A distributed Intrusion Detection and Response System based on mobile autonomous agents using social insects communication paradigm

Serge Fenet 1

Laboratoire d’Ingénierie des Systèmes d’Information
Université Lyon 1
Villeurbanne, France

Salima Hassas 2

Laboratoire d’Ingénierie des Systèmes d’Information
Université Lyon 1
Villeurbanne, France

Abstract
The ever increasing connectivity of current computer environments makes traditional Intrusion and Detection Systems more and more inefficient. The ability of moving processes across networks brings new security problems, but also gives us new ways of dealing with these environments. In this paper, we propose an architecture for a distributed stealth Intrusion Detection and Response System (IDRS) based on mobile agents mimicking behaviors of social insects. We present the motivations of an approach that solves several problems actually unchallenged and offers many new ways of thinking future IDRSs. We also depict the foundations of our architecture, discuss its main points, and expose partial results obtained from a prototype. Finally, implementation issues and future work are presented.

1 Introduction
The continuing expansion of network connectivity forces us upon a new way of thinking the computational process. Traditional schemes are no longer valid when data housed on an Australian computer can be accessed as transparently as a local disk from a French machine. We are now witnessing a maturation process that induces the disappearance of centralized monolithic systems and...
the birth of a complex global distributed environment. One example can be easily illustrated by the paradigm of mobile agents – mobile programs that can jump from a computer to another and interact with their hosts. Because of their high accessibility, huge content and wide distribution, it is more profitable to send mobile “computing units” roam vast stocks of data than tap into them and perform treatments locally. It is now often more interesting to bring the computation to data than the opposite.

Meanwhile, this evolution is not painless. As our systems are becoming more open, they are also becoming more sensible to malicious attacks. Again, mobile agents provide us good examples. Who would let an unknown piece of software enter one’s local area network and interact blindly with one’s resources? Who would trust an agent whose integrity could have been breached? From an agent’s point of view, the security threats associated with mobile agents can be divided in two great families: the attacks an agent can perform toward its environment –other agents or hosting platforms–, and the attacks that can be performed toward an agent from other agents or hosting machines. Nevertheless, in this paper we are more interested in the benefits an agent-based system can bring than in the risks endured by such systems. We do not approach the problem of mobile code protection. Obviously, mobile agents exhibit and withstand specific menaces based on their migratory abilities. But we can likewise use this unique skill to build systems able to cope with the ever changing distributed environment of a network. Moreover, we advocate the idea that in the security domain, only a distributed framework is able to deal efficiently with contemporary computer distributed environments. We present in this paper the skeleton of such a framework based on the concept of mobility. Our work is a proof of concept destined to exhibit several characteristics:

- the efficiency of a distributed detection system;
- the reactivity of a collective response system;
- the toughness and stealth of a framework of which every part is highly mobile.

All of these points are rooted in the interaction between the populations of the framework. The notion of complex collective behavior emerging from the interactions of many simple agents, the emergence, is the basement of the field of Artificial Life ([15]). The study of natural distributed systems that exhibit capabilities of complex distributed problem solving –like anthills or wasps’ nests– is a great source of inspiration for real distributed systems implementation. As our work focuses on the mechanism of total mobility, our agents should be given the ability to send messages to each other without knowing the recipient location. We base this communication on the use of one medium, the electronic version of the chemical medium used by social insects: pheromone.

This paper is organized as follows. We present in the next section a suc-
cinct background and previous works in the domain of Intrusion Detection Systems. Then we describe our approach and report in details the different agent populations and their interactions. We also criticize it and discuss briefly its advantages and drawbacks while reporting the current state of implementation. Finally, we provide an overview of our future plans.

2 Background in Intrusion Detection

Intrusion Detection is defined by [16] as “the problem of identifying individuals who are using a computer system without authorization”. In our work, we extend this definition to the detection of computer processes. These processes can materialize the interaction of an intruder with a target machine, or can be autonomous programs. Always in [16], we find a large categorization of the detection model:

- The *misuse detection model* that monitors the exploitation of known weak points in an operating system. This guarding can be done by researching a specific pattern of actions or events.
- The *anomaly detection model* that detects changes in patterns of the behavior of the system. We will relate to these detection model in Section 3.2.1.

We will show in Section 3 that an Intrusion Detection System (IDS) based on mobile reactive agents can perform surveillance in both domains. Moreover, it is easy to include in such a system a defense layer of mobile agents that attempt to stop intrusion as it occurs.

Historically, the idea of the first IDS dates back to 1980 ([1]), but the field was really born with the model of [8] and the prototypes described in [4] and [20]. These IDSs are centralized and based on a monolithic architecture. Namely, data are collected on a single machine –by looking at log files or network flow– and are analyzed on a single computer. Nevertheless, if this one weak point is successfully subverted, the attacker obtains considerable power to gain access to the entire network. We think that this risk of overthrowing is the main vulnerability of all current IDSs.

The approach depicted in [14] presents a distributed information gathering step, but the drawbacks are the same: the centralized analyzing process is hazardous and reconfigurability and scalability are limited.

Subsequent works like [24], [22] or [3] present a fully distributed architecture: data collection and information analysis are performed without central authority. The authors hence answer to the scalability problem. Despite that, the IDS itself, being static, still endures the risk of being attacked. This is even more and more conceivable when we consider the increasing of distributed attacks.

Recently, a pertinent evolution was set in motion with the work of [6], [2], and [18]. Their architectures –called respectively AAFID and EMERALD– deals

\(^3\) Internal or external to the overseen network.
with the problem of centralized processing. These systems, composed of different populations in interaction, are probably the most achieved implementation of a distributed IDS. Though, and even if the designers of AAFID mention that “agents could migrate from host to host by combining the AAFID architecture with some existing mobile-agent architecture”, the mobility aspect is not developed and no example of use of this ability is given.

In parallel, scientific investigations were fueled by the study of immune systems. In [13], the authors state similarities between the defenses of natural immune systems and computer security: both must discriminate self and non self to protect a complex system from inimical agents. Our own approach is based on the use of a chemical-like information that represent an abnormal behavior. This information could be generated easily by tracing the execution of sensible processes as described in [13]. We will now present the foundations of our system. It is based on two main principles: full distributivity and full mobility. It also sustains a reaction layer embodied in mobile defense agents.

3 Our system

This section will present our approach in detail. We will firstly describe the motivations that led its creation, then detail its facets and inner mechanisms.

3.1 What motivated us

Our motivations came from a wide variety of fields. Let us enumerate them briefly.

3.1.1 Natural systems influence

We were firstly struck by the parallel of functionality between the “real world” and the “computer world” and then inspired by problem solving methods used by social insect colonies. Studied in the field of swarm intelligence ([5] and [25]), they are very attractive in the sense that they can cope efficiently with a wide range of problems and can be built with simple basic bricks: swarms of simple agents that interact with and through their environment. This interaction is mainly based on a marking mechanism and is using as medium a chemical volatile substance called pheromone. This indirect communication mechanism, called stigmergy in [11], is the root of the emerging complex properties of systems made of unintelligent reactive agents. This mechanism is for example used to trigger and coordinate the defense of an anthill, gather food, or build complex architectural creations... We thought it would be very interesting to implement a system exhibiting the same properties of distribution, reactivity and sturdiness, applied to the domain of Local Area Network (LAN) security. This idea is not new, though. Several existing works are based on this natural metaphor. They deal with fields as varied as network management simulations ([19]), distributed constraint satisfaction problems or
routing algorithms simulations ([9]). All these domains however share common characteristics: great complexity, high dynamicity, and physical distribution.

3.1.2 Previous implementation: A.N.T.
The implementation of this IDS is based on an earlier project called *Artificial Network Termite colony* (A.N.T.) ([10]). That framework of mobile agents solves physically distributed problems and provides network control tools in using a mechanism of simulated chemical substance deposit for information sharing and remote influence. We thus implement an indirect communication technique that is performed through the environment. Our work showed that different superimposed pheromonal gradient fields can model many constraints and offer powerful solutions to problems difficult or impossible to solve with a centralized approach – like distributed optimization of multiple constraints scattered across the network.

The principle is the following: we implement on a LAN the evolution in space and time of multiple criteria in managing dynamic gradient fields. Each one represents a criterion to optimize and is emitted by computers where the corresponding criterion is optimum. Mobile agents, moving among the machines, are bathed into these simultaneous pheromone fields and actively seek optimum solutions – i.e. physical position – that satisfy these criteria. As each host knows its own and its neighbors’ pheromonal contents, basic agents will be reactive in the sense that they will only make local migration choices.

A.N.T. is implemented with the following characteristics in mind:

- the reactivity and flexibility necessary to deal with a computer network dynamic environment;
- the ability to efficiently search within a space of states restricted by any number of constraints (machine load, processes related information, document keywords, host description, etc.);
- the use of mobility to satisfy constraints: agents migrate to locations that optimize criteria they are influenced by;
- the capacity to dynamically modify the relative importance of the criteria in the migration decision;
- the light weight of small agents that only slightly increase the network load.

Our mobile agents perceive their world only in terms of concurrent gradient fields, so we must build these fields with care. This is done by implementing two mechanisms essential to the non-direct communication medium at the heart of natural colonies: *diffusion* and *evaporation*. A.N.T. is a distributed framework that any process can use to create, gather and spread information in a way suitable to exploit the social insects paradigm in a multi-pheromonal point of view. It provides an environment for “external” mobile agents (EMA) and uses mobile internal tool agents (ITA) and static internal management agents (IMA) for its inner operations.
A view of the layer-based architecture of A.N.T can be seen in Fig. 1. Each layer is associated with a distinct population of agents that furnishes services to the superior layers. Eventually, the higher one provides to users the functionalities of the whole system. There is an exploitation relationship between the layers: the mechanisms of the middle layer are based on the work of agents that are hosted in the lower one, while the higher stratum controls and monitors the two others. The lower and middle layers are distributed across the network, as well as the Communication and Intrusion Detection IMAs.

![Diagram](image)

Fig. 1. The layered architecture of A.N.T.

### 3.1.3 Pursuing an ideal: an autonomous stealth IDS

Basic requirements of an IDS, regardless of its inner principles, were described by [7]. Among them, we can find several points very difficult to achieve with a centralized system—agent based or not—, because of the constraints they induce. These points can be handily dealt with a fully distributed framework.

- “The IDS should be adaptive to network topology”.

...
It is difficult to keep track of network topology transformations unless a centralized map is steadily updated. With mobile agents, decisions can be made ‘in situ’ by the moving actors themselves, increasing the overall scalability, configurability and reactiveness of the system. They do not need to refer to a central authority. They don’t even know that the topology has changed, they just deal with what they perceive locally during their migrations. If a central map needs to be built, it can be done easily by gathering the agents topology knowledge, or by sending specific mapping agents through the network.

• “The IDS must be able to operate in a hostile computing environment, exhibit a high degree of fault-tolerance and allow for graceful degradation.”

Because of its critical role, the IDS itself is a primary target. If the centralized part of an IDS is brought down by a severe attack—like a Distributed Denial of Service, for example—the entire system is paralyzed. Bring down a distributed population of defender would require to attack successfully every part of this population. This is harder to do, but not impossible for a well organized attacker. Meanwhile, if every member of this population is mobile, how to attack it? How to damage a process that don’t stay more than handful of seconds on the same host? So the moving parts of the IDS acquire a stealth aptitude that prevent them to be harmed. If every part of the IDS is endowed with such a potential, increasing the overall unpredictability, the entire system sees its vulnerability greatly reduced. We advocate that this aptitude, exhibited by most of the natural distributed systems, can help to strengthen future IDSs and gift them with essential advantages via attack-resistant architectures. In the worst case, if an agent is destroyed, the damages are tiny: this agent won’t lay its pheromonal trail to influence its peers. As these fields are build by the entire population (see Section 3.2) the pheromonal information lost is infinitesimal in comparison with the whole trail. We can compare this to the strength of an anthill: you can destroy many ants, but the overall efficiency of the nest is unaffected. Finally, a regulation mechanism can ensure the stability of the population.

• “The IDS should be capable of providing an automated response to suspicious activity”.

The task of most traditional IDSs is to detect intrusion, but once their job is done they let human decision take the floor. Implementing an automated action of response is certainly not an easy duty. Meanwhile, some of them ([6]) implement some response mechanisms (we henceforth call them IDRS). For a traditional IDRS, such a response involves notifying the central decision core, wait its arbitration, and apply its decisions. With a distributed point of view, an agent can perform a local reaction instantaneously without referring to anyone. If the detected problem goes beyond its competence, it can still call other agents for help. This example illustrates the great

---

4 For example it is trapped on a machine that is shut down.
importance of differentiation in the agents population. It is the base of the modularity at the scale of the multi-agent system. Some small sub-populations, whose size can be dynamically adjusted to the needs of the moment, can be specialized to handle specific attacks. Our future projects (see Section 4) include the use of genetic programming to evolve counterattack agents. By using this mechanism, the IDRS will be able to adapt and tune itself without the need of human hand and in response to the state of the network.

- “The IDS should be able to perform data fusion and be able to process information from multiple sources”.
A multi-population approach is well suited to deal with heterogeneous distributed environment. Through the process of specialization, the IDS can send dedicated agents to firewalls, routers or switches. They can gather information, but in the case of an IDRS they can also update the rules of these devices to take into account the recent events detected. Moreover, as the communication between the updater and the device is local and doesn’t transit necessarily through the network, it is more secure than a distant update. It also reduces the possibility of being subject to insertion and evasion attacks.

- “The IDS should be able to work with other Commercial Off-the-Shelf security tools”.
This case is close to the preceding point. An agent can be dedicated to “talk” to a commercial tool using its specific protocol, allowing an easy interfacing between the two systems. We can even envisage a mobile agent belonging to a population common to several distributed mobile IDSs.

We can see that the benefits of a distributed mobile IDRS are numerous. Let us now present the architecture on which we built the foundations of such a system. We will then discuss its implementation characteristics and reposition ourself regarding the theoretical model.

3.2 Architecture of the system

This section present the architecture of our IDRS, a system mimicking the functioning of natural distributed systems made-up of simple reactive agents. Its purpose is to achieve the efficiency found in natural systems by using emergence and mobility.

The principle is the following: the detection of an intrusion will trigger the release through the network of an alert pheromone in a gradient pattern. Mobile response agents –the lymphocytes–, sensitive to this chemical message, will converge to its source point to initiate a defensive action. More details are presented in Section 3.2.2.

We will now detail the inner components, then illustrate their role with an example.
3.2.1 The components

The pheromone servers

These servers are the cornerstones of our distributed framework. They are sedentary agents that provide an interface between the host itself and the mobile agents. The latters communicate with these servers via meeting connection after having been authenticated as unfaked agents. During this connection, the mobile agent accesses the pheromonal information of the server –only relative to the server’s hosts and to its neighbors– and is also able to add its own deposit.

The server is aware of any information regarding its host (sub-net, what processes are running, who is connected, etc.) so it can make the equivalent pheromonal information available to each agent. For example, it automatically computes its load and codes it in a chemical form that is spread by the internal management agents via the internal tool agents (ITA).

The main functionalities of each pheromone server are embodied in a team of internal management agents (IMA). They are started up by each pheromone server at its beginning time and are in charge of the administration of internal pheromone tables. Their roles are the following:

- **Diffusion management.**
  The diffusion IMA manages how a pheromone diffuses, when the diffusion must be initialized, and how –i.e. at which distance– to perform it. The diffusion itself is ensured by a swarm of mobile internal tool agents launched by the IMA.
  The diffusion IMA of a pheromone server also notices diffusion waves launched by other servers and updates its internal tables.

- **Evaporation management.**
  This IMA is in charge of the gradual weakening of the chemical information stored by its server. This effect allows the system to “forget” bad or old solutions and to avoid uncontrolled convergence.

- **Access control.**
  The communications between the pheromone servers and their interlocutors –regardless of their origin– are monitored by these agents. Their role is to filter the access to the pheromonal information, as well as to log every dialogic act. They employ ITAs to authenticate interlocutors on the hosts they are pretending to come from.

Independently, other IMAs interact with the pheromone server. They reside in the upper layer –see Fig.1–. This layer ensures the interface between the pheromonal services and “external” applications that use its mechanisms. “Intrusion Detection and Response IMAs”, also called watchers, dwell here.

The watcher

Its role is to provide intrusion detection services. This can be done by several ways, providing either host-based or network-based identification, as
well as misuse or anomaly detection:

- Monitoring the behavior of sensitive processes by studying their system calls. The approach reported in [13] is a very interesting mechanism, as it can generate a “percentage of abnormality” that can be easily transformed in pheromone. The statistical models, in function of their size, can be stored locally on each host or in mobile agents ready to answer to a calling pheromone when it is needed by one of the watchers.

- Integrating data of mobile checking agents. Watchers can correlate data gathered by dedicated mobile checking agents, or lymphocyte agents in random checking mode (see Section 3.2.1). This method could implement a multi-point detection to deal with distributed attacks coming from machines within the LAN, with attacks directed toward the entire network, with worms, or with network scans.

- Monitoring network connections at low level by scanning TCP packets.

- Looking for the exploitation of known weak points by checking on local intrusion signatures –trojan files or root kits, for example–, files integrity, or user behavior profiles.

- Interacting with traditional ID mechanisms. Through an interface agent, the watcher can use the results of the several of-the-shelf IDSs –like SNORT [21]– and convert their outcome in a pheromonal information.

- Gathering evidence of the attacker’s activity during the time laps between the detection of the attack and the response of the lymphocyte agents.

Depending on the severity and the type of the alert, the watcher will generate the description of a pheromone. It can include the type of the attack, its gravity –coded in the intensity of the field–, a diffusion rate, a distance of diffusion, an evaporation rate, a degree of confidence, the name of the endangered machine, or whatever information an administrator judges important to incorporate. One important point is that the watcher should watch itself, as well as the pheromone server hosted on the same computer. Once the pheromone is ready to be spread, the watcher won’t contact the pheromone server directly. It will create an antibody agent.

**Antibody agents**

The function of these short-lived agents is to interact with the pheromone server to initiate the spread of a pheromone. We have chosen to embody this functionality in such a mobile agent for security reasons: if the pheromone server is itself the target of the attack, the antibody will migrate to a neighboring machine to use its server. It will also tag the pheromone to indicate that no lymphocyte should trust the pheromonal information of its original server anymore. Once the pheromone server agrees to take in hand the diffusion, the antibody self destructs.
Lymphocytes agents

These agents are the population that retort to the attacking processes. Their behavior is divided in two modes:

- The random checking mode.
  When in this mode, the agents are waiting to scent an alert pheromone but are perceiving nothing. Instead of being idle, they roam the LAN randomly to perform file integrity verification –using TRIPWIRE [23] for example–, or whichever regular checking an administrator could desire. This wandering behavior also increases the probability to perceive an alert pheromone. When it happens –the pheromonal information of every host is checked–, they toggle into the defense mode.

- The defense mode.
  When a lymphocyte finds itself bathed in a pheromone gradient, it migrates towards the source of this chemical. Not all the lymphocyte agents, however, are obliged to answer to all the chemical call of help. They can be specialized and respond only to the attacks they are trained to deal with. They can jump directly to the source machine, or follow an indirect path and climb the gradient towards the source machine. The latter solution allows to remove the name of the source from the pheromonal message. It could be a security constraint. One important point must also be mentioned: every lymphocyte has a small probability to ignore a chemical call, even if it is suited to provide an answer. This behavior inserts a small quantity of noise in the system. It allows the agent to increase their probability to catch a posterior call, paradoxically increasing the response of the system and its resistance to “decoy attacks”. Once the lymphocyte has reached its destination, it can start to react to the intrusion.

Taking defensive action is the main role of the lymphocyte agents. This response can take the form of traditional mechanisms like the interruption of the TCP connection or the update of firewall rules. But mobile defenders can also implement response schemes beyond the abilities of actual IDSs. The mobility of the lymphocyte allows them to initiate a defensive reaction on any component of the network, in a distributed way. The IDRS can then track the attacker’s path through the network –by using MAC addresses or by interrogating routers–, dynamically modify the whole network environment to resist to the incoming aggression, monitor all network traffic to and from the attacker, isolate the target or the attacking machine, or even launch a counterattack directed towards the attacker’s host or network.

The lymphocyte agents can also interact directly with the pheromone servers to emit chemical messages. This feature is important for several reasons. Within the future genetic mixing layer (see Section 4), agents will have to be able to call for mates in order to generate offsprings. Likewise, if a lymphocyte sees that the seriousness of an attack was underestimated by a watcher –i.e. that the intensity of the pheromone was too low–, it can boost it with a new
Fenet diffusion. It can also call for help or notify an administrator if a situation goes beyond its control.

There are several differences between the use of a pheromonal signal and a basic message broadcast.

(i) The chemical diffusion is controlled. When a watcher build a pheromonal signal, two characteristics are carefully chosen: diffusion and evaporation. They influence the shape of the gradient, the maximum distance at which the pheromone will be perceived, and the time curve that describes the disappearance process. Different values will be chosen according to the role of the message: call for immediate help, or just raise suspicion concerning a user in the entire network.

(ii) A broadcast is not relayed by routing devices. As the diffusion process is managed by a swarm of mobile programs, the routers or switches are not a barrier anymore. Moreover, diffusion can be done according to the description of a virtual network architecture independent of the physical connectivity.

(iii) Pheromone fields are additive. When a pheromonal alert is diffused by two neighboring machines under the same attack, the two fields merge and reinforce each other, increasing the probability to attract a lymphocyte agent. These agents will tend to converge on the most severely stricken computer, but will also initiate a response on its neighbor.

(iv) Using a pheromone gradient allows to broadcast a message activating a migration behavior without explicitly mention the destination. Only mobile agents able to climb the gradient are able to reach its source. Hence an effect of information masking that is intrinsic to the concepts of distribution and mobility. This feature is valuable in an environment where sniffing tools can be used by potential intruders.

(v) Pheromone can be altered during the very process of diffusion. Different information can be inserted in the chemical message in function of the physical environment in which the wave is spreading.

Having described its different parts, we will now illustrate the functioning of our IDRS.

3.2.2 How does it work?

It is easier to illustrate the main features of our approach by using an example. The following figures explain the basic sequencing of a reaction to an attack.

Fig.2 represent the IDRS in a guarding state. The watcher is overseeing a process, as well as its network connections. It also monitors itself and the local pheromone server. A lymphocyte agent in random checking mode is performing integrity control on the binary code of the overseen server.

In Fig.3, an intruder is trying to gain unauthorized access. It can do it either from a local process or via a network connection. The watcher detects
the intrusion and generates the pheromone that will be passed to the newborn antibody agent. The latter will then contact the pheromone server, be authenticated, initiate the diffusion of the chemical message, and die. While this time, the watcher is collecting evidences of the intrusion and is trying to trace the attacker to its source machine. This can be done by using a dedicated population of mobile agents that yet has to be built.

Fig. 3 illustrates the response to the attack. The lymphocyte agents have perceived the pheromone field deployed by the pheromone server and have converged on the threatened machine. One of them is locally trying to regain control of the attacked process (Cf. Section 3.2.1). We can also see an example of attacker traced back to its hosting machine. Once the mobile lymphocyte agent moved here, any impeding or retaliating action can be undertaken.

We have described the principles, structure, agents and behaviors our IDRS is based on. However, some components of our system are not implemented yet. A first prototype is currently tested. We will present it in the next
3.3 Actual implementation

The first prototype we are testing is intended as a proof of concept for the architecture, the pheromonal communication medium and the use of full mobility. It is implementing the main mechanisms and interactions, as well as several security-dedicated behaviors. Still, the ID characteristics of the watchers are simulated, and the pheromonal sensitivity of the lymphocyte agents are hard-coded and not dynamically configurable yet. This prototype is developed in Tcl/Tk [17] and uses the framework D’Agents [12] to provide the mobility mechanisms. It allowed us to gain valuable knowledge:

- The main idea of building a fully distributed mobile IDRS is not a chimera. By using a communication medium that permits effective non-direct communication through environment marking, mobile parts of the framework can exchange messages efficiently and independently of their physical location.
- The dynamic adaptation of a system based on mobile entities is very efficient. Our tests have shown near real-time responses and quick self-reorganization as the network topology is modified or computers are restarted. We still do not manage automatic restarting of the servers yet.
- Mobile agents provide a great ability to operate in heterogeneous environments and to be specialized. The interaction between the antibodies/lymphocytes populations and the pheromone servers was easily set up once the communication protocol was established. Likewise for direct access to local hosts informations. The parameters are currently set at the population level, but nothing prevent to adjust independently the behavior of a single specialized agent.
• The resistance of the mobile agents via stealth ability is impressive. We must however yet tune the system between two tendencies: highly mobile resistant agents consuming bandwidth, an less nomadic agents with a reduced network load.

• The size of the mobile agents’ code is currently very small, so the bandwidth consumption is mainly produced by the frequency of migrations. This point will loose importance as the average bandwidth of LANs is getting higher. Nevertheless, a commercial application will contain a larger amount of code, and the mobile ability of the watchers, for example, could be re-examined.

The effect of this prototype was to increase our motivation and the hope we placed in a mobile distributed model. However, we identified some drawbacks.

• The attack-eluding ability based on the mobility can be so efficient that it is annoying. Very mobile processes are difficult to control in a centralized way, and an attack-resistant architecture can also reveals itself as a command-resistant architecture. The consequences of this point are that we must entirely rely on the use of pheromone to control the mobile processes. When we want to kill an agent or replace it by another version, we must attract it on a trap machine before. We use a special pheromone that takes precedence over all other scents. This can be a security flaw. If this pheromone is artificially emitted by an attacker, the entire system can collapse. One way of dealing with this problem is to encrypt and authentify every dialogic act between pheromone servers, watchers and lymphocyte agents.

• Mobile code is heavily exposed to tampering. We were able to sniff the migration of an agent on an Ethernet connection. This point may be beyond the scope of this paper, we are however convinced that encryption is the main solution.

• Increasing the complexity of agents’ behavior and grouping them in different population also increase the complication of coding new agents. The greater the number of different potential interlocutors, the more complex the behavior.

This first prototype is subject to many improvements. We will now briefly describe them.

4 Future work

Our projects can be divided in three main axis:

(i) ID and Response behaviors.
   
   The Intrusion Detection aptitudes of the watchers are currently simulated. Our first project is to plug our work to the model of [13]. The

---

5 For example for a network security administrator.
information of most IDSs are symbolic and give birth to relatively big pheromonal descriptions. The measures of anomalous behaviors is based on the calculus of a numerical distance, much more suited for a pheromonal coding. This project will also let us test our architecture in real ID conditions for the first time.

Actual lymphocyte agents are specialized in reacting to specific attacks. One important enterprise will be to build a genetic framework to explore the potentialities offered by more “ALife-oriented” approaches. We hope to exhibit self-specialization faculties and attain self-regulation effects observed in natural systems.

(ii) Interface and interoperability.
We intend to develop interface agents able to exchange information between our system and current off-the-shelf products.

One big problem we had to face up is the centralizing of information relative to the state of the distributed moving brick of the system. This problem has been dealt with by introducing appropriate information-sending code into agents and by using a central tracker which role is to provide an image of the network to a central observer and to configure and initialize the system in a running state. We must now develop a neat GUI to pilot this tool.

(iii) System considerations.
All the processes involved in our framework actually run as high level interpreted code. We plan to study which parts could be inserted in kernel space.

5 Conclusion

We present an architecture for Intrusion Detection and Response Systems based on mobile agents. It is inspired from natural distributed systems and uses environment marking techniques for non-direct communication between mobile processes. The architecture is based on the interaction of several populations of agents that are described and whose roles are explained.

We relate the inner functioning of our system and picture it with an example including intrusion detection and response behavior. We also demonstrate the feasibility of this model by the implementation of a working prototype. The differences between the actual implementation and the theoretical model are also stated.

Finally, the characteristics of our distributed mobile architecture are depicted, as well as its advantages and drawbacks in comparison with traditional approaches.
References


