New Approach for Detecting Intrusions

Mohammed Chennoufi[1], Fatima Bendella[2]

(1 Laboratory (SIMPA) Dept.Informatique, Faculty of Science, University of Science and Technology of Oran-USTO
Algeria. BP 1505, 31000
chennoufimohamed5@yahoo.fr

2 Laboratory (SIMPA) Dept.Informatique, Faculty of Science, University of Science and Technology of Oran-USTO
Algeria. BP 1505, 31000
bendella@univ-usto.dz)

Abstract— This paper describes how multi-agent systems can help to solve a complex problem such as security and more precisely intrusion detection. Intrusion Detection System (I.D.S) is a component of the security infrastructure designed to detect violations of security policy. Most of the intrusions can be localized either by considering of models "pattern" of user activities (non-behavioral approach) or by considering the audit log (behavioral approach). False positives and false negatives are considered as the major disadvantages of these approaches. We consider that good I.D.S should respond to the characteristics of intelligent agents such as autonomy, distribution and communication.

For this we suggest a new approach based on multi-agent systems (M.A.S), which incorporates the characteristics of intelligent agents (automatic learning of new attacks) so that decisions taken by the system are the result of a work group of agents and makes IDS more flexible and reliable. This approach is applied to a large data source and requires a previous work (pretreatment).

Index Terms— Security, attack, I.D.S, K.D.D, M.A.S, MLP, cognitive agent, learning.

1 INTRODUCTION

When the Internet was created, the main challenge was to enable data transmission. This objective was achieved, but at the expense in accordance with the security of users and data of organizations. They agree to take the risk because the security is difficult which makes their computer systems vulnerable to attacks. Various tools to prevent these attacks or reduce their severity, but no solution can be considered satisfactory and complete. The I.D.S is one of the most effective tools to detect I ntrusions or attempted intrusions by user behavior or by the recognition of attacks from the stream of the network data. This last is to locate abnormal and suspected activities on the analysed target (network or host) [1].

Various methods and approaches have been adopted for the design of intrusion detection systems.

Our objective is to design an intelligent tool capable of detecting new intrusions while trying to solve one main problem of IDS which is the very large amount of data. For this, we suggest a new approach based on multi-agent systems (M.A.S), which incorporates the features of intelligent agents (learning new attacks). Our approach is applied to the data source KDD 99 Knowledge Discovery and Data Mining [2].

This article is organized as follows: in the first section, we present intrusion detection systems and their link with the SMA. In the second section, we discuss previous work with the scenario method. The third section is devoted to the presentation of our architecture based on M.A.S with a pre-processing module of our comprehensive data and a supervised learning of our cognitive agent. A conclusion and an outlook are presented in the fourth section.

2 INTRUSION DETECTION SYSTEM

An intrusion detection system is a tool that identifies abnormal activity on the analyzed target and to have prevention on the risks of intrusion. They are designed to analyze large volumes of data [3]. There are two main approaches to detect intrusions [4] [5] [6].

1) The behavioural approach (Anomaly Detection).
2) The non-behavioural approach (scenario).

The first approach is based on the assumption that the exploitation of a break in the system requires abnormal use of the latter and thus unusual behaviour of the user. The second approach relies on knowledge of techniques used by attackers to obtain typical scenarios. The best known and most easily understood method in this approach is pattern matching. It is based on pattern search (string or byte sequences) in the data stream.

For the advantages and disadvantages of each approach we have table 1.

TABLE 1
COMPARISON BETWEEN THE TWO APPROACHES.

	Advantage	*Disadvantage*
Behavioural Approach	-Detect unknown attacks	-Define a profile is not an easy taskToo many false positives
Not Behavioural Approach	-Can detect attacks that occurred in the past	- The attacks not listed are not detected
	- Can take into account the exact behavior of potential attackers	- It is difficult to formalize a scenario attack in a model system independent Too many false-negative

2.1 Different types of IDS

The intrusion detection system or IDS can be classified into three major categories according to whether they are committed to monitor

- Network IDS or NIDS (Network based IDS).
- System IDS or HIDS (Host based IDS).
- Hybrid IDS (NIDS and HIDS).

NIDS: are tools that analyze network traffic, they generally include a sensor that listens on the network segment to be monitored and an engine that performs traffic analysis to detect signatures attacks or differences facing the reference model.

HIDS: Their mission is to analyze system logs, control access to system calls and check file integrity. HIDS can rely on these auditing features, clean or not the operating system, for integrity checking, and generate alerts. They are unable to detect attacks exploiting the weaknesses of the intellectual property system stack, usually by denial of service as a SYN flood or other.

So a hybrid is ideal, all by improving the basic algorithms of detection and minimizes false positives, to identify complex attack scenarios. We can classify IDS according to various criteria. These can be used to select the most appropriate to the IDS needs. Some classifications are based on the behaviour of the IDS, some of their information sources; another classification based on their frequency of use of IDS with active or passive response is given.

2.2 Related works

To adapt to changing security needs due to changes in networks, new intrusion detection systems must offer features such as adaptability, flexibility, distribution, autonomy, communication and cooperation. If we compare these characteristics with the different properties of intelligent agents (autonomy, adaptability, responsiveness,), it is very clear that SMA is very appropriate to the problem of intrusion detection [7][8][9][10]. Many attacks are caused by abnormal behavior of network elements, hence the need to distribute the IDS functionalities to several entities.

In [11] the author has designed a multi-agent system for intrusion detection. This model is based on several layers according to a hierarchical model, extra and intranet.He worked on a scenario approach, his model is based on reactive agents. It does not detect new attacks.

In [12] the author has designed an architecture based on 4 well distributed agents. The approach used is based on the host, its security model an asymmetric cryptography.This key exchange between hosts can be broken if the attacker has a depth knowledge on cryptography.

Detter [13] uses an architecture based on the network by placing a agent motor at each location. It is made of layers distributed to operate over arrange of distributed agent engines. This architecture takes advantage of the mobile agent paradigm to implement a system capable of an efficient and flexible distribution of tasks of analysis, monitoring, and the integration of existing detection techniques.

In [14] the authors suggest to extend their system with a model of case-based reasoning for learning new attacks. They propose to integrate to different agents (With the exception of the agent manager for Security Policy) a learning function based on the resemblance and similarities between past attacks and new attacks. Their model did not produce a result.

Brahimi [15] has developed an IDS based on mobile agents and on data Manning, where an update to the signing table is performed by data mining.

In [16] the author uses the approach NIDS. Its architecture is based on a simulation based on KDD. He used an algorithm through reinforcement to detect new attacks, but his model did not give good results. There is a risk of convergence on unbalanced K.D.D.

Raoui [17] has developed an IDS on a distributed platform based on the M.A.S. He used two types of reactive agents to detect known attacks and cognitive agents to detect unknown attacks (one agent detects viruses, Trojans ... the other).

3 SUGGESTED ARCHITECTURE

In literature the network security experts say that 99% of new attacks that variations attack derived from known attacks; then how to detect new attacks?

The autonomy of agent such as learning can help solve this problem, since an agent perceives its environment and acts on it. So it learns from its own environment. We suggest the following architecture for NIDS (Fig1).

1. Administrator agent extracts the raw data source for the database. Some authors use the Kdd99, because it requires a prior work.
2. Then passes this data source by a decoding module (pretreatment). The purpose of this pretreatment is to make our signature-base balanced (equal connection number of TCP-IP for each class of attack) and non-redundant in order to achieve good learning. This module consists of several steps

that run in succession (to be presented in detail in the next paragraph).

3. The learning agent is used to classify the different types of attacks and normal connections by exploiting the power of the SMA (supervised learning).

4. The agent test 2 enables to the classification module generated by the learning and decoding data source module must verify the performance of the system to detect new attacks that will be stored in the database via the Administrator agent.

On a new connection, the information collected by the network will be analyzed by 2 agents. In case of detecting a new attack, the administrator agent performs an update of the database through a communication between agents.

5. The administrator agent sends a message Send () to agent test 1, asking whether or not an intrusion is in question to sent new attacks if they exist.

6. The agent test 1 sends the message Answer (Normal Attack, unknown) to the administrator. In case of a non-existing connection in the database, another communication between agents is triggered.

7. The administrator agent sends a message Send () to the Agent Test 2, requesting further attacks if they exist.

8. The agent test 2 sends the message Answer (Normal, Attack) to the Administrator (the recognized class).

9. In case of another attack detection, the administrator agent performs an update to the database.

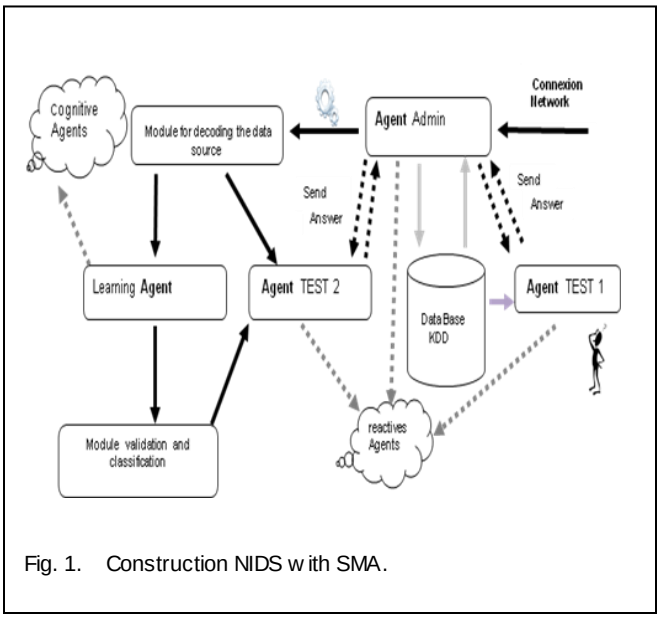

Fig. 1. Construction NIDS with SMA.

3.1 Module to decoding the data source

Due to the enormous size of the database Kddcup 99original [18] (about 5,000,000 connections, transforming the dump TCP traffic in 41 attributes), most authors have carried out their experiments using a sample of the original data set. This sample contains about 10% of connections (so far no signature-based standard. Each project implements its own base, such as SNORT [19]).

The main classes of attacks for KDD99 data set are:

1. Probing: surveillance and survey, port scan (nmap, satan ... ..)
2. DOS (Denial Of Service): Denial of Service (SYN Flooding, smurf3 ...)
3. R2L (Remote to User): unauthorized access from a remote machine (syn floo ding, smurf ....)
4. U2R (User to Root): unauthorized access for the privilege of an administrator.

We find five classes of decisions: four types of intrusions and one Normal.

Few authors use signature-based kdd99 because it requires an important previous work (pretreatment). The steps that run in succession are:

1. Elimination of redundant records: the base KDD99 contains many duplicate records.
2. Selection of attributes
3. Scanning the various fields symbolic and processing records in a format appropriate treatment.
4. Standardization of the great values

After analyzing the data using the SAS tool company (Ringuedué, 2011), we found that such as some columns do not change, for example the column "num_outbound_cmds", which means that we will eliminate this column in the training set (value = 0).

After calculation of similarity between the different columns, we noticed that some columns are similar to a rate of 99%, which implies that there is redundancy of information in the training set. Thus, these columns will be represented by a single column, which gives us a transformation of TCP dump of 41 attributes and 24 attributes.

The step of coding for kdd99 is to convert the symbolic attributes in digital. The procedure is as follow:

Initially, the names of attacks (as buffer_overflow, guess_passwd, etc...) are stored in one of five classes: 1 for Normal, 2 back, 3 for Probing, 4 and 5 U2R R2L. For the 2 nd, 3rd and 4th columns, the symbolic attributes like protocol_type (3 symbols), service (70 different symbols) and flag (11 different symbols) are transformed into integer values.

A normalization of data is required. Most columns have values between 0 and 1. Others,which represent the number of bytes, their values are between 0 and 2661205 and that represented in kilobytes. The first column (time in seconds) will be in minutes in order to reduce the values for learning and standardizing.

After considering the distribution of data according to the protocol ,the service, and the types of attacks, we found that the data are not well balanced (fig.2, fig.3). The learning of imbalanced data is very important. A data set is unbalanced when the terms distribution of the class is very different from the uniform distribution. However to achieve a good learning instead of taking 60% of KDD and 40% for the test, we took 60% of each type of attacks (60% of normal, 60% of DOS 60% of probing, 60% of U2R, 60% of R2L) and the same for the test

40% of each to ensure a balanced basis.
We noticed that U2R and R2L are less important than the other classes, that's why we took 10% for the first three classes (TAB 2).

TABLE 2
BALANCED KDD.

	Normal	Dos	Probing	U2r	R21
TCP-IP in First	97278	391458	4107	52	1126
TCP-IP after	87832	54572	2131	52	999
TCP-IP Bbalanced number 1	52700	32744	1279	31	600
TCP-IP Balanced number 2	5270 10%	3274 10%	127 10%	31	600

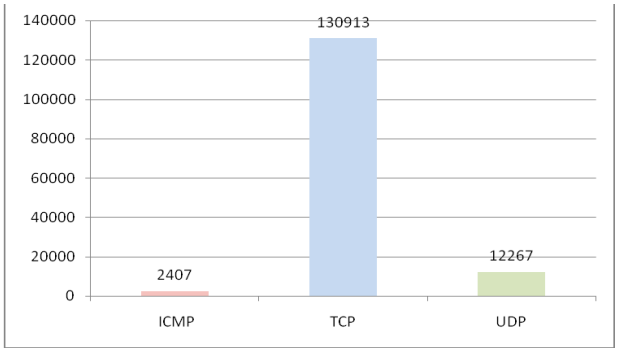

Fig. 2. Distribution of data according of protocol.

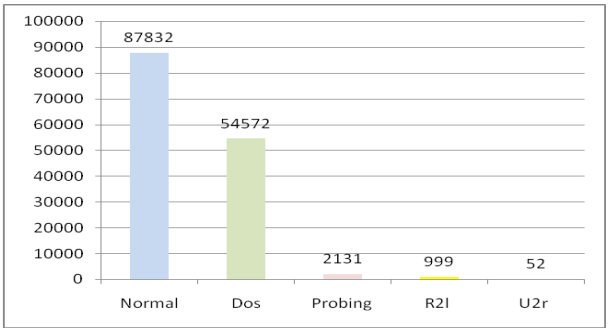

Fig. 3. Distribution of data by type of attack

3.2 Learning Agent with MLP neural networks

The autonomy of agents is considered as a powerful feature of M.A.S. We took advantage of these powerful agents for the detection of unknown attacks. We chose to use a supervised learning technique for the mutli-layer perceptrons MLP [20]. We believe that supervised learning is compatible with our architecture because the connection of the balanced K.D.D base is desired output for the 24 attributes .
Our neural network consists of an input layer of 24 neurons that represent the 24 columns (pretreated attributes K.D.D), an output layer of 05 neurons (1 + 4 normal attacks) and a hidden layer of few neurons all depending on experiments (Fig. 4).

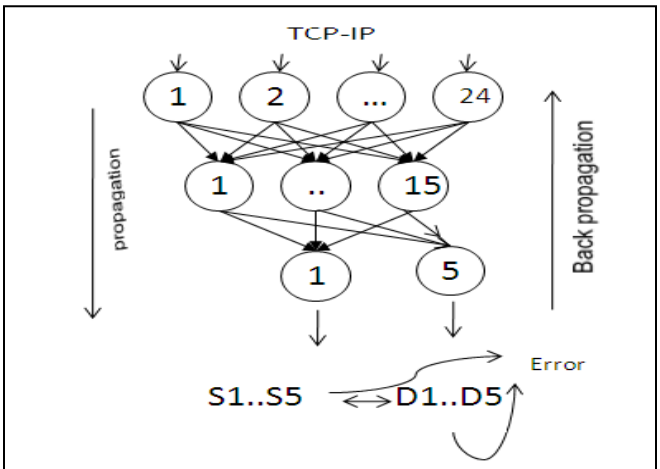

Fig. 4. MLP-NIDS.

3.3 Results

We set the threshold value to 0.0 and the value of learning step to 0.001.We varied the number of neurons in the hidden layer and the number of iterations. After several experiments, we obtained results that show the rate of learning (classification) and testing. There is an improved classification rate compared to the first tests (TAB 3).

TABLE 3
BEST CLASSIFICATION RATE.

Hidden layer	100	10	30	20	15	15
Step	0.01	0.01	0.001	0.001	0.01	0.001
Iteration	1000	100	1000	2000	1500	5000
Learning rate	79.65%	80.91%	81.96 %	84.36%	85.62 %	98.44 %
Testing rates	79.67%	80.91%	82.03 %	84.38%	85.69 %	98.44 %

Figure 5 shows the classification rate for each class for the last and best test.

We notice an improvement in the recognition of each class, except for the U2R class that represents a rate of 5.26%, which seems obvious to us for the following two reasons:

1) This is a very rare type of attack.
2) It accounts for 31 connections in our KDD (9302 connections).

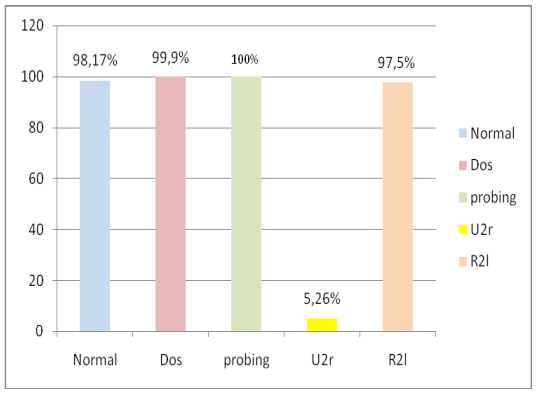

Fig. 5. The classification rate for each class

4 Conclusion

In this paper, we have first presented a statement of the art in the I.D.S. We noticed that the prior works on the non behavioural approach use reactive agents to detect know attacks. Few authors use cognitive agents to detect new attacks.

The suggest architecture is to enable us to detect new attacks as well as to take advantage of the M.A.S., like the autonomy (the agent learns from the environment and the attack classification). The administrator agent takes out the data source (sound effects K.D.D) and sends it to the decoding module (pretreatement) so as to eliminate the redundancies, the balance and the coding. The learning agent takes out the data from the pretreatment module and classifies the different types of attacks as well as the standard connections.

Ovring to the classification module, generated by the learning and the decoding data source module, the agent test 2 will allow verifying the performance of the system for detecting new attacks stored in the base.

In the future, we are thinking of another learning for our cognitive agent (Q learning), to compare it with the results of the implemented M.L.P and to make our approach active, taking into account the performance time. It is also to ensure an improvement in both the performance and the reliability of the system.

Acknowledgment

The authors are particularly grateful to Mohammed MANKOUR.

References

[1] J.S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," *Neurocomputing – Algorithms, Architectures and Appliations*, F. Fogelman-Soulie and J. Herault, eds., NATO ASI Series F68, Berlin: Springer-Verlag, pp. 227-236, 1989. (Book style with paper title and editor)

[2] W.-K. Chen, *Linear Networks and Systems*. Belmont, Calif.: Wadsworth, pp. 123-135, 1993. (Book style)

[3] H. Poor, "A Hypertext History of Multiuser Dimensions," *MUD History*, http://www.ccs.neu.edu/home/pb/mud-history.html. 1986. (URL link *include year)

[4] K. Elissa, "An Overview of Decision Theory," unpublished. (Unplublished manuscript)

[5] R. Nicole, "The Last Word on Decision Theory," *J. Computer Vision*, submitted for publication. (Pending publication)

[6] C. J. Kaufman, Rocky Mountain Research Laboratories, Boulder, Colo., personal communication, 1992. (Personal communication)

[7] D.S. Coming and O.G. Staadt, "Velocity-Aligned Discrete Oriented Polytopes for Dynamic Collision Detection," *IEEE Trans. Visualization and Computer Graphics*, vol. 14, no. 1, pp. 1-12, Jan/Feb 2008, doi:10.1109/TVCG.2007.70405. (IEEE Transactions)

[8] S.P. Bingulac, "On the Compatibility of Adaptive Controllers," *Proc. Fourth Ann. Allerton Conf. Circuits and Systems Theory*, pp. 8-16, 1994. (Conference proceedings)

[9] H. Goto, Y. Hasegawa, and M. Tanaka, "Efficient Scheduling Focusing on the Duality of MPL Representation," *Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS '07)*, pp. 57-64, Apr. 2007, doi:10.1109/SCIS.2007.367670. (Conference proceedings)

[10] J. Williams, "Narrow-Band Analyzer," PhD dissertation, Dept. of Electrical Eng., Harvard Univ., Cambridge, Mass., 1993. (Thesis or dissertation)

[11] E.E. Reber, R.L. Michell, and C.J. Carter, "Oxygen Absorption in the Earth's Atmosphere," Technical Report TR-0200 (420-46)-3, Aerospace Corp., Los Angeles, Calif., Nov. 1988. (Technical report with report number)

[12] L. Hubert and P. Arabie, "Comparing Partitions," *J. Classification*, vol. 2, no. 4, pp. 193-218, Apr. 1985. (Journal or magazine citation)

[13] R.J. Vidmar, "On the Use of Atmospheric Plasmas as Electromagnetic Reflectors," *IEEE Trans. Plasma Science*, vol. 21, no. 3, pp. 876-880, available at http://www.halcyon.com/pub/journals/21ps03-vidmar, Aug. 1992. (URL for Transaction, journal, or magzine)

[14] J.M.P. Martinez, R.B. Llavori, M.J.A. Cabo, and T.B. Pedersen, "Integrating Data Warehouses with Web Data: A Survey," *IEEE Trans. Knowledge and Data Eng.*, preprint, 21 Dec. 2007, doi:10.1109/TKDE.2007.190746.(PrePrint)