A New Distributed Intrusion Detection System Based on Multi-Agent System for Cloud Environment

Omar Achbarou, My Ahmed El Kiram, Outmane Bourkoukou, Salim Elbouanani

Computer Science Dept, Laboratory ISI, Cadi Ayyad University, Morocco

***Abstract*:**Cloud computing, like any distributed computing system, is continually exposed to many threats and attacks of various origins. Thus, cloud security is now a very important concern for both providers and users. Intrusion detection systems (IDSs) are used to detect attacks in this environment. The goal of security administrators (for both customers and providers) is to prevent and detect attacks while avoiding disruption of the smooth operation of the cloud. Making IDSs efficient is not an easy task in a distributed environment such as the cloud. This problem remains open, and to our knowledge, there are no satisfactory solutions for the automated evaluation and analysis of cloud security. The features of the multi-agent system paradigm, such as adaptability, collaboration, and distribution, make it possible to handle this evolution of cloud computing in an efficient and controlled manner. As a result, multi-agent systems are well suited to the effective management of cloud security. In this paper, we propose an efficient, reliable and secure distributed IDS (DIDS) based on a multi-agent approach to identify and prevent new and complex malicious attacks in this environment. Moreover, some experiments were conducted to evaluate the performance of our model.

***Keywords*:** Cloud computing, Intrusion detection system, Distributed system, Multi-agent systems, Mobile agent

1. Introduction

Cloud computing is based on the logic of consumption of service, implying that responsibility for the deployment, control, management and maintenance of the infrastructure, platform or software is the responsibility of the cloud service provider (CSP)[1]. Despite the enormous technical and commercial benefits of the cloud environment, security and privacy concerns are the main obstacles to its widespread adoption around the world, and particular attention should be paid to security when choosing a cloud service. In view of these security concerns, the integration of an IDS can be important for detecting attacks or other activity that can be considered suspicious or illegal.

Existing IDS solutions have been developed for conventional networks and systems, but are not easily adaptable to a dynamic environment such as cloud computing. Thus, it is necessary to develop a flexible, secure solution that is adapted to the changing and complex evolution of the cloud environment. Although IDS models have been proposed in the research literature, IDS components alone are not able to parse all of the large reports generated. Thus, these proposed solutions remain limited due to their insulation; in other words, they are not able to collaborate or cooperate with each other. Their detection results are therefore isolated, and cannot be collected and analyzed systematically. Thus, there is a need for IDS solutions based on the concepts of collaboration, cooperation, autonomy and dynamism; these concepts are needed to detect attacks effectively and to respond to intrusions by reducing response time.

In this work, we propose a solution that meets these requirements in the form of a multi-agent system-based distributed IDS (MAS-DIDS) that can identify and prevent all anomalies in a cloud environment. This system is based on a distributed architecture of IDSs that work in collaboration and communicate with each other, in order to adapt to the complexity of cloud networks. Each IDS is composed of a group of dynamic, responsive, and cooperating agents which work together to make the IDS more autonomous and flexible. The main objective of our research work is to implement a MAS-DIDS that combines the two techniques of signature-based and anomaly-based intrusion detection, in order to block both known and unknown attacks within a complex, dynamic and changing environment. Finally, the efficiency and performance of the proposed model are studied in terms of different metrics: detection rate (DR), false positive rate (FPR), and response time.

The rest of the paper is organized as follows. The next section presents a theoretical background, in which we describe the main concepts of cloud computing, IDS and our types, and multi-agent systems (MAS). We discuss several related works in the area of multi-agent IDSs in Section 3. Section 4 forms the core of this paper, and explains and describes our proposed model in detail. Section 5 presents the details of a performance evaluation and the effectiveness of our proposed model based on an experimental study. The final section summarizes the main contributions of this work.

2. Related Work

In the literature, there are many works that use an IDS with the agent approach to secure systems against attacks. However, most of these studies have developed solutions for well-defined networks and systems, and are not suitable for dynamic and complex environments such as the cloud environment. Agent-based IDS implementation is one of the new paradigms for intrusion detection in this environment, and this approach has been examined by several researchers.

Mohamed and Abdullah [2] have proposed a secure model using a mobile agent that was well-prepared with a required database. The agent consists of five processes that assess different scenarios in the wireless ad-hoc domain, and the monitoring, classification, detection, isolation, and recovery parts. This agent is configured to take a snapshot of a data recovery file when successfully attached to a new node that intends to join the wireless domain. This intelligent agent contains the gene profile, non-self profile, and detector profile. They are distributed to all nodes inside the domain upon connection.

In their article, Venkataramana and Padmavathamma[3] introduced a multi-agent intrusion detection and prevention system using agents for the detection of attacks in the cloud.

Chang and Zhu [4]an immune network algorithm, his proposition improves agent communication, but has had few contributors. Their model is composed of a set of agents in each host that cooperate, distribute and coordinate with each other to detect attacks. For self-security, the agents send keep-alive messages to their neighborhoods. The immune algorithms are used in the sandbox, study unknown intrusions and generate rules used to detect intrusion.

Singh Hada et al.[5] have proposed a secure system for cloud environment which uses agent technology as security agents to acquire useful information from the node which the user and service provider can utilize to keep track of privacy of their data. In this work, the mobile agents can dynamically move in the network, replicate itself according to requirement and execute the requested tasks such as accounting and monitoring of virtual machines for monitoring virtual machine authenticity and integrity.

In [6], the authors proposed a trust model that used mobile agent technology. In this work, mobile agents can dynamically move across the cloud network to perform certain tasks, such as accounting and monitoring the integrity and authenticity of virtual machines.

In [7], the authors have proposed a line of defense by applying Mobile Agents technology to provide intrusion detection for cloud services regardless of their positions. These works build up a robust distributed hybrid model scaled, flexible and cost-effective method based on mobile technology. Virtual machines are attached to mobile agents which detect of an attack from all the attacked instances for further auditing and analysis. This system is limited to the detection of attacks at instances. They did not think to monitor network traffic simultaneously.

Depren et al. [8] have proposed an intelligent intrusion detection system using both anomaly and misuse detection techniques, to enable a computer networks to handle attacks.

In [9], the authors have developed a collaborative system based on Hybrid-IDS and mobile agents in Cloud computing, to define a dynamic context which enables the detection of new and known attacks in this environment.

Wang and Zhou [10]presented the concept of a cloud alliance, involving communication between agents and the exchange of mutual alerts, primarily to resist DoS and DDoS attacks.

In [11], an IDS based on mobile agent technology and cryptographic mechanisms has proposed by Idrissi et al. This proposal consists on elaborating detection mechanisms, based on cryptographic traces generated by mobile agent to secure CC architecture against insider threats.

Authors Seresht and Azmi [12] proposed a hybrid IDS that analyzes the network traffic in the system environment, this analysis is performed by using virtual machines. Indeed, each instance is composed by intelligent agents to perform a defined selection algorithm. These agents communicate and cooperate with others to detect anomalies.

A thorough study of security solutions based on agent technology reveals IDS solutions that use the different properties of intelligent agents to detect attacks and respond to intrusions. Existing solutions are poorly suited to the growing complexity of cloud networks; they use centralized and non-collaborative IDSs and are not suitable for dynamic environments. Thus, they are not able to cooperate and communicate with each other to detect complex attacks. For example, if an IDS detects a new attack, it does not share this result with other IDSs in its environment.

In thispaper, we therefore propose a secure solution that meets all these requirements in the form of a DIDS based on an MAS, which can identify and prevent all attacks in a cloud environment.

3. Theoretical Background

The first part of this section defines the central concept of cloud computing, the second part describes intrusion detection systems, and the final part presents the multi-agent approach.

3.1Cloud Computing

Cloud computing (CC) is a set of virtualization technologies, providing infrastructure, platforms and applications to the customer on demand. Generally, CC involves:

- Computing "as a service", that is, on demand;
- An environment based on virtualization;
- A structure composed of three layers: infrastructure, platform and application;
- "Self-service" on a pay-as-you-go model;
- Abstraction, pooling and dynamic allocation of physical resources.

The National Institute of Standards and Technology (NIST) has defined a CC model with five characteristics, three service models and four deployment models[13], as shown in Figure 1.

The architecture of a cloud environment can be divided into three layers: the infrastructure layer (infrastructure as a service, or IaaS), the application layer (software as a service, or SaaS) and the platform layer (platform as a service, or PaaS)[14]. Each layer represents a different part of the CC stack.

- IaaS: In this layer, the cloud provider offers infrastructure, machines and other resources on demand;
- PaaS: The PaaS model consists of providing hardware and software tools as a service, enabling business users and developers to rapidly create applications;
- SaaS: This service model is characterized by the use of a shared application that runs in a cloud environment.

The cloud can also be considered in terms of five component architectures: infrastructure, servers, platforms, applications and clients. Cloud deployment models are generally categorized by the type of deployment of this environment [15][16].

- A private cloud is a cloud platform operated for a specific organization;
- In a public cloud, the cloud provider offers their resources as a service to the general public;
- A community cloud shares infrastructure among multiple organizations within a specific community with common concerns;
- A hybrid cloud is a combination of cloud deployment models (public, private, and community) that attempts to address the limitations of each approach.

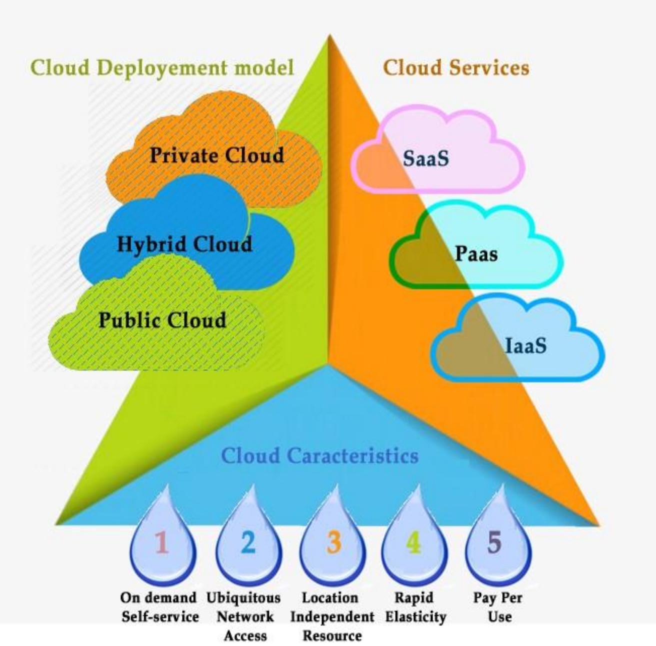


Figure 1. The architecture of a cloud environment

3.2 Intrusion Detection Systems

IDSs are software or hardware components that detect intrusions, log information about these intrusions, and generate alerts or execute predefined procedures[17]. IDSs play an important role in detecting and resisting all intrusions or attacks. Figure 2 shows different classifications of IDSs.

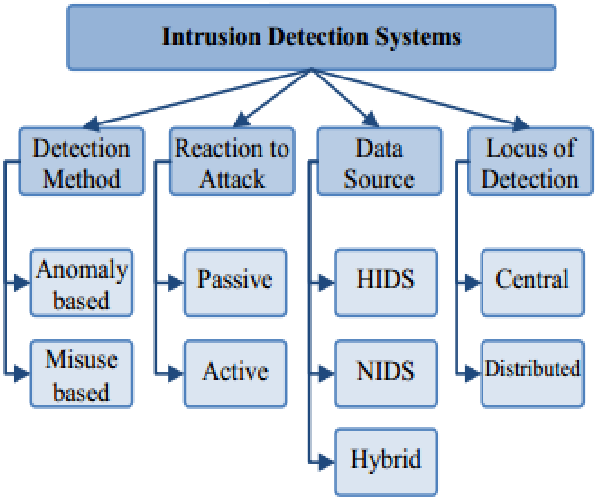


Figure 2. Classifications of IDSs

From the viewpoint of data source, there are three types of IDS in the cloud environment: host-based (HIDS), network-based IDS (NIDS) and Distrusted IDS (DIDS). An HIDS is an agent that monitors and analyzes any action, internal or external, that bypasses the system security policy, while an NIDS attempts to detect unauthorized access to a network by analyzing the network traffic for signs of malicious activity and anomalous events[18]. A distributed IDS consists of a several IDSs in the cloud network communicating with each other, or with a central point that manages that system. By distributing these cooperative IDSs on this environment to process and to analyze the collected events[19].

An IDS increases the security level of a cloud by using two main intrusion detection techniques[20]; the first is based on signatures (signature-based detection or misuse detection) and the second on behaviors (anomaly detection).

- A signature-based detection technique detects attacks by verifying that observations match known attacks. This technique therefore uses a knowledge base for the different existing attacks[21]. This principle of intrusion detection is reactive and meets several constraints; the IDS only detects attacks that have been defined.
- An anomaly detection technique is based on research on abnormal behavior, and anything that deviates from normal conditions triggers an alarm [19]. This type of detection is effective on unknown attacks but can generate a large number of false positives.

Some IDSs combine both techniques to achieve betterresults. This is approach used in our proposal, which incorporates both techniques.

3.3Multi-Agent Systems

An MAS is a system consisting of a large number of agents interacting autonomously in a dynamic environment[22]. The roles and actions of the agents must be clearly defined to solve one or more defined problems.

An agent is a real or virtual entity that can act autonomously and flexibly to achieve its goals in its environment [23]. An agent can be characterized by six properties:

- **Autonomy**: An agent should have the ability to act and cooperate without human intervention;
- **Reactivity**: An agent should be able to perceive the environment and react to changes, for instance in terms of the modification of the defined objectives or the resources available;
- **Proactive**: An agent must be able to identify the purpose of a directed behavior by taking the initiative;
- **Sociability** and **communication**: An agent should have an ability to interact with other agents. The most common agent communication protocol is Agent Communication Language (ACL);
- **Learning**: An agent should be able to memorize experiences and adapt its behavior accordingly;
- **Mobility**: An agent should be able to move from one machine to another.

4. The Proposed MAS-DIDS System

4.1System architecture

We propose an MAS-DIDS architecture, as shown in Figure 3, with a distribution and cooperation mode, which detects known attacks or new types of attack in a distributed cloud environment. This architecture is composed of a group of intelligent agents with mobility and responsiveness, which can communicate and cooperate with each other in order to effectively detect coordinated and distributed attacks in this environment.

First, as the network administrator, the CSP (Cloud Service Provider) receives the packets from different CSUs (Cloud Service User). The CSP transfers these packets to the MAS-DIDS system for analysis and detection of attacks. The first component to receive the packets is the central console; it receives packets from the CSP and transfers them to the supervisor agent (SA), which also checks and analyzes the packets before sending them to the available IDSs (IDS-1, IDS-2, ..., IDS-n) in the system. The IDSs use an interface agent (IA) as a network capture and analysis tool, allowing the capture results to be saved in a file entitled “SniffingFile.cap” for analysis by the analysis agent (AA).

The AA also communicates with the IA to parse and filter the list of packets using signatures (fingerprint attack). Then, the AA routes the hashed packets to a signature-based detection agent (SDA). This node is responsible for checking each signature in the local database (LDB), coordinating with the rules agent (RA). Two results are possible after checking a signature with RA: either the signature exists or it does not. When a signature exists in the LDB, the SDA concludes that this is proof of an ongoing known attack, and an alert is generated to initiate a response. However, when a signature does not exist in an LDB that is currently synchronized with the global database (GDB), the current packet is transmitted to the anomaly detection agent (ADA). The goal of the ADA is to detect anomalies through an analysis of possible abnormal behaviors; on this basis, it can classify a current packet as an unknown attack or a false positive. The rules obtained are automatically transmitted to the RA to update its LDB. Finally, this system generates alert reports to communicate with the central console. The latter sends these reports to the CSP, which blocks the source(s) of the detected attacks.

4.2 System components

The structure and the detailed functionalities of the different components of our proposal are described below.

CSP and CSU: As the network administrator, the CSP receives packets (TCP, UDP etc.) from different CSUs (internal or external users). It then transfers these packets to the MAS-DIDS system for analysis in terms of an attack.

Central console: The first component to receive packets is the centralconsole, which acts as the administrator and controller of the MAS-DIDS. It performs the following tasks:

- It sends the user packets to the SA;
- It receives responses from the SA;
- It sends alert reports generated by agents to the CSP;
- It ensures the secure execution of agents in the system**.**

SA: The role of this agent is to provide all the necessary information to each agent, and it controls and manages all the other MAS-DIDS agents. If the SA detects that a modification of an LDB (LDB1, LDB2 etc.) has been triggered by the RA, it automatically updates the GDB and then synchronizes its contents with all the LDBs in the system. All LDBs are enriched by the GDB, which is a central repository of the last version of the rule databases.

SDA and **RA**: The SDA is a smart agent which processes requests from the AA. This type of agent uses a signature-based detection technique. The RA is the agent responsible for investigating whether there is a similarity between the current packets and the attack fingerprints that are available in its LDB. If this is the case, the SDA triggers an attack alert for the CSP to block the source of the attack. The RA also requires frequent updates to its LDB after any change in GDP, or following the detection of a new attack by the ADA.

ADA: This agent uses an anomaly-based detection technique, which detects an intrusion according to the past behavior of the user. To do this, the IDS must first create a basic profile using self-learning mechanisms. After this learning step, the ADA agent begins to compare the traffic with the profile it has created. It triggers an alert when out-of-profile events occur. In order to avoid false positives, the ADA communicates the alert triggered to the SA, which classifies the alert by applying the following formula:

$$\frac{\#number\ of\ ADAs\ sending\ the\ same\ alert}{\#number of ADA sin the system} > 0.5 \qquad (1)$$

If the result is greater than 0.5, the packet is classified as a new type of attack to block. On this basis, the SA allows the rules obtained to be automatically added to the GDB, and communicates the alert to the CSP via the central console, in order to block the source of the detected attack. Then sent the signature of the detected attack to RA for updating its LDB.

4.3 The Proposed Procedure of the Negotiation Model

The used negotiation model is based on three main phases (Figure 4): capture and queuing, analysis and processing and reporting phase.

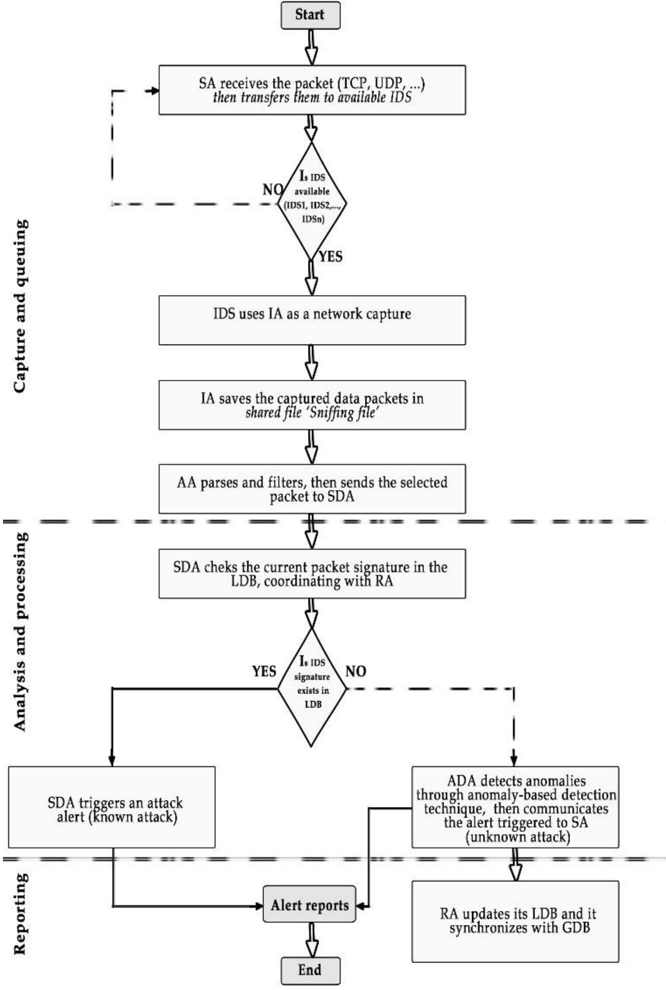

Start
SA receives the packet (TCP, UDP, ...) then transfers them to available IDS
NO
Is IDS available (IDS1, IDS2,..., IDSn)
YES
Capture and queuing
IDS uses IA as a network capture
IA saves the captured data packets in shared file 'Sniffing file'
AA parses and filters, then sends the selected packet to SDA
Analysis and processing
SDA cheks the current packet signature in the LDB, coordinating with RA
YES
Is IDS signature exists in LDB
NO
SDA triggers an attack alert (known attack)
ADA detects anomalies through anomaly-based detection technique, then communicates the alert triggered to SA (unknown attack)
Reporting
Alert reports
RA updates its LDB and it synchronizes with GDB
End

Figure 4. Procedure of the negotiation model for MAS-DIDS system

Capture and queuing: the function of this phase is received packets (ICMP, TCP, IP, UDP) and saved results in shared queue (sniffing file) for analysis.

Analysis and processing: the major functions to consider in this phase include filtering, checking and analyzing the packets of the shared queue. Through an efficient matching and analysis by AA, SDA, RA, and ADA the bad packets would be identified and will trigger alerts. the detailed algorithm of this phase for detecting attack is present in figure 5.

Reporting: the function of this phase is to prepare alert reports, an information report for the CSU and the other complete report for the CSP.

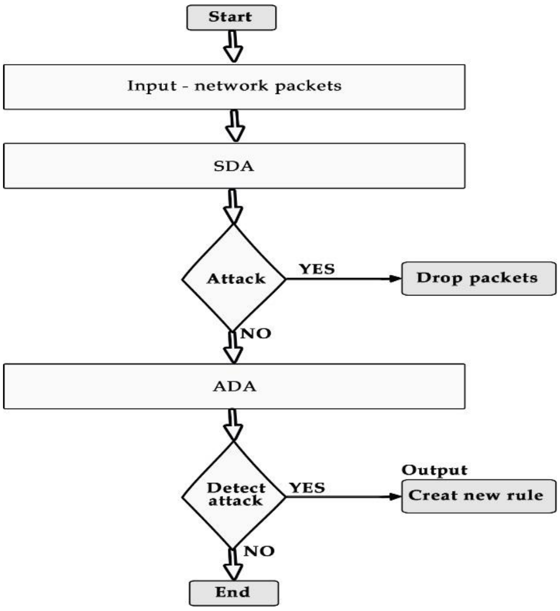


Figure 5. Algorithm for detecting attack

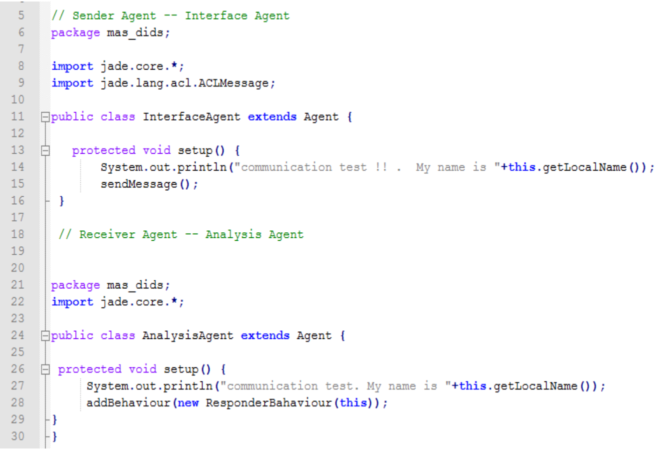
4.4 The Proposed Negotiation Protocol

The negotiation protocol defines how the agents can interact and communicate with each other in order to trigger an alert attack when an attack is detected. The different interaction of agents according to the proposed negotiation protocol are shown in Figure 6.

5. Experimental Results

Several experiments were carried out to verify the flexibility, efficiency and performance of our MAS-DIDS approach. The proposed system was implemented using the Java language, the JADE 3.7 platform, the Aglets Workbench 2.0.2 platform, and the JPCAP Framework configured on a NetBeans IDE.

The Java Agent Development framework (JADE) is a library implemented in Java for the development and execution of intelligent agents[24]. JPCAP is a project that was developed to detect intruder activity in a network based on the existing signatures of intrusion attacks or abnormal behaviors[25]. Aglets (agent applets) are mobile Java agents that can move from one host to another[26]. An Aglet is therefore autonomous, since it can resume its execution on arrival at a destination, and reactive, since it can respond (react) to events in its environment [27]. All our experiments were performed on powerful machines equipped with an Intel Core i7 2.80 GHz processor and 16 GB of RAM. We first carried out a communication model test by sending a set of test messages between the different types of agents using ACL. Figure 7 shows the source code for a communication test scenario between the IA and AA. As a follow-up, we programmed the agent interface using the JPCAP framework to capture and collect all network packets from the environment and save them in a file entitled "SniffingFile.cap". To validate and test the performance of our model, we used the following examples of attacks.



```
// Sender Agent -- Interface Agent
package mas_dids;

import jade.core.*;
import jade.lang.acl.ACLMessage;

public class InterfaceAgent extends Agent {

   protected void setup() {
       System.out.println("communication test !! .  My name is "+this.getLocalName());
       sendMessage();
 }

 // Receiver Agent -- Analysis Agent


package mas_dids;
import jade.core.*;

public class AnalysisAgent extends Agent {

 protected void setup() {
     System.out.println("communication test. My name is "+this.getLocalName());
     addBehaviour(new ResponderBahaviour(this));
}
}
```

Figure 7. Communication test with ACL

- Denial of service attacks (DOS)
- User-to-root attacks (U2R)
- Remote-to-user attacks (R2L)

Twoimportant metrics were used to evaluate the performance of our MAS-DIDS proposal: detection rate (DR) and false positive rate (FPR). DR refers to the number of true attacks detected within these detections [28], defined by:

$$DR = \frac{TP}{TP + FN} \qquad (2)$$

FPR refers to the number of instances falsely detected as attacks within all these detections[29], defined by:

$$FPR = \frac{FP}{TN{+}FP} \qquad (3)$$

where FP represents false positives, TP true positives and FN false negative. We can say that an IDS model is effective and realistic if it achieves a high DR and a low FPR.

Figure 8 shows the detection performance of our MAS-DIDS model based on the experimental results given in Table 1, which shows the relationship between the DR and FPR in our simulated cloud environment. The simulation results for this architecture show that it has a DR higher than 80% and a false alarm rate lower than 11%. It also achieves the best performance for R2L attacks.

Table 1. Experimental results

Attack type	DR	FPR
Dos	72%	11%
U2R	63,11%	18%
R2L	81%	8,3%

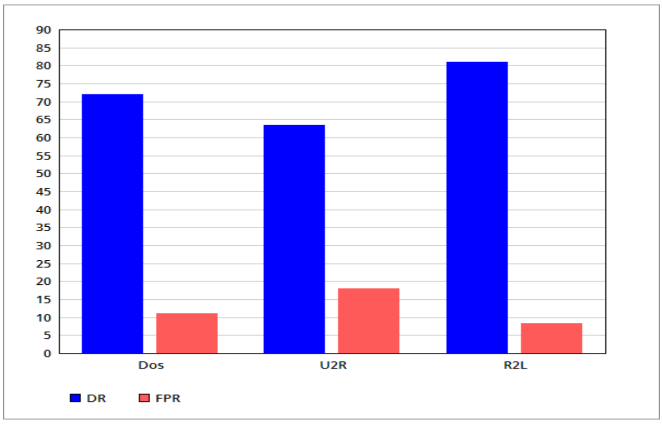


Figure 8. Performance of MAS-DIDS

These results indicate that our proposal is realistic, since the DR is increased and the FPR is decreased for tests involving simulated attacks. This means that MAS-DIDS has detected a significant number of attacks in a relatively short time. Finally, we repeated the same experiment using a classical IDS which was not based on the concept of agents, and compared the results with those of our MAS-DIDS model. The results obtained are presented in Figure 9. Our model worked better in terms of both efficiency and response time.

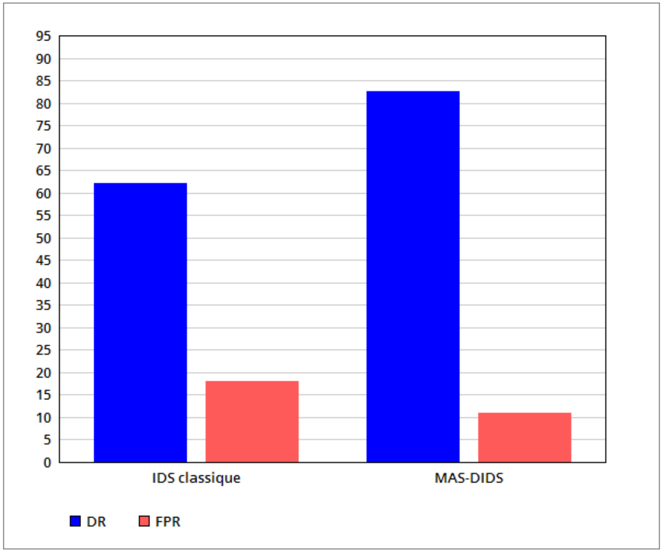


Figure 9. Performance comparison between classical IDS and MAS-DIDS

6. Conclusions

In this paper, we propose a new flexible, distributed and adaptive model, based on intelligent agent technology and DIDS, called MAS-DIDS. This model consists of a group of reactive, autonomous and cooperating agents that interact with each other to reduce the workload of an IDS, making the system more efficient and secure. The experimental results show that the MAS-DIDS can increase the intrusion detection rate and decrease the false positive rate compared to the use of a centralized IDS, meaning that our proposal is realistic and valuable for detecting both known and unknown attacks in a complex and dynamic environment such as cloud computing.

References

[1] M. Ramachandran and V. Chang, “Towards performance evaluation of cloud service providers for cloud data security,” *Int. J. Inf. Manage.*, vol. 36, no. 4, pp. 618–625, 2016.

[2] Y. A. Mohamed and A. B. Abdullah, “Implementation of IDS with response for securing MANETs,” in *Proceedings 2010 International Symposium on Information Technology - Engineering Technology, ITSim’10*, 2010.

[3] K. Venkataramana and M. Padmavathamma, “Multi-agent Intrusion Detection and Prevention System for Cloud Environment,” *Int. J. Comput. Appl.*, vol. 49, no. 20, pp. 24–29, Jul. 2012.

[4] Z. Chang and Y. L. Zhu, “The design of wireless intrusion detection system based on immune algorithm,” in *International Conference on Machine Learning and Cybernetics*, 2011.

[5] P. Singh Hada, R. Singh, and M. Manmohan Meghwal, “Security Agents: A Mobile Agent based Trust Model for Cloud Computing,” *Int. J. Comput. Appl.*, vol. 36, no. 12, pp. 975–8887, 2011.

[6] C. Saadi and H. Chaoui, “Cloud Computing Security Using IDS-AM-Clust, Honeyd, Honeywall and Honeycomb,” in *Procedia Computer Science*, 2016, vol. 85, pp. 433–442.

[7] A. V. Dastjerdi, K. A. Bakar, and S. G. H. Tabatabaei, “Distributed Intrusion Detection in Clouds Using Mobile Agents,” in *2009 Third International Conference on Advanced Engineering Computing and Applications in Sciences*, 2009.

[8] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks,” *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, 2005.

[9] H. Toumi, A. Talea, B. Marzak, A. Eddaoui, and M. Talea, “Cooperative trust framework for cloud computing based on mobile agents,” *Int. J. Commun. Networks Inf. Secur.*, 2015.

[10] H. Wang, H. Zhou, and C. Wang, “Virtual Machine-based Intrusion Detection System Framework in Cloud Computing Environment,” *J. Comput.*, vol. 7, no. 10, 2012.

[11] H. Idrissi, M. Ennahbaoui, E. M. Souidi, and S. El Hajji, “Mobile Agents with Cryptographic Traces for Intrusion Detection in the Cloud Computing,” in *Procedia Computer Science*, 2015, vol. 73, pp. 179–186.

[12] N. Afzali Seresht and R. Azmi, “MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach,” *Eng. Appl. Artif. Intell.*, vol. 35, pp. 286–298, 2014.

[13] P. Mell and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” *Natl. Inst. Stand. Technol. Inf. Technol. Lab.*, vol. 145, p. 7, 2011.

[14] O. Achbarou, M. A. El kiram, and S. El Bouanani, “Securing Cloud Computing from Different Attacks Using Intrusion Detection Systems,” *Int. J. Interact. Multimed. Artif. Intell.*, 2017.

[15] S. Singh, Y. S. Jeong, and J. H. Park, “A survey on cloud computing security: Issues, threats, and solutions,” *J. Netw. Comput. Appl.*, vol. 75, pp. 200–222, 2016.

[16] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” *Journal of Network and Computer Applications*, vol. 34, no. 1. pp. 1–11, 2011.

[17] A. Patcha and J. M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[18] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino Júnior, “An intrusion detection and prevention system in cloud computing: A systematic review,” *Journal of Network and Computer Applications*, vol. 36, no. 1. pp. 25–41, 2013.

[19] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in Cloud,” *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, 2013.

[20] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” *J. Netw. Comput. Appl.*, vol. 36, pp. 16–24, 2012.

[21] N. Keegan, S.-Y. Ji, A. Chaudhary, C. Concolato, B. Yu, and D. H. Jeong, “A survey of cloud-based network intrusion detection analysis,” *Human-centric Comput. Inf. Sci.*, vol. 6, no. 1, 2016.

[22] R. C. Cavalcante, I. I. Bittencourt, A. P. Da Silva, M. Silva, E. Costa, and R. Santos, “A survey of security in multi-agent systems,” *Expert Systems with Applications*, vol. 39, no. 5. pp. 4835–4846, 2012.

[23] Z. A. Baig, “Multi-agent systems for protecting critical infrastructures: A survey,” *Journal of Network and Computer Applications*, vol. 35, no. 3. pp. 1151–1161,

2012.

[24] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa, “JADE: A software framework for developing multi-agent applications. Lessons learned,” *Inf. Softw. Technol.*, vol. 50, no. 1–2, pp. 10–21, 2008.

[25] P. Shinde and T. J. Parvat, “DDoS Attack Analyzer: Using JPCAP and WinCap,” in *Procedia Computer Science*, 2016, vol. 79, pp. 781–784.

[26] C. J. Su, “Mobile multi-agent based, distributed information platform (MADIP) for wide-area e-health monitoring,” *Comput. Ind.*, vol. 59, no. 1, pp. 55–68, 2008.

[27] G. Fortino, A. Garro, and W. Russo, “Achieving Mobile Agent Systems interoperability through software layering,” *Inf. Softw. Technol.*, vol. 50, no. 4, pp. 322–341, 2008.

[28] G. P. Spathoulas and S. K. Katsikas, “Reducing false positives in intrusion detection systems,” *Comput. Secur.*, vol. 29, no. 1, pp. 35–44, 2010.

[29] L. Cazorla, C. Alcaraz, and J. Lopez, “A three-stage analysis of IDS for critical infrastructures,” *Comput. Secur.*, vol. 55, pp. 235–250, 2015.

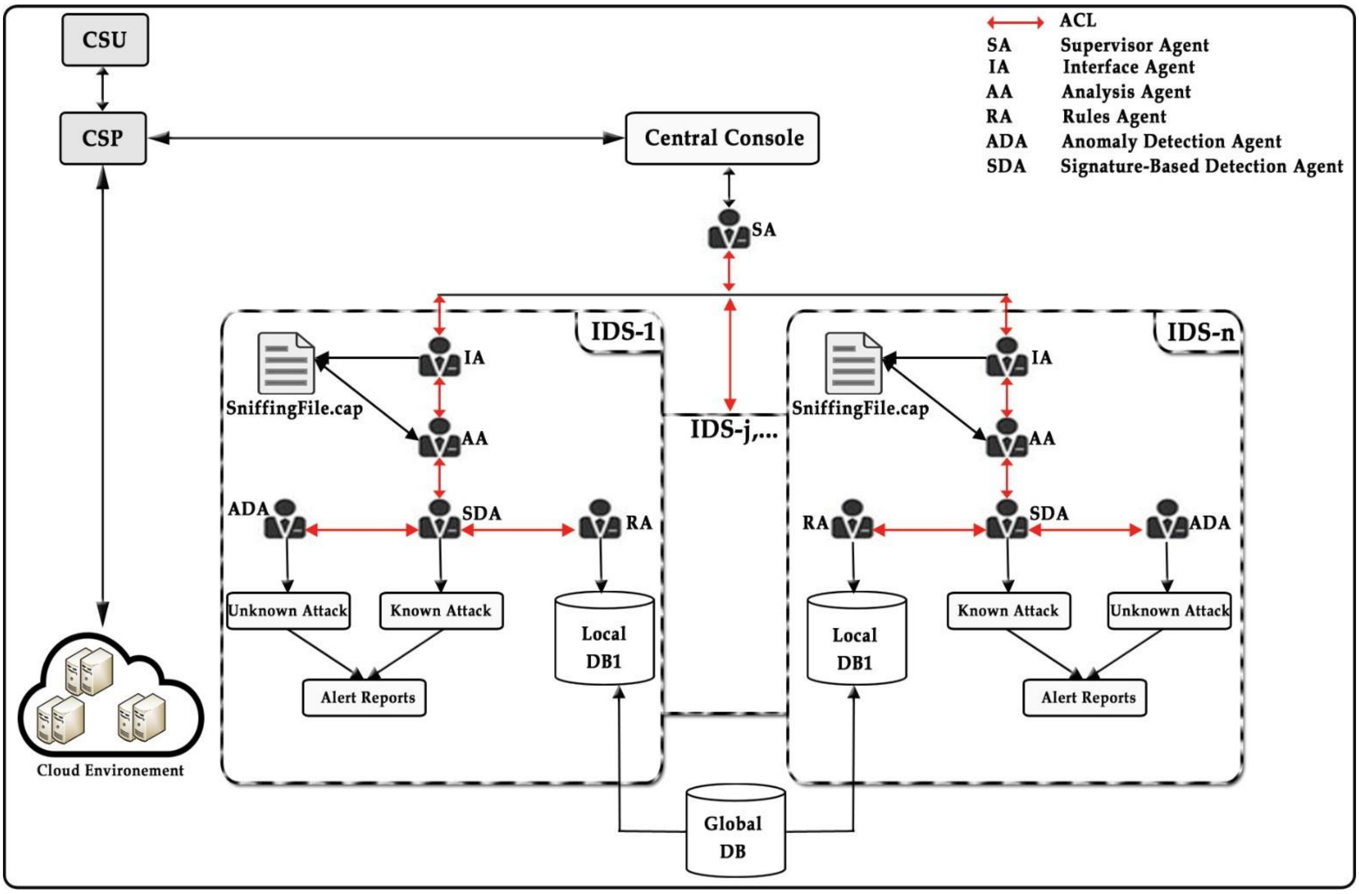


Figure 3.*Proposed MAS-DIDS Model*

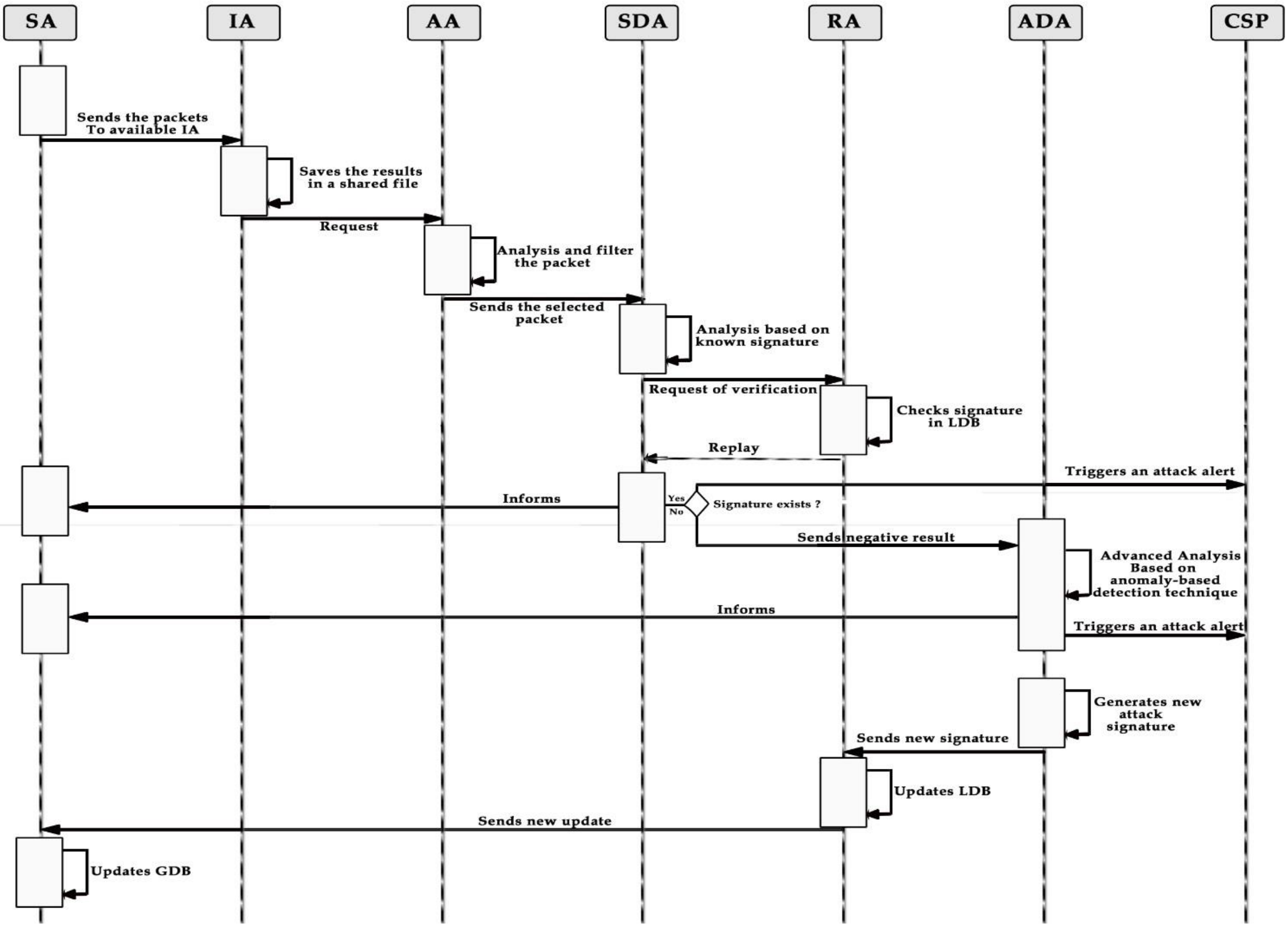

SA
IA
AA
SDA
RA
ADA
CSP
Sends the packets To available IA
Saves the results in a shared file
Request
Analysis and filter the packet
Sends the selected packet
Analysis based on known signature
Request of verification
Checks signature in LDB
Replay
Triggers an attack alert
Informs
Yes
No
Signature exists ?
Sends negative result
Advanced Analysis Based on anomaly-based detection technique
Informs
Triggers an attack alert
Generates new attack signature
Sends new signature
Updates LDB
Sends new update
Updates GDB

Figure 6.*Protocol of the negotiation model for MAS-DIDS system*