HPOVUA’2000, June 12-14 , 2000, Santorini - Greece

# An Intelligent Agent Approach for Security Management

*K. Boudaoud*
*Institut EURECOM*
*B.P. 193*
*06904 Sophia-Antipolis France*
*Phone: (33) 4 93 00 26 38*
*Fax: (33) 4 93 00 26 27*
Karima.Boudaoud@eurecom.fr

*Noria Foukia*
*University of Geneva (Centre Universitaire d'Informatique-*
*Teleinformatics and Operating Systems Group)*
*Phone : (41) 22 705 76 41*
*Fax : (41) 22 705 77 80*
*noria.foukia@cui.unige.ch*

*Z. Guessoum*
*LIP6-OASIS*
*Case 169, 4 place Jussieu,*
*75252 PARIS Cedex 05 France*
*Phone: (33) 1 44 27 87 43*
*Fax: (33) 1 44 27 70 00*
*Zahia.Guessoum@lip6.fr*

## 1. Introduction

Nowadays, the number of individual users, government agencies and companies with Internet access is expanding rapidly, and lots of them have a Web site. As a result, businesses are enthusiastic about setting up facilities on the Web for electronic commerce. But, the openness of business toward Internet is performed at the price of high security risks. Every professional knows that the only way to secure completely a private network is to make it unreachable. However, even if this solution was undertaken for many years, today it is not possible to close private network especially for business purpose. As businesses wake up to this reality, the demand for secure Web services becomes an important issue that must be considered carefully. The focus of our work concerns one critical security management issue that is intrusion detection. Some draw-backs of existing systems reveal the necessity of designing a new generation of self-adaptive systems. In fact, self-control, flexibility, adaptability, autonomy and distribution are the main features to be addressed in a suitable architecture that fulfils these requirements. The introduction of multi-agents system (MAS) in a network seems so promising to enable network entities to perform adaptive and “intelligent” behaviour. “Intelligence” means that network entities provide reasoning capabilities exhibit behaviour autonomy, adaptability, interaction, communication and co-operation in order to reach specified goals. In this context, we propose a new approach for intrusion detection using intelligent agent (IA) technology. This approach appears an appropriate candidate to make a balance between security requirements, system flexibility and adaptability for intrusion detection.

Our paper investigates some scenarios using IA technology in the context of security management. It is organised as described in the following; the first part gives an overview of frequent and recent attacks, which targets the e-commerce servers. Our multi-agents architecture for intrusion detection and the intelligent agent model are described in the second part. In the third part, we give an overall description of the implementation of our intelligent agents using DIMA*, the operational multi-agents platform we chose. We also present a case study using our proposed IA architecture to detect an ICMP flooding attack to a web server. Finally we conclude with some relevant remarks and future works.

## 2. E-Commerce Attacks Overview

With the rapid expansion of the E.Commerce, the Internet users are more and more concerned to give some personal information (address, age, credit-card number,...) when they visit the Web server or when they purchase or buy on the web servers[1]. Nowadays, lot of the E.Commerce web servers could offer a secured communication (the information confidentiality is guaranteed during its transfer from the host to the web server) with some mathematical algorithms. However, it could happen that the storage of these data which have just been confidentially transferred on the network is not secure. Like this, a french newspaper ("Le Canard Enchainé") explained on March 1999 how a freelance journalist fell completely by "chance" on the list of all the clients of a flower shop. Inside he found all client names, client addresses, credit card numbers and much more serious (according to us!), he found all the private messages the clients sent with the flowers.

Besides attacks against data, there are other attacks which target directly the service offered by the E.Commerce server: that is, instead of trying to read data or to modify them, the attacker will prevent the honest user to use the E.Commerce service or to obtain the information he needs. This kind of attack is called Denial of Service attack. Lots of similar cases with their consequences were related in the press at the beginning of this year. To understand the scenario that can happen, the figure 1 gives the example of two computers A and B which want to communicate through the Internet. For Instance the user of the computer A wants to read a web page in the computer B. Before being connected, the user A sends a SYN message and has to wait for the answer of the server B (SYN ACK). Then the user has again to send a last ACK message before beginning the communication. The server B, is able to treat simultaneously several attempts of connections. But, as its capacity is limited, if its limit is reached any other new connection won't be possible on the server B. Imagine now that an attacker initiates a large number of connections without answering to the SYN ACK message of the server B. The server B will be rapidly overloaded without being able to answer the honest users who also try to connect. This kind of attack called *SYN flooding* will block completely the server B. Another kind of denial of service attack is *ICMP flooding*. The attacker send a high rate of ICMP packets (usually ping requests but other type of requests are also possible) to flood a network and bring it down. It is quite difficult to response to such an attack because every attempt of connection can not alone be distinguished from a real demand. It is only the analysis of a larger set of connections that will help the victim to circumvent the source of the attack and/or block all the request of connection coming from this source. Unfortunately it is quite easy to complicate these kinds of attacks in

* DIMA : Development and Implementation of the Multi-agents Systems which is a platform developed by Zahia Guessoum at LIP6 - France.

order to make them impossible to be detected. One version quite dangerous is the “distributed denial of service”. The attacker enters several computers connected to the Internet and launches a coordinated attack from these computers to the same target destination. In this case, the victim will be rapidly submerged and the only thing to do is to disconnect from the Internet.

Even if we can temper the danger of such kinds of attacks because they do not provoke a real damage but just the impossibility to access temporary the server, they could become quite problematic because of the influence taken by the Internet today in our life.

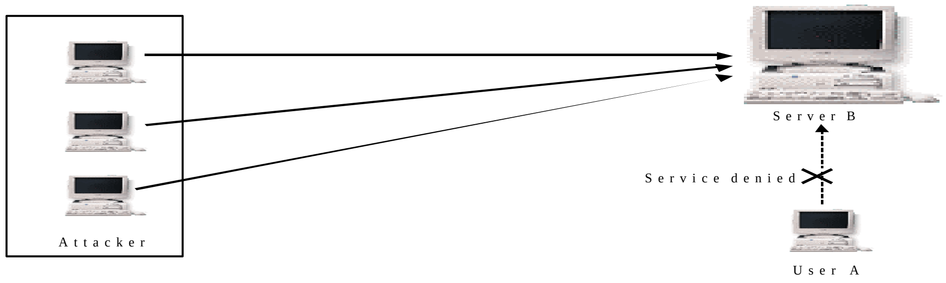

Figure 1: Denial of service attack

## 3. The Multi-agents System-based Network Security Management Architecture (MANSMA)

The key characteristics of our approach include distribution, autonomy, delegation, cooperation, adaptability and flexibility. In the proposed architecture (MANSMA), IAs, located at specific network entities, are structured hierarchically (see figure 2). In MANSMA, we distinguish two functionality layers: a *Manager Layer* and a *Local Layer* [3].

The *Manager Layer* manages the global security of a network. In this layer we identify three levels of agents: a *Security Policy Manager Agent (SPMA)*, an *Extranet Manager Agent (EMA)* and several *Intranet Manager Agents (IMA).* The *SPMA* manages the security policies specified by the administrator. The *EMA* manages the security of the distributed network. It controls *IMAs,* which report pertinent analysis. The *EMA* performs then another analysis to confirm the detection of an attack. It can also ask for more data processing and delegate new monitoring tasks to the *IMAs*. The *EMA* is also responsible for distributing a set of *Local Agents (LA)* to each *IMA*. The *IMA* manages the security of a local network. It controls *LAs* and analyzes the monitored events reported by these agents.

The *Local Layer* manages the security of a domain, constituted by a sub-set of hosts in a local network. It is composed of a group of *LAs*, which have specific functions. We distinguish three kinds of *LA*: *Extranet LA*, *Intranet LA* and *Internal LA*.

In this hierarchical multi-agent model, each *manager agent* has the ability to control specified agents and to analyze data, whereas the *LAs* monitor specified activities. In each level, agents communicate and exchange their knowledge and analysis for detecting intrusive activities in a cooperative manner.

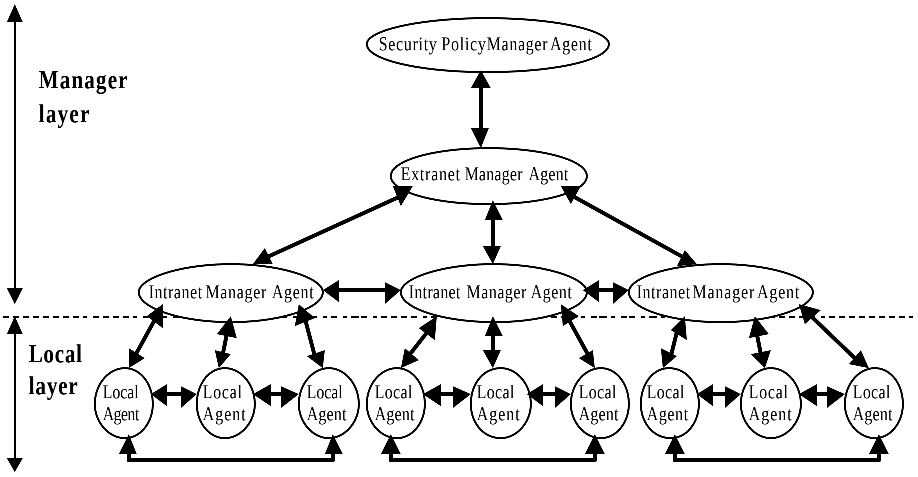

Figure 2: MANSMA Functional Architecture

## 4. The Intelligent Agent Model

To model intrusion detection, agents must combine cognitive abilities cognitive (knowledge-based) to reason about complex attacks with reactive capacities (stimulus-response) to react rapidly to the environments changes. So, an agent has three functions: a *filtering* function, an *interaction* function and a *deliberation* function. These functions are described in the following paragraphs.

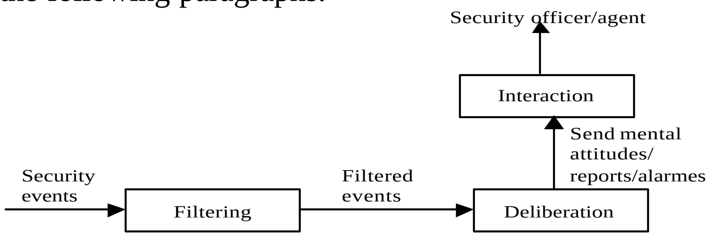

Figure 3: Interactions between agent functions

### *4.1 Event filtering function*

A *security event* is characterized by its type, its observation point, a temporal attribute (representing the event occurring moment), and a set of non-temporal attributes. According to the event type and its observation point, we identify various event classes (see diagram below).

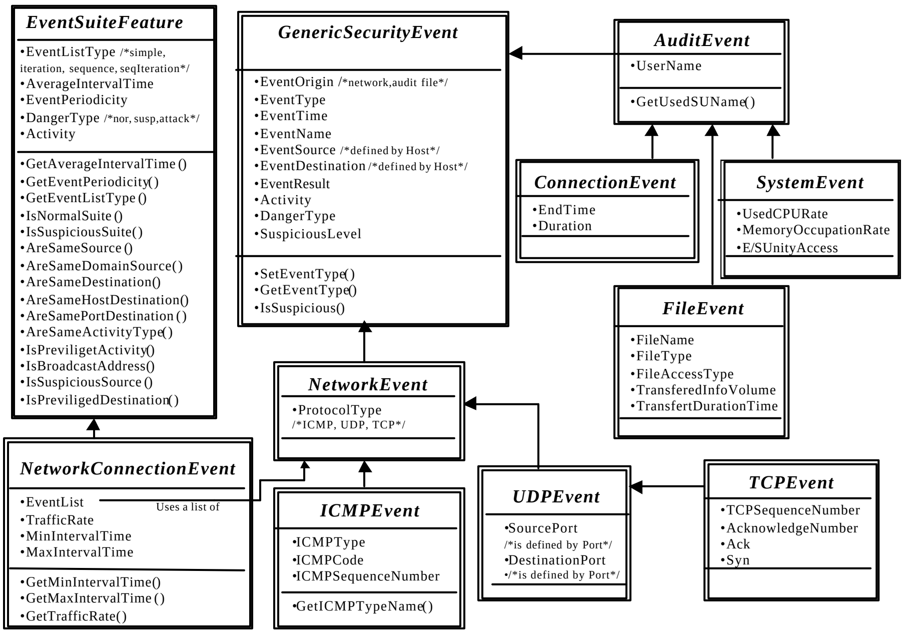

Figure 4: UML classes of security events

The *event filtering* function filters security events produced in the network, according to event classes specified in a detection goal. Indeed, the events occurring in network are not all collected. In fact, when a detection goal is sent to an agent, a set of event classes to observe is specified to it. Thus, when an event occurs in the network, the agent tests if it matches the event classes specified in the goal. If it matches, it is collected. The filtered events are then stored, waiting to be treated by the deliberation function.

### 4.2 Interaction function

This function describes interactions between the above-described agents. It allows them to communicate their analyses and knowledge and mental attitudes (beliefs, suspicions…). In fact, m*anager agents* interact with *local agents* by:

- sending goals, derived from security policies;
- delegating specific functions of monitoring/detection and specifying the various domains to monitor;
- asking particular information: the suspicion level of a specific user, the list of events generated by a user, etc.;
- and receiving the relevant reports or analyses results and alarms.

*Interaction* function also permits interactions between the security officer and *security policy manager agent/ extranet manager agent*. It ensures the reception of specifications and requests from the security officer such as security policies to apply. It allows the delivery of security reports and alarms when an attack is detected. The security officer can also ask for additional information (asking for the current security state of the network, the list of suspicious users…).

### 4.3 Deliberation function

Security management must deals with significant network characteristics such as: 1) its continuous variation, particularly in terms of users and offered services; 2) and variation of its security problems such as new vulnerabilities and increasingly complex attacks. Considering the unpredictable character of the agent environment behavior (network), we adopted a BDI solution [4][5] for modeling the security management system. Thanks to the *deliberation* function the agent is able to reason and extrapolate by relying on its mental attitudes, built knowledge and experience, in a rational way, to find the adapted answers. The agent uses its beliefs resulting from the filtered events and beliefs of the neighboring agents for reaching its specified goals. When a goal is reached (an attack is detected), it executes appropriate actions.

## 5. Implementation

### 5.1 General Description

The presented agent model has been implemented with the multi-agent platform DIMA [6]. The latter is mainly characterized by a modular agent architecture. DIMA proposes the extension of the single behavior of an active object into a set of behaviors. In our implementation, each agent has three behaviors:

- The *filtering* behavior filters security events. When an event occur in the network, it is collected only if it matches the event classes specified in the detection goal.

```
EventFilter {
Repeat
security-event := get(security-event-to-filter);
If  is-in-list-of-event-types-to-filter(security-event)
Update-list-of-filtered-event(security-event);
end repeat }
```

- The *interaction* behavior manages the interaction between the agent and the other agents. It defines the mailbox of the agent and the way the messages are received and enqueued for later interpretation. An agent may need some others information to refine its analysis. In this case, it asks other agents to give it the necessary information.
- The *deliberation* behavior represents beliefs, goals, intentions and knowledge of the agent. It is responsible 1) for generating adequate responses to the messages received from the other agents and 2) for achieving the agent goal(s).

When an agent receives a *detection goal,* it updates a set of event classes to filter. Then, when an event occurs it is filtered by the filtering module and sent to the deliberation module. This one, updates/creates agent *beliefs* and then test if this belief matches an attack . If it matches, then a *detection goal* is reached and a list of *intentions* are sent to the interaction module for being executed (see figure 4).

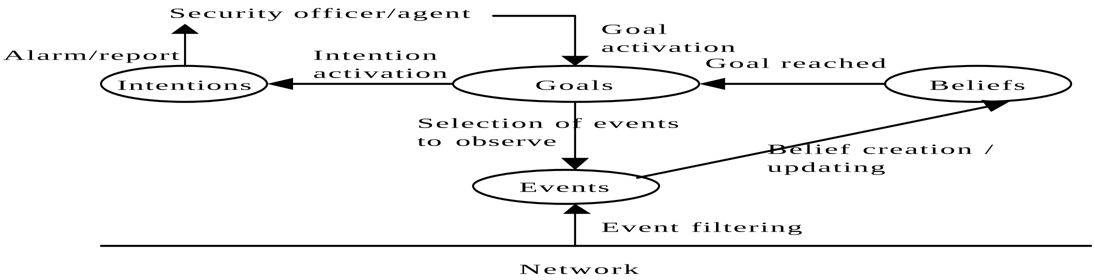

Figure 5: Interactions between mental attitudes

Our implemented system detects well-known attacks such as doorknob rattling, ping sweep and ICMP flooding attacks.

### 5.2 Case Study: ICMP flooding attack to a web server

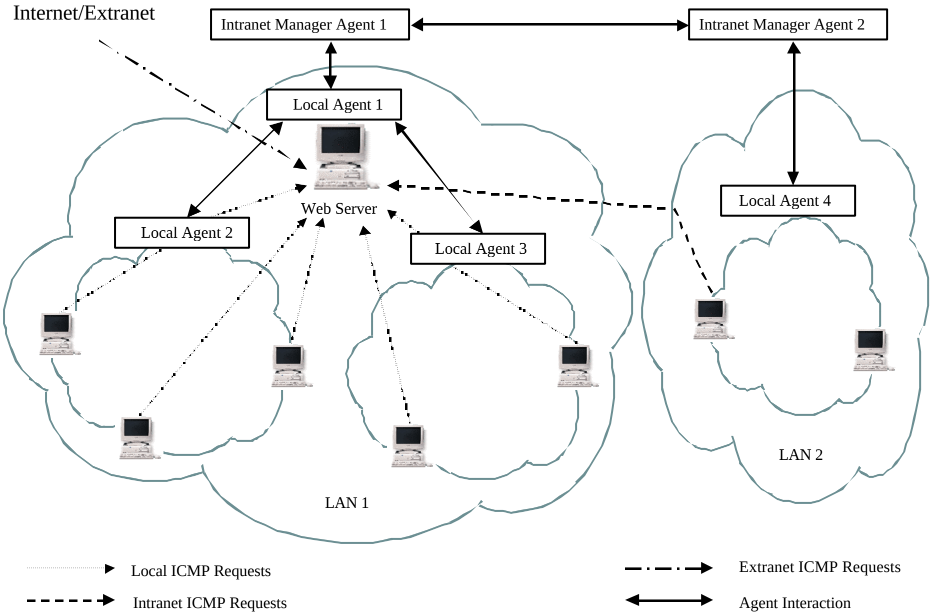

Figure 6: Scenario of ICMP flooding attack detection

This section describes a scenario for detecting an ICMP flooding attack to a web server. In the web server of a LAN 1 (Figure 6), the local agent 1 monitors all the ICMP requests entering the web server. These requests come from different sources of the network:

- Local hosts of the same LAN 1
- Local hosts of the LAN 2 belonging to the same Intranet as LAN 1
- External hosts sending requests from the outside (Extranet/Internet).

The local agent 1 of the web server plays a preponderant role because its goal is to detect as soon as possible that the number of ICMP requests it receives during a certain period of time could target an ICMP flooding attack and block the access to the server. For that purpose, the local agent 1 filters the ICMP requests arriving at the web server. The local agent 1 can also interact with other local agents (local agent 2 and local agent 3) placed in a strategic way in the LAN 1 to obtain the result of their local analysis (for instance, number of pings they collect from the hosts they supervise and send to the web server). The same interaction is possible between the local agent 1 and the Intranet manager agent 1 for obtaining the analysis of the local agent 4 through the Intranet manager agent 2. These events are used to update/create the beliefs of the local agent 1. The deliberation function tests if these beliefs match an ICMP flooding attack. If the goal is reached, an alarm is raised.

## 6. Conclusion and Future Works

In this paper, we presented a multi-agents system which aims to make security management more flexible and customizable. This approach is innovative because it allows a security management system to adapt to unpredicted complex evolution of both network environments and security attacks. To model agent knowledge, we used the BDI theoretical model, which require us a hard work to deduce a practical implementation. A prototype is currently implemented using the DIMA platform. This prototype enables the detection by the agents of well-known attacks, particularly ICMP flooding attack. Presently, we work on the adaptation of agents’ behaviors to new kind of attacks. We are regarding the eventuality to use a hybrid intelligent/mobile agent solution by integrating the mobility in our local agents, in the scope of an international collaboration (CUI - Geneva, Switzerland) [7].

## References

[1] Frédéric Schtütz and David Billard, "Securité des données et des services dans l'Internet", Revue :Bulletin SEV/VSE 7/2000 Zürich, 31th March 2000.

[2] E.Léopold and S. Lhoste, "La Sécurité Informatique, Que sais-je ?", PUF, N°3460, 1999.

[3] K. Boudaoud, H. Labiod, Z. Guessoum and R. Boutaba, "Network Security Management with Intelligent Agents", In proceedings of 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS’2000), Honolulu, Hawaii, 10-14 April 2000.

[4] A. S. Rao and M. P. Georgeff, "Modeling Rational Agents within a BDI-Architecture", Technical Note 14, 1991.

[5] A. Rao and M. Georgeff, "BDI Agents:From Theory to Practice", Tech. Note 56, 1995.

[6] Z. Guessoum and J.-P. Briot. "From Active Object to Autonomous Agents", *IEEE Concurrency,* volume 7 N° 3, pp. 68-78, July/September, 1999.

[7] J. Hulaas, A. Villazon, J. Harms, "Using Interfaces to Specify Access Rights, in Secure Internet Programming – Security Issues for Mobile and Distributed Objects", J. Vitek and C. Jensen (Eds.). Lecture Notes in Computer Science vol. 1603, Springer Verlag Inc., New York, NY, USA, 1999.