DETECÇÃO DA INTRUSÃO UTILIZANDO MODELOS NEURO-DIFUSOS

Antonio Alexandre de Castro Soares

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Jorge Lopes de Souza Leão, Dr.Ing.

Prof. Luiz Pereira Caloba, Dr.Ing.

Prof. Flávio Joaquim de Souza, Dr.

RIO DE JANEIRO, RJ - BRASIL

JULHO DE 2005

SOARES, ANTONIO ALEXANDRE DE CASTRO

Detecção da Intrusão Utilizando Modelos Neuro-Difusos [Rio de Janeiro] 2005

XV, 146 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2005)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Redes de Computadores

2. Segurança da Informação

3. Modelos Inteligentes

4. Lógica Difusa

I. COPPE/UFRJ II. Titulo (série)

A descoberta incide sobre o que já existe, atualmente ou virtualmente; portanto, cedo ou tarde ela seguramente vem.

Gilles Deleuze

Agradecimentos

À minha querida Claudia Tallemberg, por todo o seu companheirismo, carinho e suporte em todos os momentos difícies, sabendo equilibrar sugestões e críticas úteis não só ao desenvolvimento deste trabalho bem como na vida.

Aos meus filhos, Maria Alice e Dimitri que sempre me ajudaram com o seu carinho bem como o seu exemplo de força num futuro possível.

À minha amiga Jane Mourão, que com suas palavras afiadas, me ajudaram facilitando este percurso com sugestões que aplicados a vida tornaram o desenvolvimento deste trabalho um processo de auto-descobrimento.

Aos amigos José Augusto, Fernando Carvalho, Eduardo Dutra e Antonio C. M. Alvim pelo apoio e suporte no início de minha vida profissional e acadêmica.

Ao meu amigo Roberto Maia por seu apoio e companherismo tornando este trabalho um elemento concreto. Ao meu orientador Jorge Leão e toda a equipe do GTA e aos funcionários do Programa de Engenharia Elétrica da COPPE/UFRJ pelo apoio e profissionalismo no atendimento.

Aos professores Luiz Pereira Caloba e Flávio Joaquim de Souza pela participação na banca examinadora.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DETECÇÃO DA INTRUSÃO UTILIZANDO MODELOS NEURO-DIFUSOS

Antonio Alexandre de Castro Soares

Julho/2005

Orientador: Jorge Lopes de Souza Leão

Programa: Engenharia Elétrica

O uso crescente dos computadores em diversas áreas de negócios e serviços exigiu que o tema de segurança se configurasse como um dos principais problemas no universo da informação.

Atualmente o cenário de ameaças observados nos tráfegos de redes é classificado como polimórfico, ou seja, o ataque normalmente encontra-se oculto por perfis de uso fortemente auto-similares. Assim, um ataque pode ser elaborado de diferentes maneiras com o objetivo de explorar uma vulnerabilidade específica presente em um serviço disponível em um ambiente remoto, dificultando sua diferenciação de tráfegos livres de anomalias.

A motivação deste trabalho está na busca de alternativas aos atuais mecanismos de segurança, usualmente limitados a compreender a ameaça de maneira rígida e pouco flexível. Para tanto é proposto o uso da lógica difusa em associação aos mecanismos de aprendizado das redes neurais. Esta combinação permite identificar com maior precisão a presença de um ataque em meio a diversos tráfegos considerados normais. A qualidade desta avaliação é realizada por meio de um processo de decisão mais robusto e capaz de suportar diversos graus de diferenciação do ataque original e assim diminuir a quantidade de alarmes falsos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INTRUSION DETECTION USING NEURO-FUZZY MODELS

Antonio Alexandre de Castro Soares

July/2005

Advisor: Jorge Lopes de Souza Leão

Department: Electrical Engineering

The increasing use of the computers in diverse business-oriented areas and services demanded that the security subject if configured as one of the main problems in the universe of the information.

Currently the scene of threats observed in the traffics of nets is classified as polymorphous, or either, the attack normally meets for use profiles strong self-similar occult. Thus, one has attacked can be elaborated in different ways with the objective to explore a present specific vulnerability in an available service in a remote environment, making it difficult its differentiation of free traffics of anomalies.

The motivation of this work is in the search of alternatives to the current mechanisms of security, usually limited to understand the threat in rigid and little flexible way. For in such a way the use of the fuzzy logic in association to the mechanisms of learning of the neural nets is considered. This combination allows to identify with bigger precision the presence of an attack in way the diverse considered traffics normal. The quality of this evaluation is carried through by means of a process of decision more robust and capable to support diverse degrees of differentiation of the original attack and thus to diminish the amount of false alarms.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O uso crescente dos computadores em diversas áreas de negócios e serviços exigiu que o tema de segurança se configurasse como um dos principais problemas no universo da informação. As grandes organizações estão conscientes sobre os principais ataques que objetivam adquirir de maneira ilícita suas informações buscando assim prejudicar suas operações.

Atualmente estão disponíveis diversas ferramentas tais como: firewalls e sistemas detectores da intrusão (SDI), ambas concebidas com o objetivo de estabelecer uma quantidade de segurança aceitável às operações das organizações. Embora as ferramentas citadas protejam as redes de uma organização estas, não podem garantir a detecção de todos os ataques bem como estabelecer imunidade as novas estratégias todo o tempo.

Com o aumento da demanda por novas ferramentas de segurança, é observada uma lacuna expressiva do tempo de disponibilização destas facilidades, uma vez que existe a necessidade de testes efetivos antes de sua distribuição afim de assegurar sua eficácia. Associada a questão do tempo de oferta, existe o problema do modelo geral presente nas ferramentas atuais, pois todas estão centradas em questões específicas dos eventos observados localmente, desprezando o relacionamento presente entre as diversas conexões assinaladas em uma infra-estrutura. Devido as características destes modelos, constatamos deslizamentos sensíveis principalmente quando esta infra-estrutura está sob um cenário de ataque expressivo. Neste contexto as atuais ferramentas produzem um número elevado

de alarmes falsos tornando impraticável a elaboração de estratégias eficientes de reação.

1.1 Motivação

Segundo Parker[57] são seis os elementos fundamentais a serem garantidos em qualquer infra-estrutura de segurança da informação: 1) Disponibilidade, o sistema deve estar disponível para uso sempre que desejado pelos usuários; 2) Usabilidade, o sistema que opera os diversos elementos de informação deve ser adequado a necessidade dos usuários possibilitando a estes que desenvolvam suas atividades dentro de um escopo conhecido e passível de acompanhamento; 3) Autenticidade, o sistema deverá oferecer mecanismos que garantam a identidade do usuário e indique a estes os privilégios necessários ao desempenho de suas ativiades; 4) Integridade, o sistema deverá ter mecanismos internos que permitam aos usuários ter a garantia que suas transações foram realizadas com sucesso; 5) Confiabilidade, o sistema deverá garantir que somente serão oferecidos informações sensíveis a usuários que pertençam a um processo específico ou autorizado pelo proprietário da informação; 6) Auditoria, o sistema deverá prover mecanismos que permitam o entendimento das diferentes atividades de um usuário específico e seus desdobramentos.

Infelizmente os requerimentos apontados por Parker, são de difícil implementação em um cenário real, assim são estabelecidos limites práticos entre os diferentes acessos dos usuários objetivando um controle em múltiplos níveis para cada membro da rede de serviços. Tal adaptação do modelo geral estabele brechas significativas, o que permite como observado pelas estatísticas dos centros de acompanhamentos de ameaças um contínuo aumento dos eventos de segurança.

Assim, torna-se necessário o uso de sistemas adaptativos que permitam compreender os diferentes requerimentos presentes nos diversos componentes da infra-estrutura. Isto possibilita o entendimento centrado na tendência de perigo das estratégias utilizadas por atacantes internos ou externos a rede de serviços de uma organização.

Atualmente procura-se estabelecer esta segurança com ferramentas como firewalls, que operam sob regras pré-estabelecidas por uma políticas de segurança que limita-se a

bloquear ou liberar o acesso de tráfegos sem qualquer julgamento quanto a interação entre as diversas solicitações impossibilitando assim o conhecimento de qualquer estratégia.

Associadamente, emprega-se o uso de detectores especializados em examinar em tempo real a presença de anomalias, porém esta abordagem é limitada pois as ameaças necessitam ser especificadas nos bancos de dados. Portanto, esta abordagem apresenta enormes lacunas para um determinado ataque cujo mapeamento de seu comportamento não tenha sido formalmente descrito.

Por isto a motivação deste trabalho está na busca de alternativas aos atuais mecanismos de segurança, usualmente limitados a compreender a ameaça de maneira rígida e pouco flexível. Para tanto é proposto o uso da lógica difusa em associação aos mecanismos de aprendizado das redes neurais. Esta combinação permite identificar com maior precisão a presença de um ataque em meio a diversos tráfegos considerados normais. A qualidade desta avaliação é realizada por meio de um processo de decisão mais robusto e capaz de suportar diversos graus de diferenciação do ataque original e assim diminuir a quantidade de alarmes falsos.

1.2 Objetivo

O cenário de ameaças presentes nos tráfegos de redes atuais é classificado como polimórfico, ou seja, o ataque normalmente encontra-se oculto por perfis de uso fortemente auto-similares. Assim, um ataque pode ser elaborado de diferentes maneiras com o objetivo de explorar uma vulnerabilidade específica presente em um serviço disponível em um ambiente remoto, dificultando sua diferenciação de tráfegos livres de anomalias.

Devido a este cenário altamente dinâmico, a área de detecção da intrusão busca de maneira contínua desenvolver novos algoritmos que acomodem as diferentes mudanças do cenário geral de ameaças, buscando a cada tentativa melhorias operacionais, ou seja, com baixas taxas de falsos negativos e positivos.

Neste contexto a proposta deste estudo consiste na elaboração de um modelo capaz de capturar a anomalia abalizadas em regras de conhecimento construídas a partir do uso

da computação flexível, que ao contrário da computação tradicional está consolidada na lógica difusa em associação às redes neurais, acomodando assim a imprecisão inerente dos problemas complexos presentes na atualidade.

Neste trabalho foram utilizados os tráfegos de referência providos pelo laboratório Lincoln do MIT como base das atividades de pesquisa. Esta escolha possibilita a comparação dos resultados obtidos com outras abordagens e técnicas presentes no cenário acadêmico. Para tanto foi necessária a construção de uma plataforma de software capaz de suportar as diferentes informações essenciais ao desenvolvimento desta pesquisa.

Esta plataforma é composta por quatro grandes módulos organizados de maneira seqüencial. São eles: aquisição de dados, carga de dados, treinamento e decisão. O módulo de maior importância é sem dúvida o responsável pelo treinamento, onde estão presentes todas as atividades de mapeamento das características que descrevem o comportamento de cada ataque.

Como resultado deste trabalho é esperada a comprovação da eficiência do uso do modelo neuro-difuso na detecção da intrusão em ambientes computacionais conectados a redes.

1.3 Organização do Trabalho

Este trabalho é organizado em dez capítulos onde o Capítulo 2, possibilita o estabelecimento de uma visão geral dos principais mecanismos utilizados na área de segurança apontando o desenvolvimento abrangente dos principais conceitos utilizados. O Capítulo 3 apresenta o cenário onde os dados de referência utilizados neste estudo foram obtidos, observando suas estratégias de construção bem como compreendendo suas limitações de uso; o Capítulo 4 permite um aprofundamento das características particulares de cada ataque escolhido para o desenvolvimento dos referidos modelos de detecção. O Capítulo 5, estabelece a partir do entendimento da natureza de cada ataque presente no cenário de referência a arquitetura de software utilizada para abrigar todas as informações necessárias ao desenvolvimento da solução; o Capítulo 6 aprofunda as caracterís-

ticas de ataque apresentadas no Capítulo 4, classificando todas as componentes presentes nas anomalias escolhidas como objeto deste estudo.

O Capítulo 7 apresentam os modelos teóricos da lógica difusa. O Capítulo 8 detalha o esquema de treinamento utilizado para capturar as anomalias objetos deste trabalho. O Capítulo 9 apresentam os resultados alcançados e suas respectivas análises; o Capítulo 10 introduz idéias a serem complementadas ao estudo atual, possibilitando o desenvolvimento de novos trabalhos.

Capítulo 2

Aspectos Preliminares

Conforme relatório elaborado pelo Computer Emergency Response Team Coordination [3] observamos nos últimos anos um aumento significativo dos incidentes de segurança[1], este crescimento apresentado na Figura-2.1, deve-se principalmente à proliferação do número de sistemas computacionais ligados em rede. Este crescimento abrupto observado em 1999 coincide com o surgimento dos serviços on-line e a expansão da Internet.

Os invasores, usualmente conhecidos como hackers, estão constantemente alterando suas estratégias, seja para adaptar suas técnicas às novas tecnologias, ou simplesmente para evoluir os atuais programas exploratórios. Tais atividades agregam diversas experiências anteriores, o que permite a ampliação de novas possibilidades de ataque dos sistemas atuais.

No início as atividades dos invasores se restringiam à obtenção de acessos a um determinado sistema, por meio de procedimentos chamados de password guessing ou força bruta. Este ataque busca por meio de tentativas sucessivas de solicitação de serviço advinhar a senha de um usuário. Atualmente os ataques ganharam sofisticação, tornando-se mais elaborados e exigindo por parte do invasor um conhecimento técnico mais apurado. Tal conhecimento está presente nos ataques do tipo *back door*, *sniffing*, *session hijacking*

[1] Por incidente de segurança, devemos entender como sendo a tentativa de quebra das políticas de segurança que definem o oferecimento dos recursos de um determinado sistema em um ambiente compartilhado ou não.

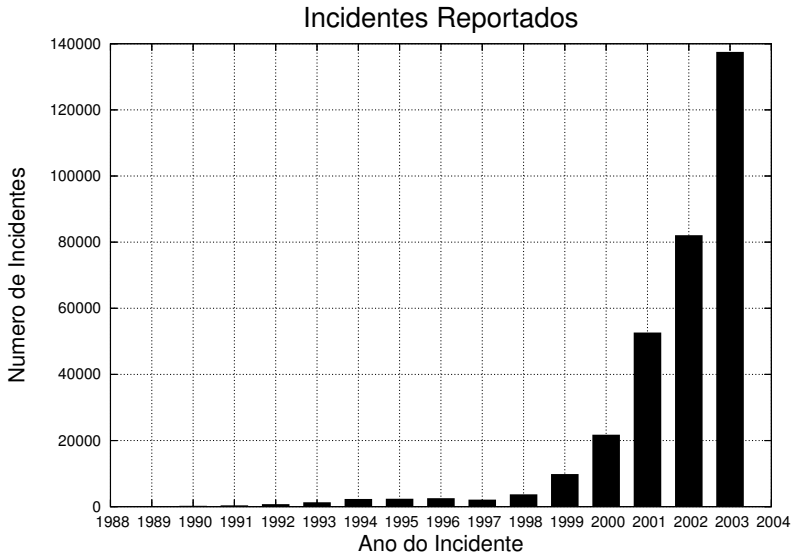

Figura 2.1: Evolução do Número de Incidentes de Segurança.

e *distributed denial of service*.

Atualmente as estratégias de ataque evoluiram com o objetivo de paralisar serviços baseados, derivados ou periféricos à tecnologia cliente-servidor, como por exemplo a geração de tráfegos expúrios a serviços e aplicações.

Com este panorama geral é possível elaborar um mapeamento desta evolução que está apresentado na Figura 2.2 onde, por meio da rápida difusão das diferentes técnicas, os invasores podem simplesmente utilizar os programas já desenvolvidos por outros atacantes mais experientes, tornando assim a possibilidade de ataque a diferentes sistemas mais usual.

As técnicas apresentadas anteriormente seguem de modo geral a estratégia apresentada na Figura-2.3; onde a primeira fase objetiva o conhecimento do contorno de um sistema. Para tanto, busca-se reunir diversas informações sobre o sistema que se deseja realizar um determinado ataque, utilizando técnicas de identificação dos diversos serviços oferecidos, mapeando suas tecnologias e fornecedores.

Na segunda fase, o invasor busca aprofundar os dados já obtidos e assim identificar a existência de alguma vulnerabilidade já conhecida para um determinado serviço. Caso este fenômeno seja verificado, aplicam-se os programas exploratórios específicos de

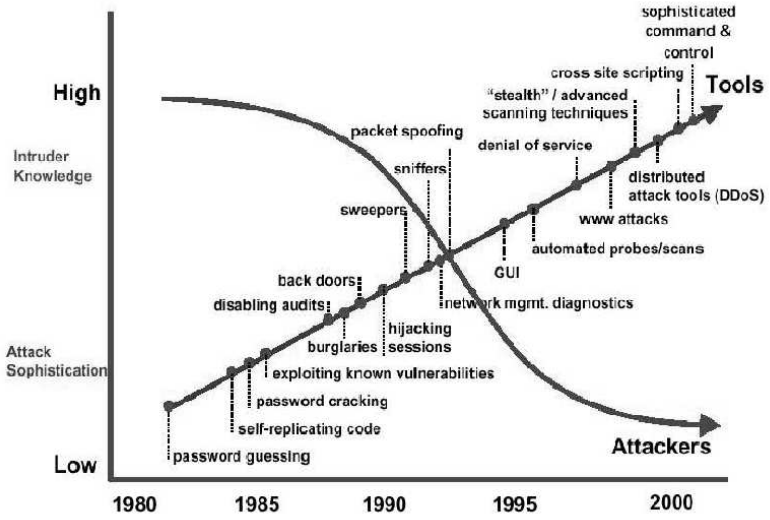

Figura 2.2: Sofisticação do ataque vs conhecimento do invasor.

maneira direta.

A terceira fase é considerada a mais agressiva por ser direcionada não apenas aos sistemas e serviços mas à infra-estrutura de suporte, tais como: roteadores, switchs, firewalls, etc. afetando um longo espectro de processos estratégicos. Logo, em caso de sucesso, a paralização torna-se extremamente crítica pois a lacuna de segurança está presente em diferentes pontos de falha tornando extremamente difícil a identificação da causa raíz.

As estratégias apresentadas anteriormente são desenvolvidas para cenários específicos que partem da premissa da existência de uma vulnerabilidade em um serviço, ou em algum protocolo. Este ciclo é apresentado na Figura-2.4 e apresenta de maneira esquemática dois processos: um de intrusão e outro de contra medida. Paralelamente ao desenvolvimento de um programa explorador existe o acompanhamento de orgãos específicos que objetivam orientar os esforços de fabricantes e comunidades de software livre, a contornar os incidentes de segurança por elas reportados.

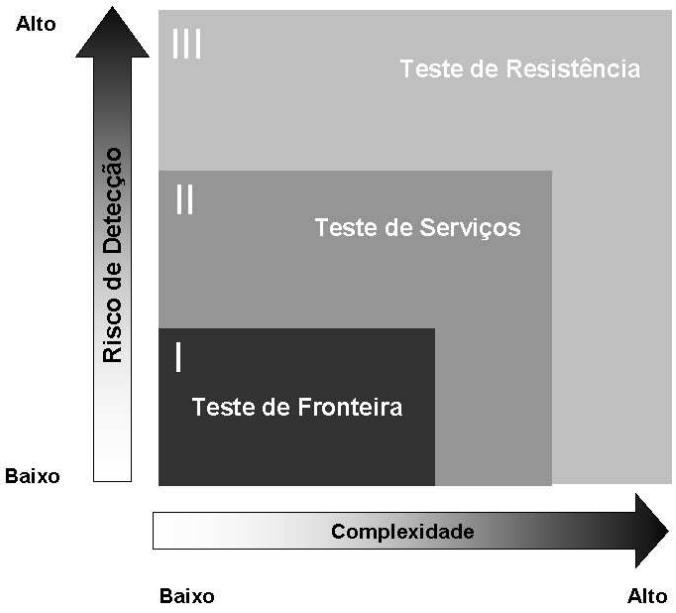

Figura 2.3: Mapeamento das fases.

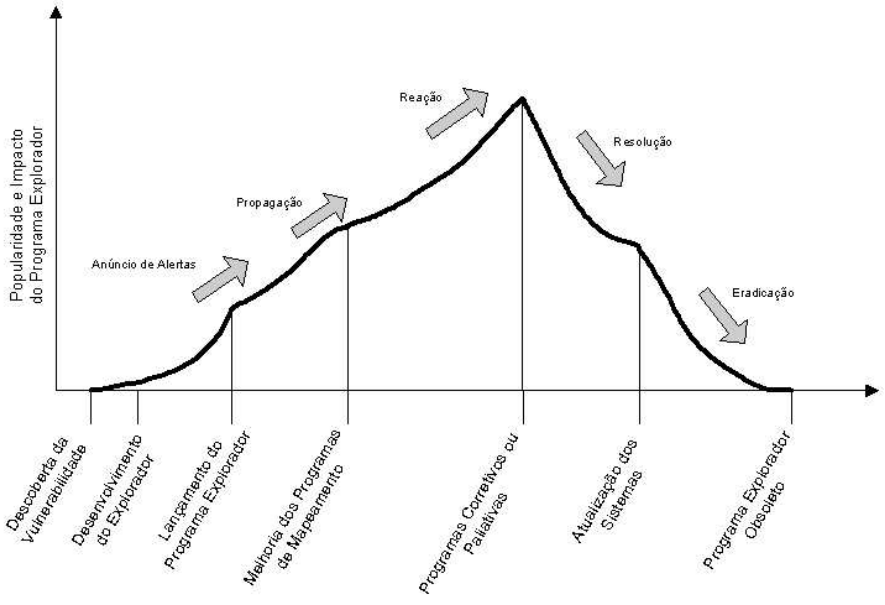

Figura 2.4: Ciclo de Vulnerabilidade.

2.1 Modelo Geral de Segurança

As diferentes ameaças têm seu ciclo de desenvolvimento localizado em ambientes de rede, e são definidas como sendo um espaço de interações complexas, onde o comportamento individual de uma ou mais entidades tais como roteadores, switches, hubs, etc definem o comportamento geral da rede[5]. É neste cenário que a área de segurança vem trabalhando em modelos que passam servir de referência para o desenvolvimento de plataformas que suportem diferentes abordagens que auxiliem a compreensão da intrusão.

Como resultado destas reflexões, foi elaborado o modelo chamado de modelo de gerações, como mostrado na Figura-2.5. Este modelo foi criado a partirdo entendimento das interações entre fronteiras e possibilidades de ameaças.

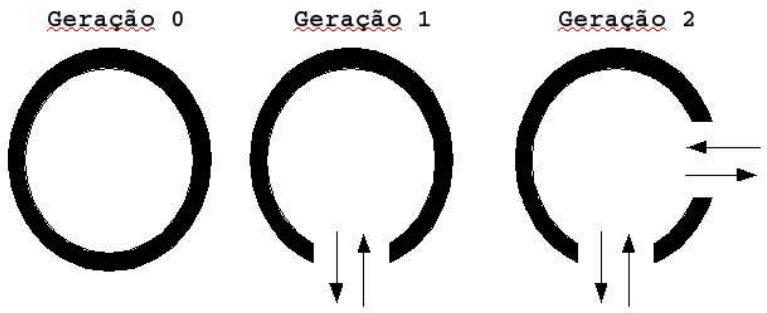

Figura 2.5: Modelo de Gerações.

A geração zero representa ambientes que não possuem conexão a redes externas e internamente não existe nenhuma conexão entre as facilidades, logo esta geração pode ser descrita como "pré-interconexão" pois depende fortemente da segurança física.

A segunda geração um representa ambientes que possuem uma única conexão externa. Internamente existe uma infra-estrutura de conexão presente entre as facilidades. Esta geração apresenta de maneira clara as fronteiras básicas do perímetro de defesa onde todas as conexões externas são fortemente verificadas, buscando garantir acessos limitados ao mínimo necessário, normalmente verificados por dispositivos de controle chamados de firewalls. Este modelo apresenta a dependência de um controle de acesso centralizado, associado a outros mecanismos de controle internos.

A geração dois representa ambientes mais complexos onde os mecanismos de controle estão oferecidos em uma arquitetura em camadas. Para cada camada existe um ou mais dispositivos de controle e análise em tempo real para identificação das diferentes ameaças a que cada camada está exposta. Assim, esta geração é fortemente dependente da coordenação por meio de políticas e processos que garantam que todos os componentes atuem de maneira organizada, visando garantir a estabilidade da infra-estrutura.

2.2 Termos e Definições

A área de segurança apresenta uma variedade muito grande de termos e definições que variam de acordo com cada autor, além de serem fortemente dependentes do contexto. As definições apresentadas nesta seção seram baseadas nos estudos de Laprie[32] aplicadas as disciplinas de TI. Associadamente apresentaremos as definições descritas no trabalho de Shirey[2].

2.2.1 Falta, Erro e Falha

Laprie[32] define como falta, um julgamento ou hipótese responsável pela causa de um erro, e erro como sendo a parte de um estado de um sistema que está pronto ou habilitado a ocorrência de uma falha, e a falha ocorre quando um serviço não se comporta conforme sua especificação.

2.2.2 Ameaça

As ameaças contra redes de computadores podem ser melhor caracterizadas a partir do entendimento da dinâmica das funções de envio e recepção de informações. Assim, dados dois sistemas A e B onde o primeiro deseja acessar os serviços oferecidos pelo segundo, observamos quatro grandes ameaças, a saber: identificação, interceptação, modificação e fabricação.

A interrupção objetiva impedir que o sistema A de alguma forma acesse os serviços

oferecidos pelo sistema B, observando os fluxos de solicitação de informação oriundos do sistema A não alcançam o sistema B, porém A tem conhecimento desta inalcançabilidade.

A interceptação objetiva que todas as solicitações provenientes do sistema A alcançem o sistema B, porém, sem o conhecimento de A, os fluxos de informação são também conhecidos por um novo sistema chamado C. Como exemplo podemos citar a atuação de um programa especial chamado *sniffer* atuando em uma rede ethernet.

A modificação atinge o sistema A que, de alguma forma está comprometido econsequentemente todos os fluxos são interceptados por um sistema C eencaminhados posteriormente ao sistema desejado. Como exemplo podemos citar aação de um vírus ou cavalo de tróia.

A falsificação atinge o sistema A, que de alguma forma está comprometido, produzindo a construção de fluxos com falsos objetivos para um sistema B. Como exemplo podemos citar o envio de emails, não autorizados.

2.2.3 Ataque

O Trabalho de Shirey[2] detalhado na RFC2828 define o ataque como sendo um evento de segurança em um sistema que basicamente deriva de uma ameaça inteligente ou uma tentativa deliberadamente articulada, elaborada por um senso técnico, método ou técnica com intuito claro de quebrar a segurança de um sistema ou serviço violando assim as políticas de segurança deste sistema.

Os ataques também podem ser categorizados pela maneira de sua condução, sendo assim classificados em ativos ou passivos. O primeiro busca alterar um determinado recurso e assim afetar sua operação; no segundo não é verificada nenhuma alteração nos recursos de um sistema, mas este é utilizado para obter informações sobre o funcionamento de um determinado serviço ou facilidade funcional.

Adicionalmente também devemos considerar a localização e a origem das atividades de ataque. Estas podem partir do interior da fronteira, ou seja, na rede interna de serviços de um determinado ambiente, ou partir do exterior da fronteira, por exemplo da Internet

destinada à rede de serviços.

2.2.4 Vulnerabilidade

Segundo Shirey[2] a vulnerabilidade é definida como sendo uma falha ou fragilidade presente na implementação ou operação de um sistema que pode ser explorada com o objetivo de violar uma política de segurança. Porém, nem sempre uma ameaça resulta em um ataque e não necessariamente um ataque é sempre bem sucedido, isto porque o sucesso é dependente da vulnerabilidade que um ataque explora e aprofunda. Isto se deve a efetividade das contramedidas de um determinado ambiente, assim podemos utilizar falha como sinônimo para vulnerabilidade.

2.2.5 Intrusão

A RFC2828[2] define intrusão como sendo um ou múltiplos eventos que articulados constituem um incidente de segurança no qual um intruso obtém acesso a um sistema ou recurso, sem que o mesmo tenha autorização para tal.

Logo, o termo intrusão é usado como sinônimo para definir sistemas que tenham sido comprometidos independentemente da localização e origem do ataque, importando apenas o resultado da ação de exploração de uma vulnerabilidade.

2.2.6 Incidente

Segundo a RFC2828, o incidente é definido como sendo um evento que envolve uma violação de segurança, em outras palavras, um evento relevante e crítico que compromete de maneira definitiva as políticas de segurança de uma determinado ambiente.

Infelizmente o conceito apresentado anteriormente não totaliza a complexidade dos processos envolvidos; sendo assim necessárias abordagens complementares que posicionem melhor a questão dos ataques que não obtiveram êxito em sua evolução ao longo do tempo. Assim, uma abordagem mais complexa seria conferir ao conceito de inci-

dente o caráter de evento adverso presente em um sistema de informação. Este pode ser um computador ou elemento de rede onde exista a ocorrência de um evento de segurança; caracterizando o incidente como um grupo de ataques que podem ser identificados a partir de outros ataques diferenciando os atacantes, tipos de ataques, objetivo e sua ocorrência no tempo.

2.2.7 Intruso

Os intrusos são classificados segundo Anderson[1], em dois grandes grupos, a saber: agentes externos que não tem qualquer autorização para acessar um determinado sistema e os agentes internos que possuem autorização de acesso e buscam de algum modo obter privilégios adicionais para que então possam realizar uma ação não autorizada.

2.3 Alarmes

Atualmente é comum que as infra-estruturas apresentem um elevado índice de gerenciamento. Esta atividade basicamente monitora de maneira contínua um conjunto de indicadores que uma vez ultrapassados os limites pré-estabelecidos, é enviado às plataformas de controle sinais ou alarmes que iniciam processos e ações corretivas.

De maneira similar, o monitoramento de segurança segue a mesma estrutura funcional, impedindo o aprofundamento dos eventos de segurança. Assim, cabe aos especialistas elaborarem regras que estabeleçam estes limites operacionais.

Os alarmes indicam cenários que serão examinados em profundidade pelos sistemas responsáveis por apontar a ameaça, classificando-as como ataque ou não ataque. Assim, um importante parâmetro da eficiência de um sistema de monitoramento de segurança é a análise do comportamento de seus acessos a partir do grau de certeza do julgamento de um determinado cenário, através de indicativos que são: falsos positivos, falsos negativos, verdadeiramente positivos ou verdadeiramente negativos.

2.3.1 Falso Positivo

Por falso positivo podemos compreender o indicativo equivocado de ataque para um cenário onde não exista a presença de um ataque.

2.3.2 Falso Negativo

Classificamos por falso negativo o indicativo equivocado da ausência de ataque para um cenário onde exista a presença de um ataque.

2.3.3 Verdadeiramente Positivo

Por verdadeiramente positivo compreendemos o indicativo correto de ataque para um cenário onde existe a presença de um ataque.

2.3.4 Verdadeiramente Negativo

Por verdadeiramente negativo compreendemos o indicativo correto da ausência de ataque para um cenário onde não existe a presença de um ataque.

2.4 Modelo Geral dos IDS

Segundo Dorothy[12], os sistemas detectores da intrusão são sistemas capazes de detectar as tentativas de penetração de um determinado sistema, possibilitando assim que seus administradores possam elaborar as respectivas contra-medidas. Porém este modelo é baseado na hipótese que toda a violação de segurança pode ser detectada pelos sistemas de monitoração de segurança e auditoria, possibilitando assim reconhecer uma atividade considerada hostil.

Ressaltamos que é suposto que o sistema de detecção deva ser independente de qualquer sistema operacional, não privilegiando nenhuma aplicação e devendo possuir meca-

nismos que identifiquem as diversas vulnerabilidades e seus respectivos tipos de ataque, oferecendo uma estrutura capaz de abrigar qualquer modelo de decisão.

2.4.1 Modelos Baseados em Servidores

A detecção da intrusão baseada em servidores teve seu início na década de 80, de maneira independente das redes de computadores que atualmente apresentam um ambiente complexo e altamente interconectado. Neste ambiente mais simples era prática comum analisar os registros de atividades consideradas suspeitas presentes nos diferentes arquivos de *logs*. Esta prática se justificava uma vez que os intrusos eram raros, através da análise do pós-fato comprovou-se como estratégia adequada para o entendimento das diferentes dinâmicas na qual estavam inseridos os sistemas computacionais deste período.

Os detectores da intrusão baseados em servidores continuam analisando os diferentes arquivos de registros para o exame da intrusão. Atualmente estes detectores são muito mais automatizados e complexos, pois evoluiram para acomodar técnicas mais sofisticadas privilegiando arquiteturas distribuídas baseadas em agentes específicos para análise contínua e não mais de um único servidor e considerando toda uma infra-estrutura. Num contexto da intrusão mais amplo, estes sistemas utilizam sofisticadas técnicas de agregação baseadas no entendimento global das ameaças.

Um método bastante elementar para detectar a intrusão é centrar a análise de uma ocorrência que está registrada em uma base de dados e observa-la no tempo compreendendo assim a sua frequência, considerando não apenas um único servidor mas todos os membros da infra-estrutura. Caso um sistema esteja sob ameaça é enviado um alarme aos administradores da infra-estrutura que procuram mapear as razões das mudanças inesperadas observadas pelas respostas automáticas do sistema.

Como vantagems desta abordagem, podemos citar:

- Verificação do sucesso ou a falha de um ataque, fornecendo uma averiguação da extensão de uma estratégia e sempre que possível fornecendo um aviso prévio do cenário da ameaça.

- Monitoramento das atividades específicas do sistema. Objetivando acompanhar as atividades de todos os usuários. Contudo limitado as atividades presentes no sistema.

- Detecção de ataques que não são identificáveis pelos sistemas baseados em rede, sendo assim recomendado o seu uso de maneira complementar aos sistemas detectores centrados na investigação dos tráfegos de rede.

- O tempo de resposta para uma ameaça é relativamente próximo ao do tempo real de sua manifestação. Em um ambiente distribuído a difusão de novas bases de conhecimento são rapidamente atualizadas, permitindo um acompanhamento mais sistematizado e eficaz das possíveis ameaças.

- Não exigem equipamentos adicionais, pois os detectores residem nos sistemas onde se encontram os diferentes serviços e facilidades.

- Baixo custo de implementação, distribuição e manutenção, sendo considerado um componente fundamental para compor um projeto de segurança dentro de uma infra-estrutura complexa.

Como desvantagens, desta abordagem podemos citar:

- As potencialidades do detector tornam-se comprometidas caso o mesmo seja invadido por um atacante.

- Os detectores da intrusão baseados em servidores são aplicações muito específicas.

- Os tempos de fornecimento dos arquivos que descrevem as assinaturas são de produção lenta, pois as ameaças e seus desdobramentos devem ser conhecidas em um número grande de implementações.

- As tecnologias de agregação ainda estão em desenvolvimento e nem sempre é completamente garantido que a ação coordenada de um atacante em uma infra-estrutra possibilite aos administradores uma visão correta do cenário de ataque.

- Os sistemas detectores apresentam dificultades para a construção de cenários de ataque quando o mesmo se encontra sob ataques de negação de serviço, o que prejudica a identificação das estrategégias de um ou mais atacantes.

2.4.2 Modelos Baseados em Rede

Segundo Biswanath[31], em essência a detecção da intrusão baseada em rede, está centrada em torno da capacidade de reconhecimento de padrões presentes no sistema de regras. Como conseqüência, este modelo é capaz de obter níveis elevados da exatidão em identificar os intrusos mais sutis, pois a identificação ocorre por níveis de similaridade.

Este modelo exige que todos os ataques no qual o sistema objetiva detectar estejam definidos em sua base de conhecimento, sendo assim impossível a identificação de novos ataques que muito embora estejam presentes no tráfego de rede não foram corretamente identificado pelos especialistas e assim não estão presentes em sua base de conhecimento. Logo a eficiência deste detector é diretamente associada ao grau de atualização do banco de conhecimento.

Como vantagens podemos citar:

- Rastreamento de padrões baseados em expressões que descrevem a ameaça.
- Difícil localização da solução dentro de uma infra-estrutura.
- Implementação não afeta os diversos aspectos funcionais de uma infra-estrutura.
- Funcionamento independente da origem e destino dos tráfegos, sendo capaz de capturar todos os pacotes independentemente do procotolo utilizado.

Como desvantagens, podemos citar:

- Dificuldade de crescimento e para alguns casos necessidade de reprojetar a arquitetura da infra-estrutura.
- Funcionalmente dependentes da descrição detalhada do comportamento dos ataques.

- Não capturam as ameaças recentes que ainda não foram mapeadas pelo formalismo de descrição.
- Necessita de atualizações constantes.

2.4.3 Modelos Baseados em Anomalia

Segundo Debar[24] a detecção centrada em rede está se tornando obsoleta, uma vez que sua base de conhecimento não apresenta capacidade de inferência suficiente para suportar a detecção de ameaças cada vez mais polimórficas[25], o novo desafio é reconhecer a intrusão por meio das diversas relações complexas presentes entre os diferentes componentes da rede que de maneira direta ou indireta fazem parte do fluxo de dados.

Conforme observado por Cannady[26], o modelo baseado em anomalia supõe que o intruso possa ser detectado observando um desvio do comportamento normal ou previsto de um sistema, logo este modelo de detecção considera como intrusão qualquer evento que não corresponde a um comportamento previamente instruído.

Como vantagens, podemos citar:

- Possibilidade de detecção de ameaças não conhecidas.
- Imunidade a ataques antigos ou já conhecidos.
- Abstração do modelo específico de mapeamento de características de ataque.

Como desvantagens, podemos citar:

- Geração de um número consideravelmente alto de falsos positivos.
- Exigência de uma grande quantidade de sessões de treinamento para melhorar a eficiência da captura.
- A partir do conhecimento do perfil usado no treinamento invasores habilidosos poderão invadir os perímetros da rede.

2.5 Taxonomia e Classificação dos IDS

O termo taxonomia é definido como "um esquema de classificação que divida um conhecimento em partes e defina o relacionamento entre estas" e classificação é definida como "a separação e organização dos diferentes objetos em classes".

A classificação pode ser executada de duas maneiras: "a priori" ou "posteriori". A classificação "a priori" é criada de maneira não empírica, assim ela está baseada unicamente na teoria. As classificações "a posteriori" são criadas de maneira empírica e baseadas em dados previamente coletados.

Como a classificação é um subconjunto da taxonomia, podemos entender que ataxonomia inclue uma descrição dos procedimentos que devem ser seguidos paracriar a classificação e para atribuir objetos as diferentes classes.

São definidas por Howard[33] seis categorias para classificação da taxomia, são elas:

1. **Mutualmente exclusivas:** cada objeto pode ser incluído em apenas uma única categoria, não havendo assim superposição.
2. **Exaustivas:** todas as categorias feitas são examinadas em conjunto e cobrem todas as possibilidades.
3. **Não ambíguas:** A classificação deve ser clara e precisa.
4. **Repetitivel:** A classificação resulta sempre no mesmo resultado.
5. **Aceitável:** A classificação é lógica e intuitiva e pode ser percebida de maneira genérica.
6. **Útil:** A classificação auxilia a obtenção de ententendimento a respeito do campo do conhecimento no qual é descrito a referida taxonomia.

A taxonomia utilizada é apresentada na Figura 2.6, sendo esta baseada em uma classificação a priori e subdividida em quatro dimensões conforme definido por Debar[34]. O primeiro componente descreve o comportamento do detector, e está dividida em três subcategorias detalhadas abaixo:

- **Ativas:** O comportamento ativo descreve um componente que pode manipular os dados deixando passar ou não um referido pacote.

- **Passivo:** O comportamento passivo descreve um componente que pode analisar os dados e executar um alerta, mas não manipula os dados nem altera qualquer estado.

- **Reativo:** O comportamento reativo descreve um componente capaz de analisar os diferentes estados presentes na sessão de comunicação e, caso seja detectada uma possível ameaça, alterar o estado desta sessão tornando-o inofensivo, podendo este componente alterar ou não a configuração de um determinado componente de rede.

A segunda dimensão da taxonomia descreve a localização da coleta e está dividida em três subcategorias detalhadas abaixo:

- **Perímetro:** Definido como limite, descrevendo todo o limite da rede de maneira independente, se é um limite externo ou interno .

- **Rede:** Toda a conexão de rede que um componente possa capturar, não importando se a rede for interna ou externa.

- **Servidor:** Por servidor devemos entender como sendo qualquer máquina que ofereça um serviço e para tanto esteja conectada a algum componente de rede, tais como roteadores, switches ou hubs.

A terceira dimensão da taxonomia descreve por qual método de pesquisa o detector irá realizar a detecção. Para tanto este está subdividido em duas subcategorias descritas abaixo:

- **Baseados em Conhecimento:** São modelos de detecção baseados na capacidade de reconhecimento de padrões presentes em um sistema de regras.

- **Baseados em Comportamento:** São modelos de detecção baseados na capacidade de identificar anomalias providos por algoritmos específicos baseados no entendimento das dinâmicas presentes nos tráfegos considerados normais.

A quarta dimensão define a periodicidade da coleta de informações, podendo ser periódica, quando a captura ocorre em períodos específicos definidos pelos administradores de um infra-estrutura ou contínua, ou seja os modelos anteriormente apresentados capturam os dados de ininterrupta.

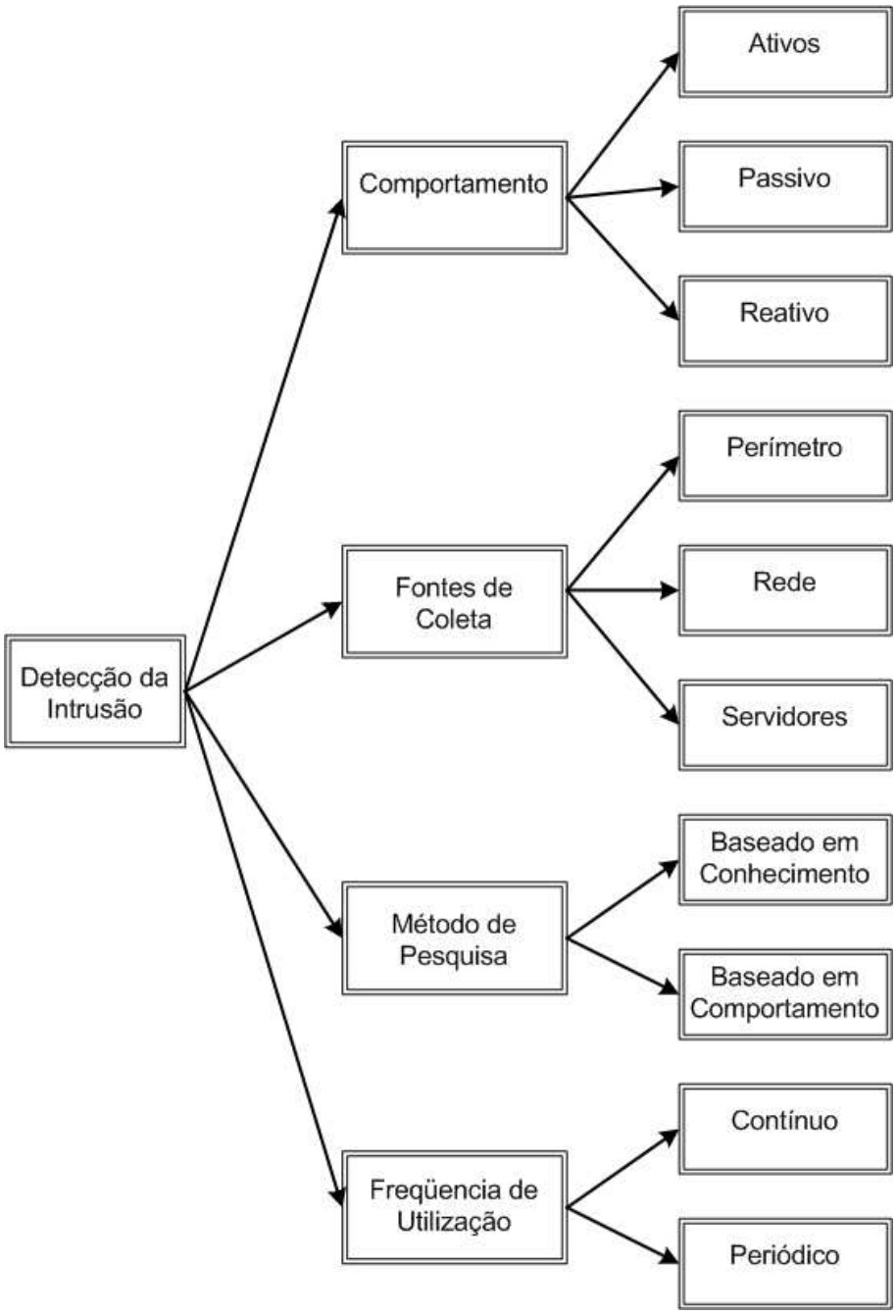

Figura 2.6: Classificação dos Sistemas Detectores da Intrusão.

2.6 Comportamento Geral dos Tráfegos

Considerando os comportamentos das entidades constituintes de uma rede e suas interelações, o comportamento das aplicações é construído por um sistema dinâmico de difícil compreensão. A Internet apresenta um conjunto complexo de padrões auto-similares[6], assim para que seja possível compreender o comportamento básico destes ambientes, torna-se necessário obter uma quantidade significativa de dados que descrevam o comportamento individual de cada componente e sua interação com os demais sistemas.

Basicamente são duas as abordagens para a construção de um sistema de inferência baseado no comportamento geral da rede. A primeira está centrada na utilização de "probes" em diferentes pontos da rede. Assim, segundo Coastes[7], este uso possibilita uma visão tomográfica do sistema e está limitada a caracterização de comportamentos não cooperativos de redes, ou seja, produz uma visão estruturada de redes que não estão sob o controle de uma autoridade central.

A segunda abordagem é a utilização de sistemas de coleta centralizados onde diversos dispositivos estão localizados em diferentes pontos, porém com agregação centralizada. Ela está restrita somente a redes onde existe uma autoridade central constituída, pois somente sob esta diretriz é possível a elaboração de regras de fusão de dados eficientes e claras o suficiente para descrever os comportamentos a serem investigados pelos sistemas de decisão.

2.6.1 Natureza dos dados coletados para análise

Basicamente podemos adotar duas grandes abordagens, a primeira está centrada na dinâmica da aplicação e a segunda no dispositivo de rede. Os dados da aplicação são formados for fluxos que são definidos como sendo uma série unidirecional de pacotes que trafegam entre dois sistemas durante um período de tempo[8]. Estes dados podem ser analisados em diferentes níveis de granularidade, porém minimamente são considerados o conjunto de informações presentes nos protocolos TCP e UDP que em última análise explicita a aplicação.

As informações fornecidas pelos dispositivos de rede apresentam os dados sobre diversos parâmetros funcionais da infra-estrutura, tais como: estado das portas físicas, disponibilidade da CPU, estado das filas, etc.

2.6.2 Definição de Anomalia de Rede

Intuitivamente consideramos tráfegos anormais como sendo aqueles que se distanciam dos parâmetros classificados previamente como normais. Contudo, identificar um tráfego normal e utilizá-lo como referência é sem dúvida uma tarefa de elevada complexidade e talvez impossível, necessitamos acomodar neste modelo todas as dinâmicas não apenas de um único sistema mas de diversos sistemas complexos.

Assim segundo Thottan[5] uma definição mais factível é compreender que a anomalia é caracterizada pela correlação da mudança de um ou mais parâmetros do sistema observado durante o evento anômalo.

2.6.3 Contribuição das Falhas Pontuais à Geração de Anomalias

Sendo uma rede um sistema complexo, são observadas diferentes dinâmicas entre os diversos elementos de rede e uma pequena falha que, muito embora não comprometa o funcionamento geral da infra-estrutura, insere pequenas anomalias que podem contribuir para a detecção de falhas generalizadas e assim pertubar todo o sistema.

2.6.4 Detecção do Ataque

Atualmente, dados os diferentes ataques praticados na Internet, concluímos que existem diversas possibilidades para a construção de um programa explorador. Isto se deve ao fato de existirem diferentes caminhos para explorar as características de uma vulnerabilidade. Portanto, o cenário onde se desenrola o aprofundamento das vulnerabilidades é altamente polimórfico, conforme descrito no trabalho de Zanero[9].

Segundo Estan[10] a rapidez da mudança torna a grande rede (Internet) um alvo cons-

tante. Para tanto basta observar a cada ano o surgimento de diferentes tráfegos de rede tais como: streaming media, CDNs, peer-to-peer, voz sobre IP, etc. A estes tráfegos são associados novos perfis que em última análise representam diferentes modos de uso que combinados a novas tecnologias de comunicação estabelecem um ambiente extremamente hostíl.

A vitalidade da Internet está na capacidade de acomodar as diversidades, porém, como conseqüência, isto obriga aos administradores o estudo dos perfis presentes em suas redes buscando entender os padrões individuais de cada novo tráfego. Assim na prática, os profissionais se utilizam de perfis de uso para construir as diferentes identidades de cada aplicação.

Assim, como conseqüência destas rápidas mudanças, observamos como padrão três formas de detecção de um ataque: o primeiro explora falhas de software no serviço oferecido pela vítima e oculta-se como um tráfego normal, o segundo está na esfera da qualidade de um programa explorador, pois o mesmo pode gerar anomalias caso seja elaborado com pouco cuidado e por fim a inserção de tráfegos que objetivam confundir os diferentes elementos de controle presentes na infra-estrutura.

2.7 Técnicas de Anti-Intrusão

Atualmente estão disponíveis um número significativo de técnicas com objetivo de proteger o perímetro de uma infra-estrutura dotando-a de diversos mecanismos de defesa. Contuto, baseado nos estudos de Lawrence [11] podemos reunirtodos estes métodos em seis grandes grupos, a saber: antecipação, prevenção, impedimentos, detecção e contramedidas que denominaremos de técnicas de anti-intrusão. A Figura-2.7 apresenta um esquemático da disposição destes grupos.

2.7.1 Antecipação

O espaço definido dentro do modelo geral anteriormente apresentado tem a função de, a partir de conhecimentos prévios das diferentes naturezas dos ataques, impedir que

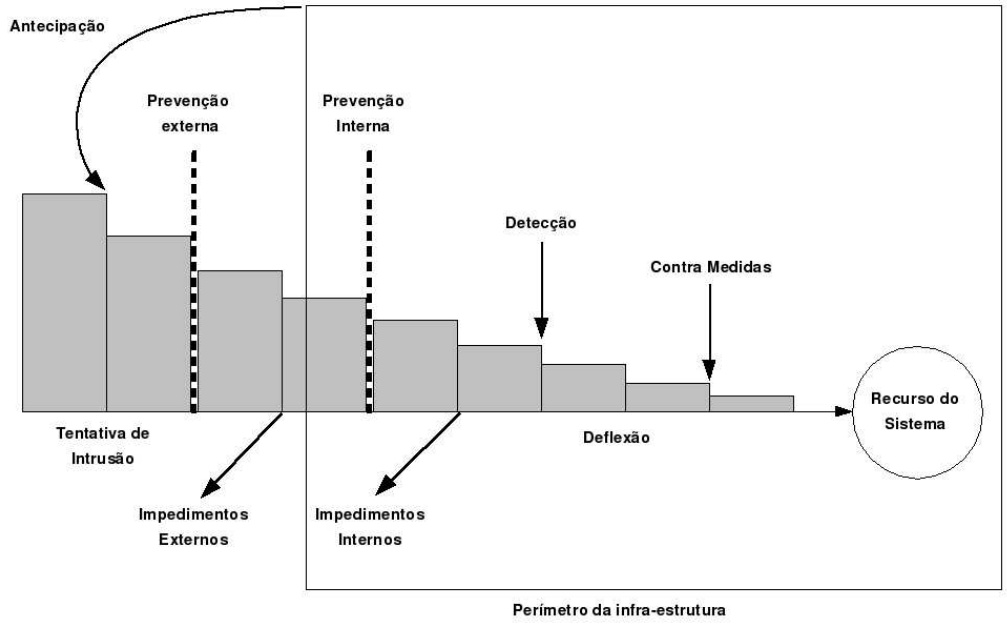

Figura 2.7: Técnicas Anti-Intrusão.

um determinado fluxo alcance um determinado recurso sistêmico que está localizado no interior da infra-estutura.

2.7.2 Prevenção

A prevenção tem a função de avaliar, a partir do conhecimento prévio dos serviços oferecidos pela infra-estrutura, os fluxos considerados normais dentro dos perfis anteriormente construídos para a infra-estrutura, alarmando sempre que houver qualquer variação deste perfil.

2.7.3 Impedimentos

O impedimento tem a função de coibir que um determinado fluxo que contenha uma ameaça já conhecida alcance um determinado recurso presente no interior da infra-estrutura. Este impedimento não depende de conhecimento prévio da dinâmica de um ataque, mas está em conformidade com a política de segurança pré-estabelecida.

2.7.4 Detecção

A detecção tem a função de identificar de maneira clara o intruso e suas intenções ao acessar de maneira duvidosa um determinado recurso localizado no interior da infra-estrutura, e assim possibilitar de maneira concreta as contra-medidas.

2.7.5 Contra-Medidas

Podemos definir como contra-medidas o conjunto de ações realizadas para impedir que os fluxos anteriormente detectados possam alcançar os recursos presentes na infra-estrutura.

Capítulo 3

Apresentação dos Tráfegos de Referência

3.1 Introdução

O laboratório Lincoln do MIT em conjunto com o laboratório de pesquisa da força aérea americana sob a supervisão do DARPA ITO (Defense Advanced Research Projects Agency), realizou em 1998 e 1999 uma avaliação off-line em uma infra-estrutura onde foram gerados diferentes tráfegos anômalos com o objetivo de apoiar e desenvolver as técnicas de detecção da intrusão.

Estas avaliações desenvolvidas por Lippmann[30] contribuiram de maneira significativa para o desenvolvimento da pesquisa de deteção da intrusão, pois forneceram resultados e dados de teste capazes de orientar diferentes esforços de pesquisa e calibração dos atuais modelos, permitindo o desenvolvimento de novas abordagens.

Esta avaliação foi projetada com o objetivo de ser simples, centrando os esforços em prover informações relevantes à área de detecção. Para tanto, utiliza tráfegos presentes na maioria das infra-estruturas, permitindo assim a participação de diferentes comunidades de pesquisa que tenham o interesse no desenvolvimento da área de segurança e privacidade.

3.2 Objetivo Técnico

Os tráfegos providos por este estudo têm a finalidade de medir a habilidade dos sistemas de detecção da intrusão e de ataques destinados a sistemas computatorizados conectados a redes. Em 1998, este estudo estava orientado apenas a estações baseadas no sistema UNIX e centradas na determinação dos seguintes eventos de ataque presente nas diversas sessões de tráfego: negação de serviço, acesso desautorizado de uma máquina remota, acesso desautorizado aos privilégios locais do super usuário por um usuário local não privilegiado, vigilância, Probe e comportamento anômalo do usuário.

As diversas sessões TCP/IP completas, que continham anomalias foram marcadas, destacando-as das sessões consideradas normais. Tais sessões são referentes ao acesso a serviços comumente observados na maioria das infra-estruturas. São eles: Telnet, HTTP, SMTP, FTP, Finger, Rlogin e outros. Estes comportamentos são baseados nos estudos observados anteriormente em bases militares da força aérea americana.

Assim, os tráfegos descritos neste estudo são projetados com o obejetivo de promover o progresso da pesquisa de detecção com os seguintes objetivos: surgimento de novas abordagens na área de deteção da intrusão, avanço da tecnologia de detecção, melhoria na performance e desempenho desta tecnologia e provimento referencial das diferentes tecnologias a partir de um mesmo ponto de comparação.

3.3 Dados de Treinamento

São providos neste estudo dois conjuntos de dados: treinamento everificação. O primeiro reune tráfegos anômalos com diferentes distribuições deataque destinados a diferentes máquinas-alvo. Associadamente estão presentestráfegos considerados normais, com a finalidade de auxiliar os pesquisadores nodesenvolvimento de seus modelos e a partir deste conhecimento aplicá-los aoconjunto de apresentações onde poderão ser avaliados diferentes fatores como:performance, qualidade da detecção, efetividade, etc.

Os tráfegos considerados normais são dados sintetizados a partir de estudos anteri-

ores que proveram o conhecimento das diferentes dinâmicas de tráfego, associados aos serviços presentes neste estudo. Os tráfegos considerados anômalos são baseados em ameaças observadas pelas principais agências de controle e alerta de anomalias, onde foram realizados pequenos ajustes e adequações.

Como suporte ao desenvolvimento da tarefa de construção de novos modelos de detecção, são oferecidos cinco arquivos: o primeiro apresenta todos os tráfegos no formato padrão *tcpdump*, o segundo chamado de *listfile*, descreve todas as sessões marcando com o nome do ataque a sessão submetida à infra-estrutura; o terceiro apresenta o arquivo padrão de auditoria disponível no sistema Solaris chamado *BSM*; o quarto apresenta a distribuição dos processos, descrevendo o seu estado minuto a minuto e; por fim, o arquivo *dump* do sistema de arquivos.

3.4 Dados de Verificação

Os dados de verificação são gerados de maneira similar aos dados do treinamento. Os formatos dos vários elementos de dados são idênticos aos dados de treinamento, com a exceção da contagem de ataques que no arquivo listfiles estarão vazios. Entretanto, uma chave resposta será distribuída junto com o desenvolvimento dos dados de verificação que descrevem a identificação da sessão, podendo esta conter 0 para o normal e 1 para o ataque, bem como o nome da referida ameaça.

Assim os dados de verificação tem a função de validar o processo de compreensão dos comportamentos anômalos presentes nos arquivos de treinamento.

3.5 Diferenças entre os testes de 1998 e 1999

Ocorreram diversas mudanças no projeto de avaliação entre 1998 e 1999. Tais alterações são resumidas na Tabela 3.1 onde são destacadas as alterações mais significativas para cada componente da avaliação.

Em 1998, os testes de avaliação enfatizavam tráfegos específicos com o objetivo de

Tabela 3.1: Diferenças entre os Testes de 1998 e 1999

Escopo	**Ano de 1998**	**Ano de 1999**
Tráfego de Fundo	Tráfego de Fundo Unix	Tráfego de Fundo NT Duas semanas de treinamento sem ataque Análise de Falsos Alarmes
Ataques	Ataques Externos 38 Tipos de Ataques Ataques Stealthy limitados Ataques contra sistemas Unix	Ataques Internos >50 Tipos de Ataques Ataque à informação Ataques contra sistemas Unix e NT Ataques Stealthy completo
Métricas	Detecção + Análise ROC (receiver operating curve)	Identificação Análise de Erros

investigar a segurança em ambientes UNIX e o comportamento de sistemasdetectores da intrusão desenvolvidos nesta plataforma. Observando o crescimento da plataforma Windows em diversos departamentos do governo e organizaçõesprivadas, os especialistas incluiram esta distribuição nos testes de 1999. Assim,são direcionados diversos tráfegos específicos do Windows NT como origem ou destino a esta plataforma.

Na plataforma Windows são avaliados os serviços de páginas web, através, o *Microsoft Internet Information Server* (IIS), e seu serviço de Mail, o *Microsoft Exchange Server*.

Muitos serviços foram redefinidos em 1999 com o objetivo de fornecer um maior realismo e uma melhor distribuição do tráfego do fundo, por exemplo, o procedimento utilizado para a geração do serviço *Telnet* foi aperfeiçoada para abrigar diferentes perfis de usuários, simulando diferentes hábitos compatíveis com cada perfil de usuário, que passa agora a executar diferentes atividades em cada sessão.

Adicionalmente foram desenvolvidos sistemas que simulavam o acesso à Internet

baseados no mesmos perfis utilizados para a geração das sessões *Telnet*. Para tanto foram desenvolvidos geradores que simulavam o comportamento dos principais navegadores da época: o *Netscape* e o *Windows Explorer*; possibilitando a presença de tráfegos http a sites artificialmente construídos.

Em resposta a diversas observações realizadas pela comunidade científica os especialistas responsáveis pelos testes incluiram duas semanas dos dados de treinamento que contêm somente o tráfego de fundo, ou seja, sem a presença de nenhuma anomalia, tornando possível o desenvolvimento de sistemas baseados em comportamento, o que em 1998 foi sériamente prejudicado.

Uma importante modificação realizada em 1999 foi a alteração do ambiente deteste que permitiu que todas as máquinas virtuais fossem capazes de iniciar ou receber tráfegos de ataque, bem como a mudança das métricas usadas na avaliação off-line da detecção da intrusão usadas em 1998. Esta métrica foi estendida em 1999 para incluir com maior exatidão a identificação do ataque.

3.6 Detalhamento da Infra-estrutura

A infra-estrutura utilizada para a geração dos tráfegos é apresentada na Figura 3.1, onde basicamente estão presentes dois ambientes: um interno, que simula as atividades de funcionamento de uma base militar americana, e o externo simulando a Internet.

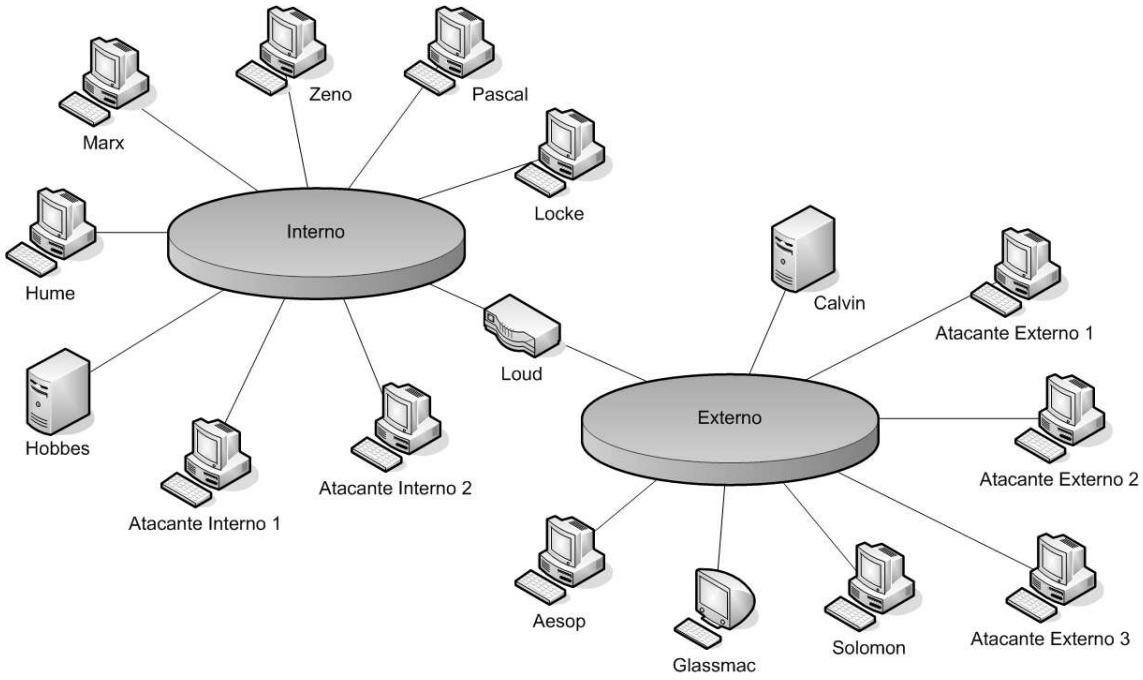

Figura 3.1: Arquitetura utilizada para geração dos tráfegos de referência.

Abaixo segue o detalhamento do ambiente interno da infra-estrutura de referência:

- **Máquina Linux**: A máquina Marx, executando um RedHat 5.0 com a versão de kernel 2.0.30 em um Intel Pentium II, 266 Mhz, tendo como aplicativos principais um apache versão 1.1.3, ssh versão 1.2.25, snmpd e adicionalmente foram oferecidos os serviços Telnet, FTP e Finger.

- **Máquina SunOS**: A máquina Zeno, executando um *SunOS 4.1.4* em uma *Sparc* 2, oferece os serviços sendmail, FTP, Telnet e Finger.

- **Máquina Solaris**: A máquina Pascal, executando um Solaris 2.5 em uma *Ultra One*, oferece os serviços *SSH*, *Telnet*, *FTP* e *Finger* e simula a ação de usuários na atividade de leitura de mail e navegação Internet utilizando o navegador Lynx.

- **Máquina Windows NT**: A máquina Hume, executando um *Windows NT Server* Versão 4.0 Build 1381 com *Service Pack* I em um Intel Pentium III, 850 Mhz, onde são oferecidos os serviços web por meio do *Internet Information Server* Versão 2.0, *FTP*, *Gopher*, *Remote Logon* e *Telnet*.

- **Máquina Geradora Interna de Tráfego**: A máquina Hobbes, executando um *RedHat* 5.0 com a versão de kernel 2.0.32, simula a ação de vinte estações de trabalho executando diferentes atividades baseadas em um modelo de perfil de uso para cada um destes usuários. Adicionalmente esta estação oferecia o serviço de nomes (*bind*) na versão 4.9.6.

- **Máquina Sniffer**: A máquina Locke, executando um Solaris 2.6 em uma Ultra One tinha a função de coletar todo o tráfego interno da rede, utilizando o programa *TcpDump* na Versão 3.3, e armazenando os dados de coleta em seus discos locais.

- **Máquinas Atacantes**: Formada por duas máquinas: uma executando o sistema operacional *Linux* e a outra *Windows NT* que periodicamente realizavam ataques supervisionados às máquinas vítimas localizadas tanto no ambiente interno como externo. Em ambas foram instaladas os programas de ataque escritos em Perl.

- **Roteador**: A máquina Loud, um roteador *Cisco* 2500, executando o IOS Versão 11.3 Revisão 4, era responsável pela integração das redes interna e externa, sendo este equipamento monitorado via *SNMP*.

Abaixo segue o detalhamento do ambiente externo da infra-estrutura de referência:

- **Ambiente Internet**:A máquina Aesop, executando um RedHat 5.0 em um Intel, onde são executados diversos ambiente virtuais o que permite a simulação de um ambiente dinâmico similar a um tráfego Internet. Este ambiente era provido por diversos serviços de páginas com tamanhos diferenciados bem como serviços simples.

- **Gerador Externo de Tráfego**: A máquina Calvin executando um *RedHat* 5.0 com uma versão de kernel 2.0.32 fornece tráfegos simulados provinientes de ambientes

virtuais com o objetivo de simular os diferentes perfis de usuários presentes na Internet.

- **Atacantes Externos**: Este conjunto era formado por três equipamentos sendo dois ambientes *Linux* e um *Windows NT* que geravam tráfegos anômalos tanto para o ambiente externo como interno. Estas máquinas utilizavam um conjunto de programas escritos nas linguagens C e **Perl** com o objetivo de automatizar todo o processo.

- **Monitoração de Tráfegos**: A máquina Glassmac, executando um *Mac* O/S 8.5 em um *Machintosh* 6100/60 era responsável por coletar informações no padrão *SNMP* oriundos do roteador e apresentar estes dados em seu web server.

- **Máquina Sniffer**: A máquina Solomon, executando um *Solaris* 2.6 em uma *Sparc* 2. Era utilizada com o objetivo de capturar os tráfegos presentes no ambiente externo e utilizava o programa *TcpDump* na Versão 3.3 e armazenava os dados de coleta em seus disco locais.

3.7 Categorização dos Ataques

Os ataques presentes nas avaliações de 1998 e 1999 foram distribuídos em categorias com o objetivo de comparar o comportamento da detecção dos diversos sistemas detectores, determinando assim a sua capacidade de identificação de cada espaço de ameaça. Esta abordagem permite aos pesquisadores classificar para qual conjunto de ataques o respectivo modelo está sendo desenvolvido.

Para ambas avaliações de 1998 e 1999, todos os componentes de ataque tiveram objetivos específicos de obtenção de informação, seja sobre um computador ou uma rede, seja de cada programa exploratório que segue a estratégia de sua categoria. Abaixo apresentamos as categorias de ataque utilizadas:

- **Denial of service (DOS)**: São ataques que têm o objetivo de limitar ou de negar o serviço fornecido por um usuário, um computador, ou uma rede. Um exemplo comum é o ataque SYN-Flood, onde o atacante inunda o computador vítima com

mais pedidos de conexão TCP do que este pode processar, fazendo com que o computador seja incapaz de responder aos pedidos válidos de serviço.

- **Probe**: São ataques que têm o objetivo de reconhecer a configuração de um sistema computatorizado ou de uma rede. Um exemplo comum é o ataque de IPSweep onde o atacante varre um ou mais endereços IP em uma dada infra-estrutura, com o objetivo de determinar a distribuição de serviços em um determinado conjunto de servidores.

- **Remote-to-Local (R2L)**: São ataques que têm o objetivo de obter o acesso local a um computador ou a uma rede ao qual o atacante teve previamente somente o acesso remoto autorizado.

- **User-To-Root (U2R)**: São ataques que têm o objetivo de obter o acesso às contas privilegiadas como root ou super-usuário, a partir de um acesso autorizado a um sistema sem estes privilégios.

Abaixo é apresentado a distribuição dos ataques, segundo as diferentes categorias de ataques descritas anteriormente.

Tabela 3.2: Lista de Ataques Presentes nos Tráfegos DARPA

DOS	**R2L**	**U2R**	**Probe**
Apache2	Dictionary	Anypw	Inside Sniffer
Arppoison	FTP Write	Casesen	IPSweep
Back	Guest	Eject	Port Sweep
Crashiis	HTTP Tunnel	Ffbconfig	LS Domain
Dosnuke	Imap	FdFormat	MScan
Land	Named	Load Modules	NTInfo Scan
Mailbomb	NCFtp	NTFSDos	NMap
Neptune	NetBus	Perl	Queso
Ping da Morte	NetCat	PS	Reset Scan
Process Table	PHF	Sechole	Saint
Selfping	PPMacro	Xterm	Satan
Smurf	SendMail	Yaga	
SSHProcess Table	SSH Trojan		
Syslog	XLock		
TCPReset	XSnoop		
Teardrop	Guess Telnet		
UDP Storm	Guess POP		
	Guess FTP		

3.8 Estatística de Tráfego

Apresentamos alguns gráficos que caracterizam os tráfegos de fundo sintetizados nos ensaios do MIT. Como não é possível incluir todos os gráficos que descrevam as diferentes facetas do tráfego, são aqui apresentados os gráficos considerados mais significativos, apresentando assim as características gerais destes tráfegos.

Os gráficos apresentados correspondem à dinâmica da primeira à terceira semana de testes onde não estão presentes os tráfegos anômalos.

A Figura 3.2, apresenta o número médio em Megabytes de tráfego por dia de simula-

ção dos protocolos TCP, UDP e ICMP, evidenciando assim a dinâmica presente na massa de dados. AFigura 3.3 apresenta o número médio de conexõesdiárias a serviços TCP, evidenciando assim a granularidade das conexões dentrodo volume de tráfego anteriormente apresentado.

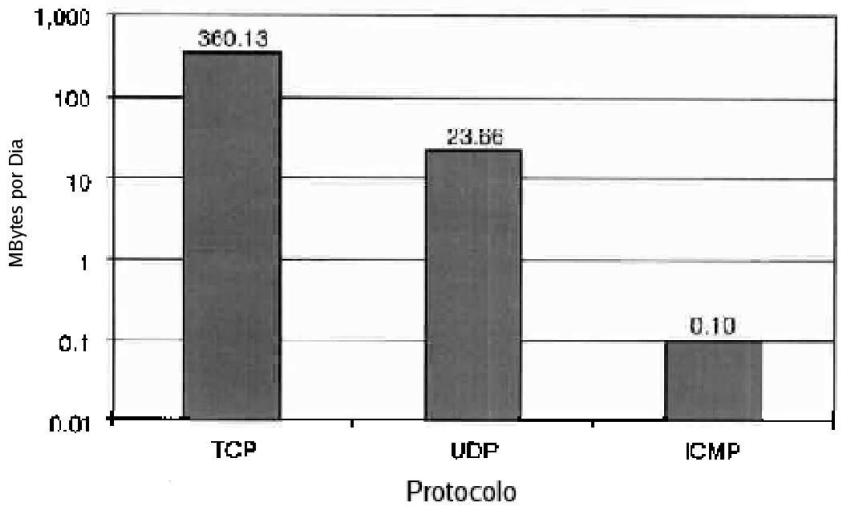

Figura 3.2: Média dos protocolos nas três semanas de treinamento.

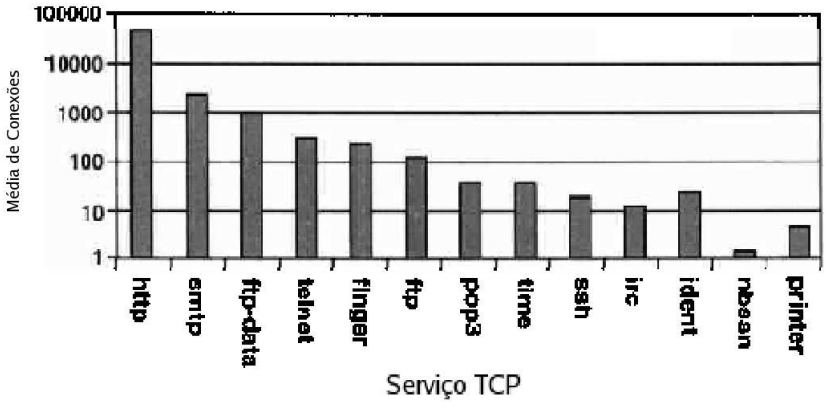

Figura 3.3: Número médio de conexões diárias de serviços TCP.

As Figuras 3.4, 3.5 e 3.6, representam a dinâmica dos tráfegos usados no experimento do MIT.

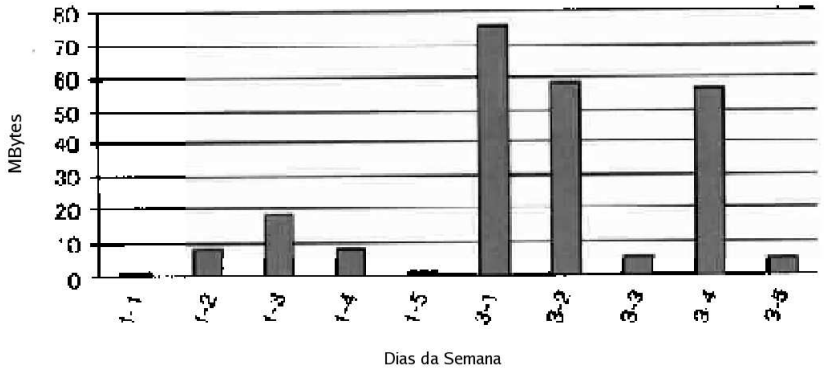

Figura 3.4: Número médio de conexões TCP.

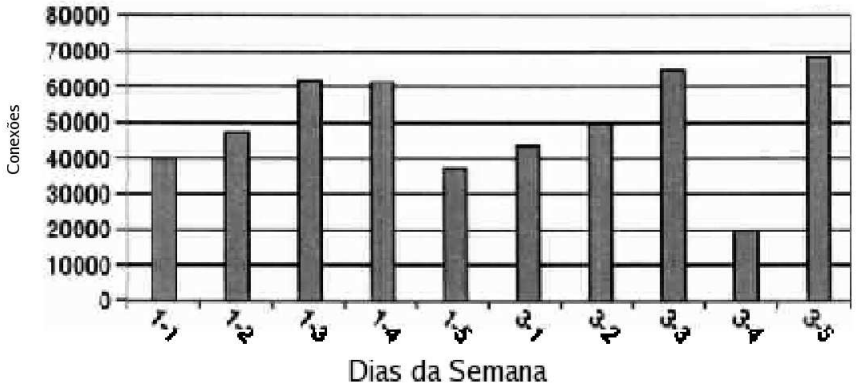

Figura 3.5: Volume médio de conexões TCP.

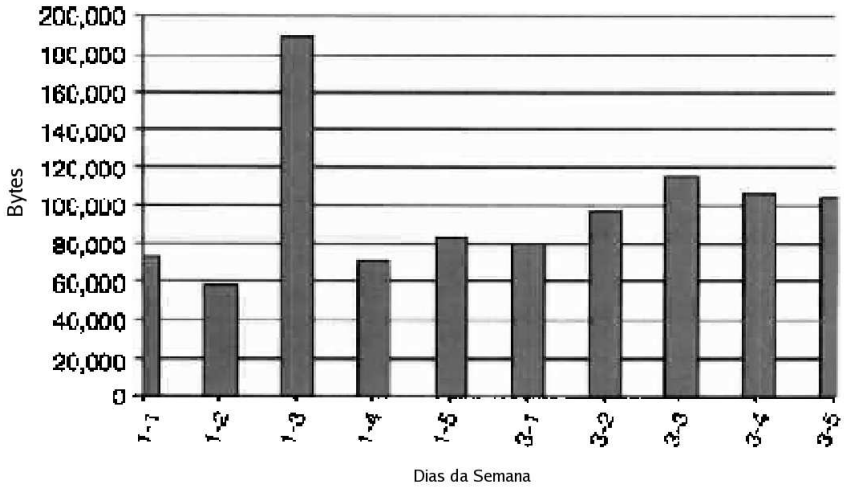

Figura 3.6: Volume médio do Tráfego ICMP.

3.9 Críticas e Considerações

Apesar dos tráfegos DARPA constituirem uma referência para comunidade acadêmica, existem diversas considerações quanto a seu uso. Tais observações foram reunidas por McHugh[27] que elaborou um estudo onde destaca diversos pontos importantes, que contribuem para ampliar o entendimento deste experimento.

Inicialmente é criticada a taxonomia utilizada, que privilegia apenas o ponto de vista de um atacante, introduzindo uma polarização das manifestações detectadas durante os testes. Associadamente é apresentada a dificuldade de comparação dos resultados, pois estes são apresentados utilizando curvas ROC (receiver operating curve) que embora sejam largamente utilizadas em outros campos da engenharia, não apresentam de maneira clara o contexto do experimento restringindo-se apenas às características de um modelo específico. Segundo McHugh, trata-se de uma questão mais ampla no sentido de qual unidade seria a mais adequada à representação dos resultados obtidos.

3.10 Trabalhos Anteriores

Estavam disponíveis na época da realização dos estudos providos pelo DARPA, uma série de trabalhos que possibilitariam a este grupo uma visão mais apurada dos diferentes cenários propostos. Um destes estudos elaborado por Puketza[28] descreve uma metodologia de teste de sistemas detectores da intrusão que reunia um conjunto de procedimentos, bem como a especificação de diferentes ferramentas de software que objetivam gerar tráfegos intrusivos e livres de ameaças de maneira controlada. Assim, esta metodologia possibilitava a criação de múltiplos cenários mais ricos em granularidade, onde tais parâmetros estariam sob o controle do gestor da experiência podendo assim ser facilmente reproduzido por qualquer grupo interessado em recriar o ambiente de teste.

Os parâmetros descritos no trabalho de Puketza, são: o número de fontes geradoras de anomalias e tráfego normal, o volume de tráfegos não intrusivos, o volume de sessões, número de sessões concorrentes, a média de perdas de sessões, o número de sessões simultâneas; nível de distribuição de ataques entre as disversas plataformas e a distribuição de severidade.

Um segundo trabaho disponível neste período foi desenvolvido pela divisão de pesquisa da IBM em Zurich[29]. Este grupo tinha como objetivo projetar e implementar ambientes de teste de intrusão em tempo real para validação de produtos de segurança da IBM, consistindo este ambiente de um número grande de máquinas clientes e servidoras, todas controladas por uma única máquina, utilizada com a finalidade de controlar todo o ambiente de teste. O controlador da experiência utilizava os mesmos parâmetros definidos por Puketza, adicionando um fator totalmente novo: a variabilidade dos testes. Este parâmetro definia a disponibilidade dos servidores e seus respectivos serviços, assim tornava-se possível avaliar os efeitos dos tráfegos intrusivos em ambientes com diferentes disponibilidades.

Através dos trabalhos citados anteriormente, podemos explicitar uma questão central ignorada pelo DARPA: a preocupação com o controle da geração dos tráfegos dentro de um ambiente heterogêneo, observando por meio de uma metodologia clara o comportamento do consumo sistêmico e os efeitos das ameaças, sendo esta tarefa possível de

reprodução por qualquer pesquisador uma vez que os parâmetros de geração são conhecidos.

3.11 Dados de Avaliação

Os dados utilizados para a geração dos tráfegos presentes no experimento DARPA, permitem apenas a identificação de dois componentes principais, são eles: tráfegos normais (livres de ameaças) e intrusivos (com a presença de ameaças). Para tanto é utilizado um parâmetro de identificação que descreve a relação de carga entre estes dois tráfegos, porém são desconhecidos os parâmetros de geração e a disponibilidade dos serviços.

Assim, conhecida apenas a relação entre os tráfegos, torna-se muito vago o acompanhamento operacional do experimento, limitando-se à contagem dos alarmes, desconsiderando o entendimento do cenário geral no qual cada alarme está associado.

3.12 Dados de Ataque

A geração da dinâmica de ataques utilizados no experimento DARPA não considerou o estudo realizado pela força aérea americana que procurou compreender o comportamento geral destes tráfegos maliciosos observado em suas bases, preferindo gerar um perfil próprio baseado em modificações de programas maliciosos disponíveis na época. Este procedimento dificulta ainda mais a possibilidade de reprodução destes tráfegos, uma vez que tais modificações não são mencionadas nos relatórios do experimento.

Os especialistas responsáveis pelo experimento optaram pela geração de seus próprios tráfegos maliciosos, injetando na infra-estrutura de referência diversos tipos de ataques, distribuidos segundo a taxonomia anteriormente apresentada. Estes ataques foram distribuídos em dez semanas de testes, onde a quantidade de ataques é mais ou menos uniforme, o que não representa um comportamento usual nas infra-estruturas.

3.13 Dados de Treinamento e Verificação

Os especialistas responsáveis pelo experimento elaboraram dois conjuntos de testes. O primeiro chamado de treinamento, e o segundo de verificação. O primeiro é obtido a partir da coleta de informações de sete semanas em um dia de vinte e duas horas e em uma semana de cinco dias. Este conjunto contém diferentes tipos de ataques e estão descritos em uma lista que permite a identificação da evidência de ataque, possibilitando assim a identificação de dados com a presença e livres de anomalias. O segundo conjunto é usado para validar o processo de detecção que teoricamente está descrito no primeiro conjunto de dados.

Não existe porém, nenhum estudo provido pelos especialistas responsáveis que garanta a existência de variabilidade suficiente para que o desenvolvimento de um sistema baseado em anomalia ou aprendizado tenha sucesso com a granularidade oferecida.

Além disto, o experimento de 1998 não é apresentado como um conjunto de tráfegos onde não existe nenhuma evidência de anomalia o que dificulta o desenvolvimento de sistemas baseados em comportamento, pois todos os conjuntos contém dados anômalos.

3.14 Características e Limitações do Coletor

Foi utilizado o *Tcpdump* como ferramenta para coletar os tráfegos brutos. Esta escolha foi apropriada por se tratar de um software de domínio público presente em um grande número de plataformas, porém os especialistas não informam os limites operacionais deste coletor.

Esta questão é significativa uma vez que foi utilizado apenas um único coletor em cada segmento e não são detalhados os procedimentos e cuidados com a sua implementação, ou seja, não é informado se a plataforma sofreu algum nível de "customização" que permitisse a esta expandir os seus limites operacionais. Desta forma, não é possível garantir que todos os pacotes tenham sido capturados com sucesso, e consequentemente, todas as sessões construídas representem o cenário completo de carga.

Outra questão relevante se refere ao tempo presente na infra-estrutura. Os especialista responsáveis não descreveram se todas as máquinas envolvidas no teste compartilhavam o mesmo domínio de tempo, introduzindo pequenos erros na agregação dos diferentes arquivos de resultados.

Capítulo 4

Detalhamento dos Ataques Estudados

4.1 Introdução

Neste capítulo os seguintes ataques serão apresentados em detalhes: varredura de portas (*portscan*); netuno (*neptune*) e convidado (*guest*). A justificativa da escolha destes ataques é baseada no fato que cada um destes integram a visão estruturada do aprofundamento dos ataques destinados a uma infra-estrutura, conforme detalhado na seção 2, que agora são combinados às ferramentas utilizados pela maioria dos ataques, conforme apresentado na Figura 4.1.

4.2 Varredura de Portas

O ataque varredura de portas (*portscan*) é classificado como um procedimento que objetiva o mapeamento das fronteiras conforme apresentado na Figura 4.1. Segundo Bailey[35] a varredura de portas é uma das técnicas mais difundidas usadas para descobrir e conhecer os serviços que estão disponíveis em uma infra-estrutura. Utilizando este método, um atacante pode então criar uma lista de fraquezas e de vulnerabilidades potenciais. Esta técnica é categorizada conforme apresentado na seção 3.7 descrevendo-a como *probe*.

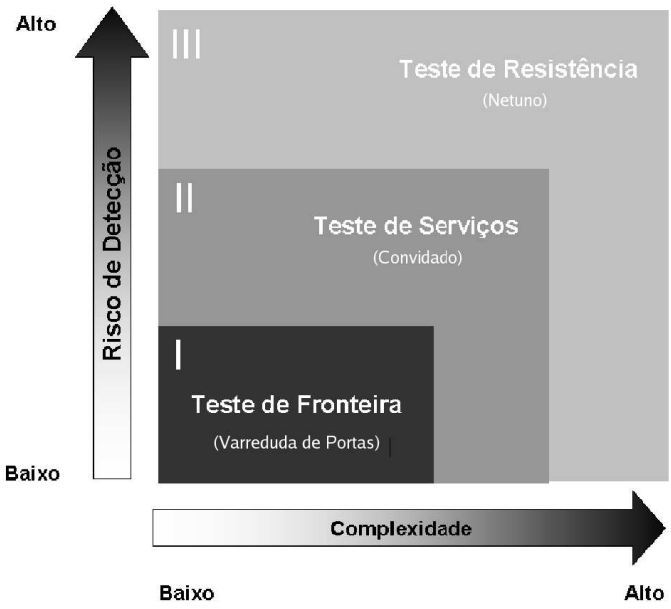

Figura 4.1: Aprofundamento dos Ataques e Ferramentas

Segundo Vinod[36], este ataque é baseado em quatro estratégias de execução que são apresentadas abaixo:

- **Varreduda vertical:** É definida como uma varredura seqüencial ou aleatória de múltiplas portas de serviço de um único endereço IP, observado durante um período de uma hora.
- **Varredura horizontal:** É definida como uma varredura baseada numa única fonte, examinando em diversas máquinas a presença de um serviço específico presente na infra-estrutura.
- **Varreduras coordenadas:** É definida como uma varredura baseada em múltiplas fontes que procuram a presença de um serviço específico presente na infra-estrutura.
- **Varredura Secreta:** É definida como uma varredura que reúne as técnicas de inspeção horizontal e vertical praticada por múltiplas fontes, procurando a presença de serviços em uma infra-estrutura, porém utilizando uma freqüência de busca muito baixa.

As diferentes técnicas presentes na estratégia de varreduras são classificados por De-

thy [37] e esquemáticamente apresentadas na Figura 4.2.

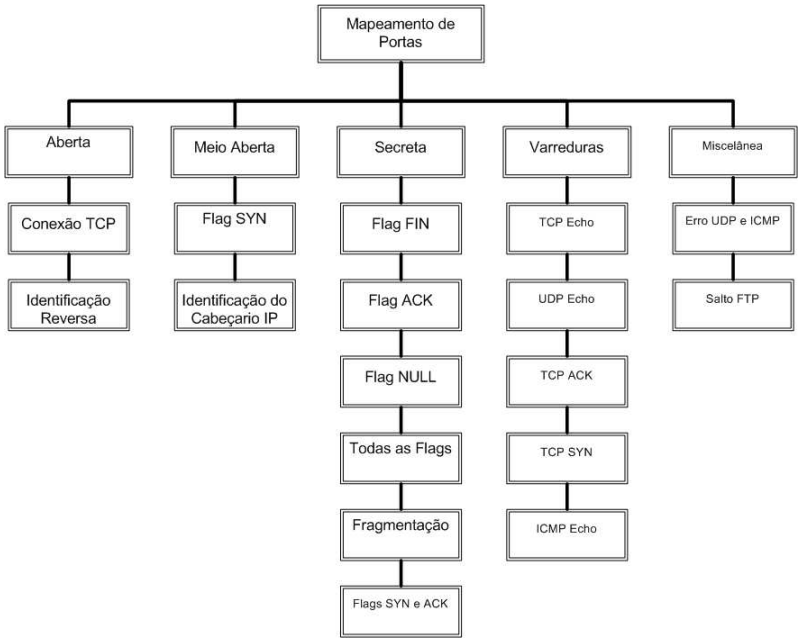

Figura 4.2: Classificação das técnicas usadas em mapeamentos de portas.

Este trabalho é limitado ao detalhamento da classificação apresentada. Tal limitação existe por entendermos que a mesma deve ter representatividade nos dados utilizados como referência. Este texto é um facilitador ao entendimento das diferentes análises a serem apresentadas ao longo desta dissertação, assim abaixo segue a apresentação das técnicas aberta, meio aberta e secreta:

- **Varredura Aberta:** As técnicas de varreduras abertas são extremamente fáceis de detectar e filtrar. Este tipo de varredura envolve abrir uma conexão com um servidor remoto onde são encontrados os fluxos padrões de negociação do TCP/IP (three-way handshake).

- **Varredura Meio Aberta:** O termo meio aberto aplica-se à maneira como o cliente termina a conexão antes que o three-way handshake esteja terminado.

- **Secreta:** O termo foi usado originalmente para descrever uma técnica que evitasse que o sistema detector da intrusão registrasse a identificação do intruso devido ao longo processo de varredura.

4.2.1 Conexão considerada normal

Para apresentar as diferentes questões que evidenciam o processo de varredura de portas utilizando as técnicas de conexão baseadas em connect, flag SYN e flag FIN, será apresentado inicialmente o que é considerado normal em uma conexão TCP. Para tanto utilizaremos um exemplo de acesso a um serviço comum de recuperação de mensagens baseado no protocolo POP (*post office protocol*), apresentado na Figura 4.3.

Nesta conexão são realizadas as atividades de comunicação, trocas de comandos e término dentro do que é esperado em um processo de transação normal tanto do ponto de vista do TCP como do protocolo POP.

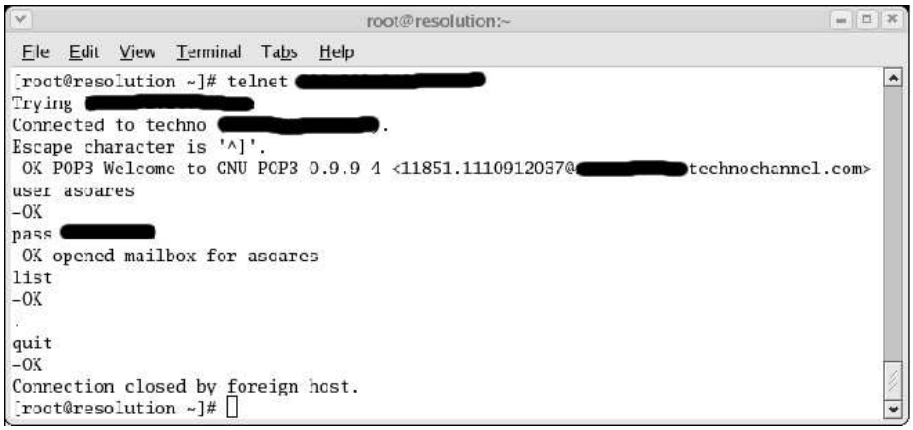

Figura 4.3: Acesso ao serviço POP.

Este acesso foi capturado por meio do programa TCPDump[38] e são apresentados por meio do programa VTA[39], que permite o exame de todos os pacotes identificando a fonte, destino e as diferentes flags utilizadas na conexão bem como a geração dos diagramas de estado TCP.

A leitura dos pacotes, conforme apresentada na Figura 4.4, muitas vezes tornam excessivamente trabalhosos os processos de identificação dos fluxos de dados. Entretanto, é possível apresentar estas informações de maneira mais clara utilizando para tanto o diagrama de fluxo de informações ao longo do tempo, conforme apresentado na Figura 4.5.

Como podemos observar, a troca de informações ocorre dentro do previsto e todas as flags presentes nos pacotes TCP estão assinaladas de maneira adequada permitindo assim

No.	Time	Source	Dest	Protocol	Info
00000001	0	192.168.0.4	200.202.248.27	TCP	32781 > 110 [SYN] Seq=3526238044 Ack=0 Win=16d0
00000002	0.062664	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK SYN] Seq=3826163581 Ack=3526238045 Win=16a0
00000003	0.062698	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK] Seq=3526238045 Ack=3826163582 Win=16d0
00000004	0.128942	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK PSH] Seq=3826163582 Ack=3526238045 Win=16a0
00000005	0.128984	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK] Seq=3526238045 Ack=3826163667 Win=16d0
00000006	4.31518	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK PSH] Seq=3526238045 Ack=3826163667 Win=16d0
00000007	4.38579	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK] Seq=3826163667 Ack=3526238059 Win=16a0
00000008	4.3904	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK PSH] Seq=3826163667 Ack=3526238059 Win=16a0
00000009	4.39042	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK] Seq=3526238059 Ack=3826163672 Win=16d0
00000010	11.5348	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK PSH] Seq=3526238059 Ack=3826163672 Win=16d0
00000011	11.6416	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK] Seq=3826163672 Ack=3526238075 Win=16a0
00000012	11.7226	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK PSH] Seq=3826163672 Ack=3526238075 Win=16a0
00000013	11.7226	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK] Seq=3526238075 Ack=3826163704 Win=16d0
00000014	15.9795	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK PSH] Seq=3526238075 Ack=3826163704 Win=16d0
00000015	16.0855	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK] Seq=3826163704 Ack=3526238081 Win=16a0
00000016	16.1136	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK PSH] Seq=3826163704 Ack=3526238081 Win=16a0
00000017	16.1136	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK] Seq=3526238081 Ack=3826164353 Win=195a
00000018	18.2802	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK PSH] Seq=3526238081 Ack=3826164353 Win=195a
00000019	18.3809	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK] Seq=3826164353 Ack=3526238087 Win=16a0
00000020	18.4868	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK PSH] Seq=3826164353 Ack=3526238087 Win=16a0
00000021	18.4868	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK] Seq=3526238087 Ack=3826164358 Win=195a
00000022	18.4901	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK FIN] Seq=3826164358 Ack=3526238087 Win=16a0
00000023	18.4901	192.168.0.4	200.202.248.27	TCP	32781 > 110 [ACK FIN] Seq=3526238087 Ack=3826164359 Win=195a
00000024	18.5532	200.202.248.27	192.168.0.4	TCP	110 > 32781 [ACK] Seq=3826164359 Ack=3526238088 Win=16a0

Figura 4.4: Lista de pacotes de uma sessão normal.

que toda a máquina de estado TCP seja executada sem observamos nenhuma anomalia, conforme apresentado na Figura 4.6.

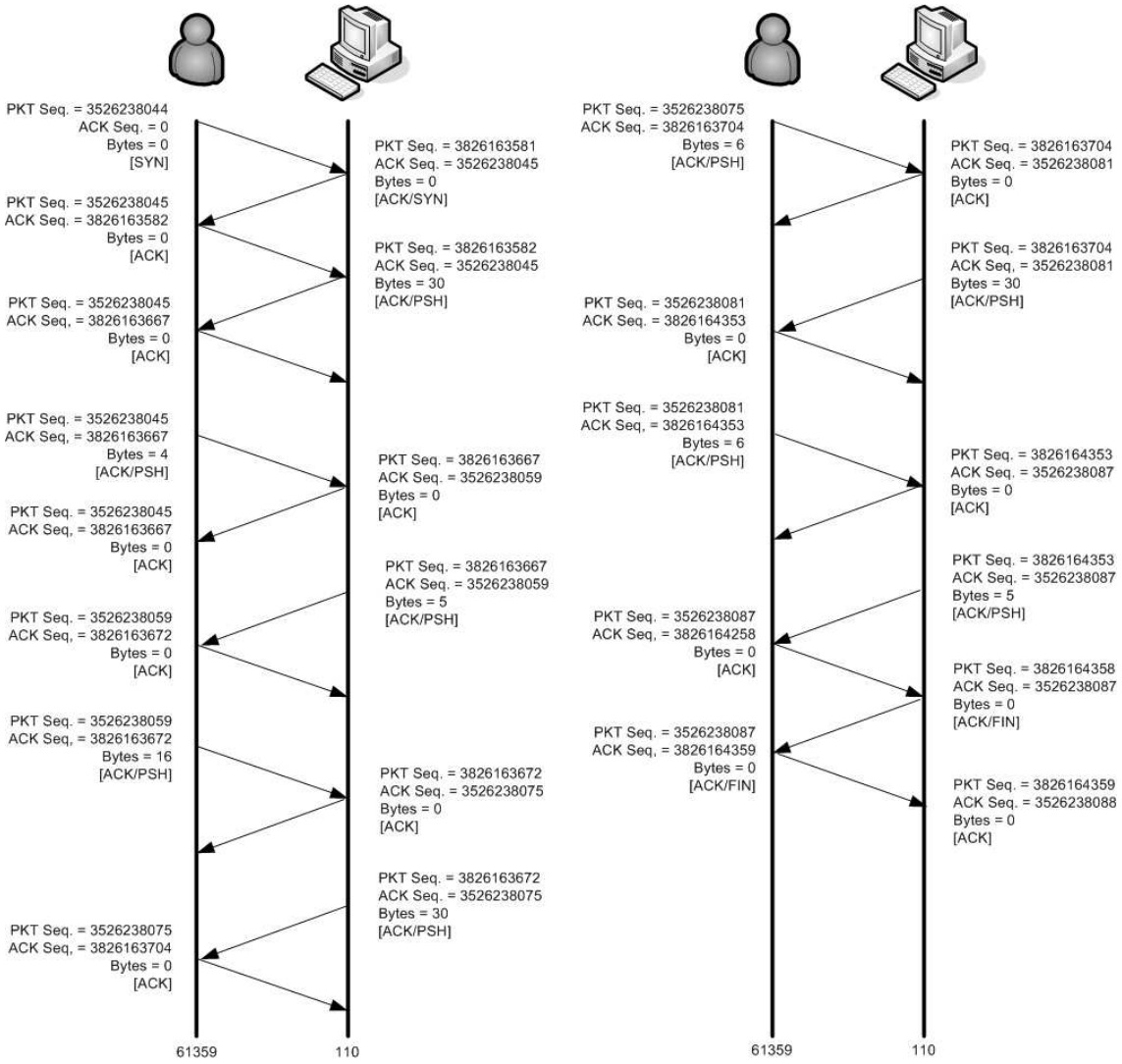

Figura 4.5: Linha de tempo de uma conexão normal.

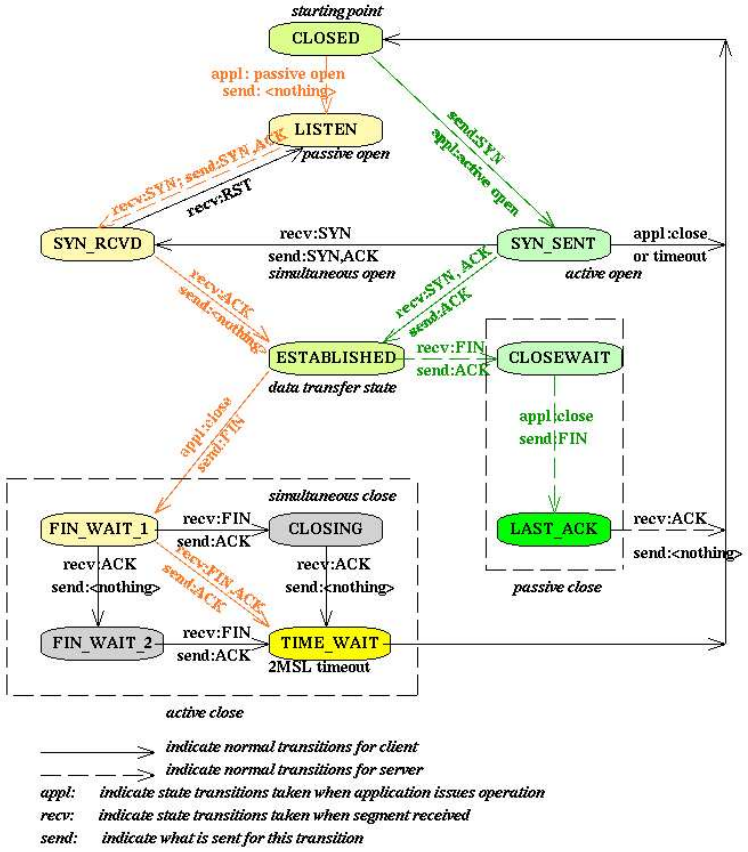

Figura 4.6: Diagrama de estados de uma conexão normal.

4.2.2 Varredura de portas usando a flag SYN

Esta técnica é também conhecida como varredura meio-aberta. Isto se deve a sua característica de não completar a conexão TCP, muito embora a inicie de modo padrão. Assim, a máquina cliente inicia por meio de um pacote endereçado ao servidor remoto onde a flag SYN está ativa e este responde com um pacote de retorno onde estão assinaladas as flags SYN e ACK. A máquina cliente então envia um novo pacote ao servidor onde a flag RST está assinalada, abortando a conexão, conforme apresentado na Figura 4.7.

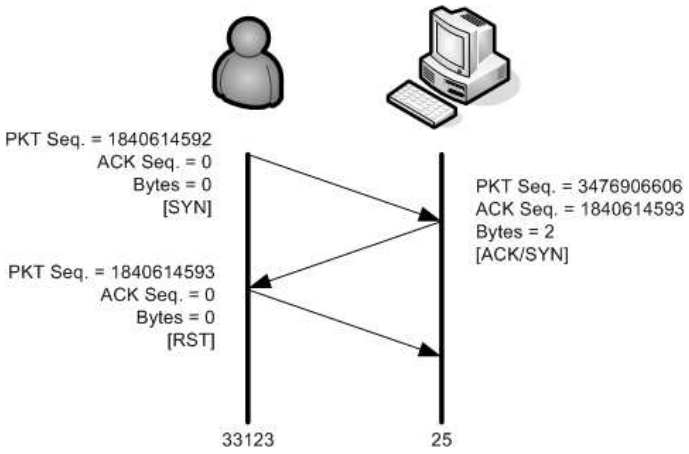

Figura 4.7: Linha de tempo de uma varredura SYN.

Na Figura 4.8 é apresentado o diagrama de estado TCP da conexão provida pela varredura SYN a um servidor que apresenta o serviço válido. Como podemos observar, a máquina de estado é percorrida de maneira que podemos considerar pouco usual, pois não são percorridos todos os estados previstos do protocolo TCP para uma conexão considerada normal.

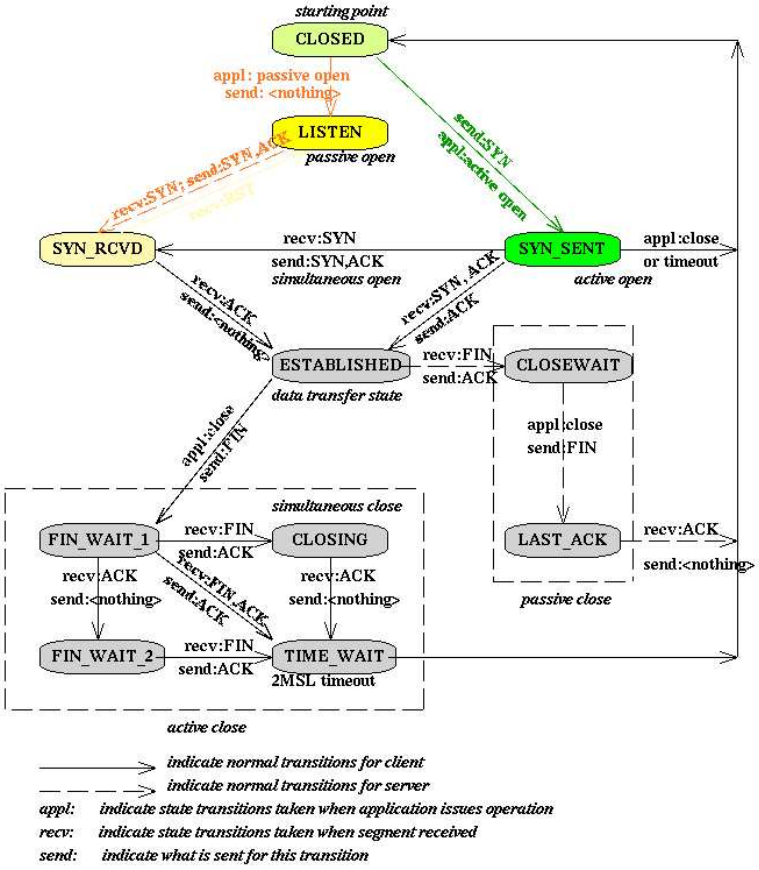

Figura 4.8: Diagrama de estados de uma conexão SYN.

4.2.3 Varredura de portas usando connect

Esta varredura é considerada a mais básica dos métodos de exploração do TCP. A chamada do sistema (*system call*) connect() fornece ao servidor todos os instrumentos necessários para abrir uma conexão e associá-la a uma porta de serviço específica considerada necessária a operação de uma facilidade.

A utilização desta técnica dispensa a necessidade de privilégios adicionais, sendo assim acessível a qualquer usuário de um sistema. Como a execução desta primitiva é rápida, torna-se possível a construção de programas especialmente elaborados com o objetivo de mapear as portas de serviço disponíveis em um servidor remoto. Para tanto podem ser utilizadas ténicas de investigação linear ou em paralelo onde são abertos diversos soquetes simultanêos.

Na Figura 4.9, observamos o desenvolvimento de uma conexão onde a máquina cliente envia um pacote TCP, onde a flag SYN está assinalada, e obtém como resposta um pacote TCP indicando o recebimento, ou seja, um ACK e o indicativo que este serviço não está disponível no ambiente remoto por meio da flag RST e assim encerra-se a conexão.

Na Figura 4.10, observamos o desenvolvimento de uma conexão onde a máquina cliente envia um pacote TCP, estando a flag SYN assinalada iniciando assim a conexão a um serviço disponível no servidor remoto. Como resposta, o servidor envia um pacote TCP com as flags ACK e SYN assinaladas em seguida o cliente responde com a confirmação do recebimento por meio da flag ACK e solicita o encerramento imediato por meio da flag RST.

Na Figura 4.11 é apresentado o diagrama de estados TCP da conexão iniciada pela chamada de sistema connect() a um servidor que apresenta o serviço solicitado. Como podemos observar, o diagrama de estados é percorrido da maneira que podemos considerar normal sem a presença de nenhum evento que impossibilite o exercício dos diferentes estágios deste diagrama.

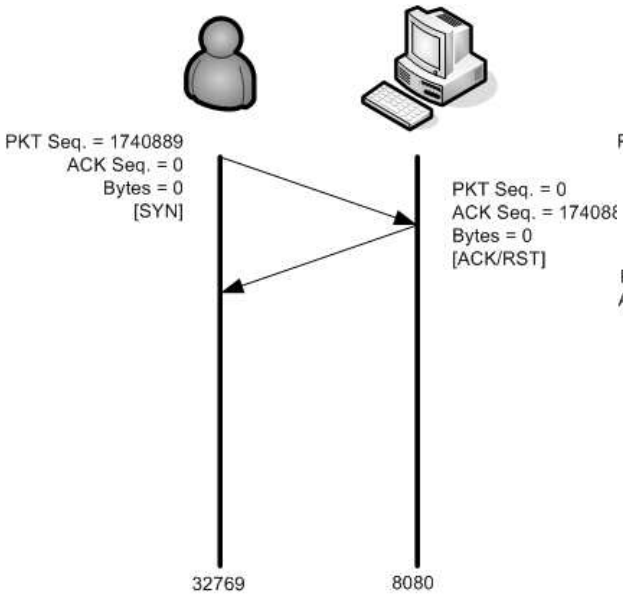

Figura 4.9: Linha de Tempo em um connect sem Serviço

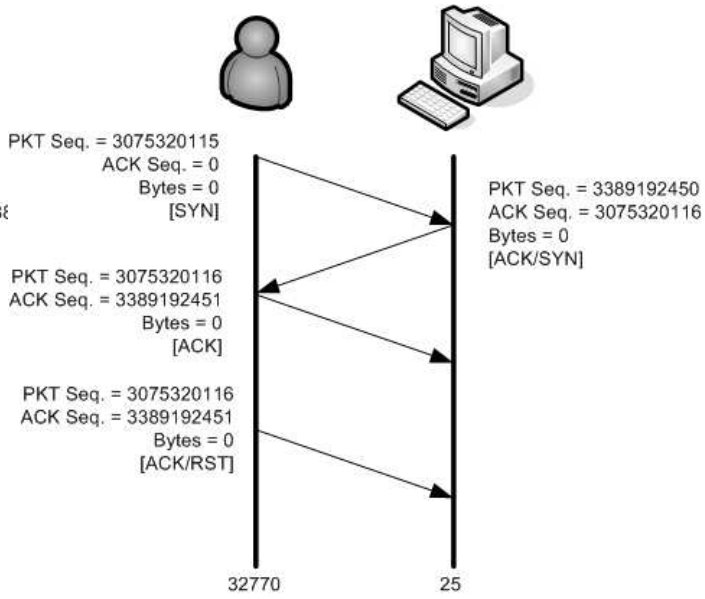

Figura 4.10: Linha de Tempo em um connect com Serviço

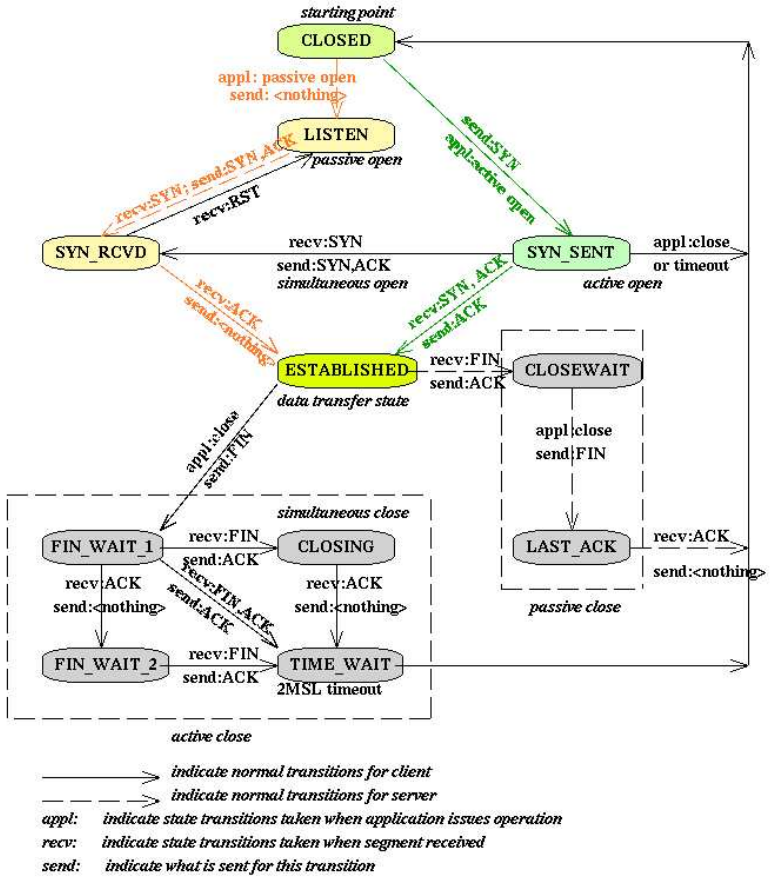

Figura 4.11: Diagrama de estado de uma conexão connect.

4.2.4 Varredura de portas usando a flag FIN

A varredura baseada na técnica de exploração SYN tornou-se atualmente pouco efetiva uma vez que a maioria dos controladores de acesso (*firewalls*) estabeleceram filtros que limitam o recebimento destes tipos de pacotes nas redes. Já a exploração baseada em pacotes FIN não sofre das mesmas restrições. Assim, há em alguns sistemas operacionais, como os *MS-Windows*, que respondem com um pacote onde a flag ACK e RST estão ligadas quando existe a disponibilidade do serviço examinado, conforme mostrado na Figura 4.13. Em ambientes Unix tal procedimento não ocorre, uma vez que a pilha TCP compreende não haver nenhum estado e consequentemente nenhuma resposta é produzida mesmo com a existência da disponibilidade do serviço, conforme mostrada na Figura 4.12.

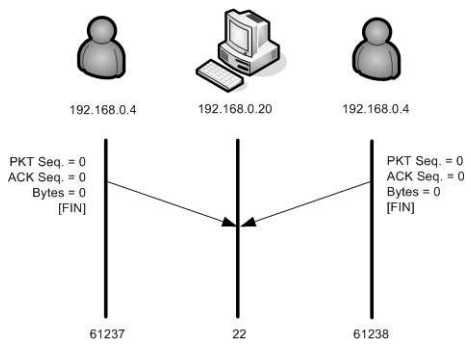

Figura 4.12: Varredura FIN em ambiente Unix

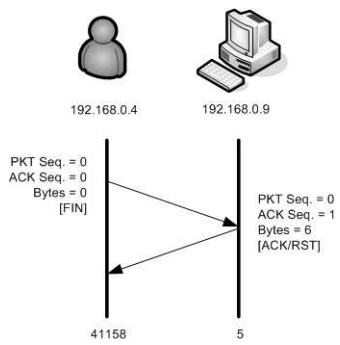

Figura 4.13: Varredura FIN em ambiente Windows

Na Figura 4.14 é apresentado o diagrama de estados TCP da conexão provida pela varredura FIN a um servidor Unix, que apresenta o serviço solicitado. Como podemos observar, nenhum estado é percorrido.

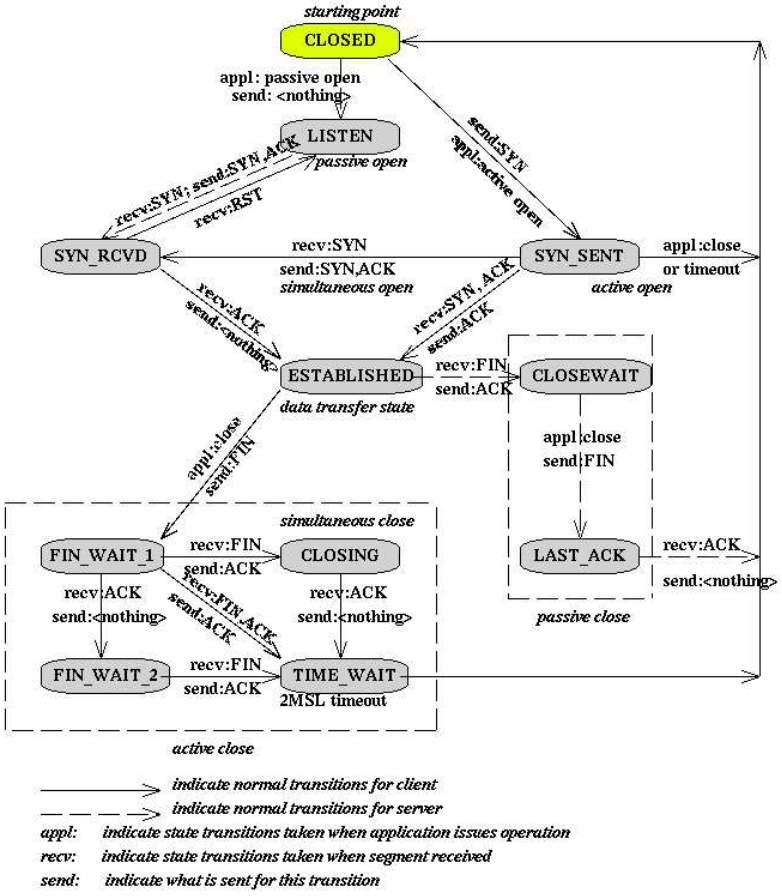

Figura 4.14: Diagrama de estado de uma conexão FIN.

4.2.5 Exemplo de uma varredura ampla

Na Figura 4.15 é apresentada uma visão esquemática e limitada da investigação de portas de serviço entre os números 20 e 23 utilizando a varredura SYN. Podemos identificar a partir deste exemplo o processo de varredura que para sua execução estabelece diferentes instâncias de exame. Posteriormente, estes resultados são agregados pelo programa explorador que apresenta o resultado final desta investigação. Este exemplo é limitado a apenas uma fonte investigando a disponibilidade de serviços de um único servidor remoto.

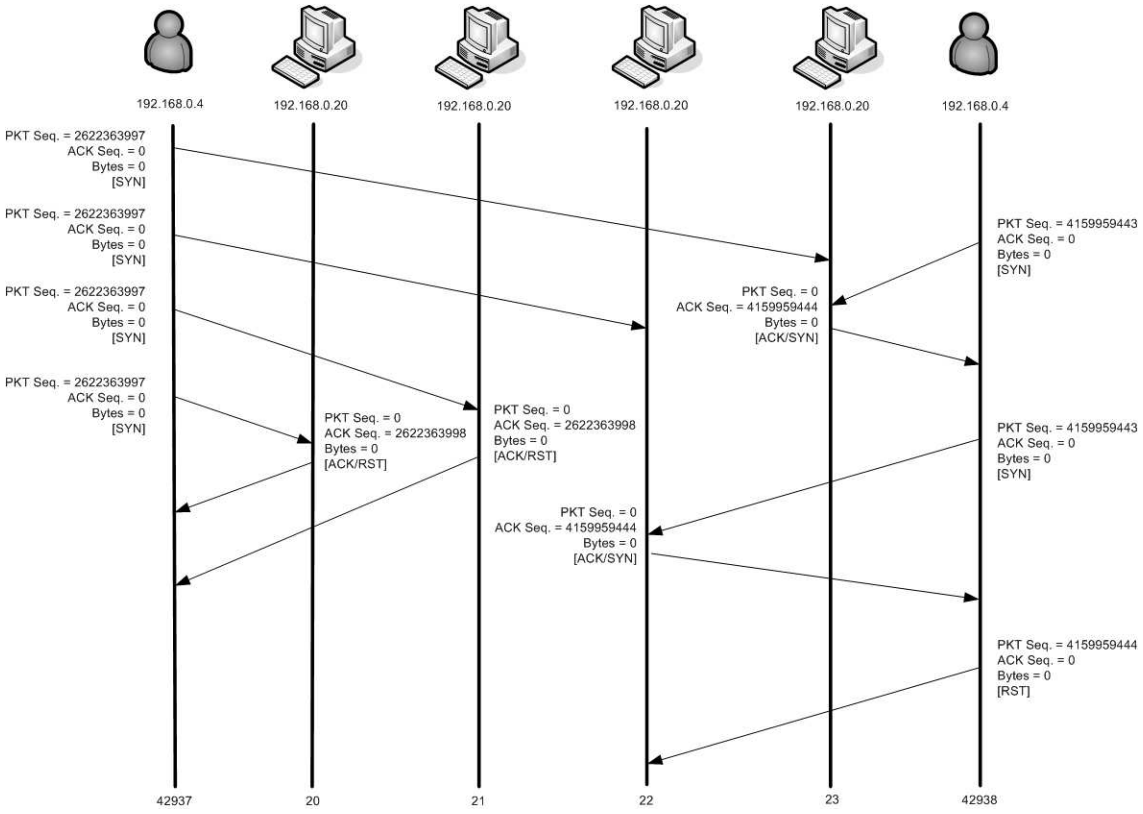

Figura 4.15: Exemplo de varredura em múltiplas portas de serviço.

4.3 Negação de serviço

Segundo Wang[42] o ataque netuno (*neptune*), é uma técnica que explora o início de conexão chamado de *three-way handshake* inundando o TCP com pacotes onde a flag

SYN está ligada. Estes pacotes são destinados a um serviço presente no servidor remoto em volumes elevados. Estes volumes impedem o funcionamento normal do serviço, constituindo assim um ataque classificado como negação de serviço (*DOS - Denial of Service*).

O impacto na infra-estutura é observado devido a espera do pacote de SYN/ACK como resposta do servidor seja reconhecido pelo cliente, assim a conexão permanece em estado meio-aberto por um período de até 75 segundos. Assim o servidor remoto tem que armazenar em sua fila de atividades uma lista que descreva todas as conexões que estão no estado meio-abertas. Como esta fila tem um tamanho limitado, quando este limite é alcançado todos os pacotes são descartados.

4.3.1 Taxonomia do Ataque

Segundo Hussain[40] os diferentes tipos de ataque de negação de serviço, podem ser classificados tanto do ponto de vista de *software* como de inundação do TCP. O primeiro o atacante envia volumes consideráveis de pacotes com o objetivo de exercitar uma falha específica seja do sistema operacional ou da aplicação presente no ambiente remoto, o que ao longo do tempo afetará a operação do serviço alvo. O segundo, um ou mais atacantes enviam pacotes destinados ao servidor remoto de maneira incessante, com o objetivo de comprometer a disponibilidade dos canais de comunicação que ligam este serviço à Internet.

Para que possamos compreender melhor a estratégia geral dos ataques baseados em negação de serviço, é estabelecido um ponto de observação localizado entre a vítma e os diferentes componentes de ataque. São adicionados nesta taxonomia dois componentes importantes, a saber:

- **Máquinas Zumbi:** São máquinas cuja segurança já está comprometida por um atacante e são utilizadas para apoiar ataques a outros ambientes.
- **Máquinas Refletoras:** São máquinas utilizadas para continuar o encaminhamento do ataque preservando a identificação do atacante.

A Figura 4.17, apresenta a ação de máquinas zumbi, a Figura 4.18 apresenta a estratégia de ataque apoiada por um número significativo de máquinas zumbi, a Figura 4.16 apresenta a estrutura mais sofisticada pois além das máquinas zumbi, utiliza máquinas refletoras como apoiadoras do ataque estabelecndo assim uma estratégia multi-camada onde a identidade do ataque se torna mais difusa.

Em todas as estratégias apresentadas, existe a possibilidade de ataques estarem sendo realizados sem que sejam observados pelo ponto de observação estabelecido. A questão central desta estratégia é garantir qual a melhor abordagem para garantir se o ataque está sendo realizado em uma única camada ou em múltiplas camadas. Hussain, propõe o uso do ID do pacote IP bem como o entendimento da duração dos ataques a partir de seu TTL (*time to live*).

4.3.2 Abordagens de Solução

Segundo experimentos de Darmohray[43] os atuais controladores de acesso (*firewalls*) estão concebidos para suportar um ataque de inundação do TCP por meio de pacotes onde a flag SYN está ligada, com uma taxa de até 14000 pacotes por segundo. Assim, como maneira de contornar esta limitação foram elaborados algumas abordagens que procuram acompanhar todo o estado TCP de uma determina sessão. Esta abordagem embora segura, impõe às infra-estruturas um custo de desempenho que muitas vezes pode ser desaconselhável dependendo do tipo de serviço oferecido.

Assim, estão disponíveis atualmente soluções tais como: o Syn Cache[44], o Syn Cookies[52], Syn Defender[45], Syn Proxying[46] e Syn Kill[47], todas baseadas em um processo de contagem dos eventos de pacotes SYN e controle das filas de conexões abertas. Tais processos são diferentes, porém são limitados a limiares fixos e pouco flexíveis na identificação das diferentes estratégias possíveis de ataques presentes nesta abordagem.

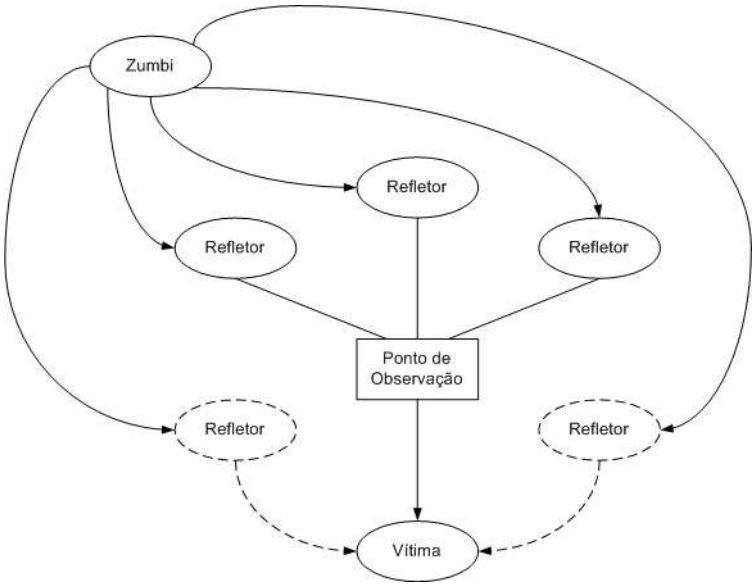

Figura 4.16: Uso de Refletores

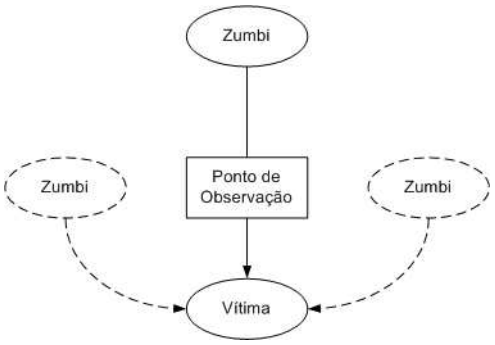

Figura 4.17: Uso com Única Fonte

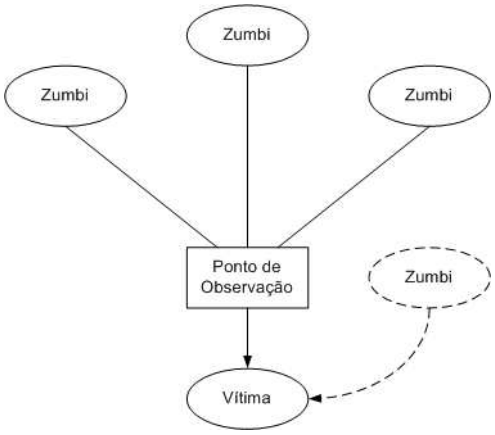

Figura 4.18: Uso com Múltiplas Fontes

4.4 Ataque Convidado

O ataque convidado (*Guest*) é baseado no fato de alguns usuários não escolherem boas senhas. Assim, um atacante busca obter acesso a um ambiente remoto no qual não tem direito de uso, mas conhece um ou todos os usuários presentes no ambiente remoto e procura obter este acesso por meio de múltiplas tentativas de acerto da senha. Para tanto, são utilizados dicionários com nomes de músicas, filmes e outros modismos associados ao momento do ataque.

O ataque convidado é derivado do ataque de dicionário sendo possível o seu emprego

nos principais serviços hoje disponíveis na Internet, tais como: telnet, ftp, pop, rlogin e imap todos estes baseados em contas e senhas.

A Figura 4.19 apresenta o comportamento de um ataque convidado destinado ao serviço POP (*post office protocol*) ao longo de três minutos. Este ataque é realizado por meio de uma única fonte a um único destino, assim podemos observar o volume de tentativas e o tempo médio de duração de cada conexão.

Atualmente, a maioria dos serviços é baseadas em autenticação fraca, ou seja, apenas em conta e senha e um limite de tentivas. Uma vez alcançado o limite de tentativas é estabelecido um período de suspensão do usuário ao referido serviço. Este tipo de acesso atualmente é de fácil detecção, uma vez que podem ser verificadas as diferentes tentativas destes usuários nos diferentes arquivos de registros (*logs*).

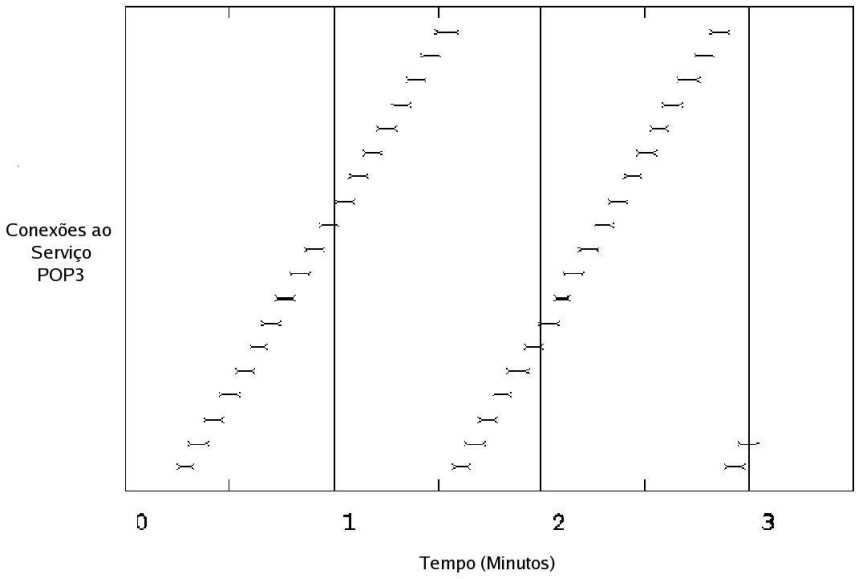

Figura 4.19: Comportamento do Ataque ao Longo do Tempo.

Capítulo 5

Plataforma de Solução

5.1 Introdução

Tendo conhecimento das características dos ataques descritos no capítulo 4, presentes nos dados de referência obtidos a partir do experimento DARPA descrito na capítulo 3. Foi desenvolvida uma plataforma de dados que possibilitou o estudo de diferentes combinações de características. Estas características foram pesquisadas com o objetivo de estabelecer o melhor conjunto de componentes capaz de descrever os ataques presentes neste estudo. Assim, neste capítulo será descrito em profundidade esta plaforma de dados.

5.2 Ambientes de Desenvolvimento

Foram utilizados dois ambientes de desenvolvimento, o primeiro responsável apenas pelas atividades de normalização dos dados de tráfego, e o segundo pelo restante do desenvolvimento das atividades de pesquisa, ou seja, levantamento das caracterítiscas, treinamento, banco de dados e investigação.

O primeiro ambiente computacional é baseado em uma arquitetura PC Intel, composta por dois processadores Xeon 2.4 Ghz, com 1 Gb de memória, e sistema operacional Linux RedHat versão 8, usado de maneira compartilhada com outros usuários. O segundo igual-

mente baseado em uma arquitetura PC Intel da SUN Modelo Cobalt LX50, utilizando dois processadores Pentium III de 1.4 Ghz com 2 Gb de memória e sistema operacional Linux RedHat versão 9.0 usado somente para esta finalidade, tendo com sistema de armazenamento um disco rígido scsi de 40 Gbytes.

5.3 Ferramentas de Desenvolvimento

Os diversos programas escritos para construção desta plataforma, foram baseados na linguagem Perl[56]. Apesar das limitações de desempenho desta linguagem, esta foi adotada devido a sua facilidade de uso, o que permitiu o rápido desenvolvimento da plataforma de dados. Outro fator importante neste processo de escolha é a simplicidade de integração com o banco de dados MySQL[55] bem como a portabilidade para múltiplas plataformas.

5.4 Premissas de Escolha do Protocolo

No presente trabalho são estudados apenas ataques transportados pelo protocolo TCP. Como podemos observar na Figura 3.2 o protocolo TCP apresenta um volume muitas vezes superior aos demais protocolos UDP e ICMP. Devido a esta granularidade de tráfegos onde são observados um número de sessões de serviço de não ataque misturados a sessões de ataque em diferentes proporções ao longo tempo é possível apurar de maneira mais eficiênte o modelo de detecção proposto.

5.5 Modelo Geral de Referência

Segundo Dihua[48], a plataforma de dados, deve ser entendida como uma estrutura utilizada com o objetivo de apoiar a classificação de dados e com isto ser suficiente para descrever as características principais encontradas em cada transação. Para tanto esta estrutura utiliza uma coleção de programas que validam as regras de captura dos compor-

tamentos presentes nas informações coletadas. Porém Panda[49] nos apresenta que nesta fase devem ser tomados diversos cuidados para garantir que as regras de classificação reflitam a dependência dos dados que são diretamente deduzidas a partir de um banco de dados.

Na Figura 5.1, são apresentados de maneira esquemática os módulos constitutivos do modelo geral da plataforma de dados.

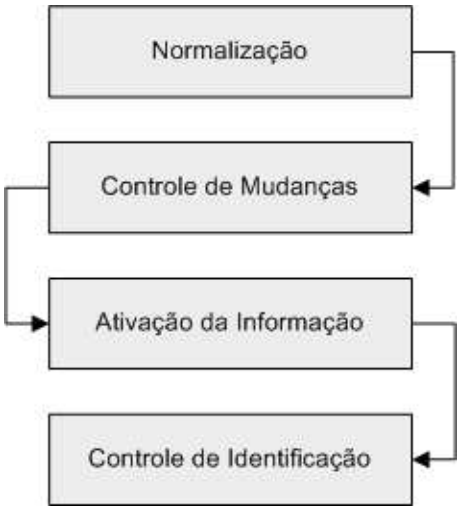

Figura 5.1: Arquitetura Geral da Plataforma de Solução.

5.5.1 Normalização dos Dados

Na Figura 5.2 é apresentado um esquemático do processo de normalização. Este módulo tem por objetivo garantir que a informação de interesse presente nos arquivos originais esteja presente nos arquivos reduzidos e associadamente transforma-las para formatos chamados planos, facilitando sua mainupulação pelos demais programas presentes nesta plataforma.

O arquivo de tráfegos brutos, está no formato do coletor de dados utilizado pelo DARPA. Este formato é de codificação binária, sendo necessário o uso do programa TCPDump[38] para acessar o seu conteúdo. Neste processo são excluídos todos os pacotes que não sejam TCP, armazenando o resultado desta redução em um arquivo temporário.

Após a operação de redução, utiliza-se o programa TCPTrace[53] onde o arquivo de entrada é o arquivo temporário gerado a partir do expurgo de protocolos indesejáveis. O objetivo desta operação é agregar todos os pacotes presentes de uma sessão em um único registro de dados. Neste registro estão presentes diversas totalizações de atributos importantes que descrevem o comportamento dos diferentes pacotes, tais como: número IP da fonte, número IP do destino, número total de pacotes, total de bytes, flags, Tempo de duração da transação e hora inicial bem como a hora final, etc.

Como resultado desta manipulação é gerado um arquivo no formato csv (*comma separated value*). A adoção deste formato, tem por objetivo assegurar que as diferentes aplicações possam garantidamente ler sem erros todos os atributos presentes no arquivo de dados. Este padrão é considerado o mais simples dentre as diversas possbilidades de troca de informações entre diferentes aplicações.

O processo de geração destes arquivos planos, foi realizado por meio de um processo em batelada (*batch*) executado em um ambiente computacional disponível nas instalações do laboratório do GTA (grupo de teleinformática e automação). Este equipamento é um PC de arquitetura Intel, composto por dois processadores Xeon 2.4 Mhz, com 1 Gb de memória, tendo como sistema operacional o Linux RedHat versão 8. Foram gastos neste processo em torno de 78 horas de processamento, onde todos os tráfegos DARPA de 1998

e 1999 foram processados.

Foi necessária uma pequena adaptação do código fonte do aplicativo TCPTrace, no geral buscou-se inserir a data e hora inicial e final no arquivo de saída padrão no formato CSV. Esta informação foi necessária para garantir o padrão de codificação do tempo utilizado em toda a massa de dados.

A normalização utilizada nos tráfegos brutos de eventos se limitou a ler o arquivo original de eventos e adequa-los ao padrão csv, facilitando a manipulação nas etapas posteriores.

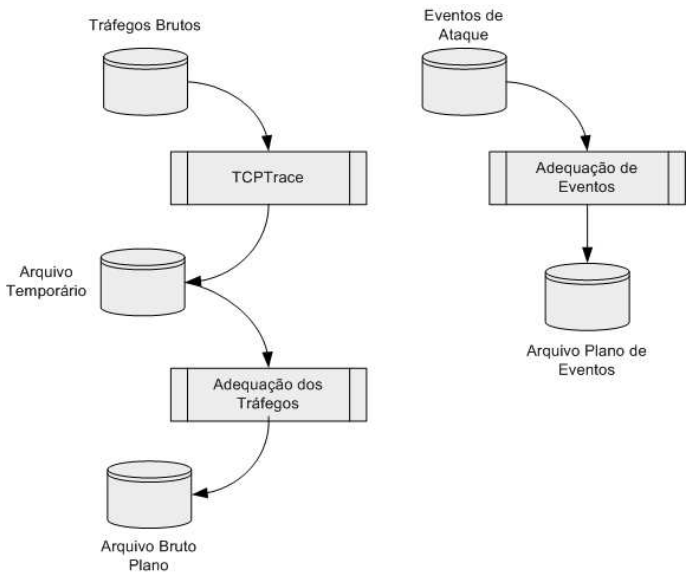

Figura 5.2: Modelo Geral de Normatização de Dados.

5.5.2 Atributos das Sessões

Na Tabela-5.1 são apresentados os atributos que descrevem uma sessão. Estes dados são observáveis após o processamento da ferramenta TCPTrace. Alguns destes atributos serão utilizadas para formação das componentes que descrevem cada ataque presente neste estudo.

5.5.3 Controle de Mudanças

Na Figura 5.3 é apresentado um esquemático do processo de controle de mudanças, que tem por objetivo realizar as alterações necessárias do arquivo plano bruto facilitando a sua integração com o banco de dados.

Como a manipulação dos diferentes eventos de ataque estão descritos pela data e hora de sua ocorrência, foi utilizada uma abordagem de traduzir o tempo no formato hora, minuto e segundo de uma a data expressa em dia, mes e ano em um único número inteiro capaz de representar o tempo.

A solução adotada foi traduzir o tempo em segundos, como era conhecido a data de ocorrência de cada registro foi utilizado o pacote DateCalc[54]. Este pacote calcula o número de dias transcorridos a partir de uma data de referência. Conhecido o número de dias transcorridos até a data descrito no registro de ataque bastava multiplicar por 86400 que descrevem o número de segundos presente em um dia, adionado a este tempo o numero de segundos transcorridos no dia do ataque.

Adicionalmente aos dados gerados pelo programa TCPTrace, foram acrescidos dois campos que descrevem o nome do ataque e sua respectiva classe. Nesta fase, estes campos são preenchidos com os dados *Não Ataque* e *Sem Classe*, pois ainda não são conhecidas as respectivas informações.

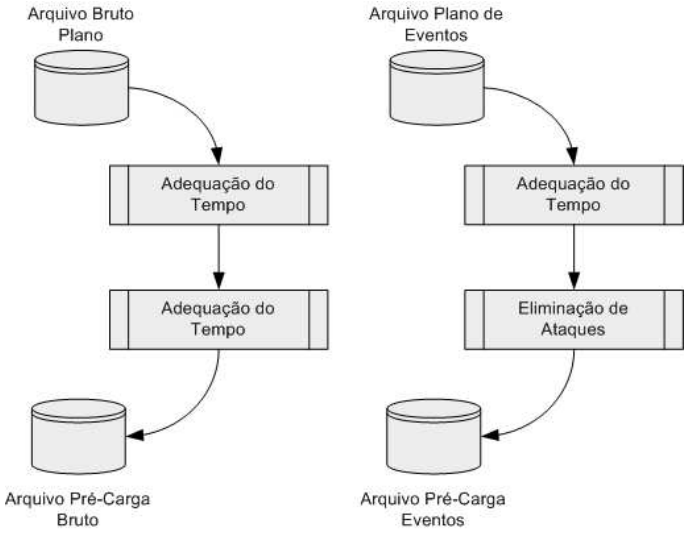

Figura 5.3: Modelo Geral do Controle de Mudanças.

Tabela 5.1: Lista de Atributos das Sessões

Start_Time	total_packets_a2b	total_packets_b2a
resets_sent_a2b	resets_sent_b2a	ack_pkts_sent_a2b
ack_pkts_sent_b2a	pure_acks_sent_a2b	pure_acks_sent_b2a
sack_pkts_sent_a2b	sack_pkts_sent_b2a	dsack_pkts_sent_a2b
dsack_pkts_sent_b2a	max_sack_blksack_a2b	max_sack_blksack_b2a
unique_bytes_sent_a2b	unique_bytes_sent_b2a	actual_data_pkts_a2b
actual_data_pkts_b2a	actual_data_bytes_a2b	actual_data_bytes_b2a
rexmt_data_pkts_a2b	rexmt_data_pkts_b2a	rexmt_data_bytes_a2b
rexmt_data_bytes_b2a	zwnd_probe_pkts_a2b	zwnd_probe_pkts_b2a
zwnd_probe_bytes_a2b	zwnd_probe_bytes_b2a	outoforder_pkts_a2b
outoforder_pkts_b2a	pushed_data_pkts_a2b	pushed_data_pkts_b2a
SYNFIN_pkts_sent_a2b	SYNFIN_pkts_sent_b2a	req_1323_wsts_a2b
req_1323_wsts_b2a	adv_wind_scale_a2b	adv_wind_scale_b2a
req_sack_a2b	req_sack_b2a	sacks_sent_a2b
sacks_sent_b2a	urgent_data_pkts_a2b	urgent_data_pkts_b2a
urgent_data_bytes_a2b	urgent_data_bytes_b2a	mss_requested_a2b
mss_requested_b2a	max_segm_size_a2b	max_segm_size_b2a
min_segm_size_a2b	min_segm_size_b2a	avg_segm_size_a2b
avg_segm_size_b2a	max_win_adv_a2b	max_win_adv_b2a
min_win_adv_a2b	min_win_adv_b2a	zero_win_adv_a2b
zero_win_adv_b2a	avg_win_adv_a2b	avg_win_adv_b2a
initial_window_bytes_a2b	initial_window_bytes_b2a	initial_window_pkts_a2b
initial_window_pkts_b2a	ttl_stream_length_a2b	ttl_stream_length_b2a
missed_data_a2b	missed_data_b2a	truncated_data_a2b
truncated_data_b2a	truncated_packets_a2b	truncated_packets_b2a
data_xmit_time_a2b	data_xmit_time_b2a	idletime_max_a2b
idletime_max_b2a	hardware_dups_a2b	hardware_dups_b2a
throughput_a2b	throughput_b2a.	Status

5.5.4 Ativação dos Dados

Na Figura 5.4 é apresentado um esquemático do processo de ativação dos dados, este módulo tem por objetivo preencher o banco de dados com as informações já processadas nas etapas anteriores. Assim, esta fase se restringe a executar o carregamento do banco de dados com as informações presentes nos arquivos brutos e de eventos.

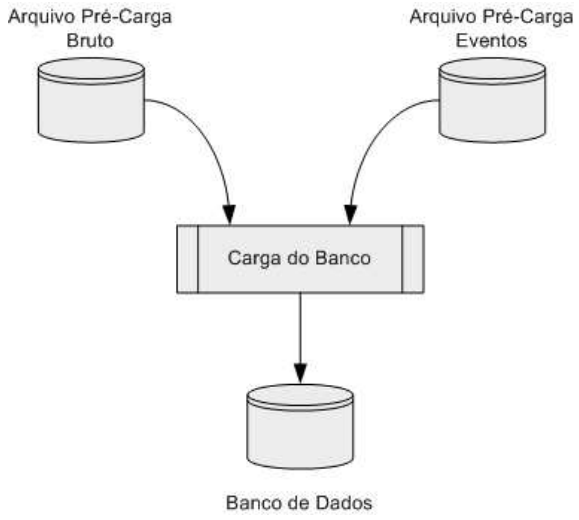

Figura 5.4: Modelo Geral da Ativação dos Dados

5.5.5 Controle de Identificação

Na Figura 5.5 é apresentado um esquemático do processo de controle de identificação. Este módulo tem por objetivo marcar os registros com ataques e adicionalmente descrever o seu respectivo nome e classe.

O processo de marcação inicia-se pela pesquisa dos ataques descritos na tabela eventos. Estes dados são investigados na tabela bruto e em caso de identificação é gerado um arquivo específico que identifica os eventos. Terminado o processo de geração deste arquivo é iniciada a atividade de inspeção manual de validação dos eventos de ataque e, caso o registro de evento não seja considera válido, este é desconsiderado. Nos casos onde estes são considerados válidos, é produzido um novo arquivo específico que é submetido ao processo final de marcação da tabela bruto do banco de dados.

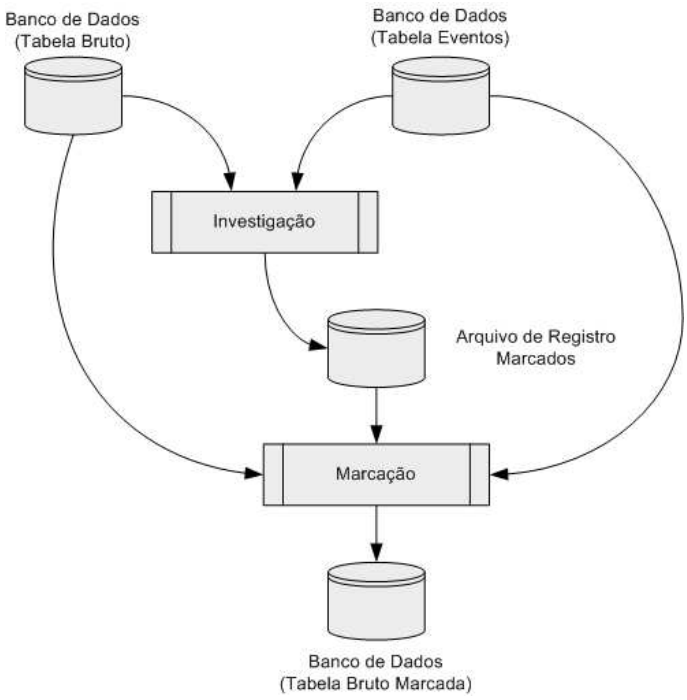

Figura 5.5: Modelo Geral do Controle de Identificação.

5.6 Especificação do Banco de Dados

Foi utilizado como banco de dados para abrigar as diferentes informações de pesquisa a versão 4.0.24 do MySQL[55]. Adicionalmente foram realizados diversos procedimentos com o objetivo de garantir no ambiente Linux o maior desempenho possível deste recurso no ambiente computacional de pesquisa utilizado.

5.6.1 Procedimentos de Melhoria de Performance

Objetivando o máximo de performance foram extendidos diversos parâmetros do segundo ambiente computacional. A Tabela 5.2 desreve os valores utilizados para elaboração deste trabalho.

Tabela 5.2: Parâmetros de Performance

Variável	Valor Padrão	Valor Utilizado
fs.file-max	63095	65535
kernel.shmmax	134217728	33554432
kernel.shmmni	4096	16384
kernel.shmall	134217728	33554432
kernel.threads-max	20206	65535
kernel.msgmnb	16384	65535
kernel.msgmax	8192	40960
kernel.msgmni	16	64
net.core.rmem_default	110592	262144
net.core.rmem_max	131071	262144
net.core.wmem_default	110592	262144
net.core.wmem_max	131071	262144

Capítulo 6

Mapeamento das Características

6.1 Introdução

Neste capítulo serão apresentadas em detalhes as características que definem os padrões de todos os ataques observados neste estudo. Tais particularidades foram apresentadas no Capítulo 4 e reunidas em uma plataforma de dados tal como descrito no capítulo 5.

6.2 Dificuldades Encontradas

Como apresentado no Capítulo 3, o experimento é composto por dois conjuntos de dados: treinamento e validação. No desenvolvimento deste trabalho foi utilizada a base de treinamento de 1998 com o objetivo de elaborar as diversas ferramentas e estruturas de apoio a execução deste estudo. Como abordagem inicial foi utilizado o arquivo de treinamento tanto para as atividades de treinamento como validação, uma vez que o objetivo desta fase era verificar o funcionamento integrado de todos os módulos que constituem a solução deste problema.

Terminada a fase de desenvolvimento das diferentes ferramentas bem como o processo automático de verificação, foi iniciado a aplicação destes conhecimentos ao conjunto de verificação de 1998, onde infelizmente não foi observada a presença de nenhuma ocor-

rência dos ataques descritos neste estudo.

Como solução foi adotada a base de validação de 1999, onde após diversas análises foi concluído que a sua utilização não traria nenhum prejuízo ao modelo desenvolvido.

Esta abordagem mostrou-se mais interessante uma vez que o contexto presente no experimento de 1999 apresenta um cenário de ataques mais diversificado se comparado ao ano anterior. Esta diferença está presente na adoção de novos ataques e diferentes freqüencias de ataques conforme apresentado no Capítulo 3.

6.3 Definição da Representação do Tempo

Compreendida a dinâmica e as características que definem os comportamentos dos diferentes ataques presentes neste estudo, concluimos que somente o uso das componentes que descrevem mais diretamente a ameaça não são suficientes para caracterizar de maneira clara um evento de segurança. Assim foi adicionada a dimensão do tempo.

Para a maioria dos ataques analisados nos baseamos em estudos desenvolvidos por Moore[41] que estabeleceu como cinco minutos um tempo suficientemente grande para sensibilizar os sentidos humanos de quantidade sem que fosse afetado de maneira imprópria os diferentes efeitos presentes na maioria dos ataques.

Na Figura 6.1 é apresentada a organização das diversas sessões no tempo. Neste intervalo são observadas diferentes sessões podendo estas serem normais ou anômalas. Doravante neste estudo, será referenciado como ataque a presença de pelo menos um conjunto de sessões cujo a quantidade da dinâmica e comportamento observados no intervalo de cinco minutos sejam capazes de sensibilizar o algoritmo do detector.

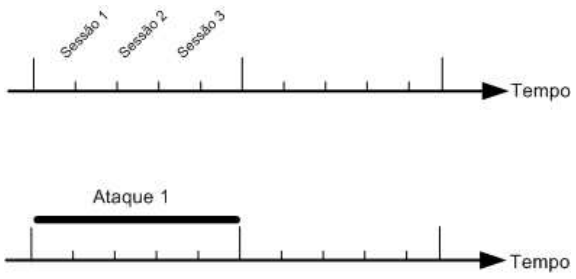

Figura 6.1: Organização das sessões no tempo.

6.4 Critérios de Contagem dos Ataques

Este processo é sensibilizado por uma certa quantidade de sessões que variam de acordo com o tipo do ataque. A disposição das sessões no intervalo de tempo afeta de maneira direta a capacidade de detecção.

Como exemplo, é apresentado na Figura 6.2 um cenário onde é suposto apenas a presença de um único ataque sendo o processo sensibilizado pela presença de quatro sessões anômalas observadas no intervalo de tempo de cinco minutos. Um exemplo com os resultados finais desta premissa é apresentado na Tabela 6.1. Estes resultados apresentam a possibilidade de um ataque presente no tráfego de rede não ser contabilizado como ataque devido a maneira com que este está distribuído ao longo do tempo.

Tabela 6.1: Resultados Finais da Contagem

Número de Ataques Reais	**Número de Ataques Sensibilizados**
1	1
1	2
1	0

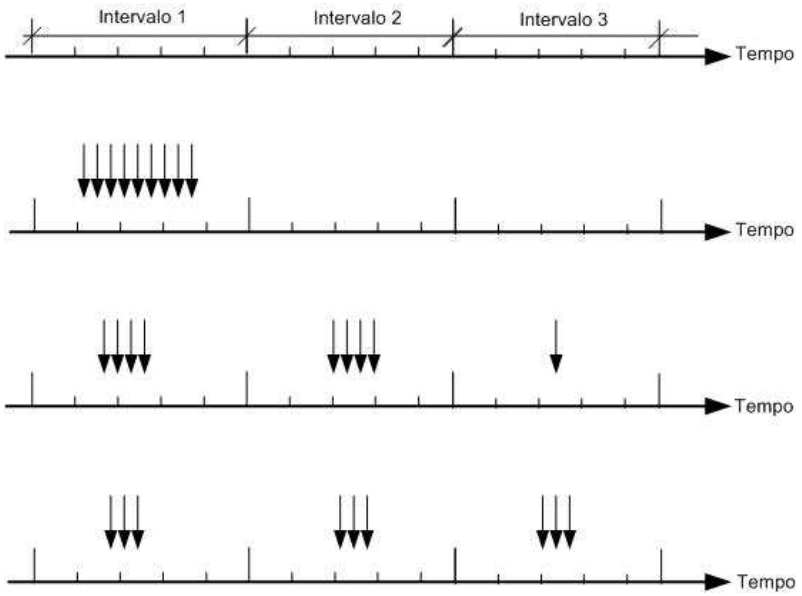

Figura 6.2: Contagem dos Ataques.

6.5 Entendimento Quantitativo dos Ataques

Os conjuntos de dados utilizados neste trabalho apresentam diferentes quantidades de sessões e observamos claramente duas classes que se distribuem com diferentes percentuais, conforme apresentado na Tabela 6.2. Como observado, a granularidade dos ataque presentes no conjuto de verificação é significativamente menor e encontra-se diluída em uma quantidade de tráfego expressiva. Esta diferença é favorável ao modelo do detector, uma vez que a quantidade de dados para treinamento é significativa, o que permite o desenvolvimento de um detector de melhor qualidade.

Tabela 6.2: Distribuição dos ataques nos diferentes conjuntos de dados

	Conjunto de Treinamento		Conjunto de Validação	
Classe	Valores Totais	Percentual da Classe	Valores Totais	Percentual da Classe
Ataque	1.569.123	65.42 %	87.039	7.27 %
Não Ataque	829.580	34.58 %	1.109.733	92.73 %

A distribuição da quantidade de ataques observados nos conjuntos de treinamento e

validação é apresentada nas tabelas 6.3 e 6.4.

Tabela 6.3: Distribuição de ataques presentes no conjunto de treinamento

Nome do Ataque	Quantidade	Nome do Ataque	Quantidade
eject-fail	1	eject	11
format_clear	1	anomaly	11
warez	1	rootkit	16
ffb_clear	1	warezmaster	19
formatfail	1	land	35
dict_simple	1	Guest telnet	50
spy	4	dict	884
perlmagic	4	ipsweep	923
phf	5	nmap	1.031
format	6	warezclient	1.654
ftpwrite	6	back	2.216
imap	6	portsweep	11.617
loadmodule	8	satan	23.948
ffb	10	neptune	1.526.643
multihop	10		

Tabela 6.4: Distribuição de ataques presentes no conjunto de validação

Nome do Ataque	Quantidade	Nome do Ataque	Quantidade
xterm1	1	sechole	9
ffbconfig	2	ps	9
imap	2	phf	11
land	2	secret	13
ls	2	httptunnel	24
selfping	2	guest	27
sendmail	2	queso	28
fdformat	3	guesspop	30
loadmodule	3	ncftp	38
named	3	ntinfoscan	45
netbus	3	ppmacro	64
sshtrojan	3	guesstelnet	67
perl	3	guessftp	80
xsnoop	3	dict	86
ftpwrite	4	back	161
dosnuke	4	portsweep	262
netcat	4	mscan	484
xterm	4	sshprocesstable	753
ipsweep	5	processtable	1113
xlock	5	mailbomb	1351
eject	8	apache2	1727
crashiis	9	satan	8488
casesen	9	neptune	72.075

6.6 Ataque Convidado

O ataque convidado conforme apresentado na seção 4.4 tem sua quantidade apresentada na Tabela 6.5. Como observado, a quantidade de ataques é muito pequena para fins de treinamento, sendo necessário o desenvolvimento de sistemas auxiliares elaborados com o objetivo de sintetizar uma quantidade maior de eventos de ataque possibilitando assim um detector de melhor qualidade.

No conjunto de validação observamos quantidades consideradas pequenas uma vez que sua ocorrência não representa um por cento dos tráfegos do conjunto. Diferentemente do conjunto de treinamento, estes valores são expressivos como fatores de medição da eficiência do algoritmo de reconhecimento utilizado no detector.

Tabela 6.5: Quantização do Ataque Convidado

Conjunto de Dados	**Quantidade de Ataques**	**Quantidade de Sessões**	**Percentual sobre o Total de Sessões**	**Percentual sobre o Total de Ataques**
Treinamento	1	50	0.0021 %	0.0032 %
Validação	4	67	0.0056 %	0.0770 %

6.6.1 Algoritmo de Mapeamento das Características

Baseado nos estudos do comportamento desta anomalia foram escolhidas duas características que melhor descrevem a ocorrência de ataque são elas: *TxGlobal* e *TxResets*. A componente *TxGlobal*, define o número de conexões destinadas ao serviço *Telnet* de um ambiente remoto. A componente *TxResets* define a quantidade de pacotes que estejam com a flag *Reset* ligada oriundas do ambiente remoto destinadas a um cliente. Estas componentes indicam que a sessão foi terminada anormalmente. Ambas as componentes foram agregadas no intervalo de cinco minutos e são apresentadas nas Figuras 6.3 e 6.4.

Os dados de ataque e não ataque são gerados em quantidades diferentes propiciando

a seletividade. Diversas quantidades foram estudadas porém, a que permitiu um melhor desempenho descreve uma proporção de sessenta por cento de ataques contra quarenta por cento de não ataques.

Esquematicamente, é apresentado na Figura 6.5 o algoritmo utilizado para construir as características do ataque convidado. Este algoritmo inicia-se estabelecendo o número de sessões de ataque e não ataque a serem gerados. em Seguida os parâmetros são calculados a partir de números randômicos definidos entre dois patamares, conforme apresentado na Tabela 6.6. Estes limites são distintos para as classes ataque e não ataque. Posteriormente ao cálculo de cada componente, os dados gerados são armazenados em arquivos distintos.

Tabela 6.6: Patamares utilizados para geração artificial do ataque convidado

	Variável TxGlobal		Variável TxResets	
Classe	Valores Mínimos	Valores Máximos	Valores Mínimos	Valores Máximos
Ataque	25	53	45	75
Não Ataque	1	24	1	44

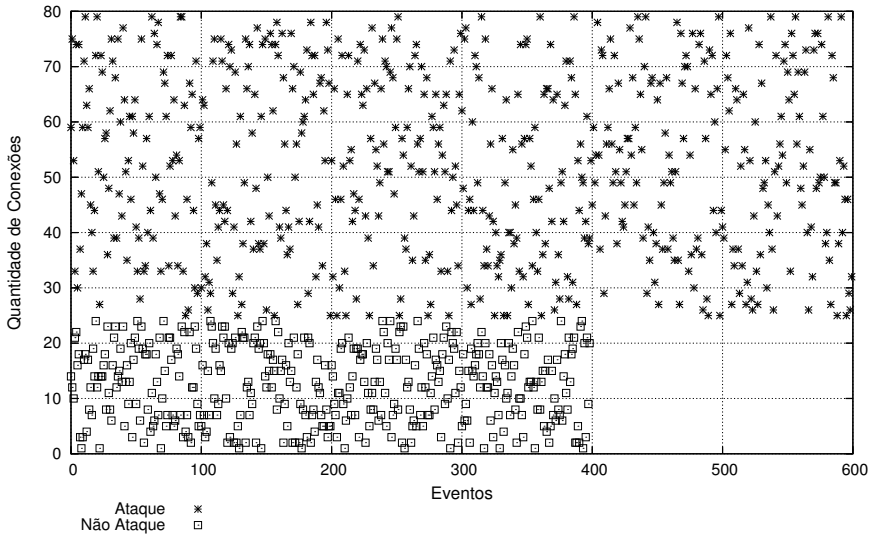

Figura 6.3: Distribuição sintética das características TXGlobal.

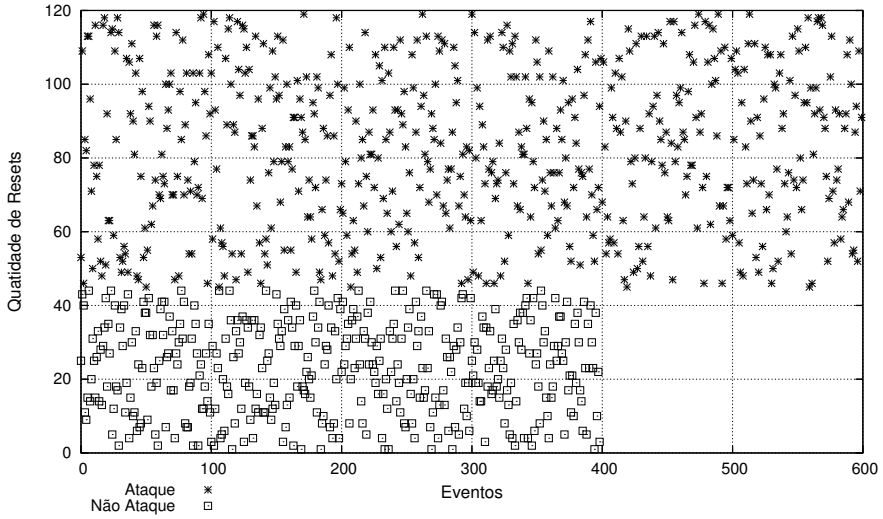

Figura 6.4: Distribuição sintética das características TXReset.

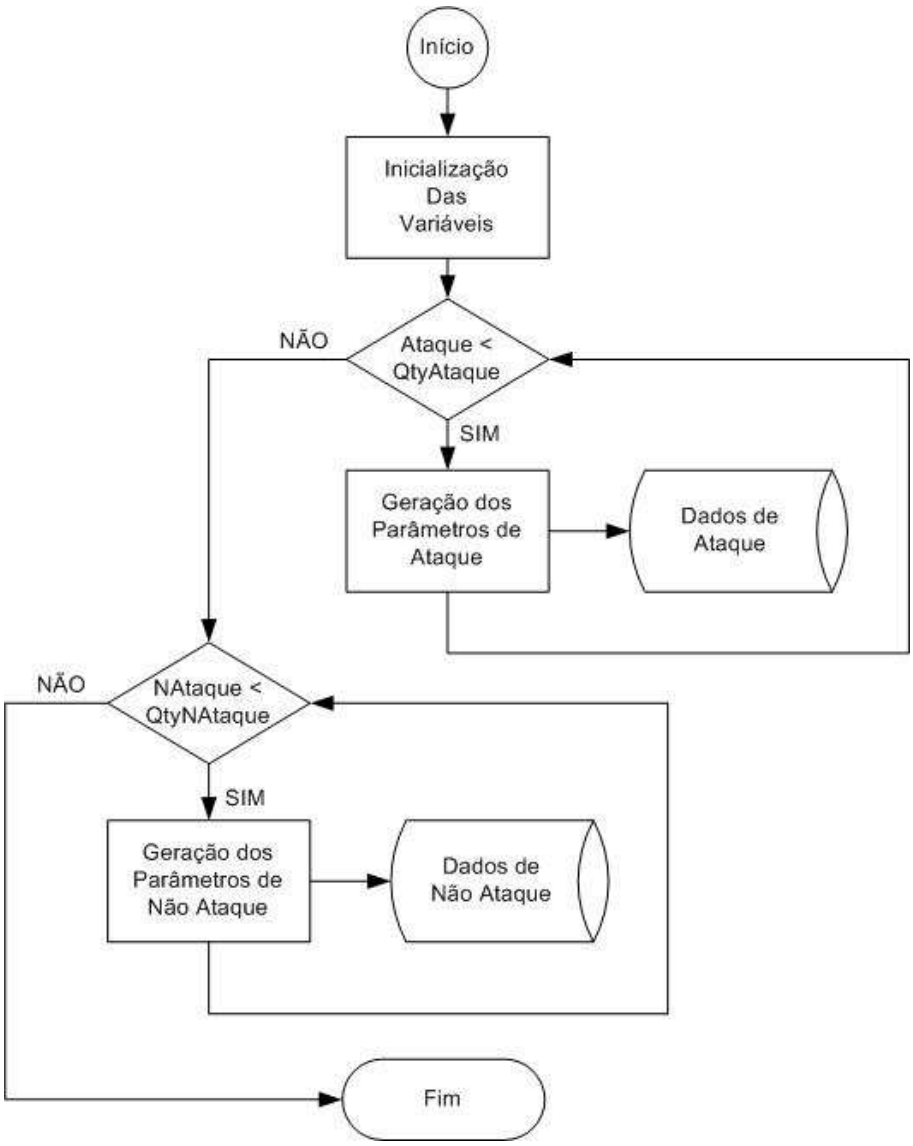

Figura 6.5: Algoritmo de mapeamento das características do ataque convidado.

6.7 Ataque Netuno

O ataque netuno conforme observado na seção 4.3 tem sua quantidade apresentada na Tabela 6.7. Como pode ser observado, a quantidade de ataques é expressiva, favorecendo o treinamento das características selecionadas para a detecção deste ataque. No conjunto de validação observamos quantidades igualmente expressivas de ataque porém, devido a característica de rápida inundação do meio de comunicação, este ataque possibilitou um exercício amplo de investigação das possíveis componentes a serem utilizadas, bem como a alteração do intervalo de tempo, que neste evento é de um minuto.

Tabela 6.7: Quantização do Ataque Netuno

Conjunto de Dados	**Quantidade de Ataques**	**Quantidade de Sessões**	**Percentual sobre o Total de Sessões**	**Percentual sobre o Total de Ataques**
Treinamento	13	1.526.643	63.6 %	97.3 %
Validação	4	72.075	6.0 %	82.8 %

6.7.1 Algoritmo de Mapeamento das Características

Baseado nos estudos do comportamento do ataque, foram escolhidas cinco características que, na visão do especialista, descrevem o comportamento deste evento de segurança. Estas componentes são referenciadas como *BPP*, *FSR_A*, *FSR_B*, *FSR_A2B* e *FSR_B2A*. Todas as componentes utilizadas são agregadas no tempo em intervalos de um minuto.

A componente BPP, descrita pela equação 6.1, é apresentada nas Figuras 6.6 e 6.7. Esta componente descreve os valores totais de pacotes que ocorreram no diálogo entre o cliente e o ambiente remoto e vice-versa.

$$\begin{aligned} BPP &= Total\_Pkts\_A2B + \\ &\quad Total\_Pkts\_B2A \end{aligned} \tag{6.1}$$

A componente *FSR_A*, descrita pela equação 6.2, é apresentada nas Figuras 6.8 e 6.9 e descreve valores o de Resets, SYN e FIN que ocorreram no diálogo entre o cliente e o ambiente remoto.

$$\begin{aligned} FSR\_A &= RST\_Sent\_A2B + \\ &\quad SYN\_Pkts\_Sent\_A2B + \\ &\quad FIN\_Pkts\_Sent\_A2B \end{aligned} \tag{6.2}$$

A componente *FSR_B* é apresentada nas Figuras 6.10 e 6.11 e tem formação semelhante a *FSR_A*, porém expressa os valores de Resets, SYN e FIN que ocorrem entre o ambiente remoto e o cliente, conforme descrito na equação 6.3.

$$\begin{aligned} FSR\_B &= RST\_Sent\_B2A + \\ &\quad SYN\_Pkts\_Sent\_B2A + \\ &\quad FIN\_Pkts\_Sent\_B2A \end{aligned} \tag{6.3}$$

A componente *FSR_A2B*, descrita na equação 6.4, é apresentada nas Figuras 6.12 e 6.13 apresentam os valores do número de Resets, SYN e FIN ponderados com o número total de pacotes transmitidos do cliente ao ambiente remoto.

$$\begin{aligned} Total\_Flags &= Reset\_Sent\_A2B + \\ &\quad SYN\_Pkts\_Sent\_A2B + \\ &\quad FIN\_Pkts\_Sent\_A2B \\ Total\_Pkts &= Total\_Pkts\_B2A + Total\_Pkts\_A2B \\ \\ FSR\_A2B &= \frac{Total\_Flags}{Total\_Pkts} \end{aligned} \tag{6.4}$$

A componente *FSR_B2A*, descrita na equação 6.5, é apresentada nas Figuras 6.14 e 6.15 agrupa os valores do número de Resets, SYN e FIN ponderados com o número total de pacotes transmitidos do ambiente remoto ao cliente.

$$\begin{aligned} Total\_Flags &= Reset\_Sent\_B2A + \\ &\quad SYN\_Pkts\_Sent\_B2A + \\ &\quad FIN\_Pkts\_Sent\_B2A \\ Total\_Pkts &= Total\_Pkts\_B2A + Total\_Pkts\_A2B \end{aligned}$$

$$FSR\_B2A = \frac{Total\_Flags}{Total\_Pkts} \tag{6.5}$$

Esquematicamente é apresentado na Figura 6.16 o algoritmo utilizado para a construção dos arquivos que descrevem o comportamento do ataque. Inicialmente é consultado no banco de dados a maior e menor hora de ataque. Em seguida, inicia-se um ciclo controlado pelas variáveis de tempo MinTime e MaxTime onde, a cada iteração, o valor de MinTime é adicionado de um intervalo de tempo de um minuto.

Como as sessões são rotuladas com o nome do ataque o algoritmo conta a quantidade de eventos netuno que ocorreram no intervalo examinado. Caso exista o ataque são calculados os valores das componentes do ataque.

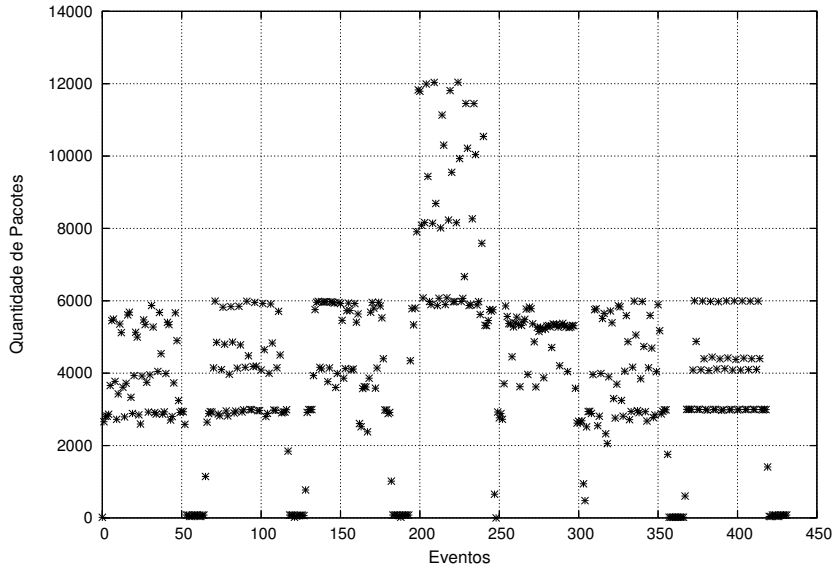

Figura 6.6: Características de BPP em cenário de ataque.

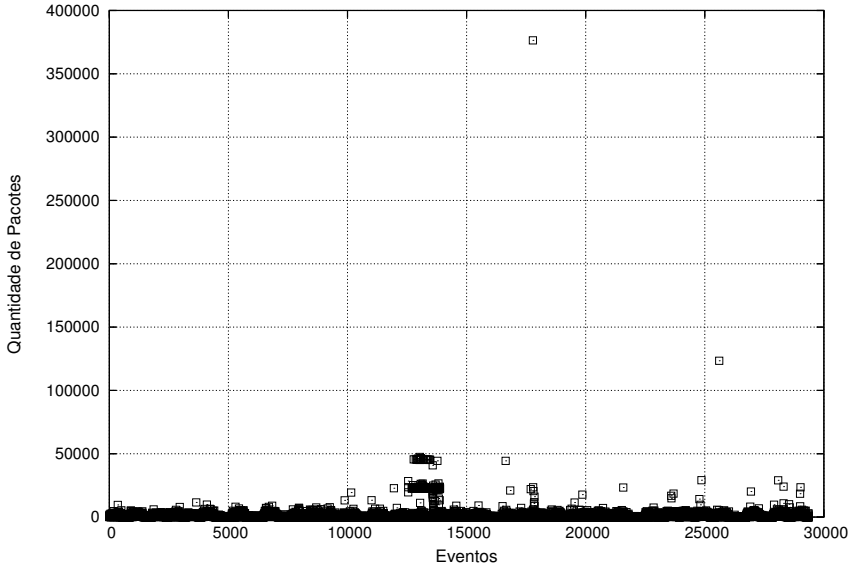

Figura 6.7: Características de BPP em cenário de não ataque.

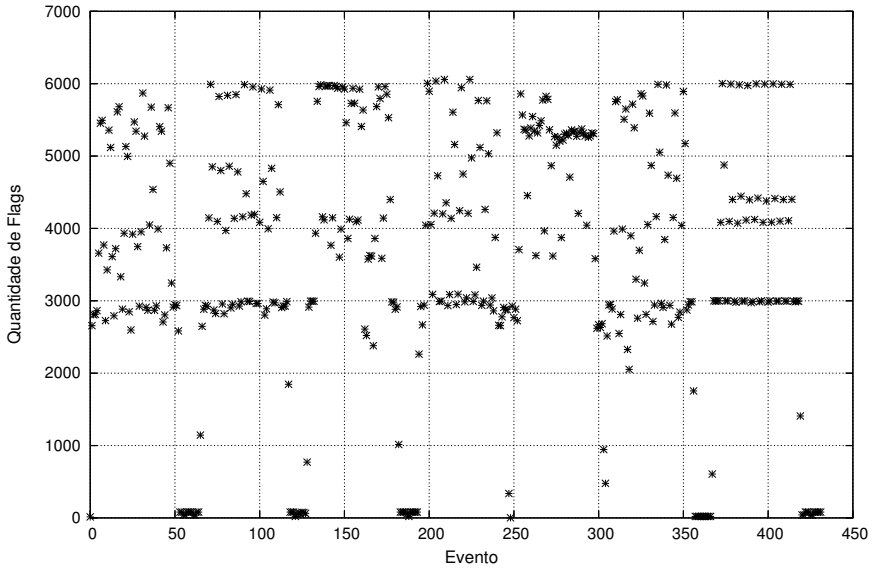

Figura 6.8: Características de FSR_A em cenário de ataque

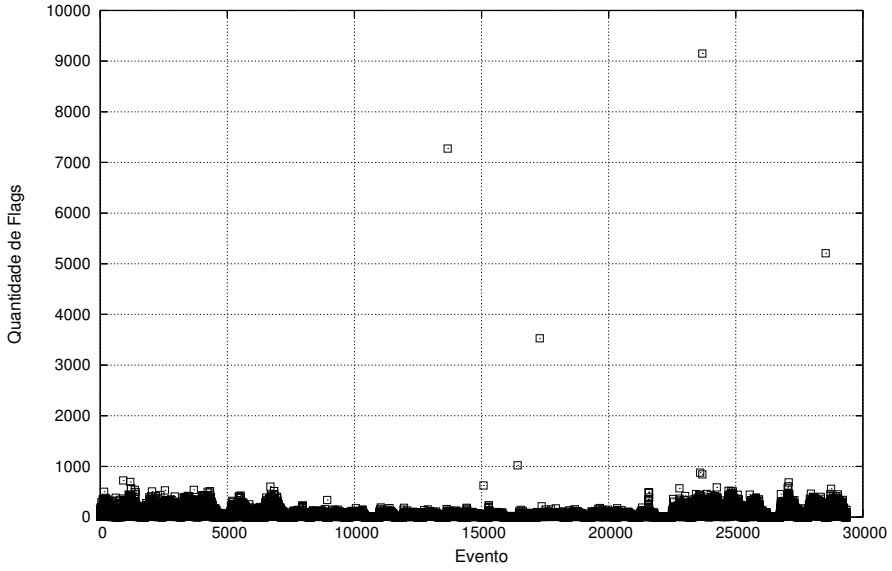

Figura 6.9: Características de FSR_A em cenário de não ataque

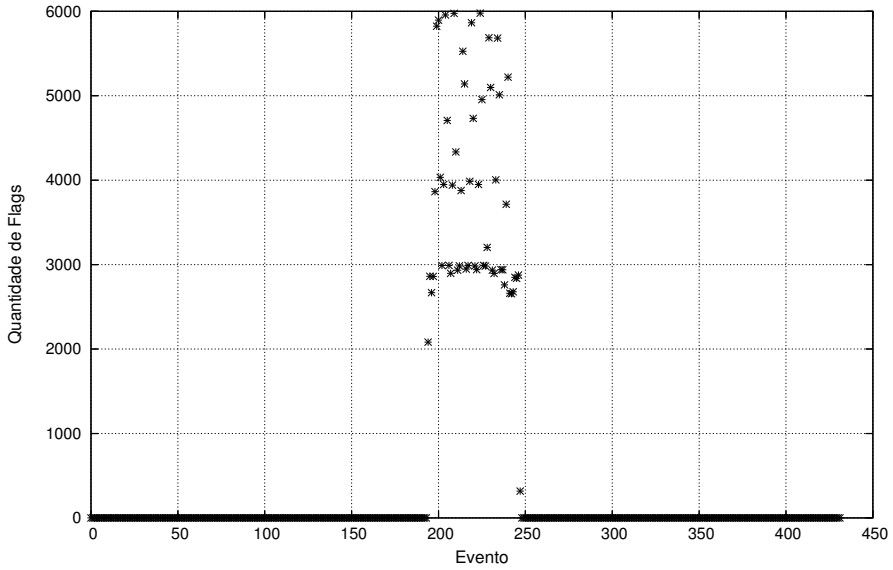

Figura 6.10: Características de FSR_B em cenário de ataque

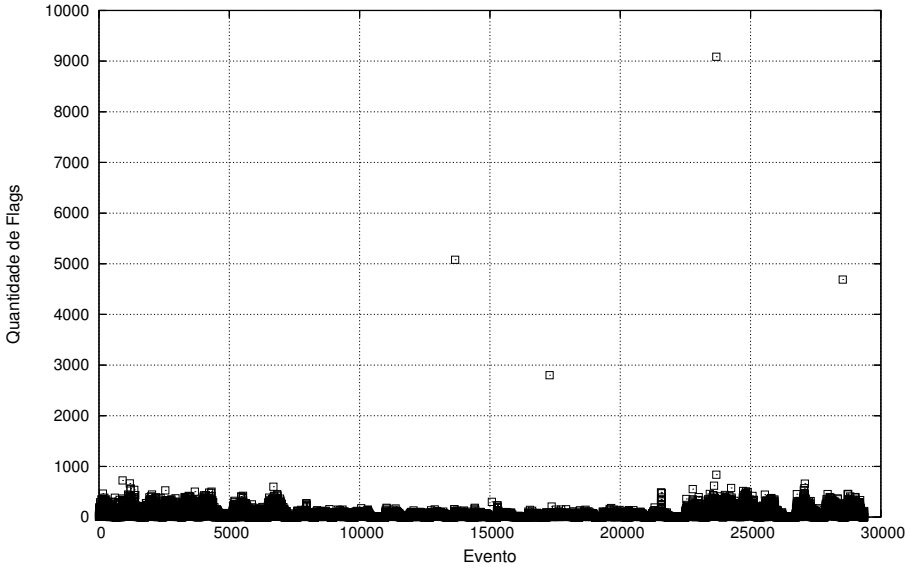

Figura 6.11: Características de FSR_B em cenário de não ataque

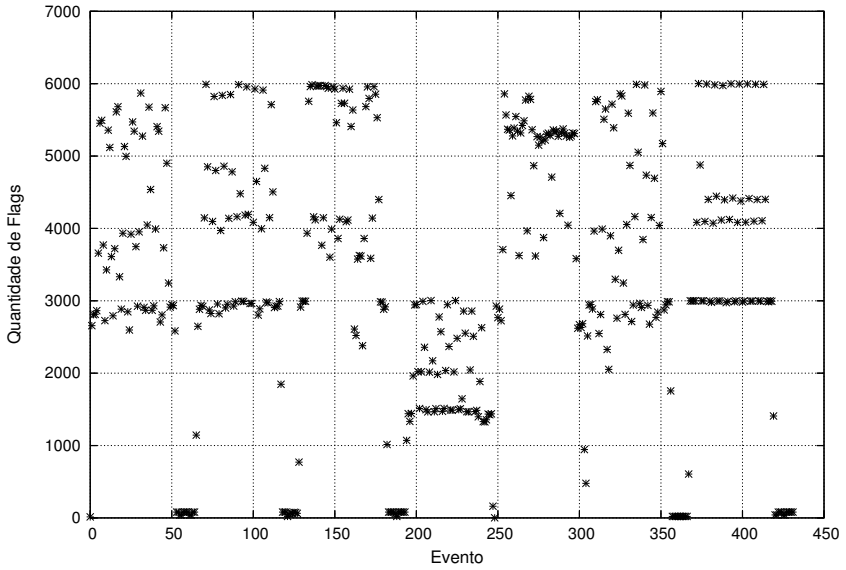

Figura 6.12: Características de FSR_A2B em cenário de ataque

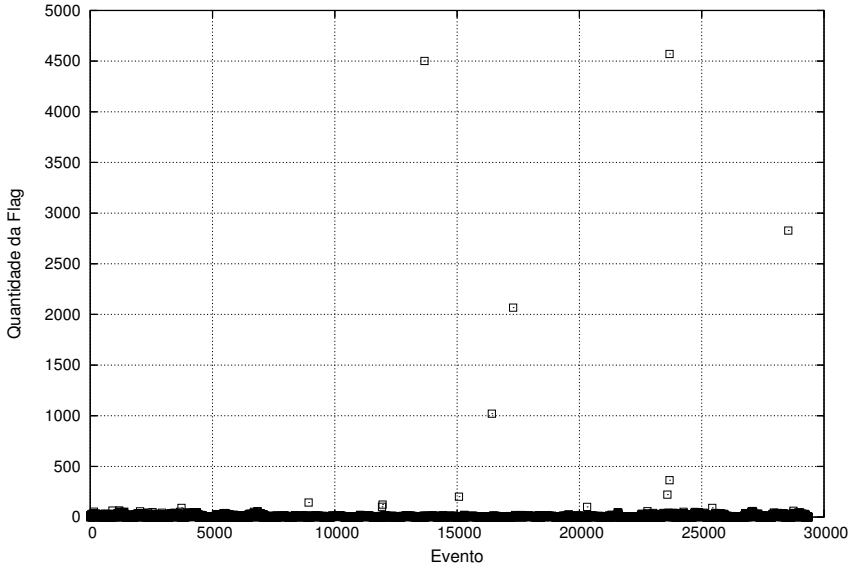

Figura 6.13: Características de FSR_A2B em cenário de não ataque

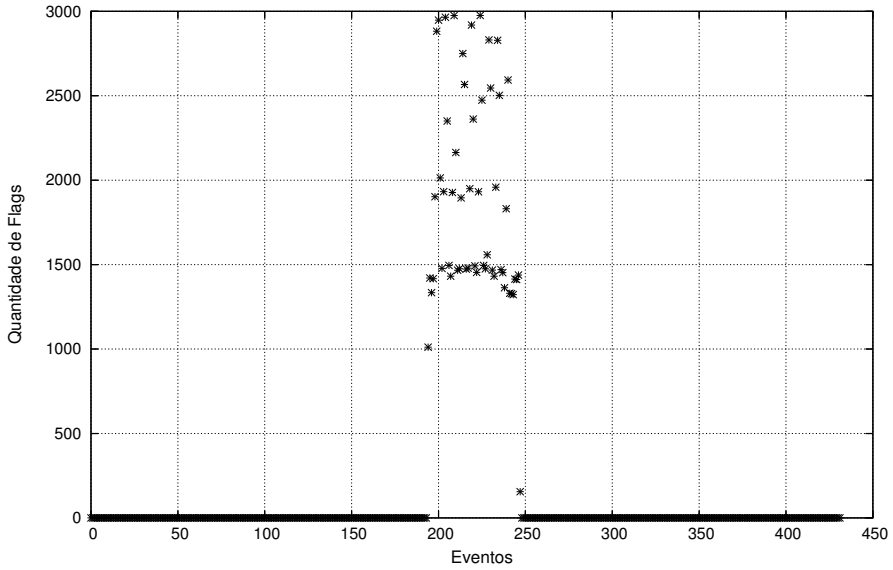

Figura 6.14: Características de FSR_B2A em cenário de ataque

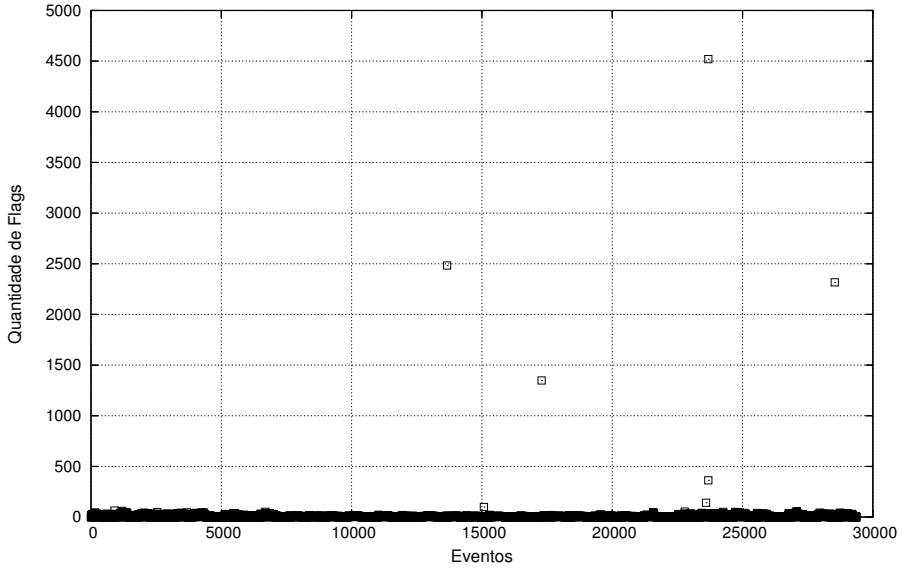

Figura 6.15: Características de FSR_B2A em cenário de não ataque

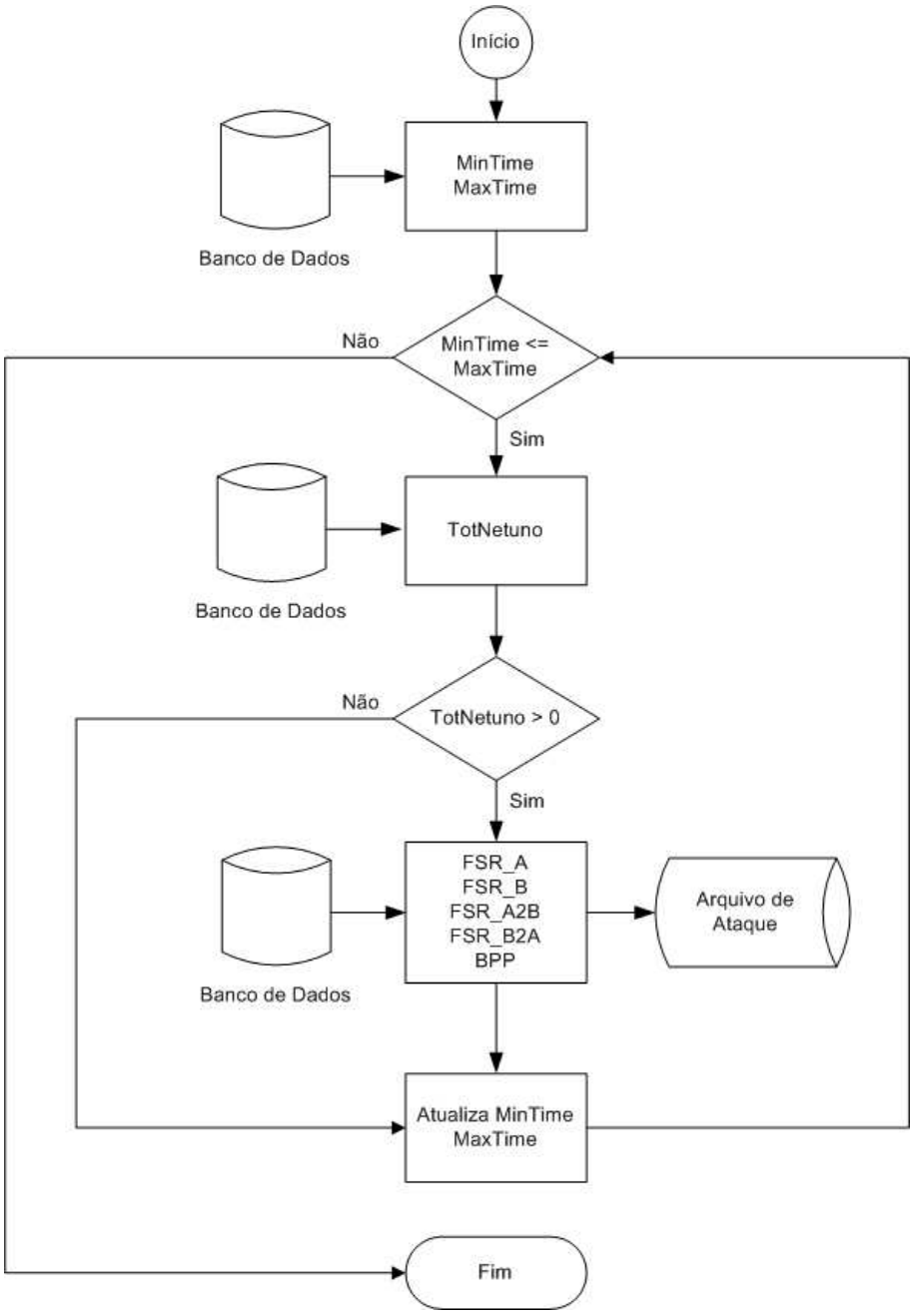

Figura 6.16: Algoritmo de Mapeamento das Características do Ataque Netuno.

6.8 Varredura de Portas

A varredura de portas, tem sua quantidade apresentada na Tabela 6.8. Os números apresentados mostram quantidades pequenas de ataques. Inicialmente acreditava-se ser necessário a produção de valores sintetizados, o que muito dificulta o correto mapeamento das diversas modalidades de ataque.

Com o aprofundamento dos estudos, foi observado que os valores percentuais apresentados são excessivamente baixos devido ao número de ataques netuno presente na base de dados de ataque serem consideravelmente elevados. Ponderando estas quantidades, observamos uma amostra cuja quantidade permitiu o desenvolvimento de um modelo de treinamento robusto capaz de capturar a presença dos eventos de varredura nos tráfegos de validação.

Tabela 6.8: Quantização do ataque varredura de portas

Conjunto de de Dados	**Quantidade de Ataques**	**Quantidade de Sessões**	**Percentual sobre o Total de Sessões**	**Percentual sobre o Total de Ataques**
Treinamento	14	11.617	0.48 %	0.74 %
Validação	13	262	0.022 %	0.30 %

6.8.1 Algoritmo de Mapeamento das Características

Baseado nos estudos do comportamento do ataque varredura de portas, foram escolhidas duas características que, na visão do especialista, descrevem o comportamento do ataque. Estas componentes são referenciadas como *SDPU* e *MBPP*. Todas as componentes utilizadas são agregadas no tempo em intervalos de cinco minutos.

A componente MBPP, descrita pela equação 6.6, é apresentada nas Figuras 6.17 e 6.18. Esta componente descreve os valores totais de pacotes que ocorreram no diálogo entre o cliente e o ambiente remoto, ponderados pela quantidade única de solicitações de um cliente a um serviço remoto.

$$MBPP = \frac{Total\_Pkts\_A2B + Total\_Pkts\_B2A}{TxGlobal} \tag{6.6}$$

A segunda componente, *SDPUnicos*, apresenta a quantidade de clientes, servidores e portas de serviço remotas únicas que ocorreram no intervalo de cinco minutos.

Esquematicamente é apresentado na Figura 6.21 o algoritmo utilizado para a construção dos arquivos que descrevem o comportamento do ataque de varredura. Inicialmente é apagada a tabela auxiliar *FuzzyFluxos* usada neste procedimento com o objetivo de realizar todas as manipulações de dados necessárias. Em seguida são consultadas no banco de dados a maior e a menor hora de ataque. O processo de investigação inicia-se a partir de um ciclo controlado pelas variáveis de tempo MinTime e MaxTime onde, a cada iteração, o valor de MinTime é adicionado de um intervalo de tempo de cinco minutos.

Neste ciclo de tempo, é criada a tabela auxiliar que conterá todos os dados a serem investigados no intervalo de tempo definido por MinTime e MaxTime bem como a consulta da existência de sessões neste intervalo. Caso exista sessões neste intervalo é obitida a lista de servidores e clientes únicos. A partir desta lista são calculados os valores de TxGlobal, SDPUnicos e BPP. A variável TxGlobal contém o valor da quantidade de sessões realizadas por um cliente e um servidor. A variável SDPUnicos contém a contagem de clientes, servidores e portas de serviço únicas no intervalo. A variável BPP contém o

somatório dos pacotes produzidos entre o cliente e o servidor definido no intervalo. Após calculados todos estes fatores, caso a variável TxGlobal tenha um valor maior que zero, é calculado o valor da variável MBPP e gerado o arquivo de mapeamento.

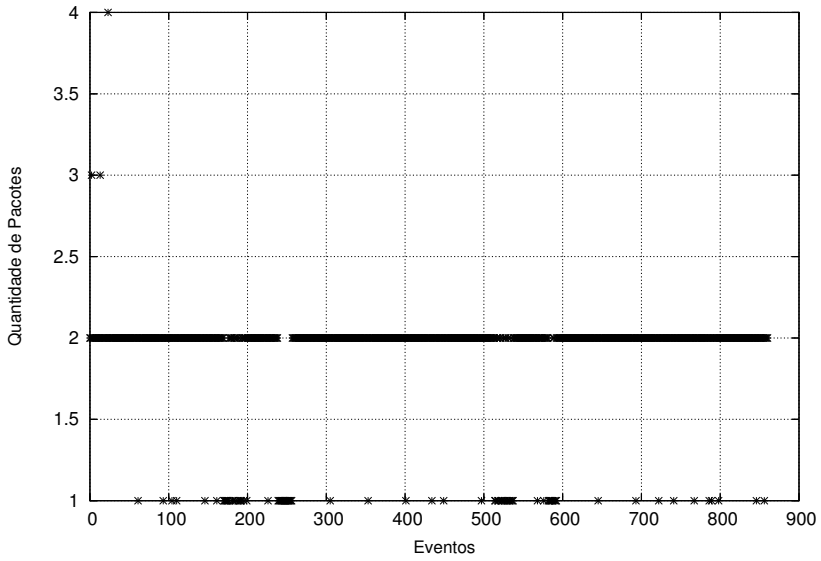

Figura 6.17: Características do MBPP em cenário de ataque.

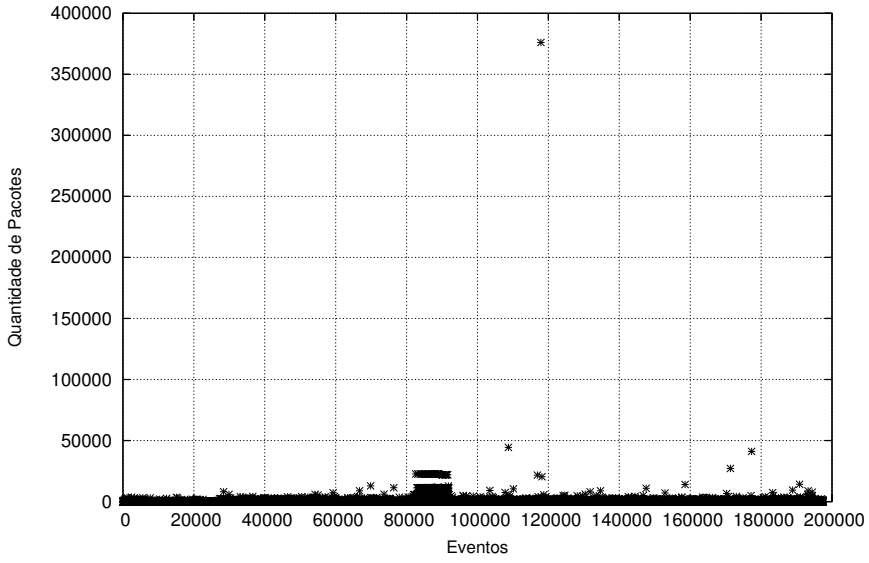

Figura 6.18: Características do MBPP em cenário de não ataque.

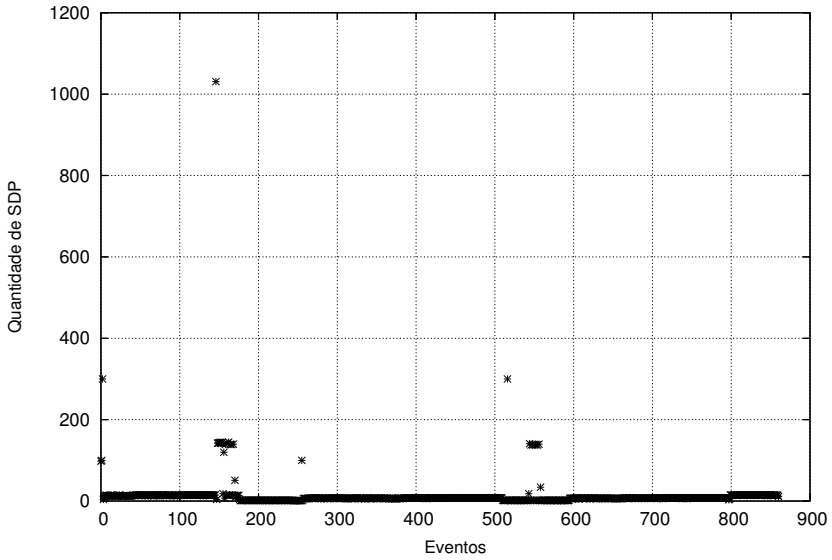

Figura 6.19: Características de SDP Únicos em cenários de ataque.

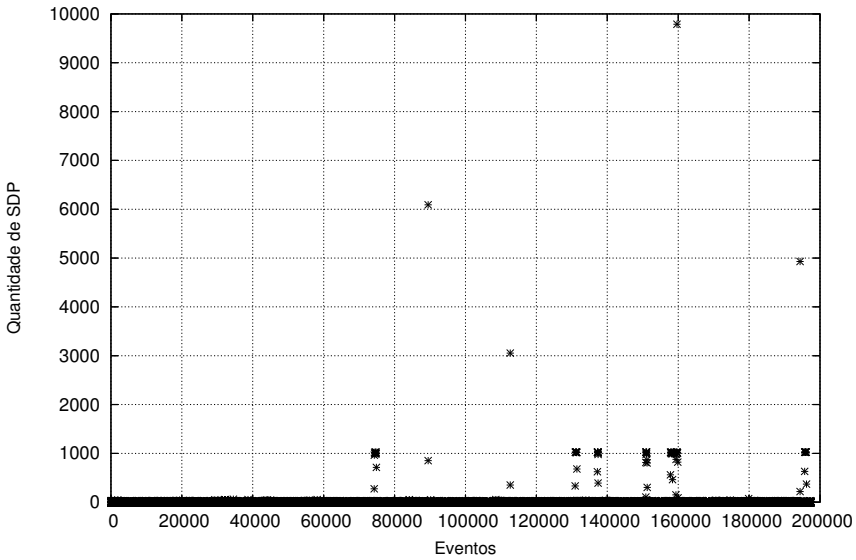

Figura 6.20: Características de SDP Únicos em cenário de não ataque.

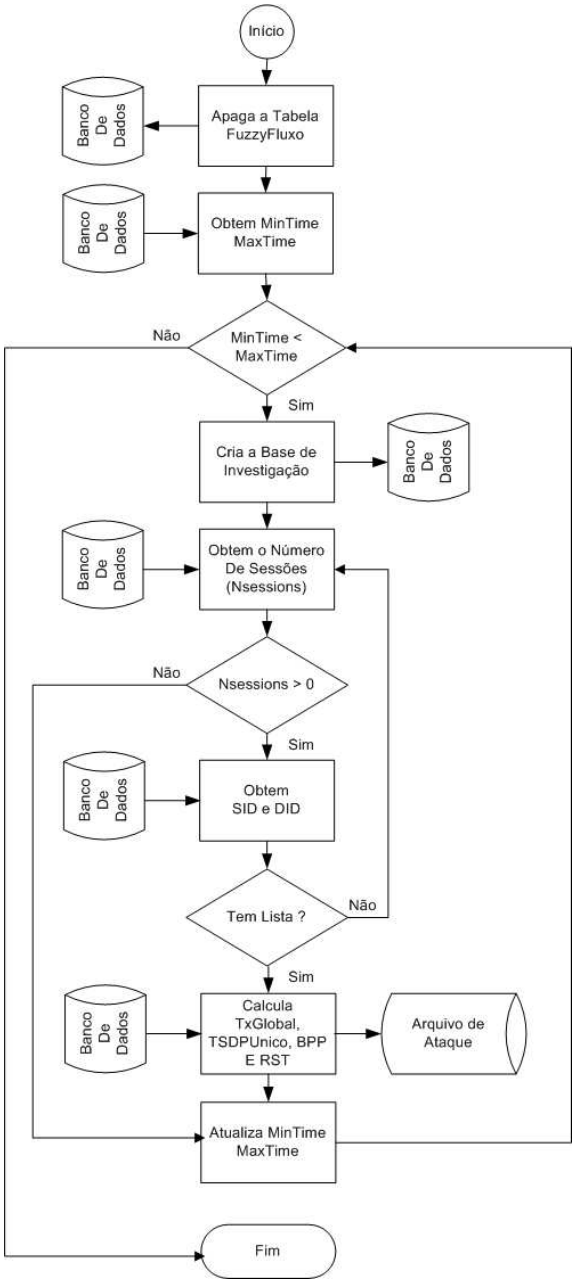

Figura 6.21: Algoritmo de Mapeamento das Características do Ataque Varredura de Portas.

Capítulo 7

Modelagem Matemática

7.1 Introdução

Neste capítulo apresentaremos o modelo NEFCLASS, que é um sistema de inferência neuro-difuso. Será assumido que o leitor já está familiarizado com os conceitos e a notação básica de conjuntos difusos, lógica difusa, regras difusas, normas-t e conormas-t. Algumas definições básicas serão apresentadas no texto com a intenção de explicitar a notação adotada. Ao leitor que desejar uma leitura introdutória é recomendada a leitura presente nas referências [64], [66] e [65].

7.2 Lógica Difusa

Na lógica difusa, uma proposição simples tem a forma: "$x(\theta_1, \theta_2, \cdots, \theta_n)$ é A" , onde $\theta_1, \theta_2, \cdots, \theta_n$ são os nomes de entidades, ou seres, de um domínio ontológico D, x é um atributo, um adjetivo, da ênupla $\theta_1, \theta_2, \cdots, \theta_n$ e A é um conjunto difuso sobre o universo U, isto é, U é a escala de medição do atributo x.

Diz-se então, "$x(\theta_1, \theta_2, \cdots, \theta_n)$ é A" é uma atribuição de uma distribuição de possibilidades A à variável $x(\theta_1, \theta_2, \cdots, \theta_n)$.

Frequentemente, o conjunto difuso A é representado por um termo com um signifi-

cado lingüístico, ou simplesmente um termo lingüístico, tal como alto, médio, baixo, etc.

A variável $x(\theta_1, \theta_2, \cdots, \theta_n)$ é chamada de variável lingüística e assume valores no conjunto de todos os conjuntos nebulosos sobre o universo U correspondente.

Com o objetivo de simplificar a notação, e quando não houver ambigüidade ou perda de generalidade, escrevemos simplesmente "x é A" , ou melhor ainda, "$x = A$" , sem denotar as entidades envolvidas.

Uma regra "se-então" (*if-then*) nebulosa tem a seguinte forma:

$$\textit{if}\ x_1 = A_1\ \textit{and}\ x_2 = A_2\ \textit{and}\ \cdots\ \textit{and}\ x_n = A_n\ \textit{then}\ y = B.$$

O raciocínio difuso, ou aproximado, é um procedimento de inferência que permite deduzir conclusões (proposições simples) a partir de premissas. As premissas são um conjunto de proposições simples (os fatos) e proposições "se-então" (as regras). Para exemplificar, dados os fatos:

$$x_1 = A_{1in} \quad e$$
$$x_2 = A_{2in}$$

e dada a regra difusa

$$\textit{if}\ x_1 = A_1\ \textit{and}\ x_2 = A_2\ \textit{then}\ y = B$$

o raciocínio difuso permite deduzir.

$$y = B_{out}$$

7.3 Classificador NefClass

Segundo Detlef[15], os classificadores Neuro-Difusos utilizam estratégias de aprendizado heurístico baseadas na teoria das redes neurais com o objetivo de apoiar o desenvolvimento de um sistema difuso. Esta abordagem foi desenvolvida com o objetivo de encontrar, para um determinado problema, as funções de pertinência mais adequadas, evitando assim que os especialistas experimentassem um processo longo e tedioso baseado no ensaio e erro.

Assim, diversos pesquisadores desenvolveram a idéia de aplicar algoritmos de aprendizado aos sistemas difusos, o que possibilitou uma nova abordagem chamada de adaptativa ou auto-organizada, conforme descrito nos trabalhos de Procyk[13] e Qiao[14].

Normalmente, os modelos adaptativos utilizam métodos baseados em conhecimento. Porém, o aprendizado dos parâmetros de um sistema neuro-difuso é obtido a partir das redes neurais, que permite o desenvolvimento de um sistema automatizado para o desenvolvimento desta tarefa.

Modernamente, as abordagens neuro-difusas utilizam a representação de multicamadas com alimentação à frente oriundo das redes neurais mas, opcionalmente, os valores difusos podem ser obtidos por outras arquiteturas neurais, tais como a dos mapas auto-organizados, descrita no trabalho de Petri[16]

Nos modelos difusos, as conexões, os pesos, a propagação e as funções de ativação são diferentemente utilizadas, se comparadas às abordagens usuais das redes neurais. Assim, buscando restringir o uso do termo neuro-difuso, seguem abaixo algumas propriedades a serem observadas:

1. Um sistema neuro-difuso é um sistema difuso, que foi treinado segundo um algoritmo de aprendizado derivado da teoria das redes neurais. O processo de aprendizado é operado com informações locais, causando apenas modificações locais, orientado pelo sistema difuso. O processo de aprendizado não é baseado em conhecimento mas orientado pela dinâmica dos dados.

2. O sistema neuro-difuso pode ser examinado como um modelo de três camadas de

alimentação para frente, onde as unidades usadas nesta rede são as *t*-normas ou *t*-conormas utilizadas como funções de ativação. A primeira camada representa os valores de entrada, a segunda representa as regras difusas e a terceira representa os resultados, ou variáveis de saída.

3. O processo de aprendizagem de um sistema neuro-difuso está baseado em propriedades semânticas. Este processo define as modificações locais aplicadas aos parâmetros do sistema.

4. Um sistema neuro-difuso aproxima funções *n*-dimensionais que estão presentes em um conjunto de treinamento, onde são exploradas e desenvolvidas regras que os representam.

Frequentemente, entretanto, tanto as entradas como as saídas de um sistema de classificação, são variáveis reais, isto é, exatas ("crisp"). Neste caso, procede-se inicialmente à "fuzificação" das entradas e posteriormente à "defuzificação" das saídas.

A "fuzzificação" de um variável exata $\hat{x} \in \mathbb{R}$, pode ser feita de diversas maneiras, como por exemplo pela função singleton, que mapeia o valor real $\hat{x}$ no conjunto difuso sobre o universo $\mathbb{R}$ dado por

$$singleton(\hat{x}) \; = \; \{(x, \mu(x)) \mid x = \hat{x} \; \textit{então} \; \mu(x) = 1, \; \textit{senão} \; \mu(x) = 0\}$$

A "defuzificação" de uma variável nebulosa de saída y também pode ser feita de diversas maneiras, como por exemplo pelo centro de gravidade, pela média dos máximos, etc. O problema de classificação com o modelo NefClass pode então ser agora definido.

Dada uma dimensão n (inteiro positivo não nulo), os padrões a serem classificados são vetores $\bar{x} = (\hat{x}_1, \hat{x}_2, , \cdots, \hat{x}_n) \in \mathbb{R}^n$. Supomos um conjunto C_i de m classes onde m é um número inteiro, positivo, maior que 1, com $i = 1, 2, \cdots, m$, tais que $C_i \subset \mathbb{R}^n$ e para todo j e k tais que $j, k \in \{1, 2, \cdots, m\}, C_j \cap C_k = \varnothing$. Além disto, $\bigcup_{i=1}^{m} C_i = \mathbb{R}^n$.

O classificador NefClass é definido por um conjunto de k regras difusas com um total de exatamente n entradas e m saídas. Os padrões são submetidos ao classificador depois de terem seus componentes devidamente "fuzificados" , isto é, as entradas fornecidas ao sistema são: $x_i = singleton(\hat{x}_i)$, onde $(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n)$ é um padrão a ser classificado.

As variáveis de saída y_i do conjunto de regras, com $i = 1, 2, \cdots, m$ são conjuntos difusos que têm como universo o intervalo [0,1] $\subset \mathbb{R}$. Como já mencionado anteriormente, as saídas y_i serão "defuzificadas" , produzindo cada uma delas um valor real $\hat{y}_i \in [0, 1] \in \mathbb{R}$.

O vetor de saídas "defuzificadas" $(\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n)$ será então interpretado, mapeando-se sua maior componente em 1 e todas as outras em 0. O sistema de inferência difusa, com a interpretação adotada para o vetor de saída é então uma aproximação da função (desconhecida) de classificação, $\psi : \mathbb{R} \rightarrow \{0, 1\}^m$.

7.4 Procedimento de Inferência

A seguir é mostrado como são calculadas as saídas y_i do sistema de inferência. Em princípio, os procedimentos empregados para calcular as saídas baseiam-se nos princípios do vínculo, na particularização, na conjunção e na projeção. Na prática, entretanto, os sistemas de inferência difusa podem ter mecanismos de dedução que não seguem estritamente as regras de inferência composicional baseadas nos princípios citados acima, mas podem ser baseados em modelos heurísticos.

Como exemplificação, é suposto um sistema com três variáveis de entrada, duas variáveis de saída e três regras difusas, ou seja, $n = 3$, $m = 2$ e $k = 3$. Podemos explicitar este exemplo nas regras abaixo:

$$R_1 : \textit{ if } x_1 = A_{11} \textit{ and } x_2 = A_{21} \textit{ and } x_3 = A_{31} \textit{ then } y_1 = B_1$$
$$R_2 : \textit{ if } x_1 = A_{12} \textit{ and } x_2 = A_{21} \textit{ and } x_3 = A_{31} \textit{ then } y_1 = B_2$$
$$R_3 : \textit{ if } x_1 = A_{13} \textit{ and } x_2 = A_{22} \textit{ and } x_3 = A_{32} \textit{ then } y_2 = B_3$$

Este sistema pode ser visto como uma rede de três camadas conforme apresentado na Figura 7.1.

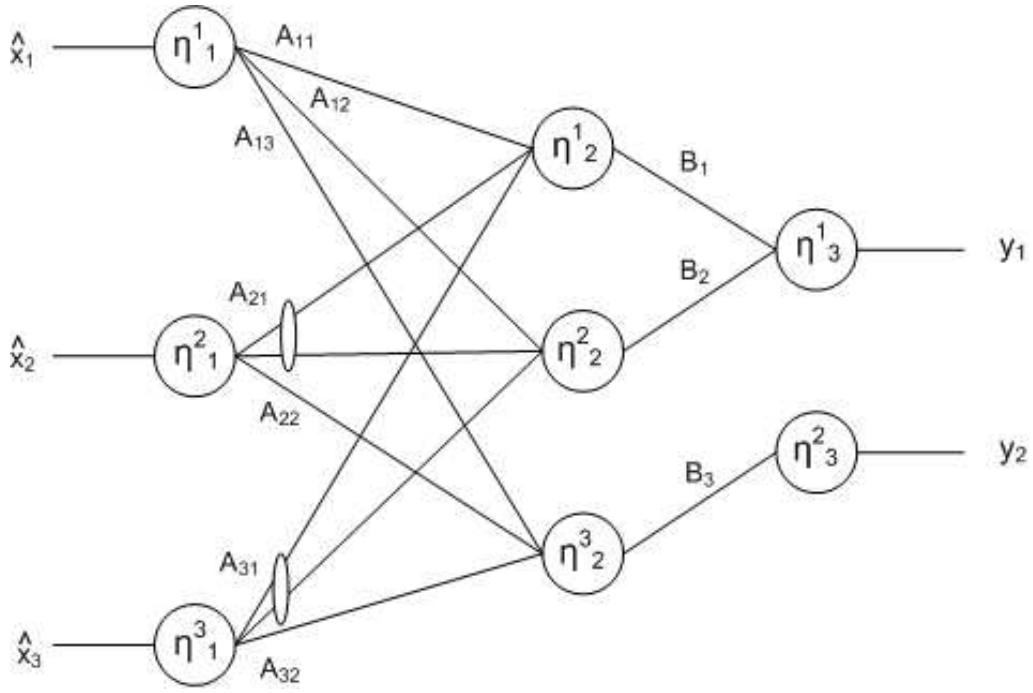

Figura 7.1: Modelo de Inferência em Três Camadas.

Para que possamos interpretar corretamente a estrutura difusa do NefClass torna-se importante identificar algumas restrições:

- Os pesos devem ser implementados de tal maneira, que para todos os termos lingüísticos idênticos sejam representados por conjuntos difusos idênticos (restrição representada no desenho pelas pequenas elipses).
- Não deve existir nenhuma regra com pesos difusos entrantes idênticos associadas a pesos de saída diferentes. Isto ajuda a estabelecer uma base de regras inconsistente.
- Em alguns casos, pode ser necessário que cada camada esteja conectada somente a uma unidade de saída.

As entradas dos nós da primeira camada são os valores exatos (crisp) da entrada $\bar{x}$:

$$input[\eta_1^i](x_i) = \hat{x}_i$$

Cada nó da primeira camada possui somente uma entrada. As saídas dos nós da primeira camada são dadas por:

$$output[\eta_1^i](x_i) = singleton(\hat{x}_i)$$

, onde i varia de 1 até n.

As entradas de cada nó da segunda camada dependem da saída dos nós da primeira camada e dos termos lingüistícos que rotulam os respectivos arcos, quando eles existem e são dadas por:

$$input_i[\eta_2^r](x_i) = \mu_{t_r^i}(\hat{x_i}) = \perp_{x_i}\{output[\eta_1^i](x_i), \mu_{t_r^i}(x_i)\}$$

onde r varia de 1 até k.

A saída dos nós da segunda camada é então dada por:

$$output[\eta_2^r] = \top_{i=1}^{n}\{input_i[\eta_2^r](x_i)\}$$

onde $\top$ é um operador norma-t.

As entradas de cada nó da terceira camada são ponderadas pelos termos lingüísticos dos conseqüentes das regras correspondentes e são dadas por:

$$input_r[\eta_3^j](y) = \top\{\mu_{B_r}, output[\eta_2^r]\} = \top_y\{\mu_{B_r}(y), output[\eta_2^r](y)\}$$

As saídas de cada nó da terceira camada são dadas por:

$$output[\eta_3^j](y) = \perp_{r=1}^{k}\{input_r[\eta_3^j](y)\}$$

onde j varia de 1 até m.

7.5 Algoritmo de Aprendizagem

O objetivo central de um algoritmo de aprendizagem é minimizar uma função de erro definido sobre a diferença entre a saída desejada e a saída atual. Entretanto, o algoritmo não executa nenhum tipo de otimização global mas computa somente modificações locais do peso. Os pesos são modificados de acordo com a informação local, distribuindo o erro global dentro da rede.

Assim, seja um sistema difuso (Mamdani) com n entradas, m saída e $k \leq k_{max}$ regras. Sejam s padrões a serem usados num treinamento supervisionado:

$$(p_1, c_1), (p_2, c_2), \cdots, (p_s, c_s)$$

onde $p_i = (\hat{p_1}, \hat{p_2}, \cdots, \hat{p_n})$ é um vetor de entrada e c_i é o índice que indica a classe à qual p_i pertence.

Além disto, supomos que, para cada entrada, seja definido um conjunto de termos linguísticos e os conjuntos difusos correspondentes:

$$t_j^i = (x_i, \mu_j^{(i)}(x_i))$$

O algoritmo a seguir constroi k regras para este sistema:

i. Selecionar o próximo padrão (p_i, c_i);

ii. Para todas as entradas x_i, selecione o termo linguístico $t_{j_s}^{(i)}$ tal que $\mu_{j_s}^{(i)}(\hat{p_i})$ = $max_j\{\mu_j^{(i)}(x_i)\}$;

iii. Se o número de nós (regras) k for menor que k_{max} e se para toda regra, $t_{js1}^{(1)}, t_{js2}^{(2)}$, $\cdots, t_{j_{sn'}}^{(n')}$ não são todos iguais aos selecionados no item (ii), então acrescentar ao nó de saída y_l tal que $c_i = l$;

iv. Repetir enquanto $k < k_{max}$ e ainda existirem padrões a serem selecionados;

O algoritmo de aprendizado das funções de pertinência do NefClass é um sistema que adopta um conjunto difuso de maneira cíclica, conforme a descrição abaixo:

i. Selecione o próximo padrão (p_i, c_i) propagando-o atraves do sistema NefClass para determinar o vetor de saída $\bar{\theta} = (\theta_1, \theta_2, \cdots, \theta_m)$;

ii. Para cada saída θ_i, determine $\delta_i = t_i - \theta_i$;

iii. Para cada regra R com $O_R > 0$;

(a) Determine delta δ_R:

$$O_{RR} = O_R.(1 - O_R)$$
$$\delta_R = O_{RR} \sum_{i=1}^{m} B_i.\delta_i$$

(b) Encontre o $x^{'}$ tal que:

$$\mu_R^{'}(x^{'}) = min_{i\in\{1,\cdots,n\}}\{\mu_R^{(i)}(x_i)\}$$

(c) Para cada conjunto difuso $\mu_R^{'}(x^{'})$ determine o delta para os parametros a, b e c usando uma taxa de aprendizado $\sigma > 0$;

$$\begin{aligned}
\delta_b &= \sigma.\delta_R.(c-a).sgn(x^{'} - b) \\
\delta_a &= -\sigma.\delta_R.(c-a) + \delta_b \\
\delta_c &= \sigma.\delta_R.(c-a) + \delta_b
\end{aligned}$$

Capítulo 8

Treinamento

8.1 Introdução

Neste capítulo será apresentada a estratégia utilizada para realizar o treinamento do detector, bem como os mecanismos automáticos de geração dos programas responsáveis pela inspeção da anomalia. Estes programas são utilizados para caracterizar a ameaça presente nos tráfegos de treinamento e validação.

Este conjunto de programas foram executados utilizando os dados de treinamento presente no experimento Darpa de 1998. No próximo capítulo, será observada a aplicação deste treinamento utilizando como dados de entrada o conjunto de dados de validação.

8.2 Esquema de Treinamento

Na Figura 8.1 é apresentado o esquemático de funcionamento da ferramenta utilizada para geração do banco de regras. O processo inicia-se informando o arquivo de dados representado pela extensão .dat que contém os valores das diferentes componentes descrevendo cada conjunto como anômalo e normal. Além disto, também deve ser informado o arquivo .ini, onde são descritas diversas informações necessárias ao funcionamento do modelo de treinamento, tais como: número de funções e suas respectivas for-

mas para as funções tanto de entrada como saída, definição de arquivos de gráficos (.gnu), regras (.nfs) e aprendizado (.log).

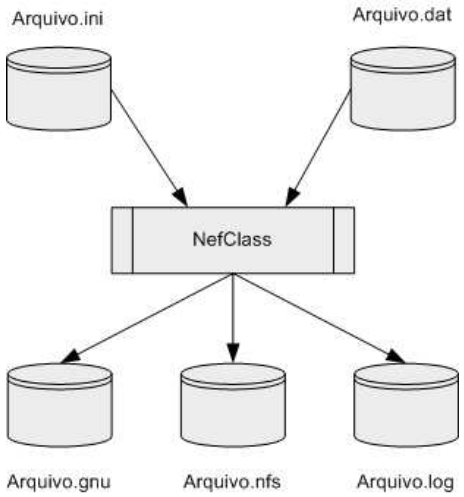

Figura 8.1: Geração do treinamento

8.3 Esquema da Geração de Regras

Na Figura 8.2 é apresentado um esquemático do ambiente utilizado para a geração automática do programa de investigação da anomalia. Este sistema inicia-se fornecendo o arquivo de regras (.nfs) proveniente da etapa de treinamento anteriormente mencionada combinada com a AI Perl library. em seguida é gerado um arquivo básico que descreve as regras nebulosas codificadas na linguagem Perl. Depois, este programa é integrado à base de dados, estando assim finalizado e disponível para uso dos especialistas avaliarem o processo de treinamento a partir dos resultados obtidos.

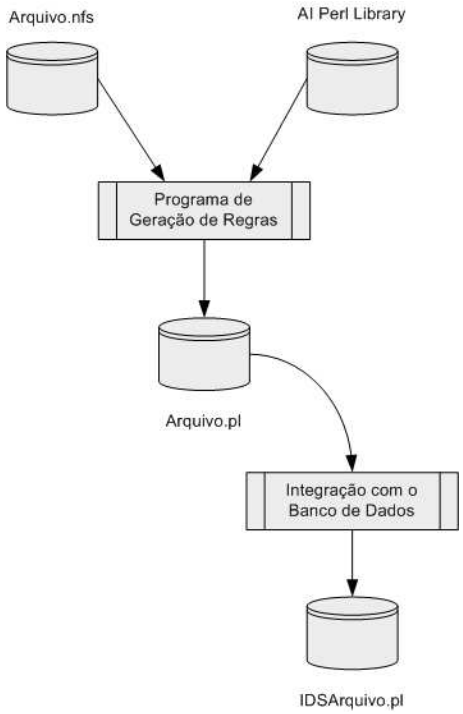

Figura 8.2: Geração automática do programa de investigação da anomalia

8.4 Ataque Convidado

8.4.1 Curva de Aprendizado

Na Figura 8.3 é apresentado o desempenho do treinamento. Como observado o modelo neuro-difuso não encontra nenhuma região de difícil convergência, mostrando um decaimento suave convergindo com poucas épocas de treinamento.

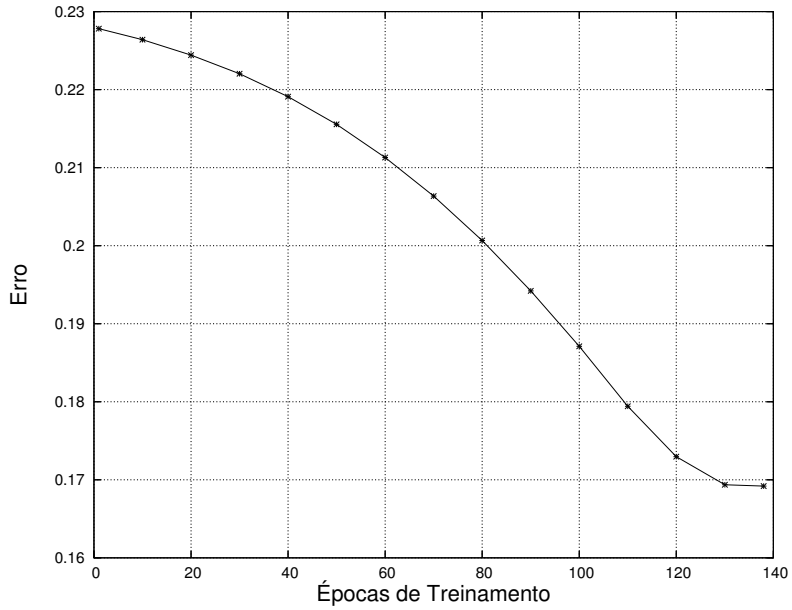

Figura 8.3: Curva de aprendizado neuro-difusa.

8.4.2 Curvas de Pertinência

As funções de pertinência utilizadas nas componentes TxGlobal e TxResets apresentadas nas Figuras 8.4 e 8.5 são da forma trapezoidal. Este perfil foi escolhido com o objetivo de estender o processo de captura do fenômeno observado. Já o perfil escolhido para mapear o espaço de saída é composto por uma combinação trapezoidal e triangular, o que permite um resultado melhor do processo de decteção.

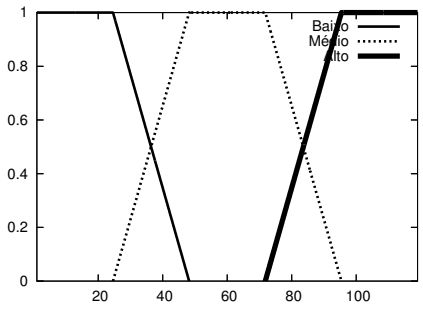

Figura 8.4: Pertinência da componente TxGlobal.

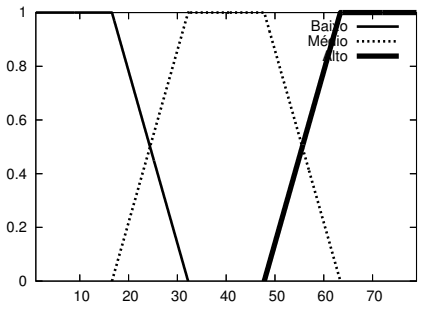

Figura 8.5: Pertinência da componente TxReset.

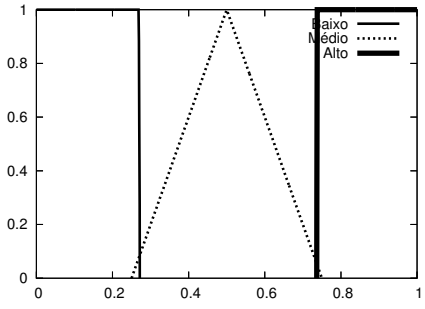

Figura 8.6: Saída do sistema neuro-difuso.

8.4.3 Regras Semânticas

O banco de conhecimento descrito nas regras semânticas produzidas a partir do mapeamento do espaço, apresentado na seção anterior, é apresentado na Tabela 8.1 onde as componentes são os antecedentes e o resultado o seu consequente em um sistema difuso do tipo Mamdani, onde a defuzificação é calculada pelo método do centro de gravidade (COG).

Tabela 8.1: Apresentação das regras semânticas

Número da Regra	Variável TxGlobal	Variável TxResets	Significado Final
1	Médio	Alto	Alto
2	Alto	Alto	Alto
3	Médio	Médio	Alto
4	Alto	Médio	Alto
5	Médio	Baixo	Baixo
6	Baixo	Baixo	Baixo

8.4.4 Distribuição Difusa da Decisão

Na Figura 8.7 é apresentado o resultado da decisão difusa para todas as sessões. Este cenário difuso permite identificar o grau de verdade mais adequado ao problema. Neste caso foi adotado o valor maior ou igual a 0.20, que é um grau de verdade significativo para indicar os elementos anômalos pois, conforme analisado, os valores abaixo deste patamar são considerados de baixa confiabilidade no julgamento do detector e assim descartados.

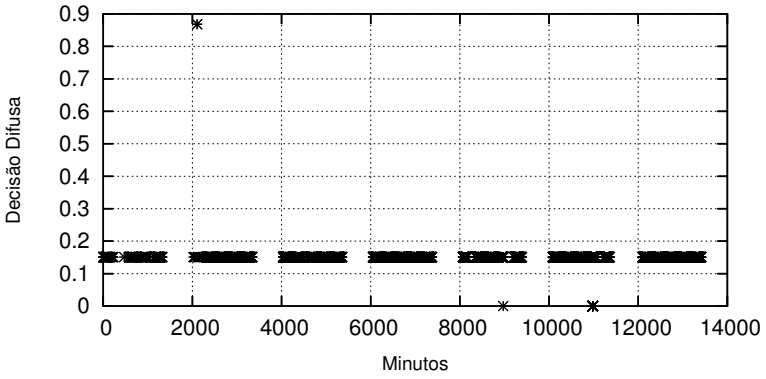

Figura 8.7: Distribuição da decisão difusa.

8.4.5 Resultados do Treinamento

Compreendidas as dificuldades discutidas na Seção-6.2, é apresentado na Tabela 8.2 o resultado do processo de treinamento.

O cenário no qual o detector modelo neuro-difuso foi treinado é expreso em 6725 unidades de tempo de cinco minutos chamadas de TAA, onde são observados apenas um único ataque convidado (TA). Assim, a diluição de ataques (DA) é de 0.0001.

Neste cenário, o detector conseguiu detectar com um percentual de acerto (PA) de cem por cento. Além disto não é observado nenhum erro, pois o percentual de erro (PE) é igual a zero, o que maximiza o percentual de acerto referencial (PAR).

Tabela 8.2: Resultados obtidos no processo de treinamento

Descrição	Resultados
Total de Amostras Analisadas (TAA)	6725
Total de Ataques (TA)	1
Diluição de Ataques (DA)	0.0001
Total de Falso Positivos (FP)	0
Total de Falso Negativo (FN)	0
Total de Verdadeiramente Positivo (VP)	1
Total de Verdadeiramente Negativo (VN)	6724
% de Acerto (PA = (VP + VN)/TAA)	100 %
% de Erro (PE = 100 - PA)	0 %
% de Acerto Referencial (PAR = VP/TA)	100 %

8.5 Ataque Netuno

8.5.1 Curva de Aprendizado

Na Figura 8.8 é apresentado o desempenho do treinamento para o ataque Netuno. Como pode ser observado, o treinamento do modelo neuro-difuso apresenta uma queda abrupta associada a uma longa cauda de duração em termos de épocas de treinamento, indicando dificuldades de convergência. Tais dificuldades estão associadas as regiões de difícil seletividade presentes nas componentes que caracterizam este ataque, conforme apresentado na Seção 6.7.1.

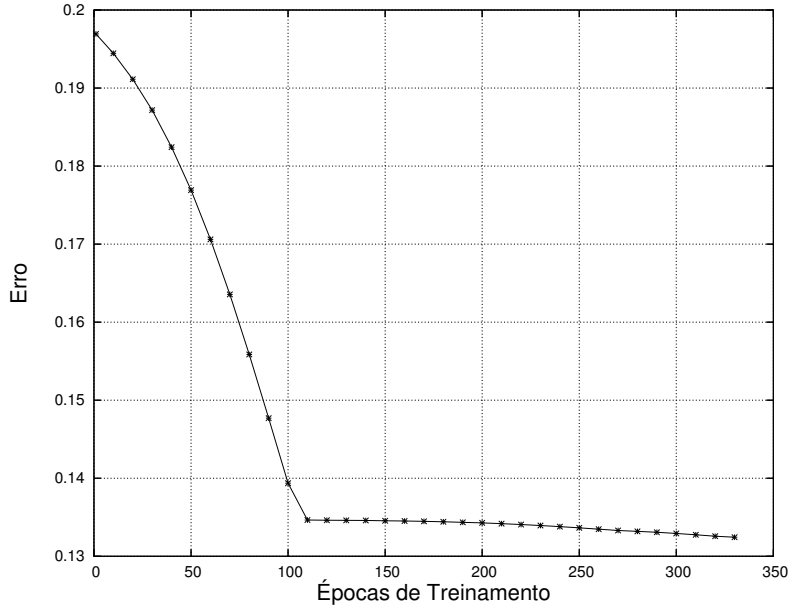

Figura 8.8: Curva de aprendizado neuro-difusa.

8.5.2 Curvas de Pertinência

As funções de pertinência utilizadas para as componentes FSR_A, FSR_B, FSR_A2B, FSR_B2A e BPP, são apresentadas nas Figuras-8.9, 8.10, 8.11, 8.12 e 8.13. Todas são da forma trapezoidal combinadas com triangular, o que permitiu um melhor resultado no processo de detecção.

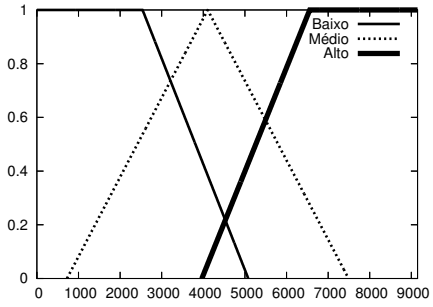

Figura 8.9: Pertinência da componente FSR_A.

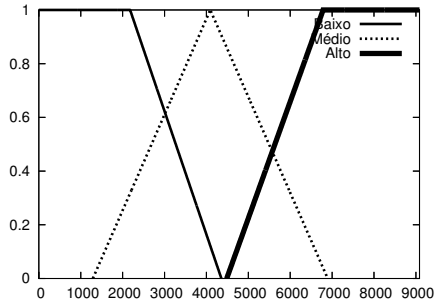

Figura 8.10: Pertinência da componente FSR_B.

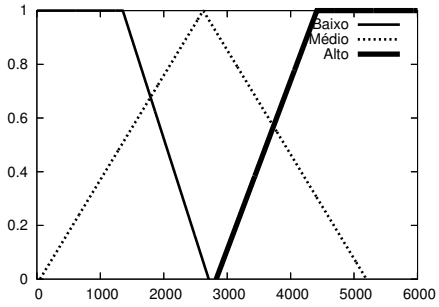

Figura 8.11: Pertinência da componente FSR_A2B.

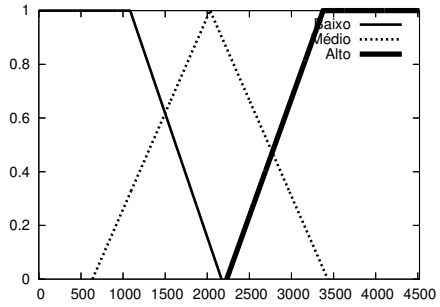

Figura 8.12: Pertinência da componente FSR_B2A.

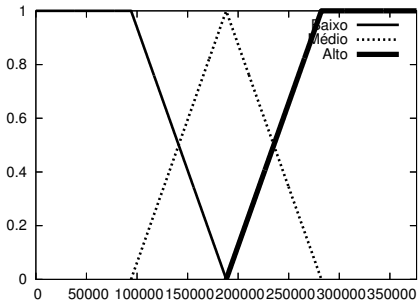

Figura 8.13: Pertinência da componente BPP.

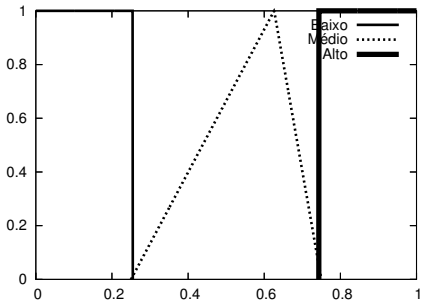

Figura 8.14: Saída do sistema neuro-difuso.

8.5.3 Regras Semânticas

O banco de conhecimento descrito nas regras semânticas produzidas a partir do mapeamento do espaço, é apresentado na Tabela 8.3.

Tabela 8.3: Apresentação das regras semânticas

Número da Regra	Variável FSR_A	Variável FSR_B	Variável FSR_A2B	Variável FSR_B2A	Variável BPP	Significado Final
1	Baixo	Baixo	Baixo	Baixo	Baixo	Baixo
2	Baixo	Baixo	Médio	Baixo	Baixo	Alto
3	Médio	Baixo	Médio	Baixo	Baixo	Alto
4	Médio	Baixo	Alto	Baixo	Baixo	Alto
5	Alto	Baixo	Alto	Baixo	Baixo	Alto
6	Médio	Médio	Baixo	Médio	Baixo	Alto
7	Alto	Alto	Médio	Alto	Baixo	Alto
8	Médio	Médio	Médio	Médio	Baixo	Alto
9	Médio	Baixo	Baixo	Baixo	Baixo	Médio
10	Alto	Médio	Alto	Médio	Baixo	Baixo
11	Baixo	Baixo	Baixo	Baixo	Alto	Baixo
12	Alto	Alto	Alto	Alto	Baixo	Baixo

8.5.4 Distribuição Difusa da Decisão

Na Figura 8.15 é apresentado o resultado da decisão difusa para todas as sessões. Este cenário difuso permite identificar o grau de verdade mais adequado ao problema e neste caso foi adotado o valor maior ou igual a 0.13. Este grau de verdade é significativo para indicar os elementos anômalos pois conforme analisado, os valores abaixo deste patamar são considerados de baixa confiabilidade no julgamento do detector e assim descartados.

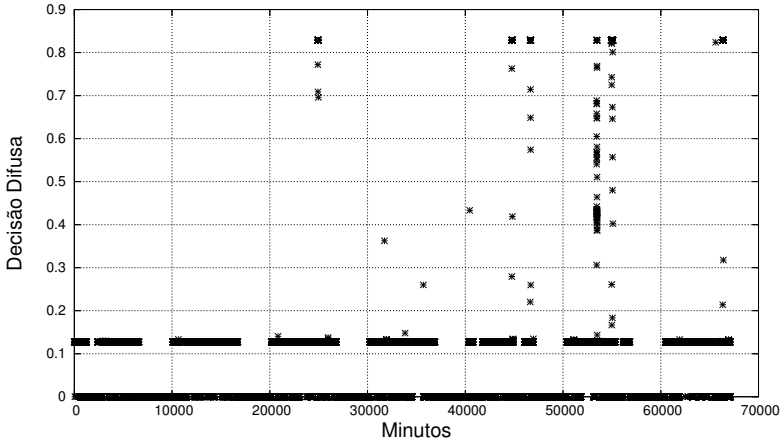

Figura 8.15: Distribuição da decisão difusa.

8.5.5 Resultados do Treinamento

É apresentado na Tabela 8.4 o resultado do processo de treinamento para o ataque Netuno. O cenário de treinamento no qual o detector neuro-difuso foi treinado é expresso em 29381 unidades de tempo de um minuto, chamadas de TAA, onde são observadas a presença de 432 ataques (TA) assim a diluição de ataques é de 0.0147.

Neste cenário, o detector conseguiu obter um percentual de acerto (PA) de 99.867 %. Além disto, é observado um percentual de erro (PE) de 0.132 %, onde o percentual de acerto de 93.75 % (PAR), mostrando um treinamento adequado.

Tabela 8.4: Resultados obtidos no processo de treinamento

Descrição	Resultados
Total de Amostras Analisadas (TAA)	29381
Total de Ataques (TA)	432
Diluição de Ataques	0.0147
Total de Falso Positivos (FP)	12
Total de Falso Negativo (FN)	27
Total de Verdadeiramente Positivo (VP)	405
Total de Verdadeiramente Negativo (VN)	28937
% de Acerto (PA = (VP + VN)/TAA)	99.867 %
% de Erro (PE = 100 - PA)	0.132 %
% de Acerto Referencial (PAR = VP/TA)	93.75 %

8.6 Ataque Varredura de Portas

Após o treinamento neuro-difuso inicial o ataque varredura de portas apresentou como dificuldade patamares de graus de verdade incapazes de capturar a anomalia. Entretanto a partir do comportamento verificado na Figura 8.16, foram verificado patamares claros que evidenciam a presença da anomalia em camadas. A ordenada foi então organizada em quatro faixas conforme apresentado na Tabela 8.5.

Tabela 8.5: Apresentação dos limites das faixas de SDP utilizadas

Faixa	Limite Inferior	Limite Superior
1	0	< 50
2	≥ 50	< 100
3	≥ 100	< 150
4	≥ 150	< 300

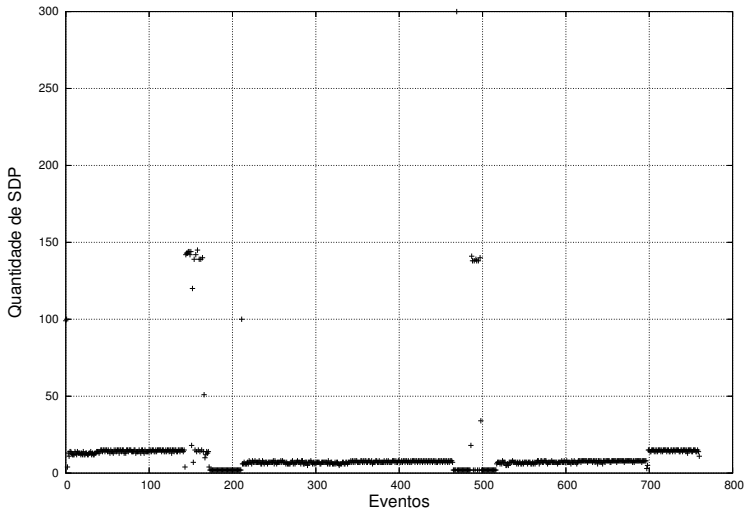

Figura 8.16: Descrição das fases difusas.

8.6.1 Curva de Aprendizado

Nas Figuras 8.17, 8.18, 8.19 e 8.20 são apresentados os desempenhos dos treinamentos das quatro faixas operacionais da solução. Como comportamento comum a todas as regiões é observado uma queda abrupta seguida de um cauda longa de épocas de treinamento. A fase 1 apresenta um comportamento de convergência diferenciado das demais, este comportamento é explicado pela natureza dos valores das componentes presentes nesta região, onde a diferenciação mostra-se mais difícil. Para todas as faixas, porém, o número de épocas de treinamento utilizado foi o mesmo.

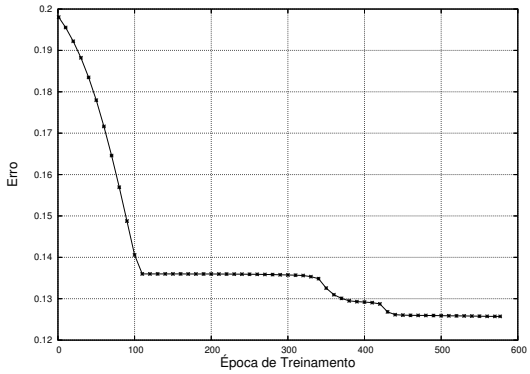

Figura 8.17: Aprendizado da fase 1

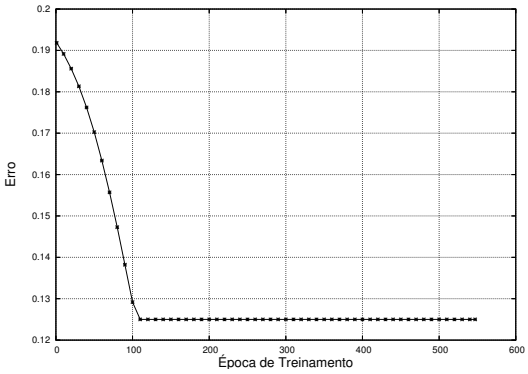

Figura 8.18: Aprendizado da fase 2

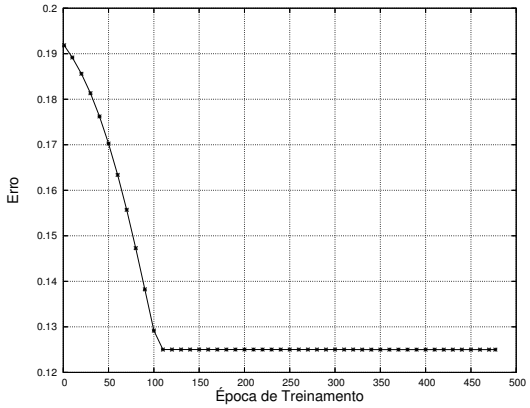

Figura 8.19: Aprendizado da fase 3

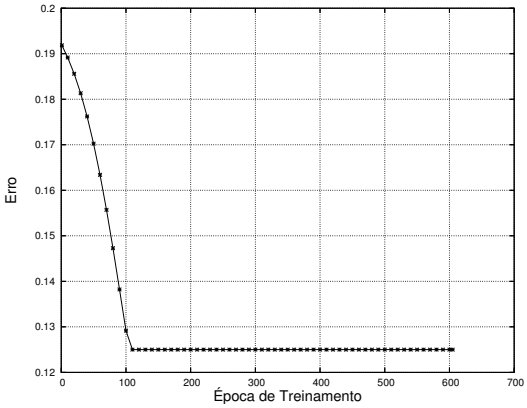

Figura 8.20: Aprendizado da fase 4

8.6.2 Curvas de Pertinência

Nesta seção são apresentadas as funções de pertinência utilizadas para as componentes MBPP e SDPU em todas as fases. A forma trapezoidal combinada com triangular foi escolhida por permitir um melhor resultado no processo de detecção da anomalia.

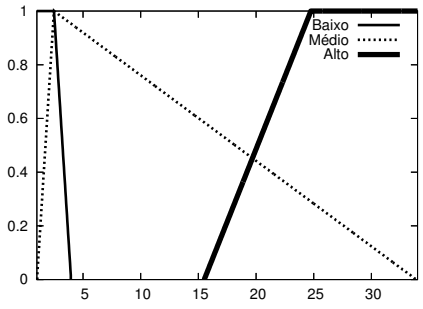

Figura 8.21: Curva de Pertinência MBPP (Fase 1)

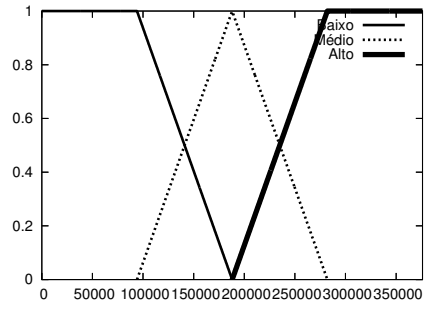

Figura 8.22: Curva de Pertinência SDPU (Fase 1)

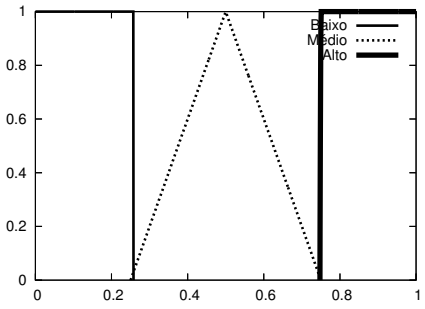

Figura 8.23: Saída do sistema neuro-difuso (Fase 1)

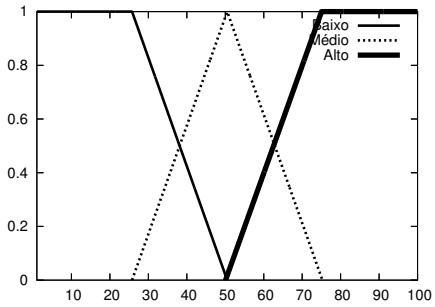

Figura 8.24: Curva de Pertinência MBPP (Fase 2)

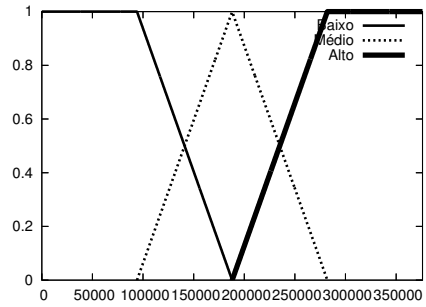

Figura 8.25: Curva de Pertinência SDPU (Fase 2)

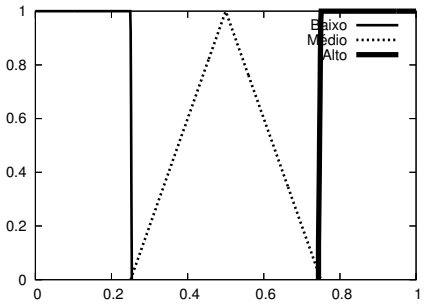

Figura 8.26: Saída do sistema neuro-difuso (Fase 2)

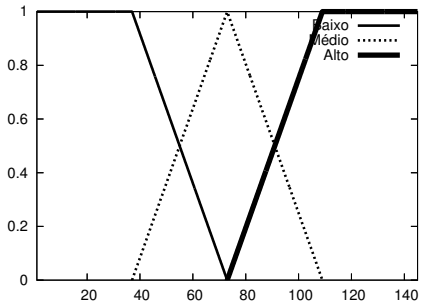

Figura 8.27: Curva de Pertinência MBPP (Fase 3)

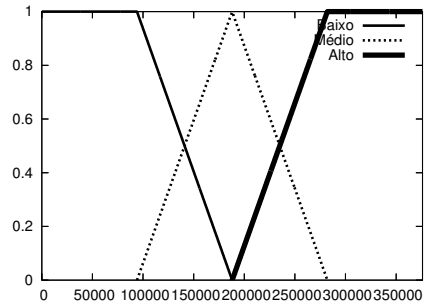

Figura 8.28: Curva de Pertinência SDPU (Fase 3)

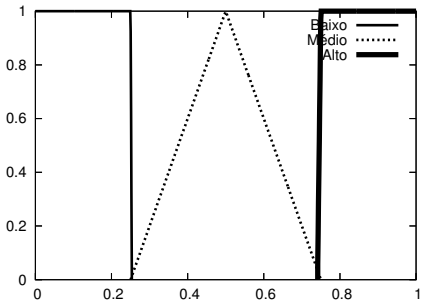

Figura 8.29: Saída do sistema neuro-difuso (Fase 3)

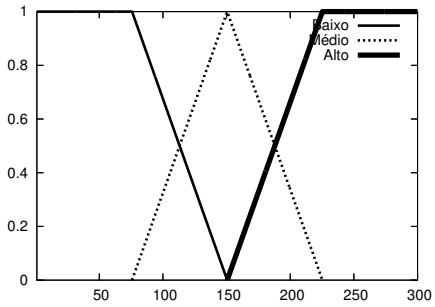

Figura 8.30: Curva de Pertinência MBPP (Fase 4)

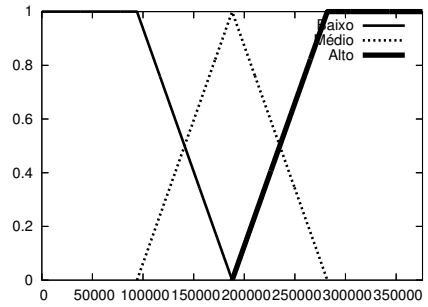

Figura 8.31: Curva de Pertinência SDPU (Fase 4)

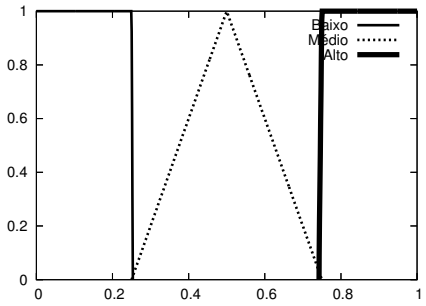

Figura 8.32: Saída do sistema neuro-difuso (Fase 4)

8.6.3 Regras Semânticas

O banco de conhecimento descrito nas regras semânticas produzidas a partir do mapeamento do espaço, é apresentado nas Tabelas 8.6, 8.7, 8.8 e 8.9. Como pode ser observado, o número de regras nas fases 1 e 2 são maiores que nas fases 3 e 4, tal comportamento é explicado devido à natureza de diferenciação mais complexa conforme apresentado no seu mapeamento.

Tabela 8.6: Regras semânticas da Fase 1

Regra	MBPP	SDPU	Final
1	Baixo	Baixo	Baixo
2	Médio	Baixo	Alto
3	Alto	Baixo	Alto
4	Baixo	Alto	Baixo

Tabela 8.7: Regras semânticas da Fase 2

Regra	MBPP	SDPU	Final
1	Baixo	Baixo	Baixo
2	Baixo	Alto	Baixo
3	Alto	Baixo	Alto
4	Médio	Baixo	Baixo

Tabela 8.8: Regras semânticas da Fase 3

Regra	MBPP	SDPU	Final
1	Alto	Baixo	Alto
2	Baixo	Baixo	Baixo
3	Baixo	Alto	Baixo

Tabela 8.9: Regras semânticas da Fase 4

Regra	MBPP	SDPU	Final
1	Alto	Baixo	Alto
2	Baixo	Baixo	Baixo
3	Baixo	Alto	Baixo

8.6.4 Distribuição Difusa da Decisão

Nas Figuras 8.33, 8.33, 8.33 e 8.33 é apresentado o resultado da decisão difusa para todas as sessões presentes em todas as fases. Este cenário difuso permite identificar o grau de verdade mais adequado como limiar para o problema, que neste caso foi de 0.20 utilizado para todas as fases. Este grau de verdade é significativo para indicar os elementos anômalos pois conforme analisado, os valores abaixo deste patamar são considerados de baixa confiabilidade no julgamento do detector e assim descartados.

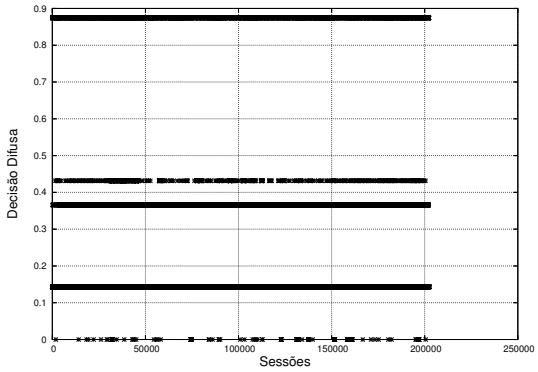

Figura 8.33: Distribuição da decisão neuro-difusa (Fase 1).

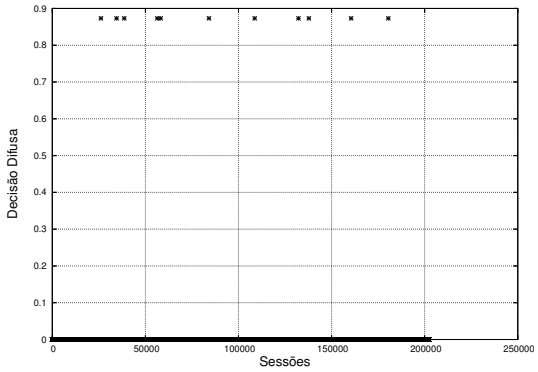

Figura 8.34: Distribuição da decisão neuro-difusa (Fase 2).

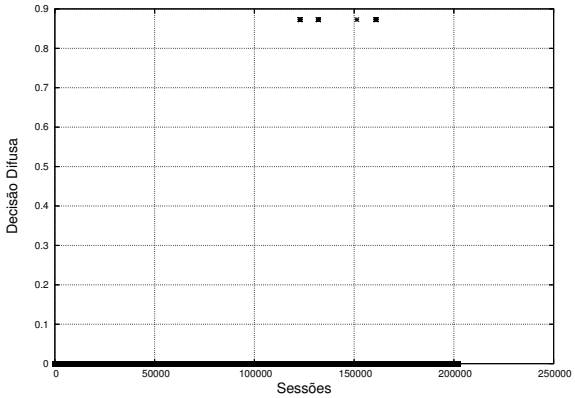

Figura 8.35: Distribuição da decisão neuro-difusa (Fase 3).

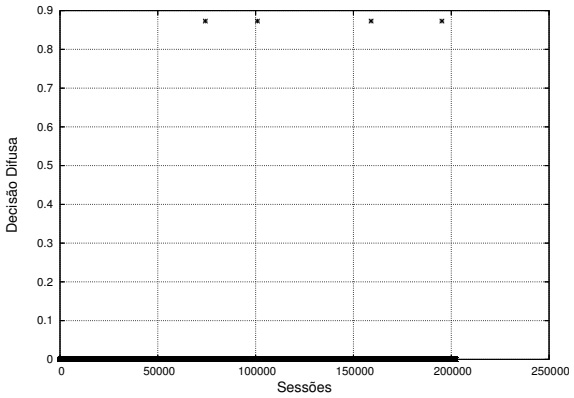

Figura 8.36: Distribuição da decisão neuro-difusa (Fase 4).

8.6.5 Resultados do Treinamento

Na Tabela 8.10, é apresentado o resultado do processo de treinamento. O cenário de treinamento no qual o detector neuro-difuso foi treinado é expresso em 6725 unidades de tempo de cinco minuto, chamadas de TAA, onde são observadas a presença de 843 ataques (TA) assim a diluição de ataques é de 0.1254.

Neste cenário o detector conseguiu obter um percentual de acerto (PA) de 37.28 %. Além disto, é observado um percentual de erro (PE) de 62.72 %, onde o percentual de acerto de 97.98 % (PAR), mostrando um treinamento adequado.

Apesar do elevado índice de erro (PE) o detector foi considerado adequado pois o acerto referencial (PAR) é elevado mostrando sua eficiência na identificação do ataque. Entretanto, o detector apresentou um elevado índice de falsos positivos (FP). Este comportamento se deve basicamente a natureza polimórfica do ataque descrito na seção 4.2 bem como ao comportamento normal de alguns tráfegos observados na rede de referência.

Tabela 8.10: Resultados obtidos no processo de treinamento

Descrição	Resultados
Total de Amostras Analisadas (TAA)	6725
Total de Ataques (TA)	843
Diluição de Ataques	0.1254
Total de Falso Positivos (FP)	4201
Total de Falso Negativo (FN)	17
Total de Verdadeiramente Positivo (VP)	826
Total de Verdadeiramente Negativo (VN)	1681
% de Acerto (PA = (VP + VN)/TAA)	37.28 %
% de Erro (PE = 100 - PA)	62.72 %
% de Acerto Referencial (PAR = VP/TA)	97.98 %

Capítulo 9

Resultados

9.1 Introdução

Neste capítulo são apresentados os resultados obtidos a partir da investigação da presença de anomalias nos dados de validação DARPA 99. Este processo é baseado na experiência contida nos dados de treinamento apresentados no capítulo 8.

9.2 Ataque Convidado

Na Tabela 9.1 é apresentado o resultado do processo de avaliação do ataque convidado. O cenário no qual o detector neuro-difuso procedeu suas análises é expresso em 1580 unidades de tempo de cinco minutos, chamadas de TAA, onde são observadas a presença de 8 ataques (TA), assim a diluição de ataques nas amostras é de apenas 0.0051.

Neste cenário, o detector conseguiu obter um percentual de acerto (PA) de 99.3038 %. Associadamente é observado um percentual de erro (PE) de 0.6962 %, onde o percentual de acerto de 25 % (PAR).

Estes resultados apresentam um claro deslizamento entre o treinamento e a dinâmica dos ataques presentes nos dados de validação. Esta diferença realça diversas dificuldades presentes no treinamento, sendo a mais significativa o equívoco em construir apenas ele-

vadas quantidades de eventos sem no entanto considerar a variabilidade da dinâmica do ataque de maneira precisa. Além disto a questão do critério de contagem no tempo apresentada na Seção 6.4 também é apontado como um problema na contagem da quantidade de ataques presente na base analisada.

Tabela 9.1: Resultados de validação do ataque convidado

Descrição	Resultados
Total de Amostras Analisadas (TAA)	1580
Total de Ataques (TA)	8
Diluição de Ataques (DA)	0.0051
Total de Falso Positivos (FP)	5
Total de Falso Negativo (FN)	6
Total de Verdadeiramente Positivo (VP)	2
Total de Verdadeiramente Negativo (VN)	1567
% de Acerto (PA = (VP + VN)/TAA)	99.3038 %
% de Erro (PE = 100 - PA)	0.6962 %
% de Acerto Referencial (PAR = VP/TA)	25 %

9.3 Ataque Netuno

Na Tabela 9.2, é apresentado o resultado do processo de avaliação do ataque netuno. O cenário no qual o detector neuro-difuso procedeu suas análises é expresso em 10163 unidades de tempo de um minuto, chamadas de TAA, onde são observadas a presença de 26 ataques (TA), assim a diluição de ataques nas amostras é de apenas 0.0026.

Neste cenário, o detector conseguiu obter um percentual de acerto (PA) de 94.6472 %. Associadamente é observado um percentual de erro (PE) de 5.3528 %, onde o percentual de acerto de 88.4615 % (PAR).

Estes resultados apresentam excelentes índices de acerto, em um cenário onde a quantidade de falsos positivos é baixa, mostrando que as componentes escolhidas são adequadas à dinâmica de ataque presente nos dados de avaliação.

Tabela 9.2: Resultados de validação do ataque netuno

Descrição	Resultados
Total de Amostras Analisadas (TAA)	10163
Total de Ataques (TA)	26
Diluição de Ataques (DA)	0.0026
Total de Falso Positivos (FP)	541
Total de Falso Negativo (FN)	3
Total de Verdadeiramente Positivo (VP)	23
Total de Verdadeiramente Negativo (VN)	9596
% de Acerto (PA = (VP + VN)/TAA)	94.6472 %
% de Erro (PE = 100 - PA)	5.3528 %
% de Acerto Referencial (PAR = VP/TA)	88.4615 %

9.4 Ataque Varredura de Portas

Na Tabela 9.3, é apresentado o resultado do processo de avaliação do ataque varredura de portas. O cenário no qual o detector neuro-difuso procedeu suas análises é expresso em 2470 unidades de tempo de cinco minuto, chamadas de TAA, onde são observadas a presença de 19 ataques (TA), assim a diluição de ataques nas amostras é de apenas 0.0077.

Neste cenário, o detector conseguiu obter um percentual de acerto (PA) de 42,8340 %. Associadamente é observado um percentual de erro (PE) de 57,1660 %, onde o percentual de acerto de 100 % (PAR).

Estes resultados apresentam excelentes índices de acerto, em um cenário onde a quantidade de falsos positivos é significativamente elevada, mostrando que as componentes escolhidas são adequadas para o recenhecimento porém em muitos cenários o mesmo se confunde com os tráfegos considerados normais, sendo necessário no futuro adicionar componentes que permitam diferenciar melhor o não ataque.

Tabela 9.3: Resultados de validação do ataque varredura

Descrição	Resultados
Total de Amostras Analisadas (TAA)	2470
Total de Ataques (TA)	19
Diluição de Ataques (DA)	0.0077
Total de Falso Positivos (FP)	1412
Total de Falso Negativo (FN)	0
Total de Verdadeiramente Positivo (VP)	19
Total de Verdadeiramente Negativo (VN)	1039
% de Acerto (PA = (VP + VN)/TAA)	42.8340 %
% de Erro (PE = 100 - PA)	57.1660 %
% de Acerto Referencial (PAR = VP/TA)	100 %

Capítulo 10

Conclusão

A principal conclusão do trabalho foi comprovar a aplicabilidade da abordagem neuro-difusa na investigação das ameaças presentes nos tráfegos de rede. Observamos entretanto que deverão ser realizados estudos complementares para reavaliação de algumas premissas utilizadas para a escolha de algumas componentes presentes em ataques onde o detector apresentou elevados índices de falsos positivos ou baixa taxa de detecção.

Como observado nos cenários finais obtidos para alguns ataques os resultados não foram exatamente os esperados. Tais dificuldades estão relacionadas a diversos fatores, alguns de caráter específico e outros de caráter geral.

Os problemas de caráter geral, citados no Capítulo 3, são devidos a diversas falhas presentes no desenvolvimento do experimento DARPA. Estas falhas prejudicaram o resultado do trabalho, sendo a de maior relevância relacionado ao processo de marcação dos eventos de ataques. Este processo apresentou dificuldades significativas de correlação com a base de tráfego, possibilitando a inserção de erros na base de dados tanto de treinamento como validação. Tais erros influenciam todo o processo, possibilitando a não marcação de tráfegos onde a evidência de ataques é concreta bem como a marcação errônea de tráfegos considerados normais como ataque.

Os problemas de caráter específicos estão relacionados aos limites detectados nos dados de treinamento que influenciaram de maneira direta os resultados obtidos em alguns ataques. Tais problemas podem ser observados no ataque varredura de portas onde a

componente MBPP apresenta um ponto entre 350.000 e 400.000 (Figura 6.18) sendo a variabilidade compreendida entre 0 e 5.000 para a maioria dos eventos. De modo similar foi observado o mesmo comportamento nos limites das componentes MBPP (Figura 6.7) e FSR_A (Figura 6.9) do ataque netuno.

O resultado alcançado para o ataque convidado mostrou-se pouco satisfatório, uma vez que o percentual de acerto em relação ao tráfego real é de apenas 25 %. Apesar deste baixo índice de acerto o detector mostrou-se eficiente ao avaliar dentro dos diversos perfis de tráfego o que não é anômalo dentro do espaço de amostras de ataque totalizando 99.3038 % de acerto. Adicionalmente, apresentamos como característica animadora o baixo índice de falsos negativos (FN) mostrando que o detector não indicou como ataque tráfegos verdadeiramente sem anomalias. Assim, devido as questões já abordadas em detalhes na Seção 9.2, acreditamos ser necessário apenas uma reavaliação das componentes ou até mesmo a inserção de uma ou mais componentes com o objetivo de aumentar a taxa de captura do ataque.

O resultado do ataque netuno é considerado satisfatório uma vez que este apresentou um percentual de 88.4615% de acerto real, em um cenário de acerto de 94.6472 % indicando um elevado índice de eficiência constatada pela quantidade de acerto dos tráfegos que verdadeiramento não continham a anomalia investigada.

O resultado do ataque verredura de porta, apesar de apresentar um acerto referencial de 100 %, indica um cenário global de desempenho pobre, pois o acerto efetivo foi de apenas 42.8340 % associado a um elevado índice de erro de avaliação dos tráfegos considerados anômalos. Aprofundando a análise dos tráfegos equivocadamente classificados como anômalos, foi observada uma grande dificuldade de captura, pois estes tráfegos realmente não continham nenhuma evidência de ameaça. Nestes tráfegos, observamos a influência do vetor de característica MBPP, capturando evidências tanto de ataque como de não ataque.

10.1 Trabalhos Futuros

10.1.1 Dissolução das Fronteiras

Uma proposta para estudos posteriores é a elaboração de mecanismos que permitam a eliminação das fronteiras rígidas de controle de ameaças hoje baseadas exclusivamente em firewalls. Estes mecanismos são centradas apenas na investigação dos cabeçalhos dos protocolos sem a visibilidade da dinâmica de ataques presentes em toda a infra-estrutura.

10.1.2 Detecção de Anomalias

Uma questão importante presente no modelo adaptativo proposto neste trabalho é compreender que as ameaças presentes nos tráfegos de rede estão sempre em constante mutação.

A evolução destes ataques podem para algumas ameaças, já mapeadas afetar, a performace de detecção apresentada anteriormente. Esta perda de performace está centrada no fato que novos perfis de tráfego podem alterar o que anteriormente era fácilmente identificável pela utilização de uma característica específica.

Assim é proposto a introdução de mecanismos automáticos que alertem aos especialistas sobre a evolução do deslizamento das ameaças em relação ao atual modelo de captura. Para tanto é sugerida a utilização de mecanismos KNN (k-Nearest Neighbor). Esta abordagem terá como objetivo capturar um determinado ataque no qual o detector não encontra uma classificação apropriada, transpondo assim a sua decisão para as fronteiras já conhecidas.

Neste processo será armazenado para futuras avaliações todo o cenário capturado permitindo assim o estabelecimento de um novo ambiente de treinamento.

Deverão ser adicionadas ao modelo geral da arquitetura de detecção mecanismos que indiquem de maneira mais automática qual o conjunto de componentes mais adequada ao problema da detecção de um determinado ataque. Assim é sugerido estudos que integrem ao atual modelo as técnicas PCD (principal component decomposition).

10.1.3 Distribuição dos Agentes Coletores

Objetivando uma visão global do cenário geral de ameaças é proposto a utilização de detectores locais distribuídos por diferentes infra-estruturas. Estes detectores teriam ação local porém seriam providos mecanismos de comunicação com um ou mais agentes responsáveis pela decisão global. Com esta arquitetura seria possível detectar estratégias globais articuladas com objetivos de desabilitar um ou mais serviços presentes em um organiazação.

Referências Bibliográficas

[1] ANDERSON, J. P. Computer Security Threat Monitoring and Surveillance Em *http://csrc.nist.gov/publications/history/ande80.pdf* (acessado em 16 de março de 2005).

[2] SHIREY, R. Internet Security Glossary Em *http://www.ietf.org/rfc/rfc2828.txt* (acessado em 16 de março de 2005).

[3] CERT COORDINATION CENTER CERT/CC Statistics Em *http://www.cert.org/stats/cert_stats.html* (acessado em 16 de março de 2005).

[4] LIPSON, HOWARD F. Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues Em *www.cert.org/archive/pdf/02sr009.pdf* (acessado em 16 de março de 2005).

[5] THOTTAN, MARINA., JI, CHUANYI Anomaly Detection in IP Networks Em *IEEE Transaction on Signal Processing* (agosto de 2003) Volume 51, Numero 8, Páginas 2191-2204

[6] LELAND, W. E., TAQQU, M., S. On the self-similar nature of ethernet traffic Em *IEEE/ACM Transaction Network* (fevereivo de 1994), Volume 2, Páginas 1-15

[7] COASTES M., HERO, A., R. N, E YU, B. Internet Tomography Em *IEEE Transaction on Signal Processing* (maio de 2002), Volume 19, Páginas 47-56

[8] BARFORD, P., PLONKA, D. Characteristics of network traffic flow anomalies Em *Proccedings of the First ACM SIGCOMM Workshop on Internet Measurement* (2001), Páginas 69-73

[9] ZANERO, S., SAVARESI, S. M. Unsupervised learning techniques for an intrusion detection system Em *Proccedings of the ACM symposium on Applied computing* (março de 2004), Páginas 412-419

[10] ESTAN, C., SAVAGE, S., VARGHESE, G. Automatically inferring pattern of resource consumption in network traffic Em *Proccedings of the conference on Applications, technologies, architectures, and protocols for computer communications*, (2003), Páginas 137-148

[11] LAWRENCE R. HALME, KENNETH R. BAUER. A taxonomy of anti-intrusion techniques Em *Proceedings of the 18th National Information Systems Security Conference* (outubro de 1995), Páginas 163-172

[12] DOROTHY, E. DENNING. An Intrusion-Detection Model Em *IEEE Transactions on Software Engineering* (fevereiro de 1987), Volume 13, Número 2, Páginas 222-232

[13] T. J. PROCYK, E. H. MANDANI. A linguistic self-organizing process controller Em *Automatica* (1979), Volume 15, Páginas 15-30

[14] WU ZHI QIAO, WANG PEI ZHUANG, TEH HOON HENG, SONG SHOU SHAN. A rule self-regulating fuzzy controller. Em *Fuzzy Sets and Systems* (abril de 1992), Volume 47, Páginas 13-21

[15] DETLEF NAUCK, RUDOLF KRUSE. What are Neuro-fuzzy Classifiers ? Em *7th International Fuzzy Systems Association World Congress IFSA 97* (1997), Páginas 228-233

[16] PETRI VUORIMA Fuzzy self-organizing map. Em *Fuzzy Sets and Systems* (setembro de 1994), Volume 66, Páginas 223-231

[17] DETLEF NAUCK, RUDOLF KRUSE. NEFCLASS - A neuro-fuzzy approach for classification of data. Em *ACM Symposium on Applied Computing* (fevereiro de 1995), Páginas 461-465

[18] EKLUND, P., KLAWONN F., NAUCK, D. D. Distributing Errors in Neural Fuzzy Control. Em *Fuzzy Logic and Neural Networks* (julho de 1992), Páginas 1139-1142.

[19] KOSKO, B. Neural Networks and Fuzzy Systems, 2^a ed. Prentice-Hall. ISBN 0136114350.

[20] JYH-SHING ROBERT JANG, CHUEN-TSAI SUN Neuro-Fuzzy Modeling and Control Em *Proceedings of the IEEE* (abril de 1999), Volume 83, Páginas 378-405

[21] MAMDANI, E.H., S. ASSILIAN An experiment in linguistic synthesis with a fuzzy logic controller Em *International Journal of Man-Machine Studies* (1975), Volume 7, Número 1, Páginas 1-13

[22] TAKAGI, T., SUGENO, M Fuzzy Identification of Systems and its Applications to Modeling and Control Em *IEEE Transactions on Systems Man and Cybernetics* (1985), Volume 15, Número 1, Páginas 116-132

[23] Y. TSUKAMOTO An Approach to Fuzzy Reasoning Method Em *Readings in Fuzzy Sets for Intelligent Systems* (1993), Páginas 523-529

[24] ANDREAS WESPI, HERVÉ DEBAR Building an Intrusion-Detection System to Detect Suspicious Process Behavior Em *Proceedings of RAID 99, Workshop on Recent Advances in Intrusion Detection* (setembro de 1999)

[25] COOPER, G. F. The computational complexity of probabilistic inference using bayesian belif networks Em *Artifitial Intelligence* (1990), Número 42, Páginas 393-405

[26] JAMES CANNADY Artificial Neural Networks for Misuse Detection Em *Proceedings of the 1998 National Information Systems Security Conference (NISSC 98)* (1998), Páginas 443-456

[27] JOHN MCHUGH Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincon Laboratory Em *ACM Transactions on Information and System Security* (novembro de 2000), Volume 3, Número 4, Páginas 262-294

[28] NICHOLAS PUKETZA, MANDY CHUNG, RONALD A. OLSSON A Software Plataform for Testing Intrusion Detection Systems Em *IEEE Software* (setembro de 1997), Volume 14, Número 5, Páginas 43-51

[29] HERVÉ DEBAR, MARC DACIER, ANDREAS WESPI, STEFAN LAMPART An experimentation workbench for intrusion detection systems Em *IBM Zurich Research Laboratory* (março de 1998), Research Report, RZ 2998

[30] R LIPPMANN, J HAINES, D FRIED, J KORBA, K DAS The 1999 DARPA off-line Intrusion Detection Evaluation Em *Computer Networks* (2000) Número 34, Páginas 579-595

[31] BISWANATH MURKHERJEE, L. TOOD HEBERLEIN, KARL N. LEVITT Network Intrusion Detection Em *IEEE Network* (maio e junho de 1994) Número 8, Volume 3, Páginas 26-41

[32] JEAN-CLAUDE LAPRIE Dependable Computing: Concepts, Limits, Challenges Em *Special Issue of the 25th International Symposium on Fault-Tolerant Computing, IEEE Computer Society Press* (junho de 1995) Páginas 42-54

[33] J. HOWARD, T. LONGSTAFF A Common Language for Computer Security Incidents Em *Sandia National Laboratories (SAND98-8667)* (1998)

[34] HERVÉ DEBAR, MARC DACIER, ANDREAS WESPI Towards a taxonomy of intrusion-detection systems Em *Computer Networks* (abril de 1999), Volume 31, Número 8, Páginas 805-822

[35] CYNTHIA BAILEY LEE, CHRIS ROEDEL, ELENA SILENOK Detection and Characterization of Port Scan Attacks Em *www.cse.ucsd.edu/users/clbailey/PortScans.pdf* (acessado em 16 de março de 2005)

[36] VINOD YEGNESWARAN, PAUL BARFORD, JOHANNES ULLRICH Internet intrusions: global characteristics and prevalence Em *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (2003), Páginas 138-147

[37] DETHY Examining port scan methods - Analysing Audible Techiniques Em *http://www.synnergy.net/downloads/papers/portscan.txt* (acessado em 16 de março de 2005)

[38] THE TCPDUMP GROUP Tcpdump/Libcap Em *http://www.tcpdump.org* (acessado em 16 de março de 2005)

[39] MTU COMPUTER SCIENCE The Visual TCP/UDP Animator Em *http://www.cs.mtu.edu/vta/* (acessado em 16 de março de 2005)

[40] ALEFIYA HUSSAIN, JOHN HEIDEMANN, CHRISTOS PAPADOPOULOS A framework for classifying denial of service attacks Em *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (agosto de 2003), Páginas 99-110

[41] DAVID MOORE, GEOFFREY M. VOELKER, STEFAN SAVAGE Inferring Internet Denial-of-Service Activity Em *Proceedings of the 10th USENIX Security Symposium* (agosto de 2001)

[42] HAINING WANG, DANLU ZHANG, KANG G. SHIN Detecting syn flooding attacks Em *In Proceedings of IEEE INFOCOM* (2002)

[43] TINA DARMOHRAY Hot Spares for Dos Attacks Em *http://www.usenix.org/publications/login/2000-7/apropos.html* (acessado em 16 de março de 2005)

[44] JONATHAN LEMON Resisting SYN flood DoS attacks with a SYN cache Em *Proceedings of the BSDCon 2002 Conference* (fevereiro de 2002)

[45] CHECK POINT SOFTWARE TECHNOLOGIES LTD. Protocols and Related Defenses - Network Layer Em *http://www.checkpoint.com/products/firewall-1* (acessado em 16 de março de 2005)

[46] JUNIPER NETWORKS Denial of Service and Attack Protection Em *http://www.juniper.net/products/integrated/dos.pdf* (acessado em 16 de março de 2005)

[47] CHRISTOPH L. SCHUBA, IVAN V. KRSUL, MARKUS G. KUHN, EUGENE H. SPAFFORD, AUROBINDO SUNDARAM, DIEGO ZAMBONI Analysis of a Denial of Service Attack on TCP Em *IEEE Symposium on Security and Privacy* (maio de 1997), Páginas 208-223

[48] LIU DIHUA, WANG HONGZHI, WANG XIUMEI Data mining for intrusion detection Em *IEEE International Conferences on Info-tech and Info-net* (2001), Volume 5, Páginas 7-12

[49] YI HU AND BRAJENDRA PANDA A data mining approach for database intrusion detection Em *Proceedings of the 2004 ACM symposium on Applied computing* (2004), Páginas 711-716

[50] LOTFI A. ZADEH Fuzzy logic, neural networks, and soft computing Em *ACM Communications* (1994), Volume 37, Número 3, Páginas 77-84

[51] BABUSKA, ROBERT Fuzzy Systems, Modeling and Identification Em *http://dutera.et.tudelft.nl/ babuska/transp/fuzzmod.pdf* (acessado em 25 de abril de 2005)

[52] D. J. BERNSTEIN SYN Cookies Em *http://cr.yp.to/djb.html* (acessado em 16 de março de 2005)

[53] SHAWN OSTERMANN TCPTrace Group Em *http://www.tcptrace.org* (acessado em 16 de março de 2005)

[54] STEFFEN BEYER Date Calc Time Manipulation Package Em *http://www.perl.com/CPAN/authors/id/S/ST/STBEY/* (acessado em 16 de março de 2005)

[55] THE MYSQL GROUP MySQL Data Base Em *http://www.mysql.org* (acessado em 16 de março de 2005)

[56] THE PERL GROUP Program Extract Report Language Em *http://www.perl.org* (acessado em 16 de março de 2005)

[57] DONN B. PARKER Demonstration the Elements of Information Security with Threats Em *Proceedings of the 17th National Computer Security Conference* (1994), Páginas 421-430

[58] JYH-SING ROGER JANG ANFIS: Adaptive-Network-Based Fuzzy Inference System Em *IEEE Transactions on Systems, Man, and Cybernetics* (1993), Volume 23, Páginas 665-684

[59] MAMDANI, E. H., ASSILIAN, S. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller Em *In International Journal of Human-Computer Studies* (1999), Volume 51, Número 2, Páginas 135-147

[60] T. TAKAGI, M. SUGENO Fuzzy Identification of Systems and Its Applications to Modeling and Control Em *IEEE Transactions on Systems, Man and Cybernetics* (1985), Volume 51, Número 1, Páginas 116-132

[61] JACQUES G. GANOULIS Engineering Risk Analysis of Water Pollution: Probabilities and Fuzzy Sets, 1^a ed. Wiley-VCH Verlag GmbH. ISBN 3527300503.

[62] ANDRAS BARDOSSY, LUCIEN DUCKSTEIN Fuzzy Rule-Based Modeling with Applications to Geophysical, Biological, and Engineering Systems, 1^a ed. CRC Press. ISBN 0849378338.

[63] CHIN-LIANG CHANG Fuzzy-Logic-Based Programming, 2^a ed. World Scientific Publishing Company. ISBN 9810230702.

[64] KAZUO TANAKA An Introduction to Fuzzy Logic for Pratical Applications, 1^a ed. Springer-Verlag. ISBN 0387948074.

[65] WITOLD PEDRYCZ, FERNANDO GOMIDE An Introduction to Fuzzy Sets Analysis and Design, 1^a ed. The MIT Press. ISBN 0262161710.

[66] HIME AGUIAR E. OLIVEIRA JR. Lógica Difusa Aspectos Práticos e Aplicações, 1^a ed. Editora Interciência. ISBN 8571930244.

[67] A. NÜRNBERGER, D. NAUCK, R. KRUSE Neuro-fuzzy control based on the NEFCON-model: recent developments Em *Soft Computing - A Fusion of Foundations, Methodologies and Applications* (1999), Volume 2, Número 4, Páginas 168-182