UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

RESPOSTAS AUTOMÁTICAS PARA MELHORIA DA SEGURANÇA EM SISTEMAS DE DETECÇÃO DE INTRUSOS

GLENDA DE LOURDES FERREIRA DOS SANTOS

São Luis
2003

RESPOSTAS AUTOMÁTICAS PARA MELHORIA DA SEGURANÇA EM SISTEMAS DE DETECÇÃO DE INTRUSOS

Dissertação de Mestrado submetida à Coordenação do Curso de Pós-Graduação em Engenharia de Eletricidade da UFMA como parte dos requisitos para obtenção do título de mestre em Ciência da Computação.

Por

GLENDA DE LOURDES FERREIRA DOS SANTOS

Novembro, 2003

Santos, Glenda de Lourdes Ferreira.

Agentes inteligentes para detecção de intrusos em redes de computadores/ Glenda de Lourdes Ferreira dos Santos. ___ São Luís, UFMA 2003.

177f.:il.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal, 2003.

1. Detecção de infecções - Computador. 2. Segurança de Rede.

I. Título

CDD 001.642

RESPOSTAS AUTOMÁTICAS PARA MELHORIA DA SEGURANÇA EM SISTEMAS DE DETECÇÃO DE INTRUSOS

MESTRADO

Área de Concentração: CIÊNCIA DA COMPUTAÇÃO

GLENDA DE LOURDES FERREIRA DOS SANTOS

Orientadores: Dr. Zair Abdelouahab

Dr. Edson Nascimento

Curso de Pós-Graduação

Em Engenharia de Eletricidade da

Universidade Federal do Maranhão

A Deus.

Aos meus pais, Antonio e Graça.

A minha filha Cecília Giovanna.

A Pietro, meu marido.

Aos Irmãos, Germana e Aarão.

Aos meus Sobrinhos Leticya, Paulo e Larissa.

AGRADECIMENTOS

A Deus, que me concedeu a vida e é digno de toda honra, glória e louvor. Sua intercessão tornou este dia uma realidade.

À Coordenação de Pós-Graduação de Engenharia Elétrica por ter-me concedido a oportunidade de ingressar no Programa de Mestrado.

Aos meus Orientadores Prof. Dr. Zair Abdelouahab e Prof. Dr. Edson Nascimento, pela imensa paciência, compreensão e dedicação dispensada à realização desse trabalho. Apesar de minhas “caminhadas em círculos”, me guiaram para o encontro da direção certa.

Aos colegas e amigos da Pós-Graduação, cujo convívio foi um grande aprendizado. Em especial a Fábio Frazão Mendes e a Nilson Santos Costa, por ter colaborado significantemente para o desenvolvimento deste trabalho. A todos muito obrigada.

Aos meus pais, pela formação do meu caráter e pela educação, cuja foi imprescindível para a apresentação deste trabalho. A todos os familiares que compartilharam comigo os momentos, bons e ruins, e ajudaram-me a vencer todas as minhas dificuldades.

A todos que, direta ou indiretamente, contribuíram para a elaboração desta dissertação.

RESUMO

O desenvolvimento de mecanismos para reações rápidas contra intrusos tem sido um dos mais importantes requisitos na defesa crítica de redes de computador, visto que estes agem rapidamente exigindo reações sem intervenção humana. Tais mecanismos devem estar habitas a, automaticamente, responder um ataque e lidar com o vários aspectos do problema de seguança de computadores, e com isso reduzir a carga de trabalho do administrador do sistema. Semelhantes características podem oferecer confiança e efetividade no processo de detecção e resposta, alta taxa de segunça a redes privadas, melhores possibilidades de defesa e, ainda, minimizar as chances do intruso.

Essa dissertação trata da especificação de uma sociedade de agentes para a avaliação e aprimoramento de sistema de resposta de intrusão em redes de computadores. A proposta de um modelo de sistema de resposta de intrusao(IRS) é baseada em várias arquiteturas disponíveis na procura da melhor solução para os problemas encontrados na modelagem de um sistema deste nível. Com isso, foi modelado um sistema que contenha as principais funcionalidades desejáveis para um de respostas ativas. O sistema, que faz parte do NIDIA(Network Intrusion Detection System based on Intelligent Agents) (Lima, 2001), é formado por uma sociedade de agentes que são responsáveis pelas funções de identificação das características do ataque, escolha da melhor estratégia de reação a pela execução resposta.A sociedade é composta por agentes artificiais aptos em determinar e aplicar automaticamente ações, corretivas e preventivas, contra ataques classificados de acordo com um modelo taxonômico de severidade. No modelo proposto procurou-se definir respostas de intrusoes por abuso e por anomalia para garantir maior robustes ao sistema.

Palavras Chaves: sistema de resposta de intrusão, detecção de intrusos, segurança de redes de computadores, sistemas multiagentes, redes neurais.

ABSTRACT

The development of approaches for proving fast reactions against intruders and attackers have been one of the most important requirements in the critical defense of computer networks, since the intrusion occur quickly, demanding reactions without human intervention. These approaches should be able to, autonomously, respond to attacks and deal with several important aspects of the computer security problem in order to reduce the system administrator's workload Such approaches can offer larger reliability and effectiveness in the detection and response processes, a higher rate of security to private networks, better defense possibilities and, in addition, minimize the intruder's change of success.

This research work deals with the specification of a society of intelligent agents for assessment and enhancement of intrusion response systems in computer networks. The proposal of the model of intrusion response system (IRS) be based on in several available architectures, in order to look for better solutions for the problems faced in the modelling of a system of that level. With that, was modeled a system to approach the main desirable functionalities for a system of active answers. The system, as part of the NIDIA (Network Intrusion Detection System based on Intelligent Agents) (Lima, 2001), is formed by a society of agents that are responsible for the functions of identification of the characteristic of the attack, choice of the best reaction strategy and for the execution of the response.The society is composed by agents able to determine and apply automatically corrective actions against attacks classified according to a given severity taxonomic model. In the proposed model was looked for to define response to intrusions for abuse and for anomaly to guarantee a lower robustness to the system

Keywords: intrusion response system, intrusion detection, multi-agents systems, neural networks.

SUMÁRIO

LISTA DE ABREVIATURAS E SÍMBOLOS

AAFID	Agentes Autônomos Para Detecção De Intrusão
AAIRS	Adptative Agent Based Intrusion Response System
AID	The Intrusion Detection System
BAM	Bi-Directional Association Memory
CERIAS	Center of Education and Research in Information Assurance and Security at Purdue University
CERT	Computer Emergency Response Team
CSI	Computer Security Institute
DFDB	Base de Dados de Incidentes de Intrusão
DoS	Denial of Service
EMERALD	Event Monitoring Enabling Responses to Anomalous Live Disturbances
HostAgent	Agente Sensor de Host
IA	Inteligência Artificial
ICMC/USP	Instituto de Ciências Matemáticas de São Carlos/Universidade de São Paulo
IDIP	Intruder Detection and Isolation Protocol
IIDB	Base de Dados de Padrões de Intrusões
LSIA	Agente de Segurança Local
MCA	Agente Controlador Principal
NetAgent	Agente Sensor de Rede
NFS	Network File System
NIDIA	Network Intrusion Detection System Based On Intelligent Agents
PA	Protection Analysis
RABD	Base de Dados de Ações Remediativas
RISOS	Research In Secured Operating Systems
SUA	Agente de Atualização do Sistema
SAI	Agente de Integridade do Sistema
SCA	Agente Controlador de Ações
SDI	Sistema de Detecção de Intrusos
SAA	Agente de Avaliação de Segurança do Sistema
SEA	Agente de Análise de Severidade
SMA	Agente de Monitoramento de Sistema
SOA	Agente Sistema Operacional
SO	Sistema Operacional
SRI	Sistema de Respostas de Intrusão
STBD	Base de Dados de Estratégias
TCP	Transmission Control Protocol
UCP	Unidade Central de Processamento
UFMA	Universidade Federal do Maranhão
UML	Linguagem de Modelagem Unificada

LISTA DE FIGURAS

CAPÍTULO 1

INTRODUÇÃO

Neste capítulo, faz-se a apresentação deste trabalho através de uma visão geral da problemática da segurança das informações, definindo-se algumas técnicas de segurança, expondo suas principais características e limitações. Em seguida, abordam-se os objetivos gerais e específicos e a organização da dissertação.

1.1 Cenário atual

Na atual ordem econômica o uso de tecnologias no armazenamento e na difusão de informação tornou–se fundamental para o progresso das empresas, provocando o crescimento da interconexão entre redes e a troca de dados através destas. Com isso é cada vez maior o acesso a serviços da Internet, como a navegação pelas páginas da World Wide Web (WWW), correio eletrônico (e-mail), Telnet e FTP (File Transfer Protocol).

Este novo panorama traz consigo muitos benefícios às organizações para conduzir os seus negócios, tais como redução de custos e o fornecimento de serviços cada vez mais atraentes aos clientes, além de promover grandes oportunidades de negócios (*business-to-business* e *business-to-customers*).

Segundo a pesquisa realizada pela Módulo Security Solutions (Módulo Security Solution, 2000), que entrevistou 165 executivos de grandes empresas, públicas e privadas, a presença de empresas brasileiras ligadas à Internet é bastante expressiva . Foi constatado que 72% disponibilizam acesso à Internet a seus funcionários através de sua rede interna, 16% permitem o acesso via modem nas empresas e 19% autorizam o acesso remoto via modem na residência. É importante ressaltar também, que 59% das organizações realizam transações eletrônicas através da Web. Dentre as mais utilizadas estão venda (25%), serviços bancários (14%) e compras (7%).

Paralelamente surge a necessidade de se utilizar mecanismos para prover a segurança das transações de informações confidenciais. Segundo o relatório do Internet Fraud Watch (IFW), no ano de 2001, fraudes na Internet geram perdas de 4,3 milhões de dólares. Na lista de fraudes estão serviços de acesso à Internet, serviços de pornografia, venda de equipamento e software, empréstimos, ofertas de cartões de crédito, oportunidades de negócio duvidosas e todo o tipo de ofertas de mercadoria.

A cada ano surgem novas técnicas de ataque e o número de incidentes cresce proporcionalmente ao crescimento da internet. De acordo com o CERT (*Computer Emergency Responce Team*) (CERT, 2002) de 1998 a dezembro de 2001, foram registrados 100.369 incidentes de segurança[1], sendo que 21.756 ocorreram em 2000 e 52.658 só em 2001.

A questão segurança é bastante enfatizada, principalmente quando imagina se a possibilidade de ter informações expostas a atacantes ou intrusos da Internet, que usam meios cada vez mais sofisticados para violar a privacidade e a segurança das comunicações. Existem mais de 80.000 sites na própria Internet dedicados a atividades de *hacking*[2], onde é possível obter ferramentas para invasão de sistemas. Essas ferramentas, muitas vezes automáticas, implementam combinações de diversos tipos de ataque, buscando falhas de configuração e vulnerabilidades[3] inerentes aos sistemas adotados nas diversas redes alvo.

A segurança deve ser entendida segundo vários aspectos, que vai depender da necessidade do ambiente e deverão ser baseadas nos objetivos das transações e no que é preciso proteger. Por exemplo, quando a prioridade é assegurar dados e que a violação de sua integridade pode acarretar prejuízos, a política de segurança estabelecida deve primar pela integridade[4] dos dados. Por outro lado, uma organização que possui informações confidenciais deve se preocupar com o aspecto confidencialidade[5], pois informações confidenciais em

[1] Incidentes de Segurança são tentativas de invasão com ou sem sucesso.
[2] Hacking é o indivíduo com um profundo conhecimento, mas geralmente sem intenções destrutivas. Seu propósito é unicamente provar que consegue invadir um determinado sistema.
[3] Vulnerabilidades são pontos fracos associados a um ativo (por ex. servidor crítico) ou grupo de ativos, os quais podem ser explorados por um atacante.
[4] Integridade é manter informações e sistemas computadorizados, dentre outros ativos (elementos aos quais a organização atribui valor e desta forma requerem proteção), exatos e completos.
[5] Confidencialidade é proteger informações confidenciais contra revelações não autorizadas ou captação compreensível.

mãos erradas podem trazer prejuízos. A disponibilidade[6] requer que o sistema funcione adequadamente fornecendo, quando requisitado, recursos aos usuários autorizados.

A cada dia surgem técnicas mais robustas de proteção de dados, Firewall, criptografia, proxy, entre outras. As técnicas de firewall são muito utilizadas para proteger uma rede contra ataques. No entanto, somente limita o acesso aos objetos no sistema, mas não restringe o que pode ser feito com os objetos em si, devido ao fato da maioria dos sistemas de firewall basear suas políticas de segurança, principalmente, na relação de confiança entre as máquinas.

Desta forma, ainda que os firewalls sejam considerados uma alternativa segura e largamente utilizada, seu uso não implica segurança total, uma vez que o mesmo não protege contra códigos móveis maliciosos[7] (applets Java e objetos ActiveX) e acesso discado. Portanto, atrelado a esse mecanismo de proteção torna-se necessário o uso integrado de diversas tecnologias para a detecção de falhas, intrusos e intrusões em um sistema.

Segundo Bace e Mell (Bace e Mell, 2001) Sistemas de Detecção de Intrusos(SDI) são sistemas de software ou hardware que automatizam o processo de monitoria de eventos ocorridos em sistema de computador ou rede. Os SDI's proporcionam maior grau de segurança às redes de computadores, detectando falhas de segurança, tentativas de intrusão[8], anomalias e abuso em sistemas computacionais e redes.

Atualmente, vistos como uma solução inovadora, os SDI´s estão sendo largamente utilizados nas corporações, instituições governamentais e redes de computadores acadêmicas, como uma poderosa ferramenta de auxílio aos administradores de segurança (Módulo Security Solution, 2001). Segundo o IDC (*International Data Corporation*), o mercado das ferramentas SDI atingiu uma venda de 136 milhões de dólares em 1998. Em 1999, foi verificado um crescimento de aproximadamente 100%, e estima-se que até 2003 atingirá um mercado de 980 milhões de dólares (PRNEWSWIRE ASSOCIATION, 1999).

---

[6] Disponibilidade de um recurso pode ser fator determinante no controle de tráfego aéreo ou possibilidade de retaliar um ataque militar (Cansian, 1997).

[7] Alguns firewalls podem prevenir a importação de código Java e ActiveX através do reconhecimento de tipos MIME. Porém, essa funcionalidade não fornece a habilidade de se distinguir entre um código legítimo e um malicioso e, com isso, tem que ser configurado para aceitar ou não todo tipo de código ActiveX ou Java(Ranum, 2000).

[8] Segundo (Crosbie e Spafford, 1995a), uma intrusão pode ser definida como sendo "um conjunto de ações que tentam comprometer a integridade, confidencialidade ou disponibilidade de recursos" em um sistema computacional.

Um sistema de detecção de intrusão pode ser baseado em rede ou baseado em host. Os sistemas baseados em rede são formados por um conjunto de sensores, em modo promíscuo, que localizados em vários pontos da rede capturam e analisam pacotes em busca de assinaturas de ataque. Este método analisa o tráfego de todos os hosts conectados no mesmo segmento de rede a procura de operações que violam a política de segurança. Um sistema baseado em host irá monitorar as atividades de um único host, examina trilhas de auditoria e arquivos de *logs* monitorando eventos locais de sistema de computador em particular, podendo detectar ataques que não são vistos pelo SDI baseado em rede (Lima, 2001).

Uma intrusão é categorizada em duas principais classes: a intrusão por abuso e por anomalia.O método de detecção de intrusos por anomalia observa os desvios de comportamentos esperados em atividades[9] relevantes dos usuários ou processos do sistema monitorado. Esses tipos de detectores baseiam-se na premissa de que um ataque difere da atividade normal, desta forma constroem perfis (*profiles*), gerados através de dados coletados durante um período normal de operação, que representam o comportamento normal dos usuários, atividades de *hosts* e conexões de rede. Depois de coletados os dados dos eventos, usando uma variedade de medidas, e comparados com os perfis construídos determinam se a atividade monitorada á anômala. Uma desvantagem deste método em SDI é que tais detectores geram um grande número de falso positivo[10], pois nem sempre uma atividade intrusiva coincide com atividade anômala. Então se um usuário modifica o seu comportamento e difere do seu perfil modelado no sistema ele é considerado intruso.

A principal característica da detecção de intrusos por abuso é a observação e codificação das principais atividades intrusivas de um ataque. Essa codificação gera um padrão, chamado de assinatura, capaz de descrever o comportamento intrusivo e que comparado com um evento suspeito pode identificar um ataque. A principal limitação desta técnica é que ela identifica somente vulnerabilidades conhecidas, e por isso precisa ser constantemente atualizado para que conheça novos ataques.

---

[9] Ocorrências detectadas pelos sensores ou analisadores como sendo de interesse do administrador. Ex.: uma sessão de telnet inesperada, usuário tentando alterar objetos sem ter privilégios, arquivos de log mostrando persistentes falhas de logon.

[10] Um evento é considerado falso positivo quando não há intrusão, mas há comportamento anormal.

1.2 Definição do Problema

Nos últimos anos tem-se um crescimento de incidentes e, além de numerosos, os ataque se tornaram mais sofisticados. Isso se deve ao fato que cada vez mais as técnicas de invasão estão acessíveis para os usuários, permitindo que agressores, com conhecimentos técnicos limitados, mas dispondo de ferramentas sofisticadas, possam efetivar ataques bem sucedidos.

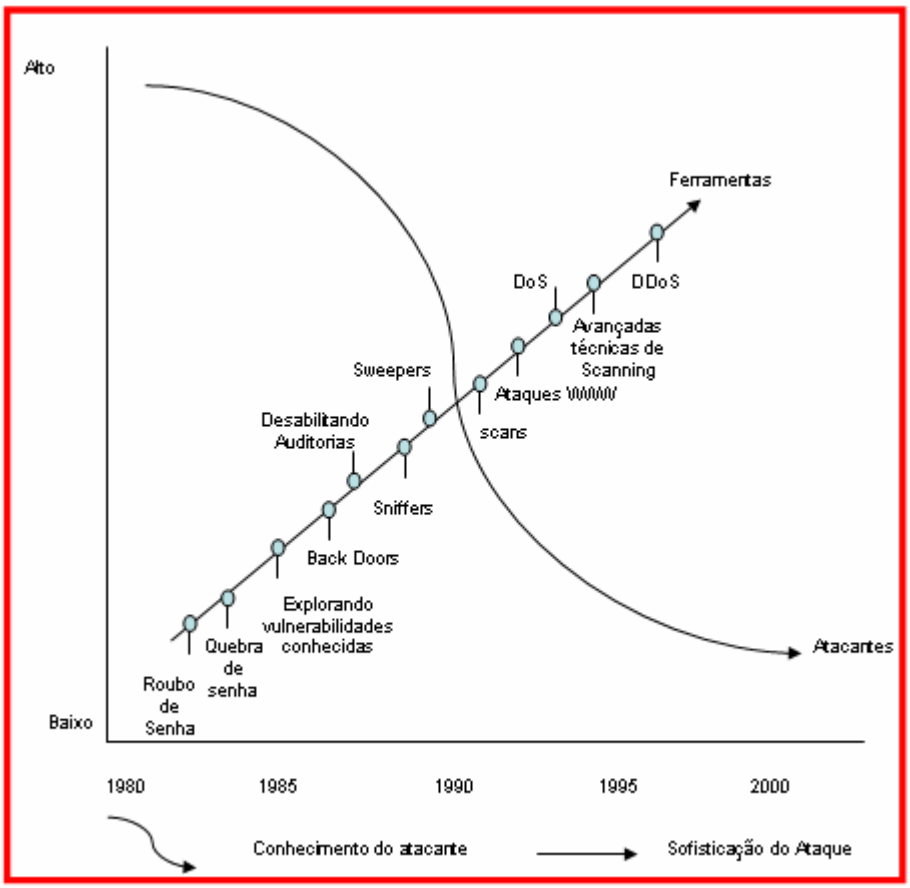


Figura 1. 1 Sofisticação de Ataques x Conhecimentos Técnicos dos Atacantes.

A figura 1.1 mostra que, ao contrário do que ocorre hoje, nos anos 80 os intrusos eram especialistas, com alto nível de conhecimento e desenvolviam seus próprios métodos de ataques para a atingir um sistema. Exploravam scripts e raramente utilizavam ferramentas automatizadas.

Apesar de fortes tentativas, os pesquisadores não conseguem acompanhar o crescimento, a sofisticação e a agilidade das ferramentas de ataques. Os atacantes usam programas modernos, que rapidamente penetram e ganham o controle de um host. Além disso, a maioria dos Sistemas de Detecção de Intrusão

utiliza como métodos de resposta de intrusão processos manuais, como por exemplo a notificação, que somente gera relatório e alarmes indicando ao administrados a existência de um possível comportamento intrusivo e que ele deve reagir. Este mecanismo têm um atraso(delay) entre a detecção da intrusão e a uma possível resposta, levando de minutos a meses. Esse espaço de tempo fornece uma ampla janela de oportunidade para o atacante operar livremente até que o administrador do sistema comece a reagir (Cohen, 1999).

O tempo de resposta a um incidente determina se o atacante terá sucesso ou não. O resultado da simulação feita por Cohen (Cohen, 1999) indica que se um atacante hábil tiver mais que 30 horas para atacar, a habilidade do administrador se torna irrelevante; o atacante terá sucesso. Por outro, se a resposta for instantânea, a probabilidade do sucesso do ataque é quase nula.

Deste modo, respostas rápidas têm se tornado a maior exigência na defesa crítica dos sistemas, adversários agem na velocidade do computador e os sistemas de segurança de rede exigem, cada vez mais, ações mais velozes sem a intervenção humana.

Visando atingir um melhor grau de segurança e melhores possibilidades de defesa, diversos laboratórios de pesquisa desenvolveram algum tipo de sistemas de resposta em seus SDI`s, dentre eles, o SRI International, Menlo Park (Neumann e Porras, 1999), a Universidade de New México (Somayaji e Forrest, 2000), os pesquisadores da Boeing Phantom Works em conjunto com a Network association NAI Labs (Schnackenberg et al., 2000), o departamento de ciência da computação da Universidade Texas A&M (Carver et al., 2000), a Cisco System (Cisco System, 2002).

A arquitetura conhecida como “Event Monitoring Enabling Responses to Anomalous Live Disturbances” (EMERALD) (Neumann e Porras, 1999), desenvolvida pelo *Laboratório de Ciência da Computação do SRI International*, é um SDI com uma estrutura hierárquica distribuída, composta por agentes, que detecta intrusões por anomalia e abuso utilizando sistema especialista. As respostas de intrusão são providenciadas pelo resolver agent. O resolver agent recebe o relatório sobre um ataque e providência a resposta levando em consideração o percentual mínimo necessário para que um incidente seja considerado uma intrusão (threshold metric) e o nível de severidade da resposta (severity metric). Para a escolha da

estratégia de resposta o EMERALD utiliza tabela decisão, o que torna o sistema de resposta inflexível. Esse mecanismo possui somente um mapa estático que faz a relação entre o ataque e a intrusão, ou seja, a incidência de um mesmo ataque por mais de uma vez em um mesmo sistema sempre terá a mesma estratégia de resposta independente do seu estado (fase inicial ou avançado) e se o ataque pode ser um falso positivo ou não.

A Universidade de New Mexico desenvolveu um projeto chamado "Automatic Response Using System-Call Delays" (Somayaji e Forrest, 2000). O sistema PH (Process Homeostaisis) apresenta mecanismos de respostas automáticas contra comportamentos anômalos, baseados em mecanismos homeostáticos[11] (Sterne et al.,2001), que permitem o computador preservar a sua integridade, detectar e parar anomalias sem comprometer o sistema alvo. O PH possui soluções somente para comportamentos anômalos, não possui respostas para ataques de rede. Ainda em fase de desenvolvimento, o modelo possui limitações quanto a obtenção de perfis de comportamento normal e anômalo.

O sistema "Intruder Detection and Isolation Protocol" (IDIP) (Schnackenberg et al., 2000) é um protocolo de camada de aplicação, organizados em comunidades (formada por host , SDI e boundary), que coordena investigação, detecção e resposta automáticas para ataques de rede. No entanto, as constantes trocas de mensagens entre as comunidades para a escolha da estratégia de resposta pode afetar a performance do IDIP levando mais tempo para o inicio de uma reação. Não possui respostas contra anomalias no sistema, ou seja, se o sistema for vítima de um intruso executando atividades ilegais a reação irá depender do administrador do sistema.

O "Adaptative Agent based Intrusion Resposne System" (AAIRS) (Carver et al., 2000) é um modelo de um sistema de resposta de intrusão que prevê a capacidade de respostas adaptativas. Portanto, se uma resposta em operação de reação, obtiver um resultado melhor sobre as outras respostas esta será a mais utilizada. Ainda está em fase de desenvolvimento e não podemos afirmar nada sobre a seu desempenho e efetividade.

O CiscoIDS (Cisco System, 2000) também conhecido como NetRanger, é um sistema de detecção de intrusão e resposta apresentado pela Cisco System.

[11] Homeostácia é a tendência do sistema de manter o seu equilíbrio interno.

O NetRanger é baseado em redes e detecção por abuso, faz monitoramento de pacotes em tempo real e quando faz a análise dos pacotes , baseada em regras, procura por padrões de intrusão por abuso e gera uma resposta. As respostas podem ser através de relatórios, alarmes, ferramentas de respostas manuais e resposta automáticas. O NetRanger ainda não é totalmente automático, sua principal forma de defesa é gerar alarmes e disponibilizar resposta manuais para que o administrador escolha alguma estratégia de resposta.

Visto a impossibilidade de se obter um Sistema Detecção completamente seguro e preciso e as limitações dos mecanismos de resposta utilizados, faz se necessário o uso de outras tecnologias autônomas e dinâmicas para aperfeiçoar o processo de reação contra incidentes. Atuando em conjunto com os SDI's, na tentativa de superar as limitações destes sistemas, um sistema de respostas automáticas pode oferecer maior confiabilidade e efetividade no processo de detecção e resposta, um elevado grau de segurança as redes privadas e melhores possibilidades de defesa, além de provocar a redução da probabilidade de um intruso obter sucesso.

Acima foram mostradas algumas pesquisas recentes sobre SDI's, que utilizam algum tipo de resposta, constatando-se a imaturidade da grande maioria das pesquisas de sistemas de resposta de intrusão. Alguns sistemas ainda permanecem utilizando processos manuais e de notificação e outros somente nos últimos anos começaram apresentar soluções mais apropriadas e robustas para sistemas de respostas automáticas.

Portanto, a continuidade dos projetos e a busca de métodos alternativos de respostas automáticas têm sido uma preocupação constante dos pesquisadores. Percebe-se a necessidade de mecanismos de resposta automática que tragam maior eficiência e confiabilidade ao processo de detecção de intrusão e resposta, promovendo melhor grau de segurança e melhores possibilidades de defesa aos SDI's. Nesse contexto, a presente proposta representa mais uma possibilidade de expansão do conhecimento teórico e prático de novas tecnologias aplicadas a essa área.

1.3 Objetivo geral e específico

Esta dissertação objetiva propor um modelo de sistema de reposta de intrusão automático, baseado na noção de sociedade de agentes inteligentes, capaz de tratar e responder automaticamente intrusões em uma rede de computadores de forma bastante flexível. O projeto denominado Agente Controlador de Sistema (SCA), que compõe a arquitetura “Network Intrusion Detection System based on Intelligent Agents” (NIDIA) (Lima, 2001), visa proteger um sistema através de medidas de emergência ou preventiva, evitando ou parando um ataque, antes que o alvo seja completamente comprometido.

Este trabalho tem como objetivos específicos:

.

- apresentar uma sociedade de agentes para fornecer respostas automáticas de acordo com a necessidade do NIDIA;
- definir as funcionalidades básicas de cada agente do modelo SCA e a interação destes em sociedade através da Linguagem de Modelagem Unificada (UML) (Booch et al., 1999);
- apresentar a implementação de um protótipo, mas especificamente do agente BAM, responsável pela identificação do nome e das características do ataque;
- demonstrar a viabilidade da adaptação do sistema de reposta de intrusão (SCA) com o sistema de detecção NIDIA , enfatizando-se a situação atual e os futuros trabalhos a serem desenvolvidos.

1.4 Organização da Dissertação

Esta dissertação encontra-se organizada em cinco capítulos. No Primeiro Capítulo é apresentado o cenário atual do mercado das ferramentas de segurança, especificamente os Sistemas de Detecção de Intrusos, o crescimento progressivo do uso da Internet voltados para atividades comerciais, informações técnicas quanto ao aumento de incidentes registrados nos últimos anos, bem como os objetivos gerais e específicos e a organização da dissertação.

No Capítulo 2, apresenta-se uma visão geral dos principais mecanismos de resposta de intrusos, características e limitações. Definindo-se também as principais técnicas de resposta e os trabalhos relacionados nessa área.

No Capítulo 3 é proposto um modelo de sistema de resposta de intrusos (SRI), denominado SCA, baseado na tecnologia de agentes. Apresenta-se a sua arquitetura e funcionalidades através de Casos de Uso da notação UML.

O Capítulo 4 apresenta a operacionalização do modelo proposto, destacando-se a implementação do parcial do agente BAM.

O Capítulo 5 apresenta as conclusões finais, enfatizando as contribuições desta dissertação e algumas indicações para trabalhos futuros.

CAPÍTULO 2

SISTEMA DE RESPOSTA DE INTRUSÃO (SRI)

Neste capítulo apresenta-se uma visão geral dos sistemas de resposta de intrusão, como ferramenta de auxílio na detecção e eliminação de intrusão e os requisitos necessários para se obter um SRI eficiente. Definindo-se também os principais mecanismos de resposta de intrusos, suas características e limitações e as principais técnicas de resposta.

2.1. Tipos de Resposta de Intrusão

Nos últimos anos, pesquisadores obtiveram um alto nível de sofisticação e eficiência na detecção de intrusões. Sistemas de detecção de Intrusão (SDI) conseguem detectar em pouco tempo, utilizando as mais variadas técnicas, diversos tipo de ataques. No entanto, a evolução dos ataques é maior que a evolução dos SDI´s; isso demanda mecanismo de resposta que permita interromper um ataque antes que este obtenha sucesso e o alvo seja comprometido. Um estudo feito por Cohen (Cohen, 1999), mostra que em apenas 8 horas, mais de 2000 tentativas de invasão foram feitas, usando ferramentas automáticas de ataque, originadas de 500 localizações diferentes contra um único sistema.

Resposta pode ser definida como um conjunto de ações que o sistema tem quando se detecta uma intrusão (Bace, 2000). Existem tipicamente dois grupos de respostas as ativas e as passivas. Uma ação ativa é disparada automaticamente pelo sistema sem a intervenção humana, e uma ação passiva é aquela que envia relatórios dos eventos ocorridos e espera uma reação do administrador de sistema .

As três principais formas de responder uma intrusão são: através de notificação, de reposta manual ou reposta automática.

Notificação

Grande parte dos sistemas de detecção de intrusão e resposta usa somente a notificação. Neste método, alarmes e notificações são geradas para informar ao administrador a ocorrência de uma intrusão.

Periodicamente são gerados relatórios, que podem ser diários ou até mensais, para auxiliar o administrador do sistema na investigação de uma potencial intrusão. A freqüência com que os relatórios são gerados delimita a janela de oportunidade que um atacante pode ter para explorar (Carver,2000), deste modo, enquanto o administrador espera por um relatório ser gerado, que pode durar meses, o intruso pode operar livremente sem que seja descoberto.

Os alarmes são gerados logo que um ataque é detectado. Apesar de menos tardio e de reduzir a possibilidade de um intruso ter sucesso, o sistema ainda depende de uma reação humana. Nos SDI's mensagens urgentes de alarme são enviadas através de e-mail, serviços de mensagem texto, da ativação de mensagem de Pager, e atualmente, através dos telefones móveis. Em uma mensagem pode conter informações do endereço IP da origem do ataque, do alvo do atacante, a ferramenta usada para atacar e a conseqüência que ataque pode trazer ao sistema.

Resposta Manual

Alguns sistemas de detecção e resposta disponibilizam um conjunto de repostas programadas para que o administrador do sistema inicie uma reação manual. Estes sistemas guiam o administrador na tomada de decisão e permite que ele tome a decisão final da resposta apropriada.

No entanto, este mecanismo ainda oferece ao intruso grandes oportunidades de ataques. Existe um espaço de tempo, entre a detecção da intrusão e o inicio da reação do administrador, que o atacante pode roubar dados críticos, instalar back doors ou usar o alvo para atingir um outro host. Além disso, o administrador pode não está no momento em que ocorrer a intrusão e então ela pode se completar antes que haja alguma reação humana.

Resposta Automática

Um sistema de resposta automático reage automaticamente a comportamento intrusivo sem precisar da intervenção humana. É um método que

permite descobrir a origem e o tipo de ataque e responde de forma rápida reduzindo a probabilidade de sucesso de um atacante.

A seguir estão relacionadas algumas habilidades que um sistema automático deve possuir:

- Habilidade para dinamicamente remover o intruso do sistema alvo;
- Habilidade para monitorar todo o tráfego da rede e identificar portas e protocolos afetados;
- Capacidade de, dinamicamente, modificar a tabela de regras de firewall e routeadores;
- Habilidade para bloquear porta, encerrar conexão usada pelo atacante e rejeitar tráfico vindo do endereço do atacante;
- Habilidade para dinamicamente alterar configurações do sistema ou prioridades do usuário.

Os sistemas de respostas automáticos, quanto à estratégia de reação, apresenta-se de duas formas: baseado em tabelas de decisão e baseado em regras.

Estratégias vindas de tabelas de decisão são inflexíveis porque sempre uma estratégia de resposta está associada a um tipo específico de ataque. Ou seja, se um mesmo ataque incidir várias vezes sobre o mesmo sistema este irá reagir sempre da mesma forma. Além disso, como as tabelas são alimentadas manualmente, na ocorrência de uma intrusão que não tem solução configurada o sistema poderá ficar resposta.

No entanto, nos sistemas baseados em regras a escolha da estratégia de resposta depende da análise dos incidentes (grau de severidade, o perigo que este oferece ao sistema, quais vulnerabilidades exploradas). Deste modo, a análise das evidências detectadas que vai determinar o tipo de resposta a ser aplicado. Ou seja, se um mesmo ataque incidir mais de uma vez no sistema ele poderá ter reações diferentes, isso vai depender do estudo da suas informações. Esse modelo aplica medidas que estejam configuradas no seu banco de dados, ou seja, solução para um novo ataque o banco de dados de reação terá que ser realimentado.

2.2. Técnicas de Resposta de Intrusão

Existe uma grande variedade de técnicas para combater ataques. Usando destas técnicas o sistema pode bloquear um usuário, cancelar uma conexão TCP ou somente gerar um alerta para o administrador. A seguir são descritas as principais técnicas de reação:

Gerar Relatório : deve ser gerado um relatório de todo comportamento intrusivo detectado para que administrador de sistema acompanhe o andamento dos incidentes. Estes relatórios irão fornecer uma visão geral do que ocorreu com o sistema e permitir a análise segura dos ataques ao longo do tempo.

Gerar Alarmes : alarmes notificam o administrador a detecção de um ataque ou de uma tentativa de ataque contra o sistema. As mensagens de alerta são enviadas através de e-mail, de janela de notificação pop up, da ativação de mensagem de Pager e ainda envio de mensagem para telefones celulares. Nem todo comportamento intrusivo é gerado alerta, só serão enviadas mensagens de alerta se incidir várias tentativas fracassadas de intrusão de um mesmo usuário ou quando se tiver certeza de intrusão.

Suspender Jobs e Encerrar a Sessão do Usuário: quando se tem operação de usuário normal e indicação de atividades intrusivas, são suspensos os jobs do usuário, a sua sessão encerrada e sua senha bloqueada antes que atividades válidas sejam corrompidas e que sejam causados danos ao sistema.

Investigação de Suspeitos: devido ao alto grau de falso positivo deve se evitar medidas severas, deste modo, quando houver suspeitas que um comportamento é intrusivo, é bloqueada a execução de certos tipos de comando sem excluir totalmente o usuário do sistema, são monitoradas as suas atividades para adquirir informações adicionais e assim classificar definitivamente o comportamento do usuário. Essa medida viabiliza reunir as informações adicionais necessárias para a classificação de um ataque ou atacante e irá reduzir a possibilidade do intruso causar dano ao sistema antes que ele seja excluído.

Bloquear Endereço IP: quando se tem um ataque de rede e a sua origem é descoberta, como resposta imediata, pode ser obstruído no roteador todo o seu tráfego. Mas essa solução é provisória e só permite que se tenha mais tempo para

reagir. Um atacante que tem seu endereço IP bloqueado pode mudar o seu IP e atacar novamente.

Derrubando o Host : a única forma de proteger um sistema comprometido contra um avançado ataque é desligando a máquina. Apesar de ser uma medida rígida, essa é a única forma de proteger de maiores danos um host tomado por um ataque.

Usar uma Ferramenta de Detecção de Intruso Adicional: devido à deficiência das ferramentas de detecção de intrusão e o alto consumo de recurso do sistema, sistemas de resposta de intrusão podem solicitar ajuda de outros meios de detecção de intrusão para a definição do estado de uma intrusão. Neste caso, ferramentas de detecção de intrusão são empregas como indicadores adicionais de que um comportamento intrusivo foi achado.

Desabilitar Porta e Serviços Afetados: quando um ataque usar um serviço ou uma porta conhecida para atingir um sistema, é possível parar efetivamente o ataque, através do bloqueio dos recursos atingidos, sem afetar qualquer outro serviço do sistema.

Investigar a Conexão: A investigação de uma conexão de rede para tentar identificar um atacante é uma reposta viável, por que além de descobrir a identidade do intruso ele pode detectar que está sendo investigado e se intimidar desistindo do ataque.

Criar Backup: ataques contra a integridade do sistema podem causam pouco impacto se forem criados backups do sistema. Os backups atualizados fazem a restauração do sistema e a comparação de arquivos. Apesar de ser impossível manter backups atualizados em tempo real, a medida que for aumentado o grau de ocorrência de incidentes deve ser reduzido o intervalo de tempo entre os backups, assim reduz a perda de dados e números de arquivos corrompidos.

Utilizar Arquivo Temporário de Proteção: segundo a proposta de Fisch (Fisch, 1996), arquivo temporário de proteção serve como um mecanismo para assegurar a integridade do sistema enquanto estiver sendo vítima de um ataque. Uma cópia do arquivo original é criada e codificada sempre que alguém tentar modificar arquivos críticos do sistema, todas as modificações são salvas em um segundo arquivo e o original permanece intacto. Uma adicional modificação é feita no arquivo temporário.

2.3. Sistemas de Detecção de Intrusão Baseados em Agentes e seus Mecanismos de Resposta

Atualmente, os sistemas de detecção de intrusos que utilizam arquiteturas monolíticas, enfrentam problemas referentes à:

i) tolerância a falhas – um sistema monolítico possui um ponto de falha, que pode ser facilmente atingido através de ataques, do tipo negação de serviço (DoS), ao servidor o qual o sistema está instalado;

ii) dificuldades em reconfigurar ou adicionar capacidades ao sistema em tempo real;

iii) eficiência - Os SDI´s precisam ser reconstruídos para poderem combater novas formas de intrusão que não estejam previstas no sistema e para serem adaptados a sistemas de respostas de intrusão automáticas;

iv) necessidade do uso de pessoal técnico qualificado e de ferramentas para manter o conhecimento;

v) escalabilidade;

De acordo (Crosbie e Spafford, 1995; Zamboni et al, 1998) segue-se que, a partir dessas desvantagens, existem alguns fatores motivadores quanto ao uso de agentes inteligentes para detectar intrusões, os quais venham garantir as características desejáveis em um SDI:

i) os agentes podem ser adicionados e removidos do sistema sem precisar reiniciá-lo, tornando fácil a sua manutenção e atualização e oferecendo flexibilidade ao sistema;

ii) os agentes tornam o sistema adaptável podendo automaticamente identificar e combater novos ataques;

iii) os agentes podem ser ativados e desativados dinamicamente para realizarem suas tarefas, proporcionando, com isso, melhor uso dos recursos do sistema;

iv) um agente pode ser configurado especificamente para as necessidade de um host ou uma rede, aumentando o poder de configuração do sistema;

v) o uso de agentes garante maior tolerância a falhas do que em sistemas monolíticos;

vi) O uso de agentes permite que respostas contra intrusão sejam aplicadas mais rapidamente, oferecendo maior eficiência ao sistema;

vii) agentes podem ser adicionados para aumentar a capacidade do SRI, permitindo que os mesmos operem em sistemas maiores, e com isso, oferecer escalabilidade ao sistema;

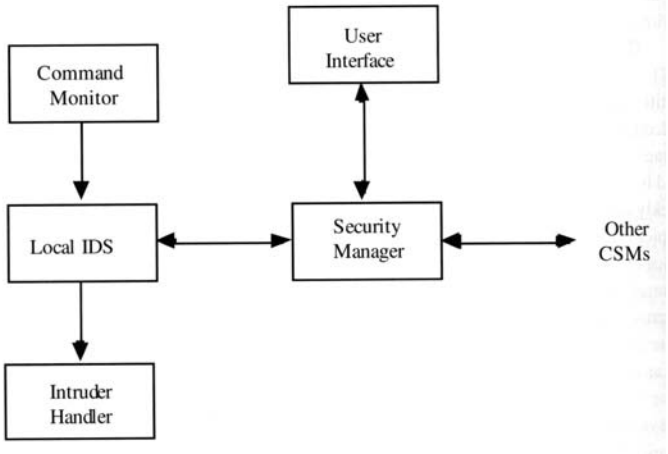
viii) O uso de agentes torna um SRI dinâmico, permitindo detectar e adaptar-se as mudanças no ambiente provocada por um incidente;

ix) Agentes podem fornecer informações suficientes para que se tenha uma resposta eficiente e em tempo-real.

A seguir é feita uma análise de alguns Sistemas de Detecção de Intrusão baseados em agentes. Serão destacadas suas principais características, seus métodos de análise e de detecção e, principalmente, os tipos de respostas por estes utilizados.

Autonomous Agents for Intrusion Detection (AAFID): (Zamboni et. al, 1998) nessa arquitetura, os agentes, responsáveis por capturar as informações dos servidores, residem em uma plataforma de agente chamada *transceivers*. Eles controlam, analisam e processam as informações recebidas pelos agentes. Nesse sistema, existe um *transceiver* para cada servidor monitorado. Os *monitors* são responsáveis pelo controle global do sistema. Eles controlam, analisam e processam as informações recebidas dos servidores monitorados, detectando, dessa forma, eventos suspeitos ocorridos em diferentes servidores. O autor destaca a necessidade de um sistema de respostas para o AAFID, mas a única forma de resposta de intrusão que este sistema utiliza são os relatórios gerados para o administrador do sistema. Esse protótipo, que atualmente está na sua segunda versão (AAFID2), foi implementado em Perl, para ser executado em ambientes Unix.

Cooperating Security Managers (CSM): (White et al., 1996) é um sistema de detecção baseada em redes e em Host, utilizando arquitetura de agentes. CSM é formado por cinco componentes mostrados na figura 2.1. O *Command Monitor* captura comando de usuários e envia para o Local IDS. O Local IDS é um sistema de detecção baseado em host que trata de intrusões no sistema. Os dados vindos da rede são enviados para o *Security Manager* que examina os dados e informa que segmento de rede deve ser vigiado. Se uma intrusão for detectada o *Intruder Handler* inicia uma reação. Para a reação o CSM irá considerar um nível de suspeita, que indica o quanto o incidente está próximo de ser um ataque. A resposta é lançada automaticamente pelo sistema e se falhar o sistema lança uma outra resposta.

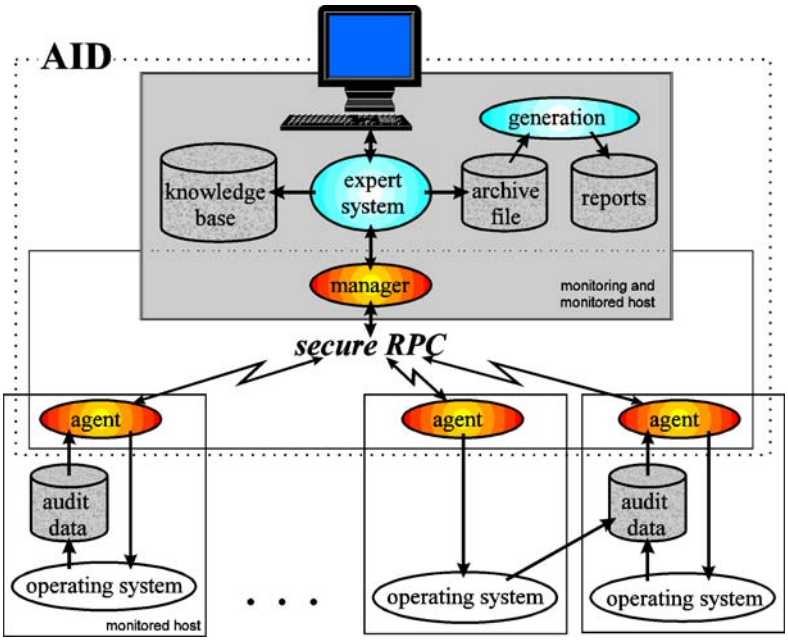


Fonte: (White et al., 1996)

Figura 2. 1 Arquitetura CSM

Adaptive Intrusion Detection (AID): (Sobirey, 2003) é um sistema de detecção baseado em regra que aplica o método de analise por abuso. O AID possui uma arquitetura cliente-servidor (figura 2.2) formada por um monitor central e vários monitores de host (agentes). O monitor central fica localizado na estação, ligado a um sistema especialista, e os agentes localizados nos hosts para fazer o monitoramento e a coleta de dados. Os agentes transformam os dados para um formato padrão e os envia para o monitor central que, através do sistema especialista, analisa as informações e identifica o ataque baseado na análise de

assinaturas. Disponibiliza uma interface gráfica pra o administrador do sistema tenha acesso as informações do ataque e gera relatório dos incidentes ocorridos.



Fonte: (Sobirey, 2003)

Figura 2.2 - Arquitetura AID

Hummingbird : (Frincke et al., 1998) este IDS utiliza os métodos de detecção por abuso e por anomalia. Emprega redes de Petri para representar padrões de ataque e perfis de usuários para detectar ataque e atividades anômalas de usuários. Hummingbird como reação contra ataque gera relatórios e alertas para classificar uma intrusão ou avisar a chegada de um possível ataque. O administrador pode também acionar algumas ferramentas para a coleta adicional de dados e para o auxílio da eliminação do intruso do sistema.

Intrusion Detection Agent (IDA): (Asaka et al., 1999) é baseado em host e faz detecção por abuso. Procura por evidencia de intrusões suspeitas e quando acha utiliza quatro agentes da arquitetura para a investigação. O manager usado para determinar se existe alguma intrusão no sistema. Os sensors responsáveis pelo monitoramento local do sistema, procura por evidencias de ataque e avisa o maneger. O trancing agent que é um agente móvel que tentar descobrir a origem do ataque. O information gatherers que é o agente acionado pelo trancing agent para coletar informações no sistema do host. O seu sistema de resposta de intrusão gera relatórios e faz investigação automática para a descoberta da origem do ataque.

A seguir uma tabela destaca e compara as principais características dos sistemas mostrados acima. Serão considerados os métodos e os tipos de análise utilizados, o tempo de coleta e os tipos de resposta implementados.

Característica / Protótipo	Método de Análise		Tipo de Análise		Tempo de Coleta		Tipo de Resposta		
	Abuso	Anomalia	Rede	Host	Real	Batch	Relatório/ Alarmes	Manual	Automática
AAFID	X		X	x	x		x		
CSM		x	X	x	x				x
AID	X			x	x		x		
Hummingbird	X	x		x	x		x	x	
IDA	X			x	x		x		x

Tabela 2.1 Comparativo entre protótipos de SDI´s

2.4. Considerações Finais

Neste capítulo foram apresentados os principais conceitos e as principais características dos Sistemas de Resposta Intrusão e das técnicas de resposta. Observou-se que os métodos de resposta de notificação e alarme são necessários, mas não são suficientes para um SRI, além de dependerem da intervenção humana, são muitos lentos para combater a velocidade e a sofisticação dos atuais ataques.

Discutem-se também as razões para o emprego da tecnologia de agentes aplicada na detecção de intrusos e resposta, mostrando-se as suas vantagens sobre os sistemas monolíticos. De fato, a abordagem de agentes é desejável, pois fornece flexibilidade a um sistema de respostas automáticas, promovendo vigilância contínua, tolerância a falhas, resistência à subversão, sobrecarga mínima e elevado nível de escalabilidade e configuração, permitindo a inclusão de novas estratégias de resposta sem que seja necessário para totalmente o sistema.

Foram mostrados algumas pesquisas recentes sobre SDI e seu mecanismo de resposta, constatando-se a imaturidade da grande maioria quanto ao desenvolvimento de sistemas de respostas automáticas. Portanto, uma solução automática de resposta que irá possibilitar a reação contra ataques a qualquer momento sem que seja necessária a intervenção humana, se mostra bastante necessária no cenário atual devido a diversos fatores discutidos acima (as vulnerabilidades dos SDI, as limitações dos mecanismos manuais de resposta e o

surgimento acelerado de novas e ágeis técnicas de ataque). A seguir será apresentada uma proposta de um sistema de respostas automáticas que representa mais uma possibilidade de expansão do conhecimento teórico e prático de novas tecnologias aplicadas a essa área. O correspondente detalhamento será apresentado no Capítulo 3.

CAPÍTULO 3

PROPOSTA DE UM SISTEMA PARA RESPOSTAS AUTOMÁTICAS

Neste capítulo é proposto um modelo de sistema de Resposta Automática de Intrusão para o NIDIA, utilizando-se a tecnologia de agentes inteligentes, com o objetivo de adicionar ao NIDIA características desejáveis para se obter um processo de reação eficiente e um SDI que ofereça maior segurança a um sistema. Será apresentada a arquitetura e as funcionalidades dos agentes em sociedade, fornecendo uma visão de modo a delimitar e justificar a contribuição deste trabalho.

3.1. Considerações Iniciais

A proposta do modelo de sistema de resposta de intrusão apóia-se em diversas arquiteturas disponíveis, a fim de buscar melhores soluções para os problemas enfrentados na modelagem de um sistema desse nível. Com isso, buscou-se modelar um sistema que abordasse as principais funcionalidades desejáveis para um sistema de respostas ativas.

No modelo proposto buscou-se definir respostas para intrusões por abuso e por anomalia para garantir uma robustez maior ao sistema. A escolha se deu em virtude do NIDIA utilizar as duas metodologias para a análise detecção dos ataques. Para isso, o sistema proposto é formado por uma sociedade de agentes que é responsável pelas funções de identificação das características do ataque, escolha da melhor estratégia de reação e pela execução da resposta.

Na Identificação das características do ataque, o sistema utiliza uma rede neural BAM (Bi-Directional Associative Memory) que ao receber a notificação da ocorrência de um ataque irá identificar as principais informações necessárias para que se tenha uma reação contra o ataque (origem, nome do ataque, vulnerabilidade explorada, entre outras).

Um agente recebe os dados da rede Neural BAM, consulta, em tempo real, um banco de dados de estratégias de reações e de acordo com as informações recebidas identifica a melhor forma de reação. Dentre os fatores que mais influenciam essa escolha temos: o tipo de vulnerabilidade explorada, o grau de suspeita do ataque e o tempo que se tem pra reagir.

As respostas serão executadas por um conjunto de agente. A escolha do agente adequado para a execução de uma determinada reação vai depender do tipo de estratégia adotada na fase anterior. As reações são as mais diversas, podendo-se simplesmente enviar notificações ao administrador ou reagir de forma mais ativa, coletando-se informações adicionais do atacante (através de um ambiente virtual – *D*ecoy Serve*r (Honeypot*s[12])), alterando-se as configurações do ambiente ou realizando-se ações diretas contra o intruso.

Um dos propósitos do modelo é possuir a capacidade de interação do Decoy Server com sistemas tipo *firewall*, no sentido de diminuir os problemas apresentados, permitindo que seja alcançado um nível de segurança desejado, uma vez que os dois sistemas possuem características complementares.

Outra característica importante da arquitetura proposta é prevê a implementação de rotinas de interação com o sistema operacional, deste modo, sempre que forem detectadas variações no comportamento de usuários ou no perfil das atividades do sistema estas rotinas serão ativadas. Atuando de forma continua tais rotinas irão fazer pequenas correções no sistema sem afetar o seu estado normal de funcionamento, para detectar e parar uma intrusão; obtendo um sistema seguro sem comprometê-lo.

3.2. O Projeto NIDIA

A proposta do sistema NIDIA (*Network Intrusion Detection System based on Intelligent Agents*) (Lima, 2001) é apresentar um sistema de detecção de intrusão, composto por um conjunto de agentes, propondo um modelo de detecção de intrusos, em tempo real, baseado na noção de sociedade de agentes inteligentes capaz de detectar novos ataques através de uma rede neural.

[12] Um ambiente projetado para atrair invasores potenciais, ao mesmo tempo em que “coleta” evidências (ou não) para levar o intruso a sofrer uma possível punição legal.

A proposta do modelo de sistema de detecção de intrusos apóia-se em diversas arquiteturas disponíveis, a fim de buscar melhores soluções para os problemas enfrentados na implementação de um sistema desse nível. Com isso, buscou-se modelar um sistema que abordasse as principais características desejáveis, principalmente quanto às respostas aos incidentes, visto que muitos sistemas não possuem respostas ativas às tentativas de intrusão.

O modelo proposto prevê a metodologia de detecção por abuso e anomalia para garantir uma robustez maior ao sistema. Entretanto, atualmente utiliza-se somente a detecção por abuso como método de análise. A escolha se deu em virtude da grande maioria dos ataques poderem ser codificados, de maneira a capturar e registrar variantes a cerca de atividades que exploram as mesmas vulnerabilidades. Para isso, diversos agentes são responsáveis pelas funções de monitoramento, análise dos dados, detecção e resposta às atividades suspeitas.

No monitoramento, o sistema adotou uma combinação de agentes sensores (*Agente Sensor de Rede e Agente Sensor de Host*) que serão instalados em pontos estratégicos da rede e em *hosts* críticos com o objetivo de capturarem pacotes suspeitos e atividades maliciosas para serem analisadas.

Os agentes coletam e analisam as informações em *batch* ou em tempo real, dependendo do ambiente e da política de segurança da organização. Os fatores que mais influenciam essa escolha são o nível de risco associado, o consumo de memória e recursos disponíveis e o papel que o próprio sistema de detecção deve desempenhar.

Outras características importantes do sistema proposto são que: *i)* o modelo é capaz de identificar intrusões que nunca tenham ocorrido, através de sua capacidade de generalização, além de poder-se adicionar novos padrões de ataques à medida que forem surgindo, com a utilização de uma rede neural; *ii)* ele pode adaptar-se às estratégias de segurança da organização, fornecendo informações a respeito de tentativas de intrusão, bem como as ações tomadas, permitindo que o administrador de segurança tome conhecimento do que está ocorrendo na rede e *iii)* ele é projetado para operar nos ambientes Unix e Windows NT.

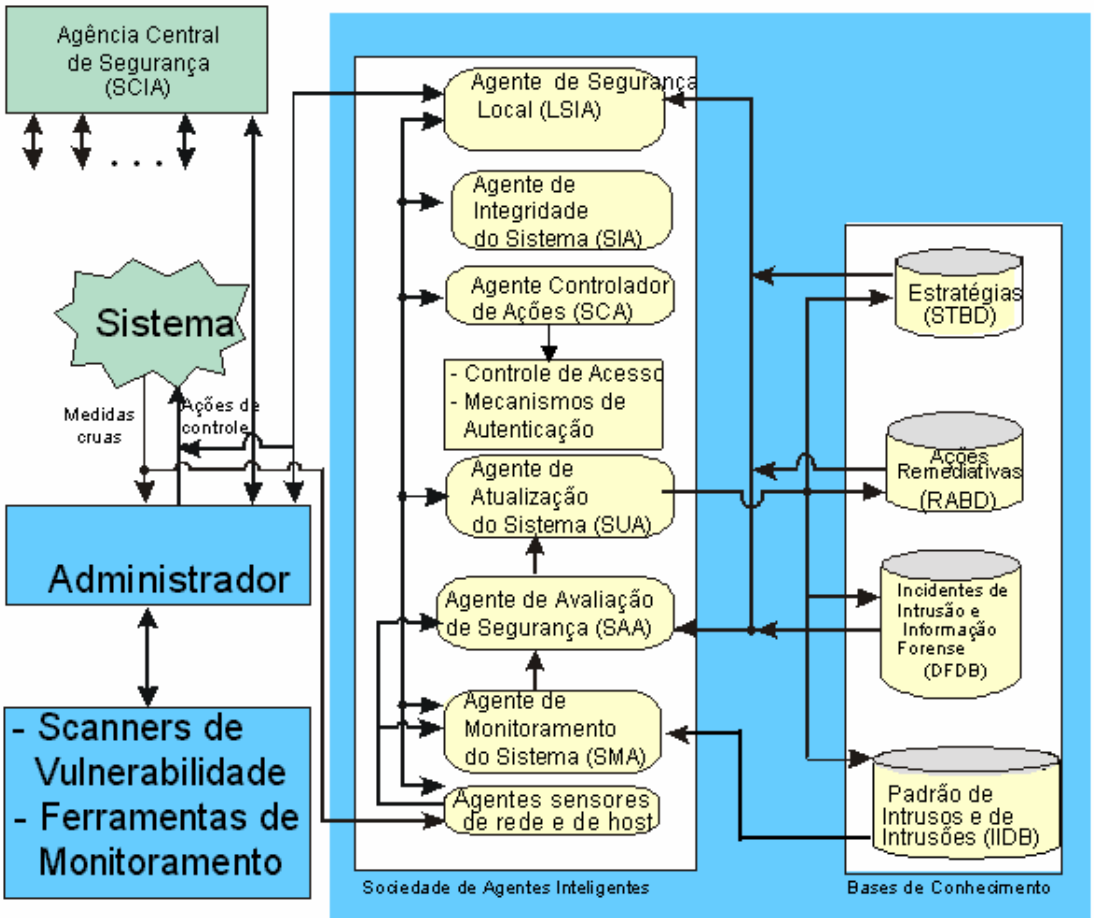


Figura 3.1 Arquitetura Geral do NIDIA

A arquitetura do sistema NIDIA (Lima, 2001) é composta pela cooperação entre agentes artificiais (*Agentes Sensores, Agente de Monitoramento, Agente de Avaliação de Segurança, Agente de Atualização, Agente Controlador de Ações, Agente de Integridade e Agente de Segurança Local*) e agentes humanos (*Administrador de Segurança e Agência de Inteligência Central de Segurança*). Os agentes sensores são localizados em pontos estratégicos vigiando a rede e realizam suas habilidades.

A seguir, descrevem-se as responsabilidades de cada agente (artificial e humano) na arquitetura proposta:

Agentes Sensores: esses agentes são responsáveis por capturar as informações que estão trafegando na rede e no servidor e enviá-las para o agente de monitoramento (SMA). O modelo propõe o uso de dois tipos de agentes sensores: **os agentes sensores de rede (NetSensor)** e os **agentes sensores de host**

(HostSensor). Os agentes sensores de rede são responsáveis pela captura dos pacotes que estão trafegando na rede monitorada (*sniffing*), enquanto que os agentes sensores de *host* coletam informações em arquivos de *log* em *hosts* críticos.

Agente de Monitoramento de Sistema (System Monitoring Agent - SMA): é responsável por organizar e formatar os eventos ou conjunto de eventos coletados, de forma que possam ser identificados padrões de ataques e comportamentos anormais na rede e servidores monitorados, de acordo com a base de dados de padrões de intrusões (IIDB) e atribuir um grau de risco representado pelo evento detectado.

Agente de Atualização do Sistema *(*System Updating Agent - SUA): é responsável por consultar e manter atualizadas as bases de dados do NIDIA. As principais são base de dados de incidentes de intrusão e informações forenses (DFDB), base de dados de padrões de intrusos e intrusões (IIDB), base de dados de ações remediativas (RABD) e base de dados de estratégias (STBD). Quando acionado informa aos outros agentes, através da consulta a base de dados, informações sobre os ataques, histórico dos incidentes e a melhor forma de reagir contra um determinado incidente.

Agente de Avaliação de Segurança do Sistema (System Security Assessment Agent - SAA): é o agente responsável por verificar se o evento corresponde, de fato, a uma tentativa de invasão ou trata-se apenas de um falso positivo. Para auxiliar nessa tarefa, ele faz uso de informações das bases de dados DFDB e estratégias STBD. Como saída, esse agente gera um nível de severidade do evento e um vetor binário contendo as informações do ataque.

Agente Controlador de Ações (System Controller Agent - SCA): esse agente é responsável pelo controle das ações que o sistema deve tomar em caso de uma tentativa de invasão. Para a tomada de decisão, são utilizadas as bases de dados de estratégia (STBD) e ações remediativas (RABD).

Agente de Integridade do Sistema (Self-Integrity Agent - SIA): esse agente é responsável por garantir a integridade do SDI. O agente de integridade busca

por eventos não esperados ou diferentes do perfil normal dos agentes ativos do sistema.

Agente de Segurança Local (Local Security Intelligent Agent - LSIA): esse agente é responsável pelo gerenciamento da sociedade de agentes e pela interface entre o SDI com o administrador de segurança. É através desse agente que o administrador gerencia o status e a configuração dos agentes, a atualização das bases de dados, o registro das ocorrências detectadas e as ações tomadas pelo sistema.

Agente Administrador de Segurança: é o agente humano que interage com o SDI através do agente de segurança local para realizar diversas tarefas, tais como: configurar a política de segurança do ambiente computacional (base de dados de estratégias) e as demais bases de dados; ativar e desativar agentes; gerenciar o *status* e a configuração do sistema, além de conhecer a situação atual do ambiente monitorado.

Bases de Conhecimento: o sistema dispõe de quatro repositórios para armazenar as informações relevantes à detecção de intrusão. O STBD é a base de dados responsável por registrar as estratégias adotadas por uma organização qualquer em relação à sua política de segurança. Ela é importante para garantir a adaptabilidade do SDI aos mais diversos casos. No RABD estão contidas as informações referentes às ações que devem ser tomadas de acordo com a severidade do ataque detectado. Também varia de acordo com a política de cada instituição. O IIDB guarda as assinaturas de intrusão que serão utilizadas para a detecção de atividades suspeitas. Ele deve ser constantemente atualizado para garantir que novas técnicas de ataque possam ser detectadas. Por fim, tem-se o DFDB, que registra os danos causados por tentativas de ataques e de ataques bem-sucedidos. Nele estão contidas informações que podem ser úteis na identificação de tentativas de ataques provenientes de uma mesma origem ou domínio ou simplesmente para serem utilizadas em investigações futuras.

A seguir é apresentado o diagrama de colaboração do NIDIA (Figura 3.2), em notação UML:

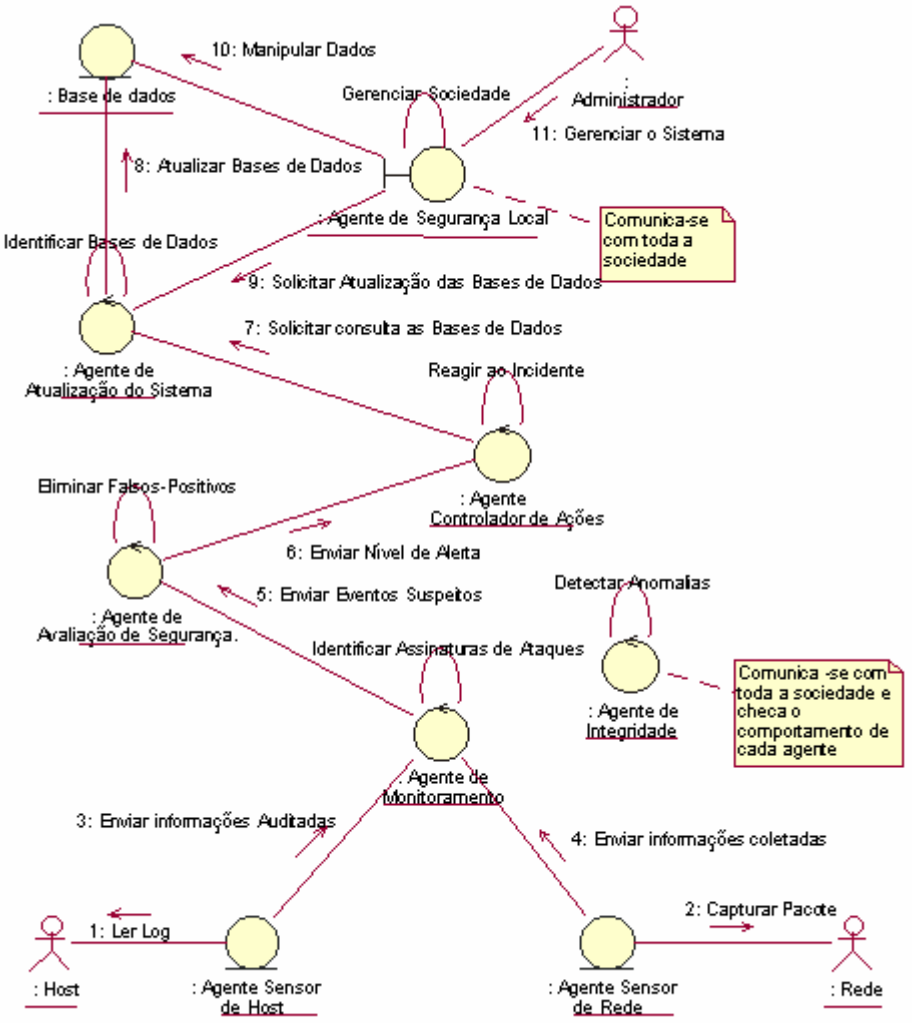


Figura 3.2 Diagrama de Colaboração NIDIA (notação UML)

Segundo o esquema acima, inicialmente os agentes sensores de host e de rede do NIDIA capturam os dados do sistema de log de um computador isolado e do tráfego da rede, respectivamente. As informações relevantes são passadas ao Agente de Monitoramento do Sistema (SMA). Esse agente efetua uma pré-análise dessas informações, verificando se os eventos ocorridos podem conter indícios de ataque, comparando-os a uma base de padrões de intrusão (IIDB). O processamento pode ser feito através de uma rede neural, o que garante uma grande flexibilidade ao processo, pois mesmo que os eventos ocorridos não sejam exatamente iguais aos do IIDB, a tentativa de intrusão pode ser detectada. Uma vez atingido um limite máximo de suspeita, tais informações são repassadas ao Agente de Avaliação de Segurança do Sistema (SAA).

Então, o SAA faz uma análise de alto nível das informações, levando-se em consideração variáveis contidas nas bases de dados DFDB e STDB, verificando se o evento corresponde realmente a um ataque. Essas variáveis podem ser: o horário em que a ação está sendo feita, o domínio ou computador de origem do evento e o usuário que está efetuando a ação. Se o SAA considerar que a segurança do sistema realmente está em perigo, então o Agente Controlador de Ações (SCA) é avisado.

O SCA, de acordo com as informações recebidas, analisa o nível de suspeita do ataque e propõe as medidas cabíveis. Em seguida, o SCA solicita que atualizações sejam feitas, a fim de registrar as informações sobre o ataque no DFDB. Isto é importante para que se tenha informação das origens das tentativas de intrusão e intrusões consumadas, bem como os danos causados e o histórico dos incidentes de segurança. Nesta arquitetura, outros agentes desempenham funções que visam garantir a qualidade, eficiência e segurança do sistema NIDIA.

3.3. Taxonomia de Resposta de Intrusão

Um SRI pode empregar um conjunto de medidas, variando do monitoramento das atividades até uma reação ativa contra o intruso, mas nem todas serão apropriadas. Deste modo, para que uma resposta seja cabível e aplicada no momento certo é necessário que se considere algumas regras para promover a categorização das respostas de acordo com as situações de ataque, resultando em respostas mais coerentes e favorecendo o aumento de chances de sucesso na operação de reação.

Uma taxonomia é um sistema que associa regras para classificação de um evento dentro de uma categoria (Carver e Pooch, 2000). A taxonomia fornece a um sistema de resposta de intrusão um esquema de categorização das possíveis respostas, ofensivas e defensivas, contra uma intrusão. O uso de taxonomia para a classificação de falhas de segurança a tempos vem sido proposto por vários pesquisadores.

Na taxonomia PA (Protection Analysis) (Bisbey e Hollingsworth, 1978) foi feita a análise de 100 falhas de segurança, em diferentes sistemas operacionais, e em seguida organizadas em 4 categorias (erro de sincronização – *improper synchronization*, erro de validação – *improper validation*, erro de proteção – *improper*

protection escolha imprópria de operação – *improper of choice of operation*). Apesar de fornecer uma classificação inicial das falhas, as categorias são muito amplas resultando em grande número de suspeitas. Tais categorias de falhas podem ainda ser subdivididas em múltiplas categorias.

A taxonomia RISOS (*Research in Secured Operating Systems*) (Aslam, 1996; Bishop e Bailey, 1995) classificou as falhas de três sistemas operacionais distintos em sete categorias: validação de parâmetro incompleta *(incomplete parameter validation)*, validação de parâmetro inconsistente *(inconsistent parameter validation)*, compartilhamento implícito de dados privilegiados *(implicit sharing of privileged data)*, validação assíncrona ou serialização inadequada *(asynchronous validation ou inadequate serialization)*, identificação, autenticação ou autorização inadequada (*inadequate identification, authentication, or autorization)*, limite violável *(violable limit)* e erro lógico explorável *(exploitable logic error)*. A principal contribuição deste estudo foi a classificação das falhas de integridade achadas nos sistemas operacionais, no entanto, as categorias ainda são muitos amplas para serem aplicadas em um sistema automático.

A taxonomia proposta por Landwehr (Landwerhr, 1994) dividiu as falhas de segurança em três categorias por gênero da falha, tempo de introdução e localização. O objetivo é identificar como as falhas foram introduzidas, quando foram introduzidas e onde elas podem ser achadas. A taxonomia de gênero da falha é a extensão das taxonomias PA e RISOS com a introdução de uma nova categoria, a de falhas intencionais, que são aquelas introduzidas em um programa intencionalmente e que são exploradas tempo depois. A taxonomia de tempo de introdução identifica quando a falha foi introduzida no sistema e a taxonomia de localização checa onde a falha de segurança ocorreu. Este modelo trouxe uma importante contribuição com a classificação das falhas em múltiplas taxonomias, no entanto, somente a taxonomia de gênero da falha poderia ser aplicada em um sistema automático de resposta, e ainda com algumas limitações. Para que fosse utilizada em um SRI seria necessário determinar qual a intenção do intruso ao gerar a falha; isso é impraticável em tempo real.

Lindqvist e Jonsson (Bishop e Bailey, 1995) apresentaram uma taxonomia que caracteriza segurança de ataque baseado na técnica usada para atacar e os resultados que o ataque provoca. Os principais objetivos são estabelecer

um estudo sistemático dos ataques de computadores, estabelecer uma estrutura para registrar incidentes e fornecer mecanismos para avaliar o grau de um ataque. Nesta taxonomia as técnicas de intrusão foram classificadas em três categorias principais: *bypassing intended controls* (nesta categoria estão incluso programas de ataque explorando autenticação fraca e tentativas de ataque contra *password*, contra *spoof* e contra programas com privilégio), *active attacks of resource* (nesta categoria está incluso ataques ativos semelhantes aos ataques buffer overflow) e *passive misuse of resources* (nesta categoria estão incluso todos os ataque scanning que tentam identificar as falhas do sistema investigado).

A taxonomia Lindqvist intrusion results (Bishop e Bailey, 1995), assim como o modelo da CIA, é baseada na confidencialidade, integridade e disponibilidade e está dividida em três categorias exposure (são ataque contra a confidencialidade dos sistemas), os denial of service (são ataque contra a disponibilidade dos sistema) e erroneous output (são ataques contra a integridade do sistema). Este estudo é importante para um sistema de resposta automático, pois fornece uma boa fundamentação teórica para a classificação das intrusões.

A taxonomia Fish DC&A (Fisch, 1996) é uma taxonomia que classifica resposta de intrusão. As respostas de intrusão são categorizadas de acordo com o tempo em que em que foi detectada (antes ou durante o ataque) e o objetivo da resposta (controle ativo do dano, controle passivo do dano, avaliação do dano ou recuperação do dano). A taxonomia Fish somente considera respostas defensivas e deixa de considerar algumas variáveis (tipo de intruso, tipo de ataque, grau de severidade do ataque, tipo de ambiente atacado) importantes para o processo de resposta automática.

O modelo proposto por Carver (Carver e Pooch, 2000) é composto de seis dimensões. A primeira é tempo do ataque (*timing of the attack*) que considera o tempo de detecção do ataque, se foi antes, durante ou depois da ocorrência do ataque. A segunda dimensão é tipo de ataque *(type of attack*); a resposta deve ser de acordo com o tipo de vulnerabilidade explorada pelo ataque. A terceira dimensão é tipo de atacante (*type of attacker*) , ou seja, as respostas devem considerar o tipo de atacante, se é de *cyber-gang* ou de uma organização militar. A próxima dimensão é implicações do ataque (*attack implications*) que analisa as conseqüências causadas pelo ataque. A quinta dimensão de taxonomia é grau de suspeita

(*Strength of suspicious*). Deve ser medido o grau de suspeita da intrusão que está ocorrendo, pois algumas intrusões são claras, já outras, existem grandes possibilidades de ser um falso positivo. A última dimensão considera o ambiente onde a resposta será lançada (*enviromental constraints*). O ambiente pode ser legal, étnico ou institucional. A seguir é apresentado um melhor detalhamento de cada dimensão:

Tempo de Resposta (*Response Timing):* É importante considerar o tempo em que foi detectado o evento, ou seja, se foi antes da ocorrência do ataque, durante o ataque ou depois do ataque. Respostas preventivas são aplicadas quando há indícios de um ataque, mas que ainda não ocorreu. Durante a ocorrência de um ataque são aplicadas respostas que tentam limitar os seus efeitos e garantir a continuidade dos serviços que estão sendo fornecidos para usuários legítimos. No entanto, quando o ataque já estiver concluído, as respostas visam reparar os danos causados pelo ataque e reunir evidências para punir o atacante.

Tipo de Ataque (Type of Attack): É necessário que haja uma clara diferenciação entre os tipos de ataques para que apropriadas respostas sejam aplicadas. A resposta deve cercar todas as possibilidades do ataque e para isso é preciso uma completa caracterização do ataque que defina quais vulnerabilidades que explora, quais os objetivos e quais os efeitos do atacante.

Tipo de atacante *(Type of Attacker):* A severidade da resposta deve ser medida de acordo com o tipo do atacante, ou seja, o tipo de resposta que se usa contra um atacante amador é diferente do tipo de resposta usado contra um ataque lançado por uma organização militar.

Grau de Suspeita *(Strength of Suspicion):* Algumas atividades detectadas são claramente intrusivas, outras apesar de anormais podem não ser intrusivas. Devido os SDI's gerarem resultado falso positivo a resposta deve ser baseada em um grau de suspeita, que determina o quanto aquele resultado está próximo de ser um ataque. As respostas devem ser baseadas no grau de suspeita da intrusão, deste modo, se o grau for baixo serão aplicadas respostas preventivas, no entanto, se o grau for considerado alto devem ser aplicadas respostas mais amplas e severas.

Implicações do Ataque *(Implications of Attack):* Um ataque tem diferente grau de importância dentro de uma organização. As implicações geradas por um incidente

em uma única estação de trabalho são diferentes das geradas por um incidente em um servidor. Portanto, é importante que alvos diferentes, vítimas do mesmo ataque, tenham respostas distintas.

Limitações do Ambiente *(Environmental Constraints):* Deve ser considerado, também, o ambiente que a resposta será aplicada. Se ele é legal, institucional ou étnico. Por exemplo, existem leis nos USA que proíbe que sejam lançados ataques contra suspeitos.

Percebemos que algumas taxonomias de segurança citadas a cima não são suficientes para serem aplicadas em nosso projeto de respostas automáticas de intrusão. Fisch somente categoriza respostas defensivas e outras simplesmente categorizam falhas de segurança e vulnerabilidades deixando de considerar pontos que podem ser decisivos em um processo de reação. No entanto, a taxonomia de Carver, que foi aplicada no AAIRS (Carver et al., 2000), apresenta um framework das possíveis respostas, ofensivas e defensivas, tornando viável a aplicação de respostas coerentes e precisas em um Sistema de Resposta Automático. Deste modo, no nosso modelo serão consideradas quatro categorias desta taxonomia; tipo de ataque, tempo de resposta, grau de severidade e implicações do ataque abordadas da seguinte forma:

Quais as Vulnerabilidades Exploradas e Suas Conseqüências: Esta informação permite estimar o impacto que o ataque pode causar ao sistema, viabiliza uma boa compreensão do problema, uma investigação automática sem intervenção do administrador e, conseqüentemente, uma resposta mais efetiva. Com esta informação será possível saber quais as conseqüências que um ataque pode provocar ao sistema e como evitá-las.

Tipo de ataque:A informação do tipo de ataque guia o sistema na elaboração da estratégia de resposta. Ou seja, é importante saber qual o objetivo do ataque, que recursos estão sendo usados e se é um ataque externo ou não. A escolha do agente para a reação irá depender desses dados, por exemplo, diante de um ataque externo que explora a vulnerabilidade do protocolo TCP/IP, o agente *Decoy Server* será o responsável pela execução da resposta; mas se for um ataque interno contra as configurações do sistema, o sistema operacional será o mais indicado.

Classificação do Grau de Severidade (normal, alerta, emergencial): Algumas atividades são claramente suspeitas, outras, apesar de apresentarem comportamento suspeito, são normais. Deste modo, a análise do índice de severidade serve para medir o grau de suspeita de um ataque e para certificar-se que não se trata de mais um falso positivo. A resposta está baseada nos níveis de classificação de suspeita (normal, alerta ou emergencial) gerados pelo Agente de Análise de Severidade. Se o sistema estiver em estado de alerta, o evento pode ainda ser um falso positivo ou o ataque estar em fase inicial. Neste caso, serão tomadas medidas menos rigorosas e com caráter preventivo. Se o nível de suspeita for de emergência, já descartada a possibilidade de falso positivo, o SCA aplica estratégias de resposta mais severas e corretivas. Caso o estado do sistema for normal, não acontece nada.

3.4. A Arquitetura Geral do Sistema

O sistema proposto é uma sociedade de agentes inteligentes que fornece respostas preventivas e ofensivas ao sistema NIDIA. O Agente Controlador Principal (MCA) coordena a comunidade de agentes e cada agente é responsável pela execução de um grupo de estratégias de respostas fornecidas pela base de dados de ações e estratégias. De acordo com a estratégia, serão ativados os demais agentes da comunidade e poderão tomar atitudes como quebrar uma conexão, remover ou alterar as permissões de usuários, reconfigurar tabelas de roteadores e firewall, ou simplesmente solicitar ajuda ao administrador em caso de dúvida na tomada de decisão.O objetivo do sistema é garantir que nenhum evento de ataque seja descartado sem solução. Para isso, através das diversas funções assumidas pelos agentes da sociedade, tenta englobar as possíveis formas de reação automática tanto ao nível de usuário como para ataques externos.

A arquitetura do sistema proposto é formada por uma sociedade de agentes artificiais (Agente Controlador Principal, Agente de Avaliação de Severidade, Agente BAM, Agente Sistema Operacional e Agente HoneyNet), apresentados a seguir na Figura 3.3, responsáveis pela execução de respostas preventivas e de emergência aos incidentes detectados. A Figura mostra os diversos níveis de profundidade existentes na arquitetura SCA e sua interação com os repositórios de dados do NIDIA.

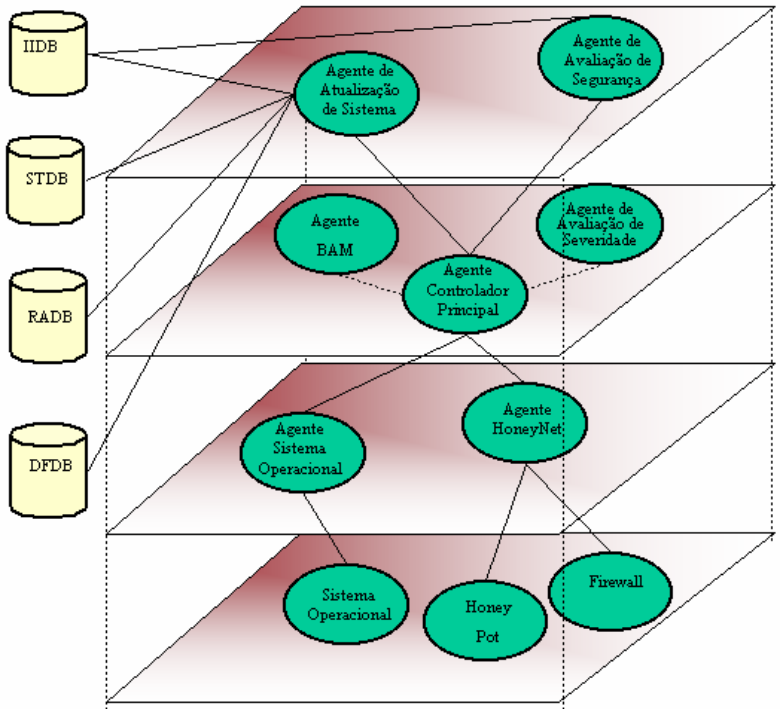


Figura 3.3 - Arquitetura do Sistema Proposto.

A seguir estão relacionados os principais agentes e suas atribuições:

Agente Controlador Principal (MCA): esse agente é responsável pelo controle e monitoramento das atividades dos outros agentes fornecendo as condições necessárias para uma resposta automática. Ele recebe informações (nome do ataque, e nível de suspeita) vindas do agente SAA, necessárias para iniciar uma reação, as envia para os outros agentes solicitando uma estratégia de reação.

Agente BAM (BA):esse agente é responsável pela identificação do tipo de ataque. Ele recebe do MCA um vetor binário de dados identifica o tipo do ataque, através de uma rede neural BAM (Bi-Directional Associative Memory), e retorna o resultado ao MCA.

Agente de Avaliação de Severidade (SEA): a função desse agente é medir o nível de perigo que uma intrusão está oferecendo para o sistema. O SAA, através de uma rede Perceptron Multi-Camadas (MLP) (Dias,2003), gera um grau de severidade, entre 0 e 1, e quanto mais próximo de 1 maior a suspeita de ataque. A partir deste índice, que é enviado pelo MCA, o agente de análise de severidade irá gerar um nível de segurança indicando o estado do sistema (normal, alerta ou emergência).

Agente Honey Net (HNA): o agente Honey Net (Oliveira, 2002) é composto de uma arquitetura, baseada em agentes inteligentes, associada a estratégias de implantação de honeypots[12] e é responsável pela investigação de atos suspeitos de atacantes potenciais e pela configuração dinâmica de firewall e routeadores. Deste modo, sempre que for detectado algum ataque externo que explora vulnerabilidades do protocolo TCP/IP as tabelas de permissão de firewalls ou de routeadores serão dinamicamente modificadas. Esse agente irá também, prover mecanismos para a descoberta dos objetivos dos atacantes, ainda não identificados pelos outros agentes, para que sejam providenciadas medidas de segurança para proteger o ambiente atacado, e ao mesmo tempo, colher informações que levem o atacante a sofrer uma possível punição legal.

A idéia consiste infiltrar estrategicamente na estrutura de uma rede decoy servers (Honeypots) e agentes visando coletar informações que possibilitem a investigação de atividades suspeitas na rede e no sistema (complementando a ação de firewalls, SDI´s e políticas de segurança adotadas). Assim, um Decoy Server seria como um fictício web site corporativo, uma armadilha, para onde os hackers serão desviados. Os agentes, estrategicamente implantados, registram as atividades do intruso e geram aos administradores da rede um registro detalhado do evento. Este registro poderá ser usado para rastrear intrusos, e ou mesmo tempo, como provas substanciais para que intrusos sejam processados em um tribunal.

A intenção não é isolar totalmente o invasor, e sim, restringir seu acesso, deixando certas áreas (honeypots) livres para sua investida, fazendo com que ele perca tempo tentando explorar as vulnerabilidades encontradas nos decoy servers, e ao mesmo tempo, os agentes o monitoram registrando toda a sua atividade e coletando informações necessárias para uma possível reação. Eventualmente, na ocorrência de uma investida criminosa e danosa ao ambiente os elementos ativos do sistema seriam acionados de modo que o administrador da rede possa se precaver desses eventuais ataques isolando totalmente o invasor, negando acesso ou outra medida cabível.

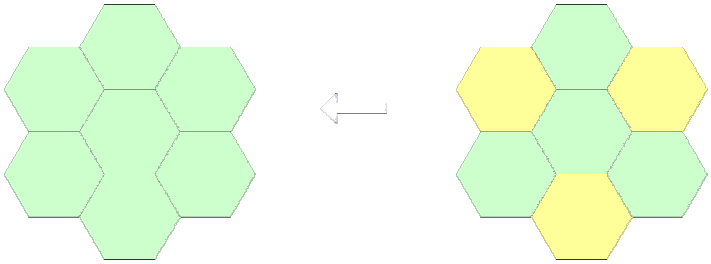


Figura 3.4 Comportamento da Rede Antes e Durante o Ataque

A Figura 3.4 mostra que inicialmente a rede global, pode ser vista tanto pelos usuários com atividades normais como pelos usuários de comportamento suspeitos (áreas verdes). No momento em que o invasor tentar explorar alguma vulnerabilidade, ou até mesmo, durante uma varredura de portas os agentes infiltrados estrategicamente irão reconfigurar os elementos de segurança para que a operação continue transparente para os usuários normais, porém, com um conjunto de restrições de acesso (áreas amarelas) aplicado ao invasor. Deste modo, o invasor continua tentando explorar as vulnerabilidades, mas sem ameaçar o ambiente.

A arquitetura Honey Net Agent é formada de uma coleção de agente com as seguintes responsabilidades:

- A responsabilidade de restringir o acesso em firewalls e roteadores e switches;
- A responsabilidade de monitorar o sistema e fazer análises off-line.
- A responsabilidade de desviar o tráfego não-malicioso para o servidor de produção;
- A responsabilidade de coletar o tráfego malicioso e encaminhar, de modo seguro, ao servidor de logs remoto;
- A responsabilidade de fazer o rastreamento do invasor e descobrir sua origem;

Com isso, esperamos um sistema capaz de executar as seguintes respostas:

- Ao descobrir a origem do ataque obstruir, no roteador, todo o tráfego do atacante. Mas essa solução é provisória e só permite que se tenha mais tempo para tomar outras providências. Um atacante que tem seu endereço IP bloqueado pode mudar o seu IP e atacar novamente.
- Derrubar o Host quando o sistema estiver tomado por um ataque avançado. Apesar de ser uma medida rígida, é a única forma de proteger o host de um ataque ativo sem maiores conseqüências.
- Quando um ataque for lançado via um serviço ou uma porta conhecida para atingir um sistema, parar efetivamente o ataque através do bloqueio dos recursos atingidos (porta e serviços), sem afetar qualquer outro serviço do sistema.
- Investigar a conexão de rede para tentar identificar um atacante e desvia-lo para o HoneyPot e descobrir as suas intenções.

Agente Sistema Operacional (SOA): agente que irá eliminar anomalias no sistema. Através de rotinas de interação com o sistema operacional irá realizar modificações nas configurações do sistema e no conjunto de prioridade dos usuários. A idéia é que sejam implementas rotinas de interação com o sistema operacional para fazer o monitoramento automático das atividades do sistema. Deste modo, sempre que forem detectadas anomalias, variações no comportamento de usuários ou no perfil das atividades do sistema, estas rotinas poderão atuar de forma contínua, evitando medidas bruscas, e fazer alterações no comportamento do sistema. Através da análise de logs, poderá provocar pequenas mudanças e/ou grandes alterações nas configurações do sistema e no conjunto de prioridade de seus usuários na tentativa de mante-lo seguro. Esperamos que esse sistema seja capaz de executar as seguintes medidas:

- Bloqueio do comportamento intrusivo ou conta de usuário comprometida, para evitar que a conta seja usada para lançar futuros ataques.
- Quando não se tem certeza de uma intrusão, é bloqueada a execução de certos tipos de comando sem excluir totalmente o usuário do sistema.

- Devido ao alto grau de falso positivo, deve se evitar medidas severas, então, para confirmar uma suspeita deve-se investigar as atividades do usuário. Isso irá reduzir a possibilidade do intruso suspeito causar dano ao sistema sem que seja excluído.

Bases de Conhecimento no SCA: o sistema proposto utiliza os repositórios de dados apresentados no NIDIA para armazenar e consultar as informações relevantes às repostas de intrusão. O DFDB o SCA utiliza para consultar as informações detalhadas dos ataques (tipo de ataque, implicações do ataque e tipo de vulnerabilidade explorada) e dos incidentes sofridos pelo sistema. Baseado nessas informações que o MCA irá identificar uma estratégia de reação.

O MCA para disparar uma resposta, precisa saber a ação a ser tomada e o agente a ser acionado, então, ao receber dados de um evento ocorrido terá que consultar um banco de dados que contenha estratégias relacionadas de acordo com o tipo de ataque, nível de suspeita, horário de ocorrência e vulnerabilidade explorada. Para isso o MCA irá consulta o RABD e no STBD. A estratégia fornecida pelo RABD poderá ser uma ação ou um conjunto de ações que juntas aplicarão medidas mais seguras e eficazes.

As respostas terão dois tipos de ações, permitir e negar, usadas tanto para conexões e usuários que combinadas oferecerão flexibilidade para suportar grande número de soluções. Note que uma ação de bloqueio pode ter vários significados. Por exemplo, para o Agente Honey Net “bloquear conexão” significa fechar conexão e não permitir mais que seja restabelecida, e isso requer funções de reconfiguração dos recursos da rede e alteração de tabelas do firewall. No entanto, bloquear usuários requer rotinas para interagir com o sistema operacional sem alterar, no entanto, o funcionamento das rotinas do sistema.

3.5. Funcionamento Genérico do Sistema Proposto

Para o melhor entendimento a seguir é explicado, com o auxílio de um diagrama de colaboração em notação UML (Figura 3.5), o funcionamento genérico do modelo de sistema de respostas automáticas proposto.

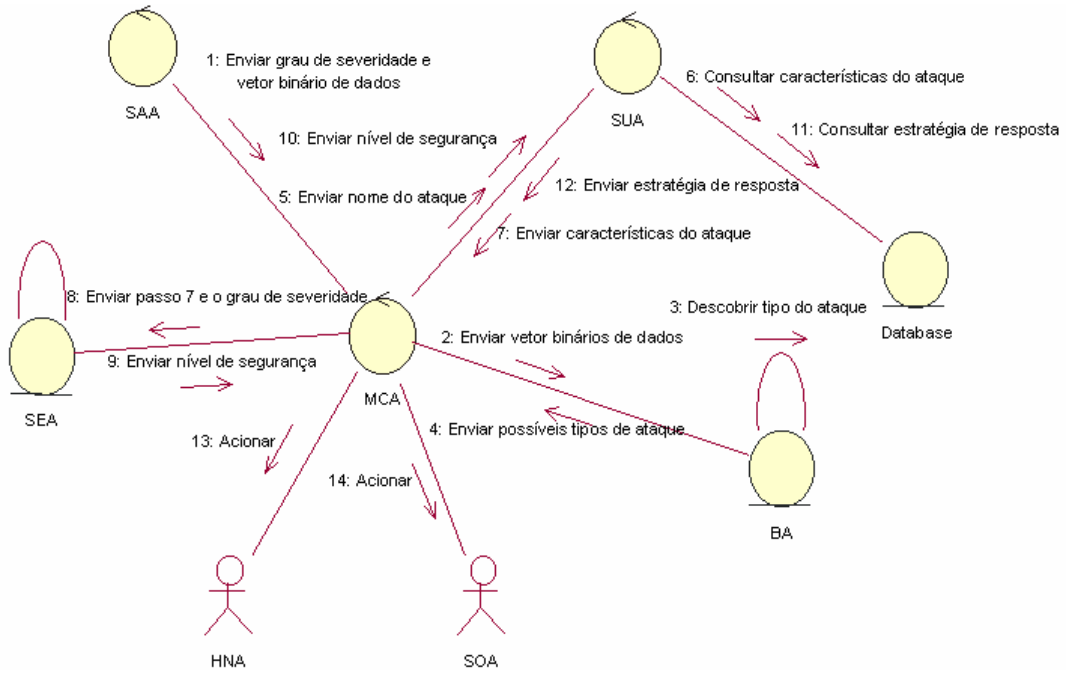


Figura 3.5 - Diagrama de Colaboração do SRI Proposto (notação UML)

Para começar a agir o Agente Controlador de Principal (MCA) vai depender do Agente de Avaliação de Segurança (SAA). Portanto, o SAA ao constatar que o sistema está em perigo envia ao agente MCA informações do incidente, através de um vetor binário de dados. Tais informações irão permitir a descoberta do nome do ataque, e o grau de severidade do padrão detectado. Em seguida, o vetor binário será enviado para o agente BAM que usando uma rede Neural BAM (Bi-Directional Associative Memory) (FREEMAN e SKAPURA, 1991) irá descobrir o nome do ataque e então enviá-lo de volta para o MCA . Após receber essa informação, o MCA a envia para o SUA solicitando mais informações sobre o ataque e sobre o atacante. Após receber as informações do SUA, o MCA as envia para o SEA solicitando um nível de segurança. Se este nível indicar que o estado do sistema está em alerta ou em emergência o MCA irá iniciar uma reação.

Em seguida, o MCA solicita ao SUA, informando o nível de segurança, uma estratégias de reação . O SUA, com base nas informações recebidas, consulta o RABD e o STDB e retorna ao MCA uma estratégia de reação. Na estratégia de reação contém o agente mais indicado para a execução e um conjunto de medidas para combater o ataque. Finalmente, o MCA envia ao agente escolhido a estratégia de resposta e espera o resultado da reação. Os agentes HoneyNet e Sistema

Operacional são os responsáveis pela execução da estratégia de resposta. Para o melhor entendimento do esquema acima segue o diagrama de seqüência do SCA.

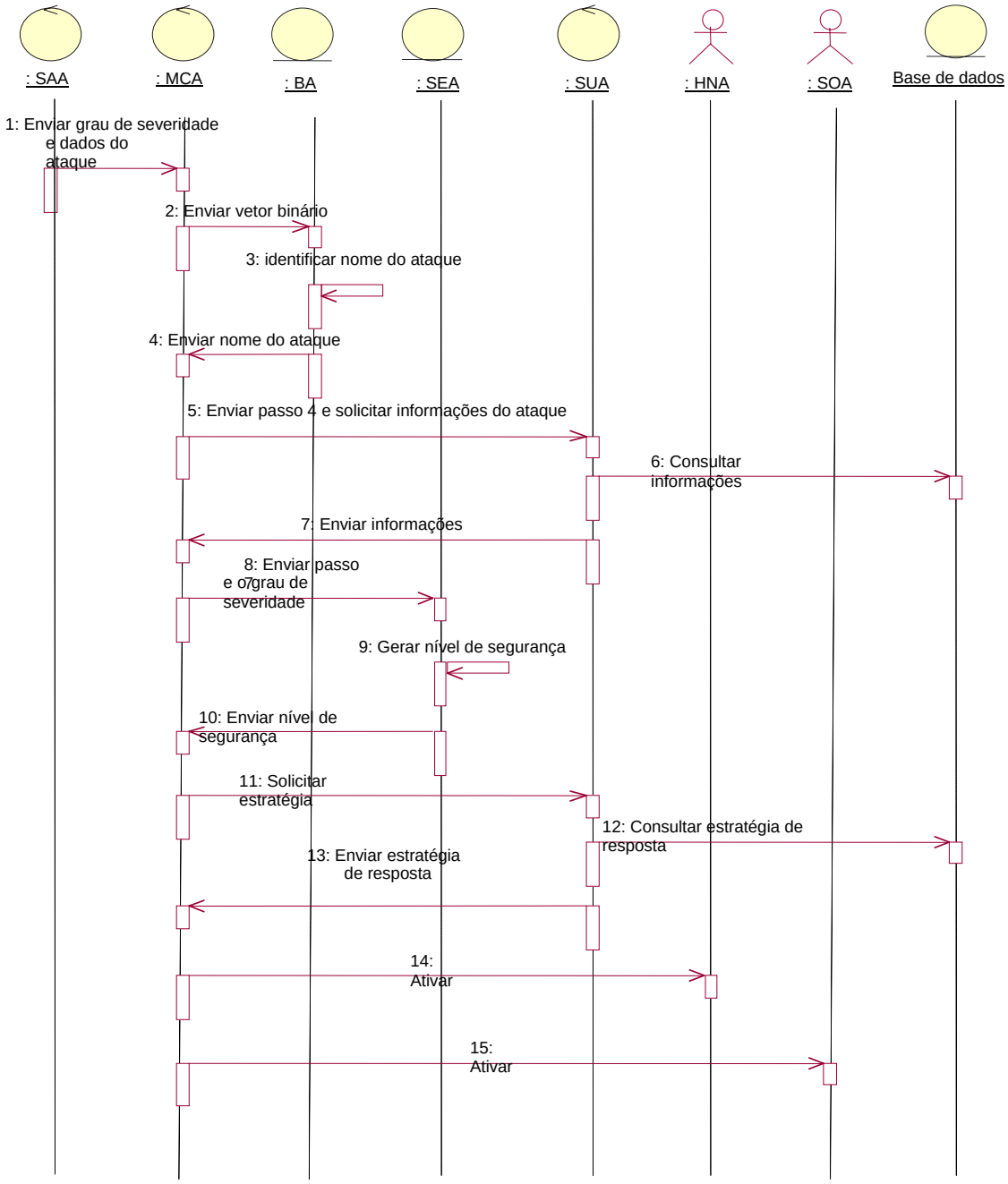


Figura 3.6 Diagrama de Seqüência do SCA

3.6. Considerações Finais

Neste capítulo foi apresentada a proposta de uma proposta para SRI automático, destacando-se os agentes de sua arquitetura e suas principais funções e o funcionamento genérico do sistema e sua interação com o NIDIA. Foi mostrado um estudo comparativo de taxonomias que vieram influenciar na modelam da proposta e na categorização das estratégias de resposta.

Tentamos abordar nesse modelo as principais formas de combater automaticamente uma intrusão. Na sua modelagem nos preocupamos com a viabilidade da implementação da arquitetura, buscando solução alternativas procurou-se superar algumas limitações contidas em determinados sistemas, e assim obter, um sistema seguro, confiável e preciso.

CAPÍTULO 4

IMPLEMENTAÇÃO DO PROTÓTIPO SCA

Neste capítulo, é apresentada a implementação do primeiro protótipo do modelo proposto através da utilização da plataforma ZEUS para a construção da sociedade de agentes. Será mostrada também a implementação de uma rede neural BAM, que é responsável pela identificação do nome do ataque.

4.1. Considerações Iniciais

A implementação do agente BAM, responsáveis pela identificação dos ataques e suas características, é o ponto de partida para a construção do sistema proposto. Para a identificação do ataque implementamos no agente uma rede BAM que irá fazer a associação entre vetores de dados binários. Ou seja, ao receber um vetor binário do MCA irá retorna o nome do ataque através da associação dos vetores de dados (esse assunto será abordado com mais detalhes a seguir). Para a implementação, foi utilizada a linguagem Java 1.3.1 e a ferramenta ZEUS.

Simulou-se parte do agente de Avaliação de Segurança (SAA), para tornar possível a comunicação inter-agentes. Esse agente será responsável por enviar os dados do ataque detectado para o agente MCA que em seguida envia para o agente BAM. Porém, não será dada ênfase à implementação de seu programa externo, em virtude de ser tema de pesquisa em desenvolvimento por outra dissertação (DIAS, 2003). Para efeito de ilustração teremos o agente MCA que mostrará o vetor de dados retornado pelo agente BAM.

O processo de construção da sociedade de agentes é baseado na metodologia de desenvolvimento de agentes ZEUS (COLLINS; NDUMU, 1999c). O processo apresenta-se em uma série de fases de desenvolvimento e consiste em atividades globais e individuais que implementam aspectos particulares de um agente (COLLINS; NDUMU, 1999a). A seguir estarão descritas as fases que iremos adotar para a criação do agente.

i) Criação da Ontologia;

ii) Criação dos Agentes;

iii) Configuração dos Agentes Utilitários;

iv) Configuração do Agente Tarefa;

v) Implementação do Agente.

4.2. Criando a ontologia

Inicialmente foi criado um projeto para a implementação do Agente BAM, associado ao arquivo BAM.def. Em seguida, o arquivo BAM.ont, foi definido como referência para as futuras ontologias definidas.

O próximo passo é a definição da ontologia, ou seja, para que agentes passem a entender um ao outro, eles precisam utilizar um vocabulário de conceitos comuns (a mesma ontologia). Desse modo, ontologia é a representação dos conhecimentos declarativos que representam os conceitos significativos, atributos e valores dentro do domínio da aplicação.

Na tela a seguir (Figura 4.1) são definidas as ontologias e os possíveis atributos (valores e domínios) de cada ontologia.

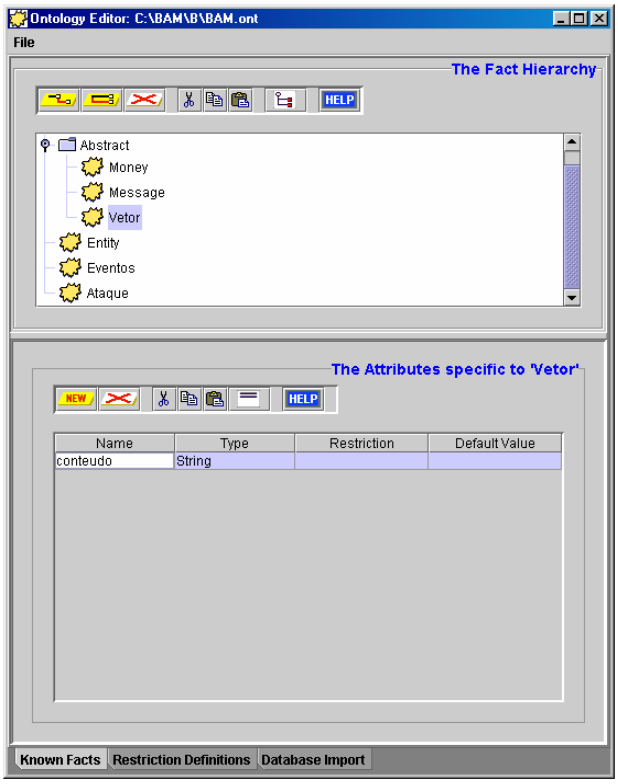


Figura 4.1 Editor de Ontologia

Na definição da ontologia foram criados os fatos, recursos disponíveis ao agente, **Vetor**, **Eventos**, **Ataque**. Tais fatos são relacionados aos agentes pertencentes à sociedade, ou seja, o Agente de Avaliação (SAA) produzirá **Vetor**, o Agente Controlador Principal (MCA) produzirá **Eventos** e o Agente BAM (BA) produzirá **Ataque**. O uso da ontologia será mostrado nas seções seguintes.

À proporção que o sistema for sendo ampliado, a ontologia (Figura 4.1) poderá ser incrementada, de acordo com as necessidades, sem causar impactos indesejáveis no sistema.

4.3. Criando os agentes

Durante essa fase, o agente é descrito de forma lógica (podendo ser reconfigurado posteriormente com as responsabilidades específicas para a aplicação a que se destina), tranformando-se em agente tarefa. Esse processo pode envolver, dependendo da natureza do agente, até quatro sub-fases (COLLINS e NDUMU, 1999c):

- a definição do agente - onde foram especificadas suas tarefas, recursos iniciais e planejamento de habilidades;
- a descrição das tarefas - onde foram especificadas a aplicabilidade e atributos das atividades do agente;
- a organização do agente - onde o contexto social de cada agente é especificado;
- a coordenação do agente - onde cada agente foi equipado com as habilidades sociais para interação.

Os nomes dos agentes da aplicação são inseridos na lista de agentes na seção "Agent Options" e suas tarefas na seção “Task options” (Figura 4.2). Foram criados três agentes tarefas: o **MCA**, o **BA** e o **SAA**. A cada agente foi associada uma tarefa, sendo elas **MostrarSaída**, **GerarSaida** e **EnviarEntrada**, respectivamente.

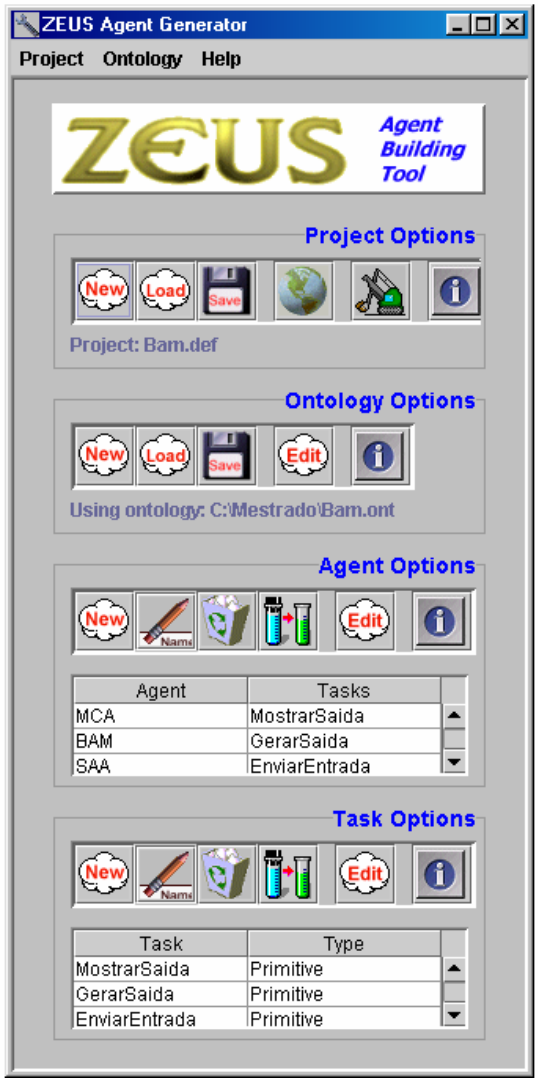


Figura 4.2 Criando os Agentes Tarefa

Em seguida é preciso relacionar as tarefas aos agentes. Isso é feito selecionando-se o agente, e então clicando-se em "Edit" na seção "Agent Options". A tela "Agent Editor" (Figura 4.3), mas especificamente na subseção "Agent Definition Panel", permite que o usuário associe as tarefas a cada agente. A organização entre os agentes, hierarquicamente, pode ser definida na subseção "Agent Organisation", onde a relação entre agentes é definida.

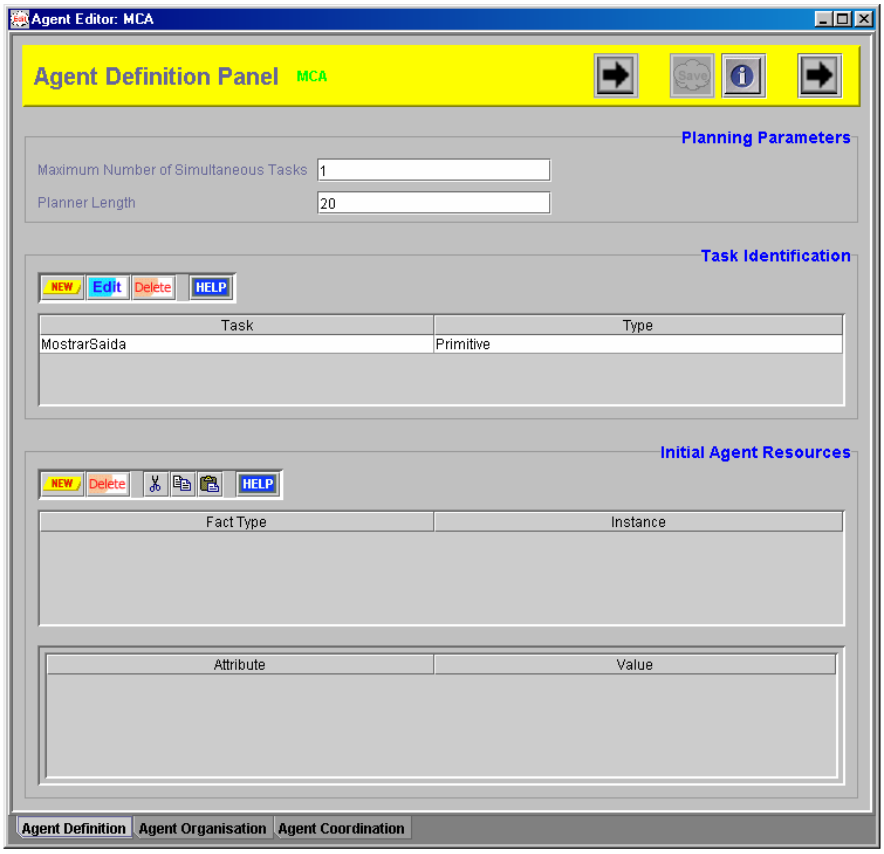


Figura 4.3 Editando os Agentes

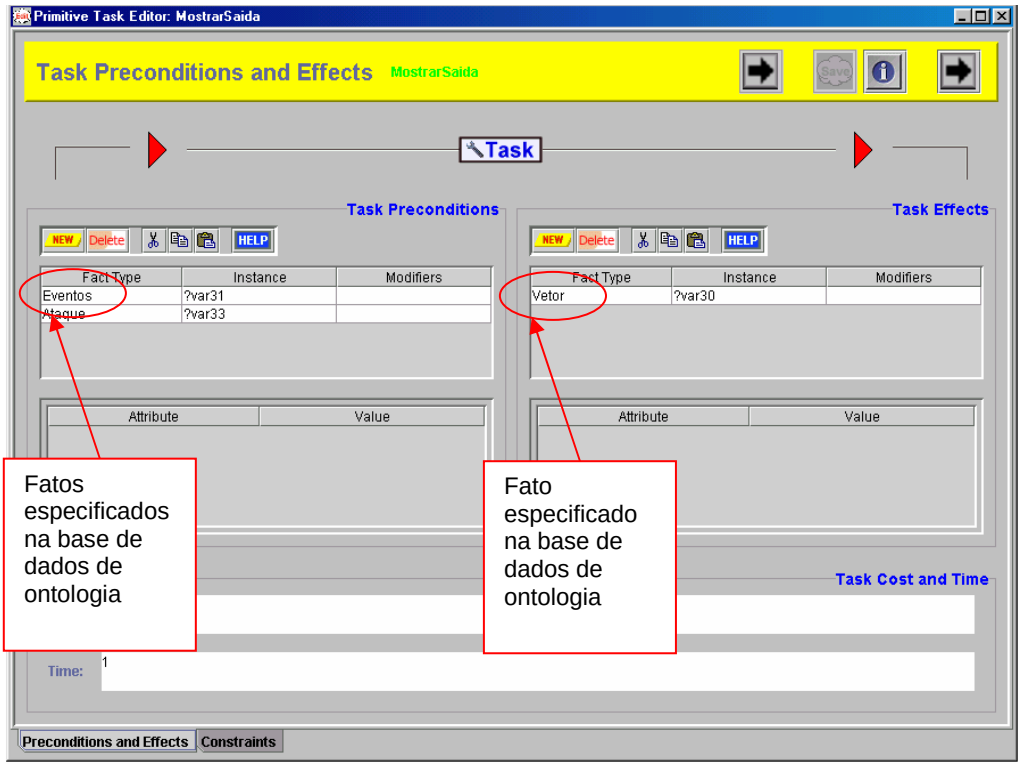


Figura 4.4 Configurando a tarefa do agente MCA

Conforme foi mencionado anteriormente, o SAA tem como saída **Vetor**, o MCA tem como saída **Eventos** e o agente BAM tem como saída **Ataque**. Portanto, na tarefa **MostrarSaida**, adicionou-se no campo *Task Effects* o fato do tipo **Vetor** e no campo *Task Preconditions* adicionou-se o fato **Vetor** e o fato **Ataque.** Ou seja, a pré-condição para que o agente MCA mostre a saída é que se tenha, inicialmente, o vetor de dados e, em seguida, o nome do ataque (Figura 4.4).

Por "default", na arquitetura ZEUS, os agentes não conhecem os nomes e habilidades dos outros agentes da mesma sociedade, desse modo, se um agente precisa do serviço de outro, deverá contactar um serviço de diretório para descobri-lo. Como opção, eles podem ter conhecimento pré-existente de outros agentes, especialmente se eles interagem com freqüência. Há quatro tipos diferentes de relações que podem ser definidos entre agentes:

- superior - possui autoridade maior que outros agentes, podendo emitir ordens, que deverão ser obedecidas;
- subordinado – tem menos autoridade que o superior e tem que obedecer suas ordens;
- *co-worker* – fazem parte da mesma sociedade e será consultado quando qualquer recurso for requerido (antes do relacionamento *peer*);
- *peer* – relacionamento default sem suposição sobre a interação do agente.

Portanto, na etapa de organização do agente, definiu-se o relacionamento de BA e SAA subordinados em relação ao MCA e associou-lhes às suas habilidades, conforme mostrado na Figura 4.5.

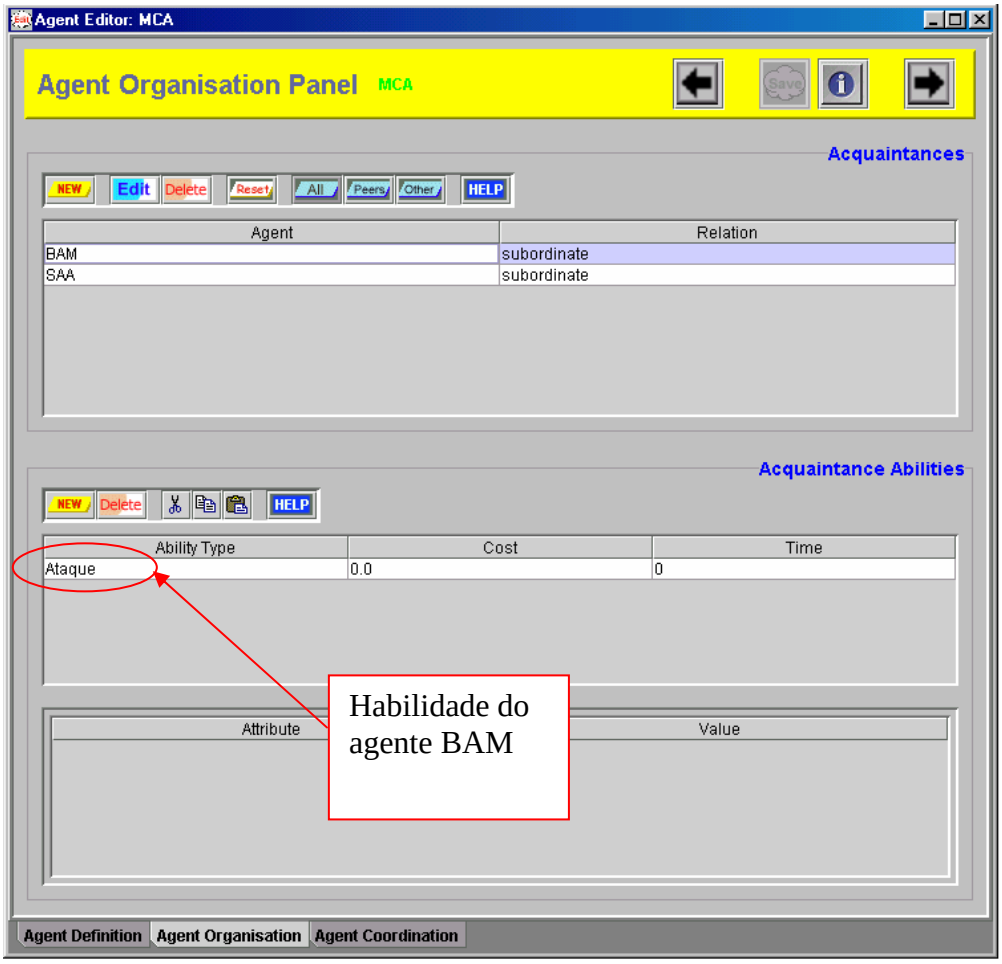


Figura 4.5 Organização dos agentes

Na etapa de coordenação dos agentes, os mesmos foram equipados com o protocolo de coordenação e estratégias, que implementam vários aspectos da conversação tipo rede de contrato (contract-net). A contratação como um mecanismo de coordenação é simbolizado pelo protocolo de rede contratual clássico (Davis e Smith, 1983), onde um agente gerenciador anuncia um contrato, recebe as ofertas de outros agentes interessados e após serem avaliadas, o contrato é entregue ao agente premiado. Esta é a abordagem utilizada pelo agente ZEUS, onde um ou mais *Iniciadores* emitem uma chamada para propostas (*cfp*) (Collins e Ndumu, 1999b), e um ou mais *Participantes* respondem à solicitação.

Dessa forma, definiu-se que MCA e BA seriam *Participantes* e o SAA foi configurado para ser *Iniciador* e *Participante* simultaneamente, conforme mostrado na Figura 4.6.

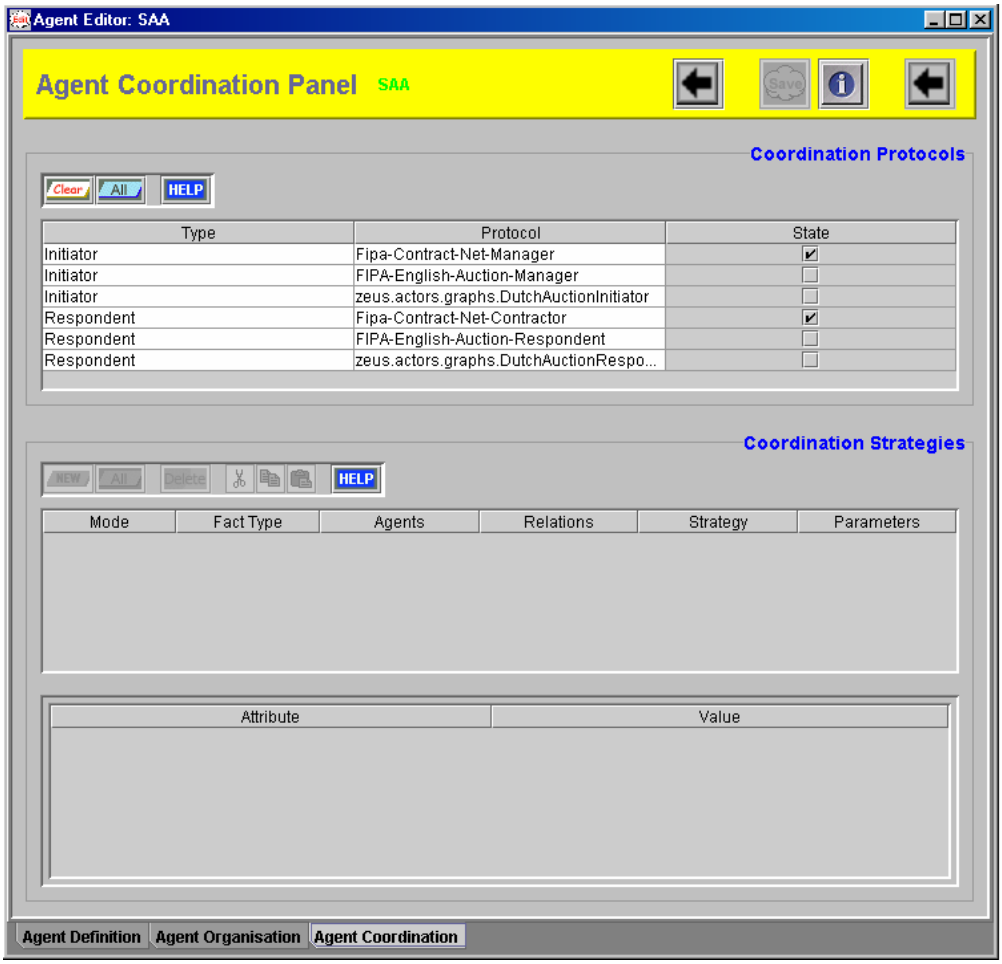


Figura 4.6 Equipando os agentes com o protocolo de coordenação

Após a fase de criação e configuração os agentes estão prontos para trabalhar em sociedade, no entanto, as tarefas específicas de cada agente precisam ser implementadas através de programas externos, caso contrário eles não serão capazes de executar-las

4.4. Configurando os Agentes Utilitários

Os agentes utilitários controlam e fazem a troca de informações entre agentes do ZEUS. Ou seja, quando um agente necessita executar uma tarefa complexa que exige a colaboração de outros agentes utiliza os agentes utilitários (Servidor de Nomes, Facilitador e Visualisador). O Agente Facilitador serve para receber e responder para os outros agentes sobre as habilidades exigidas dos outros, o Agente Servidor de Nomes determina os seus endereços e o Agente Visualisador é usado para analisar ou depurar sociedades de agentes do ZEUS

informando o comportamento coletivo dos agente.Um sistema multiagentes deve ter pelo menos um Agente Servidor de Nomes (ANS), porém não é obrigatório o uso do Facilitador e do Visualisador. Entretanto, em situações onde a aplicação precisa ser monitorada ou analisada o uso do Visualisador se faz necessário .

Agentes Servidores de Nomes possuem somente dois componentes - uma Caixa Postal e um Manipulador de Mensagens - necessários para receber e responder para os agentes as requisições de endereços de outros agentes. Ele mantém um registro dos agentes, habilitando-os a mapear identidades de outros agentes e suas localizações de rede lógica. Isto é necessário porque mesmo que os agentes conheçam os nomes dos demais, desconhecem as suas localizações.

A tela a seguir (Figura 4.7) é utilizada para a configuração dos agentes utilitários. Nela é definido o endereço do host, ou seja, a máquina local onde está a aplicação. O mesmo endereço deve ser mantido para os Agentes Facilitator e Visualiser.

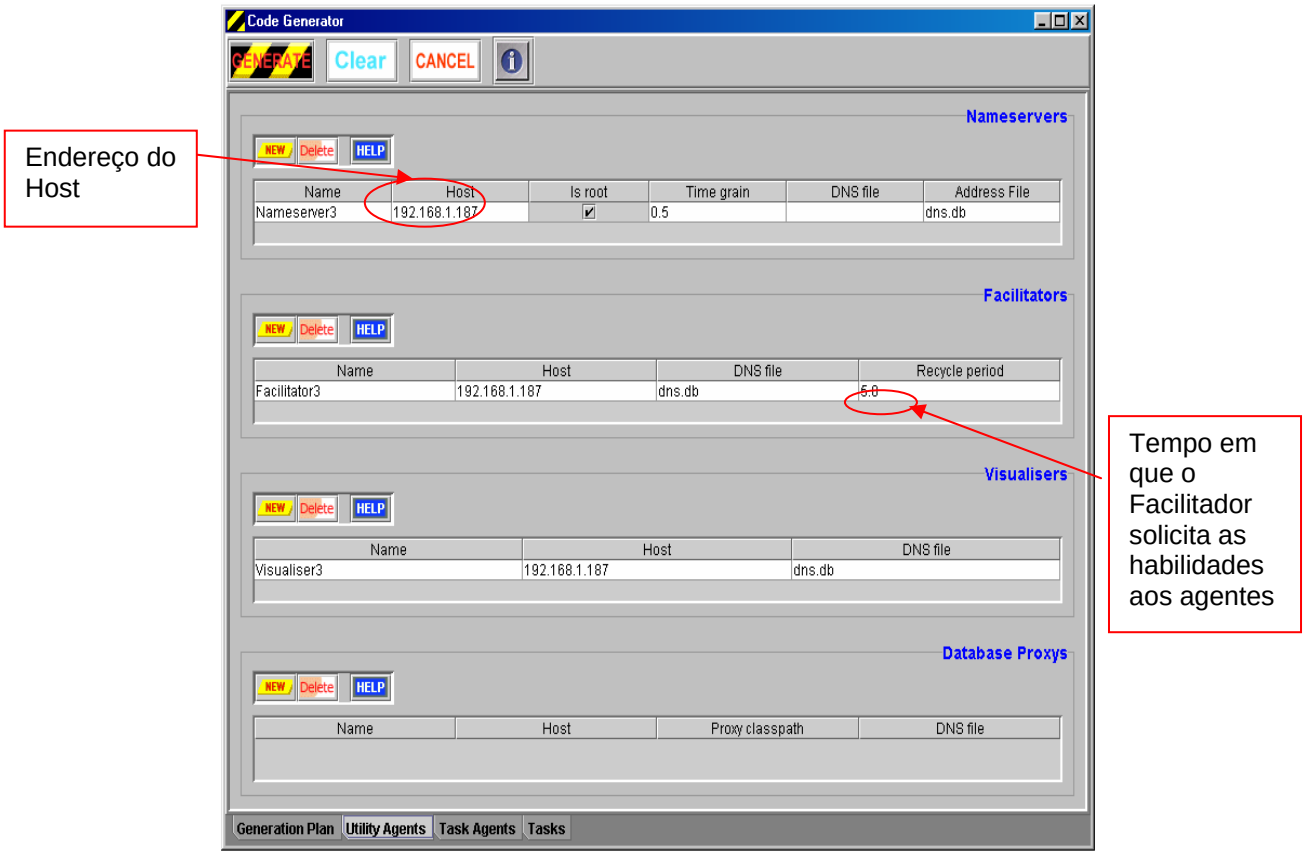


Figura 4.7 Configurando os agentes utilitários

A seguir os parâmetros de execução dos agentes tarefas foram especificados (Figura 4.8). Foi informado em quais máquinas os agentes irão executar (*host*) e quais recursos externos (*Database Extension*) e os programas (*External Program*) que eles irão acionar. Os programas externos serão detalhados nas seções seguintes.

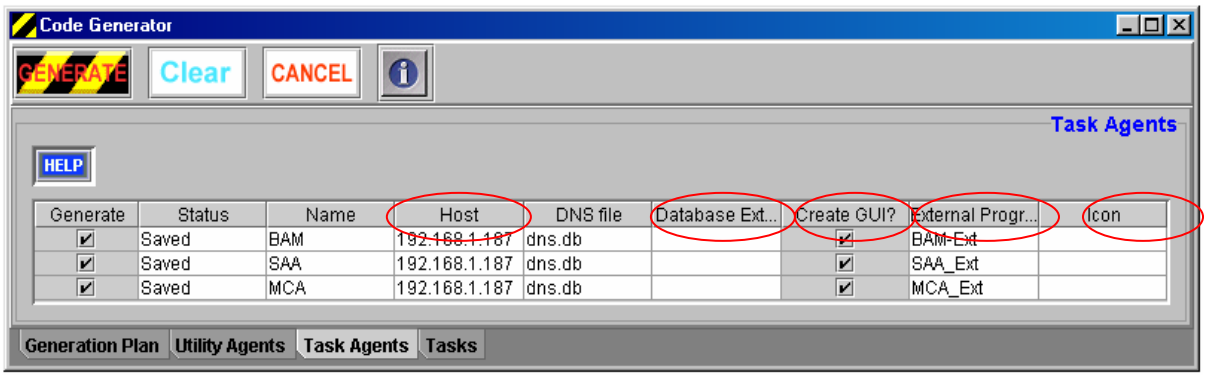


Figura 4.8 Configurando os agentes tarefas

Nesse nível de configuração é permitido optar em criar interfaces gráficas para cada agente (campo *Create GUI?*) e associá-los a um ícone (campo *Icon*). A interface gráfica permite o acesso a todos os componentes do agente, tais como Caixa Postal, Bases de Conhecimento, Máquina de Coordenação etc (Collins e Ndumu, 1999b).

4.5. Gerando os Agentes

Por fim, devemos gerar o código em Java para os agentes definidos. O *Gerador de Código* (Collins e Ndumu, 1999a) pôde ser invocado e o código fonte do agente foi gerado automaticamente (Figura 4.9). Para isso, ainda na tela da seção anterior, devemos apenas ir para a subseção "Generation Plan", e indicar o local no disco onde queremos ter os códigos gerados e o sistema operacional utilizado. Em seguida basta clicar no botão "GENERATE", para que a geração do código seja realizada automaticamente pela ferramenta ZEUS.

Com isso, foram geradas as classes *SAA.java, MCA.java, BAM.java*, *EnviarEntrada.java*, *MostrarSaida.java* e *GerarSaída.java* prontas para serem compiladas.

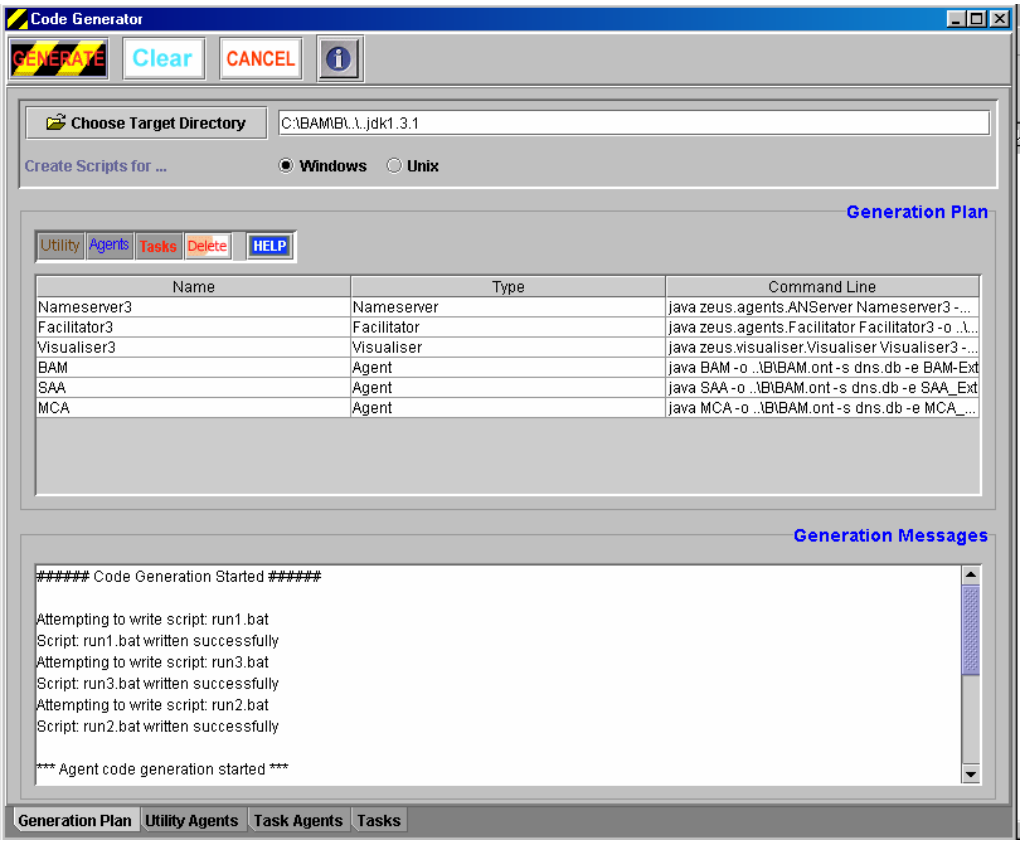


Figura 4.9 Gerando os agentes

4.6. Implementação do Programa Externo do Agente BAM

O agente BAM é fundamental no processo de reação do nosso modelo . Sua função é descobrir o nome do ataque detectado no sistema. Ele irá receber um vetor binário de dados, que caracteriza o ataque e identificar o seu nome. O procedimento se inicia quando o SAA[13] , que tem a habilidade de ativar o agente BAM, solicita ao MCA a identificação de um vetor de dados que representa uma suspeita de ataque. Ou seja, através de uma interface gráfica, o agente de avaliação de segurança ativa o seu programa externo (classe *SAA_Ext*) (Apêndice A) para que o usuário possa fornecer os valores binários a serem utilizados pelo agente MCA. Nessa mesma interface o usuário dá início a todo o processo para a identificação do nome do ataque (Figura 4.10) .

Em seguida, O Agente BAM ativa o seu programa externo (classe BAM_ Ext) (Apêndice A), onde foi implementada uma rede neural de

[13] Este agente é tema de outra dissertação (Dias,2003), por isso, para compor o cenário da comunicação apenas simulamos o SAA.

reconhecimento de padrões BAM (Bi-directional *Associative Memory*) , que fará a associação do vetor de entrada ao seu tipo correspondente de ataque.

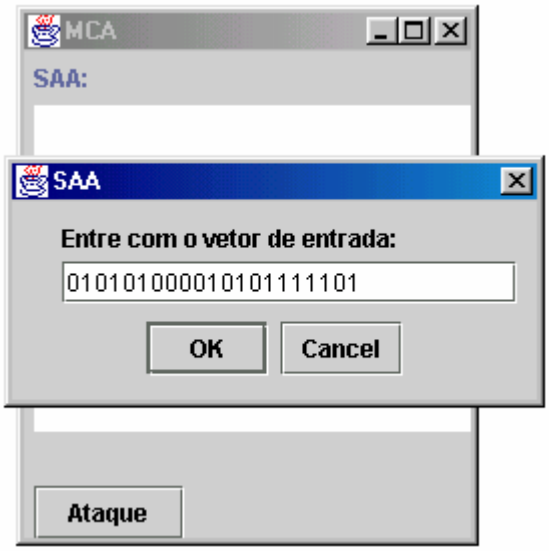


Figura 4.10 Interface gráfica do SAA

O algoritmo da Memória Associativa Bidirecional apresenta em sua estrutura ligações recorrentes com pesos associados. Consiste de duas camadas de elementos processadores, interligadas, utilizadas para estabelecer encadeamentos de associações (Figura 4.11). Na estrutura a seguir está representa a arquitetura de uma BAM, onde toda as n entradas estão ligadas as m saídas através das conexões da matriz peso W.

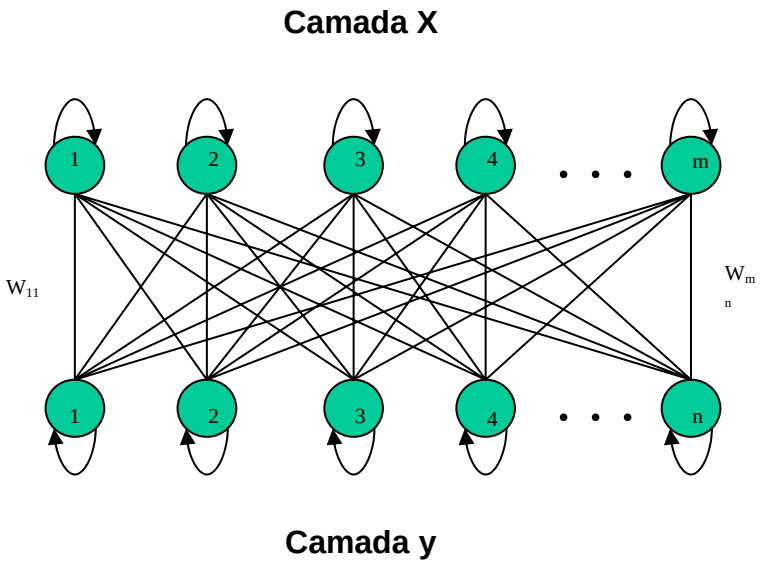


Figura 4.11 Estrutura de uma BAM

Nesse sistema a BAM irá propagar da camada X para a camada Y , ou seja, X será a entrada e Y será a saída. Portanto quando um conjunto de dados for apresentado a camada X, um conjunto de peso entre os nodos na rede são trocados por associação dos componentes, e um conjunto de dados na camada Y será recuperado. O inverso também é realizado, o que justifica o nome bidirecional.

Por exemplo, a Figura 4.12 representa a memória de uma BAM com os nomes de vários tipos de ataque. Supomos que o vetor X=(0,1,1,1,0,1,0,0,0,0) está associado ao vetor Y=(1,1,1,0,1,0,1,0). Ao apresentar o vetor X a BAM, usando memória heteroassociativa[14], ela irá retornar Y.

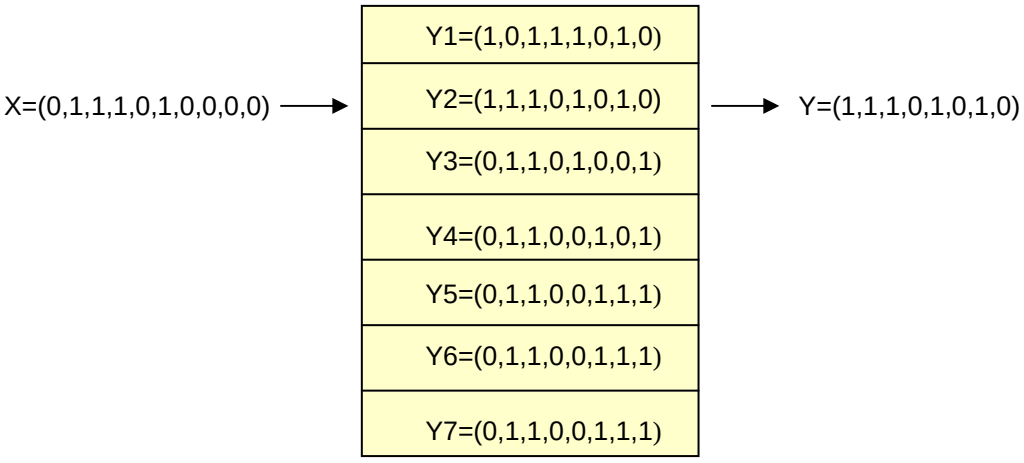


Figura 4.12 Memória da BAM

A seguir, vamos agora detalhar as etapas necessárias para a implementação da BAM. Inicialmente foram obtidos os vetores, contidos nos arquivos entrada.dat e saída.dat (no sistema representado por X e Y respectivamente), para o treinamento da rede (Figura 4.13).

```
/* Lendo Entrada e Saída*/
public void treinamento( ) {
  X = new int [TamTreinamento +1][numEntrada + 1];
  Y = new int [TamTreinamento +1][numSaida + 1];
  lendoArquivoEntrada();
  lendoArquivoSaida();
  inicializaW();
                              }
```

Figura 4.13 Lendo arquivos de Entrada e de Saída da BAM

[14] Existem dois tipos de memória associativa, são elas: autoassociativas e heteroassociativas. Uma memória autoassociativa é usada para recuperar o padrão, anteriormente arquivado, que mais se parece ao padrão que esta sendo apresentado a rede, ou seja, $X_i=Y_j$. No entanto, na memória heteroassociativa o padrão recuperado, padrão de saída, é diferente do padrão de entrada, ou seja, $X_i \neq Y_j$.

Em seguida, é feita a multiplicação dos vetores, os de entrada X com os de saída Y, e assim gerada a matriz peso W (figura 4.13). Ou seja, a matriz w é resultado da multiplicação dos i vetores X de entrada pela transposta dos j vetores Y. A expressão utiliza para o cálculo da matriz peso foi $(w_{ij}) = (x_i)(y_j^t)$ (Freeman e Skapura, 1991),onde i é a quantidade de vetores de entrada e j é a quantidade de vetores de saída.

```
/** Calcular os Pesos.**/
public void inicializaW( ) {
   int soma;
   for (int i = 1; i <= numSaida; i++)
      for(int j = 1;j <= numEntrada; j++) {
         soma = 0;
            for (int t = 1; t <= TamTreinamento; t++)
               soma = X[t][j]*Y[t][i] + soma;
                W[i][j] =  soma;
                                                  }
                                        }
```

Figura 4.14 Calcular a matriz peso W

Depois de obtida a matriz peso W, a BAM já pode ser utilizada para a recuperação de informação. Ou seja, ao receber um vetor de entrada do agente SAA, que irá representar um ataque, a propagação pode ser feita. No entanto, como existem i vetores X é preciso descobrir qual deles mais se assemelha ao vetor de entrada apresentado. Deste modo, é calculado o grau de similaridade entre o vetor de entrada e os vetores X contidos no arquivo entrada.dat.

Para isso, foi feito o cálculo da distância Haminiana (Freeman e Skapura, 1991) entre os vetores (Figura 4.15).

```
/* Função de Hamming Distance */

  public int Hamdistancia(int p) {
           int Ham[];
           int valor, posicao, soma;
           Ham = new int [TamTreinamento + 1];

            for (int t = 1; t <= TamTreinamento; t++) {
            soma = 0;
               for (int i = 1; i <= numEntrada; i++)
                    soma = soma + (X[t][i] - Entrada[p][i])*(X[t][i] - Entrada[p][i]);
             Ham[t] = (int) soma / 4;
             }
```

Figura 4.15 Calcular distância Hamminiana

O vetor que proporcionar menor valor de distância Haminiana do vetor apresentado será utilizado para calcular a Nety. A distância Haminiana pode ser definida pela expressão $H = (x1-y1)^2 + (x2 -y2)^2 + ... +(xn – yn)^2 / 4$.

Depois de descoberta a posição do vetor X mais próximo do vetor de entrada, a rede BAM é processada. Propagando de X para Y , temos Net^y calculado pela soma do produto entre os componentes de X e os pesos (Figura 4.16). Ou seja, $Net^y = \sum w_{ij}x_j$ e finalmente os elementos do vetor de saída são determinados (Figura 4.17) obedecendo as seguintes condições:

$$Y_i(t+1) = \begin{cases} +1 \text{ se } Net^y > 0 \\ Y_i(t) \text{ se } Net^y = 0 \\ -1 \text{ se } Net^y < 0 \end{cases}$$

```
/***
* Função de NETy
*/

public void NETy(int t, int p) {
 int soma;
 for (int i = 1; i <= numSaida; i++){
  soma = 0;
   for(int j = 1; j <= TamEntrada; j++)
    soma = soma + W[i][j]*X[p][j];
    if (soma > 0)
     Saida[t][i] = 1;
    else
      if (soma < 0)
       Saida[t][i] = -1;
     else
     Saida[t][i] = Y[p][i];

}

}
```

Figura 4.16 Calcular Nety

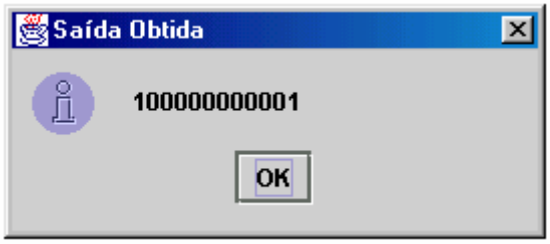


Figura 4.17 Mostrar Vetor de Saída

4.7. Conclusão

Nesse protótipo foi analisada apenas a capacidade da rede neural BAM de fazer associação e recuperar informações relevantes. Os dados vindos do SAA necessitam ser filtrados e formatados para que sejam submetidos ao agente BAM somente dados necessários para a identificação do ataque e em padrão binários. Portanto, os agentes neste trabalho implementados serviram somente para que pudéssemos testar a rede neural, tais agentes necessitam de outras funcionalidades adicionais e são temas de diversos trabalhos de dissertação.

Portanto, foi feita simplesmente uma análise significativa dos vetores gerados pelo protótipo BAM e que, de maneira geral, demonstraram que a rede BAM está apta para a associação e recuperação de informação.

Desse modo, destacamos somente a importância de tais informações, pois são o ponto de partida para a análise e tomada de decisão pelo sistema MCA no caso de uma tentativa de ataque ocorrer.

CAPÍTULO 5

CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Neste capítulo apresenta-se a conclusão sobre os principais pontos alcançados com esta dissertação e suas contribuições. Também, propõe-se futuras pesquisas para respostas automáticas em sistema de detecção de intrusões utilizando-se sistemas multiagentes.

5.1. Contribuições do Trabalho

Com esse trabalho constatamos que resposta de intrusão constitui uma problemática que gera considerável interesse na atualidade, ensejando inúmeras tentativas de implementação de mecanismos de respostas para sistema de detecção de intrusão mais eficientes. É notável que a maior parte dos SRI utilizados possuem capacidade limitada para combater um comportamento intrusivo. Vimos que os meios mais empregados são os sistemas de resposta manual e de notificação, apesar de serem soluções bastante utilizadas, apresentam um atraso entre o momento da detecção e a resposta, possibilitando ataques bem sucedidos contra qualquer ambiente.

Com isso, uma ferramenta que forneça possibilidades de respostas automáticas contra intrusões, e também, que eleve a efetividade das respostas e ofereça melhores possibilidades de defesa se mostra uma solução necessária.

Baseado nas pesquisas existentes na área de Sistema de Resposta de Intrusão (SRI), foi proposta nesta dissertação, como contribuição, uma arquitetura que ofereça respostas automáticas de intrusão, baseada na cooperação de agentes inteligentes, associadas ao uso de uma rede BAM e de um sistema para a análise do estado do ataque, para identificar as características do ataque e o nível de segurança a ele associado, respectivamente, englobando as diversas características desejáveis em sistemas dessa natureza.

Usou-se metodologia uma para a interpretação dos resultados das intrusões (taxonomia) que possibilitasse uma compreensão do problema (intrusão) possibilitando respostas de intrusões mais precisas. Por dispor de um modelo próprio para Sistemas de Respostas Automáticas aplicamos em nossa proposta a taxonomia de Carver (Carver e Pooch, 2000) que fornece um framework para organizar as respostas em categorias possibilitando a modelagem do funcionamento genérico do nosso modelo.

Alem disso, o modelo proposto pode ser facilmente incrementado para novas situações, bem como adaptar-se a novas estratégias, novos mecanismos de resposta, após o reconhecimento de novas assinaturas de intrusão, permite ainda que o administrador tenha conhecimento das situações de ataque sofridas e incrementar novas ações, sem que seja necessário parar o sistema.

Foi parcialmente implementado o agente BAM que possibilita ao sistema saber o tipo de ataque detectado e suas características. Essas informações são cruciais na escolha da estratégia de resposta e do agente que irá executa-la.

Por fim, obteve-se principalmente um modelo de Sistema de Respostas Automáticas, para ser adaptado ao sistema NIDIA, que engloba as principais formas de proteção ao sistema tanto externamente quanto ao nível de usuário. Apresenta estratégias de respostas adaptativas e, por isso, facilidade de integração a novos ambientes a baixo custo.

5.2. Considerações Finais

A construção de uma sociedade de agente para atuar na tomada de decisão fornecendo ações de controle para a correção e/ ou prevenção após uma detecção de intrusão pode fornecer robustez, confiabilidade e flexibilidade a um SDI, promovendo execução contínua, tolerância a falhas, resistência à subversão, sobrecarga mínima e elevado nível de escalabilidade e configuração, permitindo a inserção e remoção do agente de acordo com o cenário de atuação e descoberta de novas tecnologias e conhecimentos de prevenção de sistemas computacionais.

Com a utilização da plataforma ZEUS, os esforços foram concentrados no problema do domínio, mais precisamente, nas tarefas dos agentes, em detrimento da questão relacionada ao desempenho. Foram verificadas limitações quanto ao consumo de memória e velocidade de transmissão das mensagens (atualmente 24

mensagens/seg), pois o sistema necessita de mais de 64 MB[15] de RAM para oferecer um bom desempenho. Isso se deve ao fato dessa ferramenta ter sido desenvolvida com a linguagem de programação Java, e como já se sabe, apesar de fornecer um conjunto de facilidade de desenvolvimento, Java apresenta problemas de desempenho. Portanto, será assunto para estudos futuros uma forma de se obter melhor performance para o SCA.

5.3. Trabalhos Futuros

Como proposta para trabalhos futuros e sugestões para o enriquecimento deste, pode-se listar:

- Implementar o agente de Análise de Severidade para a análise do índice de segurança;
- Construir as bases de dados para arquivar os tipos de ataque e suas características;
- Construir no Agente BAM rotinas para ler o banco de dados de ataque e identificar as características do ataque;
- Desenvolver mecanismos para filtragem das informações vindas do SAA, fazendo com que chegue ao agente BAM somente os dados necessários para a identificação do nome do ataque;
- Modelar e implementar o agente Sistema Operacional. Por vários fatores (instabilidade dos sistemas, inconstância no perfil dos usuários, ambigüidade na definição dos perfis anômalos) é impraticável que se elimine completamente falso positivo dos sistemas de segurança. Neste contexto, nós propomos como forma para se detectar usos indevidos de recursos (memória, CPU, etc), bem como acessos indevidos a aplicações no sistema, a aproximação do SOA ao modelo de sistema imunológico humano. Assim, teremos a definição de um sistema artificial que substituia as ações monolíticas, tudo-ou-nada, por um conjunto de medidas de baixo impacto e contínuas para responder automaticamente comportamento anômalo. Deste modo, evitamos que ataque falso

[15] Foram feitos testes em microcomputadores com até 128 MB e, neste caso, os resultados foram satisfatórios.

positivo seja bruscamente interrompido, e ao mesmo tempo, permitimos que, apesar das flutuações do sistema, um ataque seja efetivamente parado;

- Implementar o agente Honey Net para vigiar o sistema e fazer a configuração dinâmica das tabelas de firewall e routeadores para a remoção de ataques e investigação de atacantes;
- Construir no MCA funções de gerenciamento para que ele controle toda a sociedade de agente. O MCA será o responsável pela coordenação de todas as tarefas executadas no SCA, Irá ativar e desativar agentes, comunicar com o administrador, solicitar atualização dos repositórios de dados e etc.

REFERÊNCIAS

(ABBOTT et al., 1976) ABBOTT, R. P.; CHIN, J. S.; DONNELLEY, J. E. KONIGSFORD, W.; TOKUBO, L. K.; E WEBB, D. A. **Security Analysis and Enhancements of Computer Operation Systems**, (Technical Report NBSIR 76-1041), National Bureau of Standards, Abril1976.

(AGENT WORKING GROUP, 2000) AGENT WORKING GROUP. **Agent Technology**, Object Management Group, v.1, Agosto/2000.

(ASAKA et al., 1999) ASAKA, M.; OKAZAWA, S. E TAGUCHI, A. **A Method of Tracing Intruders by Use of Mobile Agent.** In: Proceedings of the 9th Annual Internetworking Conference (INET`99), San Jose, California, Junho/1999. Disponível em http://www.isoc.org/inet99/proceedings/ 4k/4k_2.htm. Acessado em Setembro 2003.

(ASLAM, 1996) ASLAM T. **Use of a Taxonomy of Security Faults**(Technique Report TR 96-0), COAST Laboratory, Department of Computer Science, Purdue University, 1996.

(AXELSSON, 1999) AXELSSON S. **Research in Intrusion Detection System: A Survey.** Department of Computer Engineering. Chalmers University of Technology, Goteborg, Sweden. Agosto 1999.

(BACE e MELL, 20001) BACE, R.; MELL, P. **NIST Special Publication on Intrusion Detection Systems**. Agosto 2001. Disponível em: http://www.21cfrpart11.com/files/library/government/intrusion_detection_systems_0201_draft.pdf. Acessado em Setembro de 2003

(BACE, 2000) BACE, G. R. Intrusion Detection. Technology Series. Macmillan Technical Publishing. Indianapolis, USA ,2000.

(BARRUS e ROWE, 1998) BARRUS, J.; ROWE, N.C. **A Distributed Automous-Agent Network-Intrusion Detection and Response System**. In: Proceedings of the 1998 Command and Control Research and Technology. Monterrey CA, Junho –Julho,1998.

(BERNADES e MOREIRA, 2000) BERNARDES, M. C.; MOREIRA, E.S. **An Architecture for an Intrusion Detection System Based on Mobile Agents**, International Symposium on Advanced Distributed Systems, Guadalajara, Mexico, 2000.

(BERNADES, 1999) BERNARDES, M.C. **Avaliação do uso de agentes móveis em segurança computacional**. São Carlos: ICMC/USP, 1999. 105p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

(BISBEY e HOLLINGSWORTH, 1978) BISBEY, R.; HOLLINGSWORTH D. **Protection Analysis Project Final Report**, ISI/RR-78-13, DTIC AD A056816, USC/Information Sciences Institute, University of Southern California: Marina Del Rey, CA, Maio 1978.

(BISHOP e BAILEY, 1995) BISHOP M.; BAILEY D. **A Critical Analysis of Vulnerability Taxonomies,** Technical Report CSE-96-11, University of California at Davis, Setembro 1995.

(BOOCH et al., 1999) BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The unified modeling language user guide**. Addison-Wesley, Longman. 1999.

(CANSIAN et al., 1997) CANSIAN, A, M.; CARVALHO, A; BONIFÁCIO, JR., J.M. **Network Intrusion Detection Using Neural Networks**. In: Proceedings of International Conference on Computational Inteligence and Multimedia Applications, ICCIMA'97, Gold Coast, Australia, pp 276-280, Fevereiro1997.

(CANSIAN, 1997) CANSIAN, A.M. **Desenvolvimento de um sistema adaptativo de detecção de intrusos em redes de computadores**. São Carlos: IFSC/USP, 1997. 153p. (Tese de Doutorado). Instituto de Física de São Carlos, Universidade de São Paulo.

(CARVER e POOCH, 2000) CARVER, C. A.; POOCH, U. W. **An Intrusion Response Taxonomy and its Role in Automatic Intrusion Response**; in Proceedings of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, NY, 6-7 de Junho de 2000.

(CARVER et al., 2000)CARVER, JR C. A.; HUMPHRIES, J. W. ; HILL, J. M.D. **Real-Time Intrusion Detection Systems**. Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA, 2000.

(CARVER et al., 2000) CARVER, C. A.; HILL,J. M. D.; SURDU, J. R.; POOCH U. W. **A Methodology for Using Intelligent Agents to provide Automated Intrusion Response**; In: Proceedings of the IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, NY, 6-7 de Junho de 2000.

(CARVER,2000) CARVER C.A. JR. **Intrusion Response Systems: A Survey**; Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA, 2000.

(CERT, 2000a) CERT - COMPUTER EMERGENCY RESPONSE TEAM. **Defending Yourself: The Role of Intrusion Detection Systems**. Disponível em: http://www.cert.org/archive/pdf/IEEE_IDS.pdf. Acessado em Fevereiro 2003.

(CERT, 2000b) CERT COORDINATION CENTER. **CERT/CC Overview Incident and Vulnerability Trends.** Coordination Center Software Engineering Institute, Carnegie Mellon Univesity, Pittsburgh, 2000. Disponível em: http://www.cert.org/present/cert-overview-trends/index.htm

(CERT, 2002) CERT COORDINATION CENTER, **CERT - Computer Emergecy Response Team**. Disponível em: http://www.cert.org/stats/cert_stats.html . Acessado em Fevereiro 2003.

(CHECK POINT SOFTWARE TECHNOLOGIES, 2000) CHECK POINT SOFTWARE TECHNOLOGIES. **Check Point™ RealSecure™ Attack Signatures Glossary**, Version 4.1, Check Point Software Technologies Ltd, Redwood City, California, 2000.

(CISCO SYSTEM, 2002) CISCO SYSTEM. **NetRanger Overview.** Disponível em:http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids1/csidsug /overview.pdf. Acessado em Setembro 2003.

(COHEN, 1999) COHEN, F. **Simulating Cyber Attacks, Defenses, and Consequences.** Fred Cohen & Associates, Março,1999. Disponível em: http://www.all.net/journal/ntb/simulate/simulate.

(COLLINS e NDUMU, 1999a) COLLINS, J.; NDUMU, D. **The Zeus Agent Building Toolkit, Zeus Methodology Documentation**, The Application Realisation Guide, vol 3, Inteligent Systems Research Group, BT Labs, v 1.01, set. 1999.

(COLLINS e NDUMU, 1999b) COLLINS, J.; NDUMU, D. **The Zeus Agent Building Toolkit, Zeus Technical Manual**, Inteligent Systems Research Group, BT Labs, v. 1.0, set. 1999.

(COLLINS e NDUMU, 1999c) COLLINS, J.; NDUMU, D. **The Zeus Agent Building Toolkit, Zeus Realisation Guide**, The Modelling Guide, vol. 2, Inteligent Systems Research Group, BT Labs, v 1.01, set. 1999.

(CROSBIE e PRICE, 2002) CROSBIE, M.; PRICE, K. **Intrusion Detection Systems**, COAST Laboaratory, Purdue University. Disponível em http://www.cerias.purdue.edu/coast/intrusion-detection/ids.html. Acessado em Setembro 2003.

(CROSBIE E SPAFROD, 1994) CROSBIE, M.; SPAFORD, G. **Defending a Computer System using Autonomous Agents**; COAST Labotary, Department of Computer Science, Purdue University. 11 de Março de 1994.

(DEBAR et al., 1999) DEBAR, H.; DACIER, M.; WESPI, A.; **Towards a taxonomy of intrusion-detection systems.** Elsevier Science B.V., 1999

(DEITEL E DEITEL, 2001) DEITEL H.M.; DEITEL, P.J. **Java, como programar**. Porto Alegre: Bookman, 2001.

(DIAS, 2003) DIAS, R. A. **Um Modelo de Atualização Automático do Mecanismo de Detecção de Ataques de Redes para Sistemas de Detecção de Intrusão.** Dissertação de Mestrado Submetida à Coordenação do Curso De Pós-Graduação em Engenharia de Eletricidade da UFMA, Novembro 2003.

(FIRTH et al., 1997) FIRTH, R.; FORD, G.; FRASER, B.; KOCHMAR, J.; KONDA, S.; RICHAEL, J. E.; SIMMEL, D. **Networked Systems Survivability Program; Detecting Signs of Intrusion**. Carnegie Mellon University Pittsburgh, Pennsylvania. Agosto1997.

(FISCH, 1996) FISCH, E. A. **Intrusion Damage Control and Assessment: A Taxonomy and Implementation of Automated Responses to Intrusive Behavior**; (Dissertação de Doutorado), Texas A&M University, College Station, TX, 1996.

(FREEMAN e SKAPURA,1991) FREEMAN, J. A.; SKAPURA, D. M.; **Neural Networks, Algorithms, Applications and Programming Techniques;** Addison-Wesley, Março 1991.

(FRINCKE et al., 1998) FRINCKE, D.; TOBIN, Don; MCCONNELL, J.; MARCONI, J.; POLLA, D. **A Framework for Cooperative Intrusion Detection**. Proceedings of the 21 st National Information Systems Security Conference, pp. 361-373, October 1998. Disponível em: <http://csrc.nist.gov/nissc/1998/papers.html>. Acesso em 60 jul. 2000.

(HEBERLEIN L.T et al., 1991A) HEBERLEIN L.T K.N. LEVITT B. MUKHERJEE, **Towards Detecting Intrusions in a Networked Environment**. In: Proceeding of the 14th DOE Conference on Computer Security. Concord, CA, pp 17-47. Maio 1991.

(HEBERLEIN L.T et al., 1991B) HEBERLEIN L.T, K.N. LEVITT B. MUKHERJEE. **A Method to Detect Intrusive Activity in a Networked Environment**. In: Proceeding of the 14th National Computer Security Conference. Washington, DC, pp 362-371,Outubro 1991.

(IBM GLOBAL SERVICE, 2000) IBM GLOBAL SERVICE, **Denial-of-Service Attack: Understanding network vulnerabilities**. Somers , NY, USA, 2000.

(ISS TECHNICAL WHITE PAPER, 2001) ISS TECHNICAL WHITE PAPER, **The Evolution of Intrusion Detection Technology**. Atlanta. 29 de Agosto de 2001.

(JANSEN, et al., 1999) JANSEN W. et al. **Applying Mobile Agents to Intrusion Detection and Response.** National Institute of Standards and Technology Computer Security Division. NIST Interim Report (IR) – 6416, USA, Outubro/1999.

(KREMER, 1999) KREMER, H.S. **Real-Time Intrusin Detection for Windows NT Based on Navy IT –21 Audit Policy.** Naval Postgraduate School Monterey, California , Setembro /1999.

(LANDWERHR, 1994) LANDWEHR C. E. et al. **A Taxonomy of Computer Program Security Flaws**, ACM Computing Surveys, vol. 26 (3), pp.211-254 1994.

(LARSEN, 2002) LARSEN J. AND HAILE J*. **Understanding IDS Active Response Mechanisms.** Security Focus On- line. Janeiro/02. Disponível em:http://online.securityfocus.com/infocus/ 1540.

(LIMA, 2001) LIMA C.F.L.**Agentes Inteligentes para Detecção de Intrusos em Redes de Computadores.** Dissertação de Mestrado Submetida à Coordenação do Curso De Pós-Graduação em Engenharia de Eletricidade da UFMA, Maio 2001.

(LINDQVIST E JONSSON, 1997) LINDQVIST U. AND JONSSON E., **How to Systematically Classify Computer Security Intrusions**, In: Proceeding of the IEEE Symposio on Security and Privacy, Oakland, CA, pp. 154 – 163, 4-7 de Maio/1997.

(LUNT, 1990) LUNT T.F et al. **A Real Time Intrusion Detection Expert System (IDES).** Interim Progress Report, Project 6784, SRI International, Maio/1990.

(MÓDULO SECURITY SOLUTION, 2001) MÓDULO SECURITY SOLUTION. **7ª Pesquisa Nacional sobre Segurança da Informação**. Disponível em: http://www.modulo.com.br. Acesso em 2 de Agosto de 2001.

(NASCIMENTO et al.) NASCIMENTO, E.; Lima, C.F.L.; COCHRANE, E.M.; COCHRANE, J. **The NIDIA Project Network Intrusion Detection System based on Intelligent Agents**. In: Proceeding of the Tenth Latin-Ibero-American Congress on Operations Research and Systems, p. 212-217, Cidade do México, 4-8 de Setembro/2000.

(NEUMANN E PARKER, 1989) NEUMANN P. G. AND PARKER D. B. **A Summary of Computer Misuse Techniques**. In: Proceeding of the 13th National Computer Security Conference, Baltimore, MD,10-13 Outubro/1989, pp. 396-407.

(NEUMANN e PORRAS, 1999) NEUMANN, P. G.; PORRAS, P. A. **Experience with EMERALD to DATE.** Computer Science Laboratory**,** SRI International, Menlo Park CA 94025-3493,1st USENIX Workshop on Intrusion Detection and Network Monitoring Santa Clara, California, 11-12 Abril,1999, pp 73—80.

(NWANA e WOOLDRIDGE) NWANA,H.S.; WOOLDRIDGE, M. **Software Agent Technologies. in Nwana, H.S. & Azarmi, N. (eds.) Software Agents and Soft Computing: Concepts and Applications**, Lecture Notes in Artificial Intelligence Series 1198, Berlin: Springer-Verlag, 59-78.

(OLIVEIRA, 2002) OLIVEIRA, A. A. P.; **Uma Arquitetura Baseada em Agentes Inteligentes Para Investigação de Atos Suspeitos em Sistemas de Computadores**; 2º Relatório de Acompanhamento; Coordenação de Pós-Graduação em Engenharia Elétrica. Universidade Federal do Maranhão. São Luis – Ma, Setembro/2002.

(PORRAS e NEUMANN, 1997) PORRAS, P. A.; NEUMANN, P. G. **EMERALD: Event Monitoring EnablingResponses to Anomalous Live Disturbances**. In: Proceeding of the 20th National Information Systems Security Conference, Baltimore, MD, 7-10 de Outubro de 1997. Disponível em: http://www.sdl.sri.com/projects/emerald/index.html.

(PRESSMAN,1995) PRESSMAN, R. S. **Engenharia de Software.** São Paulo: Makron Books, 1995.

(PRNEWSWIRE ASSOCIATION, 1999) PRNEWSWIRE ASSOCIATION, Inc. **Plugging the Holes in e-Commerce Leads to 135% Growth in the Intrusion Detection and Vulnerability Assessment Software Market**. PRNewswire. Agosto ,1999.

(RAGSDALE et al., 2000)RAGSDALE, D.J.; CARVER, JR C. A; HUMPHRIES, J. W; POOCH, U. W. **Adaptation Tchniques for Intrusion Detection aond Intrusion Response System**. In: Proceeding Of 2000 IEEE International Conference On Systems, Man, And Cybernetics (SMC 2000), Nashville, 8-11 Outubro, 2000 P.2344-2349.

(RANUM, 1992) RANUM, M. J. **An Internet Firewall**, In: Proceedings of World Conference on Systems Management and Security, 1992. Disponível em: ftp://decuac.dec.com/pub/docs/firewallfirewall.ps. Acessado em 12 Junho 2003.

(REAMI,1998) REAMI, E.R. **Especificação e prototipagem de um ambiente de gerenciamento de segurança apoiado por agentes móveis.** São Carlos: ICMC/USP, 1998. 82p. (Dissertação de Mestrado). Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo.

(SANTOS et al., 2003); SANTOS, G. L. F.; ABDELOUAHAB, Z; NASCIMENTO, E; DIAS, R.A.; LIMA, C.F.L; .COCHRANE, E.M.; COCHRANE, J. **An Automated Response Approach for Intrusion Detection Security Enhancement.** In: Proceeding of the Software Engineering Application Conference 2003 (SEA), Marina Del Rey, CA, Novembro 2003

(SCHNACKENBERG ET AL., 2000) Schnackenberg, D.;Djahandari, K.; Sterne, D.; **Infraestructure for Intrusion Detection and Response.** In Proceeding of the DARPA Information Survivability Conference and Exposition (DISCEX) 2000. 25-27 de Janeiro de 2000.

(SCHNACKENBERG, 2002) SCHNACKENBERG, D.;DJAHANDARI, K.; STERNE, D. **Adaptative Network Defense,** Network Associates Lab. Disponível em: http://www.nai.com/research/nailabs/adaptive-network/aitr.asp. Acessado em 10 de Setembro de 2003.

(SOBIREY, 2003) SOBIREY, M. **The Intrusion Detection System AID,** Disponível em http://www-rnks.informatik.tu-cottbus.de/~sobirey/aid.e.html. Acessado em Fevereiro/ 2003.

(SOMAYAJI e FORREST, 2000) SOMAYAJI, A.; FORREST, S. **Automated Response Using System-Call Delays**. In: Proceedings of the 9th USENIX Security Symposium,. Agosto/ 2000.

(SOMAYAJI et al., 1998) SOMAYAJI, A.;HOFMERYR, S.; FORREST, S. **Principles of a Computer Immune System.** New Security Paradigms Workshop, Association for Computing Machinery, Nova York, 1998..

(STERNE et al., 2001) STERNE, D.; DJAHANDARI, K.; WILSON, B.; BABSON, B.; SCHNACKENBERG, D.; HOLLIDAY, H.; REID, T. **Autonomic Response to Distributed Denial of Service Attacks**. RA ID 2001, LNCS 2212, pp. 134–149, 2001.

(TANACHAIWIWAT et. al., 2003) TANACHAIWIWAT, S.; HWANG, K. **Adaptive Intrusion Response to Minimize Risk over Multiple Network Attacks.** University of Southern California. Disponível: ceng.usc.edu/~kaihwang/papers/ACM827.pdf. Acessado em Fevereiro/2003.

(THOMAS e TIMOTHY, 1998) THOMAS, H. P; TIMOTHY, N. N. **Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection.** Secure Networks, Inc. Janeiro, 1998.

(TOTH e KRUEGEL) TOTH, T.; KRUEGEL, C. **Evaluating the Impact of Automated Intrusion Response Mechanism.** In: Proceedings of the 18th Annual Computer Security Application Conference. Dezembro, 2002.

(WHITE et al., 1996) WHITE, G. B.; FISCH, E. A.; POOCH, U. W. **Cooperating Security Managers: A Peer-based Intrusion Detection System**. IEEE Network, vol. 10 (1), 1996, pp. 20-23.

(WINKLER, 1990) WINKLER, J.R; PAGE, W.J. **Intrusion and Anomaly Detection in Trusted Systems**. In: Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, pp.115-124. Dezembro /1990.

(ZAMBONI, 1998)ZAMBONI, D.; BALASUBRAMANIYAN, J.; GARCIA, F.; JOSE, O; SPAFFORD, E. H. **An Architecture for Intrusion Detection using Autonomous Agents.** Departamento de Ciência da Computação, Purdue University; Coast, 1998 (Technical Report - TR 98-05). Disponível em: http://www.cerias.purdue.edu/homes/aafid . Acessado em Junho 2003.