MAD-IDS: Novel Intrusion Detection System using Mobile Agents and Data Mining Approaches

Imen Brahmi[1], Sadok Ben Yahia[1], and Pascal Poncelet[2]

[1] Faculty of Sciences of Tunis, Tunisia
sadok.benyahia@fst.rnu.tn
[2] LIRMM Montpellier, France
Pascal.Poncelet@lirmm.fr

Abstract. Intrusion Detection has been investigated for many years and the field reached the maturity. Nevertheless, there are still important challenges, *e.g.*, how an Intrusion Detection System (IDS) can detect new and complex distributed attacks. To tackle this problem, we propose a novel distributed intrusion detection system, based on the desirable features provided by the mobile agent methodology. Our approach rely on: (*i*) a misuse detection mobile agent to detect known attacks, (*ii*) an anomaly detection mobile agent to detect novel kinds of attacks. Based on data mining techniques, this agent provides a high accuracy for predicting different behaviors in network computers. Carried out experiments showed the efficiency of data mining approaches integrated with mobile agent technology.

Keywords: Intrusion Detection System, Mobile Agents, Misuse Detection, Anomaly Detection, Data Mining Techniques.

1 Introduction

With the rapid growth of Internet, the security-relevant incidents have being increased. In addition, cracking technology has evolved into complex approach such as coordinated attack and cooperative attack. Under these circumstances, there is a great need for software tools that can automatically detect a variety of intrusions. As an important gatekeeper of network, *Intrusion Detection Systems* (IDS) must have the ability to detect and defend intrusions more proactively in shorter period.

Basically, two intrusion detection strategies can be distinguished: *anomaly detection* and *misuse detection* [3]. Anomaly detection systems monitor the system and try to decide whether its behavior is normal or not. This is achieved by keeping a normal user profiles. To detect abnormal activity, the predefined profiles are compared with the actual ones in use. The deviation will activate an alarm. In fact, the anomaly detection techniques can be effective against unknown or novel attacks since no prior knowledge about specific intrusion is required. However, they tend to generate more false alarms because an anomaly can just be a new behavior. Otherwise, misuse detection systems search for known *attack signatures*. A signature is a trail of a known attack. For example, it may be a specific series of bits in the header of an IP packet. A weakness of these systems is that they are not effective against novel attacks that have no matched signatures. In addition, once a new attack is discovered and its signature developed, often there is a substantial latency in its deployment.

As accuracy is the essential requirement for an IDS, its extensibility and adaptability are also critical in today's network computing environment. However, current IDS have some shortcomings as follows [11, 19]:

- Most IDS detect attacks by analyzing information from a single host, or a single network interface, at many locations throughout the network. Consequently, IDS components miss the communication and the cooperation between each other. This fact hampers the capability to detect large-scale distributed attacks;
- Most commercial IDS are built in hierarchical architecture, which is a tree structure with a control system at the top, information aggregation units at the internal nodes, and sensor units at the leaf nodes. In this kind of system, large amount of data transferred across the network may result in a network congestion;
- Because of the reliance on hierarchical structures, many IDS are susceptible to be attacked. An attacker can cut off a control branch of the IDS by attacking an internal node or even decapitate the entire IDS. Typically, such critical components have been hardened to resist direct attacks. Nevertheless, other survivability techniques such as redundancy, mobility, dynamic recovery etc, are showed to be missing in current IDS;
- Many IDS cannot adequately combine history intrusive alarms to analyze future intrusive behaviors. "*Knocking attack*" is an illustrative example. It means that many IDS have no ability to dynamically adjust detective policy by the former intrusive results.

In this paper, we investigate another way of tackling the aforementioned problems. Thus, we introduce a new distributed IDS based on the mobile agent technology using the data mining algorithms, which accurately capture the actual behavior of network traffic. In this respect, an agent is a program that can exercise an individual or organization's authority, work autonomously toward a goal, and interact with other agents [11].

Particularly, the mobile agent is an agent having the capability of moving from one host to another. The advantages of mobile agent technology includes: reducing network overload, overcoming network latency, synchronous and autonomous execution, robustness and fault-tolerance, system scalability, and operating in heterogeneous environments [9]. To this end, mobile agent technology has been shown to be very suitable to solve intrusion detection in a distributed environment [6].

Our proposed system employs this technology to coordinately process information from each monitored host. Indeed, a misuse detection mobile agent permits the detection of the known attacks. Beside, using the integration of both mobile agent methodology and data mining techniques, an anomaly detection agent provides the advantage of detecting new attacks. The resultant model is more effective since it is capable to detect both known and novel kinds of attacks in a distributed environment. Therefore, the integration of mobile agent technology and data mining techniques makes an IDS more autonomous and efficient.

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

2 Related work

IDSs have undergone rapid development in both power and scope in the last few years. IDS, in addition to protecting the system, needed to be able to resist attacks on themselves and also needed to be fault tolerant, highly adaptable and configurable. According to these characteristics, the agent-based technology seemed to be an appropriate alternative for developing IDS. Recently, several new agent-based IDS were developed [13, 16], using mobile agents [5, 6, 9, 18], using a hierarchy of static agents [8, 19] or employing a combination of both static and mobile agents [7].

Implementation of intrusion detection systems with mobile agent technology is one of the new paradigms for intrusion detection. Consequently, the application of mobile agent technology integrated with the data mining techniques to IDS gives a result to only few research projects.

One of the studies worth of mention was JAM (*Java Agents for Metalerning*) [20]. This work combines intelligent agents and data mining techniques. When applied to the intrusion detection problem, an association rule basis that exploites the relationships between the different fields in audit trails, while a meta-learning classifier learns the signatures of attacks. Features of these two data mining techniques are extracted and used to compute models of intrusion behavior.

With the aim to improve on JAM's approach, Helmer et *al.* introduced in [5] a distributed intrusion detection architecture, complete with a datawarehouse, mobile and static agents. The mobile agent system is combined with a machine learning approach to automated discovery of concise rules from system call traces, to facilitate building, monitoring, and analyzing global, spatio-temporal views of intrusions on large distributed systems.

Sodiya proposed in [18] a distributed mobile agent-based IDS, called MSAIDS, (*Multi-Level and Secured Agent-based Intrusion Detection System*). The architecture of MSAIDS is based on a multi-level methodology where the intrusion detection process is done within two levels. Each level uses a data mining algorithm, inspired by [1], to extract patterns or associations of intrusive events.

Recently, Shyu and Sainani [16] proposed a novel data mining assisted multiagent-based intrusion detection system (DMAS-IDS) architecture. DMAS-IDS integrates a classification algorithm and the multi-agent technology in a network intrusion detection. It employs three layers, called *Host*, *Classification* and *Manager layers*. Each of these layer comprises agents capable of communicating the obtained results with each other.

Table 1 summarizes the surveyed approaches dedicated to the distributed IDS. Approaches fitting in the use of multi-agent technology trend using data mining algorithms attempt to generate a more effective distributed IDS.

In this respect, the main thrust of this paper is to propose a new distributed IDS, called **MAD-IDS** (*Mobile Agent using Data mining based Intrusion Detection System*). The MAD-IDS system integrates the data mining algorithms and the mobile agent technology, whose objectives are:

1. Improving the distributed IDS performance;
2. Detection of both known and unknown attacks with a high accuracy in a distributed environment;
3. Reduction of false alarms.

	Multi-Agent technology	Data mining algorithms
JAM (Stolfo et *al.*) [20]	Intelligent static agents	Association rules Meta-Learning Classifier
Approach of (Helmer et *al.*)[5]	Mobile and static agents	Classification algorithms Genetic algorithms
MSAIDS (Sodya) [18]	Mobile agents	Modified Apriori algorithm
DMAS-IDS (Shyu et *al.*)[16]	Intelligent distributed agents	Multi-Class Supervised Classification algorithm

Table 1. Distributed IDS

3 The MAD-IDS system

The MAD-IDS system integrates the data mining algorithms and a mobile agent technology in a network intrusion detection to detect both known and novel attacks. Thus, Figure 1 provides an overall architecture of MAD-IDS. It contains various mobile agents for collecting and analyzing massive amounts of network traffic. The distributed structure of MAD-IDS comprises different cooperatives, communicants and collaborative entities which are able to move from one station to another, called respectively: *Sniffer, Filter, Misuse Detection, Anomaly Detection, Rule Mining* and *Reporter Agent*.

Indeed, the Sniffer Agent is responsible for gathering packets from the network. Beside, the Filter Agent filters the collected data. Additionally, the Misuse Detection Agent analyzes the collected and filtered network data, to detect network connections that correspond to attacks for which signatures are available, and then to remove them from further analysis. Next, the data is fed into an Anomaly Detection Agent, which combines the advantages of agent-based distributed analysis and clustering-based intrusion detection technique with unlabeled data, to detect the anomalous connections. The Rule Mining Agent aims at providing a concise representation of the network traffic. Typically, it summarizes the network connections that are ranked highly anomalous by the Anomaly Detection Agent. The obtained set of generic association rules can be periodically fed to the Misuse Detection Agent to update its signature database allowing the detection of known attacks. Finally, the Reporter Agent generates reports and logs.

Each of these agents will be individually described in the following subsections.

3.1 The Sniffer Agent

The Sniffer Agent is the first agent to work in the system, since it connects to the network and begins to read the packets moving around. It collects all the packets and stores the collected data in a "*sniffing file*". In addition, the Sniffer Agent will be cloned and distributed throughout the network. It can duplicate itself in order to lighten the network charge. For instance, the network data stored in the sniffing file includes many features such as Src_IP (the source IP), Dst_IP (the destination IP), Src_Port (the source

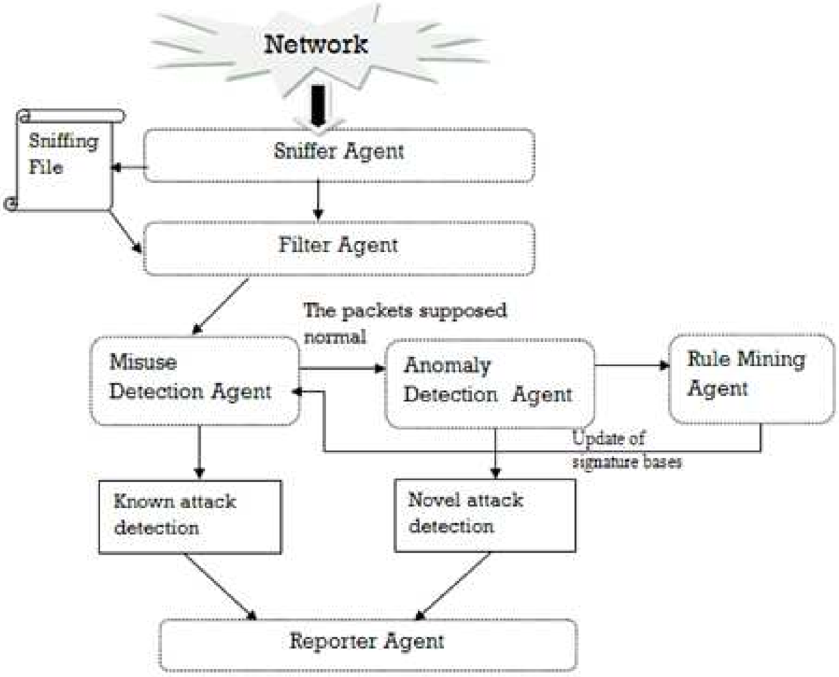


Fig. 1. The architecture of MAD-IDS

Port), Dst_Port (the destination Port) and so on. It will be the input of the next agent: the Filter Agent.

3.2 The Filter Agent

A distributed IDS must undertake to analyze a huge volumes of events collected from different sources around the network. Consequently, to be able to determine whether an intrusion is taking place, the Filter Agent aggregates and merges events stored in the sniffing file. It will treat these crude events by achieving the following tasks:

- Distinguish the various fields of the events collected in crude such as destination address and the protocol;
- Sort the events by the category of packet (TCP, UDP, ICMP, etc.) concerned by a specific kind of intrusion.

The Filter agent performs its tasks as a pretreatment phase, which precedes the analysis phase.

3.3 The Misuse Detection Agent

This kind of agent processes and analyzes the events firstly captured by the Sniffer Agent and then pre-processed by the Filter Agent. In fact, there are several techniques to perform the misuse approach [3]. Particularly, We focus on the *pattern matching* method. The Misuse Detection Agent detects known attacks in network connections. Hence, if there is a similarity between the filtered packets and attacks signatures, then the agent raises an alert to the Reporter Agent, and then removes these anomalous packets from further analysis. Attack signatures are a specific rule set provided by the Rule Mining Agent.

Although the known attacks are detected, it remains nevertheless the problem of the new attacks detection. Indeed, new kinds of attacks regularly have to be added to the signature list. The main drawback is that in case of an emerging attack, based on the recent discovery of a new security hole for instance, the IDS will ignore it since this new attack has not yet been listed in the bases of signature. Consequently, protecting a system against new attacks, while keeping an automatic and adaptive framework is of paramount importance topic in intrusion detection domain. One answer to the "adaptability" problem could rely on data mining techniques [17]. Data mining is defined as the semi-automatic discovery of patterns, associations, changes, anomalies, rules, and statistically significant structures and events in data [4]. Briefly, the data mining techniques have provided the following benefits [21]:

- *Improved variants detection*. This is especially true for anomaly detection. Not limited to pre-defined signatures, the concern with variants is not as much as before, since any deviation from a unique (normal) signature will be treated as an intrusion, including those previously unknown variants of intrusions;
- *Controlled false alarms*. Even though these are false positives, with a learning process to identify recurring sequences of false alarms, it is possible for us to filter out those normal system activities and keep the rate of false alarms at an acceptable level;
- *Reduced false dismissals*. Within data mining techniques, patterns (or signatures) of normal activities and abnormal events (intrusions) can be created automatically. It is also possible to introduce new types of attacks through an incremental learning process. As a result, more and more attacks can be detected correctly. This leads to a reduced number of false dismissals;
- *Improved efficiency*. One very attractive feature of data mining techniques is the ability to extract most meaningful information out of large amounts of data. After a step of feature extraction, the learning process can be carried out much more efficiently.

A wide variety of data mining techniques, which include unsupervised clustering [14, 21] and association rule mining [10, 12, 22] algorithms have been applied to intrusion detection. In contrast, we introduce an Anomaly Detection Agent, based on the clustering and a Mining Rules Agent, based on the association rule mining techniques.

3.4 The Anomaly Detection Agent

The Anomaly Detection Agent provides the combination of distributed IDS with clustering techniques, aiming at more efficient analysis of the massive data collected from the large network and finding new and anomalous connections (*i.e.*, anomalous behaviors).

Clustering technique groups a set of data that exhibit similar characteristics into meaningful subclasses according to some pre-defined metrics. Hence, the member, which is quite similar to another, will have the same cluster. Otherwise, the members, which are quite dissimilar each from other, will have different clusters.

Generally, the unsupervised clustering for intrusion detection aims to *i*) group behaviors together depending on their similarity and *ii*) detect groups containing only one (or very few) behavior. Such isolated behaviors are then considered as deviating from a model of normality and are therefore considered as malicious [17]. Thus, the benefits of this technique are: *i*) detecting abnormal activities automatically without too much human intervention and *ii*) the low computational complexity. Thanks to the modularized design, experimenting with clustering algorithm in unlabeled data and detection mechanisms is made easy.

Indeed, the Anomaly Detection Agent calculates the clusters C_i using an algorithm, whose first part inspired by [14]. To determine the anomalies, it assigns a number of points to each cluster C_i and sorts the clusters according to these points. The points allow to measure the degree of being a dissimilar cluster from its neighborhood. Beside, it chooses the small clusters as the candidate anomaly sets and merges these candidate sets into the clusters C_j. In order to analyze the candidate anomaly set and choosing the true anomalous connections, it measures the distance between the clusters C_j with respect their neighborhood. It chooses the shortest distance for every cluster and sorts these C_j according to the distance from large to small. Finally, the anomaly packet set will be given to a Rule Mining Agent, which is described as follows.

3.5 The Rule Mining Agent

In the intrusion detection context, the association rule mining have been found to be valuable for analyzing network traffic data [10, 12, 22]. The formalization of the association rule mining problem was initially introduced by Agrawal et *al.* [1]. Given a set of records, the objective of mining association rules is to extract all rules of the form $\mathcal{X} \Rightarrow \mathcal{Y}$ that satisfy a user-specified minimum support and minimum confidence thresholds.

Thus, a Rule Mining Agent provides the construction of a summary of anomalous connections detected by the Anomaly Detection Agent. Often times, the number of anomalous connections flagged by an IDS can be very large, thus requiring analysts to spend a large amount of time interpreting and analyzing each connection. By applying association rule mining techniques, analysts can obtain a high-level summary of anomalous connections. For example, scanning activity for a particular service can be summarized by a frequent set: $Src\_IP = \mathcal{X}, Dst\_Port = \mathcal{Y}$.

Although the association rules can detect sets of features that occur frequently in the network traffic data, the number of rules extracted can be quite large, depending on the choice of minimum support threshold. Some of the rules are redundant because they correspond to the subsets of other rules. For example, given two frequent sets:

1. Protocol=TCP, DstPort=8888,TCPflags=SYN

2. DstPort=8888,TCPflags=SYN

The first association rule is more descriptive than the second one. If the support of these two frequent sets is very close, then the second rule is redundant. Consequently, to generate association rules without redundancy, we apply the *Informative Generic Basis* ($\mathcal{IGB}$) [2]. In addition to the elimination of redundancy, the application of the $\mathcal{IGB}$ basis during an intrusion detection process provides: *the increase of the overall coverage of detectable attacks* and *the maximum convey of useful knowledge*, while being *the information lossless* [2].

An example of this rule set is shown in Figure 2.

Src_IP	Dst_IP	Src_Port	Dst_Port	Attack
206.163.37.95	160.94.179.223	192	139	Yes
206.163.37.95	160.94.179.219	195	139	Yes
206.163.37.95	160.94.179.217	180	139	Yes
206.163.37.95	160.94.179.255	199	139	Yes
206.163.37.95	160.94.179.254	186	139	Yes
206.163.37.95	160.94.179.253	177	139	Yes

$\mathcal{IGB}$ rule: {Src_IP=206.163.37.95,
Dst_Port = 139}$\Rightarrow${Attack}

Fig. 2. $\mathcal{IGB}$ rule that describe scanning activity on port 139

Therefore, once the anomalous connections are detected by the Anomaly Detection Agent, then the Rule Mining Agent is ready to mine the generic association rule set using $\mathcal{IGB}$. The benefit is the reduction of information overloading for the security analysts. Thus, the extracted rule set may be a candidate signature for addition to a signature database of the Misuse Detection Agent. This means that the database of signatures is updated regularly in order to ensure adequate protection.

3.6 The Reporter Agent

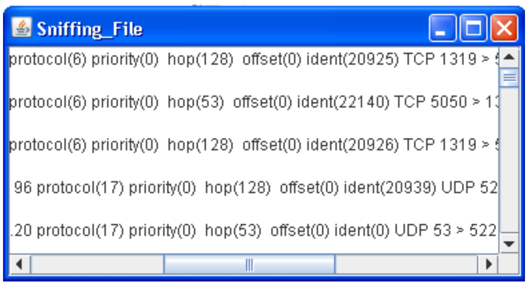
The Misuse and the Anomaly Detection Agents report their findings to the Reporter Agent which transmits them to the administrator. Whenever an intrusion is detected, it will send an alert to the system administrator. This alert can be a message on the screen or a message to a centralized machine or an alert file.

4 Experimental results

We implement MAD-IDS using Sun's Java Development Kit 1.4.1, the well known platform JADE 3.7, the Eclipse and the JPCAP 0.7.

JADE (Java Agent DEvelopment Framework) is a software Framework, which simplifies the implementation of multi-agent systems. The agent platform can be distributed by moving agents from one machine to another one. In addition, JPCAP is an open source library for capturing and sending network packets. It provides facilities to capture raw packets live from the wire and save captured packets to an off-line file.

Indeed, the Sniffer Agent based on the JPCAP library collects the network events using the "*CaptureTool*" class and saves them into a sniffing file. Thus, Figure 3 sketches an example of the sniffing file, which contain the captured packets.



```
protocol(6) priority(0)  hop(128)  offset(0) ident(20925) TCP 1319 > 5
protocol(6) priority(0)  hop(53)  offset(0) ident(22140) TCP 5050 > 13
protocol(6) priority(0)  hop(128)  offset(0) ident(20926) TCP 1319 > 5
 96 protocol(17) priority(0)  hop(128)  offset(0) ident(20939) UDP 52
.20 protocol(17) priority(0)  hop(53)  offset(0) ident(0) UDP 53 > 522
```

Fig. 3. Example of sniffing file

To assess the performance and results of the algorithms in the MAD-IDS system, we performed several experiments on a PC equipped with a 3GHz Pentium IV and 2GB of main memory running under Linux Fedora Core 6. During experiments, we partly use the traffic data DARPA [3]. It provided a standard corpus for evaluating intrusion detection systems. It also introduced more stealthy attacks, insider attacks and attacks against the Windows operating system. The audit data is split into two sets: training and test datasets. In order to simulate the distributed environment, the test data is distributed into four data sets equally which simulate the instances collected from different agents.

In fact, to evaluate the performance of an IDS, two interesting metrics are usually of use [14]: the *Detection Rate* (DR) and the *False Positive Rate* (FR).

1. The DR equals the number of correctly detected intrusions divided by the total number of intrusions in the data set;
2. the FR equals the total number of normal instances that were incorrectly considered as attacks divided by the total number of normal instances in the data set.

The value of the DR is expected to be as large as possible, while the value of the FR is expected to be as small as possible.

Table 3 show the average results for different parameter k, where k is a predefined parameter indicating the number of clusters.

Figure 4 show ROC (Receiver Operating Characteristic) [15] curve computed according to the experimental results given in Table 3, which shows the relationship between False Positive and Detection Rates. ROC curves are a way of visualizing the trade-offs between False Positive and Detection Rates.

The trade-off between False Positive and Detection Rates is inherently present in many machine learning method. By comparing these quantities against each other we can evaluate the performance invariant of the bias in the distribution of labels in the

[3] Available at: http ://www.ll.mit.edu/IST/ideval/data/data_index.html

k	Detection Rate	False Positive Rate
4	15%	0.2%
6	28%	0.8%
8	42%	1.4%
10	58%	2.3%
12	67%	5.1%
14	75%	8.2%
15	90%	16%

Table 3. False Positive *vs.* Detection Rates with the variation of k

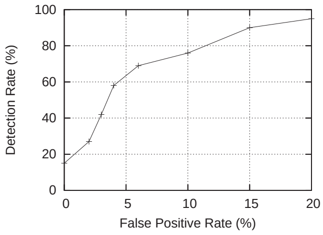


Fig. 4. The ROC Curve of False Positive *vs.* Detection Rates

data [14]. This is especially important in the intrusion detection problem. In our clustering technique, as far as the percent of largest clusters to be labeled normal was decreased, Detection Rate increased substantially since a larger number of clusters were labeled anomalous. The intrusion instances were classified correctly as intrusions. However, the False Positive Rate also increased because all the instances assigned to clusters that were previously labeled normal, were classified as intrusions as well.

5 Conclusions

In this paper, a novel distributed multi-agent IDS architecture, called MAD-IDS was presented. MAD-IDS integrates the mobile agent methodology and the data mining techniques to accommodate the special requirements in distributing IDS. We have demonstrated that the data mining techniques and in particular the unsupervised clustering algorithm and the generic association rule mining are capable of discovering anomalous connections, as well as, generating an informative summarize. The final output of MAD-IDS is concise and intuitive detection rules that can be periodically fed to the Misuse Detection Agent to update its signature database allowing the detection of known attacks. The preliminary experimental results indicated that the data mining algorithms used in MAD-IDS are feasible for detecting attacks within a distributed environment. Consequently, we can conclude that the results indicate that our proposed MAD-IDS architecture provides many favorable characteristics, such as high accuracy, good scalability and adaptability.

Other avenues for future work address the following issues:

- Testing the framework in a high speed network volume.

- Combining the data mining techniques with real homogeneous distributed system, and furthermore with heterogeneous distributed system;
- Introduction of an incremental updating mechanism for the detection agents.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings of the International Conference on Management of Data, Washington, D.C.*, pages 207–216, 1993.
2. S. Ben Yahia, G. Gasmi, and E. Mephu Nguifo. A New Generic Basis of Factual and Implicative Association Rules. *Intelligent Data Analysis (*IDA*)*, 13(4):633–656, 2009.
3. D. E. Denning. An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, 13(2):222–232, 1987.
4. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process of extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.
5. G. Helmer, J.S.K. Wong, V.G. Honavar, and L. Miller. Automated Discovery of Concise Predictive Rules for Intrusion Detection. *Journal of Systems and Software*, 60(3):165–175, 2002.
6. N. Jaisankar, R. Saravanan, and K. D. Swamy. Intelligent Intrusion Detection System Framework using Mobile Agents. *International Journal of Network Security and its Applications (IJNSA)*, 1(2):72–88, 2009.
7. P. Kannadiga and M. Zulkernine. DIDMA: A Distributed Intrusion Detection System using Mobile Agents. In *Proceedings of the 6th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2005), Towson, Maryland, USA*, pages 238–245, 2005.
8. V. Kasarekar and B. Ramamurthy. Distributed Hybrid Agent Based Intrusion Detection and Real Time Response System. In *Proceedings of the 1st International Conference on Broadband Networks, San José, California, USA*, pages 739–741, 2004.
9. F. B. Ktata, N. El Kadhi, and K. Ghèdira. Agent IDS based on Misuse Approach. *Journal of Software*, 4(6):495–507, 2009.
10. W. Lee and S. J. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th Conference on USENIX Security Symposium; San Antonio, Texas*, pages 120–132, 1998.
11. C. Li, Q. Song, and C. Zhang. MA-IDS Architecture for Distributed Intrusion Detection using Mobile Agents. In *Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004), Harbin, China*, pages 451–455, 2004.
12. T. R. Li and W. M. Pan. Intrusion Detection System Based on New Association Rule Mining Model. In *Proceedings of the International Conference on Granular Computing; Beijing, China*, pages 512–515, 2005.
13. E. Mosqueira-Rey, B. Guijarro-Berdias A. Alonso-Betanzos, D. Alonso-Ros, and J. Lago-Pieiro. A Snort-based Agent for a JADE Multi-agent Intrusion Detection System. *International Journal of Intelligent Information and Database Systems*, 3(1):107–121, 2009.
14. L. Portnoy, E. Eskin, and W. S. J. Stolfo. Intrusion Detection with Unlabeled Data using Clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia, PA*, 2001.
15. F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceeding of the 50th International Conference on Machine Learning, Madison, Wisconsin USA*, pages 445–453, 1998.

16. M.-L. Shyu and V. Sainani. A M*ultiagent-based* I*ntrusion* D*etection* S*ystem with the* S*upport of* M*ulti*-C*lass* S*upervised* C*lassification*, chapter 8, pages 127–142. Springer-Verlag US, Data Mining and Multi-agent Integration edition, 2009.
17. G. Singh, F. Masseglia, C. Fiot, A. Marascu, and P. Poncelet. Data Mining for Intrusion Detection: from Outliers to True Intrusions. In *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-09), Bangkok, Thaïlande*, pages 891–898, 2009.
18. A. S. Sodiya. Multi-Level and Secured Agent-based Intrusion Detection System. *Journal of Computing and Information Technology*, 14(3):217–223, 2006.
19. E.H. Spafford and D. Zamboni. Intrusion Detection Using Autonomous Agents. *The International Journal of Computer and Telecommunications Networking*, 34(4):547–570, 2000.
20. S. Stolfo, A.L. Prodromidis, S. Tselepis, W. Lee, D.W. Fan, and P.K. Chan. JAM: Java Agents for Meta-Learning over Distributed Databases, newport beach, california. In *Ptoceedings of the 3rd International Conference on Knowledge Discovery and Data Mining,*, pages 74–81, 1997.
21. Q. Wang and V. Megalooikonomou. A Clustering Algorithm for Intrusion Detection. In Belur V. Dasarathy, editor, *Proceedings of SPIE on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005, Orlando, Florida, USA*, pages 31–38, 2005.
22. W. Xuren, H. Famei, and X. Rongsheng. Modeling Intrusion Detection System by Discovering Association Rule in Rough Set Theory Framework. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (ACIDCA'06), Sydney, Australia,*, pages 24–29, 2006.