New Collaborative Intrusion Detection Architecture Based on Multi Agent Systems

Mohssine EL AJJOURI, Siham BENHADOU and Hicham MEDROMI

*Systems Architecture Team, Laboratory of Computer System and Renewable Energy, University Hassan II Casablanca ENSEM.
Casablanca 20200, Morocco*

Abstract: One of the most serious security challenges is to learning new attacks. Thus, the concept of learning in existing IDSs used in general to learn the normal behavior of the system to secure, then the IDS does not have the ability to detect new attacks. In this paper we propose a new architecture for intrusion detection based on multi agent systems. The agents incorporate a CBR (case-based reasoning) technique which is equipped with a learning and adaptation capacity. We present also the motivation and description of the approach. The proposed architecture is based on a hierarchical and distributed strategy separated into three layers. We focus after on the modeling of our multi-agent systems, we use the O-MaSE (organization based multi-agents system engineering) methodology.

Key words: Multi-agent system, intrusion detection, learning, case-based reasoning, o-MaSE methodology.

1. Introduction

Computer networks are increasingly likely to be the target of various disorders, against their security, such as congestion, access and malicious attacks [1]. For this purpose, it becomes inevitable to bring these systems tools and mechanisms able to inhibiting these disorders.

To detect any attempt to violate the security policy, permanent or regular monitoring systems can be implemented: it is the IDS (intrusion detection systems). The intrusion detection systems have become widely deployed in information systems and they won an important place in the design of the security strategy.

Despite the reputation of these tools, most intrusion detection systems are monolithic and centralized while collecting data on the network is distributed [1]. In addition, computer networks have become increasingly complex and therefore the number of vulnerabilities discovered in computer systems can be significant.

Thus, attacks against these vulnerabilities can be both varied and complex, they are therefore not suitable for dynamic environments and that is their main weakness, in this type of environment where security needs are constantly increasing.

In this context, and to overcome these problems, we propose intrusion detection architecture able to working in intelligence with their environment by exploiting the notion of learning agents.

Several methodologies enabling the development of MAS (multi agents systems) have been proposed. Among them, we chose the methodology O-MaSe to develop our MAS. This methodology is selected for its simplicity.

The paper is organized as follows: Section 2 introduces some theoretical background; Section 3 shows our system architecture and design; Section 4 is the conclusion and our perspectives.

2. State of Art

2.1 Intrusion Detection

Intrusion detection system is a device to monitor the activity of a network or a given host to detect intrusion attempts and possibly react to this attempt [2].

Corresponding author: Mohssine EL AJJOURI, Ph.D., research fields: intrusion detection systems, multi agents system.

Fig. 1 Simplified model of intrusion detection system [3].

Debar simplified the intrusion detection system in a detector which analyzes the information from the monitored system (Fig. 1).

There are two kinds of IDS systems, host IDS and networks IDS:

- The host IDS analyze the operation and status of machines on which they are installed to detect attacks based on demons (such as syslogd for example). The system integrity is checked periodically and alerts can be sent.
- Networks IDS, meanwhile, analyze real-time traffic they aspire to a probe (NIC in “promiscuous” mode). Then, the packets are then analyzed. In case of intrusion detection, alerts can be sent.

2.2 Methods of Analysis

IDS have two different approaches to detect intrusions:

- Signature-based IDS:

Typically, the IDS networks are based on a set of signatures each representing the profile of attack [4].

This approach consists of searching in the activity of the monitored element (Network Stream) the footprints of known attacks.

A signature is usually defined as a sequence of events and conditions relating an intrusion attempt. Recognition is then based on the concept of “pattern matching” (analysis of strings present in the package, research correspondence in a knowledge base). If an

attack is detected, an alarm can be raised (if the IDS is in active mode, otherwise it simply archive the attack) [5].

- Behavioral IDS:

Behavioral IDS main function is anomaly detection. Their deployment requires a learning phase, the tool will learn the “normal” behavior of application flows present on its network [6].

2.3 Multi Agents Systems

The agent concept is used in a general sense to refer to a natural person, a substance or a living organism. It’s an entity acting in an environment where other similar entities exist but not necessarily identical [4].

We can identify the following characteristics for the concept of agent [7]:

- Situated: the agent is able to act on his environment from the sensory entrances which he receives from the same environment.
- Autonomous: the agent is able to act without the intervention of a third (human or agent) and checks its own actions as well as his internal state [8].
- Reactive: agents perceive their environment and respond to changes that occur there. Reactivity also means the ability of an agent to alter its behavior when environmental conditions change.
- Proactive: The agent must exhibit a proactive and opportunistic behavior, while being able to take the initiative at the right time.

- **Adaptive:** The agent must be able to adapt to the environment in which it is located, and control abilities (communication and behavioral) depending on the environment in which it operates and by the agent with which it interacts.

There are three models of agents: the deliberative agents, reactive agents and interactive agents. We will briefly introduce the different types of agent models.

Deliberative agents: Deliberative agents have the ability to solve complex problems. They are thus able to reasoning on a knowledge base.

Reactive agents: Reactive architectures, also known as behavioral, are characterized by agents that have the ability to react quickly to simple problems, which do not require a high level of reasoning [9].

Interactive agents: architectures of interactive agents are characterized by agents specifically designed with capacities of coordination and cooperation [10].

2.4 A Case-Based Reasoning Technique

CBR (case-based reasoning), is an approach able to use the specific knowledge of problems (case) having already occurred in past to answer new problems. A new problem is solved by looking for a past similar case, and by reusing its solution in the present problem (Fig. 2) [11].

This definition shows that the CBR is a very natural algorithm: as a human, when a problem arises, the

idea first is to consider whether this problem (or similar event) has already occurred, and if necessary, resume the solution passed to try to adapt myself to this problem. The main steps of a CBR algorithm are [11]:

- **Retrieve:** The first step is to look in the database of issues, the case most similar to the present problem.
- **Reuse:** The second step is to reuse the previous solution for the current problem.
- **Revise:** The third step is to adapt the previous solution to the current problem, to obtain a new solution.
- **Retain:** The final step is adapted to store the solution in the case of the database.

Here is the diagram illustrating the steps of the algorithm in Fig. 2.

3. Our System Architecture and Design

3.1 Related Work

The existing detection systems have been designed for environments. They are therefore not adapted for dynamic environments and this is their main weakness. In this type of environment where security needs are constantly increasing, flexibility and adaptability become essential.

Our starting point was to make a comparative study of existing architectures of intrusion detection systems based agents, we have identified weaknesses in existing

Fig. 2 The CBR engine [11].

systems compared to changing security needs. It is interesting to note that these architectures studied may provide for many extensions, for example learning normal behavior of users, reduce the number of positive alarms.

3.2 Motivation of the Proposed Approach

In existing detection systems intrusion, the concept of learning is used to learn the normal user profiles and systems to secure. These normal profiles are then compared with the current profiles to detect potential attacks. In our architecture, we propose to use the notion of learning to learn abnormal profiles that correspond to attacks.

For an intrusion detection system, it is important to learn new structures of attacks to detect new attacks when they occur.

To add this feature, agents will use existing attack patterns and attacks have observed in the past to identify similarities with a current suspicious sequence of event which does not correspond to a

normal sequence, or a known attack.

3.3 Components of the Architecture Based Agents

The proposed architecture is based on a hierarchical and distributed strategy where features are structured and divided into 3 main layers: monitoring layer, classification and management layer.

Agents in each layer are dedicated to specific tasks; they can run regardless of their physical location, because of their own abilities as the collection, classification and data visualization. This division ensures the allocation of specific tasks according to the degree of complexity, while maintaining each level independently, allows new changes to be easily adapted.

This architecture also greatly helps balance the workload between the different entities and therefore avoids the centralization of traffic.

Different types of agents that are at different levels of the architecture can be described functionally as follows (Fig. 3):

Fig. 3 Architecture of intrusion detection proposed.

- In the monitoring layer :

Agent Sniffer: this agent is placed at the entrance of our architecture, whose goal is to capture each packet of data flow and cut into segments to send a subsequent treatment, it's a reactive agent.

Preprocessor Agent: after the division of traffic data, segments products are pretreated before analysis. Once the data have been pretreated, for this analysis a new piece of data is requested, it takes the raw packets, checks them against certain types of behavior (plug-in) such as HTTP plug-in etc. Once the packet was observed to have a particular behavior, it is forwarded to the Agent Filtering.

- In the classification layer :

Filter Agent: at this level a first analysis is performed and some suspicious events of low complexity will be detected according to the objectives of the agent who consults the rule base, but some events will be passed, an alarm will be triggered, the event will be stored in a log file.

Engine CBR Agent: This type of agent is also located in the layer classification, it's an essential component of the architecture, it is at this level that our approach will apply, the CBR algorithm will calculate the degree of similarity between past attacks and new attacks to see if the rate is high (past attack) or not (new attack), this agent is in constant interaction with the database of patterns attacks.

Decider Agent: The objective to achieve for this agent is to make sure by basing on the previous algorithm that if the degree of similarity is high, in this case the attack is already known consequently a classic launching alarm will be made.

Otherwise, it is indeed a new attack to our model, it is not known, in that case, an intervention of the generator agent of plan of attack will be done.

- In the administration layer :

Generator Agent: if it's a new attack, at this level that this agent work, the purpose is to create a new plan of attack, it will be validated by the administrator, consequently the safety policy will be updated.

MSP Agent (agent management security policy):

This agent interacts with the administrator and retrieves the attack pattern and therefore powers the DB of attack pattern, so we learn a new event.

3.4 Design of Our Architecture

To model our MAS, we used the methodology O-MaSE (organization based multi-agents system engineering) [12]. It's a new approach in the analysis and design of agent based systems. It is adapted from the OPF (OPEN process framework) [13] and takes some useful concepts from the MaSE methodology [13]. The aim of O-MaSE is to allow designers to create customized agent-based software development process based on project-specific requirements [13]. O-MaSE uses an integrated meta-model based framework that allows designers to select method fragments from a repository and construct custom processes using construction and tailoring guidelines [13]. Such as meta-model based framework is supported by a three-layer schema as shown in Fig. 4.

We present the main diagrams, namely the diagram of goals, the role diagram, diagram agents, plans diagram and protocols diagram [13].

- Goals diagram :

In our problem, the main purpose is to detect intrusions with learning new unknown attacks. This objective is the overall goal written Goal 0. The purpose 0 depends on the realization of two purposes son who are the detection of the intrusion and the update consequently the base of attacks. To detect intrusion (Goal 14) we need to capture packets (Goal 14.3), analyze (Goal 14.2) and compare with a rules base (Goal 14.1).

To updating the knowledge base of attacks (Goal 13) requires the creation of a new plan of attack (Goal 13.1) and then adding this new attack (Goal 13.1), this goal is created after the plan of attack is received (Goal 13.2.1) and confirmed by the administrator (Goal 13.2.1). To create a new plan of attack (Goal 13.1) we must compute the similarity between past

Fig. 4 O-MaSE methodology framework [13].

Fig. 5 Diagram of goals.

attacks and new attacks (Goal 13.1.2), and make after a comparison (Goal). To compare the similarity requires the consultation of the database attacks (Goal 13.1.1.1).

- Role diagram :

In this diagram (Fig. 6), our goal is to create the role diagram that describes the different roles can be played by agents in the system.

For each goal/sub-goal previously identified, we must create a role to realize it. A role in this case can

achieve two goals simultaneously.

To achieve a goal, a role should have at its disposal one or more capacities that translate, usually by execution plans.

- Agent diagram:

The diagram of agents is created, so that there is at least an agent class that has all the necessary capabilities to play each role. The philosophy of the methodology provides a O-Mase design flexibility. More done to separate the role of the agent itself reduces

Fig. 6 Role diagram.

Fig. 7 Agents diagram.

number of entities in the implementation system.

Indeed, there will be a minimum number of agents that ensure good operation of the system and change the roles depending on the situation. Thus, an optimization of the use of memory resources is ensured.

We need agents who collaborate between them for the smooth running of our architecture.

From diagram (Fig. 7), we begin to be able to see the overall structure of our architecture and its interaction with the external environment.

Plan diagrams

- Plan “Detect Intrusion”

Modeling execution plans can describe the algorithms used by the agents to achieve an objective or set of objectives. We detail in this section diagrams

Fig. 8 Plan diagram “detect intrusion”.

Fig. 9 Plan diagram “create new plan of attack”.

capability plan identified above.

Following receipt of the data stream, an analysis will be made after consultation of the rule base. If the suspicious event is low complexity it will be detected, an alarm trigger is done, and the event will storied. Otherwise, the flow will be given past for subsequent treatment (Fig. 8).

- Plan “Create new plan of attack”

In this plan the CBR algorithm will calculate the

degree of similarity between past attacks and new attacks to see if the rate is high (past Attack) or not (New attack) (Fig. 9).

- Plan “Add new plan of attack in the data base”

In this plan, a new plan of attack will be created, approved by the administrator, manager security policy agent interacts with the administrator and retrieves the plan of attack and update the data base of attacks (Fig. 10).

- Protocol diagram

The design of the cooperation between agents is defined by a sequence of message exchange. These conversations are defined protocols called protocols of cooperation or protocols conversations, which indicate the sequences authorized messages.

A protocol involving two agents is described by a sequence diagram representing interactions between the different entities in the order of showing messages exchanged.

- ✓ Protocol diagram: “Detect intrusion”
- ✓ Protocol diagram: Create new plan of attack
- ✓ Protocol: Add new plan of attack

Fig. 10 Plan diagram “add new attack in the data base”.

Fig. 11 Protocol diagram: Detect intrusion.

Fig. 12 Protocol diagram “create new plan of attack”.

Fig. 13 Protocol diagram “add new plan of attack”.

4. Conclusions

The combination of MAS and CBR technique allows us to have architecture of intelligent intrusion detection with features such as the ability to learn and reasoning. Thus, our proposed model promotes reuse of knowledge in operating problems (cases) having already happened in the past to solves new problems.

The model, for its architecture based on multi-agent systems present a distribution of sensing activities and effective interoperation between agents.

It is important to note that in contrast to a monolithic system, our model has better scalability, this greatly helps balance the workload between the various entities and avoid the centralization of traffic which helps optimize treatments.

Our starting point is to propose a new architecture of intrusion detection to address the weaknesses of existing systems with respect to changing security needs such as learning abnormal behavior.

Our future work will be to implement our architecture in order to test the various features and ensure that the objective is achieved.

References

- [1] Ghenima, B. 2007. "Un IDS Reparti Base Sur Une Societe D'agents Intelligents." Memoire de Magister, Universite M'hamed BOUGARA de Boumerdes.
- [2] Baudoin, N., and Marion, K. 2004. " NT Reseaux: IPS et IDS." Universite de Marne la Vallee, France.
- [3] Debar, H., Dacier, M., and Wespi, A. 2000. "A Revised Taxonomy for Intrusion Detection Systems." Presented at the Annales des Telecommunications.
- [4] Lasheng, Y., and Mutimukwe, C. "Agent Based Distributed Intrusion Detection System (ABDIDS)." Central South University (CSU), Department of Computer Science, Changsha, 410083, China.
- [5] Dipankar, D. 2005. "The Use of Agent Technology for Intrusion Detection." University of Memphis, WL041/Bidgoli, September 20.
- [6] Dasgupta, D., Gomez, J., Gonzalez, F., Kaniganti, M., Yallapu, K., and Yarramsettii, R. "Multilevel Monitoring and Detection System." Intelligent Security Systems Research Laboratory, Division of Computer Science.
- [7] Ravneet, K. 2012. "Role of Cross Layer Based Intrusion Detection System for Wireless Domain." *International Journal of Communications, Network and System Sciences* 5: 81-5.
- [8] Eugene, H. S., and Zamboni, D. "Intrusion Detection Using Autonomous Agents." Center for Education and Research in Information Assurance and Security, 1315 Recitation Building, Purdue University, West Lafayette, IN 47907-1315, USA.
- [9] Alvaro, H., and Corchado, E. "Multiagent Systems for Network Intrusion Detection: A Review." *Computational Intelligence in Security for Information Systems* 63: 143-54.
- [10] Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., and Yarramsettii, R. "An Agent-Based Intrusion Detection System." Intelligent Systems Research Laboratory, Division of Computer Science, the University of Memphis, Memphis, TN 38152, United States.
- [11] Aamodt, A., and Plaza, E. 1994. "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches." *AI Communications* 7.
- [12] Mguis, F., Zidi, K., and Ghedira, K. 2012. "Modelisation D'un Systeme Multi-agent Pour la Resolution D'un Probleme De Tournees de Vehicules Dans une Situation D'urgence." Presented at the 9th International Conference on Modeling, Optimization & SIMulation, Jun, Bordeaux, France.
- [13] Firesmith, D. G., and Henderson, S. B. 2002. *The Open Process Framework: An Introduction*. Harlow-England: Addison-Wesley.