ISSN:0975-9646

Security Analysis Using IDs Based on Mobile Agents and Data Mining Algorithms

Chaimae Saadi, Habiba Chaoui Mejhed, Hassan Erguig

Systems Engineering Laboratory, Data Analysis and Security TeamNational School of Applied Sciences, Campus Universitaire, B.P 241, Kénitra14000, Morocco

***Abstract*--The evolution of information systems requires the implementation of a high level of security to minimize the problems associated with these systems. Previous work have integrated intrusion detection systems ‘‘traditional’’ who have shown limited results. To remedy this problem, this work merges the two detection strategies "scenario approach and behavioral approach" in IDS chosen, by using a new IDS architecture through the development of mobile agents in the JADE platform by integrating data mining algorithms types clustering and density. Indeed, tests show that this new IDS increases the rate of detection and reduce the false positive rate, which is the perspective of performing these results.**

***Keywords*--intrusion, intrusion detection system, data mining algorithms, mobile agent.**

I. INTRODUCTION

Intrusion detection systems (IDSs) are an integral part of any complete security package of a modern, well managed network system. An IDS is used to detect the intrusions by supervising a network or a system and analyzes collected flows of audit. The implementation of IDSs by deploying mobile agent technology offers a new model for intrusion detection. In fact, more research works have developed IDS using this solution with Data Mining techniques. S.stolfo and all combined intelligent agents and data mining based on the concept of relationship between different fields of traceability at the audit [1]. Considered an improvement of the previous one, the approach proposed by the work of G.Helmer and all added a data warehouse in addition to static and mobile agents. However, implementing and coding remain difficult even with the addition architecture detection using the data warehouse [2]. Other methods [3] and [4] were based on mobile agents which detection on multiple levels. They essentially use data mining algorithms to increase the time to understand. Otherwise, another solution has been implemented by I. Brahim and all, it is based on the coupling of data mining techniques and multi-agent system (MAS) [5]. In contrast to the existing work, the IDS of latter solution, called MAD-IDS, merges the two detection strategies; scenario approach and behavioral approach to exploit their advantage while other solutions have used a single detection strategy. The inspiration for this IDS [3-5,7,8] allows us to develop The inspiration for this IDS [3-5,7,8] allows us to develop a powerful new architecture which includes mobile agents in a black box with an input (capture traffic) and an output (Alert). This combination agent is achieved through a logical operation of datamining algorithms.

This work which is theoretically inspired several systems [1-6] involves to develop mobile agents using data mining algorithms in the JADE platform (java development environment agents) and integrate and configure in a IDS that exploits the advantages of both detection strategies [9], by merging them to overcome the drawbacks corresponding to each of them using an unsupervised clustering technique [10] [11] based behavioral analysis to detect unknown intrusions and also classify packets intrusive abnormal. Thus, several tests were made to summarize the network connections and generate the corresponding rules and make a proper analysis with listening to network traffic, show that this new IDS increased detection rate since a large number of clusters, and also minimized the rate of the smallest possible false positives, because these two parameters are important for evaluating the performance of an IDS.

II. WORK ENVIRONMENT

2.1 Architecture model

JADE (Java Agent Development Framework) [12][13] is a software Framework fully implemented in Java language. It is developed by Tilab for the development of multi-agent applications based on peer-to-peer communication architecture. Latest version available is 4.0.1 released in July 2010. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the latest Foundation for intelligent physical agents (FIPA) 2000 specifications. It provides a set of graphical tools that supports the debugging and deployment phases of agent development. Jade permits the intelligence, information & resources to be distributed over the network in the form of java compatible mobile devices like PDA, pagers, cell phones, smart phones, laptops or fixed desktops etc. The communication environment evolves gradually with the appearance and disappearance of various peers, (known as agents in Jade) according to their needs and requirements.

In JADE an instance of run-time environment is called a container, as it holds all the agents created in it. Collection of such containers is called a platform and it provides a homogenous layer which hides the complexity of underlying hardware & software from agents and their developers. It is compatible with J2ME, J2EE & CLDC. Its low memory requirements make it suitable for mobile devices. Nokia, Motorola, Siemens, Compaq & Hp are some of well known brands having compatibility with JADE [14]. Following figure illustrates the architecture of JADE.

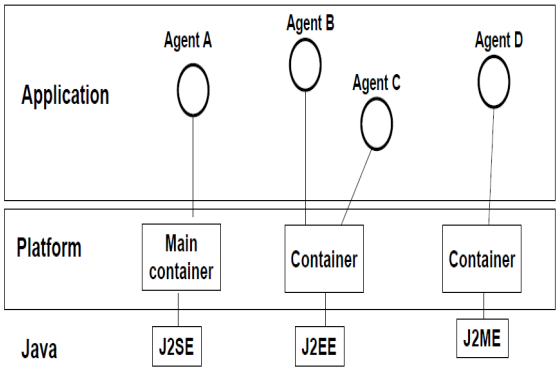

Figure1: Environment of mobile agents

2.2 Functional model

From the functional point of view, JADE provides the basic services necessary for the distributed applications Peer-To-peer in the fixed and mobile environments. JADE makes it possible for each dynamic agent, which is identified by a single name and provides a set of services, to discover other agents and to communicate with them according to the paradigm of Peer-to-Peer.

2.3 The choice of JADE

The JADE platform was selected after several studies was made on mobile agent platforms as shown in the following table ***[15]***.

Table I : platforms characteristics

Agent Development Toolkits → / Features ↓	Aglet	Voyager	JADE	Anchor	Zeus
Nature of Produce	Free, Open source	Commercial	Free, Open Source	Available in BSD license	Free, open source
Standard implemented	MASIF	.....	FIPA Compliant	SSL, X.509	FIPA compliant
Communication Technique	Synchronous , Asynchronous	All methods	Asynchronous	Asynchronous	Asynchronous
Security Mechanism	Poor	Weak	Good	Strong security	Good
Agent Mobility	Weak	Weak	Not-so-weak	Weak	Do not support
Agent Migration Mechanism	Socket	RMI	RMI	Socket	null

This table analyzed five agent development toolkits developed by different groups. On comparison Jade agent development toolkit seems most appealing. It is open source platform, purely designed in Java, provides consistency in API and supports different kinds of devices operating in internet. It provides good security features and supports sound agent mobility. Among other tools Voyager is commercial tool and doesn't comply with FIPA standards. Zeus supports FIPA standards but doesn't provide agent mobility. Aglet also doesn't comply with FIPA, lacks security and scalability. Anchor provides good security but doesn't follow FIPA specifications, thus lacks scalability. Thus JADE agent development toolkit is most balanced toolkit among the five discussed in this work.

III. NEW IDS ARCHITECTURE PROPOSED

Before planning our new architecture, we must explain the operation of mobile agents, then we will see the aspect of mobility and communication between agents.

3.1 OPERATION OF MOBILE AGENTS

The distributed structure of our system is made up of various agents co-operative, communicating and able to analyze massive quantities of the network traffic, called respectively: Interface agent, Agent of detection by Scenarios, Agent of behavioral detection, Rules extraction agent and Report agent.

3.1.1 Interface agent(IA)

At the time of its connection, IA listens the traffic network which enables him to make captures by applying filters to the capture. IA finds filters by IP address (source, destination, both), by Ethernet addresses, protocol (of any layer), etc.

3.1.2 Agent of detection by Scenarios(ADS)

ADS analyzes the collected and filtered packages, to detect connections network which correspond to attacks whose signatures are available.

These analyzes are based on the detection inferences.

Principle:

The principle of detection is based on inferences Bayes' theorem (basic result in probability theory)

Known attacks: hypotheses to explain the observed facts

- P(A |S) =P(A)×P(S |A)
- A: attack, P(A): the probability of occurrence of A.
- S: Symptoms appear as events in the audit
- P(S |A) : probability that A makes S appear
- P(A |S) : probability that S makes A appear

Method:

Calculation of the probability of each driving scenario knowing symptoms(A |S)

If the probability is highest, then an alert is triggered

3.1.3 Agent of behavioral detection(ABD)

ABD offer combination of distributed IDS with a technique of behavioral detection based on the clustering, aiming at a more effective analysis of the massive network data in order to find abnormal connections.

[16] Defines clustering as a data mining technique to group the similar data into a cluster and dissimilar data into different clusters. Clustering can be considered the most important unsupervised learning technique so as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. Clustering is "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). These clusters represents the dominant modes of the behavior of the given data objects by using a measurement of similarity [10].

Indeed, the ABD can detect the abnormal packets by using several non-supervised clustering algorithms which were

tested in [16] with the open source tool of data mining Weka.

The National Data Development and Standards Unit in Australia which describes a dataset as a set of data that is collected for a specific purpose.

The results of testing the algorithms by [16] are as following:

- All the algorithms have some ambiguity in some (noisy) data when clustered.
- The performance of K-Means algorithm is better than Hierarchical Clustering algorithm.
- K-Means and EM algorithm are very sensitive for noise in dataset. This noise makes it difficult for the algorithm to cluster data into suitable clusters, while affecting the result of the algorithm.
- K-Means algorithm is faster than other clustering algorithm and also produces quality clusters when using huge dataset.

The table [Table II] below shows a comparison between algorithms [16]:

TABLE II: COMPARISON BETWEEN THE CLUSTERING ALGORITHMS

Name	Number of clusters	Cluster instances	Number of iteration	Within clusters sum of squared errors	Time taken to build model	Log likelihood	Unclustered instances
K-means	2	42% 58%	4	2016.67520938053	0.8s		0
Em	6	5% 16% 11% 31% 15% 22%			76.94 s	-21.09024	0
Dbscan	3	40% 24% 36%			1.03s		0

IN [17] ANOTHER COMPARISON WAS MADE BETWEEN THE ALGORITHMS OF PARTITIONING: K-MEANS, K-MEDOIDS AND CLARANS.

THE TABLE [TABLE III] SUMMARIZES THIS STUDY:

TABLE III: COMPARISON BETWEEN THE PARTITIONING ALGORITHMS: K-MEANS, K-MEDOIDS AND CLARANS

Parameters	k-means	k-medoids	Clarans
Complexity	O(i k n)	O(i k (n-k)2)	O(n^2)
Efficiency	Comparatively more	Comparatively less	Comparatively more
Implementation	Easy	Complicated	complicated
Sensitive to Outliers?	Yes	No	No
Advance specification of No. of clusters 'k'	Required	Required	Required
Does initial partition affects result and Runtime?	yes	yes	Yes
Optimized for	Separated clusters	Separated clusters, small dataset	Separated clusters, large dataset

The purpose of the intrusion detection based on the K-means algorithm is [5] grouping the data according to their similarities and [6] detecting groups containing only one (or very little) of behaviors. These isolated behaviors are then regarded as deviating from the normal ones and are thus regarded as suspicious.

In this work, we will combine between several algorithms of clustering in order to profit from the advantages of the latter.

The author of [18] mixed between an algorithm based on the density: DBSCAN and an algorithm based on partitioning techniques: K-means. The k-means algorithm is sensitive to the noise, on the other hand DBSCAN gives better results concerning the generation of the clusters. Our approach is to combine these two algorithms for our tests, but also to test other improved DBSCAN versions.

3.1.4 Rules extraction agent

It aims at providing a concise representation of network traffic. Typically, it summarizes the network connections that are identified as abnormal by the ADC.

The assembly obtained of rules of generic association can be regularly added to the ADS in order to update its signature database enabling the detection of known attacks.

Although improved the detection rates and reducing the false alarm rate in the detection of attacks has been a primary goal for IDS. This task is not sufficient if the rate of network traffic is very high[14].

Therefore, when an abnormal connection is detected by the ADC, the AER is ready to use the rules of generic association using the reduction of information to analyze. The set of extracted rules can be a basis for candidate signatures to supply and update the signature database used by the ADS. This means that the signature database is regularly updated to ensure adequate protection.

3.1.5 Report agent

It is the tool of the transmission of the messages (report, alarms) to the system administrator.

3.2 NEW architecture proposed

The following figure represents the architecture which we propose to detect the intrusions and the false positive ones, while basing itself on a combination of several agents.

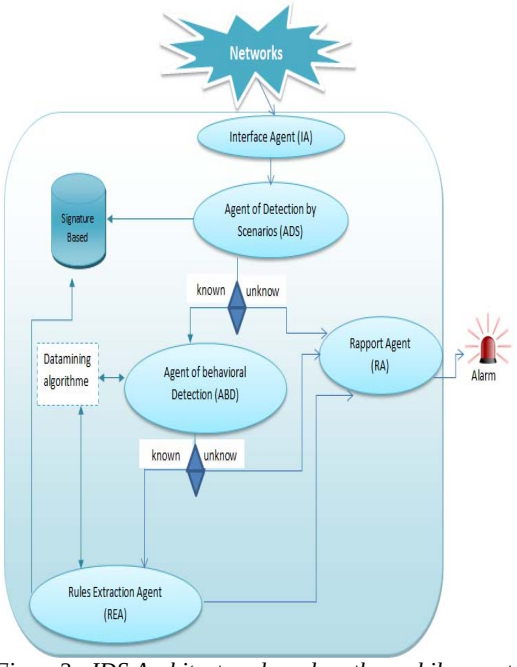

Figure2: IDS Architecture based on the mobile agents

This new architecture allows to group agents in a black box with an input (capture traffic) and an output (Alert). This combination of agents is achieved through a logical operation of datamining algorithms.

The architecture of our system contains various mobile agents for the collection and the analysis of the traffic on the network. Its distributed structure understands the various cooperatives, and entities of collaboration which are able to pass from a station to another. We integrated the data mining techniques in particular, the non-supervised algorithms and generic association rule mining which are able to discover abnormal connections.

3.3 COMMUNICATION BETWEEN MOBILE AGENTS

In this architecture, after the creation of all agents, interface agent (AI) captures packets from different network interfaces and prepares these packets to send to the agent detection scenarios. Interfaces can be Ethernet, SLIP, PPP ... etc.

When the agent detection scenarios receives the packets, it compares with existing signatures in the library of signatures intrusion; if a signature is detected, it sends a message to the agent report to create an alert and warn the administrator through different means. However, if the result of processing the detection agent scenario is considered normal, it sends the result to the agent behavioral detection.

After treatment, if this behavior is identified as normal, ADC sends a message to the AR to inform that the detected behavior is normal. Else, it sends a message to the AER to create new rules for updating the signature database and at the end, it sends a message to the AR to trigger an alert.

Log files and reports created by the reporting agent will be used thereafter by the administrator to create a comprehensive report on the treatment of this behavior manually or automatically by using a script.

Figure3 Illustrates the procedure of messages exchange between agents:

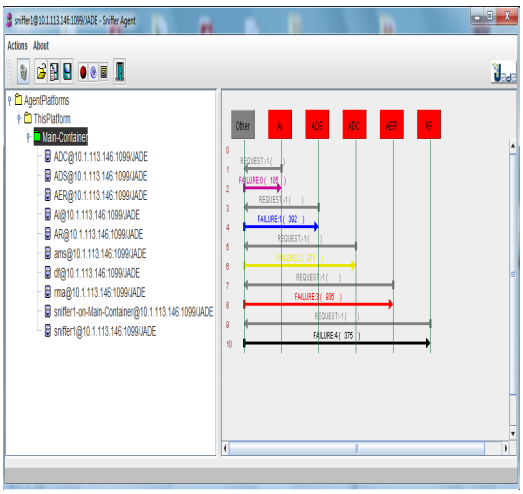

Figure 3: Exchange of message between mobile agents

The ACLMessage class represents messages which can be exchanged by the agents. The message communication is asynchronous. When an agent wants to send a message, it must create a new object ACLMessage, complete these fields with appropriate values and then call the send () method. When an agent wants to receive a message, it must use the receive () method or blockingReceive () method.

All attributes of the class ACLMessage can be obtained and modified by set / get () methods. The content of the messages is an interaction between mobile agents and their environments.

IV. TEST AND RESULTS

4.1 Test environment:

The development of this system was using Sun Java Develop Kit 7, the JADE platform 3.7 (Java Agent Development) which simplifies the implementation of multi-agent systems, and the open source library used is JPCAP0.7.

4.1.1 Working model

Architecture with which we tested our system is basically constituted of a LAN linked with a firewall which was added our intrusion detection system (IDS).

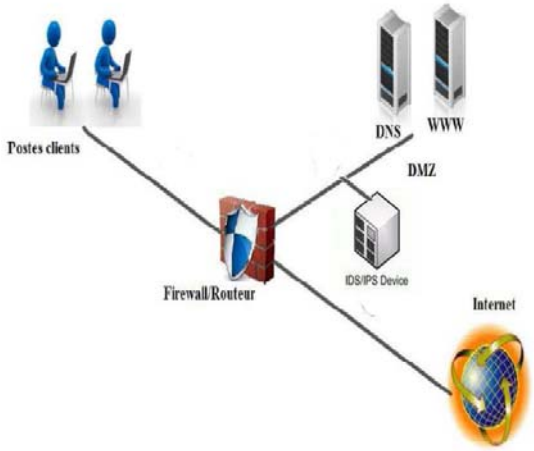

Figure 4: Proposed working

We can divide our network into three areas:

- Internal Zone: Ethernet which we put the client machine system
- External Zone: internet practically
- Demilitarized Zone: where was placed the web servers

The zones are connected together by a router and protected by our intrusion detection system

The purpose of this test was to block a website hosted locally on an Ubuntu Machine (Ubuntu 64 Bit/RAM 4096 MB/ Processors cores 2/ VT-x Technologies /AMD-V, Nested Paging, PAE / NX HDD Over 20 GO / 2 cards virtual networks).

This test applies under the TCP protocol to send a succession of SYN requests to the target.

4.1.2 Number of events detected in 24hrs

Among attacks or intrusions that have been launched since the hacker machine, we find advanced scans with the reference tool JPCAP on all machines in the network where the tests were done.

In figure 5 We report the curve of the number of events detected for 24 hours by a single detector 'sonsor '.

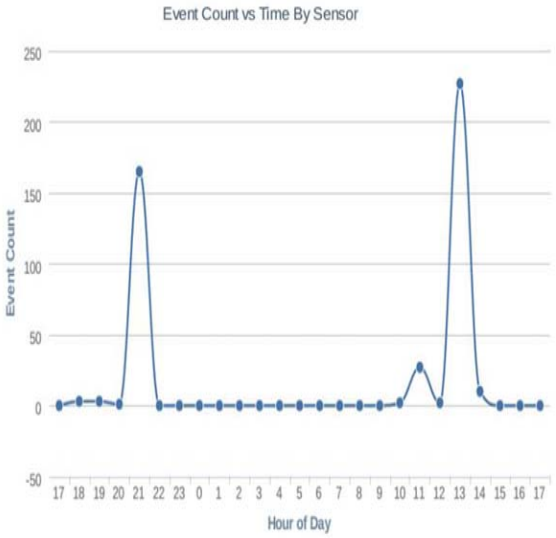

Figure 5 Number of events detected in 24 hours by a single detector (sensor)

Sniffing was launched using a network card in promiscuous mode. This mode refers to a configuration of the network card, which allows it to accept all received packets, even those that are not addressed. This mode is generally used to listen to the network traffic.

4.1.3 Type of intrusion severity

As shown in the graph of Fig.6. The detection was high in both periods of detection previously mentioned.

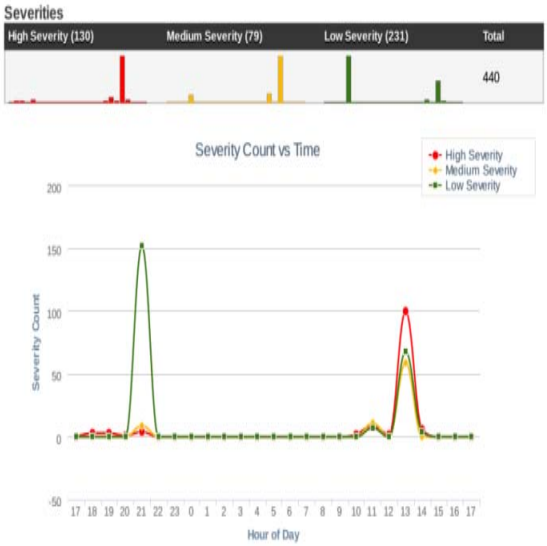

Figure 6 Type of intrusion severity detected and their name

The test for severity of attacks or intrusions detected showed three types of severity (high, medium, low). Indeed, the IDS detect 130 intrusions of high type, 79 of medium type and 231 of low type.

4.1.4 Protocol type used

A computer attack can use one of the following protocols: TCP, UDP, ICMP.

Figure7 shows the types of protocols used while the attacks or intrusions are detected.

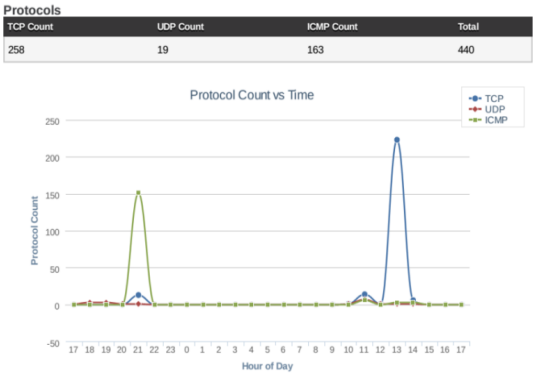

Figure 7: Type of protocols used during intrusion

After analyzing the network Traffic by JPCAP, we obtained in total 440 protocols during intrusion broken down into 258 for TCP type, 163 for ICMP type and 19 for UDP type.

4.2 Résultat

The experiments conducted in this work have used data traffic from Dataset KDD and DARPA.

DARPA'98 data provided by the DARPA Lincoln Lab School American technology MIT (Massachusetts Institute of Technology) have been formatted to create a database called KDD'99.

In this work, we use the dataset NSL-KDD, which solves some problems inherent data KDD'99 (elimination of erroneous data and repetitions ...).

The characteristics of the algorithm DensityBasedClusterer and K-Means in the process of intrusion detection applied to the NSL-KDD dataset are shown in the following tables:

Table IV : characteristics of the algorithm K-Means in the process of intrusion detection abnormal behaviors

Algorithm Name	k-means
Type	Distance
Number of iterations	9
abnormal connection	9695
normal connection	15497
Run Time	3.04 seconds

Table V : characteristics of the algorithm DensityBasedClusterer in the process of intrusion detection abnormal behaviors

Algorithm Name	DensityBasedClusterer
Type	Density
Epsilon	0.9
Minpoints	6
Number of generated clusters	168
Run Time	519.85 seconds
Number of non-clustered instance	505

After application of the K-Means algorithm and DensityBasedClusterer on the NSL KDD dataset, we

obtained the results grouped in Table 3 and illustrated in Figure 6 to evaluate the performance of an IDS, two parameters are generally used, the detection rate (DR) and false positive rate (FR).

The experiments consist in making a deterministic choice of datamining algorithms namely K-means and DensityBasedClusterer.

TABLE VI: EXPERIMENTS OF DETECTION INTRUSION AND FALSE POSITIVE

Algorithme	Full data	Normal	Anomaly	False positive	False negative
k-means	125973	62%	38%	23%	24%
DensityBasedClusterer	125973	58%	42%	14%	15%

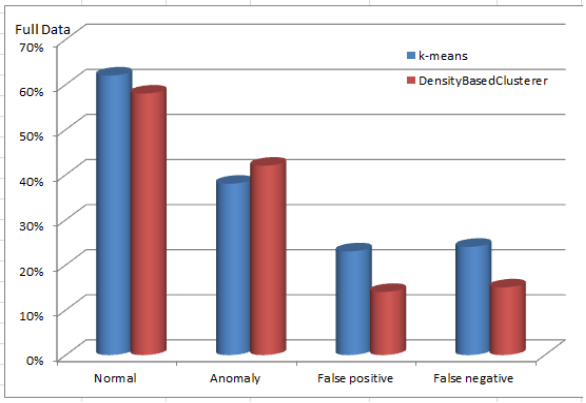

Figure 8: False positives and false negatives

It is noted that the intrusion detection with the DensityBasedClusterer is more efficient than the detection based on k-means. However, the rate of false positives and false negatives was reduced by using DensityBasedClusterer.

The system is used to detect the intrusion and to reduce the rates of false positives and false negatives by using the DensityBasedClusterer algorithm together with the interaction and the exchange of information between mobile agents.

V. CONCLUSION AND FUTURE WORKS

This paper presents a new type of distributed IDS that exploits the advantages of both detection strategies, the scenario approach and the behavioral approach. The achievement of this IDS is based on the development of mobile agents using datamining algorithms under the JADE platform. The use of unsupervised clustering and density technology allows to accurately capture the real behavior of network traffic in order to detect abnormal behavior and transform a rule to update the library signature. Mobile agents developed under the JADE platform are effective for security, flexibility and the ability of detection in distributed IDS.

This new IDS was able to achieve a high rates of detection and a reduced false positive and false negative rate as small as possible, by using K-means and density based clusterin.

From this study, it can be concluded that the results are satisfactory. However, in the next work, we will create a new data mining algorithm to performing our study.

REFERENCES

[1] 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.

[2] G. Helmer, J.S.K. Wong, V.G. Honavar, and L. Miller. Automated Discovery of ConcisePredictive Rules for Intrusion Detection. Journal of Systems and Software, 60(3):165–175, 2002.

[3] A. S. Sodiya. Multi-Level and Secured Agent-based Intrusion Detection System. JournalOf Computing and InformationTechnology,14(3):217–223, 2006.

[4] M.-L. Shyu and V. Sainani. A Multiagent-based Intrusion Detection System with the Support of Multi-Class Supervised Classification, chapter 8, pages 127–142. SpringerVerlagUS, Data Miningand Multi-agent Integrationedition, 2009.

[5] Imen Brahmi, Sadok Ben Yahia, and Pascal Poncelet. 2010. MAD-IDS: novel intrusion detection system using mobile agents and data mining approaches. In Proceedings of the 2010 Pacific Asia conference on Intelligence and Security Informatics (PAISI'10 Springer-Verlag, Berlin, Heidelberg, 73-76. DOI=10.1007/978-3-642-13601-6_9 http://dx.doi.org/10.1007/978-3-642-13601-6_9

[6] Eugene H. Spafford and Diego Zamboni, Intrusion detection using autonomous agents. Computer Networks, 34(4):547-570, October 2000.

[7] Mohamad Eid American University of Beirut, Department of Electrical and Computer Engineering, A New Mobile Agent-Based Intrusion Detection System Using Distributed Sensors P.O.Box 11-0236 Beirut 1107 2020 Lebanon.

[8] Zhihao PENG,Guanyu LI, Faculty of Information Science and Technology ,Dalian Maritime University, Department of Computer Science, An Intelligent Immunity-based Model for Distributed Intrusion Detection, Dalian 116626, China

[9] Cognitives, S., & Avanc, I. (n.d.). Études des principaux algorithmes de data mining.

[10] Guillaume CALAS Spécialisation Sciences Cognitives et InformatiqueAvancée 14-16 rueVoltaire,94270 Le Kremlin-Bicêtre,France

[11] Nguyen G., Dang T.T., Hluchy L., Laclavik M., Balogh Z.and Budinska I., 'Agent Platform Evaluation and Comparison', Published by Institute of informatics, Slovak Academy of Sciences, Pellucid 5FP IST -2001-34519, June 2002.

[12] Vitabile S., Conti V., Militello C., Sorbello F., 'An extended JADE-S based framework for Developing Secure Multi-Agent Systems', published in Computer Standards & Interfaces, Vol. 31, pp. 913–930, 2009.

[13] Bellifemine F., Caire G., Poggi A. and Rimassa G. (2003), 'JADE: A white Paper'. Available at http://exp.telecomitalialab.com , exp, vol. 3, No. 3, September 2003.

[14] Dimple Juneja1, A.K. Sharma3 M.M.University,Mullana (Ambala), Haryana, India Y.M.C.A University of Science and Technology, Faridabad , Haryana , India, International Journal of Advancements in Technology http://ijict.org/ ISSN 0976-4860 Vol 2, No 1 (January 2011) ©IJoAT 158 Agent Development Toolkits Aarti Singh 1

[15] Verma, M., Srivastava, M., Chack, N., Diswar, A. K., & Gupta, N. (2012). A Comparative Study of Various Clustering Algorithms in Data Mining Manish Verma , Mauly Srivastava , Neha Chack , Atul Kumar Diswar , Nidhi Gupta, 2(3), 1379–1384.

[16] Gandhi, G., & Srivastava, R. (2014). Review Paper: A Comparative Study on Partitioning Techniques of Clustering Algorithms, 87(9), 10–13.

[17] Erman, J., Arlitt, M., Mahanti, A., Methodologies, I. C., & Recognition, P. (n.d.). Traffic Classification Using Clustering Algorithms, 281–286.

[18] Khan, Kamran; Rehman, Saif Ur; Aziz, Kamran; Fong, Simon; Sarasvady, S.; Vishwa, Amrita, "DBSCAN: Past, present and future," Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the , vol., no., pp.232,238, 17-19 Feb. 2014 doi: 10.1109/ICADIWT.2014.6814687