Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências da Computação e Estatística

Projeto de Iniciação Científica

"Especificação e Implementação de Agentes Móveis em um Sistema de Detecção de Intrusão"

Francisco Gomes Milagres
(milagres@icmc.sc.usp.br**)**

Orientador:

Prof. Dr. Edson dos Santos Moreira
(edson@icmc.sc.usp.br**)**

USP - São Carlos
Maio de 2000

1. Introdução

A crescente necessidade de ambientes seguros para troca de informação em rede vêm cada vez mais exigindo soluções robustas contra as tentativas de invasão de sistemas e fraudes nas organizações que usam recursos distribuídos. Paralelamente, a obtenção, manutenção e disseminação da informação de forma segura tornam-se uma das grandes preocupações da sociedade em geral, ocasionando uma crescente expansão das formas de armazenamento e de distribuição através de redes de computadores.

Tentativas de ataque e invasões, bem sucedidas ou não, estão se tornando freqüentes, envolvendo cada vez mais um número maior de computadores. Dessa forma, segurança distribuída torna-se uma das palavras de ordem para a maioria absoluta das empresas, universidades e centros de pesquisa em todo o mundo. Com a utilização cada vez maior da tecnologia Internet em ambiente corporativo (as *Intranets*) e sua abertura para o mundo externo (*Extranets*) para atividades essenciais, a preocupação e o risco de invasão dos sistemas empresariais cresceram muito nos últimos anos. E não haveria de se esperar algo diferente: os crimes digitais são, na maioria das vezes, muito difíceis de serem descobertos e até mesmo rastreados. Prova disso, é que muitas empresas sofrem invasões em seus sistemas e só se dão conta do fato muito tempo depois, isso quando descobrem.

Mas o panorama está mudando rapidamente. A cada dia surgem soluções mais robustas de proteção dos dados. *Firewalls*, criptografia forte, certificados digitais, VPNs (*Virtual Private Networks*), *smart cards* e biometria (reconhecimento de alguma parte do corpo, como íris dos olhos ou a palma da mão) já fazem parte do arsenal utilizado no combate à violação de sistemas e credibilidade das transações *on-line*. O avanço dessas tecnologias deve-se principalmente ao grande incentivo às pesquisas relacionadas à área de segurança computacional nas grandes universidades e centros de pesquisas mundiais. Entretanto, ainda há uma grande demanda por pesquisas e evolução tecnológica nessa área.

A tecnologia de segurança de rede mais utilizada hoje em dia é o *firewall*. Esse sistema previne a entrada não autorizada utilizando-se de mecanismos de controle de acesso externo. Porém não existe nenhum sistema que possa ser considerado perfeito em matéria

de proteção e ainda, que forneça um elevado grau de segurança enquanto permite uma certa flexibilidade e liberdade no uso dos recursos computacionais.

Existem fatores que tornam muito difícil impedir que atacantes eventualmente tenham acesso a um sistema. A maioria dos computadores possui algum tipo de "furo de segurança" que permite a atacantes externos (ou ainda usuários internos legítimos) terem acesso a informações confidenciais. Mesmo um sistema supostamente seguro pode ser vulnerável a usuários internos abusando de seus privilégios ou ser comprometido por práticas impróprias. Em vista disso, uma vez que um ataque pode ser considerado inevitável, existe uma óbvia necessidade por mecanismos que possam detectar atacantes tentando penetrar no sistema ou usuários legítimos fazendo mau uso de seus privilégios.

Com o crescente aumento no número de ataques internos, a utilização de mecanismos como o *firewall* deve ser ampliada. Visto que este tipo de ataque, ocasionado pelos próprios usuários do sistema, não permite a localização imediata, torna-se necessário o uso integrado de diversas tecnologias para aumentar a capacidade de defesa de um *site*. Entre estas tecnologias, torna-se interessante a presença de mecanismos que acrescentem características de mobilidade no processo de monitoria do sistema. Desta forma, a introdução de agentes móveis em apoio à segurança computacional apresenta-se como uma solução natural, uma vez que permitirá a distribuição de tarefas de monitoria do sistema e a agilização no processo de tomada de decisão no caso de ausência do administrador.

Este projeto apresenta-se como uma possibilidade de expansão das pesquisas desenvolvidas pelo grupo de segurança computacional do laboratório Intermídia do ICMC-USP. O projeto visa a modelagem, implementação e validação de um cenário de execução necessário para a constituição de um SDI: a identificação de usuários anômalos.

O projeto está organizado da seguinte forma: o capítulo 2 apresenta a revisão literária dos conceitos iniciais necessários para o embasamento da proposta; o capítulo 3 apresenta a arquitetura proposta por [BERNARDES, 1999] e seu relacionamento com este projeto; o capítulo 4 apresenta a metodologia de desenvolvimento do projeto, seu cronograma e recursos financeiros associados; o capítulo 5 apresenta as referências iniciais que sedimentam essa proposta.

2. Segurança Computacional e Agentes Móveis

2.1. Considerações Iniciais

O termo segurança é usado com o significado de minimizar a vulnerabilidade de bens (qualquer coisa de valor) e recursos. Vulnerabilidade é qualquer fraqueza que pode ser explorada para se violar um sistema ou as informações que ele contém [ISO, 1989].

A segurança está relacionada à necessidade de proteção contra o acesso ou manipulação, intencional ou não, de informações confidenciais por elementos não autorizados, e a utilização não autorizada do computador ou de seus dispositivos periféricos. A necessidade de proteção deve ser definida em termos das possíveis ameaças e riscos e dos objetivos de uma organização, formalizada nos termos de uma política de segurança [SOARES, 1995]. Zorkle e Levitt ainda acrescentam que a segurança depende de mais do que a integridade do software e mecanismos de proteção do sistema operacional em uso; ela também é dependente da própria configuração e uso do software [ZERKLE & LEVITT, 1996].

2.2. Segurança Computacional

Segurança de redes de computadores é uma área de crescente interesse e preocupação, atingindo desde administradores preocupados com a segurança e o bom funcionamento de seus *sites* até *hackers* e vândalos buscando novos métodos e técnicas de ataque. O termo *hacker* deriva da década de 70, quando designava pessoas que possuíam um profundo conhecimento sobre computadores, sistemas operacionais e softwares, não tendo nenhuma ligação com os atuais significados no que se refere a atacantes e intrusos. Geralmente os termos *hacker* e *cracker* são usados indiscriminadamente, mas algumas diferenciações são encontradas na literatura.

Hacker é o indivíduo com um profundo conhecimento, mas geralmente sem intenções destrutivas. Seu propósito é unicamente provar que consegue invadir um determinado sistema e quanto mais protegido for este sistema, maior será seu empenho. De forma

oposta, o *cracker* é aquele cujo único objetivo é destruir, danificar e causar perdas. Um estereótipo típico criado é o de um adolescente em sua casa que, a partir de um computador e um modem, profere ataques aos computadores de grandes organizações. No entanto, o que se vê atualmente é o uso de ataques em espionagem, guerra cibernética e roubo de segredos industriais.

A segurança de uma rede de computadores pode ser comparada à segurança de uma casa. Não importa que grau de segurança exista, não importa que sistemas ou trancas sejam usados. Quando alguém decide com suficiente empenho, invadir, provavelmente terá êxito. De modo análogo, todas as medidas no sentido de se aumentar a segurança de uma rede tem como objetivo torná-la tão segura quanto possível, já que nenhum sistema conhecido garante o estado-da-arte em termos de proteção. Geralmente, um atacante irá analisar a relação custo/benefício, ou seja, o quão custoso e complicado será invadir um determinado sistema ponderado aos lucros que ele alcançará com tal invasão. Uma vez que esta proporção se torne inviável, pode-se dizer que foi atingido um bom grau de segurança [BONIFÁCIO, 1998].

Ao mesmo tempo em que a Internet é o meio pelo qual a maioria das intrusões e ataques ocorrem, é também através dela que são largamente disponibilizados e veiculados documentos explicando e demonstrando técnicas de *hacking*, furos de segurança e casos de monitoração de intrusões em andamento. É possível encontrar com facilidade documentos do tipo “receita de bolo”, que ensinam passo a passo técnicas de intrusão. Ainda que sejam técnicas simples, podem ser altamente destrutivas, tendo-se em vista que boa parte das redes conectadas à Internet não possuem um completo domínio das questões relacionadas à segurança.

Existem atualmente diversos sites dedicados exclusivamente a este assunto, contendo documentos que abordam desde técnicas básicas de *hacking* até conceitos avançados para se aumentar a segurança de uma rede. Exemplos de sites são: *http://packetstorm.securify.com* e *http://www.securenet.com.br*. Diariamente são divulgados relatórios com novos furos de segurança nos mais variados sistemas operacionais e softwares. Como existe uma certa demora no lançamento de *patches* de segurança e uma grande dificuldade dos administradores se manterem constantemente

atualizados, tem-se um cenário em que uma rede pode-se alcançar um estado altamente vulnerável muito rapidamente.

Este cenário de computadores, redes e comunicações inseguras deve-se, em parte, ao modo como a Internet foi projetada. O principal foco do projeto da Internet, e mais basicamente do protocolo TCP/IP, estava muito distante dos atuais usos da Internet. Seu projeto previa inicialmente o uso por instituições militares e de pesquisa. O crescimento e a popularização da Internet, o surgimento de aplicações de comércio eletrônico, a interligação das redes das diversas filiais de uma empresa e muitos outros serviços oferecidos pela Internet atualmente não eram sequer supostas pelos seus projetistas e técnicos. O crescimento da Internet levou a uma mudança no foco e no perfil dos usuários e das aplicações da rede. Em vista disso, os protocolos, serviços, sistemas operacionais como o UNIX e os softwares que são utilizados na Internet não foram projetados e especificados com as devidas preocupações com relação à segurança. No UNIX, as senhas circulam totalmente abertas pela rede, o protocolo TCP/IP não prevê nenhum esquema de criptografia dos dados ou autenticação das máquinas e usuários envolvidos em uma conexão. Os atuais sistemas, como o Windows NT, foram criados em cenários com preocupações específicas sobre segurança. Porém ainda não se mostraram soluções totalmente confiáveis devido a fatores como pouco tempo para desenvolvimento e testes, o que acarretou em falhas e furos de segurança.

2.3. Sistemas de Detecção de Intrusão

Uma intrusão pode ser definida como: "*um conjunto de ações que tentam comprometer a integridade, confidencialidade ou disponibilidade de recursos"* [CROSBIE & SPAFFORD, 1995a].

Apesar do uso dos diversos esquemas de segurança existentes, ainda existe a possibilidade de ocorrência de falhas nestes esquemas e, portanto, é desejável a existência de sistemas capazes de realizar a detecção de tais falhas e informar o administrador da rede. Tais sistemas são conhecidos como sistemas de detecção de intrusão (SDI).

Intrusões são difíceis de detectar porque existem muitas formas pela qual elas podem acontecer. Intrusos podem explorar as fraquezas conhecidas da arquitetura dos sistemas ou

explorar o conhecimento de um sistema operacional para conseguir a autenticação normal de um processo. A tentativa de retirar uma falha do sistema pode introduzir uma nova falha ou expor uma falha existente, dando a oportunidade para um novo ataque.

Tentativas de ataque acontecem de acordo com algumas técnicas de acesso e freqüentemente o invasor estava fisicamente fora do sistema sob ataque. Hoje, percebemos que boa parte, senão a maioria dos ataques, parte internamente do próprio ambiente. Os primeiros modelos de sistemas de detecção de intrusão projetados para computadores isolados usavam algoritmos básicos que incluem análise de funções multinomiais e aproximação de matrizes covariantes para detectar desvio do comportamento normal, bem como sistemas especialistas para detectar violação de políticas de segurança. Os modelos mais recentes monitoram um grande número de redes de computadores e transferem a informação monitorada para ser processada em um equipamento central que emprega técnicas de sistemas distribuídos.

A maioria dos SDIs tem um processo auditor (*daemon*) em cada máquina, responsável por capturar ações de violação de segurança dentro da máquina. Sistemas baseados em redes, ao invés de utilizar pistas de auditoria, analisam o tráfego de pacotes dentro da rede para detectar comportamento intrusivo.

As funcionalidades de um sistema de detecção tornam-se de vital importância na medida em que fornecem meios de inferir sobre o conteúdo das conexões permitidas e detectar as que apresentem um comportamento suspeito ou não condizente com a política de segurança implantada.

2.4. Agentes Móveis

Infelizmente não existe um consenso dos pesquisadores sobre a exata definição de agente. Uma definição radicalmente baseada em Inteligência Artificial é apresentada por Selker: "*Agentes são programas de computadores que simulam o relacionamento humano, por fazerem exatamente o que uma outra pessoa faria por você"* [SELKER, 1994]. Minsk define um agente como "*um sistema que pode servir como um mensageiro, pelo motivo de possuir algumas habilidades de especialista"* [MINSK & RIECKEN, 1994]. Para Genesereth & Ketchpel agentes são "*componentes de software que comunicam com seus*

pares por troca de mensagens em uma linguagem de comunicação de agentes" [GENESERETH & KETCHPEL, 1994].

O problema com a definição deve-se, em parte, à falta de coordenação entre as diversas pesquisas paralelas que foram feitas ao longo dos anos. Por outro lado, o termo agente não é propriedade dos pesquisadores da área, sendo usado diariamente no mundo real (agente de viagem, agente econômico, agente de seguros, etc.) [REAMI, 1998]. Segundo o dicionário Aurélio da língua Portuguesa, o termo agente pode ser definido como "*aquele que trata de negócios por conta alheia*" [FERREIRA, 1977]. Segundo Nwana [NWANA, 1996], existe tanta chance de se atingir um consenso sobre a definição de agente, quanto dos pesquisadores de IA têm de chegar a uma definição para inteligência artificial, ou seja, nenhuma.

Em termos gerais, um agente pode ser definido como um software capaz de executar uma tarefa complexa em nome de um usuário [ENDLER, 1998].

Assim como na definição de agentes, são apresentadas diversas propostas para definição de agentes móveis. Um dos trabalhos mais citados em toda literatura pesquisada é o relatório de pesquisa apresentado por Chess, Harrison e Kershenbaum [CHESS et al, 1995]. Segundo eles, agentes móveis *"são programas tipicamente escritos em uma linguagem script, que podem ser disparados de um computador cliente e transportados para um computador remoto para execução".* Além disso, esses elementos possuem características inerentes ao conceito de multiagentes, que proporcionam um bom desempenho em sistemas de objetos distribuídos.

2.5 - Ambientes de Agentes Móveis

A linguagem Java viabilizou a concepção de diversos sistemas experimentais de agentes móveis. Numerosos sistemas estão atualmente em desenvolvimento e a maioria está disponível via Web. Exemplificando alguns destes ambientes, pode-se citar:

- **Aglets -** Criada pela IBM Corporation, constitui-se de uma API que tenta espelhar o modelo de *Applets Java*. A proposta de seu desenvolvimento era dar mobilidade aos *Applets* [LANGE & OSHIMA, 1998].
- **Odyssey** - A General Magic Inc. criou o primeiro sistema de agentes móveis comercial, chamado Telescript. Entretanto, o Telescript teve pouco sucesso, pois era baseado em

uma linguagem e arquitetura de rede proprietárias. A Popularidade da Internet motivou a General Magic a reimplementar um sistema de agentes móveis baseado em Java chamado *Odyssey*. Este sistema simplesmente mapeou os conceitos do Telescript em classes Java, permitindo aos desenvolvedores criar suas próprias aplicações [MAGIC, 1998].

- **Concórdia** - Concebido pela Mitsubishi constitui-se de um *framework* para o desenvolvimento e gerenciamento de aplicações de agentes móveis. O Concórdia compreende múltiplos componentes, todos escritos em Java, os quais são combinados para prover um ambiente para aplicações distribuídas. Este sistema é simples e requer somente uma implementação padrão da máquina virtual Java. Seu ambiente é composto de um servidor e um conjunto de agentes. O Concórdia provê mecanismos de segurança para execução segura de agentes, além de suportar mecanismos de *checkpoint* para tolerância a falhas [MITSUBISHI, 1997].
- **Voyager** - Criado pela ObjectSpace, consiste de uma plataforma ORB (*Object Request Broker*) implementado em Java e com suporte a agentes. Voyager implementa os mecanismos tradicionais de troca de mensagens, somados à capacidade de objetos moverem-se através da rede como agentes. Uma desvantagem de Voyager é não oferecer mecanismos de segurança contra agentes não autorizados [SPACE, 1998].

3. Sistemas de Detecção de Intrusão Baseado em Agentes Móveis

3.1. Considerações Iniciais

O grau de proteção a cada ação maliciosa está diretamente relacionado ao tempo e esforços gastos construindo e gerenciando os sistemas de segurança. Utilizando-se complexas ferramentas, as quais continuamente monitoram e notificam atividades consideradas suspeitas, pode-se conseguir identificar essas atividades no momento em que elas ocorrem. Entretanto, isso envolve um alto custo em termos de tempo e dinheiro na construção e gerenciamento de sistemas de monitoria. Esses sistemas também impõem penalidades de performance no ambiente que está sendo protegido, o que pode ocasionar a sua rejeição pelos usuários.

A arquitetura monolítica de Sistemas de Detecção de Intrusão (SDI), comumente utilizada em sistemas comerciais ou de pesquisa, apresenta um número de problemas que limitam sua capacidade de configuração, escalabilidade ou eficiência.

Este capítulo apresenta a modelagem apresentada por [BERNARDES, 1999] para o desenvolvimento de um Sistema de Detecção de Intrusão não-monolítico baseado em agentes móveis e seu relacionamento com este projeto de pesquisa.

3.2. Vantagens de um Sistema Não-Monolítico

A abordagem monolítica apresenta alguns problemas práticos. Se uma nova forma de intrusão não prevista no sistema é descoberta, o SDI deve ser completamente reconstruído para conseguir tratá-la e isso, com certeza, não é uma ação trivial. [ZAMBONI et al., 1998] [CROSBIE & SPAFFORD, 1995a, 1995b].

Outra preocupação diz respeito à tolerância à falhas, uma vez que um sistema monolítico apresenta-se como um único ponto de falha e ataques. Conseqüentemente, metodologias de ataques bem conhecidas (como por exemplo, ataques de *Denial of Service*

à CNN.com, Globo.com e UOL), quando proferidas contra a máquina que hospeda o SDI, comprometem por completo a integridade do sistema.

A utilização de agentes autônomos têm sido proposta por alguns autores como uma forma de se construir sistemas de detecção de intrusão não-monolíticos [CROSBIE & SPAFFORD, 1995a, 1995b] [ZAMBONI et al, 1998]. A capacidade dos agentes autônomos de manter informação específica do seu domínio de aplicação, nesse caso, aplicação de segurança, dá a estes agentes e conseqüentemente a todo o sistema, grande flexibilidade.

Em vez de um grande módulo monolítico, este trabalho apresenta a proposta de uma abordagem modular baseada em agentes autônomos e móveis para o desenvolvimento de um SDI. Este SDI consiste de um conjunto de pequenos processos (agentes) que podem agir independentemente no ambiente em construção. Eles serão desenvolvidos para moverem-se pelo ambiente no qual estão inseridos, observarem os comportamentos do sistema, cooperarem uns com os outros via passagem de mensagens, notificarem quando uma ação for considerada suspeita e ainda, proverem ações reativas (contra-ataque).

Cada agente observa somente um pequeno aspecto de todo o sistema. Um simples agente, sozinho, não pode formar um sistema de detecção de intrusão, uma vez que sua visão é limitada a uma pequena "fatia" do sistema. Entretanto, se muitos agentes operam em um sistema e cooperam uns com os outros, então um poderoso SDI pode ser desenvolvido. Uma vez que os agentes são independentes uns dos outros, eles podem ser adicionados e removidos do sistema dinamicamente, de forma que não é necessário reconstruir todo o SDI ou ainda, interromper suas atividades. Assim, a qualquer sinal de identificação de uma nova forma de ataque, novos agentes especializados podem ser desenvolvidos, adicionados ao sistema e configurados para atender uma política de segurança específica.

Outra vantagem da abordagem descrita anteriormente é a facilidade de configuração apresentada pelo sistema em atendimento às necessidades políticas do ambiente ao qual está inserido. Isso se torna uma característica importante uma vez que, conforme visto no capítulo 2, o que é considerado uma quebra de segurança para um ambiente pode não ser em outro, em função do tipo de informação que se quer proteger.

Uma vez que a mudança é subjacente a todo trabalho de software e que esta é inevitável quando se constrói sistemas baseados em computador, outra vantagem deste

sistema é a sua alta manutenibilidade. Definida qualitativamente em [PRESSMAN, 1995] como sendo "a facilidade com que um software pode ser entendido, corrigido e/ou aumentado", esta se torna a meta primordial que orienta os passos de um processo de engenharia de um software.

Sendo dividido em módulos contendo um conjunto de pequenos agentes especializados em uma única função e que conseqüentemente apresentam uma menor complexidade lógica, o sistema procura minimizar o esforço gasto com a manutenibilidade. Isso se deve ao fato de que, desta forma, cada agente apresenta uma estrutura bem compreensível, facilitando o seu entendimento e conseqüente necessidades de manutenção. Isso é refletido diretamente em termos de:

- Tempo de reconhecimento do problema;
- Tempo de análise do problema;
- Tempo de especificação das mudanças;
- Tempo de correção (ou modificação) ativa;
- Tempo de testes locais;
- Tempo de testes globais;
- Tempo de revisão de manutenção;
- Tempo de recuperação total.

Cada uma das métricas anteriores pode, de fato, ser registrada sem grandes dificuldades. Além dessas medidas orientadas para o tempo, a manutenibilidade pode ser medida indiretamente ao considerarmos as medidas da estrutura do projeto e as métricas da complexidade do sistema, que indicarão também um ganho significativo no momento da inserção de novas funções (novos agentes).

Além dessas vantagens, [CROSBIE & SPAFFORD, 1995a; 1995b] especificam um sistema baseado em agentes autônomos no qual as capacidades dos agentes são modificadas através do uso algoritmos genéticos. Os autores reconhecem, entre outras, as seguintes vantagens de sistemas baseados em agentes autônomos sobre sistemas monolíticos:

- **Fácil configuração:** uma vez que é possível ter uma série de pequenos agentes especializados em tarefas específicas de detecção, o sistema de detecção pode ser configurado da forma mais adequada para cada caso; a adição e remoção de agentes do sistema são facilitadas;

- **Eficiência:** agentes podem ser treinados previamente e otimizados para que realizem suas tarefas de maneira a gerar a menor sobrecarga possível no sistema;
- **Capacidade de extensão:** um sistema de agentes pode ser facilmente modificado para operar em rede e permitir migração para rastrear comportamentos anômalos através da rede, ou mover para máquinas onde eles possam ser mais úteis;
- **Resistência à subversão:** caso um sistema de defesa seja subvertido, ele poderá dar a falsa sensação de segurança. Entretanto, isto se torna mais difícil, pois os conhecimentos adquiridos de um agente não fornecem o conhecimento das operações de outros, visto que eles desempenham funções diferentes;
- **Escalabilidade:** para atuar em sistemas maiores, basta adicionar mais agentes e aumentar sua diversidade.

3.3. SDI Baseado em Agentes Autônomos e Móveis

O conceito principal que envolve o SDI baseado em agentes autônomos e móveis é a simplicidade. Cada agente é uma entidade simples que irá desempenhar uma atividade específica e cooperar com outros agentes de forma mais eficiente possível. Quando uma atividade for considerada suspeita por um agente, ele irá comunicar aos demais agentes do sistema sua suspeita de uma possível intrusão. Neste momento, será acionado um agente (ou um conjunto de agentes) com um maior grau de especialização naquele tipo de suspeita.

Naturalmente um agente poderá cometer um erro, que será identificado por um agente com um nível de especialização superior. Uma vez que um número maior de agentes suspeita de uma possível intrusão, uma mensagem pode ser enviada pedindo a intervenção de um operador humano (via alguns processos de monitoramento) e agentes de reação poderão ser acionados.

Isso demonstra que uma decisão deverá ser tomada em conjunto, já que nenhum agente possui a autoridade de identificar uma intrusão por conta própria. Essa decisão será tomada com base no consenso de vários agentes no sistema. Se somente um agente suspeita de uma intrusão, ele poderá ser ignorado após uma votação dos agentes envolvidos naquela suspeita. Entretanto, se mais de um agente suspeitam de um comportamento anômalo, então há uma maior probabilidade de ser uma intrusão potencial e neste caso, poderá ser tomada a

decisão de comunicar um operador humano ou acionar agentes especializados em contra ataque. Fica claro que certos eventos podem ser mais "importantes" neste esquema do que outros. Por exemplo, 50 falhas em tentativas de *login* como *root* receberá um grau de suspeita maior que uma conexão de *FTP* externa ao domínio monitorado.

Uma arquitetura para a introdução de agentes móveis em Sistemas Detecção de Intrusão foi introduzida por [BERNARDES, 1999] e é apresentada na figura 1. As camadas são numeradas a partir da camada de Vigilância (camada 1), e cada uma delas representa um grupo de tarefas específicas desempenhadas por agentes especializados nas funções desta camada. Através do mecanismo de troca de mensagens, um agente em uma camada aciona um ou mais agentes em uma camada superior. Em outras palavras, a camada N utiliza os serviços da camada N-1, desempenha suas funções e fornece serviços para a camada N+1.

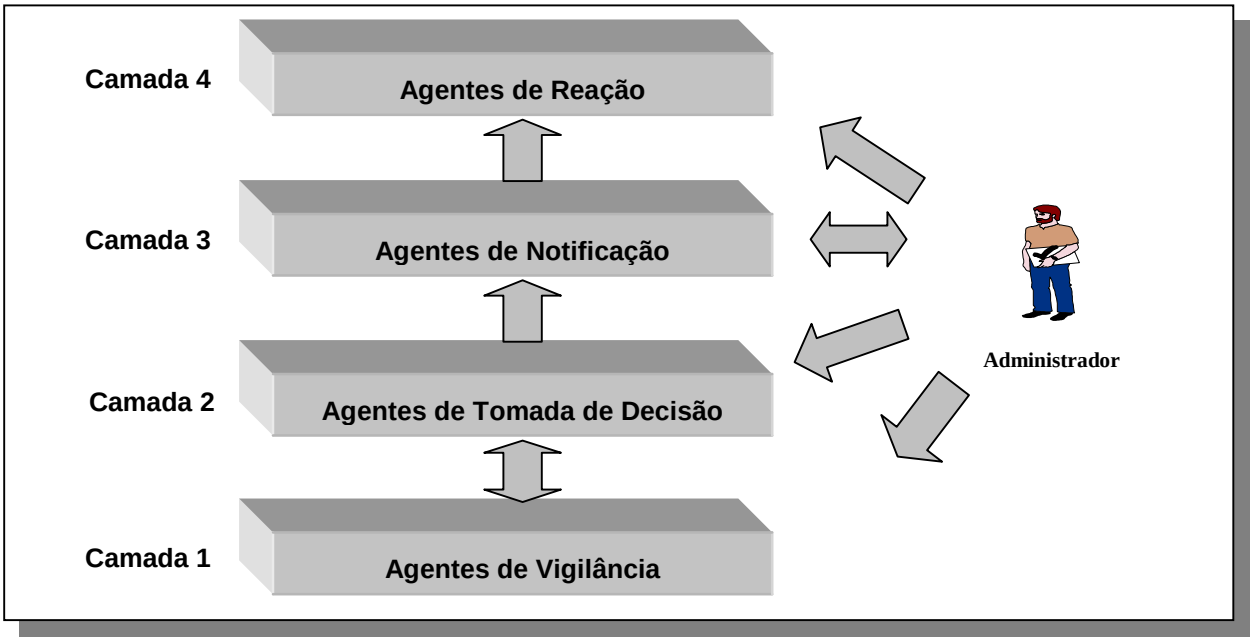

Figura 1 - Modelagem em Camadas para o Sistema Proposto

Com base em informações coletadas pelos *Agentes de Vigilância*, *Agentes de Tomada* de Decisão entrarão em ação, analisando e identificando possíveis intrusões. Caso uma ação seja considerada suspeita por estes agentes, *Agentes de Notificação* serão acionados e cuidarão de notificar o administrador da rede (via *e-mail*, *pager*, chamada telefônica, alarme, etc.) ou acionar os agentes de nível superior. Em último nível, encontram-se os *Agentes de Reação*. Estes agentes cuidarão de contra-atacar automaticamente as possíveis

intrusões, com base nas informações dos agentes de notificação ou ainda, serem acionados através de uma intervenção do administrador da rede.

Apesar do exemplo anterior exemplificar uma comunicação *bottom-up* através das camadas da arquitetura proposta, há a possibilidade de uma comunicação *top-down* entre a camada de tomada de decisão e a camada de vigilância. Exemplificando, temos um cenário em que um Agente de Tomada de Decisão, após receber uma mensagem ou um conjunto de dados dos Agentes de Vigilância, poderá no momento em que desempenhar uma análise, necessitar de mais informações. Neste ponto, novos Agentes de Vigilância deverão ser acionados e mais informações deverão ser coletadas, na tentativa de se conseguir uma decisão com um maior grau de certeza.

A ampliação do SDI para atender uma nova configuração de ataque poderá envolver o desenvolvimento e conseqüente adição de novos agentes em uma única camada ou ainda, a criação de um novo cenário que envolverá a adição de agentes em todas as camadas.

3.4. Cenários de Execução do Ambiente

Entre os trabalhos a serem realizados neste projeto, encontra-se a modelagem de um cenário de execução do sistema, visando sua melhor compreensão e uma perfeita visualização do relacionamento entre as camadas do modelo apresentado na seção 3.3.

A representação cenários será feita por um modelo de alto nível, composto inicialmente por 4 processos que estão associados respectivamente a cada uma das camadas do modelo apresentado na figura acima. Assim, a numeração utilizada em cada processo, além de identificar a seqüência lógica de execução do cenário, faz a associação do processo à sua respectiva camada no modelo. Como exemplo, temos que o processo 1 contém um ou mais agentes da camada 1 (a Camada de Vigilância) e assim sucessivamente.

Dependendo da complexidade, um processo (agente) neste modelo de alto nível pode ser “expandido” para tornar-se um novo diagrama (conjunto de agentes). O cenário representa uma fronteira de automação para o desempenho de uma função de segurança computacional no sistema proposto. Assim, a criação de um novo cenário de execução corresponderá à implementação e posterior inserção de novas funções no sistema proposto.

3.5. Considerações finais

Com base na arquitetura apresentada, cabe a este projeto as tarefas de identificação, modelagem, implementação e avaliação de um cenário de execução para constituição de Sistema de Detecção de Intrusão baseado em agentes móveis. Essas tarefas apresentam-se com uma extensão dos trabalhos desenvolvidos em [BERNADES, 1999].

Os objetivos desse projeto e suas tarefas iniciais são apresentados no próximo capítulo.

4. Proposta de Trabalho

4.1. Objetivos

O objetivo específico do projeto é fazer a modelagem, implementação e validação de um dos cenários de execução necessários para a constituição de um SDI: **a identificação de usuários anômalos**, como descrito na seção 3.4.

Entre os objetivos globais do projeto, encontram-se:

- a minimização dos custos apresentados por um SDI monolítico, através da modelagem e conseqüente implementação de alguns cenários de execução utilizando-se das técnicas de agentes móveis;
- contribuir para a avaliação e validação da arquitetura apresentada em [BERNARDES, 1999] através de sua implementação e posterior utilização no sistema acadêmico de nosso Instituto;
- proporcionar a extensão dos sistemas de detecção de intrusão existentes com a inserção da tecnologia de agentes móveis;
- contribuir para o avanço das tecnologias de segurança computacional;

4.2. Implementação do cenário *Identificação de Usuários Anômalos*

Um intruso (interno ou externo) que consiga um *username* e uma senha válidos poderá abrir uma conexão e ter acesso ao sistema. Uma vez dentro do sistema (mesmo em acesso comum, ou seja, não como *root*) poderá usar diversas técnicas disponíveis para executar suas intenções, conseguir privilégios de *root* ou ainda, agir ilegalmente em nome do usuário legítimo.

Na tentativa de identificação deste tipo de ataque, este cenário apresenta as características de um Sistema de Detecção de Intrusão baseado em anomalias. O objetivo é analisar comportamentos que possam identificar conexões que fujam dos padrões previamente estabelecidos para cada usuário. Para isso, o sistema possui:

agentes que coletam uma lista contendo informações dos usuários conectados ao sistema (horário de *login*, origem da conexão, etc); agentes que irão proceder com a identificação de possíveis anomalias tomando como base *profiles* previamente estabelecidos para os usuários; agentes de notificação do grau de suspeita e ainda, agentes para execução de medidas reativas, tais como bloqueio de conexão.

4.3. Características Desejáveis do Projeto

No desenvolvimento de um ambiente de detecção de intrusão conforme o que discutimos até o presente momento, algumas características tornam-se desejáveis e desta forma, o sistema deve:

- estar em execução contínua com o mínimo possível de supervisão humana;
- ser capaz de executar em *background* no ambiente que está sendo observado e relatar suas conclusões;
- impor o mínimo de *overhead* possível no ambiente onde está sendo executado, de forma a não interferir nas operações normais;
- apresentar facilidades de configuração para atender a política de segurança do ambiente que está monitorando;
- adaptar-se às mudanças do ambiente e dos comportamentos dos usuários através do tempo (permissão para execução de novas aplicações, usuários trocando de atividade ou novos recursos sendo utilizados);
- monitorar um grande número de *hosts* enquanto providencia resultados em tempo hábil e de maneira exata;
- apresentar uma moderada degradação de serviço, uma vez que algum componente do sistema pode ter seus serviços interrompidos por qualquer razão, o resto deverá ser afetado da menor forma possível;
- permitir reconfiguração dinâmica, uma vez que um grande número de *hosts* está sendo monitorado, torna-se impraticável reinicializar o SDI toda vez que um novo cenário for implementado.

4.4. Materiais e Métodos

4.4.1. Modelagem

No processo de modelagem, torna-se interessante a representação do cenário de execução do sistema, visando sua melhor compreensão e uma perfeita visualização do relacionamento entre as camadas do modelo apresentado. Para essa representação, deverá ser utilizada uma ferramenta lógica conhecida como Diagrama de Fluxo de Dados (*DFD)*, por ser uma ferramenta simples, de fácil entendimento e largamente utilizada na modelagem de sistemas de informação. Como os símbolos utilizados no DFD não são físicos, ele mostra a essência da lógica subjacente do sistema a ser modelado [GANE, 1988] e [GANE & SARSON, 1994].

A figura 4-2 apresenta os símbolos utilizados e sua representação neste contexto. Maiores informações sobre DFDs podem ser observadas em [GANE, 1988] e [GANE & SARSON, 1994].

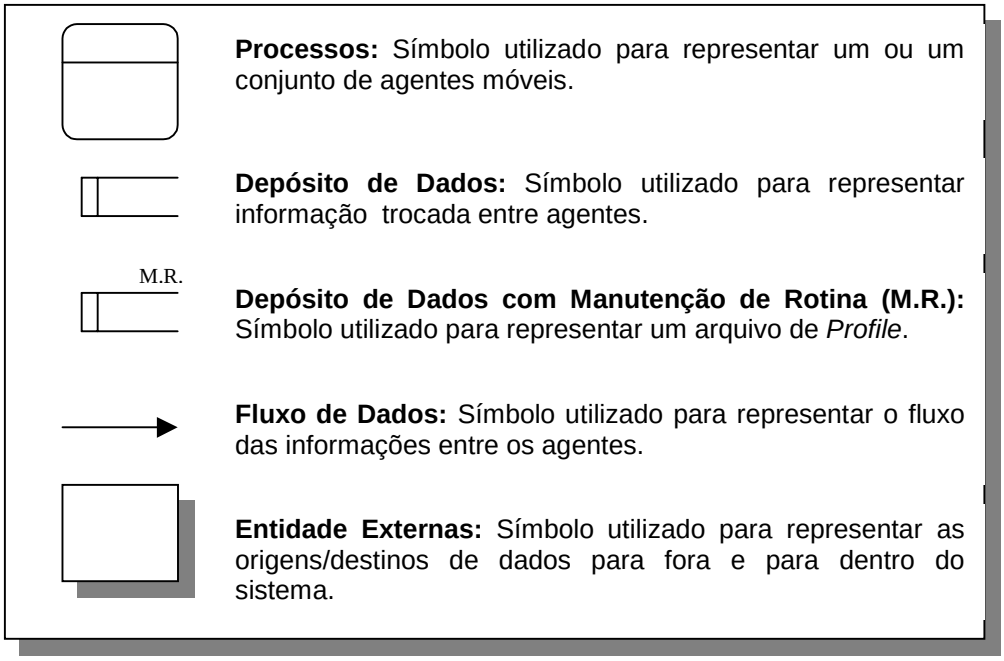

Figura 2 - Representação dos elementos de DFD

4.4.2. Linguagem

No momento de implementação, deverá ser utilizada a linguagem Java. Desenvolvida pela Sun Microsystems, Java tem chamado a maioria das atenções e expectativas em mobilidade de código. A meta original dos projetistas da linguagem foi fornecer portabilidade, clareza, fácil aprendizado e uma linguagem orientada a objetos que contribuísse para o crescimento da Internet. O compilador Java traduz programas fontes Java em um código intermediário e independente da plataforma chamado *Java Byte Code*, que é interpretado por uma *Java Virtual Machine* (JVM).

4.4.3. Ambiente de Desenvolvimento

Dentre as diversas plataformas de desenvolvimento de agentes móveis apresentadas, será utilizada a plataforma *Aglets*. Esta escolha deve-se principalmente ao fato das configurações de segurança oferecidas por este ambiente. No modelo *Aglets*, um agente móvel tem sua própria *thread* de controle, é dirigido a evento e estabelece comunicação por passagem de mensagem. Entretanto, estão sendo desenvolvidas pesquisas paralelas a este projeto [PEREIRA FILHO, 2000] que objetivam a análise e identificação das potencialidades dos sistemas de agentes para agentes móveis mais utilizados. Assim, após a modelagem e especificação, poderá ser determinada, com base nas conclusões obtidas no trabalho citado acima, a escolha de um outro ambiente de agentes, o que não implicará em mudanças nas etapas inicias.

4.4.4. Ciclo de Vida (Paradigma de Desenvolvimento)

Ao combinarmos métodos abrangentes para todas as fases de desenvolvimento do software, melhores ferramentas para automatizar esses métodos, blocos de construção mais poderosos para a implementação do software, melhores técnicas para a garantia da qualidade do software, uma filosofia de coordenação predominante, controle e administração, conseguimos uma disciplina para o desenvolvimento do software. Essa disciplina é representada na Engenharia de Software clássica como Paradigmas de Desenvolvimento. Nesse projeto, o paradigma a ser utilizado é o *Ciclo de Vida Clássico.*

Esse requer uma abordagem sistemática, seqüencial ao desenvolvimento do software, que se inicia no nível do sistema e avança ao longo da análise, projeto, codificação, teste e manutenção. Modelado em função do ciclo da engenharia convencional, esse paradigma do ciclo de vida abrange as seguintes atividades: Engenharia de sistemas, análise, projeto, codificação, teste e manutenção.

4.5. Trabalhos Relacionados Existentes

Recentemente, o COAST *(Computer Operations, Audit and Security Technology, Computer Science Departament at Purdue University*) liberou uma implementação de um ambiente que segue as idéias do sistema proposto por Crosbie e Spafford [CROSBIE & SPAFFORD, 1995a, 1995b]. Este ambiente denominado *Autonomous Agents for Intrusion Detection* (AAFID) foi implementado utilizando-se a linguagem de *script*s Perl e diversos recursos de administração de sistemas e segurança comuns em ambiente UNIX. O ambiente AAFID possui duas entidades distintas que suportam a execução dos agentes do sistema: *Transceivers* e *Monitors*, sendo estes últimos entidades de mais alto nível, que podem detectar possíveis eventos intrusivos não notados por entidades mais simples como *Transceivers*. As informações preliminares sobre o sistema AAFID podem ser encontradas em [ZAMBONI et al, 1998].

Outro trabalho recente na área de aplicação de agentes autônomos em sistemas de detecção de intrusão é apresentado por [BARRUS & ROWE, 1998]. A idéia dos autores é utilizar agentes autônomos estáticos que se comunicam através de um sistema de mensagens de alerta através de uma arquitetura distribuída. Algumas abordagens interessantes sugeridas são: a utilização de agentes especializados na detecção baseada em anomalia, detecção baseada em uso indevido e a criação de objetos específicos para tratar os diversos tipos de ataque.

Em sua tese de mestrado, [REAMI, 1998] apresenta a especificação e o protótipo de um ambiente de gerenciamento de segurança computacional apoiado por agentes móveis. O ambiente especificado e prototipado pode ser considerado um avanço importante, pois, além de fornecer recursos tecnológicos avançados e consoantes com o desenvolvimento da área, permite uma abordagem holística do problema do gerenciamento de segurança.

Em [BERNARDES, 2000] é apresentada a modelagem e implementação de um cenário de execução para um SDI baseado em Agentes Móveis. Este projeto visa contribui para sua validação, extensão e avaliação.

4.6. Cronograma

O projeto a ser desenvolvido seguirá a seguinte seqüência de eventos:

1. Análise de Requisitos: Entendimento e especificação formal das funções que o sistema deverá desempenhar.

2. Modelagem do Cenário Proposto: Representação lógica das funções a serem implementadas através de uma modelagem com DFD.

3. Implementação do Cenário Proposto: Codificação do modelo conforme item 4.3.2.

4. Avaliação do Cenário Implementado: Testes e verificação de consistência na implementação

5. Validação do Cenário Implementado: Verificação das potencialidades da implementação através de sua execução em um ambiente real.

6. Relatório Final: Redação do relatório final das atividades desenvolvidas na pesquisa.

	Agosto/2000	**Setembro/2000**	**Outubro/2000**	**Novembro/2000**	**Dezembro/2000**	**Janeiro/2001**	**Fevereiro/2001**	**Março/2001**	**Abril/2001**	**Maio/2001**	**Junho/2001**
1. Análise de Requisitos	■	■									
2. Modelagem do Cenário Proposto			■	■							
3. Implementação do Cenário Proposto					■	■	■				
4. Avaliação do Cenário Implementado								■			
5. Validação do Cenário Implementado									■	■	
6. Relatório Final											■

Estão previstos também a elaboração e submissão de artigos/relatórios técnicos a congressos e simpósios científicos.

5. Referências

[BARRUS & 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, Jun-Jul 1998.
[BERNARDES, 1999]	BERNARDES, M.C. "*Avaliação do uso de Agentes Móveis em Segurança Computacional"*. Dissertação de mestrado apresentada e Defendida no ICMC/USP em dezembro de 1999.
[BERNARDES, 2000]	BERNARDES, M.C.; MOREIRA, E. S. "*An Architecture for Intrusion Detection System Based on Mobile Agentes"*. ISADS2000, International Symposium on Advanced Distributed System, Guadalajara, Jalisco, México, March 2000.
[BONIFÁCIO, 1998]	BONIFÁCIO Jr., J. M. *Sistemas de Segurança Distribuído: Integração de Firewalls com Sistemas de Detecção de Intrusão.* São Carlos, 1998. Dissertação de Mestrado - Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo.
[CHESS et al, 1995]	CHESS, David; HARRISON, Colin; KERSHENBAUM, Aaron. *Mobile Agentes: Are They a Good Idea?* IBM Research Report. Disponível on line em: http://www.research.ibm.com/iagentes/paps/mobile-idea.ps. Visitado em 15/01/1999.
[CROSBIE & SPAFFORD 1995a]	CROSBIE, M.; SPAFFORD, E.H. *Defending a Computer System using Autonomous Agents*. Department of Computer Sciences, Purdue University, 1995. (Relatório Técnico CSD-TR-95-022; Coast TR 95-02). Disponível on-line em: http://www.cs.purdue.edu/homes/spaf/tech-reps/9522.ps. Visitado em 15/01/1999.
[CROSBIE & SPAFFORD 1995b]	CROSBIE, M; SPAFFORD, E.H. *Active Defense of a Computer System using Autonomous Agents*. Department of Computer Sciences, Purdue University, 1995. (Relatório Técnico CSD-TR-95-008). Disponível on-line em: http://www.cs.purdue .edu/homes/spaf/tech-reps/9508.ps. Visitado em 15/01/1999.
[ENDLER, 1998]	ENDLER, Markus. *Novos Paradigmas de Interação usando Agentes Móveis.* Departamento de Ciência da Computação. IME. USP. SBRC98. Disponível on-line em http://www.ime.usp.br/~endler/paperlinks/sbrcslides.ps. Visitado em 25/03/1999.
[FERREIRA , 1977]	FERREIRA, Aurélio. B. H. *Minidicionário Aurélio da Língua Portuguesa.* 1ª edição, 16ª impressão. Editora Nova Fronteira, 1977.
[GENESERETH & KETCHPEL, 1994]	GENESERETH, M.R.; KETCHPEL, S.P. *Software Agents.* Communications of the ACM, 37(7): 48-53, 1994.
[GANE, 1988]	GANE, Chris.*Desenvolvimento rápido de sistemas*. Rio de Janeiro: LTC-Livros Técnicos e Científicos Editora, 1988
[GANE & SARSON, 1994]	GANE, Chris; SARSON, Trish. *Análise Estruturada de Sistemas*. Rio de Janeiro: LTC-Livros Técnicos e Científicos Editora, 1994.
[ISO, 1989]	International Organization for Standardization / International Eletrotechnical Committee. *"Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture*". *International Standard 7498-2, 1989.*
[LANGE & OSHIMA, 1998]	LANGE, D.B; OSHIMA, M. *Programming And Deploying Java Mobile Agents with Aglets.* Addison Wesley Longman, Inc. 1998.
[MINSK & RIECKEN, 1994]	MINSK, M.; RIECKEN, D. *A conversation with Marvin Minsk about Agents.* Communications of the ACM, 37(7):23-29, 1994.
[NWANA,1996]	NWANA, H.S. *Software Agents: An Overview*. In: Knowledge Engineering Review, vol. 11, no. 3, p205-244, Outubro/Novembro 1996.
[PEREIRA FILHO, 2000]	PEREIRA FILHO, S. F. Avaliação de Ambientes Servidores para Agentes Móveis. Mini-Dissertação de Mestrado. ICMC , USP, 2000.

[PRESSMAN, 1995]	PRESSMAN, Roger S. *Engenharia de Software*. São Paulo: Makron Books, 1995.
[REAMI, 1998]	REAMI, E. R. *Especificação e Prototipagem de um Ambiente de Gerenciamento de Segurança Apoiado por Agentes Móveis.* São Carlos, 1998, 82p., Dissertação (Mestrado) – Instituto de Ciências Matemáticas de Computação de São Carlos, Universidade de São Paulo.
[SELKER, 1994]	SELKER, T. *Coach: A Teaching Agent that Learns*. Communications of the ACM, 37(7):92-99. 1994.
[SOARES, 1995]	SOARES, L. F. G., LEMOS, G., COLCHER, S. *Redes de Computadores: das LANs, MANs e WANs às redes ATM.* 2. Edição. Rio de Janeiro: Campus, 1995. 704 p.
[SPACE, 1998]	OBJECT SPACE Inc. *ObjectSpace Voyager Overview*. Disponível on-line em http://www.objectspace.com/products/vgrOverview.htm.
[ZAMBONI et al, 1998]	ZAMBONI, Diego; BALASUBRAMANIYAN, Jai; GARCIA-FERNANDES, Jose Omar and SPAFFORD, E. H.; Department of Computer Sciences, Purdue University; Coast TR 98-05; 1998. Disponível on-line em: http://www.cerias.purdue.edu/coast/projects/aafid.html, visitado em 13/01/1999.
[ZERKLE & LEVIT, 1996]	ZERKLE, Dan; LEVITT, Karl. *NetKuang - A Multi-Host Configuration Vulnerability Checker.* Departament of Computer Science. University of California at Davis. Disponível on-line em: http://seclabs.cs.ucdavis.edu/papers/zl96.ps, visitado em 13/01/1999.