On Detecting Port Scanning using Fuzzy Based Intrusion Detection System

Wassim El-Hajj
College of Information Technology
UAE University
United Arab Emirates
Email: welhajj@uaeu.ac.ae

Fadi Aloul
Department of Computer Engineering
American University of Sharjah
United Arab Emirates
Email: faloul@aus.edu

Zouheir Trabelsi
College of Information Technology
UAE University
United Arab Emirates
Email: trabelsi@uaeu.ac.ae

***Abstract*—Intrusion detection is a mechanism used to detect various attacks on a wired or wireless network. Port scanning is one of the dangerous attacks that intrusion detection tries to detect. Snort, a famous network intrusion detection system (NIDS), detects a port scanning attack by combining and analyzing various traffic parameters. Because these parameters cannot be easily combined using a mathematical formula, fuzzy logic can be used to combine them; fuzzy logic can also reduce the number of false alarms. This paper presents a novel approach, based on fuzzy logic, to detect port scanning attacks. A fuzzy logic controller is designed and integrated with Snort in order to enhance the functionality of port scanning detection. Experiments are carried out in both wired and wireless networks. The results show that applying fuzzy logic adds to the accuracy of determining bad traffic. Moreover, it gives a level of degree for each type of port scanning attack.**

***Index Terms*—Intrusion Detection System, Fuzzy Logic, Port Scanning, Snort.**

I. INTRODUCTION AND RELATED WORK

Nowadays, using computers and computer networks in all communities all over the world has made computer network security an international precedence. Because, it is not feasible to build a secure system with no vulnerabilities, intrusion detection becomes an important area of research.

An intrusion detection system (IDS) is an automated system designed to detect malicious attacks on computer systems through the Internet. The main aim of Intrusion Detection Systems (IDS) is to protect the availability, confidentiality and integrity of critical networked information systems by identifying preferably in real time, unauthorized use, misuse, and abuse of computer systems [1,2].

A typical intrusion detection system consists of three functional components [8]: an information source, an analysis engine and a decision maker. The information source provides a stream of event records. This component can also be considered as an event generator. The analysis engine finds signs of intrusions. There are two basic approaches used to detect intrusions: misuse detection and anomaly detection. A decision maker applies some rules on the outcomes of the analysis engine, and decides what reactions should be done based on the outcomes of the analysis engine [9].

As mentioned earlier, the analysis engine used two basic approaches: misuse detection and anomaly detection. Misuse detection attempts to recognize attacks that follow a certain intrusion pattern. Such patterns are stored in the form of signatures in the database. Whenever a certain pattern matches a signature in the database, an attack warning is issued. These patterns have been recognized and reported by experts, but these systems are vulnerable to attackers who use new patterns of behaviors that cannot be detected by the system. Anomaly detection, on the other hand can be identified by recording unusual behavior of operations. An anomaly is something out of the ordinary, e.g., abnormal network traffic which is actually caused by unknown attacks. An anomaly detection system models normal behavior and identifies a behavior as abnormal (or anomalous) if it is sufficiently different from known normal behaviors [3].

The main work of building an anomaly intrusion detection system is to build a classifier which can classify normal event data and intrusion event data from an original data set. In [10], the authors presented an anomaly detection method by using a Hidden Markov Model to analyze the trace of system calls coming from a UNIX system. In [11], the authors established an anomaly detection model that integrated the association rules and frequency episodes with fuzzy logic to produce patterns for intrusion detection. In [12], the authors developed an anomaly intrusion detection system combining neural networks and fuzzy logic. In [13], the authors applied genetic algorithms to optimize the membership function for mining fuzzy association rules.

Although the work presented in the above research work makes significant contributions, it still has some flaws. Some of the above research work uses artificial intelligence techniques on anomaly intrusion detection, but most of their methods depend on static input and are not integrated in practical intrusion detection systems such as Snort. So the practicality of the suggested method can not be tested in real life.

In this paper, we update Snort by integrating it with a customized Fuzzy Logic controller. We call the new system "Fuzzy Based Snort (FB-Snort)". The aim behind this merge is to better detect port scanning and to reduce the false negative and false positive alarms. Our choice for using Fuzzy Logic was based on two main reasons: (1) No clear boundaries exist between normal and abnormal events, (2) fuzzy logic rules help in smoothing the abrupt separation of normality and abnormality (anomaly).

Our strategy starts by finding the normal traffic from abnormal traffic using Snort. Then, we pass some chosen parameters (section IV) to the Fuzzy Logic controller to get one unique parameter. This parameter decides whether an attack exists or not. As a result, FB-Snort reduces the false positive and the false negative alarms.

This paper is organized as follows: section II gives a background about Fuzzy Logic, Snort, and port scanning. Section III explains FB-Snort architecture. Section IV presents the input parameters to the fuzzy logic controller and the reasoning behind choosing them. Section V discusses the fuzzy logic controller. Section VI presents the experimental results and section VII concludes the paper and discusses future work.

II. BACKGROUND

In this section, we present a brief background on Fuzzy Logic, Snort, and port scanning. The reason for that is because our suggested method integrates Fuzzy Logic with Snort in order to better detect port scanning attacks.

Fuzzy logic, a widely deployed technology for developing sophisticated control systems [5,6,7], provides a simple way to get definite precise conclusion and solutions based on unclear, imprecise, ambiguous or missing input information.

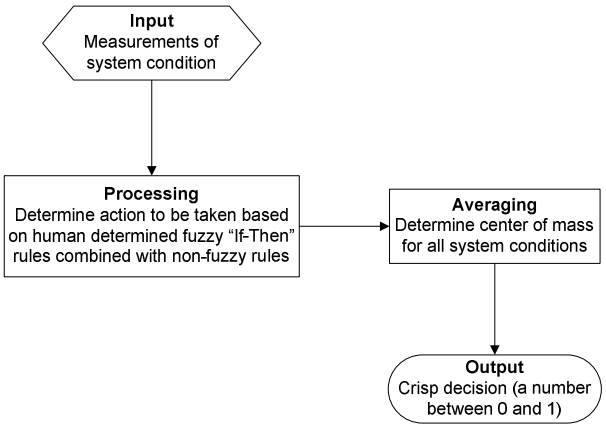

Fig. 1. Components of the fuzzy logic controller

Figure 1 shows the steps that the fuzzy logic controller is composed of [16]. The steps can be summarized as follows: (1) Receiving of one or more input values representing the measurements of the parameters to be analyzed or aggregated. (2) Subjecting the input values to fuzzy If-Then rules. The rules can be expressed in plain language words, for example, if a person is tall, back-pain is high. (3) Averaging and weighting the resulting outputs from all the individual rules into one single output decision. (4) Defuzzification of the output to get a crisp value between 0 and 1.

In general, two major components are needed to develop the fuzzy logic controller: (1) define membership functions for each input/output parameter and (2) design the fuzzy rules. The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs, define functional overlap between inputs, and determines an output response. The fuzzy logic rules use the input membership values as weighting factors to determine their influence on the output sets. In section V, we present the details about the controller that we integrated with Snort.

Snort is a Network Intrusion Detection System (NIDS); it is a signature-based NIDS which uses a combination of rule and preprocessors to analyze traffic [4]. The rules are flexible language that can describe traffic and it is used to create signatures to examine the packet. However, the preprocessor code allows deeper examination to the packets that can't be done by the rules alone. Preprocessor can perform different tasks such as port scanning detection and web traffic normalization. It gives snort the power of looking at and manipulating stream, in contrast of looking to single packet at a time as rules.

One of the attacks that Snort detects is port scanning. Attackers commonly attempt to connect to other hosts and scan their ports as starter to other attacks. Using this technique, the attacker tries to identify the existence of hosts on a network or whether a particular service is in use. Such services include email, telnet, file transfer, HTTP and DNS. Since a port is the Interface for each service within a computer, the information goes in and out of a computer through this port. Host scanning is usually characterized by an unusual number of connections to hosts on the network from an uncommon origin.

Snort attempts to detect four kinds of port scanning: (1) Portscan: it is one-to-one host scan where multiple ports are scanned on the destination host. (2) Distributed Portscan: it is a many-to-one host scanning where multiple ports are scanned on the destination host by scanner. (3) Decoy Portscan: it is many-to-one host scan where multiple ports are scanned on the destination host by scanner. It differs from Distributed Portscan in that Decoy Portscan connects to a single port multiple times. (4) Port Sweep: it is a one-to-many host scan where one host scans a few ports on each host.

In [14], the authors presented a method for detecting port scanning attacks using rule-based state diagram techniques. A set of rules corresponding with the appropriate thresholds was designed for intrusion decision. The parameters used in this work have static values, for example $\alpha = 1$ second and $\beta = 20$ packets. Many port scanning attacks occur within time more or less than 1 second, so this detection rule can not detect such

attack (scan). Also, some attacks send more than 20 packets to scan the victim. Therefore, assigning the number of attacking packets to 20 will not help in detecting port scanning, but on the other it will lead to false alarms.

In [15], the authors present a system called Fuzzy Intrusion Recognition Engine (FIRE) that uses fuzzy Logic to detect malicious activities against computer networks. To detect port scanning, FIRE uses a fuzzy logic controller that accepts three inputs and produces one output. The major problem in this approach is that it fails to detect many kinds of port scanning attacks.

As previously shown, the approaches presented above to detect port scanning have some problems. We suggest a new approach (FB-Snort) that integrates a customized Fuzzy Logic controller with Snort in order to reduce Snort false negative and false positive alarms. FB-Snort takes some input from Snort and then decides whether an attack exists or not. The architecture of FB-Snort is discussed next.

III. FB-SNORT ARCHITECTURE

Snort detects many kinds of attacks, but it gives many false positive and false negative alarms especially when detecting port scanning attacks. Furthermore, it doesn't show the levels of detected attacks. We designed Fuzzy-Based IDS (FB-Snort) to solve this problem. FB-Snort is a combination of snort and fuzzy logic. This combination will enhance the detection system within snort by reducing false alarms, and providing a system with levels of detected attacks. FB-Snort works within Snort and it is not a separate system. FB-Snort is supposed to improve on Snort by (1) adding levels to Snort alerts, (2) reducing the false positives and false negatives, and (3) generating the results efficiently. Actually, the results obtained by FB-Snort show more accurate results than Snort.

Figure 2 shows the architecture of FB-Snort in datails. It describes the flow of information between snort and the fuzzy-logic controller. As shown, the network tarffic passes through many PC's which has snort sensors IDS; these sensors collect traffic for snort so they can be analyzed. Traffic data received from the sensors are stored in snort database. Then traffic passes through snort processor which is able to analyze packets, to get IP addresses, prameters and other value which help snort to alert. From all the parameters collected by Snort, we care about the parameters that will be inputed to the fuzzy logic controller. These parameters are: (1) Average time between received packets by destination/victim (ART), (2) number of sent packets by source (NSP), and (3) number of received packets by destination/victim (NRP). These prameters are inputed to the fuzzy logic controller to calculate the attack level.

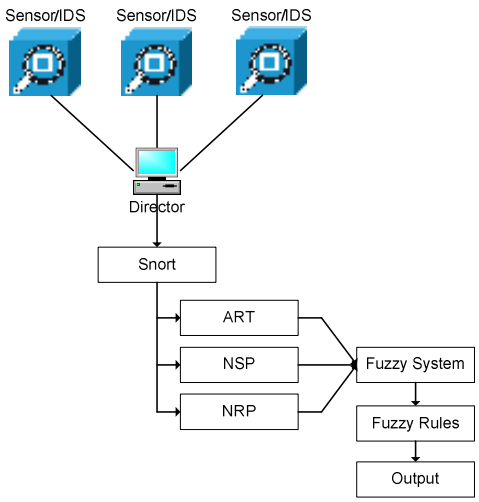

Fig. 2. FB-Snort Architecture

IV. FUZZY LOGIC PARAMETERS

As discusses above, we chose the following parameters as input to the controller in order to detect port scanning: ART, NSP, and NRP. While deciding on the parameters, we focused on the time when the packets were sent or received and their number. Time affects the detection process, because if time between received packets is too long, sources can not be considered attackers with a high probability. On the other hand, if the time is short, it means that there might be an attacker trying to scan the ports of the system. Three different experiments were performed in order to obtain threshold values for the input parameters. In these experiments, Advanced Port Scanner was used to perform various port scanning attacks. Also, *Commview* tool was used on the victim's host to monitor the parameter values.

TABLE I
RESULTS OF PORTS SCANNING EXPERIMENT USED TO BUILD FUZZY SYSTEM

Parameters/level	low scanning	medium scanning	high scanning
ART	0.00226	0.0017	0.0013
NSP	1470	2007	2794
NRP	3993	7135	8526

In the experiments, we performed three kinds/levels of scanning (1) low scanning, (2) medium scanning, and (3) high scanning. The experiments depend on the number of hosts performing port scanning. The different levels are used to assign ranges for port scanning attack. As a result, there are more chances to detect attacks. Sample results of the experiments are summarized in Table 1. When two hosts did port scanning to one host, the time between two received packets was 0.00226 sec, the number of sent packets by two hosts was 1470 packets, and the number of received packets by victim was 3993 packets. Same explanation goes for medium and high scanning attacks.

Notice that the ART value when "low scanning" is used is a large number (relatively); this means that the time between two received packets is large; therefore the level of attack is low. Similarly, the ART value when "high scanning" is used is a small number which means that there exists a high probability of having an attack because the time between two received packets is small. That is why the value of ART when "low scanning" is used is bigger than the value of ART when "high scanning" is used. Values in the "medium scanning" column mean that the time between received packets is medium, so level of attack is medium.

Looking at NSP and NRP parameters, each level of attack reflects the number of sent packets by source and the number of received packets by destination. The larger the value of NSP and NRP, the higher the probability of having a port scanning attack. These three parameters are extracted from Snort and fed to the fuzzy logic controller. The controller then combines then in an intelligent way and produces a single number indicating the level of the attack. In the next section, we discuss the fuzzy logic controller.

V. FUZZY LOGIC CONTROLLER

As mentioned in Section II, the fuzzy logic controller is composed of membership functions (for each input/output variable) and fuzzy rules. In this section we discuss these two components. The values of the parameters, taken from the experiments discussed in section IV, were used to tune the fuzzy logic membership functions and to create the fuzzy logic rules. Three input parameters are used (ART, NSP, and NRP). For each input parameter, three trapezoidal membership functions were designed: Low, Med, and High. Figure 3 shows the three trapezoidal membership functions for the NRP parameter (this snapshot was taken from Matlab-the software we used to design the fuzzy logic controller). The output parameter also has three trapezoidal membership functions distributed in the range [0.0, 1.0].

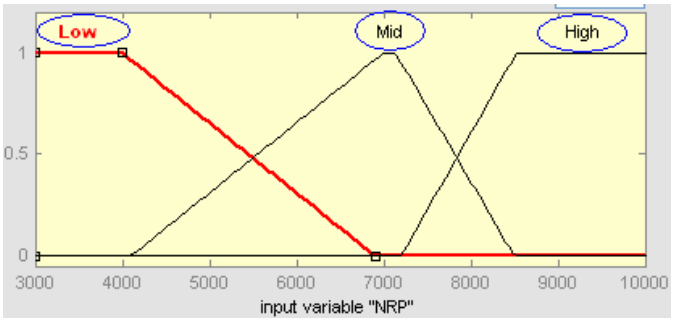

Fig. 3. NRP membership functions

After defining the input parameters, the fuzzy logic rules are designed and tested. These rules were written depending on the knowledge of detecting port scanning and the relationship between the parameters used to detect that attack (figure 4). Out of the twenty rules we designed to detect port scanning, we discuss the following three rules:

1. *If (ART is high) and (NSP is med) and (NRP is high) then (output is high).*This rule presents an attack with a high accuracy because the time between the received packets is low (high attack) and the number of packets the victim received is high.
2. *If (ART is high) and (NSP is med) and (NRP is low) then (output is low).* This rule presents an attack with a low accuracy because the time between the received packets is high (low attack) and the number of packets the victim received is low.
3. *If (ART is mid) and (NSP is high) and (NRP is mid) then (output is mid).* This rule presents an attack with a medium accuracy because the time between the received packets is medium and the number of packets the victim received is medium.

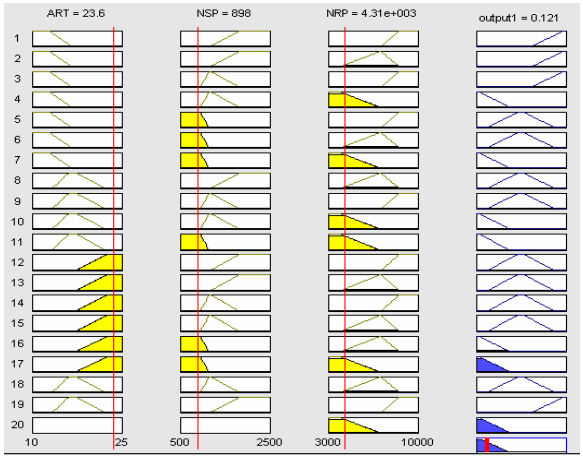

Fig. 4. The Fuzzy Logic rules used to detect port scanning. The inputs to the system are ART, NSP, and NRP.

Once Snort captures the packets, we move to the Fuzzy-Logic controller to detect port scanning. This is done by entering the parameters' values which are gathered by Snort into the fuzzy system, and then the rules will be applied on them. The output of the system shows the level of the detected port scanning attack. The next section discusses the experimental results.

VI. EXPERIMENTAL RESULTS

We installed and configured Snort on windows environment. Snort can be run in various modes: Sniffer mode, Packet logger, and network intrusion detection system (NIDS). In our system we run Snort as NIDS which is the most complex and complicated configuration mode in Snort. This mode allows Snort to analyze network traffic to be matched against a user defined rule set and performs several actions based upon what it discovers. Once Snort captures the packets, we move to the Fuzzy-Logic controller to detect port scanning. This is done by entering the parameters' values which are gathered by Snort to the fuzzy system. The output of the system shows the level of the detected port scanning attack.

In our experiments, we consider a different number of hosts communicating with each other, where some hosts are attempting to port scan other hosts. We then use Snort and FB-

Snort to analyze the traffic having in mind the following questions: Can FB-Snort detect whatever Snort detects? Does FB-Snort reduce the false alarms generated by Snort?

We start by using four hosts trying to ping and ftp each other. We place one host as a server and the others as clients. We then analyze the traffic (table 2). Our goal in this experiment is to study the traffic flow under normal conditions before introducing malicious nodes. The values presented in table 2 are used to tune the fuzzy logic parameters.

TABLE 2
PARAMETER VALUES UNDER NORMAL TRAFFIC

Parameter	Ping (sec)	FTP (sec)
ART	0.17	0.059
NSP	20	173
NRP	92	330

After tuning the fuzzy logic parameters, we conduct different port scanning attacks. Both Snort and FB-Snort are used to detect the attacks. The tools used in our testing are Snort, Snortsnarf, advanced port scan, and Commview. We did two similar experiments with different number of attackers in order to differentiate between different levels of attack. In the first experiment, 3 hosts performed port scanning on a single host and in the second experiment, 4 hosts performed port scanning on a single host.

TABLE 3
VALUES OF THE PARAMETERS GATHERED BY SNORT AND ENTERED TO THE FUZZY LOGIC CONTROLLER

Parameter\level	Medium Attack (3-to-1)	High Attack (4-to-1)
ART	16.9	15.6
NSP	1406	1925
NRP	6544	8495

In these experiments, both Snort and FB-Snort were able to detect the port scanning attacks. Table 3 presents the parameter values passed to the fuzzy logic controller. When that attack was classified as medium, the values for ART, NSP, and NRP were 16.9, 1406, and 6544 respectively. When that attack was classified as high, the values for ART, NSP, and NRP were 15.6, 1925, and 8495 respectively. Definitely, in both scenarios, an attack was taking place. But, the 4-to-1 attack was more obvious and powerful. FB-Snort was able to detect these attacks and moreover it gave an idea on how powerful or severe the attack was. In the first attack (3-to-1), the output of fuzzy system was 0.41, and in the second attack, the output was 0.877 (figure 5). The higher the number, the more severe the attack is. So, in these experiments FB-Snort outperforms Snort in the sense that it catches the attack and moreover it gives us an idea on how severe the attack is.

To show the weakness of Snort and the power of FB-Snort, we used *Frameip* tool to generate normal packets and send them to another host which is running both Snort and FB-Snort. Snort considered such normal behavior a port scanning attack (false positive alarm). On the other hand, FB-Snort did not complain on the traffic. As a result, we were able to achieve our aim which is an intelligent system that reduces false alarms. More testing need to be done, but the initial results look very promising.

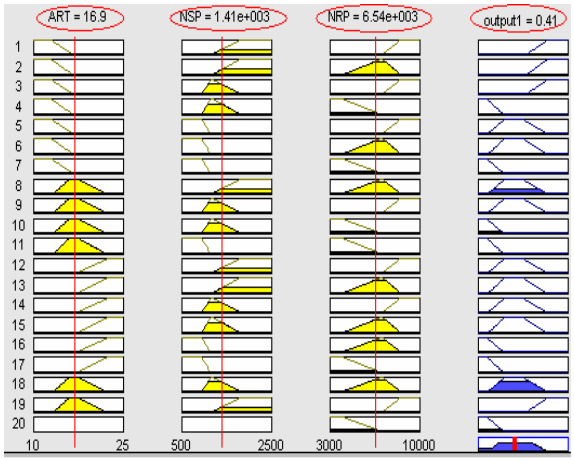

Fig. 5. The result FB-Snort produces when the parameter values presented in table 3 (medium attack) are inputted to the controller.

VII. CONCLUSION AND FUTURE WORK

As a summary we can say that anomaly-based network intrusion detection is a complex process and we focus on one of these anomaly intrusions which is port scanning. The variety in the network data stream, the amount of data to be processed, and the subtle and ever-changing ways that attackers breach systems, all conspire to complicate the task. In one of our testing, when more than five attackers did port scanning in the same time, the victim's machine had a denial of service. Therefore, this attack is really a complicated attack and it can be the starter for different types of attack. While this research does not solve the problem of finding all network-based attacks, the fuzzy intrusion detection (FB-Snort) holds promising results to be a high-level intrusion detection scheme. The use of a customized fuzzy logic controller enhances the capabilities of Snort to detect port scanning attacks. It also helps in reducing the false positive and negative alarms. The results show that the fuzzy system can be better combined with Snort to make Snort more intelligent and effective. Once we are sure that the fuzzy logic controller helps in detecting all types of port scanning attacks with no false positives and negatives, we will go ahead and completely merge it with Snort, obtaining a final usable version of FB-Snort.

REFERENCES

[1] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. Levitt, "Network Intrusion Detection", IEEE Network. 8(3):26-41, May/June 1994.

[2] D. E. Denning, \An intrusion-detection model," IEEE Trans. Softw. Eng., vol. 13, no. 2, pp. 222{232, 1987.
[3] Dae-Ki Kang, “Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation”, Proceedings of the 6th IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY,2005.
[4] http://www.snort.org
[5] ZADEH, L. A. Fuzzy sets. Information and Control 8, 3 (1965), 338.353. 118.
[6] ZADEH, L. A. Fuzzy algorithms. Information and Control 12, 2 (1968), 94.102.
[7] ZADEH, L. A. Fuzzy logic and its application to approximate reasoning. IFIP Congress (1974), 591.594.
[8] R.G. Bace, Intrusion Detection, Macmillan Technical Publishing, Indianapolis, USA, 2000.
[9] J.T. Yao, S.L. Zhao, L.V. Saxton, (2005), "A study on fuzzy intrusion detection", In *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005*, (Orlando, March 28-29, 2005), B. Dasarathy (Ed.), Proceedings of the International Society for Optical Engineering, Volume 5812, pages 23-30.
[10] Y. Qiao, X.W. Xin, Y. Bin and S. Ge, Anomaly Intrusion Detection Method Based on HMM, Electronics Letters, 38(13), 663-664, 2002.
[11] W. Lee and S.J. Stolfo, Data Mining Approaches for Intrusion Detection, 7th USENIX Security Symposium, 1998, pp. 79-94.
[12] M. Mohajerani, A. Moeini and M. Kianie, NFIDS: A Neuro-fuzzy Intrusion Detection System, Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems, 2003, pp348-351.
[13] W.D.Wang and S. Bridges, Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules, Proceedings of the 7th International Conference on Fuzzy Theory & Technology, Atlantic City, NJ, 2000, pp131-134.
[14] Urupoj Kanlayasiri, Surasak Sanguanpong and Wipa Jaratmanachot. A Rule-based Approach for Port Scanning Detection. 23nd Electrical Engineering Conference (EECON-23),Chiangmai November, 2000.
[15] J. Dickerson, J. Juslin, O. Koukousoula, and J. Dickerson, “Fuzzy intrusion detection,” Proceedings of the NAFIPS, Vancouver, British Columbia, Vol. 3, pp. 1506-1510, July 2001.
[16] Wasim El-Hajj. A Distributed Hierarchical Energy-Efficient Scheme for *Large Scale Mobile Wireless Ad Hoc Networks*. PhD thesis, Western Michigan University, 2006.