A Mobile Agents and Artificial Neural Networks for Intrusion Detection

Nabil EL KADHI, Karim HADJAR, Nahla EL ZANT
Ahlia University Manama Bahrain, EPITECH Paris France
Email: {nelkadhi, nahla, karimh}@ahliauniversity.edu.bh

Abstract— **Nowadays any intrusion detection system should include decision making feature. Each network administrator, in his everyday job, is overwhelmed with a big number of events and alerts. It is a challenge to be able to take correct decisions and to classify events according to their accuracy. That's why we need to provide the administrator with the right tools in order to help him taking the correct decision. For this purpose, we suggest an Artificial Neural Networks (ANN) architecture for decision making within intrusion detection systems. Having in mind our IMA_IDS solution [20] that presents a global agent architecture for enhanced intrusion network based solution, we are including ANN as a major decision algorithm using the learning and adaptive features of ANN. This inclusion aims to increase respectively efficiency, by reducing the fault positive, and detection capabilities by allowing detection with partial available information on the network status.**

Index Terms— **Artificial Neural Networks, Distributed Intrusion Detection System, Anomaly Detection, Signature based IDS.**

I. Introduction

Security threats for computer networks have increased significantly. Among all security issues, intrusion is the most critical and widespread. Intrusion can be defined as any action that is not legally allowed for a user to take towards an information system, compromise, or cause harm to a network. Intrusion detection, appeared in 1980 [1]. It is a process of detecting and tracing inappropriate, incorrect, or anomalous activity targeted at computing and networking resources. Abstract intrusion detection model was proposed in 1987 by Denning [2]. Intrusion Detection System (IDS) is commonly, a software that automates the intrusion detection process and detects possible intrusions. IDS are usually divided into two groups according to the analyzed events:

- Host Based IDS (HIDS): perform their analysis on information collected at a single host by the audit trails. HIDS are designed for monitoring a single computer system looking very specifically at what is happening on that machine via the log files and/or the internal auditing systems.
- Network Based IDS (NIDS): rely on information obtained by monitoring the stream of data exchanged between computers. NIDS are used to detect intrusions across an entire network. These systems must be placed in the network such that they can see all passing traffic.

The usual approach for an IDS is to set up sensors to collect the data. Then to pass it to an analyzer component which will analyze the data and issue alert. This centralized approach, used in the most known products such as Snort [4] has several flows:

- In case of a failure of a sensor there is no handover,
- This type of IDS is very sensitive to Denial of service attack [5],
- Unstable reaction to distributed attacks,
- Sensors capacity relays on computer hardware, which makes the capacity hard to extend,
- Need to update all the sniffer separately,
- Need of human expertise during all the working time,

To eliminate such defects a lot of approaches has been applied to the detection process such as Artificial Neural Networks [6], genetic algorithms [7] and agent approach [8]. Developing IDS implies taking into account contemporary computer distributed environment and distributed nature of attacks. For these reasons agents approach seems to be more suitable. We advocate the idea that mobile agents framework enhance the performance of IDS and even offer them new capabilities. Moreover agent systems are used in various applications such as workflow, scheduling and optimization [10]. Agents are defined as a distinct software process, which can reason independently, and can react to change induced upon it by other agents and its environment, and is able to cooperate with other agents [9].

In an agent based IDS there is no central station, therefore no central point of failure. Overcoming the deficiency of centralized structure is the major reason for using agents in the intrusions detection field. The agents usefulness includes also reduction of the network load, overcoming of network latency and support for disconnected operations [11].

IDS can be classified into two categories, according to the approach used in analyzing network events: those based on anomaly approach [2], and, those based on misuse approach [12]. Both anomaly and misuse approaches present advantages and disadvantages. An IDS based on misuse approach can detect only those attacks that have been defined. Anomaly approach enable us to detect attacks that are unknown in advance; this advantage causes a large number of false positives (false alarm) that may occur when an IDS alerts an event that is not an intrusion [3]. Commercial IDS products such as NetRanger [17] and RealSecure [18] work on misuse approach. In this paper and based on the MA_IDS global structure we suggest to use an Artificial Neural Networks

doi:10.4304/jsw.7.1.156-160

as a decisional process implemented by the detection (analyzer) agent. By such decision process we will be able to enhance the detection process because of :

- Learning step: Because of the ANN structure and its working shema, we will be able to customize the suggested solution from one environment to another by only adapting the learning sets. In fact, the learning allows us to move from a static attacks description (usually provided through signature) to a more general and dynamic description that may include environment specificities and particularities,
- Unexpected configuration: ANN are known to be able to adapt to new situation. In fact, if we consider a signature description, if one event is not detected the global detection process is compromised. ANN will affect partially the final decision by just decreasing the detection rate or the ANN output value. Notice that the ANN output is in general not a binary decision but a kind of fuzzy value (rate) of correctness. To illustrate this aspect let's consider a signature described by $A, B, C \Rightarrow D$ that means the attacks D is occurring if events A, B and C has been observed. With a classical detection approach if the event C for example is not captured by the collectors agents, the intrusion will not be detected. With ANN the output would be something like D is true with a rate of 0.75 for example. Defining those rates and the correct expected output value is done through first an initialisation step based on expert judgment and evaluation (fuzzification step). The initialization set will be tuned through the learning step.

II. Anomaly Approach Background

Anomaly detection has recently gained a lot of attention in many security domains. Researchers have approached this problem using various techniques such as artificial intelligence, machine learning, and state machine modeling [13]. The birth of this subject is attributed to Jim Anderson who takes the attitude that something that is abnormal is probably suspicious in the early 1980s. His purpose was to improve the computer security auditing and surveillance capability of the customer's systems. He defined a baseline behavior for normal usage of computer system and then compares new usage of that system to the baseline level in order to check whether or not the new usage deviated from the normal one. Deviation will be considered as an anomalous activity.

In 1987, Denning [2] has also described building a behavior profile of normal usage over an interval of time. An abnormal activity is then considered as any deviation from the established behavior profile. The difficult questions here are:

- How does it determine what is normal?
- When we declare without mistakes that any deviation constitutes an intrusion and not an unusual authorized activity?
- How do we distinguish between anomalies and normal behavior in noisy, high-dimensional data?

There are various anomaly detection algorithms proposed in the literature. These algorithms differ according to the information used for analysis and according to techniques that are employed to detect deviations from normal behavior. Lazarevic et al. provide classification of anomaly detection techniques into the following five groups [15]:

- Statistical methods,
- Distance based methods,
- Rule based systems,
- Profiling methods,
- Model based approaches.

Anomaly based systems have the advantage of being able to detect previously unknown attacks but they suffer from the difficulty to build a solid model of acceptable behavior and the high number of alarms caused by unusual but authorized activities. IDS designers must find ways to speed up their attack analysis techniques when monitoring a fully-saturated network with less number of false positives. Statistical algorithms are not scalable and fast enough to keep up with the gigabit networks requirements of these days. Not fast enough because the statistical processing tend to be computationally expensive due to the fact that several metrics are often maintained, and need to be updated against every systems activity. Scalability is an issue since these systems depend on the network traffic behavior and we have networks today which have diverse and different requirements at times. Besides, one of the major problem with statistical methods is that not all abrupt changes in the network are anomalies where as it declares anomaly to any abrupt changes. It is also difficult to determine the right threshold above which an anomaly is to be considered intrusive. In statistical algorithms, a bigger sampling or threshold increases the chance of false negatives, while smaller values increase the chance of false positives. Basically, these traditional methods select key statistics about network traffic as features for a model trained to recognize normal activity. Unfortunately, statistics such as packet arrival times and connection arrival times have much variation. Too much statistical variation makes models inaccurate and events classified as anomalies may not always be malicious [14].

III. IMA_IDS Architecture

As mentioned in our introduction, we are suggesting adding an Artificial Neural Networks as a decision algorithm within an already existing Agent Intrusion detection systems. The solution has been introduced by Barika & al [22], implemented and enhanced with different detection approaches in [21] and [22]. Let us briefly recall the agent IDS architecture. The distributed structure of our system consists of four levels, as shown in figure 1: the down level, the pretreatment, the kernel and the upper level.

We have four cooperatives, communicants and collaborative entities which are able to move from one station to another: Sniffer agent, Filter agent, Analyzer agent and Decision agent. Every category of agent is assigned respectively to the levels cited previously.

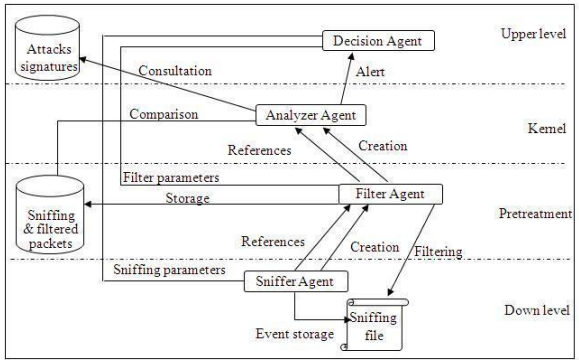

Figure 1. IMA_IDS Architecture

A. The Sniffer Agent

This kind of agent will be cloned and distributed throughout the network. This agent patrols the network, collects all the events occurred in the host to which it is related and storage the collected data in a sniffing file. The Sniffer agent can duplicate itself in order to lighten the network charge. On the down level, we are interested to collect all the events that occur through the network in real time. Sniffer is what is commonly called sensor [16].

B. The Filter Agent

Detecting intrusions in a distributed system turns out to be difficult. IDS must undertake to analyze huge volumes of events. This task becomes more difficult especially when the events must be collected from distributed sources around the network. Intrusions seek in all levels of the distributed system; each level may require monitoring. So, to be able to determine whether an intrusion is taking place, we have to aggregate and merge events collected from various sources, which is among the set of tasks allocate to the Filter agent. This agent performs its tasks in the context of the collected-events pretreatment phase, which precedes the analysis phase. The Filter agent plays the twofold role of preparing data to be analyzed, and of establishing a baseline of normal network behavior during the training period. In its first role, the Filter agent access to the sniffing file which is modified by the Sniffer agent and treats 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, IP...) concerned by a specific kind of intrusion.

C. The Analyzer Agent

This kind of agent processes and analyzes the events captured by the Sniffer agent and pre-processed by the Filter agents. Our proposal starts introduce upgrade at this level. Because of the use of ANN, this agent will now act as a classifier and pattern builder agent. We consider that attacks are divided into a set of groups according to either the concerned protocol (TCP, UDP) the observed events and events properties (IP values, PORT Values...) or the used vulnerability. This agent will in fact prepare the inputs for the decisional ANN by correctly formatting the collected elements and also describing the input/output pattern according to our ANN representation.

D. The Decision Agent

The administrator, depending on his need and requirement, can customize this agent via the training step.

In fact, in [19] Thom defines in deep the morphology for an ANN. We hold back the principal aspects. To highlight the analogy with our intrusions detection system, we demonstrate that all the keywords cited in the definition have a special reference respectively :

- A natural phenomenon: a network event,
- A case B: a segment of the network,
- A point x: a network event collected in a temporal unity,
- A regular point: a normal event, according to our definition of the normal behavior and its models,
- An open in B * T: the set of permitted behavior, considered as normal relatively in the time.

IV. ANN Architecture for Intrusion Detection Decision Making

Artificial Neural Networks have proved their abilities to correctly handle many difficult problems [23]. Many successes have been achieved with ANN in voice recognition, pattern recognition [25] and handwriting recognition. In our model, the agent gathers all the required data coming from the other agents and feed them to the ANN. The training phase consists of a supervised training, where we present to the ANN, a couple of input and desired output which consist respectively of event parameters and attacks.

A. The topology of the ANN

In our model, we choose to relay on a Multilayer Perceptrons (MLPs) ANN connected in a feed-forward way. Figure 2 illustrates the topology of artificial neural network used inside IDS model. The input layer size is depending on the collected events sent by the collector and filter agents. In figure 2 we limit (as an example) the input layers to 20 neurons. In practical implementation this size will be fixed by the network administrator according to the concerned attack category. In fact, it is commonly known that for example TCP/IP attack signatures uses no more than 10 or 15 events and so on. As known in regular ANN use [23], there is no deterministic way to decide about the number of neurons in hidden layers. After multiple tests, we find that having 3 hidden layers with respectively 5 neurons, 10 neurons and 15 neurons each, seems to be the best configuration for intrusion decision making process. Finally the output layer, that represents the set of possibly detect attacks is composed of as much neurons as the dressed (studied) attacks. In order to simplify representation, we just consider here

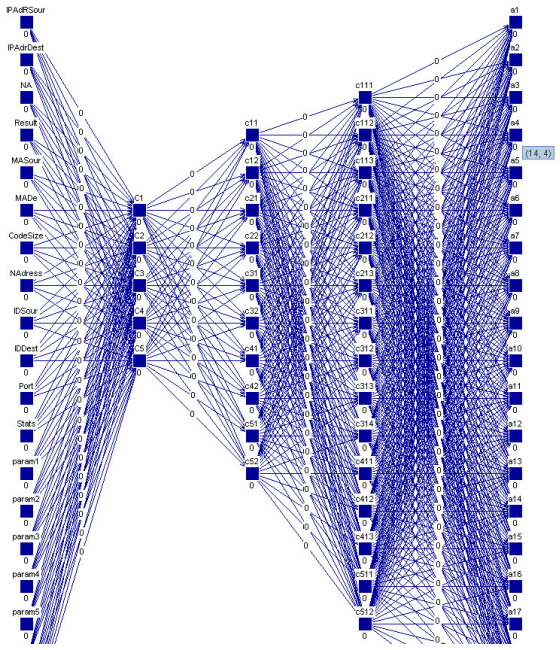

Figure 2. ANN for Intrusion detection global architecture

having 20 neurons, each one represents an attack. Let us now detail the hidden layer structure. As mentioned before, hidden layers contains respectively 5, 10 and 15 Neurons. The first hidden layer may contain more than 5 neurons. In fact this layer is for including a kind of attack classification in the attack decision making process. One neuron inside this layer is in fact one class attack category. For example we can consider that any spoofing attack is belonging to the category $C1$ in figure2 represented by the first neuron. Notice that any attack signature (of this class) may include one or more events represented by the following entries $IPAdRSour$, $IPAdrDest$, NA, $MaSour$, $MADE$, $CodeSize$. Please notice here that since the ANN is a Feed Forward one, we are focusing here only on significant entries that should have a weight higher than 0 after the training step.

B. ANN for intrusion detection

Our first prototype has been designed using Artificial Neural Nets Simulator JavaNNS [23]. In order to build the training set, we use as inputs the Snort attacks descriptions gathered by the collector agents and we convert them into the adequate input format as required by JavaNNS. The desired outputs are specified by the administrator for any collected event grouped in attack signatures. Notice that the output is nothing more than a possible attack with an estimated likelyhood produced by the ANN. In the training phase, the ANN will learn all the features from the training set by adjusting its weights. Once this phase is achieved, we experiment and test our ANN by linking it to an in use network. During this step, the ANN will use the real collected event after being classified and reformatted accordingly by the filter aget. The ANN will generate outputs corresponding to the possibly happening attacks described by the attack identification and an estimated likelyhood between 0 and 1.

V. Conclusion

In this paper we have shortly recalled intrusion detection systems principles and drawbacks. Then we have introduced the distributed intrusion detection system that we have suggested based on previous $MAFID$ architecture [21]. Based on the mentioned limitations and drawbacks in the decision step, we argue the use of Artificial Neural Networks as an alternative solution. We have presented the detailed architecture of our ANN and we also use the multilayer Perceptron paradigm. Then we have presented our ANN topology that has been simulated using JavaNNS [24]. Actually we are working on Snort signature transcription to a binary representation that could be taking as input for our ANN. The idea is to manage generating a huge test set including snort events as inputs and the corresponding signatures as possible outputs.

References

[1] J. P. Anderson, Computer security threat monitoring and surveillance, *James P. Anderson Company*, (Fort Washington, Pennsylvania, 1980).

[2] D. E. Denning, An intrusion detection model, in *IEEE Transactions on software engeneering, SE-13:222232*, (1987).

[3] G. Vigna, S. Eckmann and R. Kemmerer, Attack Languages, in *Proc. of the IEEE Information Survivability Workshop, IEEE Computer Society Press*, (Boston, MA, USA, 2000).

[4] SNORT, http://www.snort.org/.

[5] S. Specht and R. Lee, Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures, in *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems*, (September 2004).

[6] L. Vokorokos, A. Balaz and M. Chovanec, Intrusion Detection System using self organizing map, in *Acta Electrotechnica et Informatica No. 1, Vol. 6*, (2006), (ISSN 1335-8243 Faculty of Electrical Engineering and Informatics, Technical University of Kosice).

[7] W. Li, Using Genetic Algorithm for network intrusion detection, in Proc. United States Department of Energy Cyber Security Group, (Training Conference, Kansas City, Kansas, May 24–27, 2004).

[8] K. Deeter, K. Singh, S. Wilson, L. Filipozzi and S. Vuong, APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection System, in *Mobility Aware Technologies and Applications. LNCS, vol. 3284*, (Springer, Heidelberg, 2004).

[9] V. Honavar, L. Miller and J. S. K. Wong, Distributed knowledge networks, in *Proceedings, IEEE Information Technology Conference, Syracuse*, (NY, USA, September 1998).

[10] K. Ghedira, MASC : une approche Multi-Agents de problémes de Statisfaction de Contraintes, (1993).

[11] D. B. Lange and M. Oshima, Seven Good Reasons for Mobile Agents, in *Communications of the ACM, 42(3):88*, (March 1999).

[12] S. Kumar, and E. Spafford, A Software Architecture to Support Misuse Intrusion Detection, *Department of Computer Sciences, Purdue University*, (March 1995).

[13] P. Lingxi, L. Tao, L. Xiaojie, C. Yuefeng, L. Caiming, L. Sunjun, An Immune System-Inspired Paradigm for Anomaly Detection, in *Journal of Computational and Theoretical Nanoscience, Volume 4, Numbers 7-8*, (November/December 2007) , pp. 1394-1398(5).
[14] K. Das, Protocol Anomaly Detection for Network-based Intrusion Detection, SANS Institute 2002.
[15] A. Lazarevic, V. Kumar and J. Srivastava, Intrusion Detection: A Survey, book chapter, Book Managing Cyber Threats, ISSN 0924-6703, 2006.
[16] A. Cardon, A distributed multiagent system for the self-evaluation of dialogs. Proceedings of the Joint JSAI 2001 Workshop on New Frontiers in Artificial Intelligence, Springer-Verlag, 43-50 (2001).
[17] CISCO, http://www.cisco.com. Accessed March 2008
[18] RealSecure, http://www.iss.net. Accessed March 2008.
[19] R. Thom, Stabilite structurelle et morphogenese, Inter Editions, Paris, 1972.
[20] F. Barika, N. El Kadhi, K. Ghedira, Intelligent and Mobile Agent for Intrusion Detection System, ICICT 03 Egypt, November, 2003.
[21] N.EL KADHI, R. LIPS, Using Correlation Engine and Mobile Agents for Intrusion Detection System: The 12 WSEAS Conference on communication, Heraklion Greece 2008.
[22] N. EL KADHI, F. BARIKA, K.GHEDIRA, Intelligent and Mobile agent for Intrusion Detection System IMA-IDS. ICICT Conference Cairo, September 2003.
[23] P. D. Wasserman. Neural Computing: Theory and Practice, Van Nostrand Reinhold, 1989.
[24] JavaNNS, `http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome_e.html`.
[25] K. Hadjar and R. Ingold, Logical Labeling of Arabic Newspapers using Artificial Neural Nets, ICDAR'05 Seoul (Korea) 2005, pp. 426-430.