Implementation of an Intrusion Detection System Based on Mobile Agents

Mauro Cesar Bernardes
University of Alfenas
Institute of Engineering and Exact Sciences
Rod. MG 179, Km 0
Alfenas Campus – Caixa Postal, 23
CEP 37130-000 Alfenas, MG, Brazil
E-mail: mcesar@unifenas.br

Edson dos Santos Moreira
Department of Computation and Statistics
Institute of Mathematics and Computer Sciences
USP – University of São Paulo
São Carlos Campus - Caixa Postal 668
CEP 13560-970 São Carlos, SP, Brazil
E-mail: edson@icmc.sc.usp.br

Abstract

The number of security-breaking attempts originated inside the organizations are increasing steadily. Attacks made in this way, usually done by "authorized" users of the system, cannot be immediately located. As the idea of filtering the traffic at the "entrance door" (by firewalls, for instance) is not completely successful, the use of other technologies should be considered to increase the defense capacity of a site. Therefore, the introduction of mobile agents to provide the computational security by constantly moving around within the internal infoways of an organization is presented as a natural solution to prevent both external and the internal sources of intrusion. This work presents an evaluation of the use of mobile agents mechanisms to add mobility features to the process of monitoring intrusion in computational systems. A modular approach is proposed, where independent small agents will monitor the communication paths. This approach presents significant advantages in terms of minimizing overhead, increasing scalability and flexibility and providing fault tolerance.

1. Introduction

It is very well known the value of information as the vital source of knowledge and, thus, of power. The acquisition, maintenance and dissemination of information have become major concerns of society and have led to the increasing expansion of forms of storage and distribution through computer networks.

On the other hand, attempted attacks and successful invasions involving an increasing number of computers have also become frequent. Thus, security has become a key word for most companies worldwide. With the growing use of Internet technology in the corporate environment (*Intranet*) and its opening to the outside world (*Extranet*) for essential activities, concern regarding the risk of invasion of computer systems has increased tremendously in recent years. This concern is not surprising: digital crimes are generally very difficult to discover or even to trace. Proof of this is that many companies only discover that their systems have been invaded long after the fact occurs.

This state of affairs, however, is changing rapidly. Every day brings new, stronger solutions for data protection. Firewalls, cryptography of a variety of bits, digital certificates, VPNs (Virtual Private Networks), smart cards, biometry and IDSs (Intrusion Detection Systems) are already all part of the arsenal used to combat systems violations and the credibility of on-line transactions.

The most widely employed network security technology today is the firewall [11], a system that prevents unauthorized entry using outside access control mechanisms. However, no system yet exists that can be considered a panacea for protection, providing a high level of security while allowing for a certain degree of flexibility and freedom of the use of computational resources.

Certain factors make it extremely difficult to prevent attackers from possibly accessing a system. Most computers have some kind of "security loophole" that enables outside attackers (or even legitimate internal users) from accessing confidential information. Even a supposedly secure system can be vulnerable to internal users who abuse their privileges or to jeopardy from improper practices. Since attacks are apparently inevitable, there is an obvious need for mechanisms to detect attackers attempting to penetrate the system or legitimate users abusing their privileges, preferably at the moment such attacks occur.

Hence, the constantly growing number of internal attacks requires the increasing use of mechanisms such as the firewall. Because this type of attack by a system's users makes immediate location difficult, there is a need for the integrated use of several technologies to increase a site's capacity for defense. Among these technologies, it

0-7695-0634-8/00 $10.00 © 2000 IEEE

is desirable to have mechanisms that add mobility to the system monitoring process. Thus, the introduction of mobile agents to support computational security appear as a natural solution, allowing for the distribution of a system's monitoring tasks and speeding up the decision-making process in the absence of a human system's manager.

2. Intrusion Detection System Based on Mobile Agents

The degree of protection against every malicious action is directly related to the time and effort spent in constructing and managing security systems. These actions can be identified at the moment they occur using complex tools to continuously monitor and warn of suspect activities; however, this involves high costs in terms of time and money to build and manage monitoring systems. These systems also cause losses in the performance of a protected environment, which may lead to their rejection by users.

The monolithic architecture of Intrusion Detection Systems (IDSs) commonly used in commercial or research systems contain a number of problems that limit their configuration capacity, scalability of efficiency [2, 3, 12].

This section presents a model for the development of a non-monolithic IDS based on mobile agents.

Agents, Intelligent Agents and agent-based systems have attracted considerable interest from many fields of computer science [8]. Agent technology has been academically applied in a variety of fields, particularly in artificial intelligence, distributed systems, software engineering and electronic commerce. In general terms, an agent can be defined as a software program capable of executing a complex task on behalf of a user.

Mobile agents are a special type of agents defined as *"processes capable of "roaming" through large networks such as the World Wide Web, interacting with machines, collecting information and returning after executing the tasks adjusted by the user"* [9]. Although mobility is neither a necessary nor a sufficient condition to define the concept 'agent', mobile agents offer a series of advantages over similar static systems. These advantages include communications cost reductions, independence from the limitations imposed by the use of local resources, easier coordination, asynchronous computation, a natural environment for the development of electronic commerce, flexible architecture for distributed computation. They also provide an attractive and radically different approach to the applications design process. Furthermore, these elements have characteristics that are inherent to the concept of multiagents, providing good performance in distributed object systems. Thus, cooperation, autonomy and representativeness are characteristics inherited from their own origins, to which several others are added to meet the requirements of good performance of models that use this paradigm [5].

Mobile agents, therefore, constitute an interesting mechanism for the development of a non-monolithic IDS.

2.1. Advantages of a non-monolithic IDS

The monolithic approach for Intrusion Detection Systems presents several practical problems. Whenever a new form of intrusion unforeseen in the system is discovered, the monolithic IDS has to be completely rebuilt to deal with it, which is certainly not a simple task [2, 3, 12].

Another concern involves failure tolerance, since a monolithic system presents itself as a single point of failure and attack. Hence, well known attack methodologies (such as *Denial of Service*), when made against a machine hosting an IDS, completely compromise the system's integrity.

The use of autonomous agents has been proposed by some authors as a form of constructing non-monolithic intrusion detection systems [2, 3, 12]. The capacity of some autonomous agents to maintain specific information of its application domain, in this case, security application, confers great flexibility on these agents and, hence, on the entire system.

Instead of a large monolithic module, this study presents a proposal for a modular approach based on autonomous mobile agents for the development of an IDS. This IDS consists of a set of small processes (agents) that can act independently within the environment under construction. These agents will be developed to move through the environment into which they are inserted, observing the behavior of the system and the users logged into it, cooperating with each other via messages, advising each other when an action is considered suspect and engaging in reactive actions (counter-attack).

Each agent observes only a small aspect of the entire system. A simple agent, by itself, cannot form an intrusion detection system, since its vision is limited to a small "slice" of the system. However, if many agents operate within a system and cooperate with each other, then a powerful IDS can be developed. Because the agents are independent of each other, they can be added to or removed from the system dynamically, precluding the need to reconstruct the entire IDS or even to interrupt its activities. Thus, whenever any sign of a new form of attack is identified, new specialized agents can be developed, added to the system and configured to meet a specific security policy.

Another advantage of the above described approach is the system's easy configuration to meet the policy needs of the environment into which it is inserted. This is an important feature, since what is considered a breach of

security in a given environment may not be a breach in another, depending on the type of information that is to be protected and the security policy adopted.

Because change is subjacent to all software design and is an inevitable factor in the construction of computer based systems, another advantage of this system is its high maintainability. Defined as being "the feature through which a software program can be understood, corrected and/or increased" [10], maintainability is the foremost goal guiding the steps of a software engineering process.

Divided into modules containing a set of small, and hence, less logically complex agents specialized in a single function, the system seeks to minimize efforts spent on maintainability. Thus, each agent presents a structure that is easily understood and maintained. This is directly reflected in terms of:

- Time taken to recognize a problem;
- Time taken to analyze the problem;
- Time taken to prepare specifications for changes;
- Time taken for active correction (or modification);
- Time taken for local tests;
- Time taken for global tests;
- Time taken to review maintenance;
- Time taken for total recovery.

In addition to the above, non-monolithic systems based on autonomous mobile agents offer several advantages over monolithic systems [2, 3]. Among these are:

- **Easy configuration**: since it is possible to have a series of small agents specialized in specific detection tasks, the detection system can be configured more suitably for each case, with easy addition or removal of the system's agents.
- **Efficiency**: agents can be trained previously and optimized to carry out their tasks in such a way as to generate as little overloading of the system as possible.
- **Extension capacity**: a system of agents can be easily modified to operate in a network and to allow for migration to track anomalous behavior throughout the network, or to move to machines where they may be more useful.
- **Resistance to subversion**: if a defense system is subverted, it may provide a false sense of security. However, this is unlikely to occur because, since agents carry out different functions, the knowledge acquired by one agent does not include knowledge about the operation of others.
- **Scalability**: to operate in large systems, all that is required is to add more agents and increase their diversity.

2.2. Architecture of the Proposed System

The main concept of the autonomous mobile agent-based IDS is simplicity. Each agent is a simple entity that carries out a specific activity and cooperates with other agents as efficiently as possible. When an agent considers an activity suspect, it immediately advises other agents of the system of the suspected intrusion. At that moment, an agent (or a group of agents) with a higher level of specialization for that type of suspected intrusion is activated.

An agent may, naturally, make a mistake, which is then identified by a more highly specialized agent. Once a larger number of agents suspects a possible intrusion, a message can be sent requesting the intervention of a human operator and reaction agents can be activated.

This demonstrates that decisions must be made jointly. No single agent has the authority to identify an intrusion by itself. This decision has to be made based on a consensus of several agents in the system. If only one agent suspects an intrusion, it can be ignored after the remaining agents involved in that suspicion take a vote. However, if more than one agent suspect the occurrence of anomalous behavior, then there is a greater probability of a potential intrusion, in which case the decision can be reached of communicating with a human operator or activating specialized counter attack agents. It is clear that, in this scheme, certain events may be more "important" than others. For instance, fifty failed attempts to log on as a root will be more highly suspect than an FTP connection outside the monitored domain.

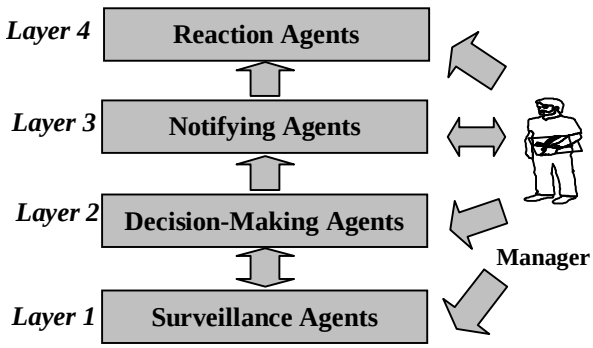

Figure 1. Layer modeling for the proposed system

The figure 1 presents the architecture (layered model) for the proposed IDS. The layers are numbered, starting from the Surveillance layer (layer 1), and each layer represents a group of specific tasks performed by agents specialized in the functions of that layer. By means of the message exchange mechanism, an agent in a layer activates one or more agents in an upper layer. In other words, layer N uses the services of layer N-1, performs its tasks and provides services to layer N+1.

Based on information collected by *Surveillance Agents*, *Decision-Making Agents* go into action, analyzing and identifying possible intrusions. If these agents consider an action suspect, they activate *Notification Agents*, which then either notify the network manager (by

e-mail, pager, phone call, alarm, etc.) or activate the agents of the upper level. On the last, top level are the *Reaction Agents*, which automatically counter-attack any possible intrusion based on the information they receive from the notifying agents, or are activated by an intervention of the network manager.

Although the above described scenario exemplifies *bottom-up* communication through the layers of the proposed architecture, *top-down* communication is also possible between the decision-making and surveillance layers. For example, let's imagine a scenario in which a Decision-Making Agent, after receiving a message or a set of data from the Surveillance Agents, requires more information to make its analysis. At this point, new Surveillance Agents may be activated and more information collected in an attempt to reach a decision based on a greater degree of certainty.

The expansion of an IDS to meet a new attack configuration may involve the development and resulting addition of new agents in a single layer or even the creation of a new scenario involving the addition of agents in every layer. The functions of each of these layers is discussed in more detail below.

2.2.1. Layer 1 – Surveillance Agents

This layer represents the first level of agents of the proposed system and is responsible for monitoring, collecting information, testing the environment and adjusting the setting based on security files strategically allocated in the environment to be protected.

An analogy of the Surveillance Agents would be to compare them to night guards in a company. These, instead of being statically allocated at strategic points in the environment, would be responsible for making the "rounds" of the system in search of open doors or patterns characterizing possible intrusions.

Among the agents of this layer there is one agent responsible for accompanying and monitoring the actions of a user considered suspect, even when the user migrates from one piece of equipment to another within the system under surveillance.

2.2.2. Layer 2 – Decision-Making Agents

This layer consists of the agents that perform the system's decision-making functions and constitute its "brain". An agent in this layer receives a message or a set of data from the agents of the layer below it (the Surveillance layer) and, based on a careful analysis of this information, it either identifies an intrusion (or attempted intrusion) at the moment it occurs or activates new Surveillance Agents to collect additional information.

When the actions are relatively simple, these agents can identify an anomaly or improper use simply by comparing the data obtained with patterns of use of the system (usage profiles). However, because this layer represents the system's intelligence, it requires the implementation of agents with artificial intelligence in order to achieve a good level of recognition of improper actions.

Among these characteristics it is desirable that these agents be capable of learning new things and adapting to new situations, rather than simply carrying out the tasks that have been allocated to them. Among the learning activation stimuli that should be used are the actions taken by the manager upon being notified by the agents from the upper level (the notification layer). Thus, there is evidently need for agents to be developed that perform functions of specialized systems, which can be done using the modern techniques of neural networks and genetic algorithms commonly referred to in reports of studies relating to Artificial Intelligence.

2.2.3. Layer 3 – Notification Agents

The agents of this layer are responsible for notifying the network manager and for activating the agents of layer 4 (Reaction Agents), based on messages received from layer 2 (Decision-Making Agents). Thus, whenever the Decision-Making Agents identify a level of danger above the acceptable limit or the need to update some new identified pattern, the Notification Agents will be activated.

One might think, in principle, that the agents of this layer perform very elementary functions, which would justify aggregating their functions to layer 4 (thus eliminating layer 3). However, a decision taken in layer 2 may require several forms of notification, occasioning the construction of a very complex agent, whether its functions are aggregated to this layer or to an upper layer. This would go against the proposed model of small agents performing specific functions in the attempt to minimize the degradation of the environment and reinforce the advantages of an intrusion detection system based on small modules that cooperate with each other.

2.2.4. Layer 4 – Reaction Agents

This layer consists of a set of agents that are activated by the layer 3 agents (Notification Agents) or yet, by direct action from a human manager. Responsible for reacting (counter attacking), recovering and reconfiguring the system, these agents represent the last instance of resources of the modeled system. Among the functions of this layer's agents are the canceling of a user's connection, blocking a user's account, reconfiguration and recovery services, the generation of log files, etc.

As a reaction example, the agent of this layer can exercise an interaction with the firewall or operating

system requesting the suspension and consequent blockade of the connection.

2.3. Classification of the Proposed Architecture

The principal methods of classification for IDSs are expressed in terms of how the system deals with the problem of detecting intrusions and in terms of how it deals with data [1,2,3].

Once the anomaly detection identifies intrusive activities as being a sub-set that does not fit normal activity patterns, a system developed according to the proposed architecture will have a set of agents that try to quantify normal or acceptable behavior, storing it in user profiles and later identifying other, irregular behaviors as intrusive. However, the system will have agents that look for attacks that can be precisely identified by the way they occur, i.e., intrusions that follow a well-defined pattern of attack (attack signatures), and these are characteristics of the model of improper usage detection (abuse).

With regard to the way it approaches the problem of intrusion detection, the proposed architecture presents itself as a hybrid between the anomaly detection model and the detection model based on improper usage. This can also be considered a significant advantage, since monolithic hybrid systems are complex, implying severe performance penalties on the environment to be monitored, which is not the case with the modular proposal.

In terms of data treatment, the architecture is a hybrid of a host-based model and a network-based model. The characteristics of a host-based system include a set of agents that look for deviations from standard behavior based on the profiles of usage of a piece of equipment, using statistical models or specialized systems. However, host-based systems also have agents that monitor network traffic, capture packages and search for the "fingerprints" of an attack in real time.

Thus, the hybrid nature of the proposed architecture allows it to make use of the advantages of each classification methodology, contributing to the development of a robust and efficient intrusion detection system.

3. Modeling of the Execution Scenarios

The modeling for implementation of the Anomalous User Identification scenario is presented below to illustrate the communication that occurs between the agents through the layers of the proposed architecture. Because it is simple, easily understood and widely used to model information systems, a logical tool known as the Data Flow Diagram (DFD) was used to model this scenario [6, 7]. Since the symbols used in the DFD are not physical, it shows the essence of the subjacent logic of the system to be modeled.

This scenario is represented by a high level model composed of 4 processes that are associated, respectively, to each of the layers of the model shown in figure 2. Thus, the numbering used in each process, in addition to identifying the logical sequence of execution of the scenario, associates the process to its respective layer in the architecture presented in figure 1.

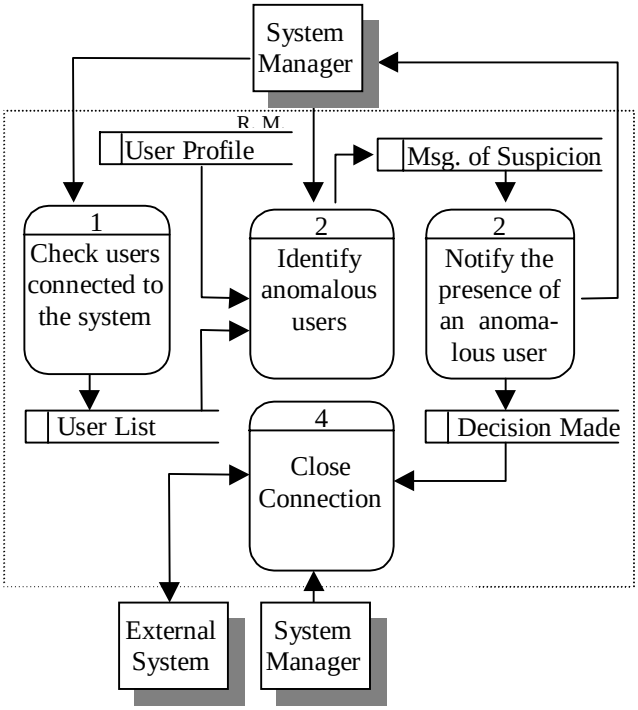

Figure 2. Modeling of the scenario of anomalous user identification

Modeling scenarios to be implemented using the DFD, in addition to clearly representing the communication that occurs among the scenario's various agents, is a very useful way to gain a better understanding of the individual functions of each agent and its relationship with the scenario.

Depending on its complexity, a process (initially seen as an agent) in this high level model can be expanded to become a new diagram (set of agents). Each scenario will represent an automation frontier for the performance of a computer security function in the proposed system. Hence, the creation of a new execution scenario will correspond to the modeling, implementation and subsequent insertion of new functions in the proposed system.

4. Implementation

For implementation of the above scenario, *Aglets Software Development Kit (ASDK),* an API Java Aglet (J-

AAPI) developed by *IBM Tokyo Research Laboratory* was used. Aglet is a mobile Java agent which supports the concepts of autonomous execution and dynamic routing in its itinerary. The Aglet term represents a combination of the words Agent and Applet. This kit can be found for download in the web site of IBM Japan (*http://www.trl.ibm.co.jp/aglets*). ASDK includes the package API Aglet, documentation, examples of Aglets and the aglets' server called Tahiti. Tahiti is a Java application which allows the user receiving, managing and sending out of Aglets for other computers that also are executing Tahiti.

4.1. Surveillance Layer

This layer, represented by process 1 in the model of figure 2, is composed by two agents. The first one, is a static agent (figure 3) whose function is to serve as user's interface and to supply the configurations for execution of the remaining agents of the scenario.

Figure 3. Communication's Interface of the mobile agent responsible for the acquisition of information from the users connected to the system

The second agent has the function of travelling in the machines that will be monitored, following the list of itineraries supplied by the agent of control or the address supplied for the administrator from the activation of the interface. After visiting a destination machine and capturing information of the users connected to the system, it will return and activate the decision taking agent sending these information in a message.

4.2. Decision Making Layer (DML)

When DML is activated by the monitoring agent, the decision-making agent (process 2 of figure 2), based on the information collected and the configuration's information of users' profile, will look for possible anomalous users. On the suspicion of a possible anomaly, this agent will invoke the notification agents of the upper layer.

In the user' profile, the information are formatted like: users#time allowed (beginning-ending)# valid origins for connection# monitoring status (0:result ignored; 1:time ignored; 2: origin ignored, 3: time and origin considered).

4.3. Notification Layer

In this scenario, this layer is made off two agents. These agents, based on the information received from the agent of decision-making, will notify the network administrator about the suspected user. The notification could be made by email (figure 4) and/or by the machine console where the administrator is connected. Next, these agents triggers the reaction agents.

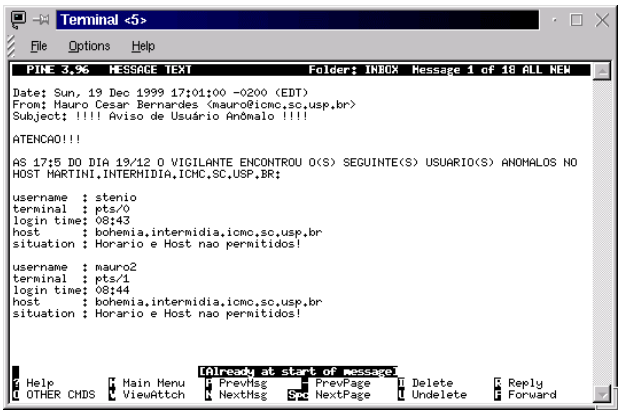

Figure 4. Notification by e-mail

4.4. Reaction Layer (counter-attack)

This layer has only one agent with the function of blocking and closing the connection for a user considered anomalous. However, before blocking up the connection, this agent will create a log file with information on the active processes for that user at that moment.

5. Conclusion

With the ever more widespread use of computer networks and the consequent opening to the outside world, computer systems have become increasingly difficult to protect. Therefore, one of the major concerns of these systems' managers has been to create barriers against outside invaders.

Recent research, however, has demonstrated that approximately 70% of attacks originate inside the organizations themselves and are made by inside users[4]. The remaining attacks, originating outside the organization, generally come through the Internet.

The creation of protection barriers against the outside world, as in the case of firewalls, is not effective. IDSs

must be in place and constantly monitoring, searching for information to identify not only outside attackers but also inside users intentionally or accidentally abusing their privileges.

The major obstacle for the detection of an internal intrusion is that it is mobile, coming from several points in the internal network. In this case, the traditional Intrusion Detection Systems (host-based, network-based or hybrid) have to be allocated to several strategic points to identify an intrusion or attempted attack at the moment it occurs.

This paper present an architecture and model of a scenario for the development of an intrusion detection system based on mobile agents. This architecture aims to minimize the costs involved in a monolithic IDS. It consists of the use of a large number of small mobile agents to perform all the tasks of monitoring, decision-making, notification and reaction to attempted intrusions. Each agent operates independently from the others; however, they all cooperate in monitoring the system, forming a complex IDS.

Based on the implementation of the presented scenario, it was verified that this approach presents significant advantages in terms of overhead, scalability and flexibility. The measurement of these parameters are going to be the subject of the further work planned for this project.

6. References

[1] A. M. Cansian,, *"Desenvolvimento de um Sistema Adaptativo de Detecção de Intrusos em Redes de Computadores"*, Tese apresentada ao Instituto de Física de São Carlos – USP, para obtenção do título de doutor em Física Aplicada, sub-área Física Computacional, 1997.

[2] M. Crosbie, and E.H. Spafford, *"Defending a Computer System using Autonomous Agents"*, Technical Report CSD-TR-95-022, Coast TR 95-02. Department of Computer Sciences, Purdue University, 1995. URL: http://www.cs.purdue.edu/homes/spaf/tech-reps/9522.ps (15/01/1999).

[3] M. Crosbie, and E. H. Spafford, *"Active Defense of a Computer System using Autonomous Agents"*, Technical Report CSD-TR-95-008, Department of Computer Sciences, Purdue University, 1995. URL: http://www.cs.purdue.edu/homes/spaf/tech-reps/9508.ps (15/01/1999).

[4] CSI/FBI. *"Issues and Trends: 1999 CSI/FBI Computer Crime and Security Survey"*, 1999. URL: http://www.gocsi.com (05/08/1999).

[5] D. Chess, C. Harrison, and A. Kershenbaum, *"Mobile Agentes: Are They a Good Idea?"*, IBM Research Report, 1998 http://www.research.ibm.com/iagentes/paps/mobile-idea.ps. (15/01/1999).

[6] C. Gane, *"Desenvolvimento rápido de sistemas"*, LTC-Livros Técnicos e Científicos Editora, Rio de Janeiro, 1988

[7] C. Gane, and T. Sarson, *"Análise Estruturada de Sistemas"*, LTC-Livros Técnicos e Científicos Editora, Rio de Janeiro, 1994.

[8] A. Lingnau, and O. Drobnik,. *"An Infraestructure for Mobile Agentes:* Requeriments *and Architecture"*, Frankfurt am Main, Germany. URL: http://www.tm.informatik.uni-frankfurt.de/ma/paper.html (28/01/1999).

[9] H. S. Nwana, *"Software Agents: An Overview"*, In: Knowledge Engineering Review, vol. 11, no. 3, p205-244, Outubro/Novembro 1996.

[10] R. Pressman, *"Software Engineering: A Practitioner's Approach"* 4th edition, McGraw Hill, 1996, 816 p.

[11] A. S. Tanembaum, *"Computer Networks"*, 3th edition. Prentice-Hall, Inc, 1997, 923 p.

[12] D. Zamboni, J. Balasubramaniyan, J. O. Garcia Fernandes, and E. H. Spafford, *"An Architecture for Intrusion Detection using Autonomous Agents"*, Department of Computer Sciences, Purdue University; 1998. URL:http://www.cerias.purdue.edu/coast/projects/aafid.html (13/01/1999).