Transactions on Information and Communications Technologies vol 6, © 1994 WIT Press, www.witpress.com, ISSN 1743-3517

Intelligent framework for security

M. Toure
Laboratoire API/Université Paul Sabatier, 50 chemin des maraîchers - Toulouse - 31 062 France

Abstract

This paper discusses an interdisciplinary approach of security. Artificial Intelligence (AI) techniques can be helpful for such security aspects as: intrusion detection, real time audit trail analysis, supervision, etc. But, without a clean and careful approach, this AI and security cooperation cannot be effective.

Our work started from two hypothesis:

First hypothesis. Security researchers focus more on tools and techniques for building secure systems rather than methods and/or methodologies for managing these tools and techniques.

Second hypothesis. Many factors (*human factors* among others) are now not or less taken into account in several security systems, because of a lack of security *Knowledge level* (such as Chandrasekaran or Newell thought it for general purpose knowledge based systems) capable of capturing all security factors.

The work reported here focuses on the concept which we call "Intelligent Security" [12], which tries to find an answer to the three questions above: 1) What knowledge sources can be relevant for a security system, 2) How to organize security knowledge sources in a divide to conquer way so as to view the whole security system as a set of dependable components [9], 3) How to build "Intelligent Security" systems out of multiple, heterogeneous components.

MAS2M, a Multi-agent model for security systems production, based also on Chandrasekaran's generic tasks is for now our thirst answer to these questions.

Introduction

Reliable security systems can now be built with tools such as: cryptography, biometric information based access control, etc. But several factors are less or merely not taken into account (human factors among others).

It's important to note that in France, more than 80% of software and hardware damages are directly or indirectly due to human factors.

In our laboratory's view, those drawbacks are in part due to a lack of global and more abstract approaches for security systems. They are also due to a hurry focus on tools for implementing security, rather than methodologies for dealing with general purpose security systems, namely, a lack of "security knowledge level".

The framework that we'll present relies on the fact that a security system can be viewed as a naturally structured set of generic tasks [3] [4] (be they knowledge based or not). A security problem can be viewed as generic tasks (such as: intrusion detection, access control, etc.) cooperating for flexible security.

It's important to recall that the framework is based on an interdisciplinary approach. Thus, our work is not limited to adding some intelligent functionalities to the security systems.

We first briefly review some terminology, so as to introduce what we mean by security. We particularly follow as far as possible the terminology issued by works in [2] [9].

In section 2, we show that an interdisciplinary approach of security can efficiently contribute to a methodology of designing what we call "Intelligent Security" [12]. Thus, we introduce a systemic approach of security.

In last sections we present MAS2M (a multi-agent based model for building "Intelligent Security" systems). How MAS2M can improve security management and integration, because we also try to tackle security integration (which includes *security of security* also called *guarding the guards*).

1. Terminology

Dependability is the generic concept including as special case such attributes as reliability, availability, safety and security [9] (see figure 1). Our security definition is one of the four dependability attributes. It's also based on the European norm of security [8]: a combination of integrity, confidentiality and security-availability attributes.

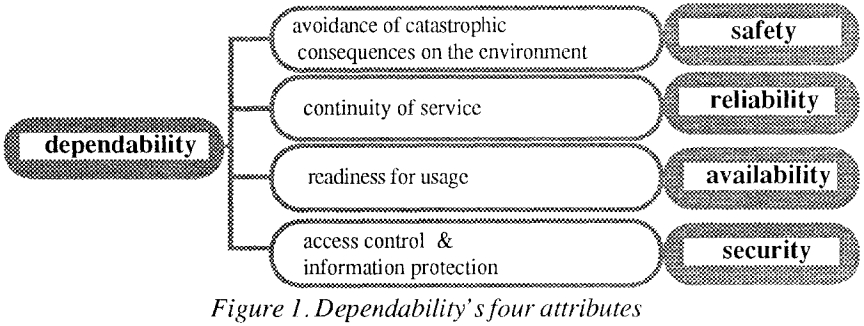

Figure 1. Dependability's four attributes

Dependability's four attributes are not mutually independent. A classical example is the antagonism between security and reliability. Security is generally obtained by access control and information protection means, whereas reliability is obtained by temporal and/or spatial redundancy-based methods. The first aims at restricting access to information and the second at making information more available. So, without a discussion between these two attributes, the security or target system's whole dependability can be compromised. In MAS2M, this problem will be tackled by both security systems design and integration. Modularity and abstraction are the main criterion for obtaining an acceptable dependability compromise.

2 Security tasks modelling

In this section, we try to explicitly define what we mean by "security task" and what can be a security task's main components. Thus, we recall how people generally deal with security problem. We first review how computer scientists generally deal with it (computer security). Next, we try to understand how people generally deal with security in their everyday life. Last, we introduce an unified and more abstract approach which seems in our view very promising.

2.1 Computer security systems

"Computer security" is computer scientists' abstract view of security systems. "Computer security" systems' principles are based on an abstraction of "real security systems". Concepts such as *subjects* can be viewed as *human intruders* or *authorized* people abstraction, *objects* as *resources* or *guarded people* (see figure 2).

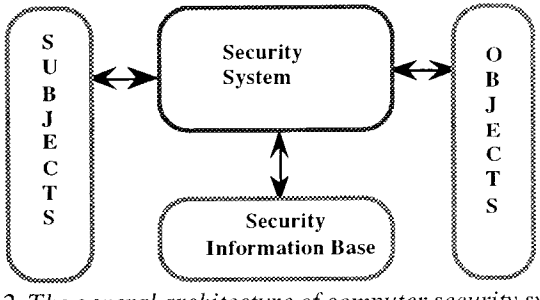

Figure 2. The general architecture of computer security systems.

Figure 2 illustrates the general architecture of a computer security system. We do not make any hypothesis on the system architecture (distributed or centralized). But it is important to note that the security system policies and strategies are embedded in the two modules (*security system model*, *security information base*). We also do not make any hypothesis on the way the security system deals with security tasks.

To day it is well recognized that this abstraction cannot capture most of the security factors because: 1) This architecture is monolithic and raises many integration problems, 2) Because it focuses more on technical tools rather than on a methodology for security. Many alternative architectures have been proposed, but they all fall in the same problem.

Since the most critical factor in security systems is human factor and in order to better understand the factors affecting a security system, we first see how humans deal with security in their everyday life (section 2.2). Following sections we'll tell us more about the approach we take for overcoming these computer security drawbacks.

2.2 Real security systems

In "real security", security systems can be viewed as networks of "guards" and security devices, in which every guard or device have at one time a precise task that can be control, supervision, etc (see figure 3). Guards need to communicate or to share information for cooperative security tasks. They also need to obtain information from security devices. Each of them can locally come to certain decisions based on its experience in security (humans generally) with respect to its environment rules and constraints or under a highest authority orders.

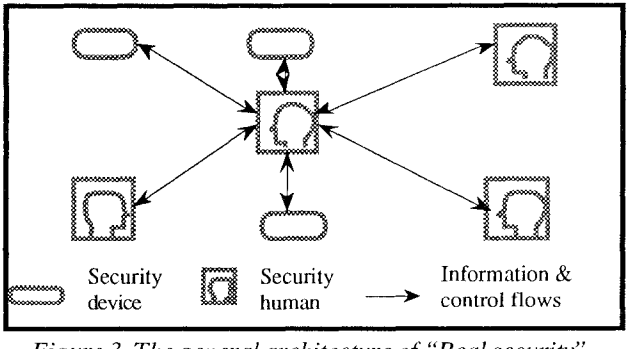

Figure 3. The general architecture of "Real security"

Figure 3 may represent a highly supervised building, an air traffic security system control. It can also represent any real world security system. This figure recalls us that human will forever be an actor of security systems. Thus, human factors must not be neglected in security systems.

In security systems we used to take into account only one facet of human factor, the faulty, the weakest. But, we recently found that there is no better remedy for human than human. Thus, human "security knowledge" integration in security systems can be interesting. In other terms, there's a need of an intelligent security framework capable of capturing a systemic approach of security.

2.3 Systemic approach of Security

The previous discussion between "computer security" and "real security" leads us to introduce the "Intelligent Security" concept which is based on a systemic approach. The systemic approach relies on the hypothesis that security involves the integration of three complex systems:

Security system, which is achieved by protection and access control based techniques (such as cryptography, access control techniques, physical security, etc.).

Environment system, which includes existing security systems (in the case of heterogeneous systems for example), but also social, philosophical, cultural factors of the environment which affect the security system [10].

Human system, the more complex, the less understood and the less taken into account in today's security systems. We mainly focus on this system and particularly on human intentional faults.

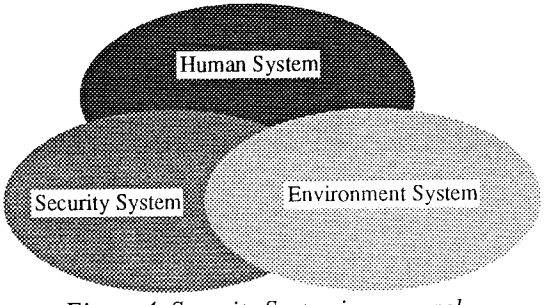

Figure 4. Security Systemic approach

In order to globally take into account these three systems, we need a general framework for integrating different knowledge and know-how sources for Human, Environment and Security systems.

2.4 Task-based approach of security systems design

Task is the abstract entity structuring our security systems. Our view of the task concept is inspired from Chandrasekaran's works on generic tasks [3] [4].

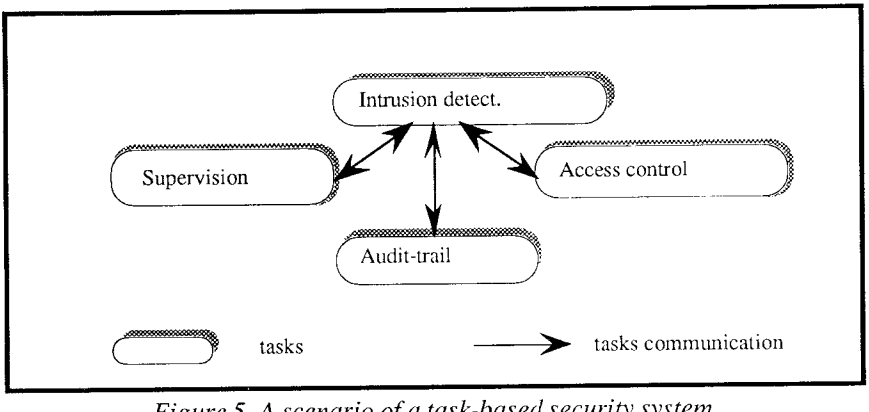

Figure 5. A scenario of a task-based security system

Tasks can be viewed as appropriate security solving units, focusing on a security problem aspect. To each security task is associated methods and pertinent information for solving the focused sub-problem. Our tasks are generic in the sense that they can deal with any security problem provided that the security system designer specifies adequately the input information required by the task.

3 MAS2M : A multi-agent framework for security systems

3.1 Architecture

MAS2M's functional architecture can be depicted as a three-layered one (see figure 6). The security system is decomposed into sub-problems *(tasks)* based on security requirements.

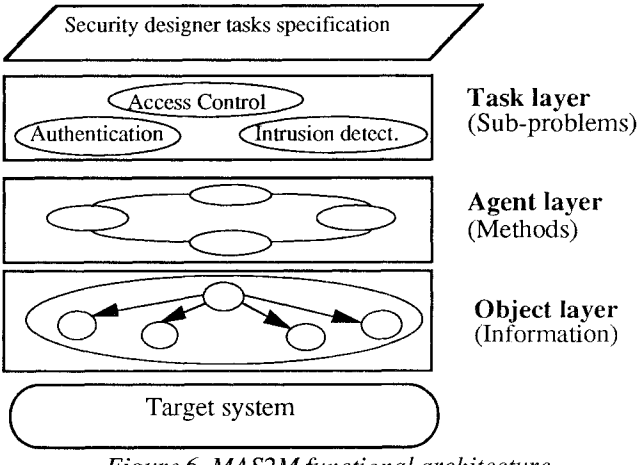

Figure 6. MAS2M functional architecture.

To each task is associated a set of competitive "security solving units" called agents. Agents are computational, at sense of Bond & Gasser [1] or Erceau & Ferber [6]. Each of them needs an appropriate set of informations encapsulated on objects.

3.2 Task layer

The task concept, can also be justified by the fact that security systems generally focus only on one or some security aspects. It's why, it's current to find in the security market or literature, specialized security systems in aspects such as intrusion detection [5] or authentication, etc. Our aim is not to cover all security aspects, but to show how security tasks can be solved by granular building blocks called agents.

Tasks communication is out of the scope of this paper. For now, tasks are our most abstract entities. But it is important to note that an event based communication can be suitable for tasks communication. Since security problems are not deterministic, because they deal with human intentional faults which are nowadays unpredictable and not understood, event modules can be viewed as local watching units, specified by the task designer.

3.3 Agent layer

Agents are intelligent and "autonomous" building blocks for the tasks. They can work either in parallel (an important factor for security system performance) or in a cooperative way, in a task solving process.

They can be modular knowledge sources corresponding to security tasks expertises or heuristics. Thus, they are capable of inferences and are referred as expertise modules in *Security*, *Environment* or *Human system* (Section 2.3) knowledge sources.

Agents can be rule or procedural based, according to the appropriateness to the problem to be solved. They can also be viewed as trusted black boxes. This is generally the case when their design is out of the security designer competence.

3.4 Object layer

Agents need certain information for solving a task. For example, an agent that suspects a user in an intrusion detection system needs to have some information about the user's activities. This information is generally encapsulated in modular information units called profiles. This information is obtained from an abstraction of the target system's security relevant activities. In MAS2M, we opted for an object oriented approach, because of its appropriateness to security. Because, for implementation purposes, once the information layer is built, topmost layers can be built as successive abstractions of the information layer.

4 MAS2M : From theoretical concepts to practice.

This section discusses the practical aspects of the MAS2M model. To make it more easier to understand, we illustrate it by the reverse engineering of a well-known intrusion detection system IDES [5].

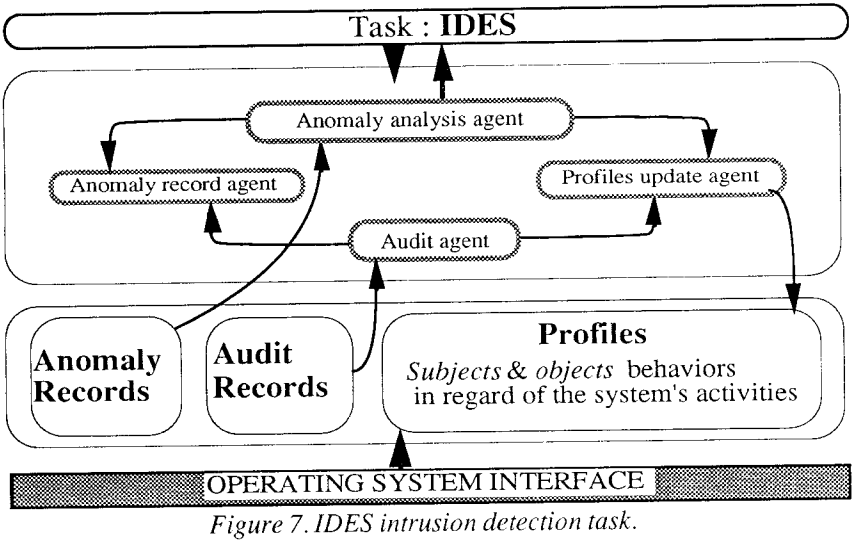

Figure 7. IDES intrusion detection task.

Figure 7 is a partial view of IDES according to MAS2M's philosphy. The basic architecture of an intrusion detection system can be viewed as four agents [5] (generally working on pipeline). Agents are in this case knowledge sources built from our experience on intrusions. Each of them needs specific information to do its computations. They also need to communicate to other agents (in the case of deviant behaviours of the system for example).

The information layer is obtained from an object oriented abstraction of the operating system. This abstraction is allowed by the building of an appropriate interface. Such interface allows MAS2M systems portability, it avoids also the operating systems kernel modification.

Conclusion

In this paper we focused mainly on general aspects of our security framework. We also insisted on the security terminology, because the term "security" and its associated attributes are employed in several literatures for different meanings.

It is important to recall that we do not pretend to replace current security systems by knowledge based ones. Our security framework can be implemented as an intelligent security layer, avoiding any change on lower layers' security.

Bibliography

1. A. H. Bond and Les. Gasser.
Readings in Distributed Artificial Intelligence.
in "Distributed Artificial Intelligence", Morgan Kaufman, Los Altos 1988.
2. Carter W.C.
A time for reflection..
IEEE Symp. on Fault Tolerant Computing. Santa Monica, California, June 82.
3. Chandrasekaran, B.
Generic tasks in Knowledge-Based Reasoning : High level building blocks for Expert Systems design. IEEE Expert - FALL 1986
4. Chandrasekaran, B.
Towards a functional archit. based on generic information processing tasks.
Intelligent Joint Conference on Artificial Intelligence - August 1987
5. Denning D.E.
An intrusion-Detection Model
IEEE Transactions on Software Engineering - Vol SE-13, n°2, February 87
6. Erceau J. Ferber J.
Introduction aux prem. journées Francophones d'I.A.D. et des S.M.A.
Premières journées IAD & SMA - Toulouse 93
7. Ferber J. Ghallab M.
Problématiques des univers multi-agents intelligents.
PRC-GRECO/IA 1988.
8. ITSEC.
Information Technology Security Evaluation.
Harmonized criteria of France, Germany, The Netherlands, UK - June 91
9. Laprie J.C.
Dependability : a unifying concept for reliable, safe, secure computing.
Algorithms, Software, Architecture - Information Processing - IFIP 92.
10. Nessett Dan M.
Factors Affecting Distributed System Security.
IEEE Transactions on Software Engineering - Vol SE-13, n°2, Feb. 87.
11. Newell A.
The knowledge level
Artificial Intelligence - Vol 18
12. Touré M.
Intelligent Security. IEEE Scalable High Performance Computing Conference. May 1994 - Knoxville-USA