAplications( )**, responsável pela conexão com o R através da rede por meio de comunicação através de sockets; em **setDataSerie( )**, responsável pela serialização dos dados de acordo com a distribuição de freqüências; **getTimeSerieR( )**, que modela a série temporal e obtém os parâmetros que descrevem o comportamento da série. Feito a geração da série a partir do coletor, pode-se obter os resíduos e suas estatísticas, que são calculados pelos métodos **getResiduosTimeSeries()** e **getStatisticResiduos( )**, respectivamente. Dentro de **getStatisticResiduos( )** é inicializado a coleta de dados de variância gerado pelos métodos **getVarianciaLearn( )**, **getVarianciaTest( )** e **getVarianciaSoma( )** para as respectivas janelas dentro do R. Os dados dos parâmetros e dados das variâncias serão comparados em testes de hipóteses para definir se o dado instante apresenta comportamento anômalo.

O teste de força implementa o cálculo de *Log Likehood Ratio* que gera um coeficiente de qualidade do tráfego de rede que é contrastado com o *Threshold* atribuído para este sistema. O valor atribuído para esta medida é de 7% do valor encontrado em aprendizado.

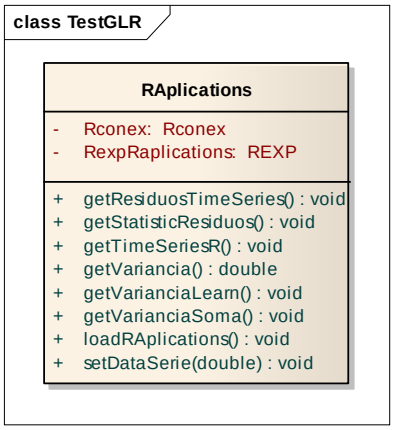

Figura 6.1 - Interface com as ferramentas estatísticas do R.

Na figura 6.2 tem-se classes que implementam o teste de hipóteses que avalia o teste de parâmetros na classe **TestParametros**, onde dentro do método **TestParametros( )** analisando os **ParametroLearn** e **ParametroTest** com fim de estabelecer se o parâmetro de teste apresentou crescimento em relação ao parâmetro de aprendizado, isto é a hipótese H1 do teste de hipóteses , caso manteve-se igual ou abaixo temos na hipótese H0.

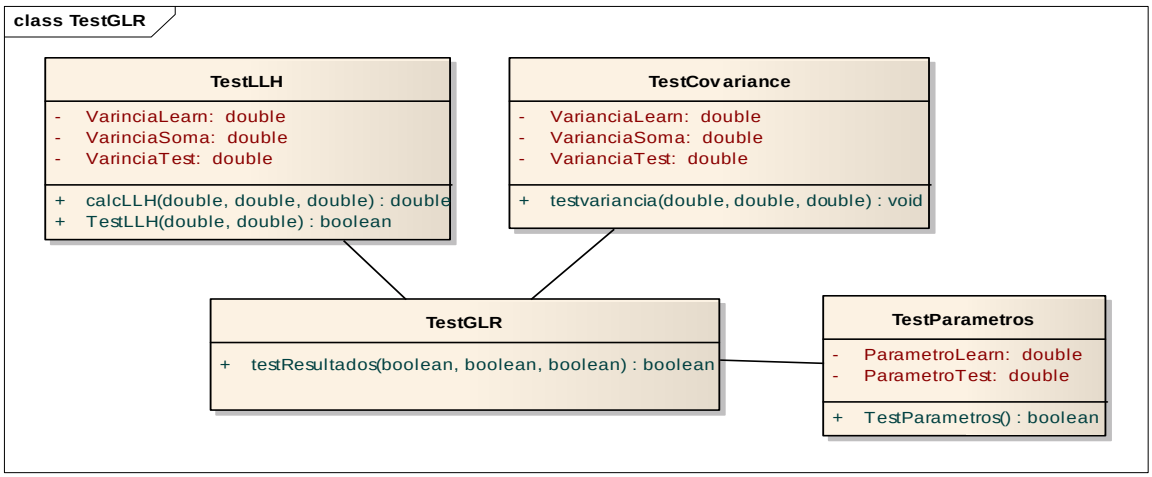

Figura 6.2 - Construção das comparações realizadas pelo teste GLR.

O teste de **TestVariance** avalia no método a hipótese referente a alterações ocorridas na **VarianciaLearn**, **VarianciaTest** e **VarianciaSoma**, testados em **testvariancia( ).** Neste teste comparamos no teste de hipóteses, H0 a equivalência das

variâncias encontradas na hipótese H1 a superação da variância de aprendizado(Learn) pela variância de Testes.

O teste de força é realizado em **TestLLH,** que aplica valores da variância das janelas **VarianciaLearn**, **VarianciaTest** e **VarianciaSoma** para a geração do LLH em **calcLLH( )** passando o valor gerado para o **TestLLH( )** com o intuito de confirmar a hipótese H1. Nesta classe também avalia-se o nível de anormalidade percebido para a janela sob teste do tráfego de pacotes.

6.4 Vetores de Anormalidade

Identificada uma anormalidade, esta é adicionada ao vetor de anormalidade (Ψ). Cada anomalia identificada é armazenada em um objeto ***Abnormality***. A Figura 6.3 apresenta as assinaturas do armazenamento de dados que contém informações sobre as anomalias (vetor de anormalidade). Para esta tarefa tem-se: **AbnLevel**, que indica o nível de anormalidade percebido; **AbnormalityIdentification**, que demonstra se fora identificada alguma anormalidade; **abnThreshold**, que armazena o valor aplicado ao *Threshold* com fim de justificar a declaração de anomalia bem como o valor de o valor de **AbnormalityLLH** é empregado no método de correlação, seção 6.5. Também adicionamos dados de variância das janelas de aprendizado, testes e soma, com **AbnvarLearn**, **AbnvarTest**, **AbnvarSoma** e tem-se ainda o instante em que foi detectada a anomalia, com **Abnormalityindice**. O índice de anormalidade é empregado para facilitar a identificação do instante de anormalidade e ajudar na construção do sistema de correlação.

Um objeto **Abnormality** é armazenado em uma classe **Vector** do **Java**, nomeada de **AbnomalityVector**, que apresenta uma atributo **AbnormalityVector** responsável por atribuir índice para os objetos armazenados.

Os vetores de anormalidade são agrupados para construir o sistema de correlação, respeitando o instante onde foi realizada a detecção, por exemplo: detectou-se anomalia no instante 1,2 e 3 para todas as variáveis analisadas, desta forma-se o sistema de correlação, caso tenhamos os instantes somente para 1, 2 e 3 para a variável TCP é impossível formar o sistema e obter o índice de correlação. Por este motivo emprega-se o filtro de duração, visto na seção 6.5. A combinação dos dados então

respeitará a existência de anormalidades em instante de tempo idêntico para os protocolos avaliados. A Figura 6.3 abaixo apresenta os vetores de anormalidade e o objeto armazenado pelo vetor.

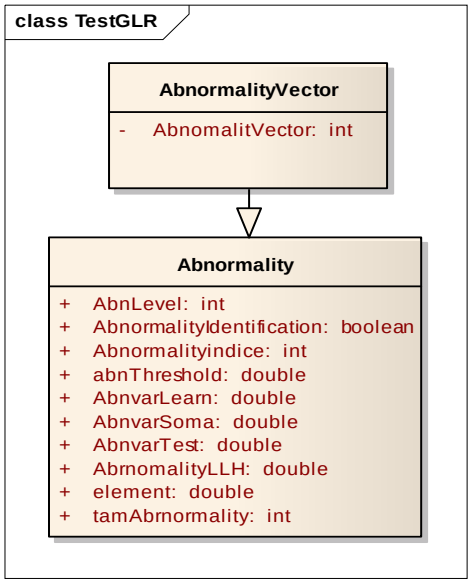

Figura 6.3 – Vetor de Anormalidade

6.5 Correlações dos Vetores

Os dados armazenados nos vetores de anomalia são processados pela aplicação de correlação. Cada vetor linha e colocada na ferramenta de que cria matrizes ***MxM,*** onde M é o número de variáveis independes utilizadas para formar o sistema, que no caso de nossa aplicação é de 3x3 devido aos serviços (TCP, ICMP e UDP). O processamento da correlação e o respectivo alarme gerado será dado após ser processado pelas seguintes aplicativos, conforme figura 6.4, onde temos: **CorrelationMatrix** que possui **VetorCorrelation**, que armazena os dados de anormalidade percebidos em **testGLR,** seção 6.3. Estes vetores de correlação são referentes às variáveis descritoras de tráfego empregadas para, nesta classe tem-se um método **NormalizeData( )** responsável por normalizar as previsões realizadas pelo ARIMA. Para este sistema não foi utilizado devido que a previsão relaciona-se a variáveis independes e uma previsão afetaria as outras variáveis de forma a levar a erro.

Também nesta classe é gerada a matriz de correlação através do método **GerateMatrix()**, contendo os dados de anormalidade referentes aos índices que apresentaram anomalias.

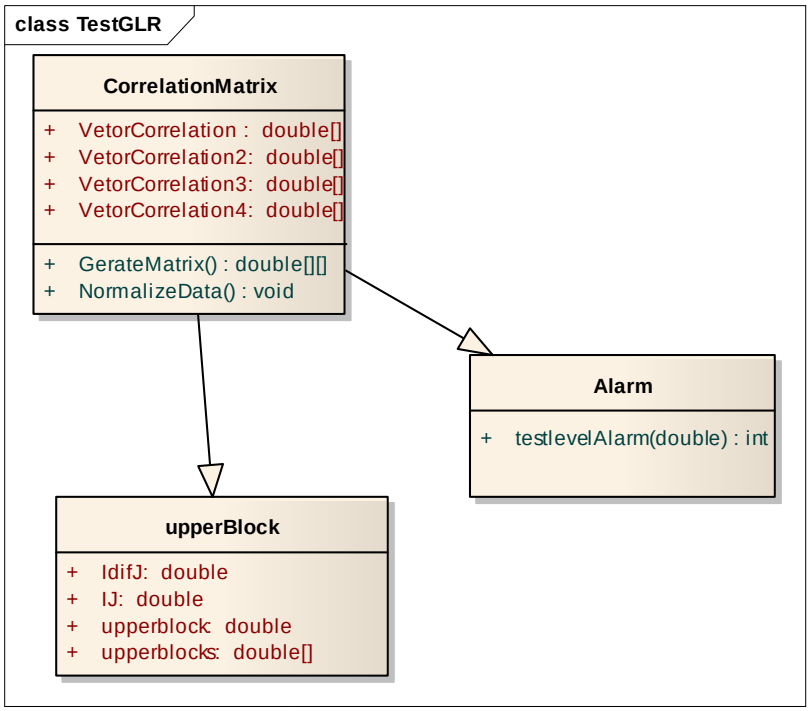

Figura 6.4 - Correlação de Vetores Anormalidade

O *upperBlocks* é a classe responsável pelo cálculo de correlação das linhas. As linhas são correlacionadas, que é aplicado um cálculo, através do método **getCalcUpperBlock( )**, dentro deste cálculo são obtidos **IdifJ**, quando temos índices I#J e **IJ** quando I=J, gerados respectivamente pelos métodos **getIdifJ( )** e **getIJ( )**. O resultado deste processo é um valor escalar que é responsável por ser operador na função quadrática empregada para operar os vetores de anormalidade.

Na figura 6.5 tem-se o desenho das classes e os relacionamentos das classes para a resolução da tarefa de detecção abrupta, englobando as classes comentadas na figuras anteriores e outras classes de entrada de dados como as que são armazenadas em **Abrupt.**

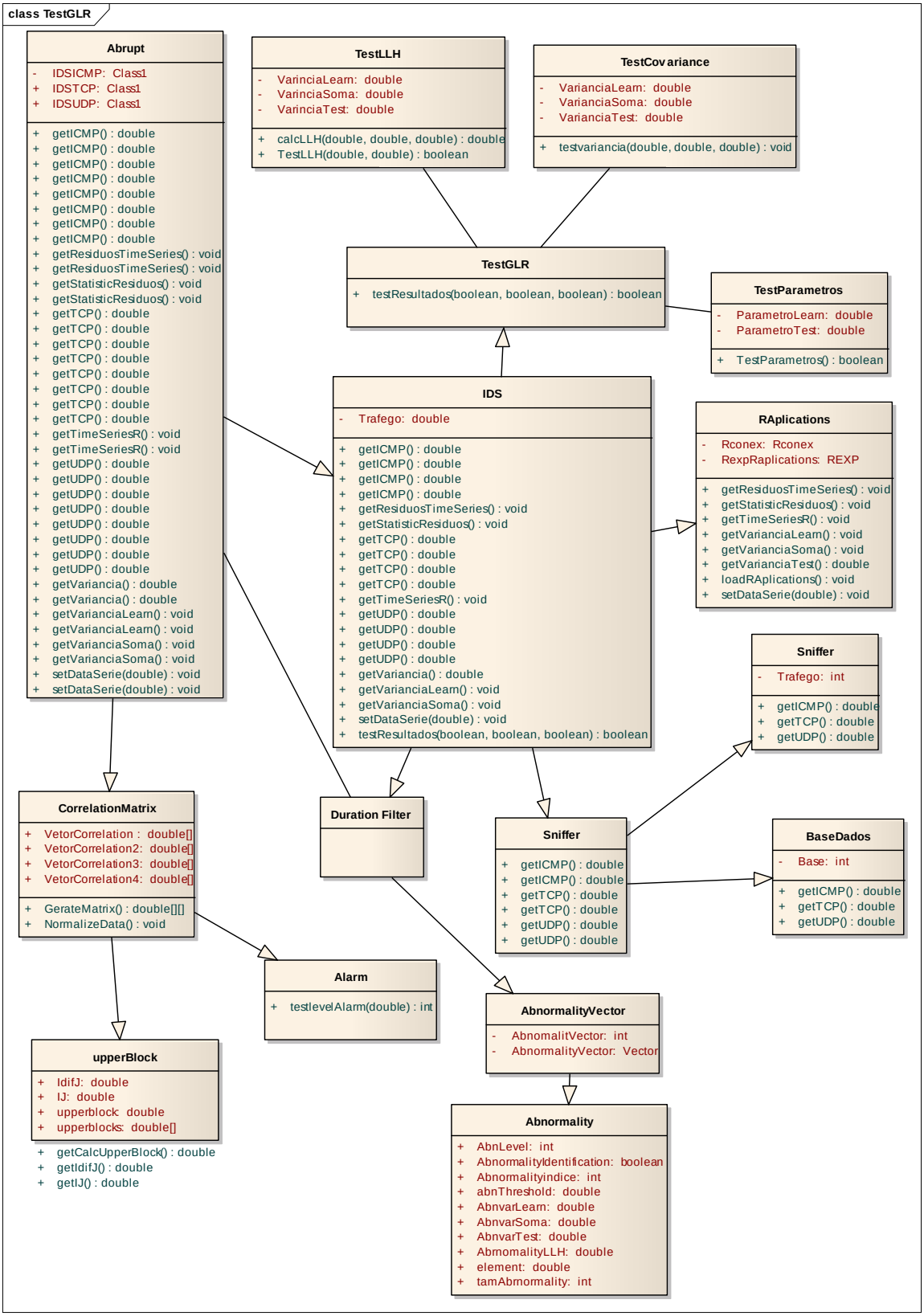

Figura 6.5 - Sistema de identificação de Intrusão

6.6 Algoritmo para implementação

Na implementação presente no algoritmo da Figura 6.6 tem-se a construção das estruturas modeladas neste capítulo onde os métodos realizam a tarefa de processamento e análise dos dados de tráfego de redes.

Mostra-se dados sendo coletados com para cada uma dos protocolos de rede avaliados (TCP, ICMP e UDP) que armazenados no seu respectivo vetor. Estes vetores são enviados para o R que realiza o processo de avaliação da série temporal para cada protocolo avaliado. Quando processada a série tem-se os parâmetros do modelo para série de aprendizado e testes e variância para as mesmas séries.

O parâmetro do modelo das séries juntamente com a variância para estas séries de dado são avaliados no teste de hipóteses. O teste de hipóteses verifica se ocorre a igualdade ou superação dos parâmetros da série de aprendizado. Verificada a superação do parâmetro do modelo e da variância de aprendizado teremos o indicativo de anomalia e para confirmar a anormalidade é calculado o LLH que é contrastado com *threshold* no teste de força confirmado a anomalia, elemento armazenado no vetor de anormalidade.

O último passo é correlacionar as anormalidades dos protocolos analisados. A correlação é realizada com o emprego de uma matriz de 3X3, visto que temos 3 protocolos. Este sistema criado determina se ocorre um ataque pela avaliação de variáveis de tráfego de rede.

6.7 Conclusões Parciais

Neste capítulo foram descritos os detalhes de implementação do algoritmo proposto no capítulo 5, o que corresponde à transposição dos modelos matemáticos para o projeto conceitual do sistema computacional. No modelo conceitual encontra-se detalhes de como foi realizada a construção do software responsável por validar o algoritmo proposto.

```
Int Janela; //valores entre 32 e 128 elementos
Double ICMP[tamanhoEntrada]; Double TCP[tamanhoEntrada]; Double UDP [tamanhoEntrada];
Double ThreICMP[tamanhoEntrada] ; Double ThreTCP[tamanhoEntrada]; Double ThreUDP [tamanhoEntrada] ;
ICMP=ColetaTrafego( ); TCP =ColetaTrafego( ); UDP=ColetaTrafego( ); Double threshold=0,07;
Double vetorAnormalidadeTCP[Janela];
Double vetorAnormalidadeUCP[Janela];
Double vetorAnormalidadeICMP[Janela];
DoubleVetorAnormalidade[Janela];
ProcessaSerieTemporal{
        Double paraModeloAprendizado;
        Double paraModeloTeste;
        Double varianciaaprendizado;
        Double varianciaTestes;
        paraModeloAprendizado=GerarParametrosModeloAprendizado(ICMP ou TCP ou UDP);
        paraModeloTeste= GerarParametrosModeloTeste(ICMP ou TCP ou UDP);
        varianciaaprendizado =ObterVarianciaAprendizado(ICMP ou TCP ou UDP);
        varianciaTestes =ObterVarianciaTeste(ICMP ou TCP ou UDP);
        geraThreshold(threshold, ICMP ou TCP ou UDP);
        TestarGLR(paraModeloAprendizado, paraModeloTeste, varianciaaprendizado, varianciaTestes)    {
         if(paraModeloAprendizado=<paraModeloTeste)&&(varianciaaprendizado=<varianciaTestes)
          {//Hipotese H1;
                testeForça(LLH,Threshold){
                        vetorAnormalidade=LLH;
                }
        vetorAnormalidadeTCP= vetorAnormalidade;
        }
        elseif(paraModeloAprendizado>=paraModeloTeste)&&(varianciaaprendizado>=varianciaTestes)
          {//Hipotese H0;
                testeForça(LLH,Threshold){
                        vetorAnormalidade=LLH;
                }
        }
        Double indiceAtaque;
        calculoCorrelacao(     vetorAnormalidadeTCP[i],vetorAnormalidadeTCP[i+1],
        vetorAnormalidadeTCP[i+2],     vetorAnormalidadeUCP[i],vetorAnormalidadeUCP[i+1],
        vetorAnormalidadeUCP[i+2],          vetorAnormalidadeICMP[i],          vetorAnormalidadeICMP[i+1],
vetorAnormalidadeICMP[i+2]){
        Double [3][3]= vetorAnormalidadeTCP[i],vetorAnormalidadeTCP[i+1],vetorAnormalidadeTCP[i+2],
                        vetorAnormalidadeUCP[i],vetorAnormalidadeUCP[i+1],vetorAnormalidadeUCP[i+2],
                vetorAnormalidadeICMP[i], vetorAnormalidadeICMP[i+1], vetorAnormalidadeICMP[i+2];
        double somatorio;
        for(int i=0,j=0; i<2;i++){
        if(i==j){
         Somatorio=Somatorio+dado[i][j];
        }elseif (i!==j ){
        Somatorio=(dado[i][j]* Somatorio);
        Somatorio =(1/Tempo)*somatorio;
        }
        }
```

Figura 6.6 – Algoritmo implementação reduzida

A fim de tornar o algoritmo reutilizável de maneira livre, na implementação foram empregados softwares externos como o pacote estatístico R, um pacote livre que implementa as bibliotecas estatísticas utilizadas. O uso desta bibliotecas viabilizou a construção do sistema que identifica alterações abruptas no comportamento do tráfego de pacotes e avalia se este comportamento verificado é originado por ataques identificáveis via análise dos protocolos de rede considerados.

7. AVALIAÇÃO DOS RESULTADOS

Neste capítulo é descrita a metodologia de avaliação do comportamento do algoritmo proposto avaliação do tráfego de rede. A metodologia aplicada divide a resolução do problema em duas fases distintas. A primeira fase é dada pela identificação de anomalias nas variáveis descritoras de tráfego e a segunda é formada pela correlação das variáveis descritora de tráfego. Com objetivo de verificar se ocorre "contaminação", um pré-alarme gerado num dado protocolo dispara um processo de análise de correlação nos outros dois protocolos. Caso identificado a correlação, evidencia-se um ataque.

A disposição dos resultados obtidos tem a seguinte organização: na seção 7.1 são apresentados os resultados obtidos na fase de detecção de anomalias de tráfego; na seção 7.2 são avaliados os resultados obtidos na fase de correlação das linhas de comunicação; e finalmente na seção 7.3 são apresentadas as conclusões parciais das análises.

7.1 Detecção de Anormalidades

A primeira fase do algoritmo é dada pela percepção da ocorrência de anomalias de tráfego de rede. Cada anomalia é percebida pelo algoritmo descrito no capítulo 5 e que gera indicadores de anormalidade. Estes indicadores são armazenados em vetores de anormalidade para serem processados pela segunda fase do algoritmo. Para que fosse comprovada a eficiência do algoritmo foram realizados testes descritos a seguir.

Os experimentos realizados avaliaram duas fontes de dados diferentes: a base de dados DARPA (MIT LINCOLN LABORATORY, 1999); e uma base de dados construída através da observação do fluxo de entrada da UFSM. Para cada uma das bases foram realizados testes com os seguintes cenários para avaliação: experimentos que avaliaram o comportamento nos protocolos TCP, ICMP e UDP, sendo que para cada experimento foi observada uma janela de dados com tamanhos de 32, 64, 96 e 128 amostras, cada uma correspondendo a quantidade de pacotes trafegados num intervalo de 1 segundo.

Para fins de comparação o modelo proposto baseado em ARIMA(p,d,q) com aquisição de parâmetros de forma dinâmica, sendo comparado com o modelo original AR(1) proposto por Thottan e Ji (2003). Nas comparações foram utilizados traços das bases de dados com tamanhos definidos em 32.000 amostras.

7.1.1 Resultados com a Base de Dados DARPA 99

Foi empregada a primeira semana da base de dados DARPA 99 como treinamento, pois esta semana da base não contém ataques. A segunda semana foi utilizada como semana de teste, por apresentar ataques marcados e identificados, o que permite o conhecimento exato das características de cada um dos ataques utilizados na base. O emprego da base DARPA 99 é reconhecido como referencial de dados em outras pesquisas com objetivo de detecção de ataques.

A Figura 7.1 apresenta o tráfego de pacotes por protocolo, sendo que o primeiro gráfico representa o tráfego de pacotes do protocolo TCP. Neste protocolo é encontrado um grande número de anomalias e algumas anomalias podem ser indícios de ataques. O segundo gráfico descreve o comportamento do tráfego de pacotes do protocolo UDP e o terceiro gráfico é o fluxo de pacotes do protocolo ICMP.

A Tabela 7.1 apresenta o número de anomalias detectadas em um traço contendo 32000 elementos da base de dados DARPA. Os resultados consideram janelas de 32, 64, 96 e 128 elementos e o emprego dos algoritmos baseados no AR(1) e no ARIMA.

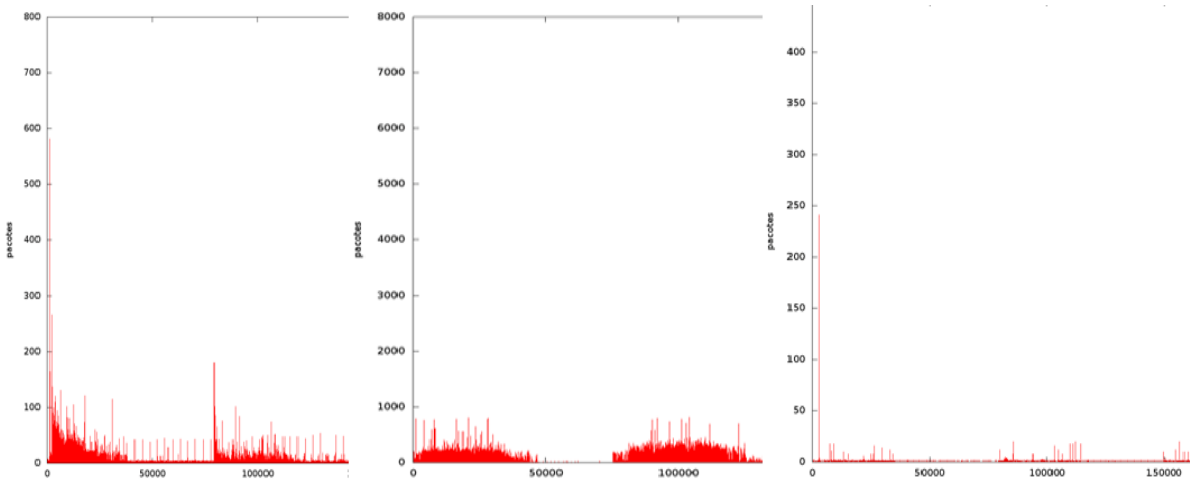

Figura 7.1 - Gráficos do tráfego Protocolos TCP, UDP e ICMP respectivamente.

A Tabela 7.1 mostra que foi verificado um grande número de detecções de anomalias utilizando qualquer tamanho de janela. Observa-se que uma anomalia corresponde a uma fuga do padrão de tráfego usual, podendo ser gerado pelo uso de aplicações legítimas. No caso da base de dados DARPA 99, existem muitos períodos de coleta que apresentam tráfego de pacotes igual a zero para o treinamento da rede, dado que a primeira semana não apresenta ataques.

Tabela 7.1 – Tabela contendo o número de anomalias detectadas por traço com 32000 amostras da base de dados DARPA.

Protocolo	**AR 128**	**ARIMA 128**	**AR 96**	**ARIMA 96**	**AR 64**	**ARIMA 64**	**AR 32**	**ARIMA 32**
TCP	140	138	687	637	2816	2297	10039	10039
UDP	6146	6146	13108	13108	9981	9981	4376	4376
ICMP	6146	6146	13108	13108	9981	9981	4376	4376

Com este fato o cálculo do *threshold* gerado para a série de dados fica prejudicado, pois a variabilidade do tráfego de pacotes muitas vezes ultrapassa o *threshold* utilizado como margem de segurança. A Figura 7.1 também mostra que as detecções realizadas utilizando o AR(1) (THOTTAN e JI, 2003) apresentaram valores idênticos para a detecção de anomalias usando o ARIMA(p,d,q), com qualquer tamanho de janela para os protocolos UDP e ICMP. Já para o TCP fora verificado igualdade de detecções apenas no caso da janela com 32 amostras. Já para janelas maiores, o número de detecções diminuiu no protocolo TCP, podendo indicar melhor resultado em termos de falsos positivos.

Para janelas de 64 elementos o número de detecções realizadas no protocolo TCP com o emprego de AR(1) indicou 2816 anomalias no tráfego e com ARIMA(p,d,q) foram indicadas 2297, isto significa uma redução de 18,43% de anomalias de tráfego.

Para a janela de 96 elementos nos protocolos TCP foi de 687 anomalias detectadas para o modelo AR e para o protocolo ARIMA(p,d,q) 637 anomalias, o que representa uma redução de 8,2% no número de anomalias detectadas. Finalizando em janelas de 128 elementos com o modelo AR(1) foi encontrada 140 anomalias e com o modelo ARIMA(p,d,q)138 anomalias, o que apresenta uma redução de 2% no número de anomalias verificadas na base DARPA.

É importante salientar que a redução de pré-alarmes falsos é uma característica importante para sistemas de detecção de intrusão.

7.1.2 Resultados com a Base de Dados UFSM 2010

Para a avaliação da base de dados UFSM 2010 foram empregados os mesmos moldes de processamento da base DARPA 99, onde foram avaliados os protocolos TCP, UDP e ICMP com janelas de 32, 64, 96 e 128 para verificar a ocorrência de anomalias que podem significar ataques e o número de amostras do traço foi de 3200. A Tabela 7.2 apresenta os resultados encontrados.

Tabela 7.2 – Tabela contendo os resultados com 3200 amostras UFSM.

Protocolo	**AR 128**	**ARIMA 128**	**AR 96**	**ARIMA 96**	**AR 64**	**ARIMA 64**	**AR 32**	**ARIMA 32**
TCP	2	2	11	11	296	296	525	525
UDP	0	0	0	0	0	0	0	0
ICMP	0	0	0	0	0	0	0	0

Os resultados obtidos aplicando a base de dados UFSM apresentaram uma particularidade quanto ao protocolo onde foram detectadas anomalias. Foram detectadas variações abruptas somente no protocolo TCP e para os protocolos ICMP e UDP não foi encontrada nenhuma anomalia. Além disto, identifica-se que quanto menor o tamanho

da janela mais anomalias são detectadas. Para as janelas de 128, 96, 64 e 32 amostras, foram detectadas 2, 11, 296 e 525 anomalias, respectivamente.

Após analisar as anomalias detectadas, percebe-se que em dados de tráfego real o comportamento de explosão de tráfego (não estacionariedade) não foi percebido e por este motivo com a aplicação de um modelo auto-regressivo consegue-se obter resultados idênticos aos obtidos com o emprego de um modelo mais sofisticado, como o ARIMA.

7.1.3 Comentários sobre a Detecção de Anomalias

Aplicado o modelo de detecção foram obtidos os resultados acima dispostos. Nestes resultados foram encontradas anomalias nos protocolos analisados para as bases de dados DARPA 99 (MIT LINCOLN LABORATORY, 1999) e dados da UFSM 2010. Para a base de dados DARPA foram identificados um comportamento inversamente proporcional em relação aos dados da base UFSM. Este comportamento é em relação ao tamanho da janela e número de detecções. Para janelas de tamanho de 128 e 96 unidades, o número de anomalias nos protocolos ICMP e UDP foi maior, enquanto para janelas de 64 e 32 amostras o número maior de ocorrências foi no protocolo TCP. Este resultado não permite concluir sobre a influência do tamanho da janela. Por outro lado, pode-se observar que o número de pré-alarmes diminui com a aplicação do modelo ARIMA, sugerindo que pode ser interessante a substituição do AR(1) pelo ARIMA dinâmico se o número de falsos positivos é grande.

7.2 Correlação das Anomalias Encontradas

A correlação proposta no trabalho utiliza como base de suas análises, os vetores gerados pela detecção de anormalidades. Os vetores são montados individualmente para cada variável de interesse (descritor de tráfego considerado) e traz como informação instantes de tempo em que foram observadas anomalias.

No caso específico dos testes realizados, cada vetor de anomalia trás informações sobre a “saúde” de rede em cada um dos protocolos. Juntos, os vetores

formam uma matriz de tamanho M x M, onde M é o número de descritores de rede avaliados. Então, foi avaliado as anomalias em três protocolos distintos: TCP, ICMP e UDP. As informações indicam qual é a intensidade do comportamento anômalo da rede. A matriz formada deve responder com um coeficiente de saúde entre 0 e 1, indicando a intensidade do ataque sofrido na rede.

7.2.1 Correlações das Anomalias da base DARPA 99

A base de dados DARPA possui ataques descritos na sua documentação, apontando o tipo e a posição dos ataques. Por exemplo, o primeiro ataque documentado ocorre após 3205 amostras. Foram utilizadas 32000 amostras da segunda semana da base DARPA (semana com ataques) para verificar o comportamento dos três protocolos utilizados como variáveis descritoras de tráfego.

Primeiramente foram avaliadas janelas de 32 amostras com detector formado pelo AR(1) avaliando janela de 32 elementos para os protocolos ICMP, TCP e UDP. Para ilustrar a identificação de correlações entre os protocolos de rede a Figura 7.2 ilustra as variáveis num trecho do tráfego de pacotes onde foi detectado um maior número de anomalias (entre as amostras 29600 a 29800). Uma anomalia só é considerado como ataque se ocorrer nas três variáveis analisadas ao mesmo tempo.

Note que nesta fase não se avalia o valor de tráfego de pacotes e sim o coeficiente LLH que se apresenta entre 0 e 1 (vide Figura 7.2). Quando correlacionamos as variáveis de tráfego é necessário que os vetores de anormalidade possuam dados em todas as variáveis ou ainda quando percebido em uma variável o comportamento anormal, este comportamento anormal seja percebido instantes após a percepção da anormalidade em uma variável, ou seja, pode se propagar para as outras variáveis. No caso do tráfego a propagação de ataques pode não ocorrer devido a construção dos ataques que não atacam em múltiplos protocolos e portas. Mesmo com esta restrição algumas anomalias apresentaram comportamento que permitem identificar ataques por meio de correlações. Nos gráficos apresentados este instante de ataque foi dado nos instantes 29650 a 29660 e alguns picos após este instante de ataque detectado.

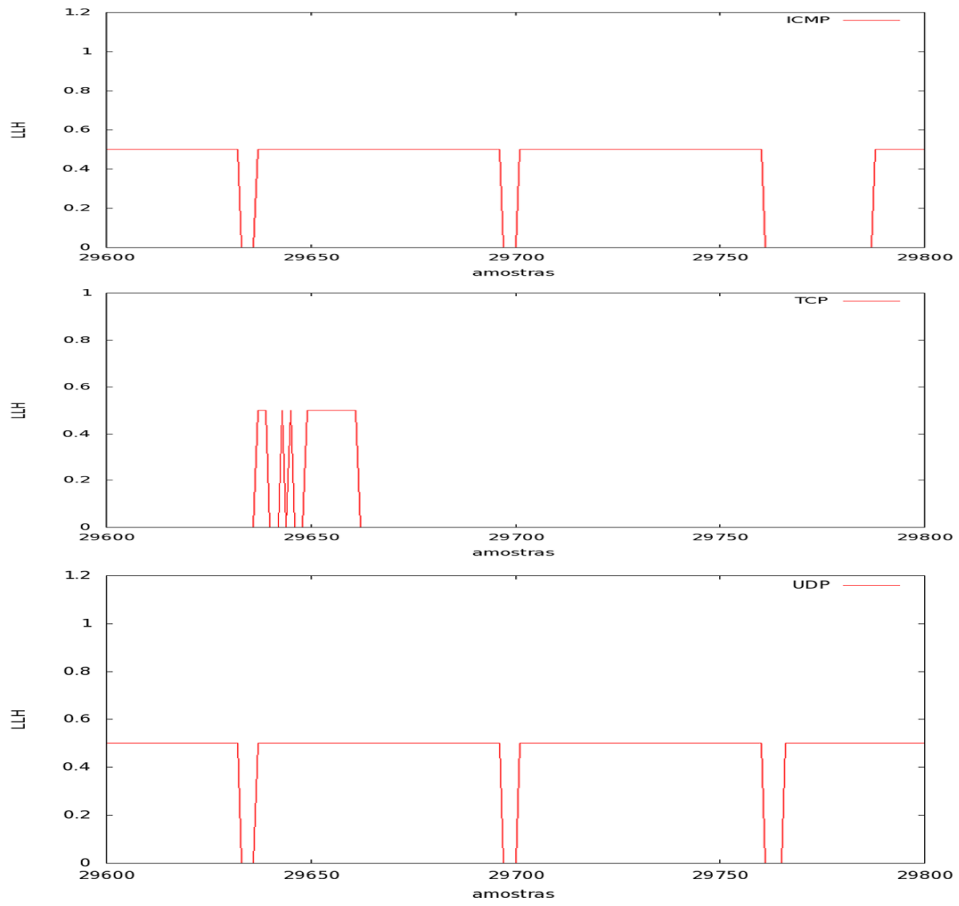

Figura 7.2 - Demonstração das correlações entre ICMP, TCP e UDP, obtidas em AR(1) com janela de 32 elementos.

As Figuras 7.3, 7.4 e 7.5 ilustram a identificação de correlação nos protocolos ICMP, TCP e UDP, sob as mesmas condições da Figura 7.2, mas considerando janelas de 64, 128 e 256 elementos, respectivamente.

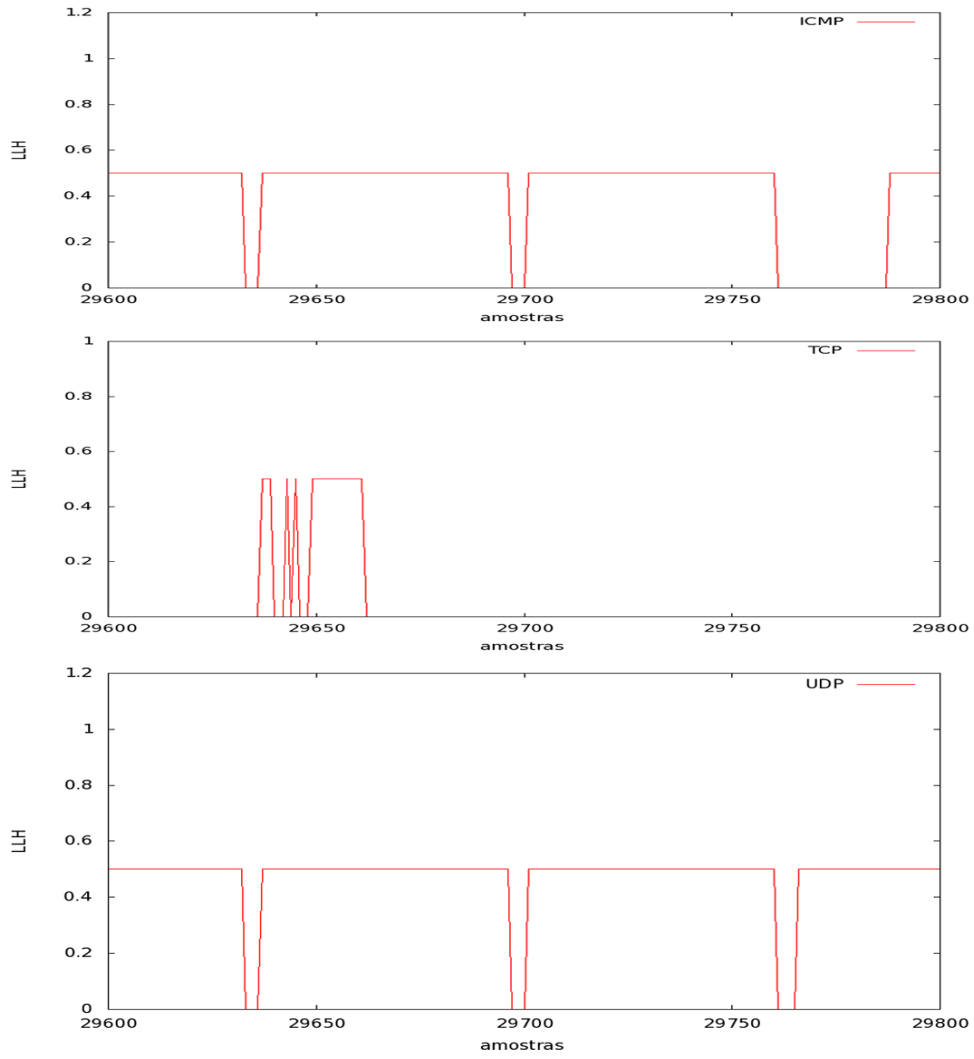

Figura 7.3 - Demonstração das correlações entre ICMP, TCP e UDP, obtidas em AR(1) com janela de 64 elementos.

A Figura 7.4 ilustra correlações entre ICMP, TCP e UDP e possui um comportamento semelhante de detecção ao encontrado na Figura 7.2: o emprego do ARIMA é viável quando obtemos rapidamente a estacionarização dos dados e neste momento consegue-se fazer uso de suas vantagens de modelagem que segue o comportamento dos dados de forma mais precisa. Visto que a implementação do algoritmo responsável pela avaliação de quais são os parâmetros que regem ao comportamento da série faz uso de uma restrição para a melhoria do desempenho do algoritmo quando ultrapassa 10 estacionarizações então é empregado o modelo AR(1).

Foram detectadas correlações entre 28225 e 28255 entre outros pontos com menor intensidade de correlação no tempo.

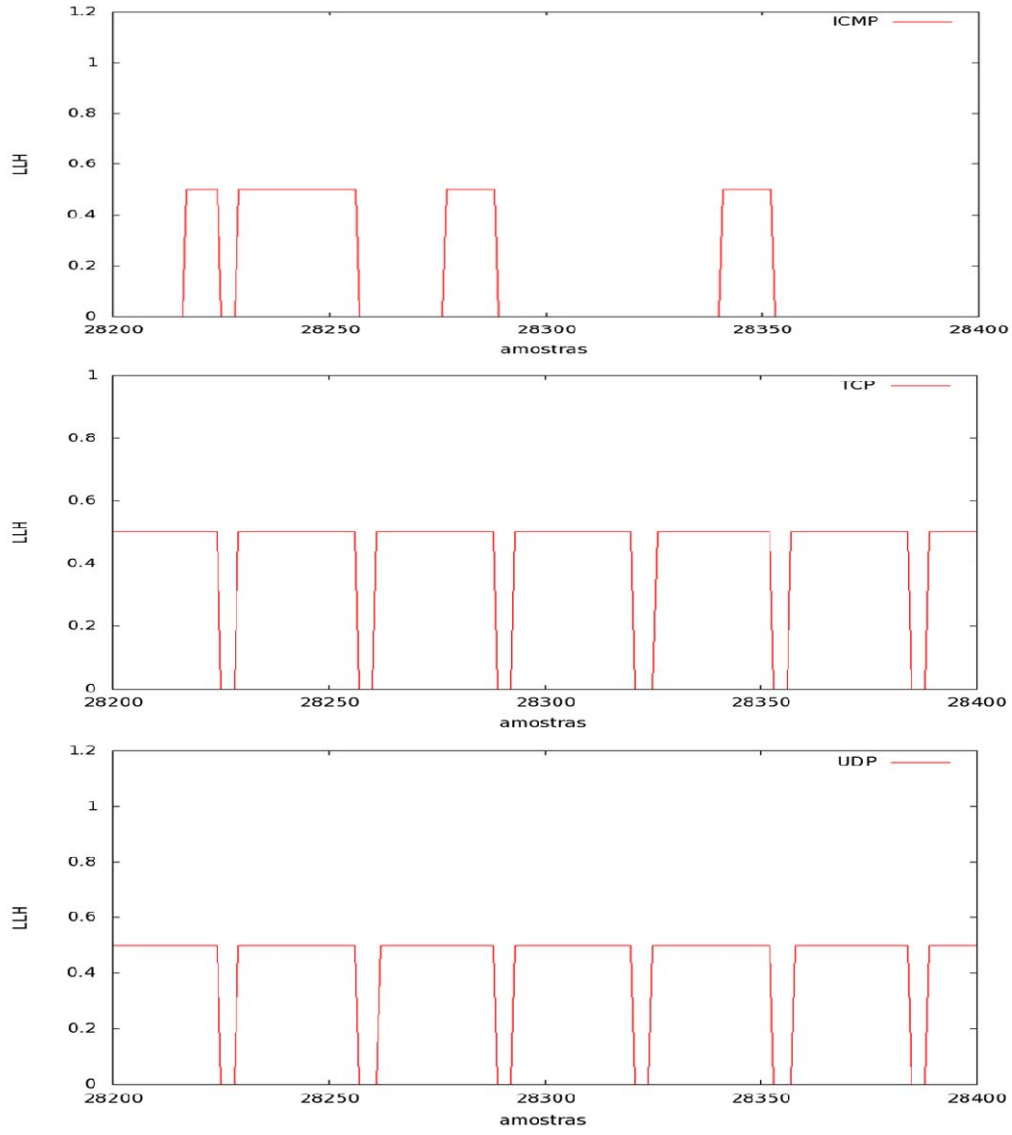

Figura 7.4 - Demonstração das correlações entre ICMP, TCP e UDP, obtidas em ARIMA com janela de 32 elementos.

Na Figura 7.5 são encontradas correlações nas amostras 29650 a 29660, e neste momento se pode gerar um alarme de ataque por terem sido encontradas correlações entre as anomalias encontradas. Porém a Figura 7.5 é semelhante ao encontrado na Figura 7.3 devido a extrapolação do limite de 10 tentativas de estacionarização com posterior adoção/aplicação do modelo AR(1) (situação de contorno).

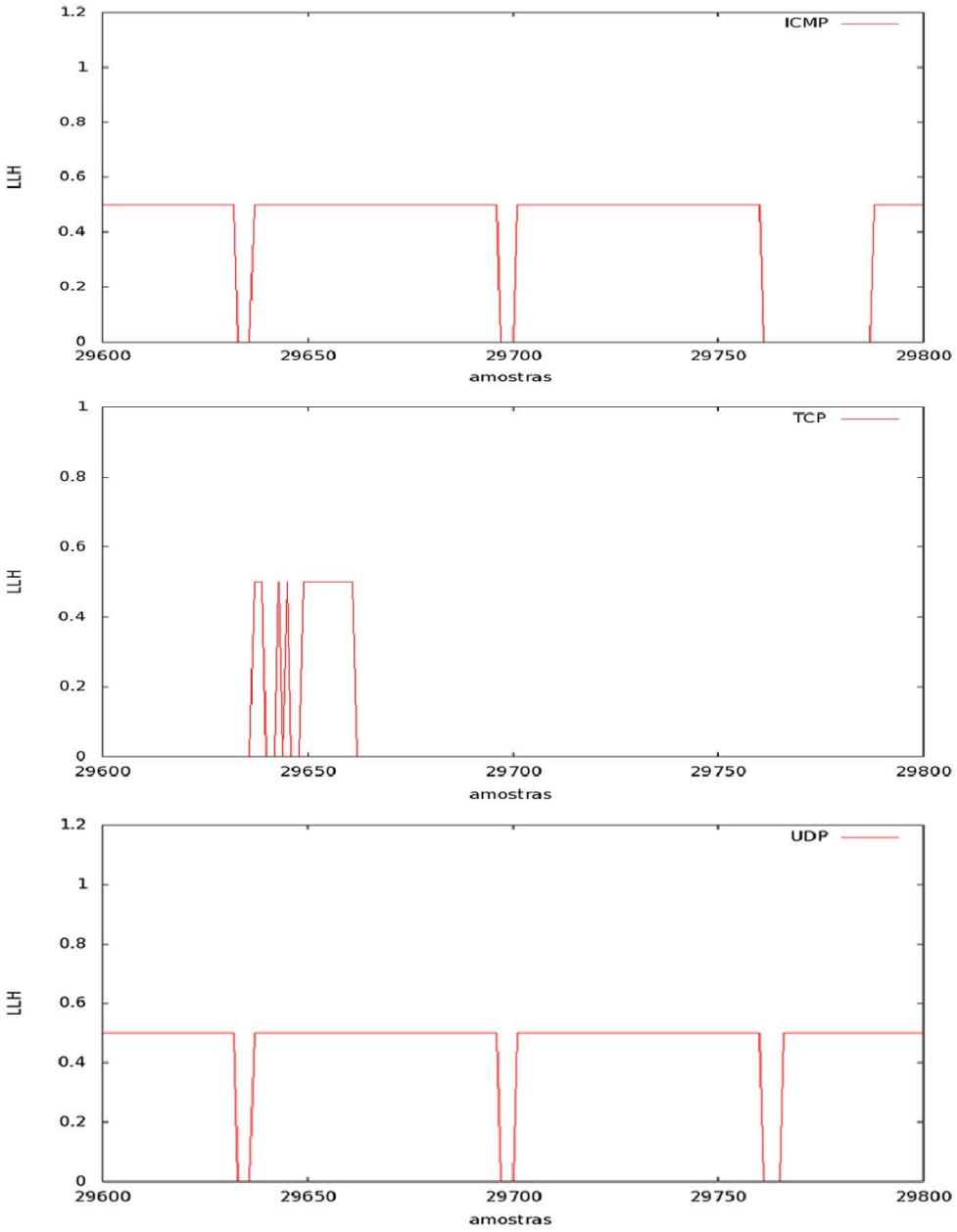

Figura 7.5 - Demonstração das correlações entre ICMP, TCP e UDP, obtidas em ARIMA com janela de 64 elementos.

7.2.2 Correlação das Anomalias da base UFSM 2010

Os testes realizados para a base de dados da UFSM empregaram janelas de 32 e 64 elementos, por apresentarem um maior de número de anomalias identificadas. Porém

o cuidado tomado para realizar as detecções na base não conseguiu ser traduzido em detecções de ataques por meio de correlação temporal.

Este fato decorre que a correlação das variáveis tráfego de rede faz uso de vetores de anormalidade como base na correlação e em cada um deles é armazenado informações sobre anomalias, onde no caso deste trabalho os vetores de anormalidade são formados pelos protocolos ICMP, TCP e UDP. A correlação obtida destes resulta em um coeficiente de uso da rede, este coeficiente apresenta-se entre 0 e 1.

Como não foram identificadas anomalias para os protocolos ICMP e UDP, sendo somente verificadas anomalias no protocolo TCP como apresentado na Tabela 7.2, não foi possível verificar se as anomalias apresentadas são originadas por ataques que possuem efeitos sobre outras variáveis formadoras do tráfego de rede.

7.2.3 Comentários sobre a Detecção de Correlações

O comportamento dos vetores de anormalidade sobre os três protocolos analisados possui características que dificultam verificar a ocorrência de ataques. Isto provavelmente decorre do fato de que os protocolos utilizados não possuem correlação entre si. Outro fator que dificultou a identificação de ataques diz respeito ao instante de ocorrência das anomalias, que não possuem correlação temporal em sua ocorrência.

A ocorrência destas duas características foi encontrada em resultados que não descrevem efetivamente o estado da rede. Como resultado da correlação das linhas de comunicação um resultado entre 0 e 1. Porém como não existe correlação direta entre os serviços avaliados e as ocorrências de anomalias percebidas no tempo, ao aplicar o algoritmo que equaciona os estados de rede não obterá como resultado final valor representativo capaz de descrever o estado da rede, visto que o estado da rede tende a 0. Valor que descreveria ausência de anomalias de rede.

Para que fosse obtido êxito na correlação dos eventos de anormalidade é necessário que os dados possuam correlação entre si e no caso das linhas avaliadas elas não apresentam correlação entre elas o que torna muito difícil obter a correlação temporal dos vetores de anormalidade criados com as anomalias percebidas. É percebido nos dados da UFSM 2010 que não existiam correlações, pois não foram detectadas anomalias em outras variáveis analisadas. O que nos leva a concluir que a

correlação direta entre protocolos não é um caminho a seguir para a detecção de anomalias e ataques.

7.3 Conclusões Parciais

Neste capítulo foi descrito como foram realizados os testes da implementação do algoritmo *DIBSETiR* para realizar os testes foi utilizada as bases de dados DARPA 99 e UFSM 2010. O capítulo foi dividido em duas etapas. A primeira formada pelo processo de avaliação do comportamento de rede e determinação de anomalias e a segunda formada pela correlação das anomalias encontradas.

Para os testes que avaliaram a ocorrência de anomalias foi obtido resultados positivos para a redução do número de detecções de anomalias, para a base DARPA e UFSM, pela substituição do AR(1) pelo ARIMA com identificação dinâmica do modelo. Ainda foi percebido que algumas detecções aconteceram em pontos semelhantes devido ao fato de a implementação do algoritmo baseado no ARIMA recair no modelo AR(1) caso não se consiga a estacionarização após 10 tentativas.

A segunda etapa dos testes diz respeito sobre a tentativa de correlacionar as varáveis de tráfego de rede. As variáveis formadas pelos protocolos de rede ICMP, TCP e UDP. Para a base de dados DARPA 99 foram identificadas algumas correlações/ataques. Identificações que ocorreram em número muito pequeno para sinalizar como uma alternativa para auxiliar a detecção de ataques por avaliação de anomalias. E para a base de dados UFSM 2010, a detecção utilizando correlação não obteve sucesso. Visto que não foram detectadas anomalias para os protocolos ICMP e UDP o que impossibilita o emprego de detecção de ataques por correlação das variáveis de tráfego de rede, requerendo outro método para a confirmação das anomalias detectadas.

8 CONSIDERAÇÕES FINAIS

Os negócios “virtuais”, que usam a internet, a agilidade e dinâmica obtidas pelas empresas e seus usuários podem gerar cobiça, principalmente por parte de concorrentes ou pessoas mal intencionadas, o que fomenta a exploração de vulnerabilidades das aplicações e da rede de computadores. Como resultado, servidores plugados na rede são alvo de ataques de negação de serviço (DoS), ataques que costumam ser observados pelo estresse indevido de protocolos de rede utilizados por servidores.

Este trabalho propõe uma maneira alternativa para detecção de ataques DoS que exploram vulnerabilidades que afetam os protocolos TCP, ICMP e UDP. A proposição é uma variante do modelo proposto em (THOTTAN e JI, 2003) e explora a modelagem ARIMA de séries temporais para identificar comportamentos que fogem ao comportamento usual do tráfego de rede de computadores, possibilitando a análise de dados não estacionários. Deste modo, o modelo ARIMA é empregado para melhorar a qualidade de representação do modelo de conhecimento do tráfego de redes aplicada na filtragem dos dados coletados. No modelo proposto, o processo de identificação de ataques foi dividido em duas etapas, a primeira formada pela identificação de anomalias de rede e a segunda formada pela determinação de ataque com base nas correlações entre anomalias encontradas em diferentes descritores de rede.

Para a identificação de anomalias é aplicado um filtro baseado no ARIMA seguido de um analisador de mudanças abruptas para geração de pré-alarmes. O método avalia a variância dos resíduos, o comportamento dos parâmetros que definem o comportamento da série e um coeficiente de intensidade, dado por LLH, sendo equivalente a análise de uma assinatura de comportamento usual contrastada com o fluxo de dados atual.

Para a identificação de ataques foi utilizado correlação das anomalias encontradas pelo algoritmo GLR. Nesta etapa foi correlacionado as variáveis descritoras de tráfego (TCP, ICMP e UDP) para confirmar que a anomalia encontrada é um ataque. Diferentemente do método proposto em (THOTTAN e JI, 2003), não foi utilizada previsão na composição dos sistemas e sim as variáveis correlacionadas diretamente.

Os resultados obtidos na primeira etapa (detecção de anomalias) mostram que o uso do ARIMA é semelhante ao do AR(1), mas em alguns casos apresentou-se uma redução no número de falsos positivos. Em algumas amostras a redução apresentou-se mais significativa que em outras situações. Para a base de dados DARPA 99 encontrou-se anomalias nas três variáveis analisadas. Porém em contra partida para a base de dados da UFSM foram encontradas anomalias somente na variável correspondente ao protocolo TCP. Para as demais variáveis descritoras de tráfego não foi encontradas anomalias. Os resultados também demonstram que a abordagem proposta não é adequada para detectar ataques em variáveis não correlacionadas, ou seja, a técnica não é adequada para identificar um ataque através do uso das variáveis independentes, requerendo uma análise prévia para seleção de variáveis dependentes.

8.1 Sugestões para Trabalhos Futuros

Foi percebido neste trabalho que lacunas ainda podem ser preenchidas, especialmente na identificação de anomalias e seleção de variáveis. O emprego de outros modelos de series temporais, tal como as que consideram a sazonalidade dos dados (modelos SARIMA), podem contribuir para uma maior eficiência na tarefa de localização de potenciais anomalias. A área de identificação de fluxos de dados correlacionados, que possibilita selecionar variáveis de interesse, também é de especial interesse, pois a identificação de correlação entre descritores pode ser chave para a redução de falsos positivos.

REFERÊNCIAS BIBLIOGRÁFICAS

AXELSSON, S. e SANDS, D. "**Understanding Intrusion Detection through Visualization**". Advances in Information Security, V. 24, 145 p. 34 illus Springer 2006.

ANT, Allan. DPRO-95367 - Intrusion **Detection Systems (IDSs): Perspective Summary Technology Overview**. Gartner Research. Janeiro de 2002.

BASSEVILLE, M.; NIKIFOROV, I. **Detection of Abrupt Changes, Theory and Application**. Englewood Cliffs: Prentice-Hall, 1993.

BECKER, F.; PETERMANN, M. **Intrusion Detection Systems Elevated to the Next Level.** 22nd Chaos Communication Congress. Dezembro de 2005.

BORBELY, Alexandre. Investimentos em acesso à Internet e seus possíveis impactos nos processos produtivos das empresas. In: **Revista da Faculdade de Administração e Economia,** São Paulo: Universidade Metodista de São Paulo, v.2, n.2, p.76-107, 2010.

BOX, G.; JENKINS, G.; REINSEN, G. **Time Series Analysis: Forecast and Control**. Prentice Hall, 592p., 1994.

CABRERA, J. B. D.; LEWIS, L.; QIN, X.; LEE, W.; MEHRA, R. K. **Proactive Intrusion Detection and Distributed Deny of Service Attacks – A Case Study in Security Management**. In: Journal of Network and Systems Management, V.10, n 2, June 2002.

CERTa. **CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack**. Fevereiro de 1996. http://www.cert.org/advisories/CA-1996-01.html Acesso 15/10/2008.

CERTb. **CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks**. Setembro de 1996. http://www.cert.org/advisories/CA-1996-21.html. Acesso 5/10/2008.

CERTc. **CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks**. Janeiro de 1998. em http://www.cert.org/advisories/CA-1998-01.html. Acesso 15/10/2008.

CERTd. **CERT Advisory CA-2000-21 Denial-of-Service Vulnerabilities in TCP/IP Stacks**. Novembro de 2000. http://www.cert.org/advisories/CA-2000-21.html. Acesso 15/10/2008.

CHEBROLU, S.; ABRAHAM, A.; e THOMAS, J. P. **Feature deduction and ensemble design of intrusion detection systems**. Computers & Security. 06 de setembro de 2004.

BOER, P.; & PELS, M.; BOER, P.; & PELS, M. **Host-based Intrusion Detection Systems**. Revision 1.10 – Amsterdam University, Fevereiro de 2005.

DARMOHRAY, T.; OLIVER, R.; **Hot Spares for DoS attacks. The Magazine of USENIX and SAGE**. V. 25, N. 4, July 2000.

DEBAR, H. **An Introduction to Intrusion-Detection Systems. IBM Research, Zurich Research Laboratory, Switzerland**. Proceedings of Connect'2000, Doha, Qatar, 2000.

DIAS, R. A.; CAMPONOGARA, E.; FARINES, J.;WILLRICH, R.; CAMPESTRINI, A. **Otimização Lagrangeana em Engenharia de Tráfego para Redes IP sobre MPLS.** XXI Simpósio Brasileiro de Redes de Computadores p. 475 - 490, maio 2003.

EHLERS, R. S. **Análise de Séries Temporais. Laboratório de Estatística e Geoinformação**. Universidade Federal do Paraná. 4 ed., 2007.

HAJJI H., **Baselining Network Traffic and Online Faults Detection**. Communications, 2003, ICC '03. IEEE International Conference on, v. 1, p. 301-308 maio 2000.

HOLANDA FILHO, R.; MAIA, J. E. B.; CARMO, M. F. F**. Identificação da Componente de Tráfego de Ataque baseada em Discriminantes Estatísticos**. Universidade de Fortaleza e Universidade Estadual do Ceará. Anais do XXVII Congresso da SBC. WPerformace – p. 690 – 720 - V WorkShop em Desempenho de Sistemas e de Comunicação Julho de 2007.

JAVA. http://java.sun.com/javase/downloads/index.jsp, último acesso abril de 2008.

LERMEN, E. G.; BRUNO, G. G. E. **Framework para Detecção e Filtragem de Alertas de Intrusão utilizando Redes Bayesianas.** VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, p. 301 – 309, Agosto 2008.

LUNARDI, R. C.; DALMAZO, B. L.; AMARAL, E. M. H.; NUNES, R. C. DIBSet: um Detector de Intrusão por Anomalias Baseado em Séries Temporais**. Universidade Federal Santa Maria, VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais WTICG p.311 - 320, Setembro 2008.**

MAFRA, P. M.; FRAGA, J. S.; MOLL, V.; SANTIN, A. O. **POLVO-IIDS: Um Sistema de Detecção de Intrusão Inteligente Baseado em Anomalias** VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, Setembro 2008.

MIT LINCOLN LABORATORY. http://www.ll.mit.edu/IST/ideval/index.html, Último acesso novembro de 2007.

POUW, K. D.; GEUS, P. L. **Uma Análise das Vulnerabilidades dos Firewalls**. Departamento de Ciência da Computação – Universidade Estadual de Campinas. Campinas, São Paulo -WAIS Março de 1996.
.

PENG, T.; LECKIE, C., RAMAMOHANARAO, K**. Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems**. Department of Computer Science and Software Engineering. The University of Melbourne, Australia. ACM. ACM-CSUR. Março 2007.

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

KAHNEY, L. A Cabeça de Steve Jobs tradução de Inside Steves Brain AGIR EDITORA LTDA Rio de Janeiro – RJ 2008.

JEMILI, F.; ZAGHDOUD, M.; AHMED, M. B. **A Framework for an Adaptative Intrusion Detection System using Bayesian Network**. IEEE Intelligence and Security Informatics p. 66 – 70. Maio de 2007.

KALINLI, A.; SAGIROGLU, S. **Elman network with embedded memory for system identification**, Journal of Information Science and Engineering 22 (6) (2006) 1555–1568.

KRUEGEL, C.; MUTZ, D.; ROBERTSON, W.; VALEUR, F**. Bayesian Event Classification for Intrusion Detection**. IEEE 19th Annual Computer Security Applications Conference (ACSAC '03) p. 14, Dezembro de 2003.

MYSQL. www.mysql.com, último acesso abril de 2008.

MORETTIN**,** P. A.; TOLOI, C. M. C. **Análise de Series Temporais**. ABE – São Paulo Projeto Fisher Editora Edgard Blücher LTDA. 2004.

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.

R. www.**r-project**.org, último acesso agosto de 2011.

RFC 791. Internet Protocol. Setembro 1981.

RFC 792. Internet Control Message Protocol. Setembro 1981.

RFC 793. Transmission Control Protocol. Setembro 1981.

RFC 2460. Internet Protocol (IPv6). http://www.ietf.org/rfc/rfc2460.txt. Dezembro de 1998.

RFC 2461. Neighbor Discovery for IP Version 6 (IPv6). http://www.ietf.org/rfc /rfc 2461.txt. Dezembro de 1998. Último acesso 26/08/2008.

SCARFONE, K. e MELL P. **Guide to Intrusion Detection and Prevention Systems (IDPS)**. National Institute of Standards and Technology 2007.

SENA, J. C., GEUS, P. L., AUGUSTO, A. **Impactos da Transição e utilização do Ipv6 sobre a Segurança de Ambientes Computacionais**. Instituto de Computação Unicamp. Campinas, São Paulo. WSeg II Workshop em Segurança *de* Sistemas Computacionais, Búzios RJ, maio de 2002.

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

SOLHA, L. E. V. A.; TEIXEIRA, R. C.; PICCOLINI, J. D. B. **Tudo que Você Precisa Saber Sobre os Ataques DDoS**. Acessado em 07/10/2008 em http://www.rnp.br/newsgen/0003/ddos.html.

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

THOTTAN, M.; JI, C. C. **Proactive anomaly detection using distributed intelligent agents**, *IEEE Network*, vol. 12, pp. 21–27, Sept./Oct. 1998.

THOTTAN, M.;JI, C. **Anomaly Detection in IP Networks**. IEEE Transactions on Signal Processing, vol. 51, 8, agosto de 2003.

TONG, X.; WANG, Z.; YU, H. (October 2009). **A research using hybrid RBF/Elman Neural networks for intrusion detection system secure model**, Computer Physics Communications 180 (2009), pp.1795-1801.

WANG, H.; ZHANG, D.; SHIN, G. C. **Detecting SYN Flooding Attack.** Department of Computer Science. University of Michigan. In Proceedings. IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. p.1530-1539, Nova York, NY, June 2002.

ZHAI, Y.; NING, P.; IYER, P.; REEVES, D.S. **Reasoning about complementary intrusion evidence.** In Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC 04) p.39-48, December 2004.