SEGURANÇA EM APLICAÇÕES BASEADAS EM AGENTES MÓVEIS

Stênio Firmino Pereira Filho, Daniel Rodrigues Ambrósio, Edson dos Santos Moreira
Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Av. Trabalhador Sãocarlense, 400
13560-970 – São Carlos - SP
{stenio, daniel, edson}@icmc.sc.usp.br

Mauro César Bernardes
Universidade de Alfenas
Instituto de Engenharia e Ciências Exatas
Rodovia MG 179, km 0
37130-000 Alfenas - MG
mauro.bernardes@unifenas.br

RESUMO

Este trabalho apresenta uma discussão sobre segurança em aplicações que implementam a mobilidade de código através da tecnologia de agentes móveis. Considerando-se que em diversas situações esta tecnologia apresenta significantes vantagens em termos de delegação de tarefas, mobilidade, overhead, escalabilidade, flexibilidade e tolerância a falhas, existe atualmente diversas propostas para sua aplicação em comércio eletrônico. Uma vez que essas aplicações necessitam de um ambiente de agência instalada em cada máquina que receberá os agentes móveis provenientes de localidades diversas, elas podem ser responsáveis por graves problemas de segurança. Dessa forma, a questão da segurança relacionada aos ambientes de agência e, em especial, aos agentes móveis é de suma importância e deve ser levada em consideração no desenvolvimento de aplicações que utilizem dessa tecnologia.

1. INTRODUÇÃO

Durante anos os sistemas computacionais foram desenvolvidos para funcionarem em plataformas centralizadas. No final da década de 80, com o advento de equipamentos pessoais mais potentes, houve o crescimento da abordagem cliente/servidor, na qual as tarefas eram divididas entre a máquina do usuário e um, ou mais, equipamentos mais potentes que serviam de apoio para operações de banco de dados, processamento de dados, etc.

Já no início da década de 90, as corporações voltaram os olhos para o novo paradigma de análise, projeto e programação orientados a objetos que veio facilitar o desenvolvimento de sistemas e melhorar a possibilidade de reutilização e a qualidade de software, etc. Além disso, o desenvolvimento de aplicações distribuídas espalhou-se com a idéia de executar as tarefas nos locais mais adequados da rede; por exemplo, a interface com usuário é tratada no lado do cliente e o processamento de dados onde os dados estão.

Nos últimos anos, com a consolidação das tecnologias orientadas a objetos, novos paradigmas vêm sendo criados para aumentar a flexibilidade dos sistemas computacionais. Estes novos paradigmas envolvem a mobilidade dos objetos implementados no sistema e a execução assíncrona de tarefas. Mais especificamente têm-se pesquisado muito sobre objetos móveis e sobre softwares denominados agentes móveis (Reami, 1998).

Agentes móveis são processos capazes de "vagar" por grandes redes como a WWW, interagindo com máquinas, coletando informações e retornando após executar os deveres ajustados pelo usuário. Apesar de mobilidade não ser uma condição nem necessária, nem suficiente para o conceito de agente, agentes móveis apresentam uma série de vantagens sobre os similares estáticos, por exemplo: redução dos custos de comunicação, independência da limitação imposta pelo uso de recursos locais, coordenação mais fácil, computação assíncrona, ambiente de desenvolvimento natural para serviços de comércio, arquitetura flexível para computação distribuída e fornecem uma nova abordagem atrativa e diferente para o processo de projeto de aplicações.

Considerando-se que em diversas situações esta tecnologia apresente significantes vantagens em termos de overhead, escalabilidade, flexibilidade e tolerância a falhas, existe atualmente diversas propostas para aplicações que fazem seu uso em áreas como: comércio eletrônico, gerenciamento de redes, integração de base de dados, coleta distribuída de informação, segurança computacional, etc.

Uma das mais promissoras áreas para a aplicação de agentes móveis é o comércio eletrônico. A idéia principal reside no fato de que o usuário pode interagir com seu agente local, delegando-lhe tarefas e conectar-se temporariamente ao sistema para disparar o agente que seguirá as instruções e o itinerário designado pelo usuário. Depois de transmitido, o agente torna-se independente do processo que o criou e pode operar de forma assíncrona e autônoma no ponto remoto. O usuário poderá conectar-se depois de um tempo e coletar o agente com o resultado da tarefa que lhe foi delegada.

Entretanto, uma vez que um agente tem um comportamento semelhante a um vírus eletrônico, deve-se ponderar as vantagens de sua utilização com as falhas de segurança que poderá impor ao sistema se não for bem configurado.

O objetivo deste trabalho é apresentar uma taxionomia para as falhas de segurança que poderão surgir com a utilização da tecnologia de agentes móveis em comércio eletrônico, formas de preveni-las e uma breve discussão dos recursos de segurança oferecidos pelos principais ambientes disponíveis atualmente.

2. AGENTES

Agentes, agentes inteligentes e sistemas baseados em agentes têm atraído um considerável interesse de muitos campos da ciência da computação (Lingnau & Drobnik, 1996). A tecnologia de agentes vem sendo aplicada academicamente nos mais diversos campos, principalmente em inteligência artificial, sistemas distribuídos e engenharia de software.

Recentemente algumas aplicações comerciais vêm sendo desenvolvidas. Dentre as aplicações desta tecnologia pode-se citar apenas como exemplos: Suporte para aplicações baseadas em fluxo de trabalho ou workflow (Nicolas, 1998), Gerenciamento de redes (Rubinstein, 2000) e serviços de telecomunicações (Corley et al. 1998), Análise de informações em aplicações de data mining, Layout de circuitos eletrônicos (Moreira & Walczowski, 1997), Busca de informações em bases de dados, Segurança Computacional (Reami, 1998) (Bernardes, 2000), Comércio Eletrônico (Rodriguez, 1999) (Valdo, 2000), Computação móvel (Lingnau & Drobnik, 1996).

Segundo (Chess et al, 1995), 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". Uma característica importante de agentes móveis, que os diferenciam das demais tecnologias cliente/servidor, é que eles não estão confinados no sistema onde começam suas execuções. Agentes móveis são livres para serem executados em qualquer máquina do ambiente ao qual estão inseridos. Ao chegar em um ponto remoto, o agente pode tomar a decisão (autonomia) de migrar para um novo computador, seguindo sua lista de itinerários.

3. SEGURANÇA EM AGENTES MÓVEIS

De fato, a segurança é um dos fatores chave no progresso da tecnologia de agentes móveis (IBM, 1998). Como qualquer código executável, os agentes móveis são uma ameaça potencial ao sistema (Lange & Oshima, 1998) além de também estarem expostos às ameaças dos ambientes onde são executados. Com base nisso e na utilização de simples mecanismos de segurança de agentes, os usuários poderão não utilizá-los ou simplesmente não aceitá-los em seu computador.

Como exemplo, supõe-se que é disparado um agente com a missão de reservar uma passagem. Além da missão, poderá ser dada a ele alguma moeda eletrônica para efetivar a transação (ex. o número de um cartão de crédito). Entre os riscos, o agente poderá ser enganado. Como resultado, é possível que ele informe resultados inválidos; por exemplo, talvez o agente retorne que a reserva foi feita quando na realidade não foi. Um servidor hostil também poderá roubar e utilizar as informações confidenciais do agente e até mesmo enganá-lo ao ponto que ele chegue a danificar bases de dados do servidor remoto que está visitando. Nesse caso o agente "inocente" pode ser considerado responsável e barrado futuramente para utilizar os serviços no servidor. Estes são precisamente alguns exemplos dos riscos que os sistemas baseados em agentes móveis sofrem.

3.1. Ameaças aos agentes

Os agentes podem sofrer ataques de diversos componentes do sistema. Os servidores hostis podem atacar os agentes na tentativa de roubar informações ou alterá-las. Ex: Um agente disparado visita um servidor não confiável que pode tentar extrair suas informações privadas ou enganá-lo. Os tipos de ataques possíveis são a falsificação das informações, execução ilegal de métodos e acesso ilegal.

Da mesma maneira que um agente pode sofrer um ataque de servidores, um outro agente pode, também, tentar obstruir a sua execução. Ex: Um agente interage com outro agente na tentativa de extrair suas informações privadas e interromper sua execução. Um tipo de ataque possível é o acesso ilegal.

Já uma entidade mal intencionada poderá alterar mensagens enviadas entre os agentes ou escutar a transmissão deles tentando desvendar o seu conteúdo. Os tipos de ataque possíveis incluem alterações e a escuta de mensagens.

3.2. Ameaças aos servidores

Os servidores também podem ser ameaçados tanto por agentes quanto por entidades externas. Quando um agente ataca um servidor, ele pode tentar acessar ilegalmente ou corromper as informações, assim como pode, também tentar interromper a execução do servidor. Os tipos de ataque possíveis incluem acesso ilegal, disfarce, Cavalo de Tróia, DoS.

Entidades externas podem tentar interromper a execução do servidor enviando uma grande quantidade de agentes (spam) podendo conseguir também um DoS. Os tipos de ataque possíveis incluem DoS e Spam.

Mas não são somente os agentes e os servidores que podem sofrer ataques. A rede também pode ser vítima. Agentes podem se multiplicar e se moverem intensivamente na tentativa de congestionar a rede causando assim um DoS.

4. TAXONOMIA DE ATAQUES

Em geral, pode-se dividir os ataques aos agentes e seus servidores em duas categorias: ataques passivos, que não modificam os agentes ou informações, e ataques ativos, que fazem alguma coisa com os agentes (Lange & Oshima, 1998). Nos ataques ativos pode-se detectar que alguma coisa aconteceu no agente, enquanto que não se pode dizer o mesmo dos ataques passivos.

Os ataques passivos baseiam-se em ataques contra o sistema de comunicação dos agentes e na sua transferência em uma rede. Estes ataques são difíceis de serem detectados porque nenhuma mudança é feita no agente. Para se proteger, pode-se utilizar um método de criptografia ou nas informações contida no agente ou no próprio agente. Os dois tipos de ataques passivos são:

Escuta: Este tipo de ataque normalmente usa um programa chamado de monitor de comunicação. Um monitor permanece observando as informações enviadas entre os sistemas de agentes, através da captura de agentes e mensagens que podem conter informações úteis. Geralmente, pode-se dizer que a informação é desvendada através da monitoração das comunicações.

Análise do Tráfego: Pode parecer fácil prevenir escutas, basta utilizar agentes cifrados. Embora esta técnica possa em muitos casos ser suficiente, os ataques passivos ainda podem quebrá-la. Análise de Tráfego, o qual é uma variação diferente de "Escuta", permite ao agressor analisar os padrões dos agentes enviados entre os sistemas de agentes (como mudanças no fluxo do tráfego ou mudanças na freqüência de agentes enviados e recebidos), possivelmente permitindo-o fazer suposições baseadas nestes padrões, podendo, às vezes, ser eficaz mesmo se o conteúdo estiver cifrado.

Os ataques ativos envolvem uma grande variedade de ameaças à segurança, abrangendo desde uma simples modificação dos dados dos agentes até a inserção de agentes mal intencionados dentro de um servido. A seguir, alguns tipos de ataques ativos:

Acesso Ilegal: Um agente acessa informações que não é permitido acessar. Isto pode acontecer quando o agente se disfarça como um usuário confiável ou é executado em um ambiente de execução sem segurança. Por exemplo, se um agente é escrito na linguagem C ele pode utilizar ponteiros aritméticos para acessar arbitrariamente localizações na memória. Não somente pode recuperar os dados privados do agente, mas é capaz, também, de se juntar aos dados. Utilizando-se Java este problema poderá ser solucionado, uma vez que nessa linguagem não é permitida a manipulação de ponteiros.

Disfarces: Uma entidade pretende ser uma outra entidade diferente. Em um exemplo típico, um agente entra em um servidor aparentemente representando uma pessoa confiável ou uma organização. Se há sucesso em enganar o servidor, o agente pode estar capacitado a utilizar um serviço livre ou roubar informações confidenciais.

Cavalo de Tróia: Um Cavalo de Tróia é um agente que é executado por um usuário legítimo mas executa alguma coisa diferente do que o usuário espera ou aprova. Um hacker pode simplesmente criar e quem sabe executar um agente de busca aparentemente inocente que na realidade causa danos para os usuários.

Alteração: Um agente ou mensagem entre dois sistemas de agentes é excluído ou alterado enquanto é transmitido. Em particular, qualquer servidor visitado no itinerário do agente pode remover dados adicionados pelo servidor anterior. Qualquer informação que é modificada de uma maneira inesperada pode transformar um agente legítimo em um outro agente mal intencionado ou pode simplesmente retornar resultados falsos.

SPAM: Uma cópia capturada de um agente recentemente enviado é retransmitida com propósitos ilegítimos. Desta maneira, a entidade mal intencionada pode estar capacitada a obter resultados idênticos ou fatalmente atrapalhar o serviço. Mesmo se o agente estiver cifrado o ataque ainda é possível, porque o agressor não precisa alterar o conteúdo do agente.

DoS (Denial of Service): Um recurso é deliberadamente utilizado exaustivamente chegando ao ponto de atrapalhar o serviço aos outros usuários. O recurso em questão pode ser uma largura de banda bem como a memória do servidor ou uma CPU. Por exemplo, se um agente mal intencionado permanece alocando memória, outros agentes ou possivelmente o servidor pode ficar incapaz de operar corretamente. Um outro exemplo é o uso de agentes que criam cópias (clones). Em pouco tempo o servidor ou a rede estará cheio de agentes.

5. SEGURANÇA EM AGENTES MÓVEIS

Embora as ameaças descritas anteriormente envolvam assuntos de segurança já conhecidos, elas apresentam um conceito relativamente novo em se tratando de tecnologia de agentes móveis. Entretanto, não há

motivos para a não utilização dessa promissora tecnologia. Utilizando-se da tecnologia de segurança existente, aliada à correta configuração dos ambientes de agência pode-se resolver a maioria deles.

O framework de segurança da linguagem Java e outras linguagens de script para "programação remota" têm permitido aos desenvolvedores alguns progressos referentes à segurança dos agentes móveis. Alguns sistemas de agentes atuais oferecem mecanismos básicos de privacidade como a possibilidade de criar um canal seguro entre máquinas através da utilização de técnicas de criptografia na transmissão dos agentes. Algumas oferecem mecanismos de autenticação e verificação de integridade através de troca de agentes e mensagens entre os servidores (mais uma vez utilizando-se de técnicas de criptografia). Outros ambientes permitem o controle de utilização de recursos pelos agentes móveis.

5.1. Serviços de Segurança

Os serviços de segurança são importantes na proteção dos agentes e servidores contra os ataques. Abaixo, segue a lista dos serviços de segurança mais comuns disponíveis para agentes e que um servidor de agentes deve oferecer para obter um sistema mais seguro:

Serviço de Autenticação: Antes de aceitar um agente em um sistema, deve-se conhecer quem o enviou. Neste caso, é necessária a autenticação do agente. Esse processo inclui a verificação da entidade que desenvolveu (programou) o agente e a entidade que instanciou e o enviou ao servidor. Antes de enviar um agente, pode haver a necessidade de verificar se o servidor de destino é sem dúvida o servidor que ele disse que é. Em uma aplicação baseada em agentes móveis, deve haver as seguintes autenticações:

- **Autenticação do usuário:** O usuário precisa se autenticar em um servidor qualquer. Uma criptografia com chave pública ou uma senha pode ser usada para este propósito.
- **Autenticação do servidor:** Antes de um servidor iniciar uma comunicação com outro servidor ou cliente, é necessário que o servidor conheça com quem ele está comunicando. Isso é importante porque não se pode assegurar a integridade e a confidencialidade sem conhecer de quem está recebendo os agentes ou enviando-os.
- **Autenticação de código:** Antes de executar um agente recebido o servidor precisa conhecer quem é responsável pela a implementação do agente. Assinaturas digitais são tipicamente utilizadas para esse propósito.
- **Autenticação do agente:** Antes de executar um agente recebido o servidor precisa conhecer quem é responsável pelo agente.

Serviço de Garantida de Integridade: Para confiar no agente é necessário certificar-se que ele não foi alterado. Checar a integridade do agente é a técnica utilizada para verificar se alterações ilegítimas foram feitas no seu estado e no seu código. O uso de bytes de paridade ou outros métodos de detecção de erros como o CRC, podem ser muito úteis neste caso.

Serviço de Confidencialidade de Informações: Um agente pode carregar uma informação confidencial mantendo-a ilegível de outros servidores ou agentes, podendo ser legível somente por servidores e agentes específicos. Um agente pode requisitar a um servidor para transportá-lo de uma maneira secreta, através do uso da criptografia, para enfrentar uma ameaça de escuta.

Serviço de Autorização: Um agente recebido deveria conceder direitos de acesso às informações de acordo com o seu gerenciador. A autorização, ou controle de acesso é a maneira de especificar e gerenciar as capacidades de acessar informações ou utilizar serviços oferecidos pelo servidor.

Serviço de Auditoria: Um serviço de auditoria grava atividades relacionadas a segurança de um agente para uma inspeção mais adiante. Os logs gerados pelo serviço podem ser muito úteis para a identificação dos métodos utilizados no ataque ou até mesmo para a recuperação do estado de execução do servidor e seus agentes para dar continuidade às tarefas.

5.2. Aplicando os serviços de segurança

É essencial estar prevenido de que há limites inerentes à segurança dos agentes. A primeira ameaça que os agentes encontram é quando eles são transferidos através da rede. Fatores externos não confiáveis podem também alterar o conteúdo dos agentes, levando aos comportamentos indesejados e inesperados dos agentes. Felizmente, este tipo de ataque é fácil de tratar e pode ser resolvido utilizando técnicas clássicas. Uma conexão segura, como SSL (Secure Sockets Layer) - que se baseia na criptografia dos dados e a checagem de suas integridade pode ser utilizada para efetivamente prevenir este tipo de ataque (IBM, 98).

A próxima ameaça é o servidor remoto que o agente visita. Quando é enviando um agente para um servidor remoto, é esperado que o servidor trabalhe com o agente adequadamente e que sirva qualquer requisição válida feita pelo agente.

O agente está totalmente nas mãos do servidor remoto. Embora freqüentemente seja afirmado que agentes são autônomos, eles não são processadores independentes que são capazes de executar seus próprios

programas. O servidor é totalmente responsável pela execução do agente e também pode dar informações erradas ao agente, na tentativa de enganá-lo. Até a transmissão do agente não é segura. Um servidor pode proibir o agente de viajar para um servidor concorrente, em vez disso, o envia para um servidor conspirador.

Toda a informação necessária para o processamento de um agente deve ser legível ao servidor. O servidor tem potencial para roubar, alterar, ou revelar qualquer informação do agente, por exemplo, se o agente carrega uma chave de acesso a um servidor específico e ela deve ser mantida em segredo dos outros servidores que o agente pode visitar. Para isto é necessária a criptografia da chave de acesso com a chave pública do servidor. Assim, somente o servidor que possui a chave pública poderá decodificar. A criptografia deve ser feita com a ajuda do servidor. Outros servidores não têm nenhuma maneira de ler o dado.

6. SEGURANÇA EM AMBIENTES SERVIDORES

Existe atualmente diversas plataformas de desenvolvimento para agentes móveis. Comerciais ou acadêmicos, elas oferecem alguns serviços de segurança que visam proteger a integridade das informações e a segurança da aplicação. Entre as principais plataformas existentes, pode-se citar o ASDK da IBM, Concordia da Mitsubishi, Grasshopper da IKV e Gypsy da Universidade de Viena. A seguir serão apresentados as plataformas e os serviços de segurança que elas oferecem.

ASDK: Aglets Software Development Kit é um conjunto de APIs, desenvolvidas pela IBM, que permitem desenvolver aglets. Os aglets são objetos Java que podem mover-se de um host para outro na rede. Em cada host deve haver o servidor de aglets denominado Tahiti. Os serviços disponíveis no Tahiti da versão 1.0 são autenticações de código, agentes e autorização. Os aglets possuem informações que auxiliam na sua identificação e na identificação do seu criador. Os aglets são identificados pelo ID e pelo code base (que é o seu código pré-compilado), já o seu criador é identificado pelo username e e-mail. Os aglets são classificados em confiáveis (trusted) ou não-confiáveis (untrusted). Para ser considerado confiável, o aglet deve estar armazenado localmente no servidor, ou seja, o seu code base deve estar armazenado no hard disk local. Todos os outros aglets, os que vêm de servidores remotos, são tratados como não-confiáveis. No servidor Tahiti do ASDK 1.0 pode-se especificar se os aglets confiáveis ou não-confiáveis estão autorizados em acessar arquivos e bibliotecas, estabelecer conexões via sockets, utilizar RMI, acessar banco de dados, criar janelas e executar comandos no sistema. A versão 1.1 possui, além de todos os serviços acima, serviços de autenticação de usuários e de domínios, serviços de garantia de integridade das informações e a adição de mais restrições no serviço de autorização (IBM, 1998).

Concordia: Concordia é um framework, desenvolvido pela Mitsubishi Eletric Information Center, para o desenvolvimento e gerenciamento de agentes móveis podendo ser utilizado em qualquer dispositivo que suporte Java. O framework é composto por APIs e um servidor para agentes móveis. O Concordia Server oferece serviços de identificação dos usuários, autenticação de agentes, garantia de integridade de agente e seus dados, autorização de classes Java dinamicamente carregadas para satisfazerem as necessidades do agente. Oferece suporte a criptografia e utiliza credenciais que possuem a identidade do usuário, importantes para sistemas que utilizam uma rede pública (Mitsubishi, 97).

Grasshopper: O ambiente Grasshopper é o primeiro a ser desenvolvido seguindo os padrões MASIF[1]. Desenvolvido pela IKV, Grasshopper oferece um suporte a segurança e facilidades para gerenciamento e a proteção oferecida é aplicável aos agentes e ao ambiente (contra agentes maliciosos). Outras capacidades essenciais do ambiente são a extensibilidade e integração das tecnologias existentes como Java e CORBA (Common Object Request Broker Architecture). Grasshopper é completamente implementado em Java. Ele oferece mecanismos de segurança externa que permite a criptografia dos agentes (código e estado) durante a transmissão, utilizando o protocolo SSL (Secure Socket Layer). Se necessário, as interações remotas entre entidades Grasshopper (agentes, agências, componentes) podem ser protegidas também via SSL. Por meios de certificados, uma agência pode identificar e autenticar um agente. Cada agência registra todos os agentes que estão em execução localmente a fim de monitorar e controlar todos os seus processos internos e a fim de controlar a interação entre agentes (IKV, 99a).

Gypsy: Gypsy é uma versão acadêmica de plataforma de agentes móveis criada por acadêmicos da Universidade de Viena. Na plataforma Gypsy, um agente móvel é um objeto Java e é executado como uma thread dedicada em places especiais. Como suporte à segurança, Gypsy oferece proteção aos hosts contra agentes maliciosos, proteção aos agentes contra outros agentes e proteção à rede de comunicação. Para a proteção dos host é utilizada a técnica de assinatura de código, onde cada código possui uma assinatura que é armazenada em uma base de dados que é consultada pelos servidores a fim de verificar se o código é confiável ou não. Para os agentes a proteção é feita através da criptografia por chave pública (RSA) ou criptografia por chave privada

[1] MASIF (Mobile Agent System Interoperability Facility): padrão de interoperabilidade para agentes móveis, definido em fevereiro de 1998 pela OMG (Object Management Group).

(IDA). Os agentes Gypsy não podem ser clonados e o ciclo de vida padrão é de 24 horas (mas pode ser alterado de acordo com as necessidades), desse jeito pode-se proteger a rede de comunicação evitando que ela se congestione (Jazayeri & Lugmayr, 99).

7. CONCLUSÃO

Com o impulso dado pela Internet para a ascensão de sistemas empresariais e o seu alcance em proporções globalizadas, fez nascer a necessidade de tecnologias independentes de plataformas que viabilizassem toda a infra-estrutura na transação entre usuário e empresa. Entre essas tecnologias surge uma das mais promissoras áreas para a aplicação da tecnologia de agentes móveis. Isso se deve principalmente ao fato de que essa tecnologia ainda apresenta características como reatividade, delegação de tarefas e execução assíncrona, além de minimizar o overhead, possuir grande flexibilidade e tolerância à falhas.

Entretanto, a utilização da tecnologia de agentes móveis sem levar em consideração aspectos de segurança computacional pode acarretar em grandes catástrofes, ao mesmo tempo em que poderia desmotivar sua utilização.

8. REFERÊNCIAS BIBLIOGRÁFICAS

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, Mar/2000

BERNARDES, M.C.& MOREIRA, E.S. Implementation of an Intrusion Detection System Based on Mobile Agents. Trabalho aceito para publicação e apresentação no 5th International Symposium on Software Engineering for Parallel and Distributed Systems and 22nd International Conference on Software Engineering (ICSE2000). June 10-11-2000, Limerick, Ireland. Patrocínio: IEEE

CHESS, David; HARRISON, Colin; KERSHENBAUM, Aaron. *Mobile Agents: 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.

CORLEY, S.; TESSELAAR, M.; COOLEY, J.; MEINKHN, J.; MALABOCCHIA, F.; GARIJO, F. *The Application of Intelligent and Mobile Agents to Network and Service Management.* Fifth International Conference On Intelligence in Services and Networks. Antuerpia, Bélgica, Maio 1998. Disponível on-line em: http://www.alcatel.be/telecom/news/isn98.

HAZELTON, K. *E-Commerce: A White Paper*. IT Architect University of Wisconsin-Madison.IT Architecture Department.1998. Disponível on-line em http://www.wisc.edu/arch/teams/ecommerce/white_paper.html

IBM. *Aglets Software Development Kit*. Disponível on-line em http://www.trl.ibm.co.jp/aglets. Visitado em 10/11/1999.

IKV++, *Grasshopper Site.* Disponível on-line em http://www.ikv.de/products/grasshopper. Visitado em 13/02/2000.

JAZAYERI, M. & LUGMAYR, W. *Gypsy: A Component-Oriented Mobile Agent System*, Technischen Universität Wien, 1999.

LANGE, D.B; OSHIMA, M. *Programming And Deploying Java Mobile Agents with Aglets.* Addison Wesley Longman, Inc. 1998

LINGNAU, Anselm; DROBNIK, Oswald. *An Infraestructure for Mobile Agentes: Requeriments and Architecture*. Frankfurt am Main, Germany. http://www.tm.informatik.uni-frankfurt.de/ma/paper.html. Visitado em 28/01/1999.

MOREIRA, D.A.; WALCZOWSKI, L.T. *Using Software Agents to Generate VLSI Layouts*. IEEE Expert. p26-32. November-December 1997.

MITSUBISHI Eletric Information Center. *Concordia - Java mobile Agent Technology*. Disponível on-line em http://www.meitca.com/HSL/Projects/condordia visitado em 11/11/1999.

NICOLAS, P.R. *Agent-based Workflow Automation*. In: WWW: White Paper: Agent-based Workflow Automation, http://www.ikonodyne.com/whitewkflow/agents.html. Ikonodyne Inc. 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.

RODRIGUEZ, Eduardo José. *Uma Modelagem para Comércio Eletrônico usando Corba e Agentes Móveis.* Campinas, 1999, 83p. Dissertação (Mestrado) – Instituto de Computação. Universidade Estadual de Campinas.

VALDO, C. A.; SOBRAL, J.B.M. *Integração de e-Commerce e ERP através de Agentes Móveis.* Anais do XXVII SEMISH, Curitiba, 2000.