Journal of Engineering and Applied Sciences 6 (1): 79-90, 2011
ISSN: 1816-949X

# Mobile Agents for Intrusion Detection System Based on A New Anomaly Approach

[1]Farah Barika Ktata, [2]Nabil El Kadhi and [2]Khaled Ghedira
[1]Higher Institute of Management, LI3 Tunis, Tunisia
[2]Department of ECCE, University of Ahlia, Bahrein, France

**Abstract:** The aim of this study is to present the performance of an agent approach for intelligent and distributed intrusion detection system based on a new anomaly detection. The performance is investigated in terms of detection delay, false alarm rate and detection rate by comparing the presented two versions MAFIDS_v1 (Mobile Agents for Intrusion Detection System) and MAFIDS_v2, respectively based on a basic statistical anomaly detection algorithm (an adaptive threshold algorithm) and a modified adaptive threshold algorithm. This novel framework incorporates parameters issued from the investigation of 2 notions: morphology and artificial emotion. The underlying idea is to describe state of agent organization by various measurements made at the agent level. A particular emphasis is on the incorporation of these measurements to the anomaly detection algorithm for detecting SYN flooding, the most common type of Denial of Service (DOS) attack and improve its performance over uctuations of real TCP traffic especially when the major shortcomings of anomaly detection are: a longer detection and higher false alarm rate.

**Key words:** Agent approach, distributed intrusion detection system, anomaly, detection, morphology, artificial emotion

## INTRODUCTION

Security threats for computer network have increased significantly. Among all security issues, intrusion is the most critical and widespread. Intrusion can be defined as any illegal action a user takes toward an information system to compromise or harm a network. Intrusion detection appeared in 1980 as a process of detecting and tracing inappropriate, incorrect or anomalous activity targeted at computing and networking resources. Abstract intrusion detection model was proposed in (Denning, 1987). Intrusion Detection System (IDS) is a software that automates the intrusion detection process and detects eventual intrusions. IDS are usually divided into 2 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 HIDS works above the network layer making it unable to detect some kind of attacks (Ranum, 2001) while NIDS infer their decision from low-level network packets traveling among hosts. Detecting unknown intrusions in network traffic can be very complicated whereas on a host there are more things to be monitored at such as processes, network accesses, system calls, etc.

Moreover, visibility/attack resistance tradeoff exists between NIDS and HIDS. Visibility makes evasion more difficult by increasing the range of analyzable events and decreasing the risk of having an incorrect view of system state. On one hand, HIDS provides good visibility (Ando *et al*., 2007). However, increasing the visibility of the target system to the IDS very often weakens the isolation between the attacker and the IDS increasing thus risks of direct attacks, i.e., on IDS. On the other hand, NIDS offer higher attack resistance at the cost of visibility.

The usual approach of an IDS is to set up sensors to collect data that will be analyzed by an analyzer component which will issue alerts. This centralized approach used in the most known products such as Snort 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

**Corresponding Author:** Farah Barika Ktata, Higher Institute of Management, LI3 Tunis, Tunisia

attack (Specht, 2004). Many IDS have hierarchical structures, giving the attackers the opportunity to the attackers to harm the IDS by cutting off a control branch or even tacking out the root command:

- Unstable reaction to distributed attacks
- Sensors capacity relies on computer hardware which makes the capacity hard to extend
- Security of each sensor has to be granted separately, there is no global security to ensure that each sensor is either corrupted or authorized on the network
- You need to update all the sniffer separately
- Need of human expertise during all the working time
- When an IDS faces to a huge number of events in the network, it may drop network packets that does not have time to process or it may cause an overall system slow down

To eliminate such defects new approaches were applied to the detection process such as neural network (Vokorokos *et al*., 2006), genetic algorithms (Li, 2004) and agent approach (Deeter *et al*., 2004). When developing IDS it is also necessary to take into account contemporary computer distributed environment and distributed nature of attacks. For these reasons agents approach is more preferred for creating the security systems. We advocate the idea that mobile agents framework enhance the performance of IDS and even offerthem new capabilities. Moreover, agent systems are used in various applications such as work flow, scheduling and optimization. Agents is defined as a distinct software process which can reason independently can react to changes induced upon it by other agents and its environment and is able to cooperate with other agents (Honavar *et al*., 1998). Agents perform different tasks and are autonomous, i.e., can act independently from other agents.

They are also robust and fault-tolerant to changing environments. Agents can be mobile migrating from an agent place to another in order to perform the work locally. The main idea of an agent based IDS is that 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 (Lange and Oshima, 1999). The research procces covers two topics: The functional structure of IDS and analysis methods, particularly anomaly detection. In fact, IDS can be classified into two categories according to the approach used in analyzing network events: those based on anomaly approach and those based on misuse approach.

**Anomaly approach:** It relies on models of the normal behavior of a computer system (Denning, 1987). Behavior profiles may be focused on the users, the applications or the network. In this approach to detect abnormal activity patterns, the predefined profile patterns are compared with the actual ones in use. The detected patterns will be considered as intrusions.

**Misuse approach:** Relies on a set of attack descriptions also called attack signatures (Kumar and Spafford, 1995). These descriptions are matched to the stream of audit data, attempting to verify that the defined signature is occurring. 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 able us to detect attacks that are unknown in advance; this advantage causes a large number of false positives (false alarm) occurred when an IDS alerts an event that is not an intrusion (Vigna *et al*., 2000). Commercial IDS products such as NetRanger and RealSecure researchers on misuse approach.

An ideal IDS offers a high attack detection rate, low detection delay and low false positive rate but in practice this is hard to achieve. Detection rate is computed as the ratio of the number of correctly detected attacks to the total number of attacks while false alarm rate is computed as the ratio of the number of normal connections (that is incorrectly misclassified as attacks) to the total number of normal connections. The researchers are also interested in detection delay, the time delay between the occurrence and detection of a change of the network which is an important metric for measuring the system responsiveness. The contribution are twofold. Firstly, we propose an implementation of a new Mobile agent for IDS model based on anomaly detection (MAFIDS_v1) we present its apabilities to detect SYN ooding attack using an adaptive threshold algorithm. Secondly, we propose an improvement of this algorithm over uctuations of real TCP traffic (MAFIDS_v2). We investigate two new concepts in the intrusion detection field namely morphology and artificial emotions. We demonstrate their usefulness theoret-ically. From the investigations we make changes to the basic anomaly detection algorithm. The implementation and test of MAFIDS_v2 reach the expected experimental results in terms of detection delay, false alarm rate and detection rate.

**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 modelling (Lingxi *et al*., 2007). The birth of this subject is attributed to James (1980) who considered that something that is abnormal is probably suspicious. The researchers purpose was to improve the computer security auditing and surveillance capabilities of the customer's systems. The researchers defined a baseline behavior for normal usage of computer system and then compare 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.

Denning (1987) also has described building a behavior profile of normal usage over an interval of time. An anomalous activity is then considered as any deviation from the established behavior profile. The questions are:

- How to define what is normal when can we declare without mistakes that any deviation constitutes an intrusion and not an unusual authorized activity?
- How to distinguish between anomalies and normal behaviors 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 the employed techniques to detect deviations from normal behavior. Lazarevic *et al.* (2003) provide classification of anomaly detection techniques into the following 5 groups:

**Statistical methods:** Various statistical variables are used to characterize the traffic behavior in any network. Statistical methods monitor the user or system behavior by measuring these variables over time (e.g., CPU usage, number of files accessed, number of login attempts, etc.). The basic models measure the means and the variances of some variables and detect whether certain thresholds are exceeded based on the standard deviation of the variable. Sophisticated statistical algorithm are developed to measure the similarity between short-term and long-term profiles.

**Distance based methods:** Require no assumptions of distribution but is based on the distances between the data samples. Data is represented as a vector of features. Having a distance measure to evaluate proximity between two points they detect outliers by computing full dimensional distance among these points. Anomaly is defined as a data point that lies far away from other data points in the feature space.

Several distance based outlier detection algorithms have been recently proposed for detecting anomalies in network traffic and overcoming limitations of statistical outlier detection.

**Rule based systems:** A set of rules is used to characterize normal behavior of users, networks and/or computer systems. Each rule is mapped to a specific operation in the system. The rules serve as operational preconditions which are continuously checked in the audit record by the intrusion detection mechanism. If the required conditions of a rule are satisfied by user activity the specified operation is authorized.

**Profiling methods:** Profiles of normal behavior are built for different types of network traffic, users, programs, etc., using different data mining techniques (Lazarevic *et al.*, 2003) or heuristic-based approaches (Mariani and Pastore, 2008) and deviations from these are considered as intru-sions.

**Model based approaches:** Many researchers have used different types of models (e.g., replicator neural networks (Hawkins *et al.*, 2002)) or unsupervised support vector machines (Scholkopf *et al.*, 2000) to characterize the normal behavior of the monitored system. In the model based approaches, outliers are those data points that do not fit the model well. We are interested on statistical methods. Especially those which can be used to detect SYN flooding attack namely Adaptive threshold (Siris and Papagalou, 2004) and CUSUM (Wang *et al.*, 2004). The simplicity of these algorithms lies in their stateless and low computation overhead.

The protocol behavior of TCP SYN-FIN (RST) pairs constitute the baseline of their detection mechanism. The adaptive threshold algorithm compares the number of SYN packets received over a given interval to the estimated number based on recent measurements. If the threshold is exceeded consecutively then an alarm is raised. The CUSUM algorithm uses the difference between the number of SYN packets in a time interval and the estimated number for the same interval as a Gaussian random variable. CUSUM relies on the fact that if a change occurs, the probability distribution of the random sequence will change.

We focus on the adaptive threshold algorithm according to Siris and Papagalou (2004) who presented and evaluated these two algorithms in terms of detection probability, false alarm ratio and detection delay. Their results show that although simple and straightforward algorithms such as the adaptive threshold algorithm can exhibit good performance for high intensity attacks (i.e., the mean amplitude of the attacks is 250% higher than the mean traffic rate). Nevertheless, the detection delay is longer than the CUSUM algorithm. Besides, the performance of the adaptive threshold algorithm deteriorates in the case of low intensity attacks.

**Anomaly detection research:** Since, Anderson's study various anomaly detection approaches have been implemented by establishing statistical models for user (Javitz and Valdes, 1994), program (Forrest *et al*., 1996) or network behavior (Lee *et al*., 1999). Recently, the researchers (Taylor and Jim, 2001) use statistical clustering techniques to learn normal behavior patterns in network data. The delivered system was able to detect most network probes and DOS attacks in the MIT Lincoln Labs data. While others (Eskin *et al*., 2002) proposed unsupervised anomaly detection algorithms with unlabeled data based on the assumption that number of normal instances is significantly larger than the number of anomalies and anomalies appear as outliers in the data. During the same year, a new approach called specification-based anomaly detection was proposed (Sekar *et al*., 2002). Its purpose was to combine the primary benefits of anomaly detection and specification-based detection, namely, good detection of new attacks and low false alarm rates.

Puttini *et al*. (2003) have presented a new anomaly IDS design using a parametric mixture model for behavior modeling and Bayesian based detection. The proposed algorithms for detection and update phases present real-time feasibility with no special hardware requirement but the parametric Gaussian model that has been used for evaluation has some limitations. An intrusion detection model based on vector quantization technique was proposed by Sun *et al*. (2004). This model is suitable for security monitoring in the grid computing environment. Experimental results based on this model have shown very promising performance in terms of high detection rate and low false alarm rate.

For the same purpose, Gao *et al*. (2006) demonstrate that with the addition of labelled examples, the anomaly detection algorithm can guided to develop better models of the normal and abnormal behavior of the data.

Propose fusions of genetic algorithm and support vector machines for efficient optimization of both features and parameters for detection models (Kim *et al*., 2005). Their method provides optimal anomaly detection model which is capable to minimize amounts of features and maximize the detection rates. A statistical approach to anomaly detection in interdomain routing was proposed (Deshpande *et al*., 2006). A time-series segmentation algorithm is used to detect instabilities triggered by events like router misconfigurations, infrastructure failures and worm attacks. The performance of the proposed algorithm is evaluated using real Internet trace data with false alarm rate as low as 0.0083 alarms $h^{-1}$. But this algorithm causes an average detection delay of approximately 50 min.

Durgin and Zhang (2005) investigate profile-based anomaly detection techniques that can be used to address the problem of recognizing and evaluating threats against networks which is complex and heterogeneous. They identify and evaluate promising techniques for data mining and machine-learning. They built a prototype anomaly detection tool that demonstrates how the techniques might be integrated into an operational intrusion detection framework.

But they have not tested datasets that contains real intrusions or used it to identify abnormal behaviors that pose a threat to the network security. Salem *et al*. (2007) provided a new framework for efficient detection and identification of network anomalies over high speed links. They applied the CUSUM algorithm and proved the capacity of early detection even for low intensity of DoS/DDoS (distributed DOS) attacks. It resulted from this framework a high sized exchanged sketch information between different monitoring nodes in different layers. Researchers also proposed as a future work to distribute hierarchically the proposed approach.

More recent work focus on the choice of features used to describe normal or intrusive traffic patterns. Tran *et al*. (2008) propose an automated feature weighting method based on a fuzzy subspace approach to vector quantization modelling that can assign a weight to each feature when network models are trained. Their method increase the detection rate and reduce false alarm rate. The idea of Shanbhag and Wolf (2008) is similar to the previous research such as the used algorithms and the detected attacks.

The researchers propose running several different anomaly detection algorithms in parallel on 1000 of different traffic subclasses. Aggregated anomaly detection results show a lower false negative (occurs when the IDS fails to detect malicious network activity) and false positive rate than any single anomaly detection algorithm.

## RESULTS AND DISCUSSION

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 researched activities. Statistical analysis are not able to detect attacks scenarios which may occur over an extended period of time. For example, an exploit using a missing command in a session can only be identified when a session is completed and will necessitate keeping track of state and context (Gong, 2003). This could affect the time performance of the IDS which corresponds to the total time needed to detect an intrusion. Times need to be as short as possible in order to allow the security analyst sufficient time to react to an attack before much damage has been done as well as to stop an attacker from modifying audit information or altering the IDS itself (Kumar *et al*., 2005).

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 system's activity. Scalability is a concern since, these systems depend on the network traffic behavior and we have networks today which have various and different requirements at times. Besides, one major problem of statistical methods is that not all abrupt changes in the network are anomalies whereas 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 so that events classified as anomalies may not always be malicious (Das, 2002).

Moreover, statistical analysis may present another disadvantage. In fact, statistical measures capturing user behavior could be gradually trained to a point where intrusive behavior could be considered normal.

**Adaptive threshold algorithm:** This algorithm relies on testing whether the traffic measurement exceeds a particular threshold (Siris and Papagalou, 2004). In case of detecting SYN flooding, this algorithm measures whether the number of SYN packets exceeds a particular threshold over a given time period. In order to account timely variations, the value of threshold is set adaptively based on an estimate of mean number of SYN packets. If $x_n$ is the number of SYN packets in the nth time interval and $\mu_{n-1}$ is the mean rate estimated from measurements prior to n, then the alarm condition is $x_n$ $(\alpha+1)$ $\mu_{n-1}$ then alarm signalled at time n where, $\alpha>0$ is a parameter indicating the percentage above the mean value that we consider to be an indication of anomalous behavior. The mean μn can be computed from some previous time frame or using an Exponential Weighted Moving Average (EWMA) of previous measurements:

$$\mu_n = \beta\mu_{n-1}+(1 - \beta)_{xn}$$

where, β is the EWMA factor. Direct application of the above algorithm would yield a high number of false alarms. In order to avoid this inconvenience, a simple modification is to signal an alarm after a minimum number of consecutive violations of the threshold. In this case, the alarm condition is given by if:

$$\sum\nolimits_{i=n-k+1}^{n} 1\{x_i \geq (\alpha+1)\mu_{n-1}\} \geq k$$

then alarm at time n where, $k>1$ is a parameter that indicates the number of consecutive intervals the threshold must be violated for an alarm to be raised. The tuning parameters of the above algorithm are the amplitude factor  for computing the alarm threshold, the number of successive threshold violations k before signaling an alarm, the EWMA factor  and the length of time interval over which measurements (number of SYN packets) are drawn. Panoptis (Spinellis and Gritzalis, 2002) is an example of a system that applies a simple threshold based anomaly detection algorithms to the number of bytes, packets and flows observed in time intervals of constant length in order to detect DoS and DDoS attacks. The system Open-Eye (Androulidakis *et al*., 2004) follows the same concept but applies the threshold algorithm to different metrics. Both Panoptis and Open-Eye do not provide the possibility to run multiple detection modules simultaneously.

**Proposed approach:** We have advocated the idea that in the security domain, especially when we are faced to the contemporary computer distributed environment, a mobile agent's framework enhances the performance of IDS and even offer them new capabilities. We have argued this challenge (Barika *et al*., 2003). Nevertheless in the intrusions detection domain, systems faced many facts:

- The fuzzy, unexpected and uncertain hazards
- A dynamic and a continual evolution of the environment where they perform there tasks

Besides this facts, the intrusions must be detectedas soon as possible to allow efficient and immediate counter measures to be accomplished. Hence, it becomes a contradiction in these conditions to entirely, aim at a usual Multi-agent system which has a structure, a behavior and a goal fixed in advance. The goal is to reach to endow IDS with the capacity to infer a decision or result which is not strictly and purely formal calculated

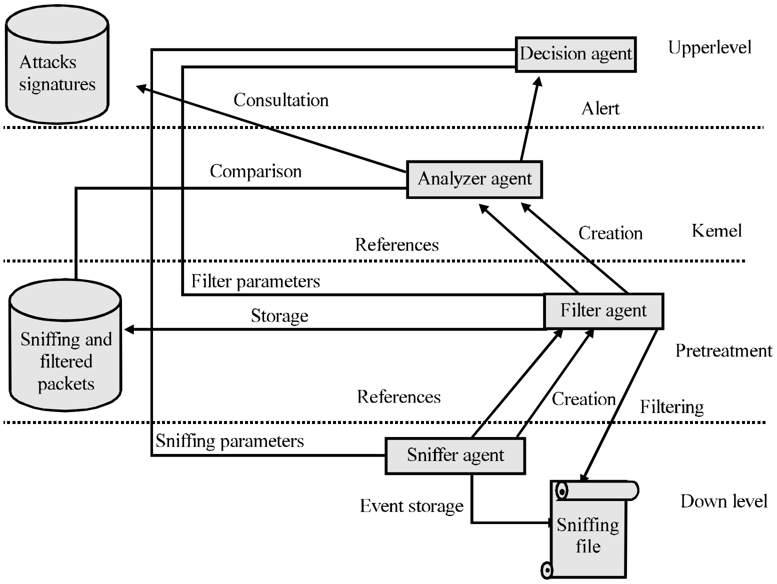


Fig. 1: MAFIDS Architecture

from knowledge and rules formally defined. In other words, we aim to make the system able to learn from the previous experience to take advantage of it to use the indicators of the proprieties of the environment which we control leading to describe the global form of the system in order to make in the best way as possible, a true representation of the actual scene (the environmental neighborhood, the system proprieties).

For this purpose, we argue and demonstrate the utilities of the morphological aspect and the artificial emotions by introducing new metrics to make the system self adaptive and its decision more efficient.

**The system: architecture overview:** The distributed structure of the system consists of four levels as shown in Fig. 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.

**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 stores the collected data in a sniffing file. The Sniffer agent can duplicate it self in order to lighten the network charge. On the down level, we are interested to collect all the events occurred through the network in real time. Sniffer are what is commonly called sensor (Cardon, 2001).

**The Filter agent:** Detecting intrusions in a distributed system turns out to be difficult. IDS must under take to analyze a huge volumes of events. This task becomes more difficult especially when the events must be collected from distributed sources around the network. Intrusions seep 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

Of equal importance to its first role is the Filter's responsibility to establish a traffic baseline under normal (non-flood) network operating conditions. Normal operating conditions are defined as average traffic and application flow crossing the network edge devices averaged over time while the network is not under attack. The basic idea is to compute statistical values of relevant features which will identify SYN flood attack (e.g., protocol distribution). The training period lasting 2 weeks during which the Filter agent measures TCP packet count for every hour interval of 6 work days week$^{-1}$. Mean values and adaptive thresholds are calculated and stored in a local data structure for every interval.

**The Analyzer agent:** This kind of agent processes and analyzes the events captured by the Sniffer agent and pre-processed by the Filter agents. We adopt the threshold algorithm to detect the SYN flooding attack. While the Filter agent performs its major tasks in the training period, the Analyzer agent operates in the detection period. It constructs current traffic profiles. At each given period the Analyzer agent calculates and stores the deviation between the normal and current value (number of SYN packets). It sends an alarm after a minimum number of consecutive violations of the threshold.

**The Decision agent:** The administrator depending on his needs and requirements can give some parameters relative to the full detection process. These parameters are saved on a configuration file which is consulted by the Decision agent in order to sort them by kind of treatment. In fact, we consider sniffing parameters such as the address of the monitoring hosts and filter parameters like the target protocol. Furthermore, the Analyzer agent report their findings to the Decision agent which transmits them to the administrator.

**The morphologic aspect:** The morphology is defined as scientific study of the links between the shape of an object and its environment. To enrich this definition we focus on the research of Thom who points out that an object is always subject to disruptive in uence from its environment which has an effect on his shape, even they are weak. We are interested in the morphology in the context of the objective to be able to make a precise, exact and correct response to this question: when we can consider the system unstable. In fact, Thom defines in detail the morphology. We hold back the principal aspects. To highlight the analogy with the 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 the 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

Moreover, the point x is considered irregular when according to its neighborhood, it deviates from a regular scheme (normal behavior) already defined. In the contrary case, the point x is considered as a regular event. Thus, we recognize the system unstable by occurring un irregular event.

**The usefulness of the morphology:** The research is applying the morphology as the study of the shape and the behavior of the system in order to emphasize the result of the anomaly approach. When we plan to control a system, starting from the observation phase then the analysis phase until making the decision which is relative to its state, the study the shape of this system turn out to be an important phase. According to the morphology allows to highlight some proprieties of the system which are not perceptible at first sight. Furthermore, the fact of consider the shape of the system provide us a global vision of its state and its behavior which allows us to argue and enrich the singular observed proprieties and events. Moreover, the concept of instability can be directly detected starting from the study of the shape of the system which will help us to determine its vulnerabilities aspects. The morphology with the global vision which it offers, permits to mitigate the difficulties to analyze and to control the system which is heterogenous according to the principal characteristic of the network. The artificial emotion recently, some researcher in the community of artificial intelligence look into the concept of emotion which belongs to neurobiology, especially Minsky (2006) who leads to results defending this hypothesis: in order to make the artificial entities more autonomous and self-adaptive, it will be necessary to provide them with emotions.

The emotion associates a physical and a mental representations. Moreover, it can be consider as a cognitive evaluation of stimulations or situations. The emotion also represents a behavior preparation and a relation between situation and reaction. Nevertheless, concerning the research we do not consider the emotion as troubles caused by the fear or happiness or other feeling but as:

- A systemic reaction to a stimulus
- A calculable and conceptual entity
- An organizational movement in a dynamic and complex system

- The quality of some re-organizational movement of the entities in a system

The emotion has a dynamic process which can be represented by a curve of effect in the space (Minsky, 2006). Moreover, we have a set of descriptive parameters relative to emotion particularly its duration, its intensity, its frequency, its progression and its regularity. The artificial emotion in the detection process traditionally, computer science treats automatically the information in order to lead to calculated results which are useful for performance goal.

Often the automatic treatment process has less importance than the result. In the context of the intrusions detection, it is important to take an interest in the result and the progress of the detection phases too.

Many experiences have proved the fact that a detection exclusively automatic and formally inferred depending to a preconceived knowledge base is not always exact provoking false positive and false negative. An IDS performs in an opened environment which is uctuate, uncertain and undetermined. In this case, it becomes essential to provide the system with the capacity of selfadaption. According to Campagne and Cardon (2003) and Cardon *et al*. (2005), the emotion belongs to the set of the fundamental mechanisms for a system which aims at autonomy and self-adaption. Indeed in the intrusion detection domain, the major challenge is to be able to face the new situations (new attacks, changes in the normal behavior of users) which frequently occurred. This reality makes us convinced that an optimized knowledge based reasoning (models for normal behavior) do not cover this unexpected and sudden situations. Furthermore, an emotional reasoning, precisely intentional has to bring a decisional advantage to the system and permit its backup by the generation inspired actions plans which are continually adapted and quickly defined without care of the best calculated choice.

**Modified adaptive threshold algorithm:** We propose new metrics issued from the investigation of morphology and artificial emotions. We also take into account new features in the detection process in order to improve its precision, i.e., its ability to correctly detect intrusion in a short time.

**Event correlation engine:** The goal is to enrich the pretreatment phase by adding new module checking for suspicious events. For us a suspicious event is any event liable to be part of an attack signature. Due to the widespread prevalence of Snort, its signatures comprise the most comprehensive signature set that is openly available. Consequently, Snort which is one of the most popular open source security tools (Timofte, 2008), serves us as a reference. We parse the SNORT signatures database for DOS attacks. We pick out the most common attributes in signatures which are Source Port Fields (SPF), Destination Port Fields (DPF) and Packet Data Payload (PDP). We construct correlation rules (such as the example below) composed from all possible combinations of the values of the picked attributes. There is an additional field in the TCP packet that is the result of the module checking; Priority. Priority has a binary value. The Filter agent affects the value 1 if it verifies at least one of the correlation rules, 0 otherwise. IF SPF = 12754 and DPF = 139 and PDP = FF FF FF FF FF FF then Priority = 1 (Table 1).

Table 1: Common attributes values in TCP packets

| Port types | TCP |
|---|---|
| Source port | 20432, 12754, 15104 |
| Destination port | 27665, 12754, 7070, 8080, 135, 139, 3372, 6004, 6789, 6790, 80, 179, 515, 646,21513, 3128, 9191, 443, 3101, 25 |
| Packet data payload | FF F4 FF FD 06, FF FF FF FF FF FF, 00 03 00 00 00, 05 00 00 03 10 00 00 00, 00 00 00 00, 01 06 00 00 00, FF FF FF, 00 00, 3A, 13, 0A, 00 |

**Agents synergy:** The aim is to reach a global state vision of the agent system by favoring agents synergy in order to emphasize the result of anomaly detection by monitoring the agents own progress and the whole system. The underlying idea is that intrusion affects both system and agent behaviors especially in the case of DOS attack. We advocate the fact that to detect failures, an agent must have information about the whole agent system behavior. Given that every agent in our MAFIDS has in his knowledge database, a set of metrics that indicate the ideal state of the system (e.g., maximum number of cloned agents, average of agent response time), it compares these metrics to the agents actual behavior to detect discrepancies indicating possible failure. We define a set of messages and metrics that illustrate the agent's synergy and describe their state. The Filter agent can send the following urgent messages to the Analyzer agent:

- Filter syntactic abnormal event when the Filter agent can not identify the different fields of packet (such as IP address and port). The exact number and nature of the fields depends on the type of the event
- Filter semantic abnormal event when the Filter agent finds abnormalities in the packet field value such as unusual long or short field lengths which can indicate an attacker is attempting to introduce a buffer over flow

- Filter suspicious event when the Filter agent put the value 1 to the priority field of the event
- Filter count abnormal X event when the Filter agent find an unusual number of occurrences of particular event
- Filter pb access resources when the Filter agent can not access to the sniffing file

When receiving one of these urgent messages, the Analyzer agent increments its Counter of Urgent Notifications (CUN). We also define the following metrics which will be considered by the Analyzer agent in its anomaly detection algorithm:

**Latency Time of Response (LTR):** Periodically, the Analyzer agent send messages to call others agents. Given the total number of running agents, the Analyzer agent can deduce the number of agents which do not respond.

This metric can indicate a critical overload of an agent or an unexpected agent crash which could be symptom of DOS attack. The Analyzer agent measures the total latency time of response which will be multiplied by the number of agent with no response given LTR as a result. During its detection process the Analyzer agent will compare this metric with the average latency time stored during the training period.

**Number of Cloned Agent (NCA):** Given the normal traffic flow, Decision agent knows the maximum number of Cloned agent. During the training period, we store normal traffic flow which is the number of packets of a given protocol travelling between a source and a destination IP/port pair within a certain period of time. The Sniffer and the Filter agents are cloned depending on the size of the sniffing file in order to lighten the network charge. If the actual Number of Cloned Agent (NCA) exceeds the maximum number of cloned agent then the Decision agent sends anurgent message to notify the Analyzer agent. We change the alarm condition of the threshold algorithm taking into account the defined metrics as follow:

$$\left(\sum_{i=n-k+1}^{n} 1\{x_i \geq (\alpha+1)\mu_{n-1}\} \geq k\right)$$

and;

$$(\mathrm{LTR}_i > \mathrm{LTR}_{\mu_{n-1}})$$

or;

$$\left(\mathrm{NCA}_i > \mathrm{NCA}_{\mu_{n-1}}\right),(\mathrm{CUN} > 0)$$

then alarm at time n. The goal of these changes is to speed up the detection process and at the same time reduce the false positive rate. We evaluate the performance of the modified adaptive threshold algorithm by comparing it with the original one.

**Implementation and performance eveluation:** We implement two versions MAFIDS_v1 (Mobile Agents for Intrusion Detection System) and MAFIDS_v2 based on an adaptive threshold algorithm and a modified adaptive threshold algorithm, respectively. The parameters we consider for both these algo-rithms were $\alpha = 0.5$, $k = 4$ and $\beta = 0.9$. We use Sun's Java Development Kit version 1.4.1 the framework Aglets Workbench 2.0.2., the Netbeans 3.4. and the Jpcap 0.01.16.

All the experiments were conducted on equivalent machines equipped with a pen-tium dual core processor running at 1.66 GHz and 1.99 Gb of main memory. The system performs its tasks over any number of hosts in the network. Each host can receive any number of Sniffer agents that monitor all events occurring in it. In a first phase, we test the communication model by sending a set of messages between the four agents classes. We also test the mobility of these agents by dispatchingthem and retracting over three hosts.

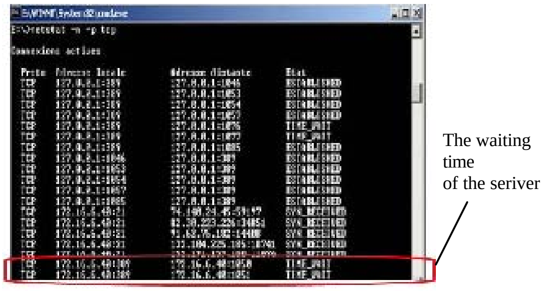
In a second phase, we run separately the two versions in order to learn the normal packet attribute values during the attack-free period (2 weeks) of inside training data which consist of 8,983,528 traffic packets in order to come up with the normal traffic profile based on distinct packet field values for each of the host in the network. These profiles then are classified by time of day, day of week.

We consider only working days (from 8:00 a.m.-6:00 p.m.). In a third phase, we run the two versions, in the same condition. We randomly inject SYN flood attack by using the HPING tool which is able to send custom TCP/IP packets to network hosts. All experiments run on three machines:

- 172.16.0.41: Attack host
- 172.16.6.220: Web client (IP address which we usurped)
- 72.16.6.40: Web server (the victim machine)

We usurp the IP address of the web client host and send a large number of SYN packets to the web server via this command: hping - S - i u10 - p 80 - a 172.16.6.220 172.16.6.40.

At the same time we disconnected the web client host, preventing thus, the machine from answering the packets sent by the web server. Otherwise, it would send TCP RST packets which would stop the connection attempt. The web server machine waits for confirmation



```
E:\>netstat -n -p tcp

Connexions actives

  Proto  Adresse locale         Adresse distante         Etat
  TCP    127.0.0.1:389          127.0.0.1:1046           ESTABLISHED
  TCP    127.0.0.1:389          127.0.0.1:1053           ESTABLISHED
  TCP    127.0.0.1:389          127.0.0.1:1054           ESTABLISHED
  TCP    127.0.0.1:389          127.0.0.1:1057           ESTABLISHED
  TCP    127.0.0.1:389          127.0.0.1:1076           TIME_WAIT
  TCP    127.0.0.1:389          127.0.0.1:1077           TIME_WAIT
  TCP    127.0.0.1:389          127.0.0.1:1085           ESTABLISHED
  TCP    127.0.0.1:1046         127.0.0.1:389            ESTABLISHED
  TCP    127.0.0.1:1053         127.0.0.1:389            ESTABLISHED
  TCP    127.0.0.1:1054         127.0.0.1:389            ESTABLISHED
  TCP    127.0.0.1:1057         127.0.0.1:389            ESTABLISHED
  TCP    127.0.0.1:1085         127.0.0.1:389            ESTABLISHED
  TCP    172.16.6.40:21         74.140.24.45:59197       SYN_RECEIVED
  TCP    172.16.6.40:21         82.30.223.226:34854      SYN_RECEIVED
  TCP    172.16.6.40:21         91.62.75.132:1440        SYN_RECEIVED
  TCP    172.16.6.40:21         133.104.225.185:10740    SYN_RECEIVED
  TCP    172.16.6.40:21         [illegible]              SYN_RECEIVED
  TCP    172.16.6.40:389        172.16.6.40:1050         TIME_WAIT
  TCP    172.16.6.40:389        172.16.6.40:1051         TIME_WAIT
```

Fig. 2: Result of the SYN flood attack

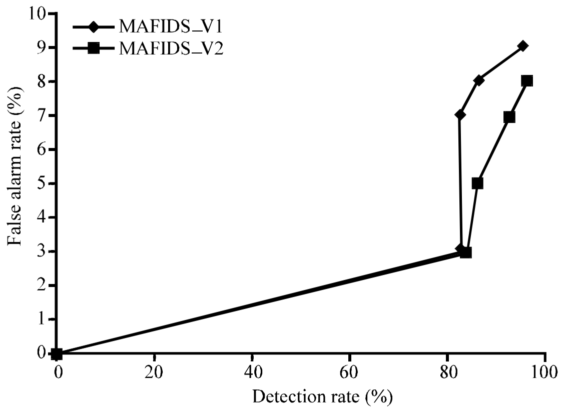


Fig. 3: ROC curves for the two system versions in the case of low intensity attacks

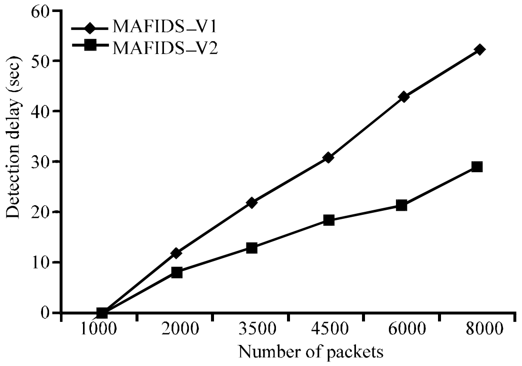


Fig. 4: MAFIDS_v1 vs. MAFIDS_v2 in terms of detection delay

that never arrives. Hence, the attack succeeds (Fig. 2). We evaluate the two system versions performance with Detection Delay (DD) detection False Positive Rate (FPR) and Detection Rate (DR). We are also interested in two cases: high intensity attack (with mean traffic load of 400 tcp packets sec$^{-1}$) and low intensity attack (with mean traffic load of 30 tcp packets sec$^{-1}$). Figure 3 shows the results in the case of low intensity attacks for MAFIDS_v1 and MAFIDS_v2. Performance is shown by plotting Receiver Operating Characteristic curves (ROC) which show the detection rate versus the false alarms rate produced by each system. MAFIDS_v2 demonstrates better performance in reducing false alarm rate and increasing detection rate. The average of false alarm rate is about 4.6 with 89% detection rate. MAFIDS_v2 exhibits improvements of 15 and 3.8%, for false alarm rate and detection rate, respectively compared to the results of MAFIDS_v1 (the average of false alarm rate is about 5.4 with 85.75% detection rate). Such superior performance of MAFIDS_v1 may be explained by the fact that We strengthened the adaptive threshold algorithm by more criteria (LTR, NCA) to generate the ALARM and in the same time we consider more effects which can indicate DOS attack (CUN). In the case of high intensity attacks, the two versions realize a detection rate of 100% and false alarm rate of 0%. As can be seen from the Fig. 4, MAFIDS_v2 has the best detection delay performance. We generate a set of packets varied from 1000-8000. For each set, we simulate the SYN flood attack and we calculate the detection delay. MAFIDS_v2 is much faster than the MAFIDS_v1. For example, in the case of 6000 packets, we observe that detection delay is reduced by 49% (43 vs. 21 sec). This can be explained by the fact that the Filter agent simplifies and facilitates tasks of the Analyzer agent especially when it includes the priority field. Besides, agents exchange exactly what they need as urgent messages, no more nor less given that sending too much messages between agents leads to network overload.

## CONCLUSION

Intrusion detection systems must handle masses of information (in real-time) so as to report the abnormal use of networks and computer systems. We are interested in anomaly detection methods which allow us to detect new types of attacks. But their major drawbacks are: a longer detection and higher false alarm rate. Mobile agents could offer a valuable addition to the intrusion detection field. We designed and developed MAFIDS_v1 (Mobile Agents for Intrusion Detection System) based on adaptive threshold algorithm.

Being convinced that anomaly detection is not always about detecting unexpected activities but also about detecting state changes, we defined new metrics issued from the investigation of two new notions: morphology and artificial emotion. We integrate these new metrics into the anomaly detection algorithm (modified adaptive threshold algorithm) and developed MAFIDS_v2. Experimental results demonstrate that

MAFIDS_v2 presents better performance in increasing detection rate, reducing false alarm rate and detection delay. For the future research, more research can be done testing the modified adaptive threshold algorithm against more attacks and exploring how mobility and self-clone ability would enhance the survivability of IDS.

## REFERENCES

Ando, R., Y. Kadobayashi and Y. Shinoda, 2007. Asynchronous pseudo physical memory snapshot and forensics on paravirtualized VMM using split kernel module. Lecture Notes Comput Sci., 4817: 131-143.

Androulidakis, G., V. Chatzigiannakis, M. Grammatikou and F. Stamatelopoulos, 2004. Network flow-based anomaly detection of DDoS attacks. Proceedings of Trans-European Research and Education Networking Association (TERENA) 2004, Rhodes, Greece, June 2004.

Barika, F., N. El-Kadhi and K. Ghedira, 2003. Intelligent and mobile agent for intrusion detection system. ICICT Conference, 03 Egypt, November, 2003.

Bordogna, J.T., D.E. Brown and J.H. Conklin, 2007. Design and implementation of an automated anomaly detection system for crime. Proceedings of the Systems and Information Engineering Design Symposium, April 27, Charlottesville, VA USA., pp: 1-6.

Campagne, J.C. and A. Cardon, 2003. Artificial emotions for robots using massive multi-agent systems. Proceedings of the Social Intelligence Design International Conference, London.

Cardon, A., 2001. A distributed multi-agent system for the self-evaluation of dialogs. New Front. Artif. Intell., 2253: 43-50.

Cardon, A., J.C. Campagne and M. Camus, 2005. A self-adapting system generating intentional behavior and emotions. Second GSFC/IEEE WRAC 2005: Workshop on Radical Agent Concept, NASA Goddard Space Flight Center.

Das, K., 2002. Protocol anomaly detection for network-based intrusion detection. SANS Institute, GSEC Practical Assignment Version 1.2f, http://www.sans.org/reading_room/whitepapers/detection/protocol-anomaly-detection-network-based-intrusion-detection_349.

Deeter, K., K. Singh, S. Wilson, L. Filipozzi and S. Vuong, 2004. APHIDS: A mobile agent-based programmable hybrid intrusion detection system. Mobility Aware Technol. Appl., 3284: 244-253.

Denning, D.E., 1987. An Intrusion-Detection Model. IEEE Trans. Software Eng., SE-13: 222-232.

Deshpande, S., M. Thottan, T.K. Ho and B. Sikdar, 2006. A statistical approach to anomaly detection in interdomain routing. Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems, Oct. 1-5, San Jose, CA. USA., pp: 1-10.

Durgin, N.A. and P.C. Zhang, 2005. Profile-based adaptive anomaly detection for network Security. Sandia National Laboratories Technical Report, SAND2005-7293, November 2005.

Eskin, E., A. Arnold, M. Prerau, L. Portnoy and S. Stolfo, 2002. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. In: Applications of Data Mining in Computer Security, Barbara, D. and S. Jajodia (Eds.). Kluwer Academic Publishers, Boston.

Forrest, S., S.A. Hofmeyr, A. Somayaji and T.A. Longstaff, 1996. A sense of self for Unix processes. Proceedings of the IEEE Symposium on Securiry and Privacy, May 6-8, Oakland, CA. USA., pp: 120-128.

Gao, J., H. Cheng and P.N. Tan, 2006. A novel framework for incorporating labeled examples into anomaly detection. Proceedings of the 6th SIAM International Conference on Data Mining, April 2006, Bethesda, MD, pp: 594-597.

Gong, F., 2003. Deciphering detection techniques: Part II anomaly-based intrusion detection. White Paper, McAfee Network Security Technologies Group, March 2003.

Honavar, V., L. Miller and J. Wong, 1998. Distributed knowledge networks. Proceedings of the IEEE Information Technology Conference, Sep. 1-3, Syracuse, NY USA., pp: 87-90.

James, P.A., 1980. Computer security threat monitoring and surveillance. Technical Report, James P. Anderson Company, Fort Washington, Pennsylvania. http://csrc.nist.gov/publications/history/ande80.pdf.

Javitz, H.S. and A. Valdes, 1994. The NIDES statistical component: Description and justifi-cation. Technical Report, Computer Science Laboratory, SRI International, Menlo Park, California.

Kim, D.S., H.N. Nguyen, S.Y. Ohn and J.S. Park, 2005. Fusions of GA and SVM for anomaly detection in intrusion detection system. Adv. Neural Networks ISNN., 3498: 415-420.

Kumar, S. and E. Spafford, 1995. A software architecture to support misuse intrusion detection. Technical Report, Department of Computer Sciences, Purdue University, (CSD-TR-95-009).

Kumar, V., J. Srivastava and A. Lazarevic, 2005. Managing Cyber Threats: Issues, Approaches and Challenges. 1st Edn., Springer, New York, pp: 330.

Kussul, N., A. Shelestov, A. Sidorenko, V. Pasechnik, S. Skakun, Y. Veremeyenko and N. Levchenko, 2003. Multi-agent security system based on neural network model of users behavior. Int. J. Inform. Theories Appl., 10: 184-188.

Lange, D.B. and M. Oshima, 1999. Seven good reasons for mobile agents. Commun. ACM., 42: 88-89.

Lazarevic, A., L. Ertoz, V. Kumar, A. Ozgur and J. Srivastava, 2003. A comparative study of anomaly detection schemes in network intrusion detection. Proceedings of the 3rd SIAM International Conference on Data Mining, http://www.citeulike.org/user/horeis/article/549060.

Lee, W., S.J. Stolfo and K. Mok, 1999. Data mining in work ow environments: Experiences in intrusion detection. Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD-99).

Li, W., 2004. Using genetic algorithm for network intrusion detection. Proceedings of the United States Department of Energy Cyber Security Group, Training Conference, May 24-27, Kansas City, Kansas.

Lingxi, P., L. Tao, L. Xiaojie, C. Yuefeng, L. Caiming and L. Sunjun, 2007. An immune system-inspired paradigm for anomaly detection. J. Comput. Theoret. Nanosci., 4: 1394-1398.

Mariani, L. and F. Pastore, 2008. Automated identification of failure causes in system Logs. Proceedings of the 19th International Symposium on Software Reliability Engineering, (ISSRE'08), Washington, DC USA., pp: 117-126.

Minsky, M., 2006. The Emotion Machine. Simon and Schuster, New York.

Puttini, R.S., Z. Marrakchi and L. Me, 2003. A bayesian classification model for real-time intrusion AIP Conf. Proc., 659: 150-162.

Ranum, M.J., 2001. Experiences benchmarking intrusion detection systems. NFR Security.

Salem, O., S. Vaton and A. Gravey, 2007. A novel approach for anomaly detection over high-speed. Proceedings of the EC2ND : European Conference on Computer Network Defense, Heraklion, Greece, October 2007.

Scholkopf, B., R. Williamson, A. Smola, J. Shawe-Taylor and J. Platt, 2000. Support vector method for novelty detection. Adv. Neural Inform. Proc. Syst., Vol. 12.

Sekar, R., A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang and S. Zhou, 2002. Specifi-cation based anomaly detection: A new approach for detecting network intrusions. Proceedings of the 9th ACM Conference on Computer and Communications Security, Nov. 18-22, Washington, DC, USA., pp: 265-274.

Shanbhag, S. and T. Wolf, 2008. Massively parallel anomaly detection in online network measurement. Proceedings of 17th IEEE International Conference on Computer Communications and Networks, Aug. 3-7, St. Thomas, US Virgin Islands, pp: 1-6.

Siris, V.A. and F. Papagalou, 2004. Application of anomaly detection algorithms for detecting SYN fooding attacks. Proceedings of the Global Communications Conference, Nov. 29-Dec. 3, Dallas, TX., pp: 2050-2054.

Specht, S.M., 2004. Electrical engineering, princeton university, ruby b lee, electrical engineering, princeton university, distributed denial of service: Taxonomies of attacks, tools and countermeasures. Proceedings of 17th International Conference on Parallel and Distributed Computing System, 2004, IEEE Xplore, pp: 543-550.

Spinellis, D. and D. Gritzalis, 2002. Panoptis: Intrusion detection using a domain-specific language. J. Comput. Security, 10: 159-176.

Sun, H.W., K.Y. Lam, S.L. Chung, M. Gu and J.G. Sun, 2004. Grid and cooperative computing. Proceedings of the 3rd International Conference Wuhan, China, October 21-24, 2004.

Taylor, C. and F. Jim, 2001. NATE-network analysis of anomalous traffic events, A low-cost approach. New Security Paradigms Workshop.

Timofte, J., 2008. Intrusion detection using open source tools. Informatica Economica J., 2: 75-79.

Tran, D., W. Ma and D. Sharma, 2008. Automated feature weighting for network anomaly detection. Int. J. Comput. Sci. Network Security, 8: 173-178.

Vigna, G., S. Eckmann and R. Kemmerer, 2000. Attack languages. Proceedings of the IEEE Information Survivability Workshop, (ISW'00), IEEE Computer Society Press, Boston, MA, USA., pp: 163-166.

Vokorokos, L., A. Balaz and M. Chovanec, 2006. Intrusion detection system using self organizing map. Acta Electrotechnica Informatica, 6: 1-6.

Wang, H., D. Xhang and K.G. Shin, 2004. Change-point monitoring for the detection of DOS attacks. IEEE Trans. Dependable Secure Comput., 1: 193-208.