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

Joelson de Alencar Degaspari

Análise Comparativa dos Métodos de Detecção de Intrusão Logcheck e Rede Neural

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

Prof. Dr. João Bosco Mangueira Sobral

Florianópolis, Fevereiro de 2009

Análise Comparativa dos Métodos de Detecção de Intrusão Logcheck e Rede Neural

Joelson de Alencar Degaspari

Esta Dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Dr. Frank Augusto Siqueira

Banca Examinadora

_____

Prof. Dr. João Bosco Mangueira Sobral

_____

_____

Prof. Dr. Joni da Silva Fraga

_____

Prof. Dr. Mauro Roisenberg

_____

Prof. Dr. Ricardo Felipe Custódio

Para adquirir conhecimento, é preciso estudar;
mas para adquirir sabedoria, é preciso observar.
Autor desconhecido

Dedico este trabalho em memória de meu grande amigo Laend Vilan, por ter me ensinado os primeiros passos na computação.

Agradecimentos

Agradeço primeiramente a Deus por ter me permitido viver com saúde até hoje para assim poder cumprir mais esta etapa em minha vida.

A minha família, pelo constante incentivo aos estudos, desde os anos iniciais de minha vida escolar. Além da compreenção nos diversos momentos de estresse.

A Giselle Thais Kramer de Jesus pelo seu apoio durante todo este grande projeto, além de não medir esforços em sua ajuda no entendimento e escrita da fundamentação biológica.

Ao meu professor e orientador João Bosco Mangueira Sobral pelo seu apoio e confiança nestes vários anos de estudo, não medindo esforços nas orientações nem mesmo nos finais de semana e feriados.

Aos co-orientadores e amigos Igor Vinícius Mussoi de Lima e Renato Bobsin Machado pelas excelentes sugestões e orientações, sempre direcionando o trabalho para o rumo certo.

Aos amigos e companheiros de laboratório, Clytia Higa Tamashiro, Eder Contri e Vivian Cremer Kalempa pelas discussões técnicas, cooperação e aprendizado mútuo.

Aos amigos conquistados na UFSC, Claudio Biazus, Claudio Flores, Jeferson Luiz Rodrigues Souza, Lucas Guardalben, Vinicius da Cunha Martins Borges. Obrigado pela amizade, troca de conhecimento e dos momentos de descontração.

A todos que direta ou indiretamente fizeram parte deste projeto.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ACL . . . . . Agent Communication Language

CERT . . . . . Computer Emergency Response Team

CERT.br . . . . Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

CIDF . . . . . Common Intrusion Detection Framework

DNS . . . . . Domain Name System

FTP . . . . . File Transfer Protocol

HIDS . . . . . Host Based Intrusion Detection System

HTTP . . . . . Hyper Text Transfer Protocol

IP . . . . . Internet Protocol

NIDS . . . . . Network Based Intrusion Detection System

RNA . . . . . Rede Neural Artificial

SDI . . . . . Sistema de Detecção de Intrusão

SIA . . . . . Sistema Imunológico Artificial

SIH . . . . . Sistema Imunológico Humano

SSL . . . . . Secure Sockets Layer

TCP . . . . . Transmition Control Protocol

VPN . . . . . Virtual Private Network

Resumo

Este trabalho mostra uma abordagem de detecção de intrusão bioinspirada. O modelo utiliza os princípios e conceitos de defesa do sistema imunológico humano para a realização de transporte e armazenamento de `patógenos` (sessões intrusivas) além de reatividades passivas e ativas buscando minimizar os danos aos sistemas atacados. A base para os diagnósticos são os fluxo de dados capturados de uma rede de computadores, os quais são analisados por uma rede neural artificial. As sessões capturadas são transportadas por meio de agentes móveis até uma estação segura com a rede neural, estes agentes também são responsáveis pelo armazenamento da `memória imunológica` e da efetivação das reações. Esta abordagem tem arquitetura baseada em rede e distribuída, utilizando método de detecção por abuso e anomalia. Os resultados obtidos por meio dos experimentos permitiram verificar a possibilidade de detecção de variantes de ataques conhecidos além da vantagem expressiva em relação a sistemas de detecção baseados puramente em regras, onde necessitam de extensas bases de conhecimento. Através dos resultados dos experimentos foi possível estabelecer uma análise comparativa com o método de detecção de intrusão Logcheck.

Palavras Chaves: *Detecção de Intrusão, Segurança de Redes, Sistemas Imunológicos Artificiais, Redes Neurais Artificiais, Agentes Móveis, Logcheck*

Abstract

This work shows a bioinspired intrusion detection approach. This model use principles and concepts of defense of the human immune system for implementation of transport and storage of pathogens (intrusive sessions) beyond passive and active reactivity, seeking to minimize the damage to infected systems. The basis for the diagnoses are the data flow captured from a computers network, which are analyzed by an artificial neural network. The captured sessions are carried by mobile agents to a secure station that contains the neural network, these agents are also responsible for the storage of immunological memory and the realization of reactions. This approach has a network-based and distributed architecture, using misuse and anomaly detection method. The results obtained by the experiments showed the possibility to detection of variants of intrusion, beyond the expressive advantages in relation to intrusion detection systems purely based on rules, which require extensive knowledge databases. Through the results of experiments were established a comparative analysis with the method of intrusion detection Logcheck.

Key Words: *Intrusion Detection, Network Security, Artificial Immune Systems, Artificial Neural Network, Mobile Agents, Logcheck.*

Capítulo 1

Introdução

A partir da evolução tecnológica pode-se observar a real necessidade de melhora na segurança dos sistemas envolvidos. A acelerada disseminação da Internet, aliada à facilidade do uso, possibilitou a exposição de dezenas de milhares de computadores a iminentes invasores. Esta utilização indiscriminada ocasiona também o aprendizado e aperfeiçoamento de técnicas intrusivas [Deng et al. 2003].

Por outro lado, tem-se as técnicas de defesa, que procuram impedir ou ao menos minimizar os danos causados aos sistemas. Uma infraestrutura de rede para ser considerada segura, deve atender, ao menos, os requisitos de integridade, privacidade, disponibilidade e autenticação. Para proporcionar estes requisitos, costuma-se usar sistemas especialistas ou mineração de dados.

O crescimento dos incidentes reportados todos os anos ao CERT (Computer Emergency Response Team) [CERT/CC 2007], cria a necessidade de existência de uma segurança de redes mais eficaz. Assim, pesquisas que contribuam com o aumento da segurança das redes de computadores têm se tornado cada vez mais frequentes.

As abordagens atuais de detecção de intrusão estão procurando utilizar a técnica de reatividade por meio de ações do próprio sistema, minimizando o tempo de resposta. Entre os diferentes métodos que vêm sendo pesquisados, muitos são inspirados em conceitos biológicos, criando-se soluções computacionais bioinspiradas que têm proporcionado bons resultados ao longo das pesquisas. Assim, redes neurais artificiais

(RNA), algoritmos genéticos, sistemas imunológicos artificiais (SIA) e comportamentos de formigas [Castro 2001] têm sido usados para a solução de determinados problemas de segurança computacional.

Com base neste cenário, esta pesquisa busca a integração, visando a evolução, de dois projetos anteriormente desenvolvidos. Ambos baseados em inteligência artificial, procuram resolver problemas em segurança computacional. E, a partir disto é possível estabelecer os devidos comparativos com as soluções anteriores.

1.1 Computação com Inspiração Biológica

Pesquisadores da área de inteligência artificial(IA), têm utilizado conceitos biológicos como inspiração para resolução de problemas computacionais. Esta rica fonte de inspiração, pode por exemplo, trazer para o meio computacional um certo nível de inteligência através do estudo dos neurônios e suas sinapses, ou até mesmo aperfeiçoar as técnicas de proteção de redes de computadores, utilizando como base as proteções do corpo humano [Castro 2001].

Seres vivos que possuem sistemas nervosos são capazes de armazenar conhecimento, ou seja, aprendem a partir de experiências vividas. Esta capacidade de aprender se deve à plasticidade do cérebro. Outro quesito relevante é a velocidade de processamento, que, devido os neurônios cerebrais trabalharem em paralelo, deixam para trás qualquer computador utilizado atualmente.

O cérebro humano possui circuitos neurais complexos, os quais são formados pela interligação de neurônios. Um neurônio recebe informações de vários outros neurônios e esta passagem de sinal entre eles é conhecida como sinapse [Lent 2001]. Usando modelos matemáticos baseados nesta constituição estrutural do cérebro, as Redes Neurais Artificiais (RNA) puderam ser construídas. Assim, conseguindo esta modelagem computacional consegue-se gerar comportamento inteligente em máquinas [Haykin 2001].

O sistema imunológico humano (SIH) [Castro e Zuben 1999] tem características funcionais e desejáveis a sistemas de proteção computacional. Uma rede de computadores pode ser comparada ao corpo humano, neste caso os computadores seriam

os órgãos. Estes orgãos devem ser protegidos de agentes agressores externos, da mesma forma que os computadores. É interessante também que esta proteção aos computadores possa ser adaptativa e possibilite reagir a agressões inéditas, como nos SIHs.

Baseado nestas características foi possível o desenvolvimento dos sistemas imunológicos artificiais (SIA) [Dasgupta et al. 1999], os quais podem ser utilizados em diversas áreas da computação. Em especial, pode ser citada a área de segurança [Castro 2001]. Aliado aos SIAs podem-se utilizar as RNAs, podendo assim agregar as funcionalidades de aprendizado e generalização para reagir a ataques inéditos.

1.2 Trabalhos Relacionados

Este é o quarto projeto de pesquisa em segurança de redes com bio inspiração, desenvolvido no PPGCC - Programa de Pós Graduação em Ciência da Computação na UFSC (Universidade Federal de Santa Catarina), no âmbito do *Distributed Mobile Computing and Network Security Research Group*. Esta linha de pesquisa teve início com [Juca 2001], que avaliou problemáticas de segurança de redes, observando a geração, recepção e interpretação dos dados das atividades anômalas, dentro da idéia do uso de um sistema imunológico artificial definido e utilizando os sistemas *Syslog-ng* e *Logcheck* para respectivamente gerar e analisar os *logs* registrados no *host*.

Teve sequência com [Machado 2005], seguindo a inspiração biológica, com a arquitetura baseada em *host*, o método de detecção baseado em anomalias. Como evolução da pesquisa, utilizou-se componentes distribuídos para armazenamento de *logs*, reações pró-ativas e análise de detecção em tempo contínuo. Foram usadas tecnologias de agentes móveis para obter esta solução.

Outro projeto bio-inspirado está em [Lima 2005], que fez uso de redes neurais artificiais (RNA) como mecanismo para detecção de intrusões. O método de detecção é híbrido, pois a solução incorpora características de ambos os métodos, detecção por abuso e por anomalia e a arquitetura é baseada em rede, ou seja, analisa-se o tráfego gerado na rede por meio de um *sniffer*.

No trabalho de [Mukkamala e Sung 2003] é mostrado um estudo com-

parativo entre o uso de Máquinas de Vetores de Suporte (MVSs) e Redes Neurais Artificiais (RNAs) para detecção de intrusão. É verificado o desempenho dos SDIs com SVM e com RNA usando um conjunto de avaliação conhecido e capturado pelo DARPA. Por meio dos experimentos comparativos os autores verificaram que o uso das MVSs supera a performance das RNAs em pelo menos três aspectos críticos. É obtido uma exatidão muito maior nas respostas das Máquinas de Vetores de Suporte do que das Redes Neurais melhores treinadas. O tempo de treinamento e de teste das MVSs são muito menores do que das RNAs. As MVSs tem uma escalabilidade maior que as RNAs, podendo ter um treinamento com um maior número de padrões, sem levar um tempo absurdamente grande, nem falhar na convergência. É mostrado na pesquisa que as MVSs permite que seja alcançada uma grande exatidão na detecção de todas as classes de ataque.

Em [Yu et al. 2006] é descrito uma abordagem de detecção de intrusão usando um modelo híbrido de rede neural. O modelo une uma rede neural do tipo Perceptron de Múltiplas Camadas (MLP) e uma rede neural caótica para detecção de eventos por anomalia. Neste trabalho, o autor expõe que normalmente são utilizadas redes MLP em SDI baseados em rede neural, que possibilitam a detecção de novos ataques, mas são gerados altas taxas de alarmes falsos. A utilização do modelo híbrido proporciona a união das características de detecção de ataques em tempo real das redes MLP, com a possibilidade de detecção de ataques compostos por uma série de eventos anormais das redes neurais caóticas, com uma menor taxa de alarmes falsos.

Em [Al-Subaie e Zulkernine 2006] é mostrado que os sistemas de detecção de intrusão por anomalia (Anomaly Intrusion Detection Systems - AIDSs) possuem um grande potencial para detectar ataques inéditos, mas apesar disto padecem com a falta de capacidade de generalização e com os altos índices de alarmes falsos. Algumas técnicas são propostas na literatura para superar esta falta de capacidade de generalização, que são as técnicas estáticas que realizam o reconhecimento do padrão estrutural. Mas estas técnicas não são capazes de modelar eficientemente a relação sequencial entre os padrões de eventos que constituem os comportamentos normais e anormais. Os autores mostram com a pesquisa, que esta propriedade é vital para detecção de intrusão por anomalia e que quando modelada eficientemente é possível produzir robustos AIDSs. Para

provar estas afirmações são testados duas técnicas diferentes de detecção, o modelo oculto de Markov (Hidden Markov Models - HMMs), que produzem mecanismos de aprendizado sequencial e redes neurais artificiais com algoritmo de retropropagação de erros que produzem técnicas de reconhecimento de padrões estruturais. Com os experimento fica provado que o desempenho dos classificadores HMMs supera o desempenho obtido com a RNA com MLP para determinado conjunto de testes.

A pesquisa descrita em [Mo, Ma e Xu 2008] relata que devido a mobilidade, os agentes autônomos podem prover conveniência, eficiência e robustez a programação para aplicações distribuídas. Verificando estas características, aplicou-se elas a sistemas de detecção de intrusão e assim foi proposto um novo modelo de detecção de intrusos. Este novo modelo baseado em agentes móveis, combina a idéia de distribuição plenamente utilizada a inteligência, mobilidade, cooperação e heterogeneidade. Os experimentos mostraram que o IDS foi capaz de capturar os dados intrusivos a medida que eles apareceram, e que o seu uso não comprometeu os recursos de CPU e memória.

A pesquisa relatada em [Michailidis, Katsikas e Georgopoulos 2008] avalia a habilidade em encontrar ataques de um Sistema de Detecção de Intrusão de rede que possui como mecanismo de análise uma Rede Neural Evolucionária - RNE (Evolutionary Neural Network - ENN). O treinamento da RNE é realizado pelo algoritmo de otimização de enxame de partículas (Particle Swarm Optimization - PSO). São relatados bons resultados com os experimentos para o reconhecimento de ataques conhecidos e desconhecidos, com uma excelente exatidão para os ataques conhecidos. Com ataques DoS, de varredura e promoção a super usuário foram verificados melhores resultados que os obtidos na competição KDD99, que foi baseado na DARPA 1998. No entanto, nos ataques onde se tem que ganhar o acesso inicial a máquina houveram dificuldades para se identificar.

1.3 Proposta de Dissertação

Este projeto representa uma continuidade de trabalhos desenvolvidos no PPGCC - UFSC, na linha de pesquisa de segurança computacional. Assim é proposta uma abordagem simplificada, a qual faz uso da potencialidade das redes neurais artifi-

ciais para o reconhecimento de padrões [Haykin 2001] e a funcionalidade dos sistemas imunológicos artificiais para reagir a prováveis invasores [Castro 2001].

Através desta proposta com inspiração biológica é desenvolvido um SDI que, a partir do protótipo desenvolvido em [Machado 2005], tem a substituição das ferramentas *Syslog-ng* e *Logcheck* pela RNA de [Lima 2005]. Assim é possível classificar os eventos intrusivos utilizando-se de uma ferramenta de inteligência artificial. O tráfego da rede é monitorado e as sessões armazenadas em arquivo para posterior análise. Este arquivo é apresentado para uma rede neural artificial previamente treinada para reconhecer padrões intrusivos em determinados protocolos. A tecnologia de agentes móveis é utilizada para implementar a solução imunológica artificial.

Desta forma se torna possível estabelecer um comparativo válido entre os dois protótipos analisados, pois as variáveis a serem comparadas são as mesmas. São analisados os resultados das detecções em cada um dos protocolos escolhidos. Estes resultados comparativos são verificados estatísticamente para se obter o grau de significância das diferenças obtidas nestes grupos de dados.

1.4 Organização do Trabalho

O trabalho está organizado como segue. No capítulo 2, tem-se uma breve conceituação de segurança em redes de computadores, tipos de ataques e políticas de segurança, além da explanação sobre sistemas de detecção de intrusão, principalmente os que se utilizam de inteligência artificial.

O capítulo 3, apresenta os conceitos bio-inspirados utilizados neste trabalho, desde o sistema imunológico humano até os sistemas imunológicos artificiais, mostrando suas propriedades, processos, princípios básicos e anatomia. São mostradas as semelhanças encontradas entre as necessidades de defesas de um corpo humano e de uma rede de computadores. Também são apresentadas as redes neurais artificiais, descrevendo suas propriedades e utilizações além de uma conceituação da origem dessas propriedades matemáticas. Neste capítulo, é demonstrada a importância da aplicação da inteligência artificial para solução de problemas que necessitam de aprendizado e generalização.

No capítulo 4, está a descrição da proposta deste trabalho. É construído um protótipo de detecção de intrusão com inspiração biológica. Um sistema que se utiliza de deteção neural de eventos intrusivos aliado ao paradigma dos sistemas imunológicos artificiais e agentes móveis para a realização das reações inteligentes.

No capítulo 5, tem-se a apresentação e discussão dos resultados obtidos durante os testes realizados de acordo com cada situação de configuração e treinamento.

O capítulo 6 descreve as conclusões obtidas através dos testes realizados no capítulo anterior. São discutidas os desafios e as realizações alcançadas com o modelo. Também são lançadas propostas para trabalhos futuros, que poderiam dar sequência à pesquisa.

Capítulo 2

Segurança e Detecção de Intrusão em Redes de Computadores

Administradores de rede costumam questionar se existem motivos, e quais são eles, para que suas redes sejam invadidas. Os motivos podem ser os mais variados, desde um simples teste de conhecimento, obtenção de informações sigilosas ou até mesmo a busca por destruição, causando prejuízos ao ambiente atacado. Por menor que seja a rede administrada, o responsável não deve descartar a possibilidade de invasões.

A invasão de uma máquina pode ter outro interesse além de uma simples cópia ou destruição de dados armazenados no disco rígido. O interesse pode ser a largura de banda para internet que o computador tem disponível. Um atacante que tenha vários computadores sob seu controle, ainda mais possuindo conexões de alta velocidade, pode, por exemplo, enviar *spams*, fazer escaneamento de portas (*portscans*) ou lançar ataques coordenados contra grandes servidores. Todos esses ataques terão como origem o endereço *IP (Internet Protocol)* do computador controlado pelo atacante.

Ataques e invasões são baseadas principalmente em vulnerabilidades. Estas vulnerabilidades surgem devido a erros de programação em softwares ou protocolos, aplicações mal configuradas, além da falta de uma política de segurança bem definida também pode contribuir para o comprometimento da integridade de uma rede de computadores.

2.1 Segurança Computacional

A crescente necessidade de uso das redes de computadores, cujo tráfego das informações, muitas vezes, deve ser sigiloso, faz com que aumente a preocupação com a segurança envolvida. É possível observar na Figura 2.1 que, mesmo com grandes esforços para possibilitar melhoras na segurança computacional, o número de incidentes reportados ao CERT.br - *Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil* [CERT.BR 2007] aumentou significativamente nos últimos anos.

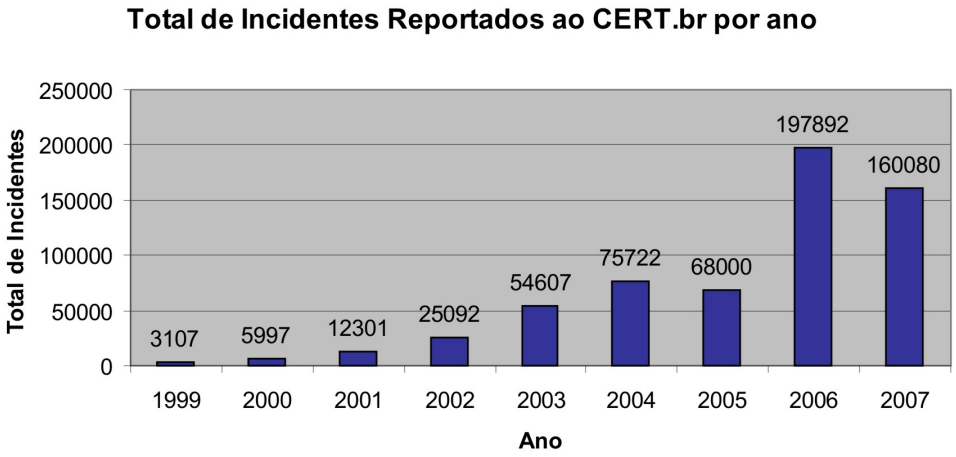

Figura 2.1: Estatística de Incidentes Reportados ao CERT. Fonte:[CERT.BR 2007]

A segurança de uma rede de computadores é avaliada constantemente, e pode-se afirmar que o administrador não está sozinho neste serviço pois nem sempre é ele quem realiza a avaliação. De acordo com o CERT.br [CERT.BR 2007], computadores ligados à internet continuamente sofrem tentativas de ataques e violações. Alguns destes ataques podem ser simples varreduras, para as quais pode-se ter facilmente uma defesa, outros podem ser mais complexos e acabam pegando o administrador de surpresa.

É difícil afirmar o quanto uma rede é segura ou vulnerável. É necessário entender primeiramente que nada é inviolável e que nenhum componente isolado será capaz de defender uma rede de computadores perfeitamente. No entanto, com a união de componentes é possível se alcançar a proteção almejada. *Firewalls*, *Virtual*

Private Networks(VPN) - Redes Privadas Virtuais, antivírus e sistemas de detecção de intrusão(SDI) proporcionarão melhores resultados quando utilizados em conjunto.

Para se ter segurança de computadores e suas redes, primeiramente deve-se ter em mente a necessidade de boas práticas com relação à configuração, administração e operação das mesmas. Estas práticas podem não ser suficientes em todas as situações, mas é o mínimo a ser adotado para se reduzir problemas de segurança ou apenas facilitar a administração.

O aumento da quantidade de ataques realizados se deve, principalmente, à facilidade em encontrar ferramentas especializadas para este fim [Tanenbaum 2003]. Pode-se ver na Figura 2.2 que a sofisticação dos ataques tem crescido consideravelmente, enquanto o conhecimento técnico do intruso tem diminuído. Desta forma, com o passar dos anos tem ficado mais fácil para curiosos com pouco conhecimento causarem grandes problemas [Schulter 2006].

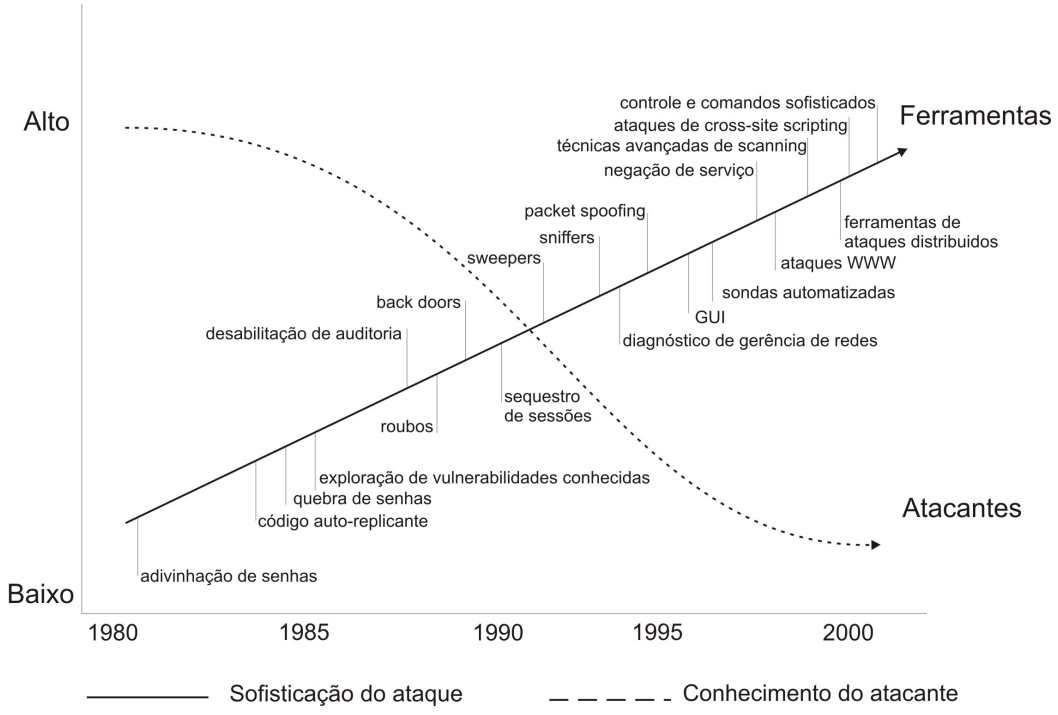

Figura 2.2: Sofisticação do Ataque vs. Conhecimento Técnico do Invasor. Adaptado de [Lipson 2002]

Para se ter um ambiente seguro e um maior controle sobre os recursos computacionais é necessário que se atenda a alguns requisitos. Estes princípios básicos de segurança computacional, podem ser citados como confidencialidade, integridade, disponibilidade e autenticidade.

- Autenticidade - Esta propriedade está diretamente associada à certificação da identidade. A legitimidade de um usuário ou sistema ao tentar acessar outro deve ser comprovada. Desta forma, fica assegurado ao receptor de uma mensagem que o emissor realmente é quem diz ser.

- Confidencialidade - A confidencialidade diz respeito à definição de quem pode ter acesso a informações, assim dados sigilosos são protegidos contra a revelação para alguém não autorizado pelo proprietário. Pode ser definida como o dever da preservação das informações íntimas e privadas.

- Integridade - Integridade significa manter um dado intacto, ou seja, protegê-lo contra corrupção intencional ou acidental. A informação não pode ser modificada sem a permissão do proprietário. Significa, desta maneira, ter disponíveis informações corretas, confiáveis e utilizáveis.

- Disponibilidade - Consiste em garantir que dados, sistemas e computadores estarão acessíveis quando forem solicitados. Garantindo assim, a proteção dos serviços prestados, de forma que não sejam degradados nem fiquem indisponíveis. Quanto maior for a disponibilidade desejada, maior deve ser a redundância e consequentemente os custos.

2.1.1 Ameaça à Segurança

Ameaças podem ser definidas como qualquer evento que possa explorar vulnerabilidades dos sistemas de segurança, e que ao serem explorados causem perdas ou danos. É preciso também identificar quem poderá explorar essas vulnerabilidades e quais serão os objetivos. Um atacante pode ser classificado pelo seu nível de conhecimento técnico em computação, que pode ser principiante, intermediário ou avançado.

Um atacante principiante geralmente não tem noção do que está fazendo, nem mesmo das consequências de seus atos. Apenas usa ferramentas prontas de terceiros. Os de nível intermediário possuem algum conhecimento em programação mas também utiliza alguma ferramenta de terceiro. Os atacantes avançados, em geral são programadores experientes e têm um alto grau de conhecimento em protocolos e infraestrutura.

Os principais tipos de ameaças: *Interrupção, Interceptação, Modificação, Fabricação* podem ser observados na Figura 2.3 e são definidos em seguida.

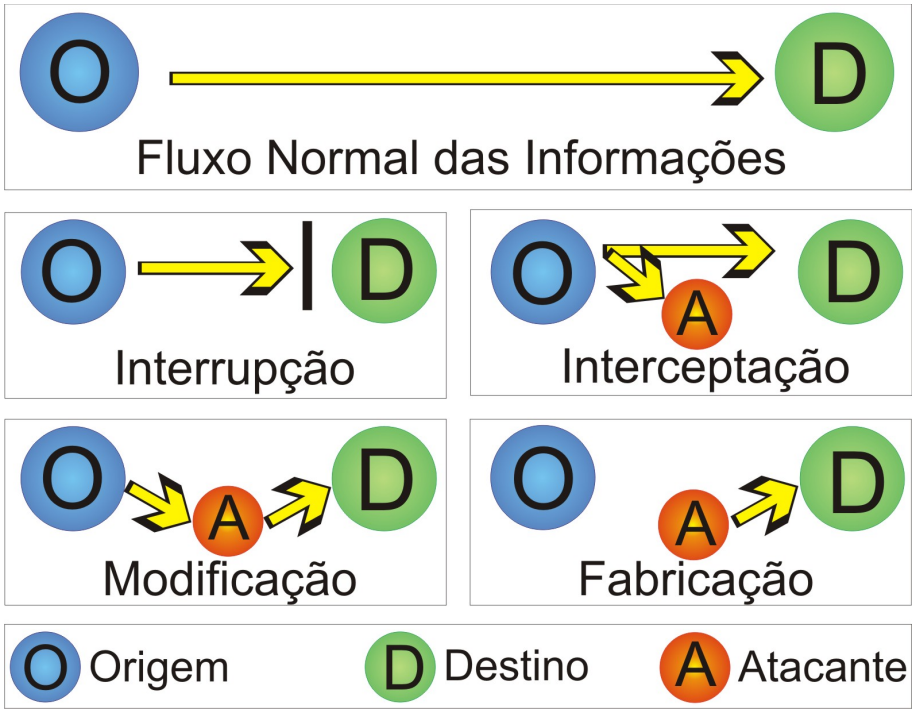

Figura 2.3: Tipos de Ameças à Segurança. Adaptado de [Machado 2005]

- *Interrupção* - Interromper um fluxo de dados, significa não permitir que as informações percorram o seu trajeto desde a origem até o seu destino. Desta maneira, dados podem ser perdidos ou sistemas se tornarem indisponíveis. Este é um ataque contra a disponibilidade dos recursos.

- *Interceptação* - Diz respeito à observação e captura do fluxo de informações. Todo o tráfego é verificado pelo atacante antes de chegar em seu destino. Para se concre-

tizar basta ter uma interface de rede em modo promíscuo para se copiar os dados. Este ataque é do tipo passivo, pois os dados não são alterados e viola o princípio da confidencialidade. É o mais comum e de difícil detecção pelos verdadeiros donos das informações.

- *Modificação* - Modificação é uma variação com agravante da *Interceptação*, pois além de ver informações alheias, as modifica, fazendo com que o destinatário acredite ser a mensagem original. É um ataque do tipo ativo, visto que interage com fluxo de dados. Viola os princípios da confidencialidade e integridade.

- *Fabricação* - Na fabricação, o fluxo de dados é criado pelo atacante que se faz passar por outra pessoa ou sistema. Simula desta forma para o destinatário, uma mensagem legítima. É um ataque ativo e afeta a autenticidade das informações.

2.1.2 Tipos de Ataques

Um ataque pode ser definido como sendo uma ação maliciosa que viola as políticas de segurança, compromete a integridade e/ou disponibilidade dos recursos computacionais utilizando basicamente de exploração de vulnerabilidades existentes em sistemas e protocolos. Isto pode resultar em uma invasão ou mesmo causar indisponibilidade dos sistemas alvo [Barbosa e Moraes 2000].

Baseado nesta definição, observa-se que os ataques podem ser divididos em três classes de acordo como agem nos sistemas. Sondagem, comprometimento de recursos e penetração são definidos e exemplificados a seguir [Souza, Silva e Cansian 2002].

- Ataques de sondagem são os que utilizam métodos investigativos para encontrar alvos na rede ou suas vulnerabilidades. Com estas buscas é possível obter dados essenciais para a escolha da ferramenta mais adequada à exploração da vulnerabilidade encontrada.

  – Sniffer - Um sniffer é um programa criado para auxiliar na gerência de redes de computadores, no entanto tem sido, maliciosamente, utilizado por pessoas

mal intencionadas. Através de um sniffer, é possível capturar todos os dados que trafegam na rede e analisar os que não estejam criptografados. Logins, senhas e qualquer outra informação que esteja em texto plano podem ser furtadas.

  - Scanner de Portas - São programas que fazem varreduras em busca de portas abertas nos computadores para tentar uma invasão. Sistemas de detecção de intrusão acusam estes testes de verificação quando feitos de forma seguida, por isso os scanners mais sofisticados levam dias fazendo testes em horários aleatórios. A partir do momento em que o atacante tem a lista de portas abertas de um servidor alvo, ele verifica quais serviços e suas versões estão rodando nestas portas. Assim, é possível definir qual *exploit* melhor explorará as vulnerabilidades.

- Comprometimento de recursos é o tipo de ataque que visa impossibilitar a utilização de um sistema, através da sobrecarga do alvo. Ataques deste tipo não necessariamente implicam na invasão do servidor, resultando em furto, eliminação ou alteração de dados.

  - Negação de Serviço - O ataque de negação de serviço (*Denial of Service - DoS*), tem como objetivo degradar o tempo de resposta de servidores de rede com sua lentidão, ou até mesmo tirá-los temporariamente do ar. Tem como alvos servidores Web, de e-mail e também de DNS (*Domain Name Server* - Servidores de Nome de Domínio). Geralmente, tem-se aliado a estes ataques a técnica de mascaramento do endereço de IP origem (*IP Spoofing*) [Barbosa e Moraes 2000], que altera o cabeçalho dos pacotes para que um computador se faça passar por outro. Exemplos de ataques DoS são *TCP SYN Attack*, que basicamente explora a forma como são iniciadas as conexões entre um cliente e um servidor, pois, a cada pedido, o servidor aloca recursos e com um número grande de requisições ele fica sobrecarregado e recusa novos pedidos. *Flood Attack*, semelhante ao ataque definido anteriormente cujo servidor alvo é inundado por certos tipos de requisições. *Smurfing* é tipo de ataque cujo

atacante envia muitas seqüências de solicitações de *Ping* para um endereço de *broadcast*, assim, usando *IP spoofing*, as respostas são todas encaminhadas para a vítima.

  - Negação de Serviço Distribuído - Este ataque, também chamado de *Distributed Denial of Service(DdoS)*, é uma variação do DoS que agrega poder ao ataque. Nesta classe de ataque, o atacante invade uma máquina que passa a ser seu computador mestre. A partir desta máquina-mestre outros são invadidos e assim se tornam seus zumbis. O princípio do DDoS é que o ataque seja lançado de vários computadores distintos que podem estar em qualquer parte do mundo. É coordenado pelo atacante através do computador mestre, mas executado pelos zumbis. Desta maneira, além de ser um ataque mais robusto e sincronizado, o verdadeiro responsável fica escondido.

- A penetração em um sistema acontece após a exploração de uma vulnerabilidade existente ou a inserção de programas que a ocasionem. Assim proporciona a aquisição de privilégios, recursos e dados, geralmente o resultado final neste tipo de ataque acaba sendo a obtenção do controle da máquina ou sistema invadido. Geralmente este tipo de ataque é realizado em cinco principais fases. Fase de reconhecimento, na qual é escolhida a rede ou máquina alvo. Fase de sondagem (*Scanning*), como descrito anteriormente, é a busca por vulnerabilidades. Obtenção de acesso, a qual realmente se caracteriza pela penetração, fase em que o atacante utiliza-se de ferramentas de exploração que o permitirá entrar na máquina. Nesta próxima fase, o atacante busca recursos que lhe permitam facilitar os acessos posteriores a esta máquina, seja criando usuários, mudando permissões ou mesmo inserindo *backdoors* (portas de fundos). E por fim, o atacante limpa seus rastros para dificultar que o administrador responsável descubra a invasão.

  - Cavalo de Tróia - Programa malicioso que executa alguma tarefa disfarçadamente. Na maioria das vezes está anexado a uma foto, música ou outro arquivo, e quando executado também executa o trojan que abre uma porta TCP

do computador para uma futura invasão. Existem cavalos de tróia específicos para roubar e enviar por e-mail senhas e dados sigilosos.

2.1.3 Etapas de um Ataque

Invasores têm como objetivo principal ganhar acesso a sistemas ou aumentar os privilégios já conseguidos [Stallings 2003]. Isto geralmente requer o furto de informações que deveriam estar protegidas. Para atingir esse objetivo final, geralmente um ataque é caracterizado por uma sequência de passos.

Como passo inicial, o atacante precisa descobrir possíveis alvos que quando encontrados são investigados ao máximo, com intuito de se obter a maior quantidade de informações. A posse de informações do tipo sistema operacional e versão utilizados podem ser consideradas valiosas para uma investida, pois através disto se pode encontrar uma vulnerabilidade existente.

Ainda à procura de vulnerabilidades, o atacante pode executar um scanner de portas buscando listar as portas abertas desta máquina. Esta informação, aliada à versão de serviço que está rodando em determinada porta, pode resultar em um ataque bem sucedido. Devido à periculosidade de uma investigação deste tipo, sistemas de detecção de intrusão atuais estão aptos a detectar e alertar administradores. Alertas como este, devem ser considerados e tratatos para que não proporcionem uma invasão.

Após determinar as falhas que uma máquina possui, o atacante a explora e penetra no sistema. Dentro do sistema, o invasor buscará conseguir privilégios e, provavelmente, instalar programas que facilite o seu retorno quando necessário.

Para fechar o ciclo, o invasor busca limpar seus rastros. Com isso ele manipula *logs* e registros para que não se perceba o que aconteceu. Uma penetração em sistemas alheios tem como objetivo desde o roubo de informações até o uso de uma série de máquinas invadidas para ocasionar problemas em sistemas mais robustos, por meio de um ataque em massa.

2.1.4 Políticas de Segurança

A correta definição de uma política de segurança é fundamental para o sucesso do ambiente de segurança computacional. Quando bem escrita conterá a definição do "que" fazer de forma que o "como" poderá ser identificado, medido ou avaliado [Northcutt et al. 2002].

De acordo com Cert.br [CERT.BR 2007], políticas de segurança definem quais direitos e responsabilidades têm os usuários de recursos computacionais da instituição e das informações contidas neles, além de definir suas atribuições em relação à segurança destes recursos utilizados. Esta política deve conter o que pode e que não pode ser feito dentro da instituição, explicando também as penalidades a que serão sujeitados os que infringirem as regras. É considerado um incidente de segurança qualquer descumprimento à política de segurança.

Para uma política de segurança ser respeitada e se tornar efetiva, ela deve ser aceita por todos os níveis hierárquicos da instituição. E para que isso aconteça, a gerência corporativa deve oferecer o suporte necessário desde o desenvolvimento à implantação das políticas.

2.2 Detecção de Intrusão

Intrusão pode ser definida como sendo qualquer ação cujo objetivo seja comprometer algum dos princípios básicos de segurança computacional [Heady et al. 1990, Paula 2004] já definidos na seção 2.1 (Página 9), violando desta forma as políticas de segurança.

Modelos informatizados que visem identificar este tipo de violação, ou que também respondam, de alguma forma, para impedir ou minimizar os problemas ocasionados são chamados Sistemas de Detecção de Intrusão - SDIs (Intrusion Detection System - IDSs).

Baseado na definição de defesa em profundidade [Northcutt et al. 2002] os SDIs fazem parte da última linha de defesa computacional e possuem especial im-

portância devido à possibilidade de reações pró-ativas contra invasores, mostrando-se uma ferramenta preventiva contra ações consideradas maliciosas.

2.2.1 Estrutura dos SDIs

Devido a existência de diversas ferramentas e formas de detecção de intrusão, existe uma forte necessidade para padronização destas abordagens, nomenclaturas e seus componentes [Machado 2005]. Isso facilita a interação entre diferentes SDIs, trazendo mais possibilidades de uma detecção bem sucedida.

Um destes padrões é o *Common Intrusion Detection Framework - (CIDF)* [Kahn et al. 1998], que mostra com Figura 2.4, que a relação entre os componentes não segue uma seqüência rígida de passos, mas obrigatoriamente precisa ter as fases de geração e análise de eventos. Este padrão utiliza a *Common Intrusion Specification Language - (CISL)* como linguagem para comunicação entre componentes e especificação de eventos [Staniford-Chen et al. 1998, Barbosa e Moraes 2000, Machado 2005].

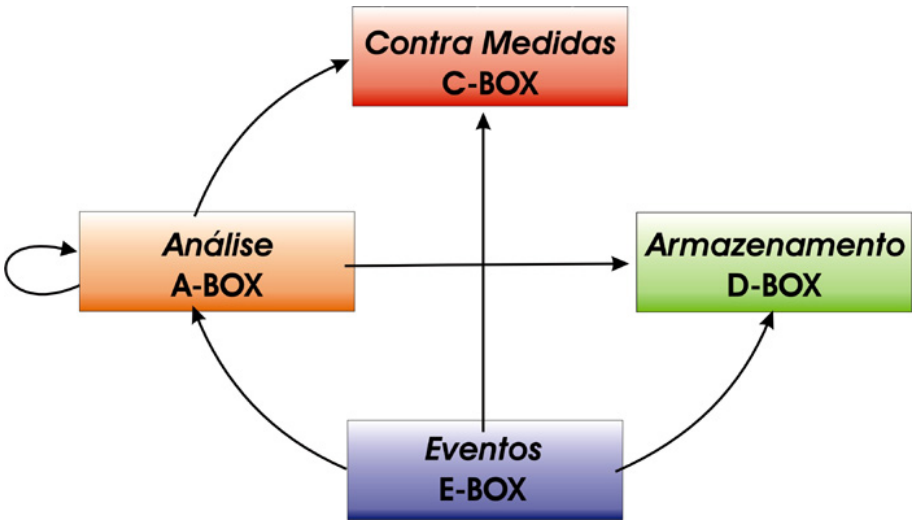

Figura 2.4: Elementos da Padronização CIDF - Adaptado de [Lima 2005]

Os componentes, que podem ser observados na Figura 2.4, são definidos como sendo: Caixa E (Geradores de eventos), Caixa A (Analisadores dos eventos), Caixa D (Mecanismo de armazenamento) e a Caixa C (Mecanismo de contra medidas)

[Militelli 2006].

- Caixas E (Geradores de Eventos)- São responsáveis pela captação dos eventos ocorridos nos *hosts* ou na rede. Esta caixa é composta de sensores devidamente posicionados visando capturar ocorrências de anomalias. *Logs* de sistemas e tráfego de rede podem ser considerados exemplos de geradores de eventos. Preparam os eventos para serem recebidos pelas A-Boxes.

- Caixas A (Analisadores de Eventos) - Responsáveis pela análise dos eventos gerados nas E-Boxes em busca de padrões pré-definidos de ataques. É nesta fase que a intrusão é detectada. A análise pode ser baseada nos métodos por anomalia, abuso ou híbrido.

- Caixas D (Mecanismo de Armazenamento) - As caixas D realizam o armazenamento dos eventos intrusivos detectados, compondo desta forma a base de dados de conhecimento do sistema de detecção. Através desta base de conhecimento é possível reconhecer e reagir mais rapidamente em uma situação hostil.

- Caixas C (Mecanismo de Contramedidas) - São responsáveis pelas reações que o SDI deve ter. Este componente contém as contramedidas que serão executadas para minimizar os problemas de uma invasão. O quanto antes estas respostas forem executadas, menores serão os efeitos ocasionados. Entre as ações a serem tomadas podem estar por exemplo o desligamento de servidores, emissão de alertas por e-mail ou até mesmo o contra-ataque à origem [Barbosa e Moraes 2000].

2.2.2 Classificação dos SDIs

Os componentes dos SDIs são desenvolvidos de acordo com diferentes abordagens. A utilização destas abordagens determina as características específicas de cada sistema de detecção e influencia na forma como o SDI funciona [Paula 2004].

A arquitetura de um SDI é classificada de acordo com quatro itens: Método de Detecção, Arquitetura, Comportamento Pósdetecção e Frequência de Uso [Campello e Weber 2001], as quais descrevem como o *hardware* e *software* que compõem

o sistema fará as detecções e como se comportará após elas. A figura 2.5 mostra como se dá a relação entre os itens da classificação dos SDIs.

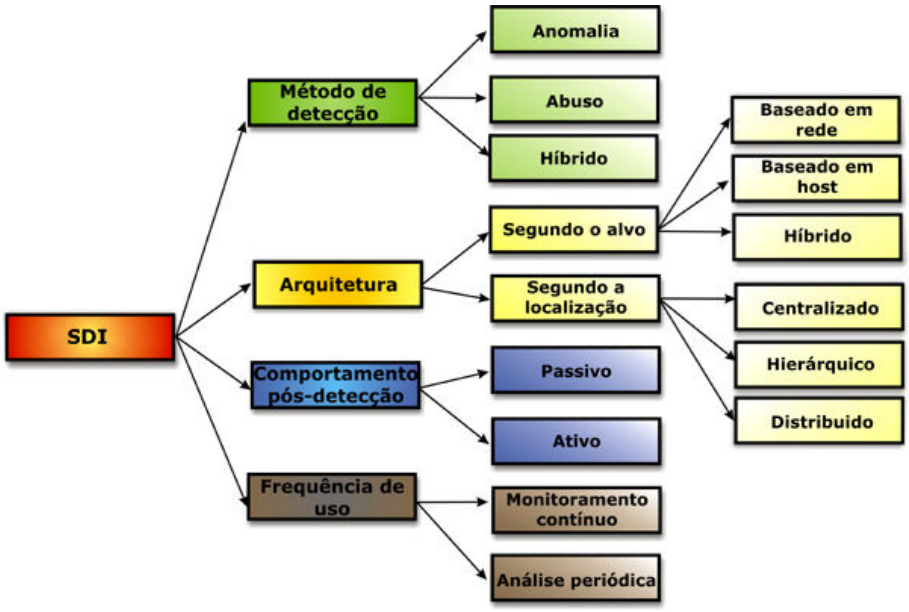

Figura 2.5: Classificação dos SDIs - Adaptado de [Campello e Weber 2001]

2.2.2.1 Métodos de Detecção

O método de detecção mostra o que o SDI verificará para conseguir encontrar um padrão intrusivo. Quanto ao método, um sistema de detecção, pode buscar anomalias nos comportamentos, assinaturas de intrusão na base de conhecimento ou unir ambas em uma abordagem híbrida [Laureano, Maziero e Jamhour 2007].

- Detecção por Anomalia

  Abordagem também conhecida por ser baseada em comportamento, devido à análise feita em busca do comportamento usual do sistema ou *host* monitorado. É traçado um perfil padrão com a observação das informações coletadas durante o uso normal, para que posteriormente situações analisadas nas quais ocorram qualquer desvio sejam consideradas ataques. Para a construção dos perfis podem ser utilizadas técnicas com base em regras, estatísticas, algoritmos genéticos, redes neurais e

sistemas imunológicos artificiais. Neste modelo é considerado que toda atividade intrusiva é anômala [Campello e Weber 2001, Paula 2004, Machado 2005].

A análise feita com base em comportamentos classifica-os em normais e anômalos, mas este tipo de avaliação permite a geração de alarmes falsos. Estes alarmes podem ser divididos em quatro categorias:

- Anômalo e Intrusivo: Atividade intrusiva e devidamente reconhecida e tratada pelo IDS. São conhecidas por verdadeiros positivos.
- Não anômalo e não intrusivo: Atividades denominadas verdadeiros negativos. São eventos não intrusivos e que não são detectados pelo sistema.
- Anômalo e não intrusivo: Atividade não intrusiva, mas reconhecida de maneira errada como anômala. Assim o sistema trata como se fosse um ataque. Conhecida também como falso positivo.
- Não anômalo mas intrusivo: Evento intrusivo mas não considerado pelo SDI como anômalo, assim não tem um devido tratamento de uma invasão. Esta categoria é a pior, pois permite que sistema fique vulnerável. É também denominado como falso negativo.

- Detecção por Abuso

  Esta abordagem busca identificar ataques com base em assinaturas pré-definidas. Um banco de dados armazena as assinaturas de ataques, as quais são formadas por uma sequência de eventos ou estados que consiste em uma tentativa de violação das políticas de segurança [Souza, Silva e Cansian 2002, Paula 2004, Machado 2005, Lima 2005].

  É um método mais simples e tradicional para detecções de intrusão. É semelhante aos procedimentos dos sistemas anti-vírus de computadores. Esta técnica tem a vantagem de uma detecção mais eficaz e rápida quando o padrão é conhecido, mas por outro lado é ineficaz com padrões desconhecidos ou mesmo em variações das assinaturas de ataques [Cansian 1997].

- Detecção Híbrida

  Um sistema que use uma abordagem híbrida combina a vantagem da utilização da detecção por anomalia, que busca encontrar comportamentos não usuais nos sistemas, detectando ataques desconhecidos, com a utilização da detecção por abuso, que utiliza as assinaturas de instrusão para encontrar os invasores. Esta união traz vantagens significativas, apesar do aumento da complexidade, tornando o SDI mais eficaz [Cansian 1997].

2.2.2.2 Arquiteturas de Detecção

A arquitetura dos SDIs se refere à forma com que os sistemas desenvolverão o trabalho para o qual foram destinados. É dividida de acordo com o alvo a ser analisado e a localização onde residirá o sistema.

- Segundo o alvo

  A detecção segundo o alvo diz respeito a que tipo de análise será feita para avaliar possíveis invasões. SDIs podem buscar rastros de invasores analisando informações em um *host*, nos dados que trafegam na rede ou em ambos [Barbosa e Moraes 2000], [Campello e Weber 2001], [Laureano, Maziero e Jamhour 2007].

  – Detecção Baseada em Rede

    Chamados de *Network Based Intrusion Detection System - NIDS* buscam detectar invasões ou suas tentativas através dos pacotes que trafegam na rede [Cansian 1997]. Isso é feito a partir de uma auditoria no conteúdo dos pacotes capturados em busca de comportamentos anormais. A captura é feita através de um *sniffer* estrategicamente posicionado em pontos da rede. Uma desvantagem deste método é a impossibilidade da análise dos dados que trafegam criptografados devido à incapacidade de compreensão das assinaturas de ataque. Outro problema é que cada vez mais são utilizados *switchs* nas redes, e eles não possibilitam a captura do tráfego em modo promíscuo pois chaveiam a comunicação apenas entre as máquinas proprietárias das informações.

  - Detecção Baseada em *Host*

    Os SDIs baseados em *host* são também denominados *Host Based Intrusion Detection System - HIDS*. A análise deste tipo de sistema é feita no próprio *host* onde se encontra, com base em dados gerados na própria estação. Os dados a serem analisados podem ser *logs* de sistema, dados de usuários, processos e serviços locais ou mesmo pacotes de rede destinados a esta máquina. Tem-se como vantagem uma maior facilidade para reações a invasões e a possibilidade de se detectar ataques em um ambiente de rede criptografado, devido à captura ser realizada depois da decifragem. Por outro lado possui a dificuldade de reagir a ataques ao próprio SDI, além da maior complexidade de configuração e manutenção.

  - Detecção Híbrida

    Sistemas de detecção que se utilizam desta abordagem buscam reunir o que cada uma tem de melhor. Assim podem tanto avaliar as *logs* de sistemas como o tráfego da rede, sempre procurando uma maior robustez para o SDI.

- Segundo a localização

  A classificação das arquiteturas de detecção segundo a localização se refere ao posicionamento dos componentes funcionais dos sistemas. Esta disposição dos componentes tem grande infuência no desempenho e nas garantias de funcionamento do SDI. Desta forma os sistemas podem ser organizados em três grupos: os centralizados, os hierárquicos ou parcialmente distribuídos e os distribuídos [Campello e Weber 2001, Raguenet e Maziero 2006].

  Um sistema centralizado em um *host* pode trazer vantagens para os desenvolvedores, devido a sua simplicidade na implementação, e para os operadores, possibilitando uma maior facilidade na instalação e configuração, além do desempenho que se obtém em relação a outras abordagens. Apesar disto, a busca por mais segurança e tolerância a falhas inviabilizam a utilização de modelos centralizados.

  Através de uma estrutura distribuída é posssivel garantir uma maior robustez que

a arquitetura centralizada, mas a busca pela segurança necessária através de criptografia, assinaturas digitais e o uso de técnicas de detecção de falhas oneram os sistemas e aumentam consideravelmente a complexidade de desenvolvimento.

Um arquitetura parcialmente distribuída visa encontrar um ponto de equilíbrio entre as duas outras abordagens. Módulos do sistema ficam distribuídos, mas interagem entre si através de uma forte relação hierárquica possibilitando mais facilidades para detecção de falhas. A desvantagem é que se um módulo do topo da cadeia de hierarquia falhar, compromete todo o sistema de detecção.

2.2.2.3 Comportamento Pósdetecção

O comportamento de um SDI, após uma detecção, pode ser passivo ou ativo. É considerado uma reação passiva quando o sistema apenas emite alertas para o administrador revelando o acontecimento e aguarda uma intervenção. Uma reação ativa responde automaticamente ao detectar uma atividade intrusiva.

2.2.2.4 Frequência de Uso

A frequência de uso dos sistemas de detecção se refere ao período de tempo em que ele fica ativado. Este período em que o SDI adquire e analisa dados pode ser contínuo, ficando o tempo todo em execução, ou pode ter intervalos de verificação pré-determinados, apenas realizando suas tarefas com base na configuração do administrador.

2.3 Segurança e Detecção de Intrusão Bioinspirados

Uma considerável gama de problemas computacionais têm sido resolvidos através de analogias com alguns conceitos e mecanismos da natureza. Pode-se citar como exemplos já consolidados os algoritmos genéticos, o comportamento das formigas, as redes neurais e os sistemas imunológicos.

Dentre os exemplos citados, os dois últimos paradigmas são amplamente utilizados para a resolução de problemas de segurança. No caso das redes neu-

rais artificiais a utilização se deve ao seu poder de aprendizado e generalização que ao ser aplicado na detecção de intrusão pode permitir a identificação de invasores através da observação de padrões que apenas se assemelhem aos padrões de violação de segurança.

Em relação aos SIAs, as vantagens existentes para a segurança computacional se devem ao fato de ter sido inspirado no SIH, o qual protege o corpo humano dos patógenos utilizando uma variedade de mecanismos de defesa.

2.4 Considerações Finais

O crescimento exponencial da Internet posssibilita um maior acesso a ferramentas de exploração e ataques com uma menor exigência de conhecimento, agregado a isto temos a constante necessidade da melhoria da segurança dos sistemas envolvidos na tentativa de evitar problemas.

Neste capítulo foram apresentados conceitos inerentes à segurança de redes de computadores e aos sistemas de detecção de intrusão. Dados que motivam o estudo desta área da computação também são mostrados aqui, além dos tipos e etapas de um ataque e da descrição das arquiteturas, estruturas e classificações dos SDIs.

Ao final deste capítulo foram apresentadas considerações sobre o estudo da segurança com inspiração biológica. A segurança bioinspirada em Redes Neurais e no Sistema Imunológico Humano.

No Capítulo 3, são apresentados os conceitos biológicos e seus respectivos estudos artificiais na computação, que são a base da conceituação para este trabalho. Esta descrição é realizada com base nos projetos desenvolvidos em [Lima 2005] e [Machado 2005]

Capítulo 3

Sistemas de Detecção Bioinspirados Analisados

Problemas computacionais têm sido resolvidos de maneira eficiente através de inspirações biológicas. Aplicações como gerenciamento de banco de dados, reconhecimento de padrões, detecção de vírus e modelos de segurança mais eficazes podem ser implementados usando estas técnicas.

A justificativa do uso das redes neurais, se deve ao seu poder de reconhecimento de padrões, aliado à capacidade de aprendizado e generalização. Uma RNA é uma grande estrutura altamente conectada de processamento paralelo e por isso tem a capacidade de executar tarefas de forma mais rápida. Os sistemas imunológicos têm a especial e complexa função de defesa do corpo humano, assim o aproveitamento de suas propriedades e formas de defesa contra agentes agressores é de grande relevância para a proteção computacional.

Neste capítulo são descritos os modelos de Sistema de Detecção de Intrusão bioinspirados analisados, além da conceituação dos processos biológicos e artificiais necessários ao entendimento do modelo. São analisados um SDI que tem como base das detecções uma Rede Neural Artificial e outro SDI que utiliza a ferramenta Logcheck para suas detecções.

3.1 Sistema de Detecção de Intrusão com Rede Neural

A literatura mostra que as redes neurais artificiais (RNA) têm tido relativo sucesso no reconhecimento de padrões. O diferencial é o seu uso para o aprendizado, generalização e reconhecimento de padrões intrusivos em protocolos de comunicação de redes de computadores.

O conceito das RNA's envolve técnicas computacionais desenvolvidas através de modelos matemáticos baseados na constituição estrutural do cérebro humano. Esta inspiração se deu pelo fato do cérebro possuir capacidades de processamento e organização poderosas e, principalmente, por ser responsável pelo comportamento inteligente do indivíduo. Assim supõe-se que reproduzindo suas características, pode-se extrair resultados inteligentes.

O uso destas redes tem se mostrado interessante e eficaz, pois permite tornar inteligente o comportamento de máquinas. E através desta inteligência artificial é possivel a detecção de intrusos sem a utilização de extensas e pesadas bases de assinaturas. Além disto, as informações a serem transferidas para a rede são mínimas, pois ela é capaz de aprender e generalizar conhecimento.

3.1.1 Inspiração Biológica para o Modelo

O sistema nervoso tem como unidade sinalizadora o neurônio, a célula especializada que possui diversos prolongamentos para recepção de sinais e um único para emissão de sinais. Portanto, sua função é de receber, processar e enviar informações [Lent 2001].

São células excitáveis com comunicação entre si e com outras células musculares e secretoras que utilizam como linguagem a eletricidade. Estes pulsos elétricos são propagados através das regiões do neurônio, que são: os dendritos, a soma ou corpo celular e o axônio, como podem ser observados na Figura 3.1.

Os dendritos são responsáveis pela recepção dos estímulos, transformando-os em alterações de potencial de repouso da membrana. O corpo celular é

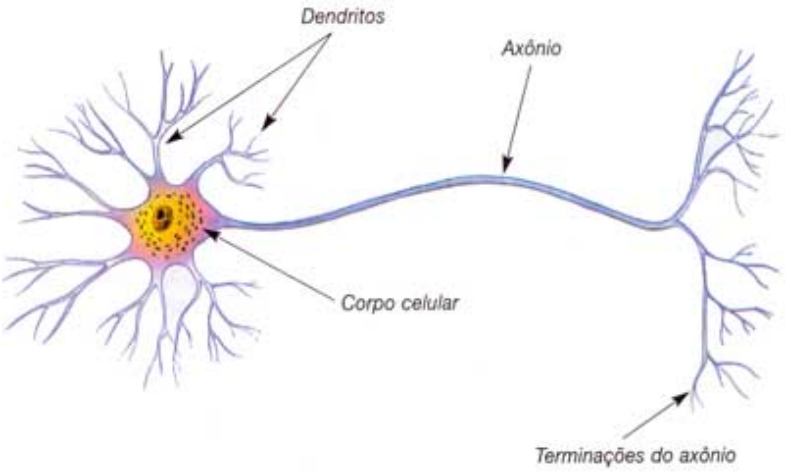

Figura 3.1: Neurônio Biológico. Adaptado de [Machado 2003]

onde se encontram os componentes comuns a todas as células, o que difere um neurônio das demais células animais é sua adaptação para processamento de informações. E por fim, o axônio que é responsável pela transmissão das informações para as outras células do circuito neural [Lent 2001, Machado 2003].

O local de contato entre um neurônio e outro é chamado sinápse, constituindo uma região especializada de processamento da informação que passa pelo sistema nervoso. Nesta região os sinais não são apenas transmitidos, mas transformados, podendo ser multiplicados ou bloqueados [Lima 2005].

Dentro da evolução do sistema nervoso pode-se distinguir três tipos de neurônios: neurônio aferente ou sensitivo leva ao sistema nervoso central (SNC) informações de modificações que ocorrem no meio interno e externo, normalmente situados junto ao SNC; neurônio eferente ou motor conduz os impulsos nervosos aos músculos ou glândulas, determinando assim uma contração ou secreção, podem se encontrar no sistema nervoso autônomo ou central e os neurônios de associação que são responsáveis pelos padrões de comportamentos mais elaborados do SNC [Machado 2003].

A passagem de informações entre um neurônio e outro depende da combinação de alguns fatores, como tipo e quantidade do neurotransmissor na

sinápse e sensibilidade da membrana dendrítica a excitações. Modificando-se a intensidade com que um neurônio pode excitar ou inibir outro neurônio, mudam-se os valores das conexões sinápticas e, desta forma, modifica-se o comportamento da rede. Isto significa um aprendizado da rede.

Aprendizagem e Generalização

Através do processo de aprendizagem obtém-se mudanças na habilidade de realização de tarefas novas que anteriormente não podiam ser realizadas, ou melhorar a realização de tarefas antigas. Nas redes neurais artificiais, o comportamento pode ser modificado em razão dos estímulos produzidos pelo ambiente. Desta forma, a força da conexão entre as unidades de processamento é regulada através da adaptação dos pesos sinápticos [Wasserman 1989].

O processo de aprendizagem é decorrente do treinamento da rede por meio da apresentação de padrões para as suas unidades. Assim, são atribuídos pesos sinápticos com valores apropriados, de forma que seja produzido um conjunto de saída adequado com o intervalo de erro estabelecido. Em resumo, todo o aprendizado consiste na busca dos pesos ideais através de aplicações de regras que definam a aprendizagem [Másson e Wang 1990].

O processo de ajuste dos pesos sinápticos, objetivando a adequação das saídas é obtido através do treinamento e pode acontecer de três formas. Sem treinamento, caso em que os valores são explicitamente determinados. Com treinamento supervisionado, através da apresentação dos conjuntos de entrada e os respectivos valores de saída esperados. Com treinamento não supervisionado, no qual apenas os conjuntos de entrada são apresentados para que as características destes conjuntos sejam extraídas. A partir destas características, os dados são organizados em classes. Este é um processo auto-organizável [Wasserman 1989, Másson e Wang 1990].

Outra característica importante das RNAs é o poder de generalização que possuem, a habilidade em generalizar sobre o domínio do problema. Sua sensibilidade a variações que possam ocorrer em suas unidades de entrada possibilitam o reco-

nhecimento de ruídos e distorções. Esta capacidade de adaptações às novas situações, gerando as respostas esperadas, é de vital importância para aplicação do modelo em ambientes do mundo real.

Aplicações

É grande a variedade de problemas que podem ser resolvidos através utilizando-se as redes neurais. Devido aos grandes esforços realizados em pesquisas, as características funcionais destas redes vêm sendo amplamente exploradas. Tarefas como o reconhecimento de padrões, a classificação, a transformação de dados, a previsão de séries temporais e a aproximação de funções podem ter sua resolução pelas RNAs.

A abordagem de reconhecimento de padrões visa a separação dos dados em classes ou categorias. Neste modelo, o sistema é treinado por meio de exemplos a fazer a identificação de determinados padrões. Após generalizar o conhecimento, ele é capaz de reconhecer novamente, mesmo que o exemplo apresentado não seja exatamente o visto no treino [Osório e Vieira 1999].

O reconhecimento e classificação de padrões por meio de inteligência artificial para a segurança computacional tem sido amplamente pesquisado nos últimos anos. Desta forma, tem sido possível otimizar os métodos convencionais de detecção.

Utilizando-se uma RNA é possível realizar detecções de intrusões com menos custos computacionais que os modelos de detecção convencionais, pois o conhecimento da rede neural se encontra nos pesos sinápticos e assim não se faz necessária a realização de comparações exaustivas entre as bases de assinaturas e padrões de comportamento.

Outra vantagem está no fato de que os SDIs tradicionais incorporam sistemas especialistas, os quais se baseiam em conjuntos de regras, e para a detecção de novos ataques é necessário constantes atualizações. Já com as redes neurais é possível se obter o reconhecimento de variações de ataques conhecidos devido o poder de generalização. Para a detecção de comportamentos completamente diferentes é necessário um re-treino da rede [Lima 2005].

3.1.2 Propriedades

O modelo de detecção de intrusão de [Lima 2005] tem como subsídio para as análises o fluxo da rede de computadores, sendo capaz de realizar a classificação das atividades normais e intrusivas com base nos eventos capturados. O sistema faz a análise por meio de uma rede neural.

Os eventos são capturados por meio de um *sniffer* com base nos filtros que o administrador estabelece. Após capturadas, as sessões passam por um análise prévia, onde são comparadas com uma base de conhecimento formada por *strings* de caracteres retiradas de sessões intrusivas. Ao final deste comparativo é gerado um vetor de estímulo para a rede neural com as palavras chave consideradas suspeitas.

A rede neural efetua a classificação dos eventos em normais e intrusivos a partir dos pesos sinápticos ajustados durante o treino. No treinamento é exposto para a rede neural ambos padrões e suas respectivas classificações para que ela possa extrair características e consequentemente aprender.

Este modelo de [Lima 2005] tem a classificação de sua arquitetura baseada em rede, utiliza métodos de detecção por abuso e anomalia, tem frequência de uso periódico pois necessita de intervenção do administrador para o início de seu uso e possibilita apenas reações passivas pois apenas informa o administrador sobre os eventos intrusivos.

3.1.3 Arquitetura

A arquitetura deste protótipo está modelada sobre os componentes E-BOX e A-BOX do padrão CIDF, pois a principal intenção é efetuar o reconhecimento dos padrões capturados na rede de computadores. Cada componente do padrão CIDF corresponde a um módulo da estrutura da solução proposta.

O primeiro módulo que é correspondente a caixa E-BOX está no nível mais baixo, e é onde são captados os eventos que transitam na rede. O módulo A-BOX tem uma maior complexidade pois é onde acontece toda a pré-classificação, ordenação e classificação final pela RNA. É dividido em sub-módulos, onde cada um tem responsabi-

lidades específicas.

O sub-módulo de classificação busca encontrar palavras-chave contidas nas sessões capturadas em concordância com a lista de base de conhecimento. É montada uma representação intermediária com informações do estabelecimento da conexão, as *strings* suspeitas e seus respectivos códigos binários.

O sub-módulo de Pós-Processamento é responsável pela formação do vetor de estímulos a partir da representação intermediária feita no sub-módulo anterior. A composição do vetor de estímulos depende de sua finalidade, caso a intenção seja treinar a rede, o vetor é formado com as representações binárias das sessões e as respostas do supervisor. Caso a necessidade seja analisar dados, o vetor formado conterá apenas as representações binárias.

Por fim tem-se o sub-módulo mais importante do modelo, o sub-módulo da rede neural. Neste componente que está a RNA, e ele é responsável pela indicação do nível de suspeita das sessões analisadas. Este nível pode ser ajustado pelo administrador no intuito de minimizar a incidência de falsos.

3.2 Sistema de Detecção de Intrusão com Logcheck

O protótipo de [Machado 2005], que foi utilizado nas análises, tem como ferramenta para detecção dos intrusos o Logcheck. Esta ferramenta verifica as *logs* em busca de atividade maliciosas através de comparações com palavras-chave e expressões contidas em seus arquivos e que foram gerados pelo Syslog-ng. De acordo com o modelo CIDF, o logcheck funciona como o analisador pertencente a caixa A-Box.

Durante o processamento, o que for considerado normal pelo Logcheck é descartado e tudo que não pertencer ao conjunto normal é catalogado em relatórios. Estes relatórios são divididos em ataques, violações de segurança e eventos de segurança, de acordo com os arquivos de configuração que são baseados nas políticas de segurança adotadas.

Neste modelo, [Machado 2005] utiliza uma inspiração biológica para definição dos seus componentes. Desta maneira é feito um comparativo com os sistemas

imunológicos humanos para possibilitar melhorias nos procedimentos de defesa computacional. Esta seção é destinada a mostrar os conceitos utilizados, as propriedades e a arquitetura do modelo em questão.

3.2.1 Inspiração Biológica para o Modelo

Sistemas imunológicos humanos (SIHs) protegem o corpo humano de agentes agressores externos, como bactérias, vírus e parasitas [Roitt, Brostoff e Male 1999, Schindler et al. 2005]. Sistemas computacionais necessitam de uma proteção semelhante contra seus agressores. Todo o processo de análise, detecção e reação imunológica abstraído dos SIHs tornaram possíveis a existência dos sistemas imunológicos artificiais (SIAs). Esta linha de pesquisa possibilita o desenvolvimento de sistemas de defesa computacional mais robustos, além de contribuir em outras áreas, como reconhecimento de padrões, detecção de vírus, detecção de falhas e detecção de anomalias em dados de séries temporais [Dasgupta et al. 1999].

3.2.1.1 Sistema Imunológico Humano

Por definição, imunidade é a propriedade de um ser vivo de ficar livre de determinadas perturbações. Com origem no Latim, immunitas, pode ser interpretado como direito ou vantagem concedida a alguém, devido a seu cargo. Pode também significar a tolerância a alguma coisa sem ser afetado por ela.

Historicamente, a primeira citação da relação entre imunidade e infecções pode ser atribuída a *Thucidides* na Grécia antiga. Mas o primeiro exemplo registrado de manipulação do sistema imunológico de forma controlada é atribuído a *Edward Jenner*, no fim do século XVIII. *Jenner* provou através da vacinação a possibilidade de se adquirir imunidade a uma infecção [Castro e Zuben 1999].

O poder de suportar agentes patogênicos sem desenvolver doenças, se deve ao sistema imunológico. Este sistema é uma complexa rede de orgãos, células e moléculas responsáveis pela defesa do organismo contra partículas estranhas, assim estão em constante busca a agentes agressores, para sua captura, identificação e eliminação.

Uma das principais características do SIH é a sua capacidade de distinção entre moléculas próprias (moléculas self) e adquiridas. Através desta capacidade o SIH pode distinguir o que não pertence ao corpo (moléculas nonself), fazendo desta forma o reconhecimento dos patógenos [Forrest et al. 1996, Machado 2005, Machado et al. 2005].

3.2.1.2 Imunologia Computacional

É possível observar na natureza uma grande diversidade de sistemas complexos formados por uma completa interação cooperativa e competitiva de elementos simples, possuindo como principal característica a robustez. Esta robustez se deve à capacidade de tolerância a pequenas perturbações justamente devido à simplicidade das tarefas executadas por cada elemento, além da existência de elementos redundantes.

Através destas definições é possível observar que certos sistemas biológicos possuem uma capacidade muito superior a de qualquer tecnologia atual. Assim, a observação e detalhamento dos princípios de comportamento destes sistemas possibilitaram o desenvolvimento de paradigmas computacionais eficientes. Entre os quais pode-se citar os sistemas imunológicos artificiais [Castro 2001].

Diversos trabalhos estão procurando pesquisar e aperfeiçoar os conceitos inerentes à imunologia computacional, além de suas aplicações. Uma interessante definição para os SIAs foi dada por [Dasgupta et al. 1999].

"*Os sistemas imunológicos artificiais são compostos por metodologias inteligentes, inspirados no sistema imunológico biológico, para a solução de problemas do mundo real.*" [Dasgupta et al. 1999].

As aplicações são variadas e em diversas áreas da computação, a seguir são citadas algumas delas: métodos computacionais inspirados em princípios imunológicos; SIA aplicados ao reconhecimento de padrões; sistemas baseados em imunologia para a detecção de falhas e anomalias; sistemas auto-organizados baseados em imunologia; métodos de busca e otimização baseados em imunologia; sistemas imunológicos para proteção contra vírus computacionais; mineração de dados [Machado 2005]; além da aplicação utilizada neste projeto, os sistemas imunológicos artificiais aplicados à seguran-

ça de redes e detecção de intrusão.

3.2.2 Propriedades

O protótipo de detecção de intrusão analisado em [Machado 2005] tem como base para as análises as *logs* geradas pelo sistema *Syslog-ng*. Estas *logs* são analisadas com o analisador de *logs Logcheck* em busca de sinais indicativos de atividades anormais, e todo o processo consiste em comparações entre as *logs* e palavras chave pré configuradas de acordo com as políticas de segurança adotada.

O *Logcheck* utiliza sua base de dados de palavras e expressões para gerar relatórios de acordo com o nível de periculosidade encontrado nas comparações. O conteúdo das *logs* que o *Logcheck* considerar como sendo normal é descartado, e o restante é separado em três tipos de relatórios, ataques, violações de segurança e eventos de segurança.

Ataques e violações de segurança são padrões conhecidos como anômalos pelo *Logcheck* e os eventos de segurança são ações desconhecidas e por isso não podem ser classificada como normais. Os relatórios produzidos no processamento do *Logcheck* servem de entrada para o sistema de agentes móveis que incrementam os padrões de reatividade do modelo [Machado 2005, Boukerche et al. 2007].

Os agentes móveis definidos em [Machado 2005] monitoram os arquivos gerados pela ferramenta *Logcheck* e viabilizam a distribuição e o armazenamento destes dados em estações seguradas e em sequência iniciam procedimentos de reação contra as situações críticas.

3.2.3 Arquitetura

O protótipo de [Machado 2005] tem arquitetura baseada em host, utiliza método de detecção por anomalia, tem frequência de uso continuo e permite respostas passivas e ativas.

O modelo possui implementações de todas as caixas do padrão CIDF, pois realiza as funções de captura, análise, armazenamento e reatividade previstas para

sistemas de detecção de intrusão. Como mostrado na subseção 3.2.2, a funcionalidade da caixa E-Box é realizada pela ferramenta de geração de *logs syslog-ng*. A caixa A-Box tem suas funções executadas pela ferramenta de análise de *logs Logcheck* no intuito de reconhecer violações e intrusões.

As caixas D-Box e C-Box são implementadas com a tecnologia de agentes móveis, que respectivamente realizam o armazenamento com a garantia da integridade e segurança das *logs* e a geração de respostas inteligentes.

3.3 Considerações Finais

Neste capítulo foram mostradas as características dos modelos estudados que possibilitaram a realização dos comparativos entre as duas tecnologias. Além de serem apresentados conceitos da biologia e suas respectivas aplicações computacionais importantes ao entendimento deste trabalho. Estas aplicações biológicas têm fundamental relevância quando aplicadas ao estudo da segurança de computadores devido a sua grande afinidade com a mesma. As redes neurais trazendo à computação a inteligência humana, possibilitando o reconhecimento de padrões e os sistemas imunológicos com a defesa do corpo semelhante a proteção às redes de computadores.

No capítulo 4 temos a apresentação da implementação do modelo estudado. São mostradas suas caraterísticas com base nas teorias descritas nos capítulos anteriores, além da modelagem com base na abstração biológica.

Capítulo 4

Protótipo de Detecção Bioinspirado

Nos capítulos anteriores, foram descritos importantes conceitos que nortearam o desenvolvimento do protótipo para um SDI bioinspirado. Com isso, assuntos relacionados à segurança de redes de computadores e SDIs, além dos conceitos biológicos necessários ao entendimento das analogias entre a computação e o corpo humano foram discutidos em seus respectivos capítulos.

Este capítulo descreve um protótipo baseado na aplicação de uma RNA como mecanismo de detecção em uma rede de computadores dentro da abordagem de [Machado 2005, Machado et al. 2005]. Assim tem-se um sistema de detecção de intrusão que utiliza como forma de detecção a RNA de [Lima 2005, Degaspari, Lima e Sobral 2008] e tem reações computacionais inteligentes utilizando reações análogas às naturais do Sistema Imunológico Humano. Para que se tornassem possíveis as reações do SDI na rede de computadores foi utilizada a plataforma de agentes móveis *Grasshopper*.

Similar aos sistemas imunológicos humanos que são responsáveis pela defesa do corpo humano contra ataques de invasores externos, o protótipo é um sistema imunológico artificial para defesa computacional, onde agentes móveis nos permitem combinar características desejáveis e necessárias de mobilidade e as redes neurais artificiais viabilizam o aprendizado e generalização, contribuindo com o aumento das detecções de intrusão [Boukerche et al. 2007].

O modelo de detecção projetado buscou implementar requisitos como a

identificação e análise de agentes invasores, memorização e distribuição dos resultados e a geração de respostas. É apresentado primeiramente apenas com as definições computacionais padrões dos SDIs. Em uma segunda parte, é definida e descrita as analogias com o corpo humano necessárias ao desenvolvimento da proposta de SDI bioinspirado.

4.1 Tecnologias Utilizadas

Trabalhos como [Machado 2005, Boukerche et al. 2007, Lima 2005], [Degaspari, Lima e Sobral 2008], desenvolvidos nesta linha de pesquisa, foram as bases para a definição tecnológica do presente modelo.

Para a implementação da caixa E-Box do padrão CIDF foi utilizado um sniffer de rede baseado no projeto *tcpflow* [Elson 2003]. A caixa A-Box é composta pela RNA desenvolvida no projeto de [Lima 2005]. Atendendo assim os requisitos do SDI proposto.

A implementação das caixas D-Box e C-Box, visando o atendimento aos requisitos de armazenamento, persistência e geração de respostas, tiveram seu desenvolvimento baseado em agentes móveis [Machado 2005]. Desta maneira, foi possível realizar analogias aos princípios do Sistema Imunológico Humano.

Para realização dos testes, foi necessário que dentro do segmento de rede observado houvessem três máquinas do projeto. A primeira delas, considerada o servidor, oferecia os serviços a serem monitorados. São eles: DNS, FTP, POP3, SMTP e HTTP. Este servidor possui as seguintes características:

- Processador INTEL *Pentium* III 1.0GHz, 256MB de memória RAM.
- Sistema Operacional *Linux*, *Kernel* 2.4.27, distribuição Debian Potato.

A segunda máquina, considerada o sensor de captura de tráfego, possui as seguintes características:

- Processador AMD *Athlon* XP 2.4GHz, 1,5GB de memória RAM.
- Sistema Operacional *Linux*, *Kernel* 6.08, distribuição *Ubuntu*.

- Máquina Virtual Java 1.5.0
- Plataforma de Agentes Móveis *Grasshopper* 2.2.4

A terceira máquina, a estação de gerência, tem as seguintes características:

- INTEL *Centrino* 1.6GHz, 1,5GB de memória RAM.
- Sistema Operacional *Microsoft Windows XP Professional*.
- Máquina Virtual Java 1.5.0
- Plataforma de Agentes Móveis *Grasshopper* 2.2.4
- C++ *Builder* 6

4.2 Método Estatístico Utilizado

A aplicação de um método estatístico visa a obtenção do grau de significância das diferenças obtidas entre os conjuntos amostrais. Este grau de significância deve ser observado dentro de um intervalo de confiança, o qual representa o percentual de amostras que pertencem ao intervalo em questão.

Com base nas características dos dados amostrais foi escolhido a teste exato de fisher, que busca calcular o *p-valor* testando a hipótese nula de que as frequências encontradas serão iguais a partir de pequenas amostras de dados.

O *p-valor* é a probabilidade de se encontrar uma diferença igual ou maior que a observada em uma pesquisa e quanto menor for o seu valor, menor é a chance de ter sido causada ao acaso. Nesta pesquisa foi utilizado a convenção de que um *p-valor* ≤ 0.05 mostra uma diferença significativa entre os conjuntos, por outro lado, um *p-valor* maior indica a inexistência de diferenças significativas [Filho 1999].

4.3 Modelo Computacional

Este modelo é a proposta de um SDI mais funcional, pois tem-se a retirada de ferramentas proprietárias como *syslog-ng e logcheck* [Machado 2005] para que o modelo tenha em substituição um intrumento de inteligência artificial. Com isso obtém-se uma evolução nesta linha de pesquisa.

O mecanismo responsável pela detecção dos eventos intrusivos passa a ser a rede neural. A análise é feita nos dados capturados do tráfego da rede, com o auxílio de um *sniffer* [Lima 2005]. Após a captura, estes fluxos são organizados em um arquivo texto que depois de tratado é entregue para a rede neural. Para que a RNA esteja capacitada a fazer detecções é necessário primeiramente treiná-la. Assim um conjunto contendo dados normais e outro com dados intrusivos são apresentados juntamente com sua classificação. Desta maneira, tem-se um aprendizado supervisionado.

Uma vez treinada, a rede neural é capaz de analisar quantos conjuntos de dados forem necessários. Fica sob responsabilidade do administrador retreiná-la quando perceber a existência de ataques inusitados.

O modelo tem como base a abstração de propriedades e processos do SIH [Machado 2005] que se utiliza da proposta arquitetural de SIAs para a segurança computacional [Somayaji, Hofmeyr e Forrest 1997]. Por meio destas abstrações obteve-se subsídios para o desenvolvimento dos componentes que possibilitam monitoramento, distribuição e persistência das informações além das devidas reações.

A seguir são mostradas as características do modelo baseado nas definições dadas na seção 2.2 (Página 17):

- Método de Detecção Híbrido - O método de detecção utilizado é considerado híbrido, pois incorpora as características de ambos os métodos para detecção de intrusos: abuso e anomalia. Neste modelo a detecção dos eventos intrusivos é o resultado propriamente dito da análise neural dos vetores de estímulo. Estes vetores são os resultados da formatacão dos fluxos de dados após a captura e pré-seleção dos mesmos na rede.

  A pré-seleção é a primeira classificação dos dados em busca de sessões suspeitas.

Esta análise acontece por comparações diretas com assinaturas de intrusão contidas em uma base de conhecimento. Desta maneira temos uma detecção por abuso.

Já a análise neural, que finaliza o processo de detecção, é capaz de encontrar atividades suspeitas em casos inéditos devido à generalização do conhecimento. Este processo acontece a partir de inteligência artificial, caracterizando o método de detecção por anomalia.

- Arquitetura Baseada em Rede e Distribuída:

  A busca e análise por sessões suspeitas é feita com base nos pacotes que trafegam na rede. Os sensores podem ser posicionados de forma estratégica para que se possa isolar determinados segmentos da rede analisada, ou até mesmo configurá-los para que sejam capturadas as sessões contendo serviços ou protocolos específicos. Com base nesta definição, o modelo possui arquitetura (segundo o alvo) baseada em rede.

  Em relação à arquitetura segundo a localização, pode-se dizer que o modelo é do tipo distribuído. Esta classificação se dá devido os computadores responsáveis pelo fornecimento de diversos serviços poderem estar espalhados pela rede. Desta maneira, os componentes de detecção, análise, armazenamento e de reação podem trabalhar distribuídos e de forma cooperativa sem manter uma relação hierárquica.

- Frequência de Análise Periódica

  Este protótipo é caraterizado pela detecção de intrusão com base na análise do fluxo de pacotes de uma rede de computadores. Para isso o modelo necessita capturar e organizar as informações primeiramente, deixando um intervalo de tempo entre as fases de captura e sua respectiva análise. Este modo de funcionamento é classificado como sendo baseado em um período de ativação, no qual o administrador precisa configurar os períodos de funcionamento de cada fase.

- Geração de Respostas Passivas e Ativas:

  As respostas são executadas quando o SDI classifica um evento que esteja acontecendo como sendo intrusivo. São definidas como passivas, as respostas que se limitam a informar ao administrador a ocorrência de um evento através de um *e-mail*, o

qual é disparado a partir de uma máquina segura. As respostas ativas visam agregar uma maior autonomia ao sistema. Isto significa possibilitar uma reação automática para assegurar a integridade da máquina invadida, e é implementada através de um agente móvel que interrompe o serviço não permitindo a continuidade do ataque. Estas respostas visam incrementar a característica pró-ativa ao modelo.

4.4 Arquitetura do Modelo Computacional

A arquitetura deste protótipo foi estruturada a partir da análise do modelo em [Machado 2005] visando a evolução e aperfeiçoamento. No modelo analisado já eram buscados os requisitos desejáveis em um SDI, com base na padronização CIDF (subseção 2.2.1, página 18). Diferentemente do modelo desenvolvido em [Machado 2005], cujas geração e análise de eventos eram realizadas por ferramentas proprietárias (*Syslog-ng e Logcheck*), tem-se aqui a inclusão de um *sniffer* de rede e uma RNA.

A utilização de RNAs para a realização das detecções de intrusos em SDIs é interessante devido à forma de representação do conhecimento, que se dá através dos pesos das conexões sinápticas e não com base em regras ou assinaturas de intrusão. Desta forma, o espaço necessário ao armazenamento deste conhecimento é mínimo e invariável. Estes pesos têm seus valores ajustados durante o treinamento da rede neural.

O restante deste modelo continua a implementar persistência, robustez, disponibilidade e confiabilidade conforme experiência discutida em [Machado 2005]. Desta forma finaliza a utilização dos componentes da padronização CIDF e pode ser observado todo o fluxo da realização das tarefas a partir da Figura 4.1.

É possivel observar no topo esquerdo da Figura 4.1 os serviços analisados(*FTP, POP, HTTP, SMTP E DNS*) a partir do fluxo de dados capturados na rede de computadores. Nesta rede analisada se encontra o *sniffer* responsável pela captura dos eventos, o qual realiza a função das *E-Boxes* do padrão CIDF. Abaixo da rede na Figura 4.1 encontram-se os componentes que atuam com a funcionalidade das *A-Boxes*, em especial cita-se a rede neural artificial [Lima 2005]. As *D-Boxes* e as *C-Boxes* ainda são implementadas por um sistema de agentes móveis que possibilitam a segurança e integri-

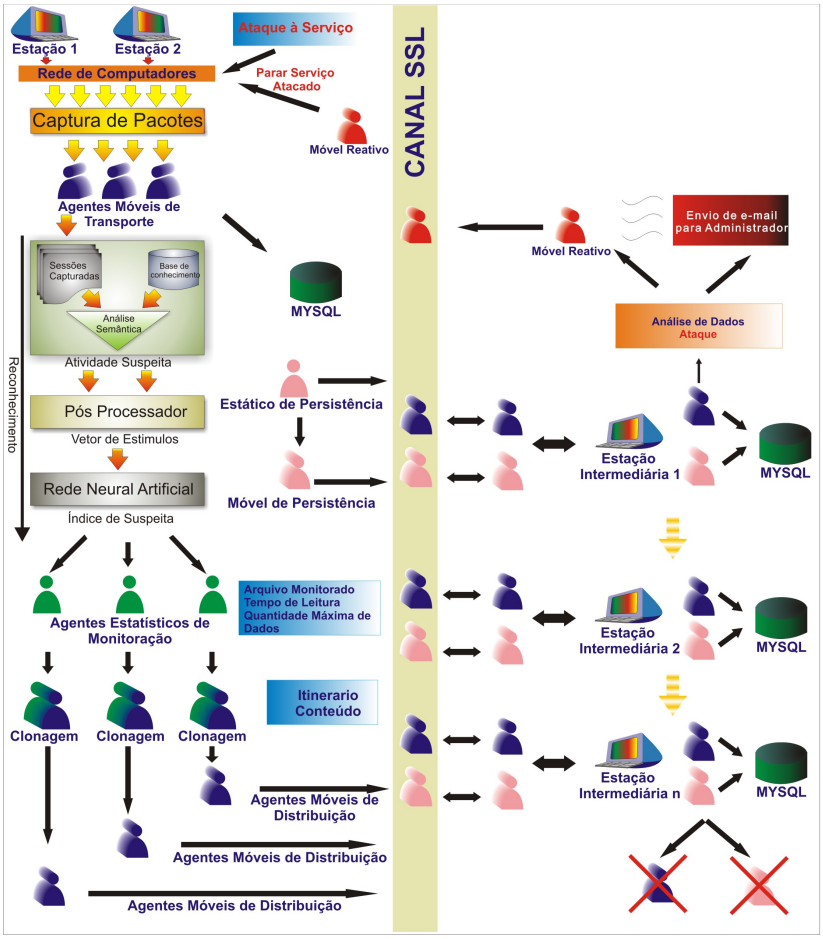

Figura 4.1: Arquitetura do Modelo Computacional. Adaptado de [Machado 2005]

dade das informações, distribuição e persistência dos dados, além de garantir a existência das respostas computacionais [Machado 2005].

4.4.1 Serviços Monitorados

Visando um estudo com a maior abrangência possível para os experimentos deste trabalho, observou-se os serviços mais utilizados na maioria dos ambientes computacionais. Foram analisados protocolos de serviços de acessos a *web sites*, de envio e recebimento de *e-mail*, de transferência de arquivos e de troca de informações sobre domínios e servidores na Internet.

A seguir, tem-se uma breve descrição sobre os serviços verificados por

este protótipo de SDI.

- *Hyper Text Transfer Protocol* - O HTTP é o protocolo utilizado em toda a *Web* e que especifica as mensagens trocadas entre os navegadores e os servidores. As solicitações são do tipo ASCII, as quais recebem uma resposta semelhante ao MIME (Multipurpose Internet Mail Extensions). O estabelecimento do contato entre as partes normalmente se dá por uma conexão TCP na porta 80.

  Tabela 4.1: Métodos internos de solicitação HTTP. Fonte: [Tanenbaum 2003].

Método	**Descrição**
GET	Solicita a leitura de uma página Web
HEAD	Solicita a leitura de um cabeçalho de uma página Web
PUT	Solicita o armazenamento de uma página Web
POST	Acrescenta a um recurso (por exemplo, uma página Web)
DELETE	Remove a página Web
TRACE	Ecoa a solicitação recebida
CONNECT	Reservado para uso futuro
OPTIONS	Consulta certas opções

  Na versão 1.0 do protocolo HTTP, apenas eram trocadas uma mensagem e sua resposta e então era encerrada esta conexão. Para solucionar o problema de excesso de conexões necessárias ao transporte de todos os ícones e imagens contidas em um página foi lançado o HTTP 1.1, que possibilita conexões persistentes.

  As solicitações consistem em uma ou mais linhas de texto ASCII, em que a primeira palavra da primeira linha é o nome do método solicitado. Na Tabela 4.1, pode-se observar os métodos internos de solicitações HTTP. A maioria das solicitações a servidores tem a forma de métodos *GET*, o qual é a solicitação de envio de uma página ou objeto.

- *Simple Message Transfer Protocol* - SMTP é o protocolo padrão para envio de *e-mails* na *Internet*, no qual o servidor opera sobre a porta TCP 25. É relativamente

simples, e seu funcionamento se dá a partir do estabelecimento da conexão entre a máquina transmissora, que opera como um cliente, e a receptora como um servidor. Em seguida, é esperado que o servidor forneça sua identidade e informe se está preparado para o recebimento de mensagens.

O aceite do servidor, possibilita que o cliente se identifique, anuncie a origem e o destino desta mensagem. Caso o destinatário exista no receptor, o servidor permite que o cliente envie a mensagem. Este modelo funcionou bem até determinado momento na evolução da *Internet*, quando as máquinas transmissoras e receptoras estavam o tempo todo *on-line*. Com o crescimento e a possibilidade de que um usuário, destinatário de uma mensagem, estivesse desconectado da *Internet*, foi necessária a criação de agentes de transferência de mensagens em máquinas servidoras constantemente ligadas. Desta forma, o *e-mail* é passado a este intermediário que espera conexões na porta TCP/IP 25 [Tanenbaum 2003].

Este agente de tranferência aguarda que o real destinário da mensagem entre em contato e estabeleça uma conexão para ter seu correio eletrônico entregue.

- *Post Office Protocol version 3* - O POP3 é o protocolo utilizado para o acesso remoto a uma caixa de e-mail e por meio dele as mensagens contidas em uma caixa de correio possam ser sequencialmente transferidas para um computador pessoal. O POP3 não tem como intenção prover operações de manipulação complexas no servidor, normalmente após a transferência as mensagens são apagadas.

  O servidor inicia o serviço POP3 escutando a porta TCP 110. Assim que um cliente deseje usar o serviço, ele solicita um estabelecimento de conexão TCP. Após o estabelecimento, a conexão passa por três estados. A autorização se refere à validação do nome de usuário e senha. A transação realiza a transferência de uma cópia das mensagens do servidor para a máquina do usuário, além de marcá-las para, ao final desta etapa, apagá-las. E por fim, o estado da atualização exclui as mensagens do servidor [Tanenbaum 2003].

- *File Transfer Protocol* - FTP é um serviço para transferência de arquivos entre com-

putadores. Funciona de acordo com o modelo cliente e servidor, além de ser o protocolo de transferência de arquivos mais utilizado na Internet, com uma forma de transferência bem rápida e versátil. É o protocolo padrão da pilha TCP/IP para estas transferências, apesar de ser anterior ao TCP e ter passado por adaptações para se tornar compatível. É genérico e independente de hardware e sistema operacional.

A conexão acontece entre um cliente e um servidor e se dá após a confirmação de dados passados pelo utilizador. Este deve passar um nome de usuário e senha, além do endereço IP correto de um servidor. O acesso acontece através de uma porta, normalmente a porta 21, e pode ocorrer através de uma interface ou linha de comando. No segundo modo, apesar de mais complicado, está presente em qualquer distribuição *UNIX-like* ou *Windows*.

- *Domains Name System* - O DNS é um protocolo existente na camada de aplicação utilizado para dar suporte necessário a fim de que certas aplicações funcionem. Ele cuida das nomenclaturas na *Internet*.

  A criação do DNS se deve, principalmente à dificuldade de memorização dos endereços binários de redes pelas pessoas. E, além disso, caso um domínio tenha seu servidor mudado para uma máquina diferente com outro endereço IP, seria necessário toda a divulgação para os acessos com o novo endereço IP. Assim foram introduzidos nomes em *ASCII* para que exista uma separação dos nomes das máquinas e seus respectivos endereços. A partir disso, outro problema surgia devido à rede apenas reconhecer endereços numéricos. Assim se fez necessária a criação de um mecanismo de conversão de *ASCII* em endereço de rede numérico.

  A base do funcionamento do DNS é um esquema hierárquico de atribuição de nomes com base no domínio e de um sistema de banco de dados distribuídos para a implementação do esquema de nomenclatura. Em suma, para o mapeamento do nome em endereço IP, o programa aplicativo invoca um procedimento chamado “resolvedor” e entrega o nome como parâmetro. Este procedimento passa os dados ao servidor DNS local, que procura e responde o endereço IP correspondente. Por fim, o “resolvedor” retorna o endereço IP ao aplicativo que fez a chamada, que de posse

deste endereço pode estabelecer uma conexão TCP.

4.4.2 Captura dos dados

Os dados que trafegam na rede, quando pertencem a aplicações que não se utilizam de técnicas de criptografia, podem ser facilmente capturados e entendidos. Esta possibilidade, aliada a técnicas de pré-seleção com base em combinações de filtros, permite que estes fluxos possam ser posteriormente analisados.

A captura do fluxo da rede é possível através de um *sniffer*, o qual na abordagem de [Lima 2005] foi desenvolvido a partir do projeto *tcpflow* [Elson 2003], que dispõe de diversos recursos para captura de pacotes. Recursos tais, que permitem a geração de um arquivo contendo a atividade da rede, devidamente mapeada e formatada com base nos filtros utilizados.

Para que seja possível o acesso aos pacotes do tráfego é necessário que a rede esteja operando em modo *broadcast*, ou seja, os ativos de rede permitam que todos os pacotes sejam replicados a todas as estações. Desta maneira, a estação de captura não precisa estar envolvida diretamente na conexão a qual a atividade monitorada pertença. Além disso, a interface de rede da estação de captura deve estar em modo promíscuo (forma de funcionamento da interface onde é retido todos os pacotes que passarem pela rede, mesmo os que não forem destinados a esta estação).

O tratamento inicial dos pacotes capturados consideram certos parâmetros em determinadas áreas do datagrama. São analisados cabeçalhos IP, TCP e o *payload*. Desta forma, tem-se os subsídios necessários aos tratamentos posteriores. Para a composição da representação do fluxo da rede, primeiramente é gerado um arquivo para cada sessão e sentido do fluxo de dados [Lima 2005].

Ao término do processo de captura é realizado um processamento em todos arquivos obtidos com a finalidade de organizar e então gerar um único arquivo contendo toda a comunicação bidirecional da rede.

Neste arquivo, as sessões são organizadas sequencialmente e demarcadas no início e fim com a repetição de caracteres “-”. Como pode ser observado na Tabela

Tabela 4.2: Sessão FTP não intrusiva.

Sessão não intrusiva na porta 21

```
------------------------------------------------------------
#####_I-IDS_#####
TCP 10.1.1.12:37201 -> 192.168.2.1:21
#####_I-IDS_#####:5:
I-IDS<-220 ProFTPD 1.3.0 Server (Debian) [::ffff:192.168.2.1]
I-IDS->USER joelson
I-IDS<-331 Password required for joelson
I-IDS->PASS joelson@gmail.com
I-IDS<-530 Login incorrect...
I-IDS->SYST..
I-IDS<-530 Please login with USER and PASS...
I-IDS->QUIT..
I-IDS<-221 Goodbye...
I-IDS:END
------------------------------------------------------------
```

4.2, cada sessão tem em seu cabeçalho informações relevantes às posteriores análises. Na sessão em questão, pode-se verificar que representa a comunicação TCP com origem no *host* com endereço IP 10.1.1.12 na porta 37201 e destino o *host* com endereço IP 192.168.2.1 na porta 21. A direção do fluxo é mostrada pelos caracteres "`->`".

A sessão observada é a quinta sessão do fluxo de dados e isto pode ser visto no número 5 que consta entre os caracteres ":" na quarta linha. A direção do fluxo é dada pelo marcador `I-IDS->`, representando que o fluxo ocorreu no sentido de quem originou a sessão para o endereço destino. Por outro lado, quando os eventos forem precedidos pelo marcador `I-IDS<-`, o fluxo foi gerado na máquina destino em direção

à origem da sessão. A sessão tem seu encerramento com o marcador `I-IDS:END`.

As sessões capturadas são transportadas de forma segura através de *ssl* da máquina que contém o sensor para a estação de gerenciamento, utilizando para isto agentes móveis. Assim, além da automatização da transferência das sessões capturadas, mais um elemento no SIA é implementado.

4.4.3 Análise do Tráfego

Todo o processo descrito na seção 4.4.2 representa a implementação de uma caixa E-Box do padrão CIDF [Kahn et al. 1998], ficando responsável pela captação dos eventos ocorridos na rede analisada. Em sequência, para realizar a representação da caixa A-Box, tem-se a implementação de um modelo de análise neural [Lima 2005], capaz de identificar atividades intrusivas com seus respectivos níveis de suspeita. Os dados são previamente tratados no intuito de buscar assinaturas de intrusão já conhecidas, converter estes dados para binário e preparar o vetor de estímulos esperado pela RNA.

A busca por assinaturas de intrusão é uma forma de realizar uma separação inicial entre as sessões com características em comum. Esta organização se dá com o auxílio de um arquivo contendo uma lista de palavras. Normalmente estas palavras quando utilizadas de forma isolada não representariam problema algum, mas as combinações entre elas aliadas a outros parâmetros podem mostrar diferentes maneiras de violação. Em busca da atualização deste arquivo durante o desenvolvimento deste projeto, foram adicionadas palavras encontradas no campo *content* das últimas atualizações da base de regras do SDI *snort*.

Para possibilitar um melhor treinamento e, consequentemente, resultados mais confiáveis nas análises da RNA, a conversão das sessões para binários devem seguir um importante princípio. Caso as sequências de bits que representem as palavras não tenham uma diferença em um número mínimo de *bits*, a rede neural pode ser induzida a aprender de forma errada a característica da sessão. Assim, os códigos binários que representam cada categoria possuem uma *distância de hamming* entre si, possibilitando que todas as representações tenham o mínimo de diferença.

Tabela 4.3: Representação binárias dos protocolos

Protocolos
TCP 0101010101010101
UDP 1010101010101010

Tabela 4.4: Representação binárias das portas

Portas
21 0000000011111110
25 0011001100110010
53 0011001111001101
80 0011110011000010
110 0101010110101011

Como conteúdo, cada arquivo de apoio possui as representações necessárias às devidas conversões binárias.

Como pode ser observado na Tabelas 4.3 e 4.4, as representações binárias respectivas dos protocolos e portas analisadas são suficientemente diferentes umas das outras. A partir das conversões, é feita uma varredura no arquivo de sessões em busca de combinações específicas que tornem a atividade suspeita.

Esta busca obtém como resultado um arquivo com sessões devidamente tratadas e preparadas para que sejam gerados os vetores de estímulo da rede neural. No tratamento, as combinações das palavras-chave encontradas são separadas e catalogadas em sequência, além de suas representações binárias.

Tabela 4.5: Sessão FTP não intrusiva após tratamento.

```
----------------------------------------------------------------
#2:5
#TCP
#21
#PASS
#Login incorrect
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0
0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 0
0 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
----------------------------------------------------------------
```

Na Tabela 4.5, pode ser vista a sessão exemplificada na Tabela 4.2, já tratada. Apesar de ser uma sessão não intrusiva, a busca por *strings* suspeitos identifica alguns comandos isolados. Estas sessões possuem dados relevantes à construção do vetor de estímulos.

No início, as descrições são precedidas pelo identificador #, significando que são dados retirados das sessões originais. Na primeira linha é mostrado `#2:5`, representando que foram encontradas 2 *strings* suspeitas e que esta é a quinta sessão analisada. Na segunda e terceira linha, são dados o protocolo e a porta, respectivamente, utilizados na sessão em questão. Nas próximas linhas, precedidas pelo identificador, são listadas as palavras encontradas.

A seguir, vêm as representações binárias do protocolo, porta e palavras-chave encontradas. Cada sessão tratada pode ter 16 sequências de 16 *bits*, desta forma, pode armazenar até 14 combinações de palavras suspeitas. Caso a sessão tenha menos palavras suspeitas, como no exemplo da Tabela 4.5, são completados com a sequência `0000000000000000`.

A partir das representações binárias é montado o vetor de estímulos em um segundo arquivo, o qual possui em sua primeira linha o número de sessões a serem analisadas. Em seguida o número de entradas por sessão, e após, separados linha por linha, está cada *bit* de todas as representações binárias das sessões. Assim o vetor de estímulos possui 256 entradas para cada sessão.

O vetor de estímulos é gerado da mesma forma caso a intenção seja o treinamento da RNA. A diferença é que para cada sessão é perguntado ao supervisor a classificação do padrão, intrusivo ou não. Ao informar, a resposta é anexada ao final das 256 entradas da sessão, 1 para as intrusivas e -1 para as sessões normais.

4.4.3.1 Rede Neural

Após preparadas, as sessões podem realmente ser analisadas pela rede neural e assim informados os níveis de suspeita de cada uma. O principal motivo para utilização de uma ferramenta de inteligência artificial como as RNAs para detecção de

intrusão, é a possibilidade de se conseguir o aprendizado a partir de uma certa quantidade de assinaturas de intrusão conhecidas e assim obter a generalização do conhecimento para as próximas análises.

Outra vantagem está no fato de que o conhecimento armazenado pela rede neural se encontra nos pesos sinápticos ajustados durante o treinamento. Assim o tamanho do arquivo que o armazena é mínimo e invariável. Esta característica torna as RNAs mais interessantes que os sistemas de detecção de intrusão baseados puramente em regras, pois estes últimos necessitam de constantes atualizações além de um maior espaço necessário para o armazenamento.

Com base na formação do vetor de estímulos, que possui 256 bits de entrada e a possibilidade de classificação da saída ser apenas 2, intrusiva ou não intrusiva, a rede utilizada tem 256 neurônios na camada de entrada, 21 na camada intermediária e 1 na camada de saída. Os valores de saída desta rede variam de -1 a 1. Assim na configuração inicial, os valores no intervalo de -1 a 0 representam respostas caracterizando sessões normais e os valores no intervalo de 0 a 1 representam as sessões intrusivas.

A interface de análise permite que um ajuste seja feito pelo administrador para modificar o índice que leva a rede a classificar os padrões. Modifica-se assim o limiar de separação entre o classificado como intrusivo ou não intrusivo. Desta forma, é minimizada a ocorrência de falsos positivos ou negativos.

A rede utilizada no protótipo foi uma *feedforward* do tipo *multilayer perceptrons* com algoritmo de treinamento *backpropagation* (retropropagação de erro). A função de ativação é a função tangente hiperbólica que possibilita as saídas no intervalo de -1 a 1.

A fase de treinamento tem importância primordial e influência direta na qualidade dos resultados das análises. Por este motivo, na interface de gerenciamento é possível realizar diferentes configurações, como a quantidade de épocas a serem realizadas ou um valor de erro quadrático médio a ser alcançado, e isso pode ser acompanhado por gráficos que mostram a curva de aprendizado ou o ponto de convergência do processo de treinamento [Lima 2005].

Neste modelo desenvolvido, além possibilitar que as sessões intrusivas

sejam detectadas, tenham suas devidas reações realizadas e sejam armazenadas para posteriores estudos de comportamentos, o conhecimento pode ser replicado a outras estações de análise na mesma rede ou em diferentes segmentos de rede. Sendo assim, uma vez treinada a rede e de posse do arquivo de pesos sinápticos, os agentes móveis podem permitir que a análise seja feita em diferentes sessões e por diferentes estações de gerência ao mesmo tempo. Compartilhando assim o que foi aprendido.

4.4.4 Persistência e Reatividade do Modelo

A persistência das informações intrusivas e a reatividade incorporada ao modelo são implementadas pelos agentes móveis. São definidas em [Machado 2005], as características das quatro classes diferentes de agentes: agentes de monitoração, distribuição, reativos e de persistência. A plataforma de agentes utilizada na implementação é a *Grasshopper 2.2.4*. São mostrados também o método de mobilidade e o gerenciador de banco de dados utilizados.

Com isto, tem-se a totalidade da analogia entre os componentes e a definição dada no padrão CIDF, com as caixa D-Box para a persistência e C-Box para reatividade. Agentes estáticos monitoram a captura do tráfego da rede e instanciam agentes móveis para transportá-las de forma segura até a estação de gerência. As sessões são tratadas e, em seguida, analisadas pela Rede Neural Artificial.

Outros agentes de monitoramento aguardam o final da análise para, ao encontrar sessões intrusivas, instanciar novos agentes móveis que as transportam e registram em outras máquinas confiáveis. São instanciados também agentes móveis responsáveis pela reações passivas e ativas do modelo. Nas reações passivas, os agentes enviam *e-mails* aos administradores. Nas ativas, percorrem a rede até a máquina que está sofrendo agressão e param o serviço atacado. Estes agentes móveis reativos também são capazes de inserir regras no *firewall* que impeçam a utilização dos serviços a partir de endereços *IP* de atacantes.

A arquitetura dos agentes é apresentada na Figura 4.2, na qual pode ser observada a disposição dos seus componentes. A Região SIA representa todo o domínio

da aplicação e é composta por uma Agência de Origem, várias Agências de Destino, Canal de Comunicação e diferentes Classes de Agentes.

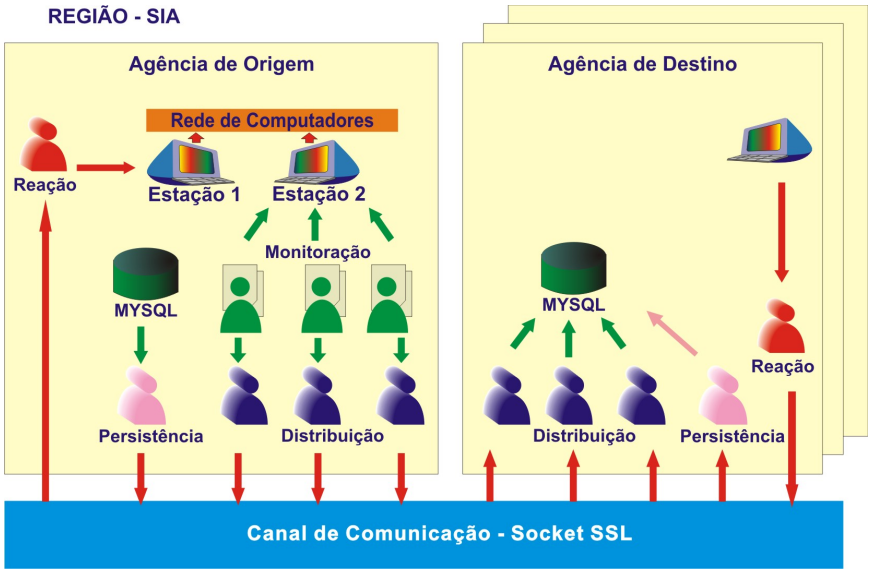

Figura 4.2: Arquiteturas dos agentes. Adaptado de [Machado 2005]

A Agência de Origem representa a estação que possui o sensor. Ela é composta por agentes estáticos de monitoração, agentes móveis de distribuição e estáticos de persistência. As Agências de Destino se encontram em máquinas seguras. São nelas que os agentes móveis provenientes da Estação de Origem registram as sessões intrusivas. É na primeira estação considerada segura que acontece a análise dos dados, armazenamento das informações na base de dados e são iniciados, se necessário, a replicação de armazenamento de sessões intrusivas, além das diferentes formas de reatividade. O canal de comunicação seguro (*socketssl*) é por onde os agentes móveis são transportados e visam a obtenção dos requisitos de segurança.

Com as definições dadas anteriormente, é caracterizado o sistema de detecção de intrusos a que se refere este projeto. Um SDI com arquitetura baseada em rede e distribuída, utilizando o método de detecção por abuso e anomalia, sendo executado em períodos de tempo pré-definido e com reatividades passivas e ativas. Assim, tem-se obtenção dos requisitos funcionais das caixas do padrão CIDF, geração de eventos, análise, armazenamento e reatividade. Estes componentes computacionais tiveram como

base os princípios abstraídos do Sistema Imunológico Humano.

4.5 Modelo Biológico Artificial

A linha de pesquisa seguida neste trabalho tem sido amplamente pesquisada [Mukkamala e Sung 2003, Yu et al. 2006, Boukerche et al. 2007, Mo, Ma e Xu 2008, Degaspari, Lima e Sobral 2008], pois tem uma forte correlação com o sistema imunológico humano. Sendo assim, problemas com segurança computacional tem suas soluções encontradas nas analogias com o SIH, possibilitando detecções de intrusos (`patógenos`) e reações inteligentes (`respostas imunológicas`).

No modelo descrito por [Machado 2005], foram estabelecidas metáforas com o SIH que buscavam implementar um sistema em camadas semelhante à proteção humana. Nele são comparados todos os mecanismos relacionados à controle e proteção computacional com as funcionalidades naturais de defesa do `corpo`.

Neste trabalho, procurou-se seguir as mesmas definições das analogias com os conceitos imunológicos. No entanto, algumas alterações foram feitas para a evolução do modelo. A captura do tráfego da rede passa a fazer parte da definição do sistema imunológico *inato*. No sistema imunológico *adaptativo*, tem-se a inclusão de uma RNA como ferramenta de inteligência artificial aplicada à análise e reconhecimento de `patogenias`, distinguindo entre `antígenos self` e `nonself` e possibilitando que haja armazenamento da memória imunológica e geração de respostas.

Para fazer o reconhecimento nos padrões, o `organismo` (rede de computadores) através dos `macrófagos` (agente móveis), entra em contato com `antígenos self` e `nonself`, os quais são diferenciados pelas `células T-Helper` (Rede Neural). As sessões capturadas da rede são consideradas os `antígenos`.

4.6 Considerações Finais

Neste Capítulo foram apresentadas as definições computacionais no modelo implementado e suas analogias com o SIH. Foi possível observar todo o con-

texto do trabalho. No Capítulo 5, são apresentados e discutidos os resultados dos testes e análises.

Capítulo 5

Discussão dos Resultados

No Capítulo 4, foram definidas as soluções tecnológicas que formam o protótipo, com base nas abstrações feitas do SIH para a incrementação no modelo de SIA definido por [Machado 2005]. Sendo assim, foi possível ter uma evolução no atendimento aos requisitos básicos de um SDI.

Neste Capítulo, são realizadas as análises de experimentos para a detecção de intrusos com a utilização da RNA de [Lima 2005] inserida no modelo de SIA baseado em agentes móveis. As análises executadas buscaram a minimização de falsos positivos e negativos.

A captura das sessões foi realizada em um ambiente computacional corporativo, utilizando suas políticas de segurança, perfis de usuários e os níveis de controle. No intuito de otimizar os filtros de protocolos e portas escolhidas para os testes, foi isolado um segmento de rede pertencente a um departamento. O tráfego normal capturado é resultante da utilização da rede por quinze usuários. O tráfego intrusivo é resultante de simulações de comportamento hostil por meio de ferramentas próprias para esta finalidade. Os tráfegos normal e intrusivo foram capturados através do sensor que monitorava este segmento da rede.

Após capturados os comportamentos, foi possível converter para binário, treinar e analisar com a rede neural. Para realizar os testes foram formados dois conjuntos, ambos divididos em dados de treino e de teste.

Os dados obtidos com ambas tecnologias podem ser analisados estatisticamente, no intuito de se obter o grau de significancia das diferenças dos conjuntos de dados. Os comparativos foram realizados por protocolos para que fosse possível a verificação do impacto de detecção de intrusão por ferramenta em cada serviço analisado. Tabelas e gráficos demonstrativos ilustram os resultados destas comparações ao longo deste capítulo.

5.1 Formação das Assinaturas

A composição do conjunto de treinamento precisa ser com ambos padrões, metade do conjunto contendo sessões normais e a outra metade com sessões intrusivas. Isto se deve à necessidade de não permitir que o aprendizado da rede neural seja influenciado por um dos comportamentos.

Os padrões não intrusivos foram capturados a partir da utilização normal da rede de computadores por meio de monitoramento de tarefas, como acesso a páginas *web*, envio e recebimento de *e-mail* e troca de arquivos por FTP. Durante o processo de captura deste tipo de comportamento foi tomado o cuidado para observar que estes dados não possuíam nenhum nível de periculosidade.

Durante a captura das sessões intrusivas foram utilizadas ferramentas de varredura, exploração de vulnerabilidades, promoção a super usuário e instalação de *rootkits*, como SAINT [SAINT 2004], NESSUS [Deraison 2004], além do sistema operacional BACKTRACK [BackTrack 2007], cujo foco são os testes de penetração. Este processo foi realizado com o isolamento das estações envolvidas para garantir que o único tráfego de dados existente na rede no momento era intrusivo.

Foram catalogadas 288 sessões distintas, sendo 144 sessões com comportamento normal e 144 com comportamento intrusivo. A Tabela 5.1 mostra a divisão entre ambos os padrões em relação ao protocolo utilizado. A diferença entre as quantidades de sessões de uma porta para outra se deve às semelhanças encontradas nas sessões, pois alguns protocolos trabalham com pouca diversidade de comandos. Resultando, muitas vezes, apenas um padrão utilizável em meio a diversas sessões capturadas.

Tabela 5.1: Quantidade de Sessões Utilizadas.

Porta	Sessões Intrusivas	Sessões Normais	Total
21	40	40	80
25	12	12	24
53	15	15	30
80	70	70	140
110	7	7	14
Total	144	144	288

Um maior detalhamento é dado na Tabela 5.2, na qual ficam expostas as quantidades de padrões, normais e intrusivos, que foram utilizados para os processos de treinamento e teste. Por meio desta tabela é possível ver quantos padrões de determinado comportamento em determinada porta foi usado para treinar ou testar a rede.

A divisão dos conjuntos foi realizada aleatoriamente, visando novamente a não indução de aprendizado de dados específicos. Foram utilizados cerca de 75% dos padrões capturados para realizar o processo de treinamento, visando incorporar o conhecimento na rede neural. Os 25% restantes foram usados para efetuar os testes que possibilitaram mensurar o quanto a rede foi capaz de aprender e de generalizar.

Para realizar um comparativo que permita verificar as diferenças no desempenho da rede durante situações adversas, foram formados dois conjuntos de dados com diferentes organizações a partir das mesmas 288 sessões. Esta organização também ocorreu de forma aleatória para evitar a formação de grupos de dados tendenciosos.

5.2 Treinamento

O processo de treinamento possibilita que a rede neural adquira conhecimento sobre determinadas tarefas, como visto na seção 3.1.1, página 29. Isto acontece neste modelo de RNA por meio da retropropagação de erros, que a cada ciclo ajustam

Tabela 5.2: Divisão das Sessões para Treinamento e Testes.

	Treinamento			***Testes***			
Porta	**Ataque**	**Normal**	**Total**	**Ataque**	**Normal**	**Total**	**Total**
21	30	30	60	10	10	20	80
25	9	9	18	3	3	6	24
53	11	11	22	4	4	8	30
80	53	53	106	17	17	34	140
110	5	5	10	2	2	4	14
Total Geral	108	108	216	36	36	72	288

os pesos sinápticos da rede em busca de que as respostas sejam estatisticamente mais próximas das desejadas [Haykin 2001].

Todo este processo acontece em ciclos sucessivos, chamados épocas, no qual todos os 216 padrões de comportamento que formam o conjunto de treino são apresentados para a rede juntamente com o valor esperado. Pela diferença entre o resultado gerado pela rede e o valor de saída esperado é calculado o erro quadrático médio.

Os 2 conjuntos de treinamento foram submetidos a 350 épocas de treino utilizando diferentes taxas de aprendizado, em busca da melhor taxa a ser aplicada para o experimento definitivo. O valor de taxa que melhor apresentou resultados, observando as diferentes convergências da rede para os valores configurados e considerando ambos conjuntos foi 0.05. Com isso, o treino definitivo teve este valor fixado.

A Figura 5.1 contém o gráfico com a evolução do treinamento do conjunto 1. Como pode ser visto, o gráfico representa a relação entre o erro máximo encontrado e a época em que ele ocorreu. No início do treinamento, o erro máximo que a rede pode ter é 2, devido o uso da função de ativação tangente hiperbólica que varia de -1 a 1.

Durante a evolução do treinamento observa-se o processo de convergência, significando que a rede está sendo capaz de aprender e com isso seus erros ficam menores ao término de cada época. O processo de convergência, no conjunto 1, iniciou-se em aproximadamente 220 épocas de treino. Na Figura 5.2, é mostrado o gráfico relativo ao treinamento utilizando o conjunto 2.

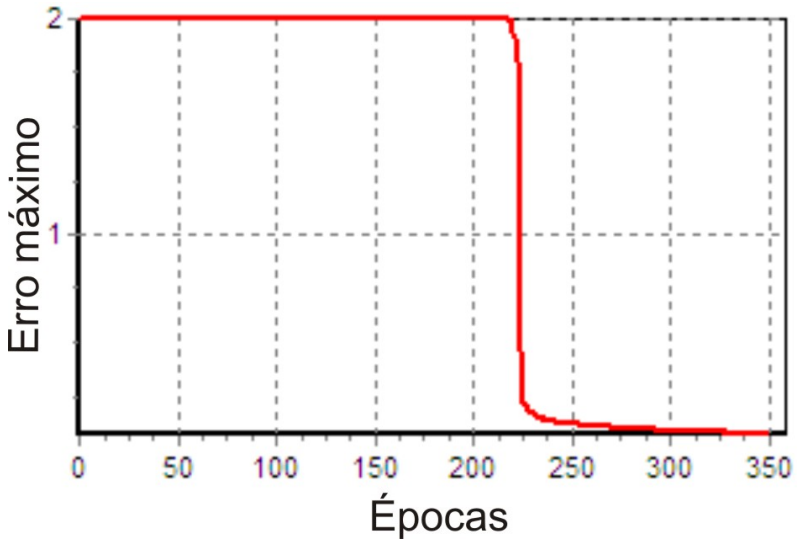

Figura 5.1: Evolução do Treinamento da Rede com o Conjunto 1.

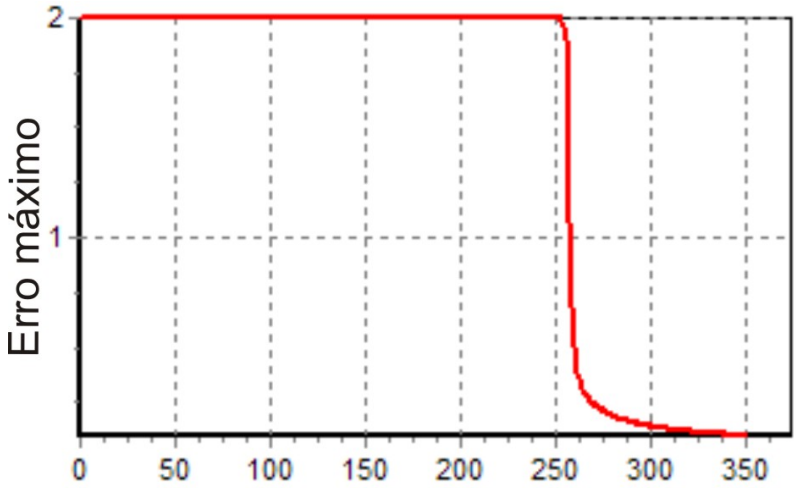

Figura 5.2: Evolução do Treinamento da Rede com o Conjunto 2.

Ao comparar os gráficos de treinamento de ambos conjuntos pode-se observar que o início do processo de convergência do conjunto 1 se deu antes do que o início do processo no conjunto 2. O processo de convergência, no conjunto 2, iniciou-se em aproximadamente 260 épocas de treino.

Testes realizados utilizando o próprio conjunto de treinamento mostraram que a rede obtém 100% de acerto e um erro quadrático médio muito pequeno. Ao

final do treinamento com 350 épocas utilizando o conjunto 1, o maior erro obtido foi 0,031, enquanto que com o conjunto 2 foi 0,097.

5.3 Validação do Treinamento

Este processo é realizado para verificar quanto válido está sendo o treino, durante o acontecimento do mesmo. A rede neural foi treinada com diferentes quantidades de épocas e, ao final de cada treinamento, o conjunto de testes era apresentado para a rede e calculado o erro quadrático médio. Com isto, foi possível observar a capacidade de aprendizado que a rede obteve durante a evolução das épocas. Para esta avaliação foram utilizadas 690 épocas de treinamento.

Esta validação não teve nenhuma influência sobre o processo de treinamento, pois os erros quadráticos obtidos com a apresentação do conjunto de testes não foi usado para atualização dos pesos sinápticos da rede. Serviu apenas como índice para estimativa de desempenho da rede.

Por meio da Figura 5.3, pode ser visto o gráfico que representa a variação dos erros quadráticos ao passar das épocas durante a validação do treinamento utilizando o conjunto 1. Neste gráfico, pode ser observado que o decrescimento dos erros quadráticos médios ocorre durante todo o processo de validação, além de ter apresentado um ponto de convergência a um erro mínimo no treino.

Na Tabela 5.3 podem ser observados todos os erros quadráticos médios em suas respectivas épocas utilizadas para esta validação e também o desvio padrão destes erros quadráticos. Estes dados formaram o gráfico ilustrado na Figura 5.3.

A Figura 5.4 e a Tabela 5.4 representam respectivamente o gráfico e os dados relativos à validação do treinamento utilizando o conjunto 2. Observando o comportamento ocorrido durante o treino dos dois conjuntos, é possível concluir que a formação do conjunto de treino influi diretamente na capacidade que a rede terá em classificar corretamente os padrões apresentados nos testes. Uma generalização efetiva só é possível caso a rede tenha conseguido extrair características dos padrões e com isso tenha aprendido.

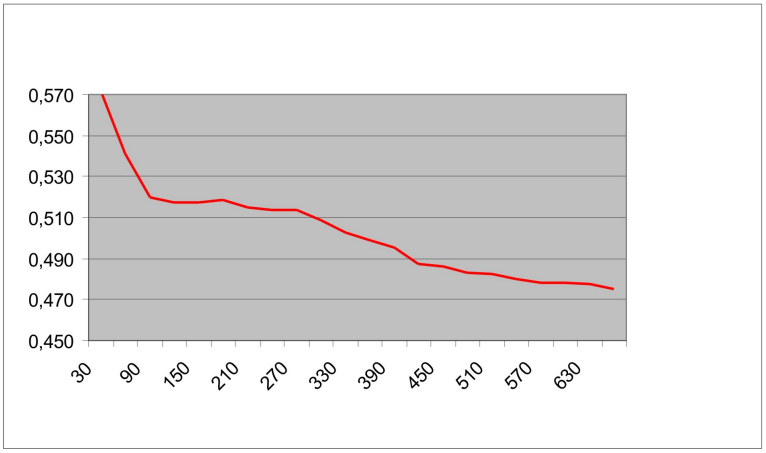

Figura 5.3: Gráfico dos Erros Quadráticos Médios por Época - Conjunto 1.

Tabela 5.3: Tabela dos Erros Quadráticos Médios por Época - Conjunto 1.

Conjunto 1		
Épocas	**Erro Quadrático Médio**	**Erro de Classificação**
30	0,571397114371293	28,57 %
60	0,541123383176532	27,06 %
90	0,519856633455523	25,99 %
120	0,517468463316065	25,87 %
150	0,517599254674574	25,88 %
180	0,518401879098553	25,92 %
210	0,514696511731063	25,73 %
240	0,513973227703981	25,70 %
270	0,513785852534358	25,69 %
300	0,508522317522205	25,43 %
330	0,502464704187843	25,12 %
360	0,499044201383946	24,95 %
390	0,495188793826557	24,76 %
420	0,487297930348881	24,36 %
450	0,485907726820149	24,30 %
480	0,482829497000035	24,14 %
510	0,482559702658554	24,13 %
540	0,480108119110326	24,01 %
570	0,478379104877847	23,92 %
600	0,478199498187487	23,91 %
630	0,477794111088129	23,89 %
660	0,475374268275306	23,77 %
690	0,474245676785102	23,71 %
Desvio Padrão:	**0,023967065306947**	

5.4 Resultados da Análise Neural

Como já visto na seção 5.2 (página 60), a rede neural é capaz de classificar corretamente 100% dos padrões vistos durante o treinamento com um erro quadrático

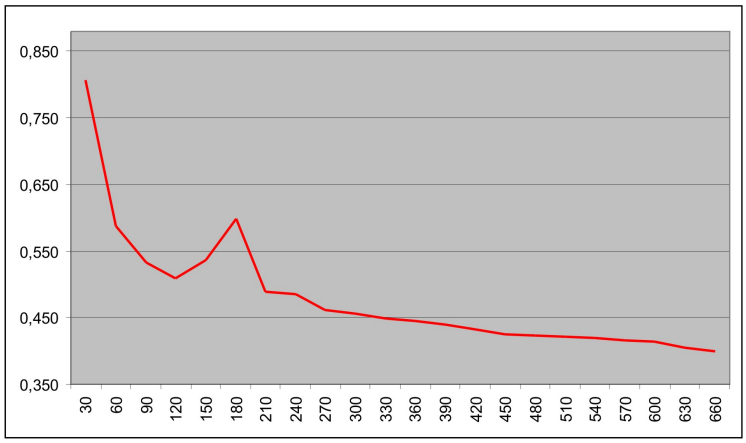

Figura 5.4: Gráfico dos Erros Quadráticos Médios por Época - Conjunto 2.

Tabela 5.4: Tabela dos Erros Quadráticos Médios por Época - Conjunto 2.

Conjunto 2		
Épocas	**Erro Quadrático Médio**	**Erro de Classificação**
30	0,806400723051632	40,32 %
60	0,587646959080701	29,38 %
90	0,533164162958709	26,66 %
120	0,508682728721874	25,43 %
150	0,536385892943530	26,82 %
180	0,598615323997714	29,93 %
210	0,488566605312641	24,43 %
240	0,485565342771927	24,28 %
270	0,461572367152890	23,08 %
300	0,456787890726962	22,84 %
330	0,448129664165042	22,41 %
360	0,444775954229990	22,24 %
390	0,438647976309683	21,93 %
420	0,431456098142428	21,57 %
450	0,424549959585621	21,23 %
480	0,423614758044353	21,18 %
510	0,421988181934272	21,10 %
540	0,419891743500198	20,99 %
570	0,416292842899283	20,81 %
600	0,413261428361712	20,66 %
630	0,405400175510276	20,27 %
660	0,399225572778616	19,96 %
690	0,389209725405938	19,46 %
Desvio Padrão:	**0,092434342559501**	

médio insignificante. Mas isto ocorre porque ela foi ensinada a partir destes padrões.

Para verificar a efetividade da rede, é necessário treiná-la com certos padrões e testá-la com outros não vistos na fase anterior. Assim, as saídas da rede neural

realmente podem ser consideradas classificações dadas com base em um aprendizado.

A rede neural teve melhor índice de classificação dos padrões de uso normal do que dos padrões intrusivos. Para demonstrar como foi a classificação, a Tabela 5.5 contém um dos padrões não intrusivos utilizados para testar a rede.

Tabela 5.5: Padrão Não Intrusivo Analisado.

Padrão não intrusivo:
#TCP
#21
#PASS
#Login incorrect
#QUIT
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0
0 1 1 0 1 0 1 1 1 0 0 0 1 0 0 0
0 0 1 1 0 0 0 1 1 0 1 1 0 1 0 0
0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Saída Desejada: -1
Saída Informada: -0,999698400497437

Nesta tabela está descrito o padrão como ele foi apresentado, a saída esperada e a saída informada pela rede neural. A RNA o classificou corretamente, pois era esperada como saída -1 e a rede informou -0,999698, que é um número mais próximo de -1 do que de 1. Neste caso, a rede classificou com 99,98% de certeza que o padrão era de uso normal.

A Tabela 5.6 mostra um padrão intrusivo apresentado à rede neural durante a fase de teste. Este padrão foi classificado corretamente, a resposta da rede foi 0,817563. A classificação realizada teve 90,88% de certeza que o padrão era intrusivo.

Em uma escala percentual fica mais visível a comparação do desempenho de classificação entre os dois conjuntos. É possível observar na Tabela 5.7 que a porcentagem de erros de classificação do conjunto 2, é menor que a porcentagem de erros

Tabela 5.6: Padrão Intrusivo Analisado.

Padrão intrusivo:
#TCP
#53
#|23|list
#|d840 cd80 e8d9 ffff ff|/bin/sh
#|ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff|
#|ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff|
#Bad
#|90 90 90 90 90 90 90 90 90|
#|ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff|
#|ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff|
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 1 1 0 0 1 1 1 1 0 0 1 1 0 1
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 0 1 1 0 1 0 1 0 1 1 1 0
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 1 1 0 1 1 0 0 0 0 1 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Saída Desejada: 1
Saída Informada: 0,817563498020172

do conjunto 1.

Com isso foi possível concluir que, com a validação do conjunto 2 (Figura 5.4) mesmo não tendo um comportamento semelhante a validação do conjunto 1 (Figura 5.3), com uma queda contínua nos erros quadráticos médios durante a passagem das épocas, as análises feitas foram mais satisfatórias. Pois, nas análises feitas com o conjunto 2 foram obtidos 19,46% de erros contra 23,71% das análises feitas com o conjunto 1. Assim, pode-se dizer que, a formação dos conjuntos teve influência no treinamento e consequentemente na aprendizagem e na capacidade de generalização para classificar corretamente os dados.

De forma geral, esta rede neural foi capaz de classificar corretamente em média 78,42% dos padrões apresentados durante os experimentos.

Tabela 5.7: Erros de Classificação da RNA.

Conjunto 1	Conjunto 2	Média
23,71%	19,46%	21,58%

5.5 Resultados da Análise com *Logcheck*

No intuito de realizar um comparativo com os resultados obtidos por meio da abordagem de [Machado 2005], as *logs* geradas a partir das sessões utilizadas como base para os experimentos serviram de subsídio para testes utilizando o analisador de *logs Logcheck*.

Para poder estabelecer o comparativo, as logs foram analisadas da mesma forma com que estavam agrupadas as sessões. Dois grupos de logs, contendo cada um, informações referentes as 72 sessões utilizadas nos experimentos anteriores. Ambos conjuntos são formados por metade dos dados gerados a partir de comportamentos normais e outra metade por comportamentos intrusivos.

O sistema de análise de *logs Logcheck* realiza sua classificação com base em palavras-chave que compõem os arquivos de configuração, organizando cada evento em diferentes categorias, da mais importante para a menos importante. Vão desde ataques em andamento a eventos gerados por uso normal. As categorias deste analisador são ataques, violações de segurança, eventos de segurança e eventos normais [Machado 2005].

As três primeiras armazenam os conjuntos que devem ser reportados ao administrador, de modo que, ataques e violações de segurança são classificados pelo *Logcheck* como anômalos, já os eventos de segurança são comportamentos desconhecidos, e assim não podem ser classificados como normais.

Levando em consideração as decisões de projeto de [Machado 2005], os dados classificados como ataques e violações de segurança são eventos intrusivos, ou verdadeiros positivos. Os dados classificados como eventos de segurança normalmente caracterizam erros de utilização e eventos usuais, e são chamados de falsos positivos. O

restante dos dados são considerados comportamentos de uso normal e são omitidos dos relatórios enviados ao administrador.

Nos experimentos comparativos do conjunto 1, o analisador classificou corretamente 69,45% dos comportamentos. A separação dada pelo *Logcheck* foi de 43,05% dos dados para o conjunto de eventos normais, 25% para o conjunto de eventos de segurança, 26,39% para o conjunto de violações de segurança e 5,56% para o conjunto de ataques.

Nas análises realizadas com o conjunto 2, foram classificados 72,22% dos dados corretamente. A classificação do *Logcheck* foi de 45,83% dos dados para o conjunto de eventos normais, 23,61% para o conjunto de eventos de segurança, 27,78% para o conjunto de violações de segurança e 2,78% para o conjunto de ataques. Por meio da tabela 5.8 pode ser visto a tabulação da classificação dos dados referentes a ambos os conjuntos testados.

Tabela 5.8: Classificação Realizada pelo *Logcheck*.

	Conjunto 1	***Conjunto 2***
Eventos Normais	43,05%	45,83%
Eventos de Segurança	25%	23,61%
Violações de Segurança	26,39%	27,78%
Ataques	5,56%	2,78%

O percentual de classificação dos eventos pelo analisador *Logcheck* esta ilustrado graficamente na Figura 5.5.

5.6 Comparativo entre os Métodos de Análise

Visto que os testes foram realizados com os mesmos tipos de variáveis, é possível estabelecer o comparativo desejado. Esta comparação é mostrada na tabela 5.9, por meio da qual pode-se verificar qual foi o percentual de erros de cada ferramenta de detecção. Assim, pode-se afirmar que a substituição da geração e análise dos eventos utilizado no projeto de [Machado 2005], pela rede neural avaliada teve resultados superiores,

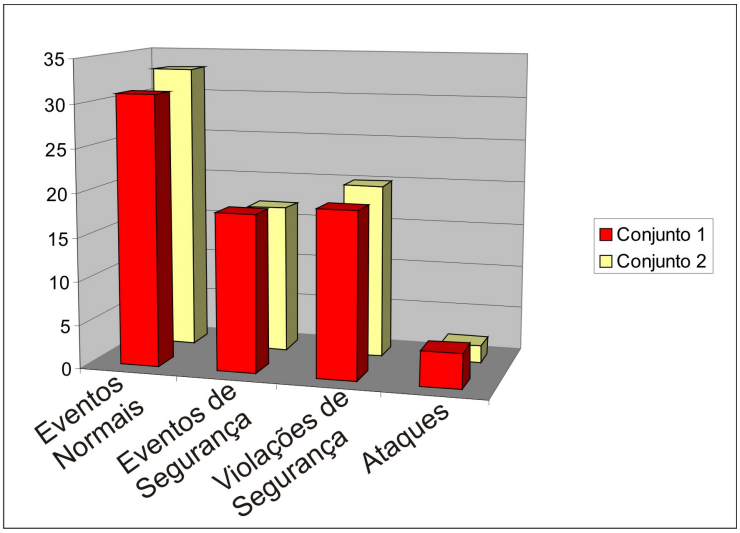

Figura 5.5: Classificação Realizada pelo *Logcheck*.

agregando a este protótipo uma ferramenta de inteligência artificial não proprietária.

Tabela 5.9: Percentual de Erros de Classificação nas Análises.

Dados Analisados			
RNA		***Logcheck***	
Conjunto 1	Conjunto 2	Conjunto 1	Conjunto 2
23,71%	19,46%	30,55%	27,78%

A classificação de eventos normais e intrusivos possibilita a ocorrência de alarmes falsos. Os eventos podem ser definidos corretamente, como verdadeiros positivos e verdadeiros negativos, ou incorretamente, como falsos positivos e falsos negativos. Uma comparação entre as classificações dadas em ambos os métodos para ambos os conjuntos é apresentada na Tabela 5.10.

Na Figura 5.6 é apresentado graficamente os dados mostrados na Tabela 5.10. É possível visualizar que a análise dada pela Rede Neural possibilitou um menor índice de alarmes falsos, consequentemente suas inferências quanto a característica dos dados analisados mostraram o real nível de periculosidades dos eventos.

Tabela 5.10: Percentual de Classificação dos Eventos.

	Dados Analisados			
	RNA		***Logcheck***	
	Conjunto 1	Conjunto 2	Conjunto 1	Conjunto 2
Verdadeiro Positivos	44,44%	45,83%	30,56%	30,56%
Verdadeiro Negativos	40,28%	43,06%	38,89%	41,67%
Falsos Positivos	9,72%	6,94%	26,39%	23,61%
Falsos Negativos	5,56%	4,17%	4,17%	4,17%

5.6.1 Classificação dada pelos Métodos de Análise

Com o método de detecção híbrido, utilizado neste trabalho, foi possível a classificação dos eventos em normais e anômalos. Para a RNA os eventos anômalos são os classificados como padrões intrusivos e os normais como padrão de uso normal, para o Logcheck os eventos anômalos são os eventos que foram organizados em suas categorias e os normais os que foram descartados. A classificação pode ter sido feita corretamente, gerando os verdadeiros positivos e negativos, ou incorretamente gerando os falsos positivos e negativos.

A classificação realizada em ambos os conjuntos de teste, foi dada pela Rede Neural nas sessões capturadas pelo *sniffer* e pelo analisador Logcheck nas *logs* geradas pelo *Syslog-ng*. O percentual de eventos normais e anômalos em cada conjunto dado por cada um dos métodos de análise é apresentado na Tabela 5.11.

Tabela 5.11: Classificação dos Eventos Normais e Anômalos.

	Dados Analisados			
	Conjunto 1		***Conjunto 2***	
	RNA	Logcheck	RNA	Logcheck
Eventos Normais	45,83%	43,06%	47,22%	45,83%
Eventos Anômalos	54,17%	56,94%	52,78%	54,17%

Na Figura 5.7 é ilustrado graficamente a relação dos percentuais mostrados na Tabela 5.11. Fica evidenciado que para ambos os conjuntos, a RNA classificou

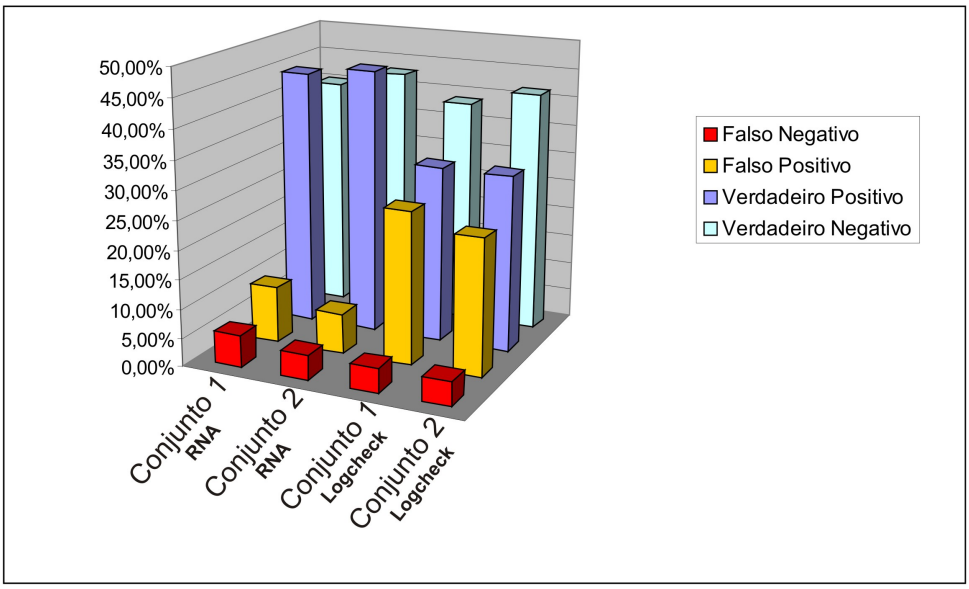

Figura 5.6: Percentual de Classificação dos Eventos.

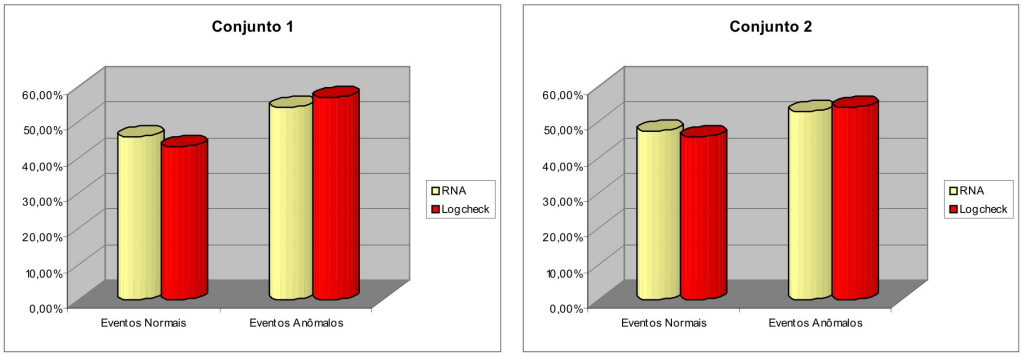

Figura 5.7: Percentuais de Eventos Normais e Anômalos.

uma maior parte dos dados como sendo eventos de uso normal, enquanto que o Logcheck acusou uma maior quantidade de dados anômalos.

Os eventos capturados foram agrupados por serviço e puderam ser avaliados separadamente. Desta forma, para cada um dos protocolos analisados, HTTP, FTP, DNS, POP3 e SMTP foram estabelecidos comparativos entre as classificações.

Os eventos classificados corretamente correspondem aos não intrusos e não anômalos na forma de verdadeiros negativos e aos intrusos e anômalos na forma de

verdadeiros positivos. Os eventos classificados incorretamente correspondem aos intrusos e não anômalos na forma de falsos negativos e aos não intrusos e anômalos na forma de falsos positivos.

5.7 Análise do Serviço HTTP

Na Figura 5.8 são mostrados os percentuais de classificação dos eventos do protocolo HTTP para ambos os conjuntos de testes. Analisando os percentuais fica evidenciado um melhor comportamento por parte da RNA, pois obteve um maior número de verdadeiros enquanto que o Logcheck um maior numero de falsos.

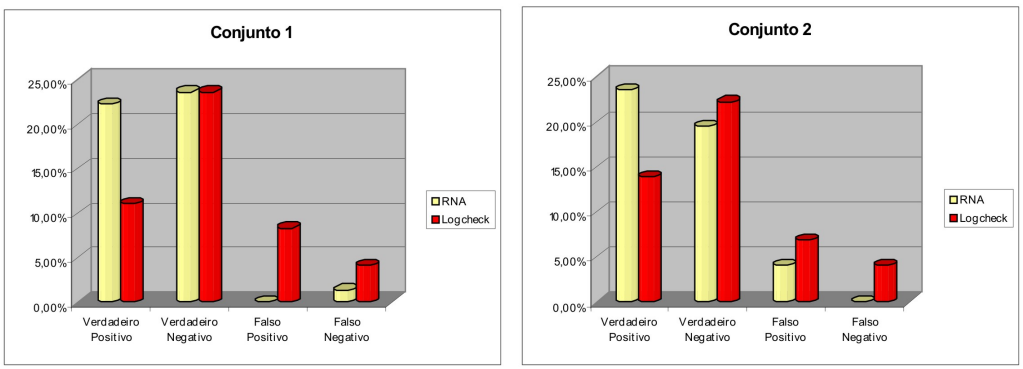

Figura 5.8: Percentual de Classificação dos Eventos do Protocolo HTTP.

Na Figura 5.8 conjunto 1, o percentual de verdadeiros positivos foi de 22,22% para a RNA e 11,11% para o Logcheck. O percentual de verdadeiros negativos teve o mesmo índice para ambos os métodos de análise, 23,61% dos eventos. O percentual de falsos positivos obtidos com a RNA foi nulo e de 8,33% com o Logcheck, e o de falsos negativos foi de 1,39% com a RNA e 4,17% com o Logcheck.

Os resultados obtidos com o conjunto 2, exibidos na Figura 5.8, foram semelhantes ao do conjunto 1, pois de maneira geral a RNA obteve menores níveis de respostas incorretas (RNA com 4,17% de falsos positivos e 0,00% de falsos negativos e o Logcheck 6,94% e 4,17% respectivamente) e um maior número de respostas corretas com os verdadeiros positivos (23,61% da RNA contra 13,89 do Logcheck). Em relação aos

verdadeiros negativos, o Logcheck teve um desempenho levemente superior, o percentual obtido foi de 22,22% contra 19,44% da RNA.

5.8 Análise do Serviço FTP

Na Figura 5.9 são exibidos os percentuais dos resultados obtidos com os eventos do serviço FTP. De forma geral, resultados foram equilibrados nos testes em ambos conjuntos.

Para o conjunto 1 da Figura 5.9, foi registrado o percentual de 11,11% de verdadeiros positivos para a RNA e 8,33% para o Logcheck. Mostrando assim, um maior número de detecções de eventos intrusivos neste serviço. O logcheck teve um melhor comportamento em relação a classificação dos eventos normais (verdadeiros negativos), pois classificou 11,11% enquanto a RNA 8,33%. Em relação as respostas incorretas, os falsos positivos tiveram 5,56% de classificação pela RNA e 8,33% pelo Logcheck. E os falsos negativos foram 2,78% das classificações da RNA e 0,00% das classificações do Logcheck.

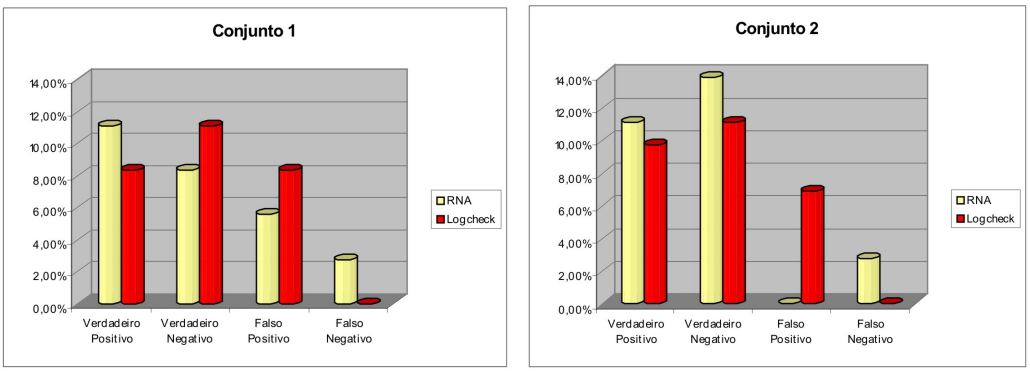

Figura 5.9: Percentual de Classificação dos Eventos do Protocolo FTP.

Nas análises com o conjunto 2, mostrado na Figura 5.9, os resultados obtidos com a RNA foram sensivelmente melhores. A Rede Neural possibilitou uma maior quantidade de respostas corretas, o percentual de verdadeiros positivos da RNA foi de 11,11% enquanto que do Logcheck foi de 9,72%. E o percentual de verdadeiros

negativos nas análises com a RNA foi de 13,89% e do Logcheck foi de 11,11%. O percentual de falsos positivos obtidos com a RNA foi de 0,00% e com o Logcheck 6,94%, e o percentual de falsos negativos foram de 2,78% obtidos com a RNA e 0,00% com o Logcheck.

5.9 Análise do Serviço DNS

Na Figura 5.10 é ilustrado os resultados obtidos nas análises com o serviço DNS. De forma geral, para o conjunto 1, os resultados ficaram equilibrados entre as ferramentas de análise. Para o conjunto 2, foi visível o melhor desempenho com a análise feita pela Rede Neural.

No conjunto 1 da Figura 5.10, os percentuais encontrados com as análises foram: 4,17% de verdadeiros positivos com a RNA e 5,56% com o Logcheck. 5,56% de verdadeiros negativos com a RNA e 1,39% com o Logcheck. 0,00% de falsos positivos com a RNA e 4,17% com o Logcheck e 1,39% de falsos negativos com a RNA e 0,00% com o Logcheck.

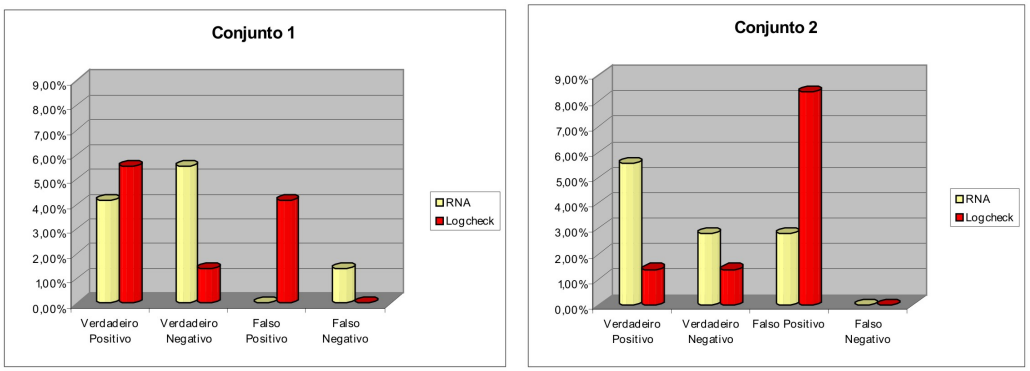

Figura 5.10: Percentual de Classificação dos Eventos do Protocolo DNS.

O melhor desempenho encontrado com a RNA no conjunto 2, exibido na Figura 5.10, fica evidenciado com um maior número de ocorrências de respostas corretas e um menor número de respostas incorretas em relação as análises com o Logcheck. Nos experimentos com este conjunto, a RNA classificou 5,56% dos dados como verda-

deiros positivos, enquanto que o Logcheck classificou 1,39% dos dados. Os verdadeiros negativos foram 2,78% classificados com a RNA e 1,39% com o Logcheck. A RNA classificou 2,78% dos eventos como falsos positivos e o Logcheck 8,33% dos eventos. Ambos os analisadores obtiveram o mesmo percentual de classificação em falsos negativos, 0,00% dos dados.

5.10 Análise do Serviço POP3

Na Figura 5.11 é exibido graficamente os percentuais dos resultados das análises com os eventos do Protocolo POP3. Especificamente para este serviço, o comportamento do Logcheck teve um melhor desempenho que o da RNA.

O percentual de verdadeiros positivos, exibidos na Figura 5.11 conjunto 1, foi de 2,78% com a RNA e 1,39% com o Logcheck. A Rede Neural não classificou nenhum evento como verdadeiro negativo enquanto o Logcheck classificou 1,39%. As respostas dadas incorretamente tiveram os mesmos percentuais em ambos analisadores. Os falsos positivos foram 2,78% das classificações da RNA e do Logcheck. Não houveram classificações do tipo falso negativo com nenhum dos analisadores.

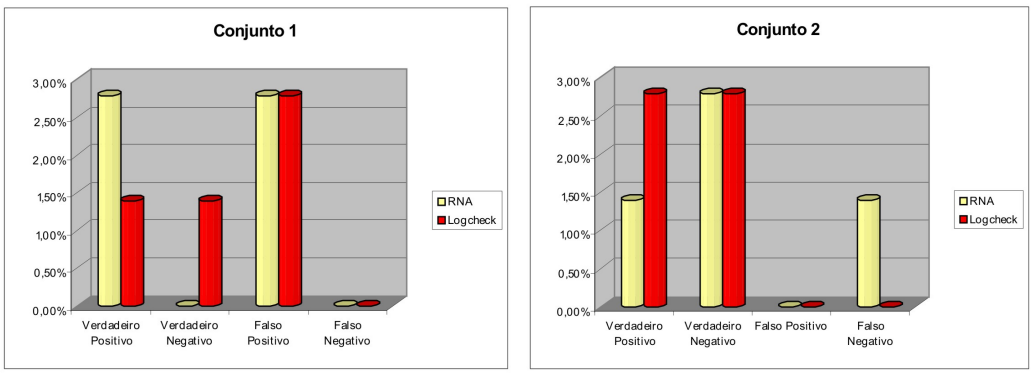

Figura 5.11: Percentual de Classificação dos Eventos do Protocolo POP3.

Com o conjunto 2, mostrado na Figura 5.11, foi obtido 1,39% de verdadeiros positivos com a RNA e 2,78% com o Logcheck. O percentual de verdadeiros negativos obtidos foi igual com ambos analisadores, 2,78% dos eventos. Não foi classi-

ficado nenhum evento como falso positivo com nenhum analisador. Os falsos negativos foram classificados em 1,39% dos casos com a RNA e em nenhum caso com o Logcheck.

5.11 Análise do Serviço SMTP

Na Figura 5.12 é exibido graficamente os percentuais dos resultados das análises com os eventos do Protocolo SMTP. De forma geral, com o conjunto 1 foram obtidas melhores classificações com a rede neural. Com o conjunto 2 foram obtidos índices semelhantes com ambos analisadores.

Nas análises com o conjunto 1, exibido na Figura 5.12, foram obtidos os mesmos percentuais de verdadeiros positivos com ambos analisadores, 4,17% dos eventos. O percentual de verdadeiros negativos obtidos foi de 2,78% com a rede neural e de 1,39% com o Logcheck. Foram obtidos 1,39% de falsos positivos com a RNA e 2,78% com o Logcheck. O percentual de falsos negativos obtidos com ambos os analisadores foi nulo.

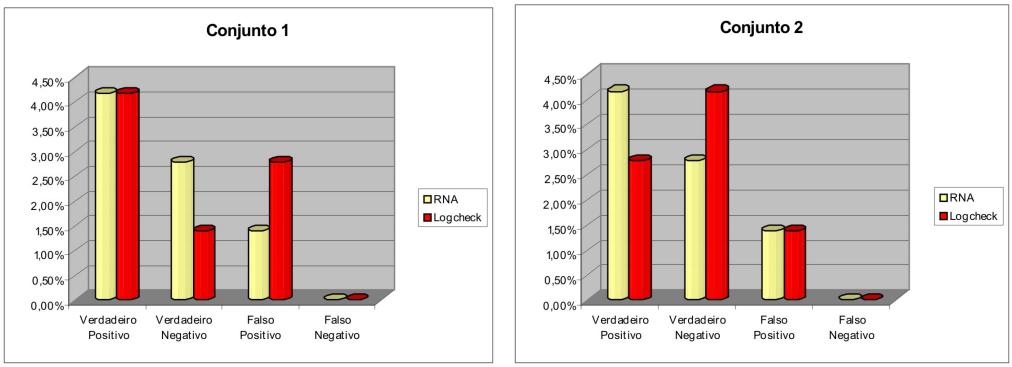

Figura 5.12: Percentual de Classificação dos Eventos do Protocolo SMTP.

Com o conjunto 2, mostrado na Figura 5.12, foram obtidos 4,17% de verdadeiros positivos com a RNA e 2,78% com o Logcheck. Os verdadeiros negativos foram 2,78% das classificações feitas pela RNA e 4,17% das classificações realizadas com o Logcheck. As respostas incorretas dadas por ambos analisadores foram iguais. Foram classificados 1,39% dos eventos como falsos positivos e as classificações do tipo

falsos negativos foram nulas.

5.12 Tabela Comparativa dos Percentuais

As Tabelas 5.12 e 5.13 mostram um resumo dos dados obtidos nas análises realizadas nas seções 5.7 a 5.11.

Tabela 5.12: Comparativos entre os protocolos - Conjunto 1

	Conjunto 1									
	HTTP		***FTP***		***DNS***		***POP3***		***SMTP***	
	RNA	Logcheck	RNA	Logcheck	RNA	Logcheck	RNA	Logcheck	RNA	Logcheck
Verdadeiro Positivo	22,22%	11,11%	11,11%	8,33%	4,17%	5,56%	2,78%	1,39%	4,17%	4,17%
Verdadeiro Negativo	23,61%	23,61%	8,33%	11,11%	5,56%	1,39%	0,00%	1,39%	2,78%	1,39%
Falso Positivo	0,00%	8,33%	5,56%	8,33%	0,00%	4,17%	2,78%	2,78%	1,39%	2,78%
Falso Negativo	1,39%	4,17%	2,78%	0,00%	1,39%	0,00%	0,00%	0,00%	0,00%	0,00%

Tabela 5.13: Comparativos entre os protocolos - Conjunto 2

	Conjunto 2									
	HTTP		***FTP***		***DNS***		***POP3***		***SMTP***	
	RNA	Logcheck	RNA	Logcheck	RNA	Logcheck	RNA	Logcheck	RNA	Logcheck
Verdadeiro Positivo	23,61%	13,89%	11,11%	9,72%	5,56%	1,39%	1,39%	2,78%	4,17%	2,78%
Verdadeiro Negativo	19,44%	22,22%	13,89%	11,11%	2,78%	1,39%	2,78%	2,78%	2,78%	4,17%
Falso Positivo	4,17%	6,94%	0,00%	6,94%	2,78%	8,33%	0,00%	0,00%	1,39%	1,39%
Falso Negativo	0,00%	4,17%	2,78%	0,00%	0,00%	0,00%	1,39%	0,00%	0,00%	0,00%

5.13 Análise Estatística dos Métodos

Conforme análise estatística aplicada aos resultados obtidos com ambos conjuntos de teste, por meio do método estatístico descrito na seção 4.2, página 39, obteve-se o *p-valor* igual a 0.0464 para o conjunto 1 e igual a 0.0194 para o conjunto 2.

A Figura 5.13 mostra a relação entre as respostas corretas e incorretas dadas pelo métodos de análise RNA e Logcheck no conjunto 1.

A Figura 5.14 mostra a relação entre as respostas corretas e incorretas dadas por ambos os métodos de análise no conjunto 2. Os p-valores obtidos se encontram no intervalo de $0.05 \geq$ *P-valor* > 0.01.

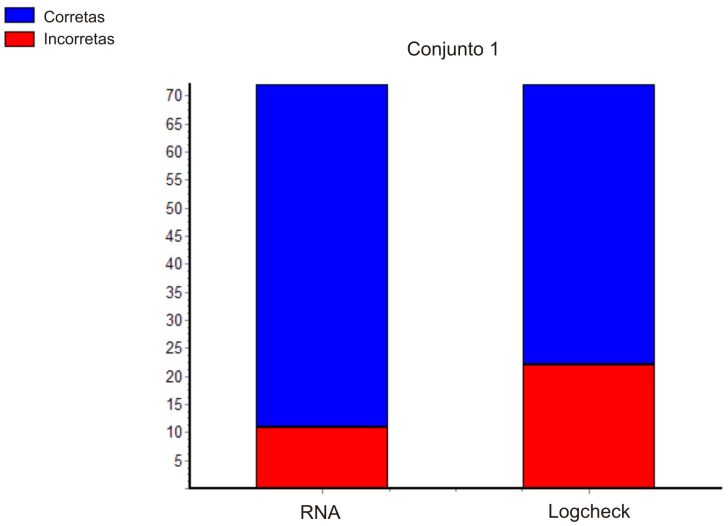

Figura 5.13: Comparação Estatística - Conjunto 1.

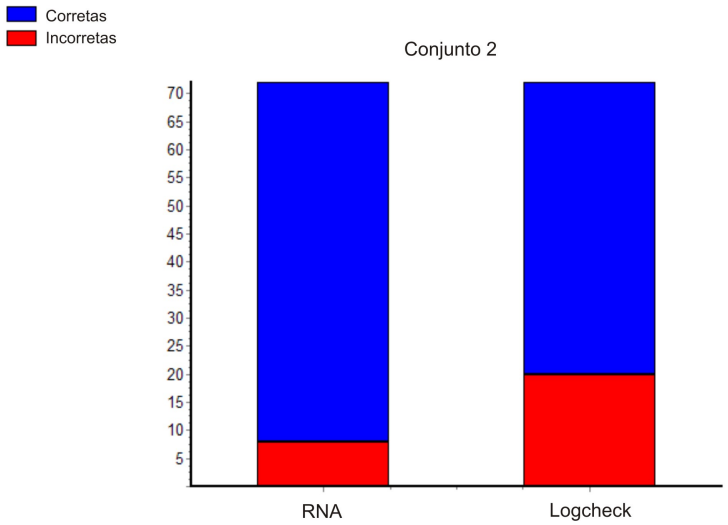

Figura 5.14: Comparação Estatística - Conjunto 2.

Com estes valores fica comprovado com um intervalo de confiança de 95%, pelo teste exato de fischer, que as diferenças entre os métodos de análise para detecção de intrusão estudados são consideradas estatísticamente significantes.

5.14 Aplicação do Modelo de SIA

A realização dos experimentos neste modelo de SIA, cujo método de detecção é uma RNA, permitiu uma otimização no modelo computacional. Foi possível avaliar toda a solução em um ambiente real. Este protótipo foi testado em um ambiente corporativo, visando a obtenção de dados reais para a realização dos experimentos, pois esta exposição do modelo a situações reais permitiu avaliar seu comportamento diante de situações inusitadas.

O sistema imunológico artificial pôde ser representado por meio dos agentes móveis, região, agências de origem e destino e seus arquivos de configuração. Na Figura 5.15, pode ser vista a região SIA com todas suas agências controladas: `Órgão`, `Timo` e `Medula`.

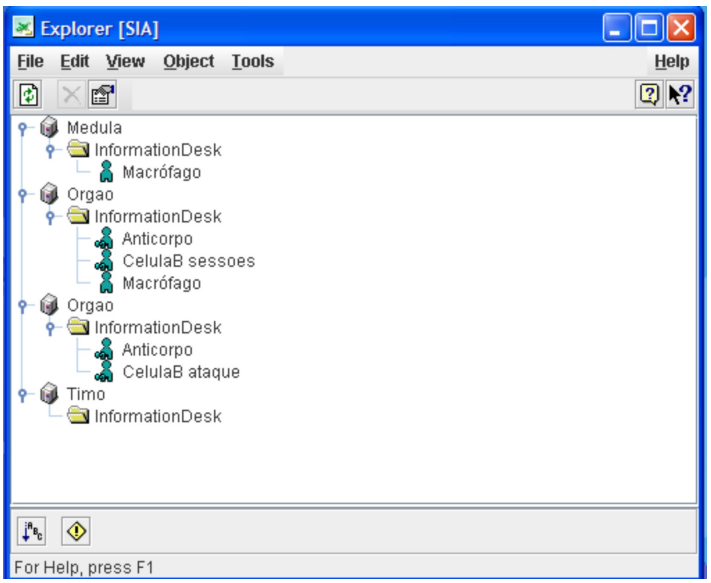

Figura 5.15: Região SIA com as Agências Controladas.

O servidor descrito na seção 4.1, teve seus serviços disponíveis analisados com a observação do tráfego da rede durante 3 meses. Assim foram catalogados dados que representavam diversas situações de utilização real da rede, e estabelecer os comparativos necessários à definição dos elementos `self` e `nonself`. Desta maneira, as técnicas de ataques à rede têm seus paralelos aos `patógenos` que agridem o sistema

imunológico humano.

A captura das sessões pelo *sniffer* é um processo análogo à atividade dos macrófagos que coletam resíduos, fragmentando antígenos para a apresentação aos linfócitos. A funcionalidade do macrófago é complementada com um agente móvel que transporta o fluxo capturado para entregar às células T-Helper (rede neural).

Para a realização deste processo, na agência de origem (estação sensor), um agente de monitoração (célula B) foi instanciado para monitorar a existência de dados capturados. Caso encontre um novo conjunto de sessões, este agente instancia agentes móveis de transporte (macrófago) para levar as sessões até a estação de gerência (agência Medula).

As células T-Helper fazem a distinção entre antígenos self e antígenos nonself, classificando-os de acordo com o nível de periculosidade encontrado. Antes de analisar as sessões com a rede neural, o administrador pode alterar o índice de ajuste. Aumentando o valor deste índice, diminui a quantidade de eventos reportados como sendo intrusivos.

Na estação de gerência, outro agente de monitoração (célula B) é instanciado para verificar a existência de comportamentos intrusivos. Quando encontrados os patógenos (sessões classificadas como intrusivas), este agente instancia agentes de distribuição (plasmócitos) que percorrem o itinerário até a agência de destino (agência Timo) e armazenam estes dados. Caso haja alguma instabilidade na rede, os agentes de persistência realizam o armazenamento local das sessões intrusivas até que seja reestabelecido o funcionamento.

São instanciados também os agentes móveis de reatividade (anticorpos), que em função dos dados classificados como ataque se deslocam até a máquina atacada e param o serviço explorado. Alternativamente, os agentes reativos foram configurados para inserir regras no *Firewall*, que impeçam o acesso ao servidor por parte dos endereços *IP* dos atacantes.

5.15 Considerações Finais

Neste Capítulo, foram mostradas as formas de realização dos testes, além de discutidos os resultados obtidos por meio destes experimentos. Foi possível obter um comparativo na análise neural a partir de treinamentos usando conjuntos com diferentes formações.

Os experimentos possibilitaram a análise de todo o modelo computacional. Verificando, desta forma, que a inclusão de uma rede neural que permita detectar variações de ataques além dos ataques já conhecidos, agregou características desejáveis em sistema de detecções de intrusos. Os testes mostraram que este modelo de sistema imunológico artificial pode ser aplicável em um ambiente real.

Outra característica relevante é que a estação de análise neural possui um índice de ajuste (subseção 4.4.3.1, página 52), que permitiu diferentes configurações na interpretação da saída da rede neural. Assim, foi possível a redução na quantidade de falsos positivos e negativos.

Experimentos comparativos permitiram verificar o desempenho dos diferentes métodos de análise utilizados em [Lima 2005] e [Machado 2005]. Estabelecidas as devidas métricas, os dados puderam ser analisados pela rede neural e pelo *Logcheck*. Assim foi possível comparar os percentuais de erros de classificação de cada método, além de mensurar o que foi classificado como verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo.

Uma análise mais minuciosa foi realizada para possibilitar um comparativo entre as respostas corretas e as incorretas dadas pelo analisadores para cada serviço utilizado. Estes experimentos visaram encontrar as melhores análises feitas em cada protocolo. De forma geral, a rede neural teve desempenho superior ao Logcheck, classificando um maior número de verdadeiros e um menor número de falsos. Em poucos casos os resultados foram mais favoráveis ao Logcheck.

Para se ter uma prova estatística das diferenças entre as análises feitas com os dois métodos foi aplicado o teste exato de fischer. Assim fica provado com um intervalo de confiança de 95% que as diferenças são consideradas estatisticamente signi-

ficantes.

No Capítulo 6, são apresentadas as conclusões obtidas a partir destas experiências. Tem-se também a descrição de propostas para trabalhos futuros.

Capítulo 6

Conclusão

A grande capacidade do sistema imunológico na proteção do corpo humano oferece uma fonte de inspiração para o desenvolvimento de soluções para a detecção de intrusos. Aliado a esta característica é possível utilizar o poder de reconhecimento de padrões das redes neurais artificiais com sua capacidade de aprendizado e generalização.

Por meio destas características foi possível a construção de um modelo bioinspirado para detecção de intrusão. Assim, tem-se como resultado da abordagem um sistema baseado no método de detecção por abuso e anomalia, que analisa as sessões capturadas da rede. Possuindo, portanto, uma arquitetura baseada em rede e distribuída, executada em períodos de tempo pré-definido e com reatividades passivas e ativas.

A análise do modelo proporcionou uma avaliação minuciosa, tendo como subsídios os dados analisados pela rede neural artificial e os padrões de armazenamento e de reação realizados pelos agentes móveis. Assim, foi possível mensurar os resultados obtidos.

Com esta abordagem, além de não se fazerem necessários grandes recursos computacionais para o armazenamamento de bases de regras, tem-se a possibilidade de replicação do arquivo de pesos sinápticos, que é o conhecimento armazenado, para outras redes semelhantes por meio dos agentes móveis.

As análises mostraram a possibilidade de detecções de intrusos com níveis satisfatórios, permitindo classificar corretamente 100% dos dados que fizeram parte

do conjunto de treinamento. Em relação às sessões inéditas ou que eram variações das sessões contidas no treino, a rede neural teve em média 78,42% de classificações corretas. A solução ainda permite o ajuste na escala de classificação, assim o administrador pode modificar o índice do que a rede classifica como normal ou intrusivo. Desta forma, é reduzido o número de falsos positivos e falsos negativos.

Os experimentos foram complementados por meio de análises de *logs* para que fosse possível estabelecer comparativos entre os resultados, e com isso mostrar as vantagens que uma rede neural pode agregar ao modelo de imunologia artificial desenvolvido em [Machado 2005]. As *logs* geradas no servidor, durante os testes com comportamentos normais e intrusivos que proporcionaram a captura do tráfego da rede, possibilitaram o uso da ferramenta *Logcheck* para realizar uma classificação paralela.

Com a análise das *logs* conseguiu-se o percentual médio de 70,84% de classificações corretas. A partir da comparação dos percentuais obtidos foi possível mostrar que o método de análise neural das sessões possibilitou melhor desempenho. Estes experimentos demostraram que o comportamento da solução foi satisfatório com a integração dos modelos existentes. Desta forma foi possibilitado que uma rede neural artificial fosse utilizada como técnica de detecção dentro do contexto de um modelo de SIA.

Trabalhos Futuros

Apesar das funcionalidades agregadas ao modelo de detecção de intrusão, algumas deficiências necessitam de atenção para futuras propostas de abordagens. Estes estudos podem viabilizar a otimização da solução, incorporando novos métodos e técnicas de detecção de intrusão. Assim, citam-se algumas propostas para trabalhos futuros:

- Viabilizar a captura de tráfego criptografado para as posteriores análises neurais.
- Implementação da análise neural com base em registros de *logs*, pois assim além de possibilitar a detecção de intrusos que não se utilizem da rede para as investidas,

permitem que dados que trafegam criptografados na rede sejam analisados, já que são observados após serem decriptados.

- Incrementar métricas baseadas em sistemas especialistas para realização de uma pré-seleção nos eventos, diminuindo assim o esforço da análise.
- Incrementar os padrões de reatividade, diversificando-os e tornando o modelo mais semelhante ao modelo biológico.
- Possibilitar que as contra medidas sejam mais adequadas para cada tipo de intrusão, fazendo com que as medidas extremas só sejam tomadas em casos que realmente as necessite.

A pesquisa para o atendimento destas funcionalidades visa buscar soluções mais robustas de reconhecimento e de reações inteligentes contra intrusos em redes de computadores.

Referências Bibliográficas

[Al-Subaie e Zulkernine 2006]AL-SUBAIE, M.; ZULKERNINE, M. Efficacy of hidden markov models over neural networks in anomaly intrusion detection. In: *Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International*. [S.l.: s.n.], 2006. v. 1, p. 325–332.

[BackTrack 2007]BACKTRACK. *BackTrack 3 Beta*. Dezembro 2007. Acessado em 10/01/2008. Disponível em: <www.remote-exploit.org>.

[Barbosa e Moraes 2000]BARBOSA, A. S.; MORAES, L. F. M. de. *Sistemas de Detecção de Intrusão*. Dezembro 2000. Seminários Ravel - CPS760: Laboratório de Redes de Alta Velocidade, UFRJ.

[Boukerche et al. 2007]BOUKERCHE, A. et al. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications*, v. 30, p. 2649–2660, march 2007. Disponível em: <http://www.sciencedirect.com/science/article/B6TYP-4NBH21F-1/1/81412ca7b05c6b7c924b9ab96304e01a>.

[Campello e Weber 2001]CAMPELLO, R. S.; WEBER, R. F. Sistemas de detecção de intrusão. In: *Anais do Simpósio Brasileiro de Redes de Computadores*. [S.l.: s.n.], 2001. Instituto de Informática UFRGS.

[Cansian 1997]CANSIAN, A. M. *Desenvolvimento de um Sistema Adaptativo de Detecção de Intrusos em Redes de Computadores*. Tese (Doutorado) — Universidade de São Paulo - USP, 1997.

[Castro 2001]CASTRO, L. N. de. *Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais*. Tese (Doutorado) — Departamento de Engenharia da Computação e Automação Industrial - Universidade Estadual de Campinas, Maio 2001.

[Castro e Zuben 1999]CASTRO, L. N. de; ZUBEN, F. J. V. *Artificial Immune Systems: Part I - Basic Theory and Applications*. [S.l.], 1999.

[CERT.BR 2007]CERT.BR. *Documentos do CERT.br*. Novembro 2007. Acessado em 23/11/2007. Disponível em: <http://www.cert.br/docs>.

[CERT/CC 2007]CERT/CC. *CERT/CC Statistics 1988-2006*. Computer Emergency Response Team (Coordenation Center), Fevereiro 2007. Acessado em 17/02/2007. Disponível em: <http://www.cert.org/stats/cert_stats.html/>.

[Dasgupta et al. 1999]DASGUPTA, D. et al. *Artificial Immune Systems and Their Applications*. [S.l.]: Springer-Verlag Berlin Heidelberg, 1999.

[Degaspari, Lima e Sobral 2008]DEGASPARI, J. A.; LIMA, I. V. M. de; SOBRAL, J. B. M. Intrusion detection through artificial neural networks. In: *11th IEEE/IFIP Network Operations and Management Symposium - NOMS 2008*. [S.l.: s.n.], 2008.

[Deng et al. 2003]DENG, Y. et al. An approach for modeling and analysis of security system architectures. In: *IEEE Transactions on Knowledge and Data Engineering*. [S.l.: s.n.], 2003. p. 1099 – 1119.

[Deraison 2004]DERAISON, R. *NESSUS Project - Project founder and current leader*. January 2004. Acessado em 10/11/2007. Disponível em: <www.nessus.org>.

[Elson 2003]ELSON, J. *tcpflow – A TCP Flow Recorder*. [S.l.], Agosto 2003. Acessado em 26/11/2007. Disponível em: <http://www.circlemud.org/jelson/software/tcpflow/>.

[Filho 1999]FILHO, U. D. *Introdução à Bioestatística - Para simples mortais*. [S.l.: s.n.], 1999.

[Forrest et al. 1996]FORREST, S. et al. A sense of self for unix processes. In: *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*. [s.n.], 1996. p. 120–128. Acessado em 30/01/2008. Disponível em: <citeseer.ist.psu.edu/forrest96sense.html>.

[Haykin 2001]HAYKIN, S. *Redes Neurais - Princípios e Práticas*. Segunda edição. [S.l.]: Bookman, 2001.

[Heady et al. 1990]HEADY, R. et al. *The Architecture of a Network Level Intrusion Detection System*. [S.l.], 1990.

[Juca 2001]JUCA, K. R. L. *Uma Abordagem de Detecção de Intrusão Baseada no Sistema Imunológico Humano*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, Dezembro 2001.

[Kahn et al. 1998]KAHN, C. et al. A common intrusion detection framework. *Journal of Computer Security*, July 1998. Disponível em: <http://gost.isi.edu/cidf/papers/cidf-jcs.ps>.

[Laureano, Maziero e Jamhour 2007]LAUREANO, M.; MAZIERO, C.; JAMHOUR, E. Protecting host-based intrusion detectors through virtual machines. *Elsevier Computer Networks Journal*, v. 51, p. 1275–1283, April 2007. Disponível em: <http://www.sciencedirect.com/science/article/B6VRG-4M4CTMM-1/1/cbf622e17c767e6d6bc967ae7ddc78c2>.

[Lent 2001]LENT, R. *Cem Bilhões de Neurônios - Conceitos Fundamentais da Neurociência*. [S.l.]: Atheneu, 2001.

[Lima 2005]LIMA, I. V. M. de. *Uma Abordagem Simplificada de Detecção de Intrusão Baseada em Redes Neurais Artificiais*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, Fevereiro 2005.

[Lipson 2002]LIPSON, H. F. Tracking and tracing cyber attacks: Technical challenges and global policy issues. 2002. Acessado em 27/01/2008. Disponível em: <http://www.cert.org/archive/pdf/02sr009.pdf>.

[Machado 2003]MACHADO, A. B. M. *Neuroanatomia Funcional*. Segunda edição. [S.l.]: Atheneu, 2003.

[Machado et al. 2005]MACHADO, R. et al. A hybrid artificial immune and mobile agent intrusion detection based model for computer network operations. In: *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. [S.l.: s.n.], 2005. p. 191a–191a.

[Machado 2005]MACHADO, R. B. *Uma Abordagem de Detecção e Intrusão Baseada em Sistemas Imunológicos Artificiais e Agentes Móveis*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, Fevereiro 2005.

[Michailidis, Katsikas e Georgopoulos 2008]MICHAILIDIS, E.; KATSIKAS, S.; GEORGOPOULOS, E. Intrusion detection using evolutionary neural networks. In: *Informatics, 2008. PCI '08. Panhellenic Conference on*. [S.l.: s.n.], 2008. p. 8–12.

[Militelli 2006]MILITELLI, L. C. *Proposta de um Agente de Aplicação para Detecção, Prevenção e Contenção de Ataques em Ambientes Computacionais*. Dissertação (Mestrado) — Universidade de São Paulo, 2006.

[Mo, Ma e Xu 2008]MO, Y.; MA, Y.; XU, L. Design and implementation of intrusion detection based on mobile agents. In: *IT in Medicine and Education, 2008. ITME 2008. IEEE International Symposium on*. [S.l.: s.n.], 2008. p. 278–281.

[Másson e Wang 1990]MáSSON, E.; WANG, Y.-J. Introduction to computation and learning in artificial neural networks. *European Journal of Operational Research*, v. 47, p. 1–28, 1990. Acessado em 28/01/2008. Disponível em: <http://www.sciencedirect.com/science/article/B6VCT-48NBFK6-1JW/2/437c3588d7b7ae2181e489d44119fed7>.

[Mukkamala e Sung 2003]MUKKAMALA, S.; SUNG, A. Artificial intelligent techniques for intrusion detection. In: *Systems, Man and Cybernetics, 2003. IEEE International Conference on*. [S.l.: s.n.], 2003. v. 2, p. 1266–1271vol.2.

[Northcutt et al. 2002]NORTHCUTT, S. et al. *Desvendando Segurança em Redes*. [S.l.]: Campus, 2002.

[Osório e Vieira 1999]OSóRIO, F. S.; VIEIRA, R. Sistemas híbridos inteligentes. In: *Encontro Nacional de Inteligência Artificial - ENIA 99*. [S.l.: s.n.], 1999.

[Paula 2004]PAULA, F. S. de. *Uma Arquitetura de Segurança Computacional Inspirada no Sistema Imunológico*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, Junho 2004.

[Raguenet e Maziero 2006]RAGUENET, I. F.; MAZIERO, C. Um modelo de composição de detectores de intrusão heterogêneos baseado em conjuntos difusos. In: *VI Simpósio Brasileiro de Segurança da Informação e Sistemas - SBSeg*. [S.l.: s.n.], 2006.

[Roitt, Brostoff e Male 1999]ROITT, I.; BROSTOFF, J.; MALE, D. *Imunologia*. [S.l.: s.n.], 1999.

[SAINT 2004]SAINT. *Security Administrator's Integrated Network Tool*. January 2004. Acessado em 17/10/2007. Disponível em: <www.saintcorporation.com>.

[Schindler et al. 2005]SCHINDLER, L. et al. *Understanding The Immune System*. January 2005. Public Health Service. Acessado em 25/4/2008. Disponível em: <www.nci.nih.gov/cancertopics/understandingcancer/immunesystem>.

[Schulter 2006]SCHULTER, A. *Integração de Sistemas de Detecção de Intrusão para Segurança de Grades Computacionais*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, Fevereiro 2006.

[Somayaji, Hofmeyr e Forrest 1997]SOMAYAJI, A.; HOFMEYR, S.; FORREST, S. Principles of a computer immune system. In: *Second New Security Paradigms Workshop*. ACM, 1997. p. 75–82. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.253>.

[Souza, Silva e Cansian 2002]SOUZA, M. de; SILVA, A. R. A. da; CANSIAN, A. M. Proposta de um modelo padronizado para representação de assinaturas de intrusão. In: *IV Simpósio de Segurança em Informática - SSI*. [S.l.: s.n.], 2002.

[Stallings 2003]STALLINGS, W. *Cryptography and Network Security: Principles and Practice*. [S.l.: s.n.], 2003.

[Staniford-Chen et al. 1998]STANIFORD-CHEN, S. et al. The common intrusion detection framework (cidf). In: *Information Survivability Workshop*. [s.n.], 1998. Acessado em 15/03/2008. Disponível em: <http://gost.isi.edu/cidf/>.

[Tanenbaum 2003]TANENBAUM, A. S. *Redes de Computadores*. [S.l.: s.n.], 2003. 4th Ed.

[Wasserman 1989]WASSERMAN, P. D. *Neural computing: theory and practice*. [S.l.]: Van Nostrand Reinhold Co., 1989.

[Yu et al. 2006]YU, Y. et al. Anomaly intrusion detection approach using hybrid mlp/cnn neural network. In: *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on*. [S.l.: s.n.], 2006. v. 2, p. 1095–1102.