An Efficient Framework for Network Intrusion Detection

Hoque Mohammed Moshiul, and Alam Md. Mahbub

Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, Chittagong-4349, Bangladesh, E-mail: moshiul_240@cuet.ac.bd

Abstract

The computer networking is increasing day by day. With the growth of computer network, e-commerce, web service computer security is very important. The detection of attacks against computer network is becoming a harder problem to solve in the field of network security. Network intrusion detection (NID) is the process of identifying network activity that can lead to the compromise of a security policy. The Fuzzy Intrusion Recognition Engine (FIRE) is a network detection system that uses fuzzy system to assess malicious activity against computer networks. This paper describes an efficient intrusion detection system with the fuzzy logic. We specify a set of fuzzy rules that can be used to detect an intrusion in the network. This paper presents the design of the FIRE system and discusses how the fuzzy agents are used to perform correlation of multiple inputs for intrusion detection. All agents communicate with a fuzzy evaluation engine that combines the results of individual agents using fuzzy rules to produce alerts that are true to a degree. We have tested our model on some data obtained from networks under simulated attacks. In this paper, we present some experimental results that indicate the fuzzy system can easily identify port scanning and unauthorized servers.

Keywords: *Intrusion, Fuzzy Logic, Fuzzy Classifier, Attack Classifier and Fuzzy Rules.*

1. Introduction

Computer security plays an important role in modern communication. To get an efficient network it should required making the network free from hackings and other evil tasks. If it is not possible the important data may spread out overall underworld and the peace and mankind may be at some risk. In our modern world everybody tries to communicate each other but not try to open his task for others. For this reason an access control list is always maintain in the host net and also in the server net. But sometimes some people try to access the computer of others illegally and try to make the data be corrupted and to make sure the system may fall. To stop this efficient step should be taken. Network intrusion detection is the process of identifying network activity that can lead to the compromise of a security policy. Intrusion Detection attempts to detect computer attacks by examining data records observed by process on the same network. These attacks are typically split into two categories, host-based attacks and network-based attack. Host-based attack may detect routines normally use system call data from an audit process that tracks all system calls made on behalf of each user on a particular machine. These audit process usually run on each monitored machine. Network based attack detection routines typically use network traffic data from a network packet snuffer [1]. Anomaly Detection identifies activities that vary from established patterns of users, or groups of users. Maintaining a profile of each authorized user or group is useful in detecting anomaly attacks in systems, but it is difficult to maintain a behavior profile for each legitimate user when the number of users increases. Behavior patterns also change with the mental state of the person and thus detecting anomaly attacks by comparing patterns with user profiles gives rise to a large number of false positive alerts. In the field of computer security, one of the most damaging attacks is masquerading, in which an attacker assumes the identity of a legitimate user in a computer system. The attacks typically occur when an intruder obtains a user’s password or when a user leaves their workstation unattended without any sort of locking mechanism in place.

It is difficult to detect this type of security breach at its initiation because the attacker appears to be a normal user with valid authority and privileges. The next generation of intrusion detection tools will need to be able to perform correlation analysis of multiple inputs.

The field of intrusion detection is continually evolving. During the last five years, there has been a significant increase in the level of interest in anomaly detection and misuse detection. Researchers are trying to give more and more effective intrusion detection system to face the changing network environment. The problem of intrusion detection has been mentioned in details in computer security [2, 3, 4].The network information is discussed before the fuzzy logic is applied [5]. But it is more efficient not only discard network data but also use network information [6]. The detecting system changes its state and the attacker also changes their pattern. So, more effective system has to be applied. To face this demand, pattern-matching model has used in intrusion detection [7]. But the anomaly intrusion is increased day by day. Therefore, the fuzzy logic has been used in the field of intrusion detection in the network. In fuzzy logic intrusion detection fuzzy controller used to formulate the rule and got its accuracy [8]. But it cannot face the changing environment. Therefore, an autonomous agent used to learn system [9]. Among this back propagation techniques are more efficient in detecting intrusion [10]. Now a day's different approach are trying to use to detect the problem of anomaly [11, 12]. Artificial immune systems have been applied successfully in anomaly based computer network intrusion detection [13, 14, 15]. Data mining based on fuzzy association rules and statistical techniques has used for intrusion detection [16, 17]. In this paper, we proposed an intrusion detection system using a set of fuzzy rules for network intrusion detection. Our system can detect port scan and unauthorized servers. The goal of intrusion detection is to build a system which would automatically scan network activity and detect such intrusion attacks. Once an attack is detected, the system administrator can be informed who can take appropriate action to deal with the intrusion. The intrusion detection system is a system that has been used for detecting the intrusion in the network by using fuzzy logic. Fuzzy systems have the several important characteristics that suit intrusion detection very well. Fuzzy systems can readily combine inputs from widely varying sources. Many types of intrusion cannot be crispy defined and the degree of alert that can occur with intrusions is often fuzzy. The paper is structured as follows. In section 2 presents the some preliminary concepts of fuzzy logic and implications. Section 3, describes fuzzy c-means clustering algorithm. Section 4 explains the how anomaly detects with fuzzy inference rules. A proposed framework for intrusion detection is presented in section 5. Section 6, evaluates our system through experiments. Finally, we present our conclusion in section 7 with some discussions.

2. Fuzzy Logic and Operator

A fuzzy system is based on the concept of fuzzy logic. In fuzzy logic, objects can belong to a set and cannot belong to the set at the same time. Fuzzy sets define the linguistic notions and membership functions define the true value of such linguistic expressions. The membership degree to a fuzzy set of an object defines a function so called membership function where the universe of discourse is the domain and the interval [0,1] is the range. That function is called membership function [18]. A standard fuzzy space using the triangle function is shown in Fig. 1. With this linguistic concept, atomic and complex fuzzy logic expressions can be built.

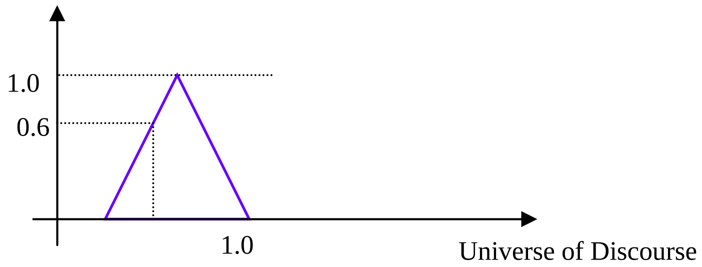


Fig. 1: Triangular membership function for a fuzzy set

For each classical logic operator (and, or, negation), there is a common fuzzy logic operator. The fuzzy logic operator has been presented in the Table 1. Where, condition is a complex fuzzy expression, consequence is an atomic expression and weight is a real number that defines the confidence of the rule.

Table 1: Fuzzy Logic Operators

Logic Operator	Fuzzy Operator
a AND b	Min (a, b)
a OR b	Max (a, b)
NOT a	1- a

2.1. Implication Relation

Fuzzy if/then rules are conditional statements that describe the dependence of one or more linguistic variable on another. The underlying analytical form of an if/then rule is a fuzzy relation called the implication relation. Some implication operators are given below. The Zadeh *max-min* implication operator can be represented as

$$\Phi_m[\mu_X(a),\mu_Y(b)] = (\mu_X(a) \wedge \mu_Y(a)) \vee (1-\mu_X(a))$$

Thus the membership function of the implication relation can be stated as

$$\mu(a,b) = (\mu_X(a) \wedge \mu_Y(b)) \vee (1-\mu_X(a))$$

The Mamdani *min* implication operator is a simplified version of Zadeh max-min proposed by Mamdani in connection with fuzzy control and is defined as

$$\Phi_C[\mu_X(a),\mu_Y(b)] = \mu_X(a) \wedge \mu_Y(b)$$

Where, $\mu_X(a)$ = Membership function of a, $\mu_Y(b)$ = Membership function of b, $\vee$ = Fuzzy OR operation, and $\wedge$ = Fuzzy AND operation respectively.

3. Fuzzy C-Means Clustering

Fuzzy c-means algorithm generates fuzzy sets for every observed feature [19].The fuzzy c-means (FCM) algorithm minimizes the objective function [20].

$$J(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^{m} \|x_k - v_i\|^2$$

Subject to $u_{ij} \in [0,1]$ and $\sum_{i=1}^{c} u_{ik} = 1 \forall k$. U is the partition matrix that shows to what degree k-th data point x_k belongs to each cluster as measured by its distance from the prototype of the *i*-th cluster, V_i and *m* is a weighting exponent, *c* is the number of clusters and *n* is the number of data points. FCM is an iterative algorithm to find cluster centers that minimize a dissimilarity function.

4. Anomaly Detection with Fuzzy Rules

In the self or non-self space $[0,1]^n$, an element x in this space is represented by a vector $(x_1,....x_n)$where $x_i \in [0,1]$. A fuzzy detection rule has the following structure:

If $x_1 \in T_1 \wedge ....x_n \in T_n$ **then** non_self

Where, $(x_1,....x_n)$: element of the self/non-self space, T_i: fuzzy set, $\wedge$: fuzzy conjunction operator (in our case, min (x)). The fuzzy set T_i is defined as a combination of basic fuzzy sets (linguistic values). Given a set of linguistic values S = {S_1,…,S_m} and a subset $\hat{T}_i \subseteq S$ associated to each fuzzy set T_i,

$$T_i = \bigcup_{S_j \in \hat{T}_i} S_j$$

Where $\bigcup$ corresponds to a fuzzy disjunction operator. We used the addition operator defined as:

$$\mu_{A \bigcup B}(x) = \min\{\mu_A(x) + \mu_B(x), 1\}$$

An example of fuzzy detector rules in the self/non-self space with dimension n = 3 and linguistic values

S = {L, M, H}: **If** $x_1 \in L \wedge x_2 \in (L \cup M) \wedge x_3 \in (M \cup H)$ **then** non_self

In our experiments, the basic fuzzy sets correspond to a fuzzy division of the real interval [0.0, 1.0] using triangular and trapezoidal fuzzy membership functions. Fig. 2 shows an example of such a division using five basic fuzzy sets.

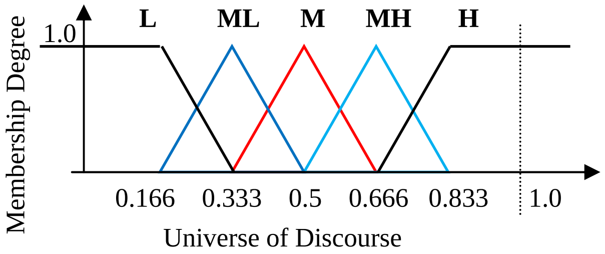


Fig. 2: Partition of the interval [0.0, 1.0] in basic fuzzy sets.

Given a set of rules R_1, R_2,…., R_m, the degree of abnormality of a sample x is defined by

$$\mu_{non\_self}(x) = \max_{i=1,\dots m}\{eval_{R_i}(x)\}$$

Where $eval_{Ri}$ (x) represents the fuzzy true value produced by the evaluation of the condition of fuzzy rule R_i, and μ_{non_self} (x) represents the degree of membership of x to the non-self set; thus, a value close to zero means that x is normal and a value close to 1 indicated that it is abnormal. The purpose of the fuzzy logic operator is to use imprecise and heuristic knowledge to describe the state of the system as normal or as a specific attack (if the attack is known) or just as an attack (if the attack is unknown).

4.1 Fuzzy Inference Rules

The fuzzy rules are derived on the basis of system design. They are base on the tuning of the various combinations. From various combinations the best matches have to be taken as rule. If the source and destination port is low so there is a possibility of intrusion. A collection of fuzzy rules with the same input and output variables is called a fuzzy system. Fuzzy rules have the following forms:

IF condition THEN consequent [weight].

Where, condition is a complex fuzzy expression, consequent means an automatic expression and weight is real number that defines the confidence of the rule. For example: R: IF *a* is HIGH and *b* is LOW THEN pattern is normal [0.4]. As the difference between the normal and the abnormal are not distinct, but rather fuzzy, this module can reduce the false signal in determining intrusive activities. We have proposed a set of fuzzy rules for calculating system data in Table 2. The system data then have used for intrusion detection. The Fuzzy Controller takes two inputs from System Reader then applies the rules.

Table 2: Fuzzy Rules for Calculating System Data

Rule no.	Set of fuzzy rules
1	IF SD=L AND SP=L THEN R=H
2	IF SD =L AND SP=LM THEN R=MH
3	IF SD =L AND SP=M THEN R=M
4	IF SD =L AND SP=MH THEN R=LM
5	IF SD =L AND SP=H THEN R=L
6	IF SD =LM AND SP=L THEN R=H
7	IF SD =LM AND SP=LM THEN R=MH
8	IF SD =LM AND SP=M THEN R=M
9	IF SD =LM AND SP=MH THEN R=LM
10	IF SD =LM AND SP=H THEN R=L
11	IF SD =M AND SP=L THEN R=M
12	IF SD=M AND SP=LM THEN R=MH
13	IF SD=M AND SP=M THEN R=M
14	IF SD=M AND SP=MH THEN R=LM
15	IF SD=M AND SP=H THEN R=L
16	IF SD=MH AND SP=L THEN R=H
17	IF SD=MH AND SP=LM THEN R=MH
18	IF SD=MH AND SP=M THEN R=M
19	IF SD=MH AND SP=MH THEN R=LM
20	IF SD=MH AND SP=H THEN R=L
21	IF SD=H AND SP=L THEN R=MH
22	IF SD=H AND SP=LM THEN R=M
23	IF SD=H AND SP=M THEN R=LM
24	IF SD=H AND SP=MH THEN R=L
25	IF SD=H AND SP=H THEN R=L

5. Proposed Framework for Intrusion Detection

A typical intrusion detection system consists of three functional units: an information source, an analysis engine and a decision maker [21]. The information source provides a stream of event records as well as event generator. It monitors different data sources and generates data that are well formatted and suitable for analysis. The analysis engine finds signs of intrusions. 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. The proposed intrusion detection system is a blackboard-based intrusion detection system that learns new attack patterns and later identifies them in the computer network. In the system architecture, there are six autonomous agents that interact with the blackboard. The agents are Network Reader (A1), Initial Analyzer (A2), System Data Reader (A3), Attack Classifier (A4), Autonomous Agent (A5) and the Report Generation Agent (A6). The overall proposed architecture of intrusion detection system is presented in Fig. 3. In this architecture, A1 and A2 process information which are collected from Network and work as an event generator. A3 and A4 work as an analysis engine unit with system data. However, A5 and A6 work as a decision making unit.

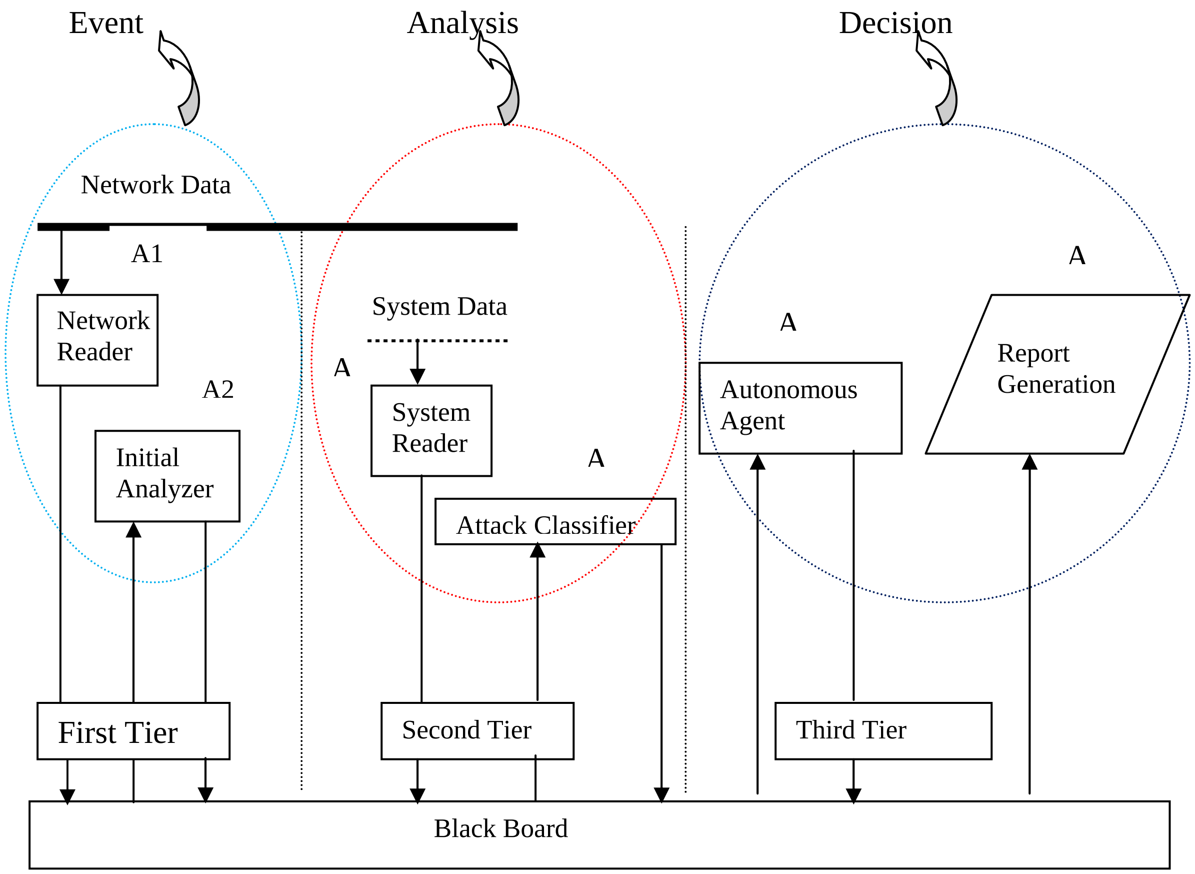


Fig. 3: Proposed Intrusion Detection System Architecture

5.1. Network Reader

A1 is the Network Reader. It collects network data with the help of a program called tcpdump. Tcpdump is a network utility tool that records network data in a specific format. The A1 autonomous agent collects network data in groups of 1000 data packets (network activity information) and pastes them on the blackboard. The network reader pastes the seen data stream on the blackboard.

5.2. Initial Analyzer

A2 is the initial analyzer. It calls a Rule-based classifier system that is written. This classifier system analyzes the network data in Fig. 3. This knowledge-base analyzer identifies different attacks in the system and posts alerts, if any, on the blackboard. The Predefined Feature (K) is in the form of 6-tuples of parameters. Its characteristic can be defined system as follows.

$$K = (SYN, DA, DP, W, SEQ, T)$$

Where, SYN is a flag on TCP header, which identifies a connection request. DA is a destination IP address of packet. DP is the destination port number. W is a window size of TCP segment. SEQ is an interval value of two sequence numbers. T is an interval time of two consecutive TCP segment. It analyzes events with a set of detection rule. To identify intrusive patterns, by considering the first 4 features of K, sequence of packets must conform to several conditions. First, a packet must have the SYN flag in its TCP header. Secondly, each packet has the same destination IP address (DA) and the same destination port (DP). Finally, the packet must have the same window size (W).

5.3. System Reader

A3 is the System reader that gathers system specific information on the protected system and posts it on the blackboard. These system data are very helpful in detecting the extent of damage caused by any attack. The type of information gathered includes available network bandwidth, CPU usage, network packets/second, memory usage, number of connections, connection attempts, protocol, source address to destination ports ratio (variety of ports accessed) and packet length. This system is read the input membership value for the next agent. The system reader from the system reads the membership value.

5.4. Attack Classifier

A4 is the attack classifier that identifies different sub-classes of intrusions present in the network data. This agent sends the system information from the blackboard to a fuzzy classifier that uses the multiple-fault diagnosis concept to perform the above function and posts its result back to the blackboard. The result states which kind of attack is present and what its probability of presence in the dataset is. In our case study, there are two significant features, SD and SP. SD is a value of source and destination port TCP segments and SP is the service port of two consecutive TCP segment.

5.5. Autonomous Agents

A5 is the autonomous agents that give full details of the attacks. For this system we have used here the back propagation algorithm [1]. The algorithm calculates the weight for the next layer. The required equation for calculating weight for the next agent is given below.

$$W(N+1) = W(N) - 2[T - \Phi_N]\Phi_N[1 - \Phi_N]\Phi_{N-1}$$

Where, W (N+1) = Next weight or output, W (N) = Previous weight or output taken from the attack classifier. Φ_N = Queue length for present system. Φ_{N-1} = Queue length for previous System. T = Packet loss for the current System. On the basis of this weight value the system can take the decision of the intrusion. These weight values have to send in the next agent and the next agent will create a warning message.

5.6. Report Generation

A6 is the Report Generation agent displays a complete report of the analysis to the user. The simple Network intrusion Simulator gives average value for intrusion for an intrusion possibility. If this value is greater than a certain limit then the system will in a danger zone. On the basis of this warning message the administrator cut the particular connection. For every connection the percentage value has to be calculated and then the warning message gives warning to the administrator at every time.

5.7. Blackboard Architecture

The blackboard architecture is considered one of the most general and flexible knowledge system architectures for building decision-based applications. As a result, the blackboard-based architecture is considered a good solution in developing our proposed Intrusion Detection System. The proposed architecture will also include the use of Autonomous Agents that are software agents, which perform certain security monitoring functions at a host. The proposed architecture consists of autonomous agents that are integrated in a blackboard-based architecture and placed in a tier form.

6. Experimental Results

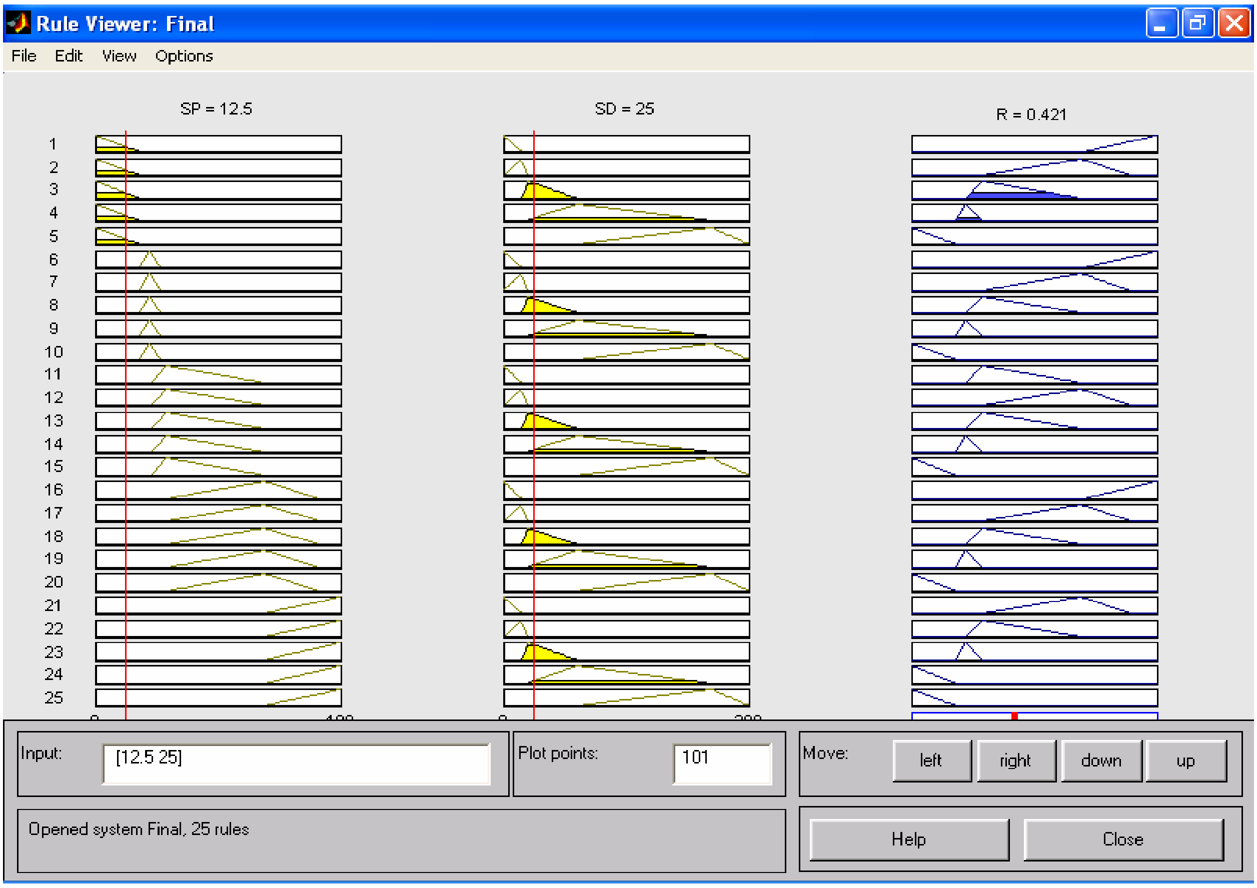
In the experimentation, the rules are created using the fuzzy system editor contained in the Mathlab Fuzzy Toolbox. This tool contains a graphical user interface that allows the rule designer to create membership functions for each input or output variable. In this experimentation we have used fuzzy inference system.

6.1 Data Collection

A monitor was developed to combine the outputs for detecting attacks. The monitor determines the fuzzy threats present by applying fuzzy rules to the inputs observed form the Fuzzy Intrusion Recognition Engine (FIRE). Choosing the best data elements to monitor in the network stream is critical to the effectiveness of the intrusion detection system. The system monitors all traffic during a two week sliding collection window. The data is logged on a data collection host where one or more FIRE agents are present. The individual FIRE agents each monitor a specific type of data to create a set of observed metrics about its data source. We use observation time interval of 10 minutes, though this time is chosen somewhat arbitrary. Al together, the five FIRE agents monitor a total of 64 separate metrics. Over time, the metrics are used to establish the values for the fuzzy sets in the intrusion recognition engine.

6.2. Port Scanning

Port scanning is one of the serious attacks that intrusion detection tries to detect. The scans may utilize a variety of protocols. We use SDP to identify a single connection among source, destination and service port. A complete fuzzy system for the rule set is shown in Fig. 4. Each system has five membership functions with triangular distribution of each. The input domain has clipped in leftmost and maximum at the rightmost rules. Therefore, within the rule domain, distribution has too high or low values.



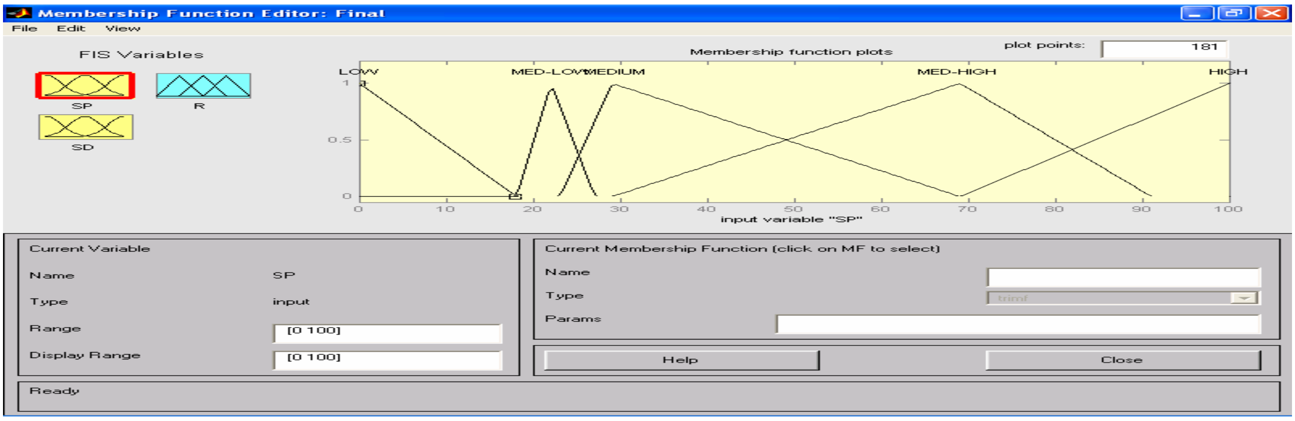
Input: [12.5 25]

Plot points: 101

Opened system Final, 25 rules

Fig. 4: Ports scan detection system. Inputs are the number of source or destination, or port combinations and the number of service ports observed and the number of destination hosts.

For evaluation each type of scan for each fuzzy set, we have to apply fuzzy c-means algorithm to the data gathered during the observation period before running the attacks. This produces the values used to quantify the fuzzy rules. Representing resulting membership functions for the number of service ports observed are shown in Fig. 5.



plot points: 181

Current Variable

Name SP

Type input

Range [0 100]

Display Range [0 100]

Current Membership Function (click on MF to select)

Name

Type trimf

Params

Ready

Fig. 5: Membership functions for the number of observed service ports

6.3. Unauthorized Servers Detection

Another intrusion detection scenario that is potentially more damaging that the prior scenario is the situation where an attacker has managed to invade a system and install a backdoor or Trojan horse program that can lead to further compromise. The control surface for this system is shown in Fig. 6.

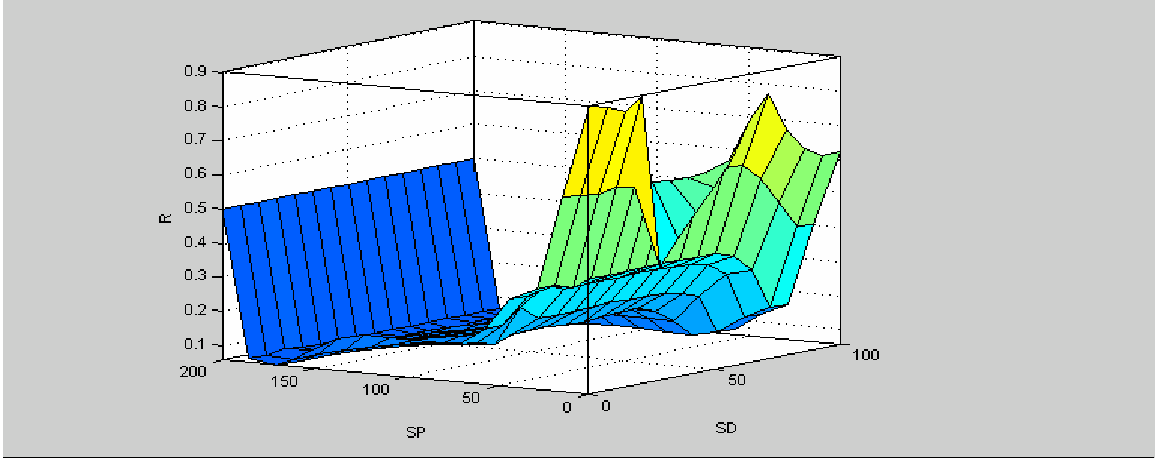


Fig. 6: Surface viewer for calculating Intrusion.

This surface viewer gives representation for input and output. The front axis is the number of source and destination port and retreating axis is the number of service port observes for the system. Table 3 shows that the simple network intrusion simulator gives average value for intrusion is 59.76% in test case 1, which indicates an intrusion possibility in the existing network and average value for intrusion is 15.84% in test case 2 that indicates there is no intrusion possibility in the existing network.

Table 3: Data sets for Intrusion detection

Test Cases	Case 1	Case 2
Total no. of packets that required service	1404	1233
No. of packet routed	1200	1193
Total packet drop	124	0
Packets left in queue	80	40
No. of times packet drop occurred	10	0
Percentage of packet routed (%)	85.47	96.75
Percentage of packet dropped (%)	8.83	0
Average value for intrusion (%)	59.76	15.84
Possibility for intrusion	Intrusion	No intrusion

7. Conclusion

Reliance on internet and world wide connectivity has raised the potential damage that can be inflicted by attacks over internet against remote systems. Successful attacks inevitably occur despite the best security measures. Therefore, intrusion detection has become a necessary element of information security to detect malicious attacks with the aim of preserving valuable systems from widespread damages and identifying vulnerabilities of the intruded system. Anomaly-based network intrusion detection is a complex process. The variety in the network data stream, the amount of data to be processed, and he subtle and ever-changing ways that intruders penetrate systems all conspire to complicate the task. The system successfully detected the intrusion. If the output value is 33% the system will give a warning message. If the value is greater than 66% then the system will in a danger zone and required to disconnect the system. This system will run independently and will run every moment. So it can detect more accurately than existing system. It provides faster system than previous existing system. It works well on basically anomaly detection and misuse detection. Most importantly, this research has laid a solid groundwork for fuzzy intrusion detection and revealed promising areas of continued exploration. The proposed IDS are suitable for large organization. Therefore, it is very much important to modify the architecture for the small organization. But it will depend on some factor. The carrier frequency will play a key factor in intrusion detection system and the frequency distribution will be important in future. Another possibility is to use data mining algorithm to detect intrusion. Machine learning technique, fuzzy association rule based classification with the combination of neural network and fuzzy logic may be the greatly improve the intrusion detection efficiency.

References

[1] Giles, L. and Mark W., “Routing in optical multistage interconnection networks: a neural network”, Journal of Lightwave Technology, 1995, vol 13, no. 6, pp. 1111-1115.

[2] Amoroso, E., “Intrusion detection”, New Jersey: Intrusion.net Books, 1999, 1st edition.

[3] Allen, J., Alan, C., William, F., John, M.’ Jed, P., and Stoner, E., “State of the practice of intrusion detection technologies”, Technical Report, Networked Systems Survivability Program, Carnegie Mellon, Software Engineering Institute, Pittsburgh, Pennsylvania, 2000, pp. 220.

[4] Stefan, A., “Intrusion detection systems: A survey and taxonomy”, Technical Report, Dept. of Computer Engineering, Chalmers University of Technology, G¨oteborg Sweden, 2000, pp. 27.

[5] Kumar, S., “Classification and Detection of Computer Intrusions”, Ph.D. Thesis, Department of Computer Sciences, West Lafayette, Purdue University, 1995.

[6] Mukherjee, B., Heberlein, L. and Levitt, K., “Network Intrusion Detection”, IEEE Network, 1994, vol. 8, no. 3, pp. 26-41.

[7] Kumar, S. and Spafford, E. H., “A pattern matching model for misuse intrusion detection”, National Security Conference,tWest Lafayette 1994, pp.11-21.

[8] Dickerson, E. J., Juslin, J, Koukousoula, O. and Dickerson, J. “Fuzzy Intrusion Detection,” Electrical and Computer Engineering Department, Iowa State University, USA.

[9] Crosbie, M. and Spafford, “Active Defense of a Computer System using Autonomous Agents”, Technical Report, COAST Group, West Lafayette, Dept. of Computer Science, Purdue University, 1995, pp. 14.

[10] Dasgupta, D. and Forrest, S., “Novelty detection in time series data using ideas from immunology,” International Conference on Intelligent System, Reno, Navada, 1995, pp. 82-87.

[11] Keogh, E., Lonardi, S. and Chiu, B., “Finding surprising patterns in a time series database in linear time and space,” International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, ACM, 2002, pp. 550-556.

[12] Hofmeyr, S. and Forrest, S., “Architecture for an artificial immune system,” Journal of Evolutionary Computation, 2000, vol. 8, no. 4, pp. 443-473.

[13] Dasgupta, D. “Artificial immune system and their application,” New York, Springer-Verlag, 1999.

[14] Kephart, J., “A biologically inspired immune system for computers,” In Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems Artificial Life, Cambridge, MA, 1994, pp-130-139.

[15] Bezdek, J. C., “Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[16] Tajbakhsh, A., Rahmati, M., Mirzaei, A., “Intrusion Detection using Fuzzy Association Rules, Journal of Applied Soft Computing, 2009, vol. 9, pp. 462-469,.

[17] Biermann, E., Cloete, E., Venter, I. M., “A Comparison of Intrusion Detection System, Journal of Computer Security, 2001, vol. 20, no. 8, pp. 676-683.

[18] Tsoukalas, L. H. and Uhrig, R. E., “Fuzzy and Neural Approaches in engineering,” John Wiley & Sons, Inc. New York, USA, 1996, 1st edition.

[19] Chimphlee, W., Abdullah, A. H., Sap, M. N. M., Chimphlee, S. and Srinoy, S., “Integrating Genetic Algorithms and Fuzzy c-Means for Anomaly Detection”, IEEE Indicon Conference, Chennai, India, 2005, pp. 575-579.

[20] Kanlayasiri, and Sanguanpong, “Network-based Intrusion Detection Model for Detecting TCP SYN flooding”, National Computer Science and Engineering Conference, Bangkok, Thailand, 2000, pp.148-153.

[21] Bace, R.G., Intrusion Detection, Macmillan Technical Publishing, Indianapolis, USA, 2000.

Amount of Figures: six (06)
Amount of Tables: Three (03)

Article received: 2009-04-23