Volume 1, No. 1, May-June 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

RULE-BASE IDS FOR APPLICATION LAYER USING FUZZY LOGIC

S.Sangeetha*, M.S.Vinu, S.Haripriya and S.G. Mohana Priya

Department of CSE,

Angel College of Engineering and Technology,

Tirupur, India

visual.sangi@gmail.com

Abstract: The objective of this paper is to develop a Fuzzy Rule-Base Based Intrusion Detection System on Application Layer which works in the application layer of the network stack. It consists of semantic IDS and Fuzzy based IDS. Rule based IDS looks for the specific pattern which is defined as malicious. A non-intrusive regular pattern can be malicious if it occurs several times with a short time interval. At application layer, HTTP traffic's header and payload are analyzed for possible intrusion. In the proposed misuse detection module, the semantic intrusion detection system works on the basis of rules that define various application layer misuses that are found in the network. An attack identified by the IDS is based on a corresponding rule in the rule-base. An event that doesn't make a 'hit' on the rule-base is given to a Fuzzy Intrusion Detection System (FIDS) for further analysis. In a Rule-based intrusion detection system, an attack can either be detected if a rule is found in the rule base or goes undetected if not found. If this is combined with FIDS, the intrusions went undetected by RIDS can further be detected. These non-intrusive patterns are checked by the fuzzy IDS for a possible attack. The non-intrusive patterns are normalized and converted as linguistic variable in fuzzy sets. These values are given to Fuzzy Cognitive Mapping (FCM). If there is any suspicious event, then it generates an alarm to the client/server. Results show better performance in terms of the detection rate and the time taken to detect. The detection rate is increased with reduction in false positive rate for a specific attack.

Keywords: Semantic Intrusion detection, Application Layer misuse detector, Fuzzy Intrusion detection, Fuzzy Cognitive Mapping, HTTP intrusion detection.

I. INTRODUCTION

Intrusion detection is the process of monitoring computers or networks for unauthorized entrance, activity, or file modification. Most of the commercially available Intrusion Detection Systems (IDS) work in the network layer of the protocol stack. This paves way for attackers to intrude at various other layers, especially in the application layer namely HTTP [11]. Application layer based IDS blocks the application layer based attacks that network layer intrusion detection system can't block. Firewalls works in network layer and prevents the attacks entering the network through unauthorized port. Some comlex threats can enter through authorized port (HTTP 80) and they go undetected. These threats can be detected by Application layer IDS. Misuse detection uses rule based detection that follow a signature-match approach where attacks are identified by matching each input text or pattern against predefined signatures that model malicious activity [8]. The pattern-matching process is time-consuming. Moreover, attackers are continuously creating new types of attacks. These attacks can be detected by the IDS, if it knows about the attack in the form of signatures. Attack signatures can be specified either in a single-line [12] or by using complex script languages [10]and are used in rule base to detect attacks. Because of the continuously changing nature of attacks, signatures and rules should be updated periodically on IDS.

Rule-based Intrusion Detection System (RIDS) looks for specific pattern that are defined as malicious. A non-intrusive regular pattern can be malicious if it occurs several times with a short time interval. The non-intrusive patterns are checked by the fuzzy component of the proposed architecture for a possible attack. The detection rate increases by checking the non-intrusive patterns using the fuzzy component. This paper proposes a Rule-Base Ids For Application Layer Using Fuzzy Logic which aims at designing the above solution for one of the application layer protocols namely HTTP (Hyper-Text Transfer Protocol). Such an HTTP based semantic IDS detects both the header based attacks and payload (which typically consists of HTML and script) based malicious content.

II. ARCHITECTURE OF THE FUZZY RULE-BASE IDS

The architecture of the system is as shown in Figure. 1. The block diagram shows the order in which the different modules process the incoming payload. The HTTP data capture block collects the application-layer traffic from the network. Captured data is then separated into the header and payload parts and are forwarded to separate buffers [13].

The Header parser [6] module reads the header and prepares a list of the objects in the HTTP packets. Each object represents a field of the HTTP protocol [1] and is a five tuple *<message-line, section, feature, operator, content>*.

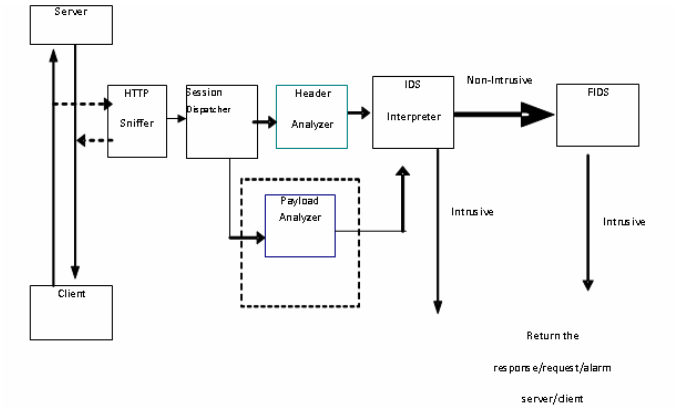

Figure 1. Block diagram of Fuzzy Rule-Base IDS

This sequence of objects is given to the IDS interpreter that refers to the rule-base and correlates the different objects to trigger one or more of the rules. Simultaneously the HTML parser parses the HTML data and searches for misappropriate usage of tags and attributes and also observes for the javascript based attacks injected in the HTTP [7]. The state transition analysis is done by defining states for every match. The incoming pattern is semantically looked-up only in specified states, and this increases the efficiency of the IDS pattern-matching algorithm [8]. If the pattern matches with some predefined pattern then it generates intrusion alert to client/server. If non-Intrusive, the output of the rule-based IDS goes to the Fuzzy IDS for further analysis [9]. Fuzzy Cognitive Mapping captures different types of intrusive behavior as suspicious events and generates an alert to the server/client, if there are any attacks.

III. FUZZY COMPONENT FOR NON-INTRUSIVE TRAFFIC

Parts of traffic that get past the rule-based intrusion detection system with no matches of intrusion are fed into the fuzzy component for further analysis. A functional block diagram of the fuzzy component is shown in Figure. 2.

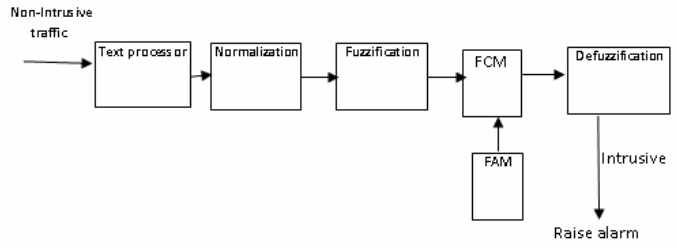

Figure 2. Functional blocks of FIDS

The traffic is first given to a text processor such as *awk*, which helps in finding the number of occurrences of a specific pattern in it. These nos. are later normalized to keep the obtained values in a specific range to aid relative comparison. The normalized values are fuzzified into linguistic terms of fuzzy sets before feeding to the Fuzzy Cognitive Mapper (FCM) [4]. The output of the text processor for Denial of Service attack. The output of this is normalized between 0.0 and 1.0 which then goes for fuzzification. Fuzzification converts a normalized value into linguistic terms of fuzzy sets. The output of the fuzzification is given for Fuzzy Cognitive Memory [2,3] which makes use of Fuzzy Associative Memory (FAM). Figure 3 shows the output of Text Processor for Denial of Service Attack.

When an IDS fails to detect intrusive pattern, it is called as a false negative and when it detects a legitimate traffic to contain intrusive patterns, it is called as false positive. In a signature based IDS, false alarms can either be a result of wrongly coded signatures or due to the presence or absence of signatures for specific patterns. FCM is used to calculate the rate of false alarms by drawing a map that correlate various events, thereby evaluates the rules that gets triggered. Defuzzification uses a mean of maxima method to get the final crisp value to classify the pattern as intrusive or non intrusive using a threshold value.

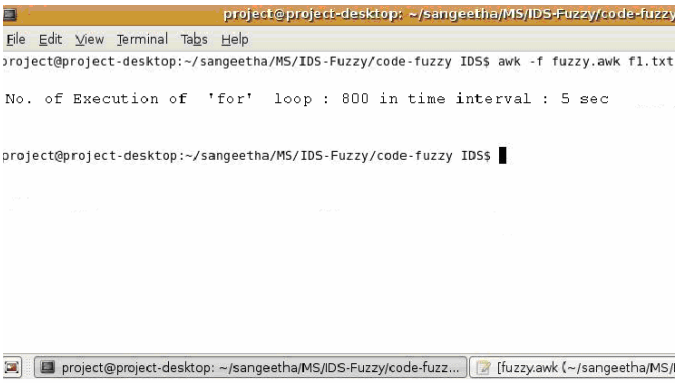

Figure 3. Output of Text Processor for Denial of Service Attack

A. WORKING OF FUZZY COGNITIVE MAPPER IN IDS

Fuzzy rules are constructed based on a map of multiple inputs to a single output. For eg., No. of login failures, time interval between any two login failures, time duration of a login session, etc. Malicious activities that are defined by one or more fuzzy rules are mapped using the FCM. The FCM uses a Fuzzy Associative Map (FAM) to evaluate the fuzzy rules to generate an alert that could fall under either of *very high, high, medium, low* or *very low* categories, based on the severity of the attack.

The following example demonstrates the sequence of events in the fuzzy component identifies a brute-force attack, where an intruder tries to login with several users' passwords and fails. This attack can be identified by observing the number of login failures and the time interval between each failure.

FCM for *login_failure* is shown in Figure. 4, which shows that if *login_failure* is very high for small *interval* of time and for *same machine*, then there is a suspicious event. ++, +, ∈, -& -- represents *very high, high, medium, low & very low* respectively. In Figure. 4, the *time interval* for login failure is *small* which is represented by '-' and no. of login failure is *high* which is represented by '+'.

Fuzzy rule: no. of *login_failure* is *very high* AND *time interval* is *small* is triggered which identifies that the specific scenario may be due to a brute-force attack. FAM table for a brute force attack as shown in Table 1 is used to evaluate this rule.

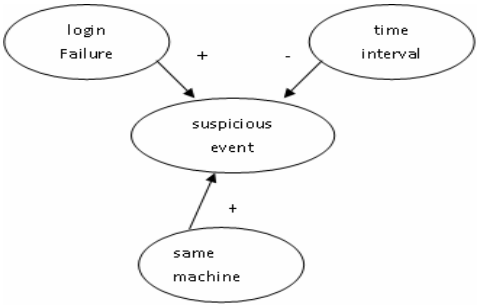

Figure 4. FCM for login_failure

B. FUZZY ASSOCIATIVE MEMORY BY FUZZY RULES

Fuzzy Associate Memory (FAM) is used to map fuzzy rules in the form of a matrix. These rules take two variables as input and map them into a two dimensional matrix. The rules in the FAM follow a simple *if-then-else* format. Fuzzy Associative Memory facilitates the conclusion of the rate of false negatives for few attacks such as Denial of Service (DoS) and brute force attacks.

Table 1. Fuzzy Associative Memory for a Brute force attack

t ***x***	***VS***	***S***	***έ***	***H***	***VH***
VS	VS	VS	S	έ	έ
S	VS	Έ	S	S	έ
έ	H	H	έ	έ	S
H	VH	H	H	έ	VS
VH	VH	VH	H	VS	VS

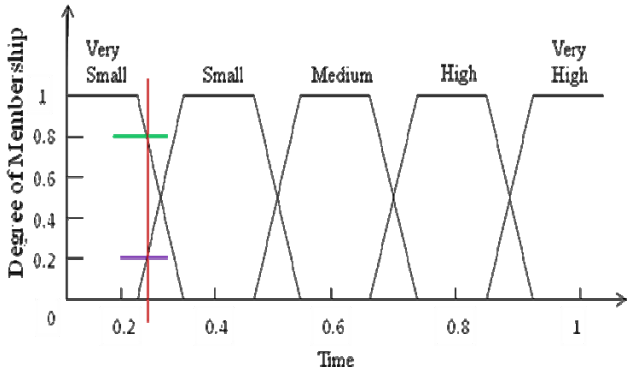

Figure 5. Linguistic representation of time interval during Brute force attack

Table 1 shows that the FAM table for a Brute force attacks in a matrix format. Rows in this table represent the rate of *no. of login failure* and the columns represent the rate of *time interval* between each failure. A linguistic representation of the same is as shown below in Figure. 5.

The *time interval* between each login failure is taken in X axis as a normalized value. The degree of membership is taken in Y axis. The min-max normalization scheme is used to normalize the time interval for login failure to a common range i.e., between 0 and 1. Figure. 6 shows the time values assigned to the linguistic variables (very small, small, medium, high and very high). Figure.6 shows the *login_failure* values assigned to the linguistic variables.

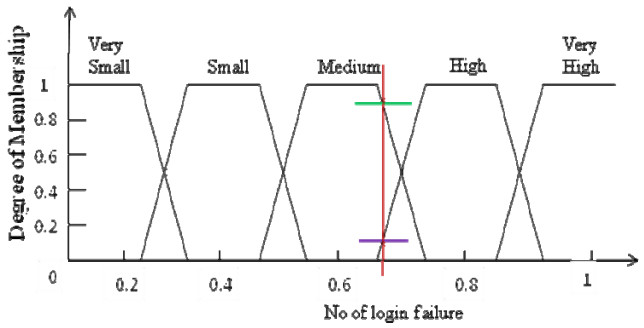

Figure 6. Linguistic representation of no. of login failures during Brute force attack

Consider a scenario in which the *time interval* between login failures is *very small* and *no. of login failures* is *very high*. From Table 1, we can conclude that the possibility of such a scenario being detected as an intrusion is *very high*.

Table 2 shows that the FAM table for Denial of Service attacks in the form of a matrix. Rows in this table represent the no. of execution of a loop and columns represent the time interval between occurrences of such loops.

Table 2. Fuzzy Associative Memory for a DoS attack

t ***x***	***VS***	***S***	***έ***	***H***	***VH***
VS	VS	VS	S	έ	έ
S	VS	έ	S	S	έ
έ	H	H	έ	έ	S
H	VH	H	H	έ	VS
VH	VH	VH	H	VS	VS

The *time interval* for execution of loop (t) is taken in X axis as a normalized value. The degree of membership (DoM) is taken in the Y axis. The min-max normalization scheme is used to normalize the time interval between execution of loops to a common range i.e., between 0 and 1. Figure 7 and 8 show the *time interval* and *no. of execution of loops* being assigned to the linguistic variables (very small, small, medium, high and very high), respectively.

Consider a scenario in which the time duration for execution of *for* loop is "Small" and No. of execution of *for* loop is "Small". From Table 2, it can see that the output "intrusion" is "Medium". Figure 9 shows the output of Fuzzy Intrusion System.

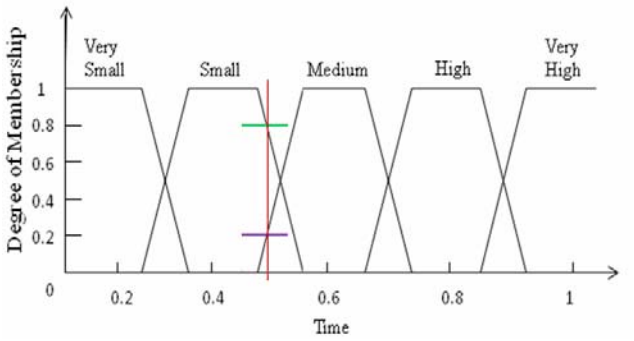

Figure 7. Linguistic representation of the time interval between loops for a DoS attack

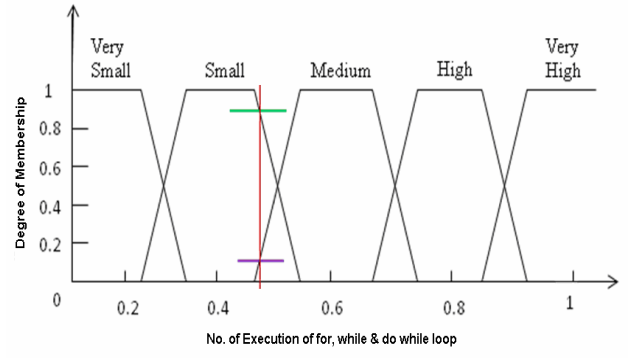

Figure 8. Linguistic representation of the number of execution of loops for a DoS attack

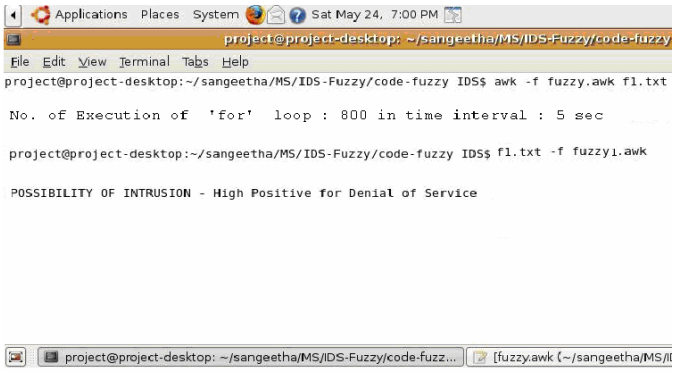

Figure 9. Output of the fuzzy component of the Intrusion Detection System

C. ALGORITHM FOR FUZZY INTRUSION DETECTION SYSTEM

The following algorithm presents the step in Fuzzy Intrusion Detection System.

Step 1: Let x = *set of number of login failures*
t= *time interval*

Step 2: x = normalization of (x)= (x-*min*)/(*max*-*min*)
where,
min is the minimum value of x
max is the maximum value of x

Step 3: Give x and t to FCM to select the appropriate fuzzy rules (Refer Table 3) from FAM table which has the following format:

IF condition AND condition THEN consequent

where, *condition* is a complex fuzzy expression that uses fuzzy logic operators (Refer Table 4), *consequent* is an atomic expression.

Step 4: Perform Mean of Maxima defuzzification

$$(D_{MM}) = \text{sum } \Sigma x_i / |X|$$

where, x_i belongs to X

Table 3 Fuzzy rules for detecting intrusions

Rule No.	***Rules***
Rule 1	If (*x*==*very small*) AND (t ==*very small*) THEN (I==*very small*);
Rule 2	IF (x==*very small*) AND (t ==*high*) THEN (I==*small*);
Rule 3	IF (x==*medium*) AND (t ==*high*) THEN (I==*high*);
Rule 4	IF (x==*very high*) AND (t ==*very small*) THEN (I==*medium*);
Rule 5	IF (x==*very high*) AND (t == *very high*) THEN (I==*very high*);

Table 4 Fuzzy logic operators

Logical Operator	**Fuzzy Operator**
x AND t	$min\{x, t\}$
x OR t	$max\{x, t\}$
NOT x	1.0 - x

Several methods are available in the literature for defuzzification. Some of the widely used methods are centroid method, centre of sums, and mean of maxima. In mean of maxima defuzzification method, one of the variable value for which the fuzzy subset has its maximum value is chosen as crisp value. According to the FAM table, the defuzzification graph is obtained and is shown in Figure. 10.

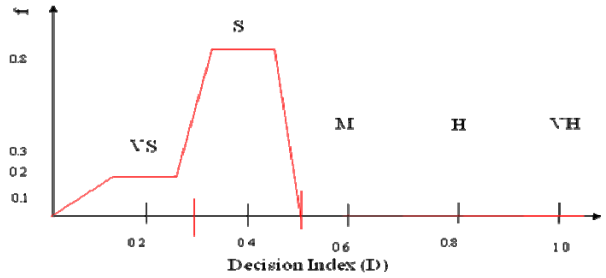

Figure. 10 Defuzzification

$$Defuzzification = \frac{\sum_{x_i \in M} x_i}{|M|} = \frac{(0.3+0.5)}{0.2} * 100$$

$$= 40\%$$

The defuzzification value thus calculated for Brute Force attack is 40%. In many situations, for a system whose output is fuzzy, it is easier to take a crisp decision if the output is represented as a single scalar quantity. For this reason, defuzzification value is calculated. Based on the defuzzification value, decision is taken if the traffic contains intrusive pattern or not.

The output of the rule-based intrusion detection module is non-intrusive for few attacks such as DoS, login failures. In DoS attack, instead of having infinite loop, the intruder will execute the loop for larger number of times. There is a bigger class of attacks which doesn't have a clear rule entry in the rule base can also be detected. These patterns are checked by the fuzzy IDS for a possible attack. Fuzzy Cognitive Mapping is used to capture different types of intrusive behavior as suspicious events.

IV. RESULTS AND ANALYSIS

The objects in each of the protocol field that are to be searched is plotted in Figure. 11. It is observed that if the number of objects to be matched in each protocol field is increasing the Response time increases linearly.

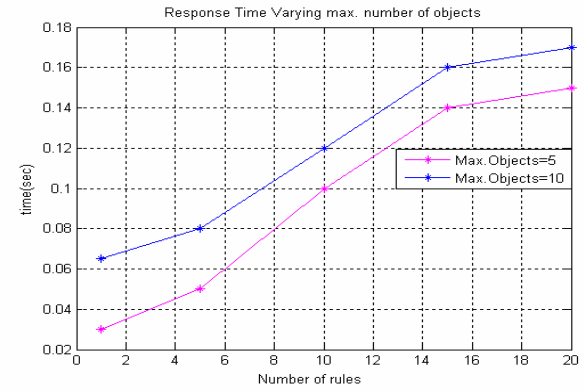

Figure 11. Response time vs. Rules with different number of maximum objects for each protocol field

But the response time tends to saturate after a specific number of rules. This is because it is expected that the rules contain some common objects which are to be checked once thus improving the response time. Figure.12 shows the detection rate with various components of IDS. From the figure. 12, the detection rate increases by combining HTTP header and payload (HTML and Scripts).

Figure 13 shows the comparison of Fuzzy based Misuse Detection and Regular Misuse Detection for various attacks. It shows the detection rate of fuzzy based misuse detection is high when compared to the regular misuse detection for some attacks such as Dos, Brute Force, Directory Traversal Attacks.

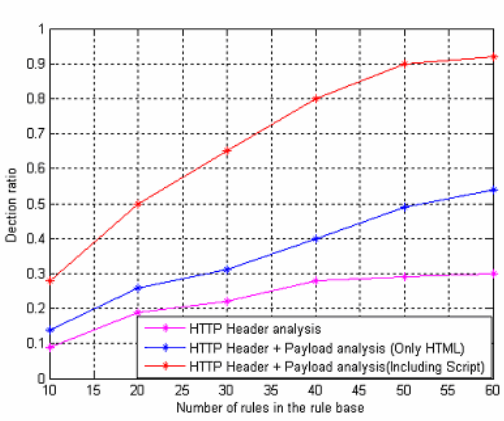

Figure 12. Detection Ratio with various component of IDS

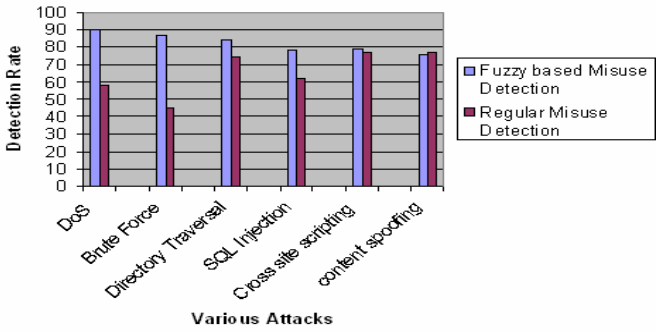

Figure 13 .Comparison of Fuzzy based Misuse Detection and Regular Misuse Detection

V. CONCLUSION

The rule-based semantic intrusion detection system proposed in this thesis has an efficient memory usage since the amount of memory needed for working of the IDS depends on the rule table size. A fuzzy component that is added to this rule based semantic IDS as proposed in this thesis uses Fuzzy Cognitive Mapping (FCM) in order to have an accurate prediction. Thus, the system proposed in this thesis namely Fuzzy aided Application layer Semantic Intrusion Detection System draws advantages from two different concepts. The semantic rule base keeps the rules updated for detecting newer intrusions by semantically matching the patterns. The Fuzzy component contributed to improving the detection rate by scanning through the traffic for attacks which goes undetected by a typical rule based IDS. The results show better performance in terms of the detection rate and the time taken to detect an intrusion. The semantic rule base can be appended with more number of semantic parameters by way which improving the accuracy of attack detection of the system is possible. Also that, more number of application layer protocols like FTP, SMTP, etc can be considered for implementation and the performance of the concept of application layer semantic intrusion detection can be validated with these protocols.

VI. REFERENCES

[1] Abbes T., Bouhoula A. and Rusinowitch M. (2004), 'Protocol Analysis in Intrusion Detection Using Decision Tree', In the Proceedings of International Conference on Information Technology: Coding and Computing (ITCC' 04), IEEE.

[2] Ambareen Siraj, Susan M. Bridges and Rayford B. Vaughn (2001), 'Fuzzy Cognitive Maps For Decision Support In An Intelligent Intrusion Detection System', In the proceedings of 20th International Conference of North American fuzzy information (NAFIPS), vol. 4, pp.2165-2170.

[3] Brubaker D. (1996), 'Fuzzy cognitive maps', EDN access.

[4] Carvalho J.P. and Jose A.B. Tome (1999), 'Rule-based fuzzy cognitive maps and fuzzy cognitive maps – a comparative study', In the proceedings of the 18th International conference of the North American fuzzy information (NAFIPS), pp. 115-119.

[5] Sangeetha S., Vaidehi V. and Srinivasan N. (2008), 'Implementation of Application Layer Intrusion Detection System using Protocol Analysis', Proceedings of International Conference on Signal Processing, Networking and Communications, ICSCN 2008, pp. 279-284.

[6] Cgisecurity.com (2002), William Bellamy Jr., 'TCP Port 80 - HyperText Transfer Protocol (HTTP) Header Exploitation'.

[7] Hallaraker O. and Vigna G. (2005), 'Detecting malicious JavaScript code in Mozilla', In the Proceedings of the 10th International Conference on Engineering of Complex Computer Systems (ICECCS 2005), pp. 85-94.

[8] Krugel C. and Toth T. (2003), 'Using decision trees to improve signature-based intrusion detection', In the Proceedings of the 6th International Workshop on the Recent Advances in Intrusion Detection (RAID'2003), LNCS Vol. 2820, pp. 173-191.

[9] Susan M. Bridges, Rayford B.Vaughn and Ambareen Siraj (2002), 'AI Techniques Applied to High Performance Computing Intrusion Detection', Proceeding of the Tenth International Conference on Telecommunication Systems, Modeling and Analysis, Monterey CA, Vol. 2, pp. 100-114.

[10] Eckmann S.T., Vigna G. and Kemmerer R.A. (2000), 'STATL: An Attack Language for State-based Intrusion Detection', In the Proceedings of the ACM Workshop on Intrusion Detection Systems, Athens, Greece.

[11] Lee T.B., Fielding R. and Frystyk H. (1996), 'RFC 1945 – Hypertext Transfer Protocol – HTTP/1.0'.

[12] Roesch M. (2006), 'Snort Users Manual', www.snort.org.

[13] Sangeetha S., Vaidehi V. (2010), 'Fuzzy aided Application Layer Semantic Intrusion Detection System-FASIDS', Proceedings of International Journal of Network Security and its Application (IJNSA April 2010), Vol.2 pp. 39-56.