A Multi-agents Intrusion Detection System Using Ontology and Clustering Techniques

Imen Brahmi, Hanen Brahmi, Sadok Ben Yahia

A Multi-agents Intrusion Detection System Using Ontology and Clustering Techniques

Imen Brahmi[1], Hanen Brahmi[1], and Sadok Ben Yahia[2]

[1] Faculty of Sciences of Tunis. Computer Science Department.
Campus University, 1060 Tunis, Tunisia.
imen.brahmi@gmail.com
[2] Institut Mines-TELECOM, TELECOM SudParis,
UMR CNRS Samovar, 91011 Evry Cedex, France.
sadok.benyahia@fst.rnu.tn

Abstract. Nowadays, the increase in technology has brought more sophisticated intrusions. Consequently, Intrusion Detection Systems (IDS) are quickly becoming a popular requirement in building a network security infrastructure. Most existing IDS are generally centralized and suffer from a number of drawbacks, *e.g.*, high rates of false positives, low efficiency, etc, especially when they face distributed attacks. This paper introduces a novel hybrid multi-agents IDS based on the intelligent combination of a clustering technique and an ontology model, called OCMAS-IDS. The latter integrates the desirable features provided by the multi-agents methodology with the benefits of semantic relations as well as the high accuracy of the data mining technique. Carried out experiments showed the efficiency of our distributed IDS, that sharply outperforms other systems over real traffic and a set of simulated attacks.

Keywords: Intrusion Detection System; Multi-agents; Clustering; Ontology.

1 Introduction

As far the cost of information processing and Internet accessibility is dropping, more and more organizations are becoming vulnerable to a wide variety of cyber threats. Therefore, network security is becoming a major challenge. Consequently, software tools, that can automatically detect a variety of intrusions, are of a compelling need. An *Intrusion Detection Systems* (IDS) has been of use to detect and defend intrusions more proactively in short period.

Even that IDSs have become a standard component in security infrastructures, they still have a number of signicant drawbacks [14]. Indeed, they suffer from problems of reliability, relevance, disparity and/or incompleteness in the presentation and manipulation of knowledge as well as the complexity of attacks. This fact hampers the detection ability of IDS, since it causes the generation excessive of false alarms and decreases the detection of real intrusions. In addition, most of the IDSs use centralized architectures. Unfortunately this strategy has

several drawbacks [4]. Indeed, the central processing node can lead to a single point of failure. Clearly, whenever the central processing node is attacked, then the whole IDS has been damaged. Besides, the transfer of all the information at a central processing unit implies a great need on network resources and leads to much network load on the system. Consequently, the centralized IDS suffers from scalability problems [4]. Moreover, the communication and cooperation between a centralized IDS components are badly missing. To palliate these problems, the integration of a multi-agents technology within the IDS seemed to be an appropriate solution. In fact, the use of multi-agents system for intrusion detection offers a new alternative to the IDS with several advantages listed in literature, *e.g.*, independently and continuous running, minimal overhead, scalability, *etc.*, [4]. Therefore, multi-agent technology makes the resilience of the system strong and thus ensures its safety [6]

Alongside, the concept of ontology has emerged as a powerful method for domain knowledge representation and sharing. It can improve the intrusion detection features giving the ability to share a common conceptual understanding threats and design the signature rules [1, 7, 10, 13, 20]. In fact, the use of the ontologies and OWL (*Ontology Web Language*) within the intrusion detection context has different advantages: (*i*) Grasping the semantic knowledge about the intrusion detection subject; (*ii*) Expressing the IDS much more by building better rules of signatures using the SWRL (*Semantic Web Rule Language*) [9]; and (*iii*) Making intelligent reasoning [6, 10]. In this respect, it is possible to design a multi-agents architecture based on a knowledge basis represented as an ontology. The use of such architecture reveals conducive to the development of IDSs [6].

In this paper, we investigate another way of tackling the aforementioned problems. Thus, we introduce a new distributed IDS, called OCMAS-IDS (*Ontology and Clustering based Multi-AgentS Intrusion Detection System*). OCMAS-IDS is based on the integration of the multi-agents technology, the ontology and the clustering technique. In this respect, our proposed system uses a set of agents that can be applied to a number of tasks, namely: data capturing, detecting the known and unknown attack categories and ultimately alerting the administrator. Through extensive carried out experiments on a real-life network traffic and a set of simulated attacks, we show the effectiveness of our proposal in terms of (*i*) the scalability and (*ii*) the detection ability of our system.

The remaining of the paper is organized as follows. Section 2 sheds light on the related work. We introduce our new distributed intrusion detection system based on the multi-agents technology in Section 3. We then relate the encouraging results of the carried out experiments in Section 4. Finally, Section 5 concludes and points out avenues of future work.

2 Scrutiny of the related work

Recently, few approaches, within the intrusion detection field, are dedicated to the integration of multi-agents technology and ontology model. Approaches

fitting in the distributed IDS trend using ontological structure attempt to enhance the IDS accuracy and performing intelligent reasoning.

Worth of mention that the first research of applying ontology within intrusion detection context was done by Undercoffer et *al.* [20] in 2003. In this respect, the authors developed an ontology focused on the target (*centric*) and supply it within the format of the logical description language DARPA *DARPA Agent Markup Language + Ontology Inference Layer* (DAML + OIL). This ontology allows modeling the domain of computer attacks and facilitates the process of reasoning to detect and overcomes the malicious intrusions.

Mandujano [13] proposed a detection tool composed of a multi-agents architecture and an ontology focused on attacker, called FROID (*First Resource for Outbound Intrusion Detection*). FROID attempts to protect a set of nodes in a network using the ontology OID (*Outbound Intrusion Detection*). The proposed system is characterized by its intention to detect known attacks based on signatures. Thus, the main drawback of FORID system is that in case of an emerging attack, it will ignore it since this new attack has not yet been listed in the base of signatures.

In addition, Abdoli and Kahani [1] proposed a system, called ODIDS. The system includes two types of agents: IDSAgent and MasterAgent. Based on the techniques of the semantic web, they have built an ontology for extracting semantic relationships between intrusions. The main moan that can be addressed to the ODIDS system stands in the fact that the MasterAgent is a central point of failure. Hence, if an intruder can prevent it from working (*e.g.*, blocking or slowing the host where it is running), the entire system will be damaged. Another criticism of the ODIDS system is time wasting, since the system needs more time to make a connection between the MasterAgent and the IDSAgents on the network and to send and receive messages between them.

Azevedoln et *al.* [3] proposed an autonomic model, called AutoCore, which includes a set of intelligent agents as well as a domain ontology CoreSec, in order to perform intrusion detection independently. The system makes use of CoreSec as an ontology knowledge base with high-level concepts for information [3]. The agents are then responsible for enabling the analysis of network traffic and the detection of malicious activities. However, the approach does not consider the secure state which is important to judge false positive alerts and successful possibility of attacks [12].

In [7], Djotio et *al.* proposed a MONI system based on an ontology model, called NIM-COM. The MONI system includes a multi-agents IDS to achieve a distribution of the detection activities. In addition, MONI is endowed with a *Case Based Reasoning* (CBR) mechanism to learn new attacks. Even though CBR is considered as a powerful reasoning paradigm and easy to set up, it suffers from re-engineering problems [14]. This lack of flexibility of the knowledge representation is with no doubt an inherent CBR limitation.

With the same preoccupation, Isaza et *al.* [10] developed a multi-agents architecture for the detection and prevention of intrusions, called OntoIDPSMA. The representation of known attacks has been designed using a semantic model

based on ontology specifying signatures and reaction rules. The authors integrated an Artificial Neural Network (ANN) technique and the clustering algorithm K-means for the identification of new attacks. However, the most significant disadvantage of ANN relies on the fact that its ability to identify an intrusion is completely dependent on the accurate training of the system, data and the methods that are used. Moreover, the configuration of an ANN is delicate and can significantly affect the results [18]. In addition, the performance of K-means and its effectiveness as a method for detecting new attacks depends on the random selection of the number of initial groups. Therefore, a "bad choice" of this number will decrease the detection of actual intrusions and increase the generation of false alarms [4].

Due to its usability and importance, detecting the distributed intrusions still be a thriving and a compelling issue. In this respect, the main thrust of this paper is to propose a hybrid distributed IDS, called OCMAS-IDS, which integrates : (*i*) a multi-agents technology; (*ii*) an ontology; and (*iii*) an unsupervised clustering technique. The main idea behind our approach is to address limitations of centralized IDSs by taking advantage of the multi-agents paradigm as well as the ontological representation.

3 The OCMAS-IDS system

Agents and multi-agents systems are one of the paradigms that best fit the intrusion detection in distributed networks [4]. In fact, the multi-agents technology distributes the resources and tasks and hence each agent has its own independent functionality, so it makes the system perform work faster [6].

The distributed structure of OCMAS-IDS is composed of different cooperative, communicant and collaborative agents for collecting and analyzing massive amounts of network traffic, called respectively: SnifferAgent, MisuseAgent, AnomalyAgent and ReporterAgent. Figure 1 sketches at a glance the overall architecture of OCMAS-IDS.

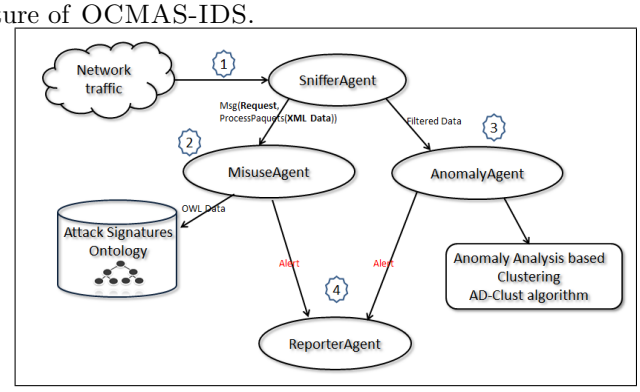

Fig. 1. The architecture of OCMAS-IDS at a glance.

Worth of mention that the combination of the detection known attacks as well as the unknown ones can lead to improve the performance of the IDS and

enhances its detection ability [11]. Consequently, OCMAS-IDS efficiently merges the detection of both types of attacks. It incorporates a MisuseAgent specialized on known attacks detection, as well as an AnomalyAgent competent on unknown attacks detection. The processing steps of OCMAS-IDS can be summarized as follows:

1. The SnifferAgent captures packets from the network. Indeed, a distributed IDS must undertake to analyze a huge volumes of events collected from different sources around the network. Consequently, the SnifferAgent permits to filter the packets already captured. Besides, it converts them to XML, using the XStream library[3]. Finally, the pre-processed packets will be sent to others agents to be analysed;
2. The MisuseAgent receives the packets converted to XML from the SnifferAgent. It transforms these packets to OWL format in order to be compatible with the SWRL rules stored in the ontology. Now, it is ready to analyze the OWL packets to detect those that correspond to known attacks. Indeed, the MisuseAgent searches for attack signatures[4] in these packets, by consulting the ontology ASO (*Attack Signatures Ontology*). Consequently, if there is a similarity between the OWL packets and the SWRL rules that define the attack's signatures, then the agent raises an alert to the ReporterAgent;
3. The filtered network packets are fed into an AnomalyAgent, which uses the clustering algorithm $\mathcal{AD}$-$\mathcal{C}$lust to detect the unknown attacks. Likewise, the agent sends an alert to the ReporterAgent, if an attack is identified;
4. Finally, the ReporterAgent generates reports and logs.

OCMAS-IDS detects the known attacks through the intelligent agent MisuseAgent, which uses an ontology to enrich data intrusions and attack signatures by semantic relationships. In what follows, we present the proposed ontology used within our system OCMAS-IDS.

3.1 The Attack Signatures Ontology (ASO)

Since last few decades, Raskin et *al.* [16] opened a new field, that focuses on using *Ontology* within information security and its advantages. In fact, ontologies present an extremely promising new paradigm in computer security domain. They can be used as basic components to perform automatic and continuous analysis based on *high-level* policy defined to detect threats and attacks [10]. Moreover, they enable the IDS with improved capacity to reason over and analyze instances of data representing an intrusion [7, 20]. Furthermore, the interoperability property of the ontologies is essential to adapt to the problems of the systems distribution, since the cooperation between various information systems is supported [3, 7].

[3] Available at: http://xstream.codehaus.org/.

[4] An attack signature is a known attack method that exploits the system vulnerabilities and causes security problem [4].

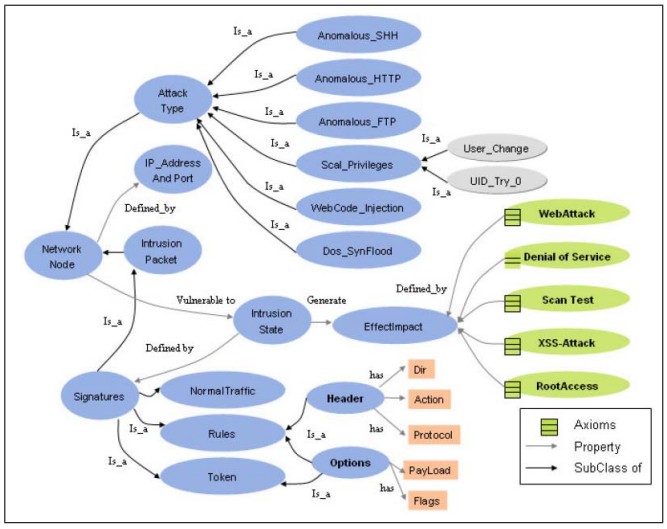

Fig. 2. The Attack Signatures based Ontology ASO.

Within the OCMAS-IDS system, an ontology, called ASO (*Attack Signatures based Ontology*), is implemented, in order to optimize the knowledge representation and to incorporate more intelligence in the information analysis. Moreover, OCMAS-IDS integrates into its internal structure the interoperability between agents since they use the same model of ontology. The ASO ontology is characterized by network components, intrusion elements, classification defining traffic signatures and rules classes and instances. Figure 2 depicts a fragment of the ontology ASO, which implements the intrusion detection knowledge. The ASO ontology allows the representation of the signatures basis for known attacks, used with the agent MisuseAgent. The power and usefulness of ontology, applied to the signature basis issue, provide a simple representation of the attacks expressed by the semantic relationships between intrusion data. We can also infer additional knowledge about intrusion due to the ability of the ontology to infer new behavior by reasoning about data. Therefore, this fact improves the process of decision support for an IDS [1, 6, 20].

The signature basis incorporates rules provided by the ASO ontology, that allows a semantic mean for reasoning and inferences. In fact, the rules are extracted using the SWRL language (*Semantic Web Rule Language*). The latter extend the ontology and enriches its semantics by the deductive reasoning capabilities [9]. It allows to handle instances with variables (?x, ?y, ?z). Thus, the SWRL rules are developed according to the scheme: *Antecedent* $\rightarrow$ *Consequent*, where both antecedent and consequent are conjunctions of atoms written $a_1 \wedge ... \wedge a_n$. Variables are indicated using the standard convention of prefixing them with a question mark (*e.i.*, "?x"). The following example shows a rule represented with SWRL.

Example 1. NetworkHost(?z) $\wedge$ IntrusionState(?p) $\wedge$ GeneratedBY(?p,?z) $\wedge$ SQLInjection(?p) $\wedge$ Directd_To(?p,?z) $\rightarrow$ SystemSQLInjectionState(?p,?z)

Using this syntax, a rule asserting that the composition of the network host(z) and an intrusion state(p) properties implies the attack "*SQL Injection*" property.

When constructing our ontology, we designed and implemented multiple rules to define various attacks and signatures. The defined rules allow properties inferences and reasoning process. The attack properties, *e.g.*, *WebAttack*, *SQLInjection*, *DoS*, *dDoS*, and so on, are defined as ontology' attributes identifying the type of an intrusion.

Even thought, the known attacks are detected, it remains nevertheless the problem of the new attacks detection. In this respect, additionally to the MisuseAgent, based on the ontology, OCMAS-IDS uses an AnomalyAgent based on the clustering analysis. The algorithm is described in the following subsection.

3.2 The clustering algorithm AD-Clust

Needless to remind that the application of the data mining techniques within the intrusion detection context can effectively improve the detection accuracy, the detection speed, and enhance the system's own security [2]. Thus, as an intelligent analysis task, the AnomalyAgent provides the crossroads of multi-agents systems with the clustering technique, in particular the $\mathcal{AD}$-$\mathcal{C}$lust algorithm. The idea behind this technique is that the amount of normal connection data is usually overwhelmingly larger than that of intrusions [5]. Whenever this assumption holds, the anomalies and attacks can be detected based on cluster sizes, *i.e*, large clusters correspond to normal data, and the rest of the data points, which are outliers, correspond to attacks [19].

$\mathcal{AD}$-$\mathcal{C}$lust, (*Anomaly Detection-based Clustering*), is an unsupervised clustering algorithm introduced by Brahmi et *al.* in [4, 5], to improve the quality of the K-means algorithm applied within the intrusion detection context. Indeed, the latter suffers from a greater time complexity, which becomes an extremely important factor within intrusion detection due to the very large packets sizes [15]. Moreover, the *number of clusters dependency* and the *degeneracy* constitute the drawbacks that hamper the use of K-means for anomaly detection [15]. In this respect, the $\mathcal{AD}$-$\mathcal{C}$lust algorithm combines two prominent categories of clustering, namely: distance-based [19] as well as density-based [8]. It exploits the advantages of the one to palliate the limitations of the other and vice versa.

The processing steps of our algorithm $\mathcal{AD}$-$\mathcal{C}$lust can be summarized as follows [4]:

1. Extraction of the density-based clusters that are considered as candidate initial cluster centers. The density-based clustering is used as a preprocessing step for the $\mathcal{AD}$-$\mathcal{C}$lust algorithm;
2. Compute the Euclidean distance between the candidate cluster center and the instance that will be assigned to the closest cluster. For an instance x_i and a cluster center z_i, the Euclidean distance is defined as:

$$distance(x_i, z_i) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2} \tag{1}$$

3. The size of a neighborhood of instances is specified by an input parameter. We use the k' parameter to distinguish it from the k parameter used by the K-means algorithm. Hence, k' specifies the minimal number of instances in a neighborhood and controls the granularity of the final clusters of the clustering-based density. If k' is set to a large value, then a few large clusters are found. To reduce the number of candidate clusters k' to the expected number k, we can iteratively merge the two most similar clusters. Otherwise, if k' is set too small, then many small clusters will be generated. The clusters will be split, new clusters will be created to replace the empty ones and the instances will be re-assigned to existing centers. This iteration will continue until there is no empty cluster. Consequently, the outliers of clusters will be removed to form new clusters, in which instances are more similar to each other. In this way, the value of initial cluster centers k will be determined automatically by splitting or merging clusters;
4. Within the detecting phase, the $\mathcal{AD}$-$\mathcal{C}$lust algorithm performs the detection of intrusions. Thus, for each novel instance I the algorithm proceeds as follows:
   (a) Compute the Euclidean distance and find the cluster that presents the shortest distance with respect to I.
   (b) Classify I by the category of the closest cluster. Clearly, if the distance between I and the cluster of "normal" instances is the shortest one, then I will be a normal instance. Otherwise, I is an intrusion.

4 Experimental results

In order to assess the overall performance of OCMAS-IDS in a realistic scenario, a prototype of the proposed architecture was implemented using Sun's Java Development Kit 1.4.1, the well known platform JADE[5] 3.7, the Eclipse and the JPCAP[6] 0.7. The ontology ASO is designed using Protégé[7].

Through the carried out experiments, we have to stress on evaluating the performance of our system in terms of (*i*) the scalability-related criteria such as network bandwidth, detection delay and system response time; and (*ii*) the detection ability. During the evaluations, we compare the results of the OCMAS-IDS system *vs.* that of the centralized IDS Snort [17] and the multi-agents based ontology one MONI[8] [7]. All experiments were carried out on equivalent machines equipped with a 3GHz Pentium IV and 8GB of main memory. We used machines that were connected via a switch, thus forming a switched network. Moreover, we simulated attacks using the well known tool *Metasploit*[9] version 3.5.1. The simulated eight different attack types are:

[5] Available at: http://jade.tilab.com
[6] Available at: http://netresearch.ics.uci.edu/kfujii/jpcap/doc/
[7] Available at: http://protege.stanford.edu/download/download.html
[8] We thank Mrs. Djotio et *al.* [7] for providing us with the implementation of MONI system.
[9] Available at: http://www.metasploit.com/

- **attack1**: DoS Smurf;
- **attack2**: Backdoor Back Office;
- **attack3**: SPYWARE-PUT Hijacker;
- **attack4**: Nmap TCP Scan;
- **attack5**: Finger User;
- **attack6**: RPC Linux Statd Overflow;
- **attack7**: DNS Zone Transfer; and
- **attack8**: HTTP IIS Unicode.

4.1 The scalability evaluation

In order to test the scalability of OCMAS-IDS, we study the relationship between the bandwidth consumption and a number of attack types. Moreover, the variation of the detection delay according to the number of packets is evaluated. Additionally, we assess how the response time varies with respect to eight attack types.

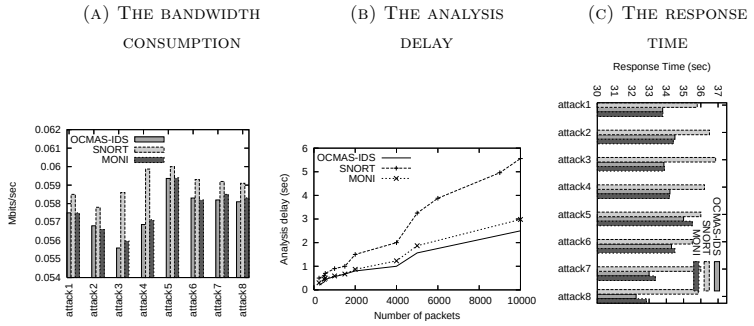

Fig. 3. The bandwidth consumption, the analysis delay and the response time of OCMAS-IDS *vs.* Snort and MONI.

As depicted in Figure 3 (a), the maximum bandwidth consumed by OCMAS-IDS and MONI is lower compared to that of Snort. For example, the maximum bandwidth consumed by OCMAS-IDS is 0.06 Mbits/sec, which is very low as well. The reduction of the network bandwidth consumption is owe to the use of the multi-agents system. Thus, the OCMAS-IDS system is not greedy in bandwidth consumption, which is definitely a desirable feature for any distributed system [4].

Besides, Figure 3(b) plots the detection delay against the number of packets, using the OCMAS-IDS, MONI and Snort systems. According to this figure, we can answer the question: why the realization of the multi-agents IDS is advantageous? Clearly, the results show that the detection delay of both systems linearly increases with the number of packets. Moreover, the gap between both curves related to the detection delay of OCMAS-IDS and MONI is small, since both systems are based on multi-agents technology. In addition, Figure 3(b) highlights that our proposed system OCMAS-IDS is faster than the system Snort. This can be explained by the fact that agents operate directly on the host whenever an action has to be taken, their response is faster than systems where actions were taken by the central controller, *i.e.*, Snort.

Figure 3 (c) illustrates the response time required by OCMAS-IDS with respect to the attack types. On the one hand, we remark that the detection of all attack types, on average, result in lower response time compared to that of Snort, due to its centralized detection engine. In addition, this figure proved how fast our system respond. For example, the response time of OCMAS-IDS was 35 seconds for attack5, which is absolutely negligible.

On the other hand, within MONI, the ontology model is developed under JADE. Differently, the ontology ASO of our system OCMAS-IDS is designed under Protégé and queried with SWRL. The response time of OCMAS-IDS is better than that of MONI. The main reason is that in the case of OCMAS-IDS, the inferred model is computed only once before the matching starts and used throughout all the queries. Thus, the figure indicates that OCMAS-IDS outperforms MONI and permits the exploitation of the semantics of ASO.

To sum up, it is clear from the obtained results that the performance of the OCMAS-IDS will not deteriorate too much with the increase in the number of attacks, which is justified by its low bandwidth consumption, reduced detection delay and quick response time. Likewise, in case of more machines are connected to the network, the OCMAS-IDS system still withstand the load and swiftly deliver the results.

4.2 The detection ability

In order to evaluate the detection ability of an IDS, two interesting metrics are usually of use [4]: the *Detection Rate* (DR) and the *False Positive Rate* (FPR). Indeed, the DR is the number of correctly detected intrusions. On the contrary, the FPR is the total number of normal instances that were "incorrectly" considered as attacks. In this respect, the value of the DR is expected to be as large as possible, while the value of the FPR is expected to be as small as possible.

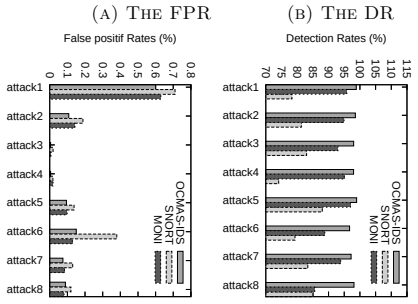

Fig. 4. The FPR and the DR of OCMAS-IDS *vs.* Snort and MONI.

With respect to Figure 4 (a), we can remark that the FPR of OCMAS-IDS and MONI is significantly lower compared to that of Snort. This fact is due to the adaptive mechanisms used by the agents, enabling both systems, *i.e.*, OCMAS-IDS and MONI, to better suit the environment. Consequently, the

false alarms can be reduced correspondingly. For example, for attack3 the FPR of SNORT can reach values as high as 0.019% compared to 0.007% of MONI and 0.005% of OCMAS-IDS.

Moreover, Figure 4 (b) shows that the DR of OCMAS-IDS is higher than that of MONI. Moreover, among the three investigated IDS, SNORT has the lowest DR. For instance, for attack3, whenever OCMAS-IDS and MONI have the DR 97.9% and 94.9%, respectively, SNORT has 74.1% DR. This is due to his centralized architecture.

Knowing that a main challenge of existing IDSs is to decrease the false alarm rates [4], the main benefit of our system is to lower the false alarm rate, while maintaining a good detection rate.

5 Conclusion

In this paper, we focused on a distributed architecture and multi-agents analysis of intrusions detection system to tackle the mentioned above challenges, *i.e.*, the high detection delay, the high bandwidth consumption as well as the low detection ability. Thus, we introduced a multi-agents intrusions detection system called $OCMAS\text{-}IDS$ based on an efficient ontology model, called ASO, as well as a clustering algorithm called $\mathcal{AD}$-$\mathcal{C}$LUST. The carried out experimental results showed the effectiveness of the OCMAS-IDS system and highlighted that our system outperforms the pioneering systems fitting in the same trend.

Future issues for the present work mainly concern: (i) the alert correlation techniques by using the multi-agents system and ontology [12].

References

1. F. Abdoli and M. Kahani. Ontology-based Distributed Intrusion Detection System. In *Proceedings of the 14th International CSI Computer Conference CSICC 2009, Tehran, Iran*, pages 65–70, 2009.
2. C. Azad and V. K. Jha. Data Mining in Intrusion Detection: A Comparative Study of Methods, Types and Data Sets. *International Journal of Information Technology and Computer Science(IJITCS)*, 5(8):75–90, 2013.
3. R.R. Azevedoln, E. R. G. Dantas, R. C. Santos, C. Rodrigues, M.J.S.C. Almeida, F. Freitas, and W.C. Veras. An Autonomic Ontology-Based Multiagent System for Intrusion Detection in Computing Environments. *The International Journal for Infonomics*, 3(1):1–7, 2010.
4. I. Brahmi, S. Ben Yahia, H. Aouadi, and P. Poncelet. Towards a Multiagent-Based Distributed Intrusion Detection System Using Data Mining Approaches. In *Agents and Data Mining Interaction, ADMI, Revised Selected Papers*, pages 173–194. Springer, 2012.
5. I. Brahmi, S. Ben Yahia, and P. Poncelet. $\mathcal{AD}$-$\mathcal{C}$LUST: Dtection des anomalies base sur le Clustering. In *Atelier Clustering Incrémental et Méthodes de Détection de Nouveauté en conjonction avec 11ème Conférence Francophone d'Extraction et de Gestion de Connaissances EGC 2011, Brest, France*, pages 27–41, 2011.

6. K. Brahmkstri, D. Thomas, S. T. Sawant, A. Jadhav, and D. D. Kshirsagar. Ontology Based Multi-Agent Intrusion Detection System for Web Service Attacks Using Self Learning. In N. Meghanathan, D. Nagamalai, and S. Rajasekaran, editors, *Networks and Communications (NetCom2013)*, pages 265–274. Springer, 2014.
7. T. N. Djotio, C. Tangha, F. N. Tchangoue, and B. Batchakui. MONI: Mobile Agents Ontology based for Network Intrusions Management. *International Journal of Advanced Media and Communication*, 2(3):288–307, 2008.
8. L. Duan. Density-Based Clustering and Anomaly Detection. In M. Mircea, editor, *Business Intelligence - Solution for Business Development*, pages 79–96, 2012.
9. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. Available at: `http://www.w3.org/Submission/SWRL/`.
10. G. A. Isaza, A. G. Castillo, M. López, and L. F. Castillo. Towards Ontology-Based Intelligent Model for Intrusion Detection and Prevention. *Journal of Information Assurance and Security*, 5:376–383, 2010.
11. G. Kim, S. Lee, and S. Kim. A Novel Hybrid Intrusion Detection Method Integrating Anomaly Detection With Misuse Detection. *Expert Systems with Applications*, 41(4, Part 2):1690 – 1700, 2014.
12. W. Li and S. Tian. An Ontology-Based Intrusion Alerts Correlation System. *Expert Systems with Applications*, 37(2010):7138–7146, 2010.
13. S. Mandujano, A. Galvan, and J.A. Nolazco. An Ontology-Based Multiagent Approach to Outbound Intrusion Detection. In *Proceedings of the International Conference on Computer Systems and Applications, AICCSA'05, Cairo, Egypt*, pages 94–I, 2005.
14. C. I. PinzóN, J. F. De Paz, Á. Herrero, E. Corchado, J. Bajo, and J. M. Corchado. idMAS-SQL: Intrusion Detection Based on MAS to Detect and Block SQL Injection Through Data Mining. *Information Sciences*, 231:15–31, 2013.
15. R. Ranjan and G. Sahoo. A New Clustering Approach For Anomaly Intrusion Detection. *International Journal of Data Mining and Knowledge Management Process (IJDKP)*, 4(2):29–38, 2014.
16. V. Raskin, C.F. Hempelmann, K.E. Triezenberg, and S. Nirenburg. Ontology in Information Security: A Useful Theoretical Foundation and Methodological Tool. In *Proceedings of the 2001 workshop on New security paradigms, NSPW'01, Cloudcroft, New Mexico*, pages 53–59, 2001.
17. M. Roesch. Snort - Lightweight Intrusion Detection System for Networks. In *Proceedings of of the 13th USENIX Conference on System Administration (LISA'99), Seattle, Washington*, pages 229–238, 1999.
18. A. Sodiya, O. Ojesanmi, O.C. Akinola, and O. Aborisade. Neural Network based Intrusion Detection Systems. *International Journal of Computer Applications*, 106(18):19–24, 2014.
19. I. Syarif, A. Prugel-Bennett, and G. Wills. Unsupervised Clustering Approach for Network Anomaly Detection. In *Proceedings of the 4th International Conference on Networked Digital Technologies (NDT 2012), Dubai, AE*, pages 135–145, 2012.
20. J. Undercoffer, A. Joshi, and J. Pinkston. Modeling Computer Attacks: An Ontology for Intrusion Detection. In *Proceedings of the 6th International Workshop on the Recent Advances in Intrusion Detection, Pittsburgh, PA, USA*, pages 113–135, 2003.