An Approach for Improving Performance of a Packet Filtering Firewall Based on Fuzzy Petri Net

Ali A. Ali, Saad M. Darwish, and Shawkat K. Guirguis

Abstract—**With the rapid development of Internet, the security of networks becomes an important issue today and monitoring network traffic is necessary for realizing different purposes such as system performance, network debugging and/or information security. As a major measure to implement enterprise security, firewall technique ensures the security of local networks. Traditional firewall technologies have their own weaknesses in architecture, configuration, monitoring and management that affect to firewall performance. Furthermore, it lacks to deal with vague and uncertainty associated with filtering packets from outside. Architecture of a new kind of firewall, intelligence firewall is presented in this paper. The main contribution is to utilize Fuzzy Petri Net as a tool for modeling discrete event systems characterized by an imprecise knowledge. The graphical power of Petri Nets makes the packet filtering model easy to design, test, improve and maintain. Another contribution is to present 2-level fuzzy filtering algorithm to enhance ordering of filtering rules list that permits us to model the dynamic behavior of monitoring system concerning uncertainty associated with packet filtering. Experimental results for local network are given, which show the effectiveness of the suggested approach and demonstrate the enhancement of the firewall sensitivity against the risk coming from network traffics.**

Index Terms—**Firewall, fuzzy petri net, packet filtering, access control list (ACL).**

I. Introduction

With the advancement of computer technology and the wide spread use of computer networks, the security of internal network against attacks, illegitimate traffics and unauthorized access can be crucial to the success of the entire business operation. As firewalls are core element in network security, over the years, firewalls technology has become a major area of research in network security. A firewall is a combination of hardware and software used to implement a security policy governing the flow of traffic between two or more networks [1]. In its simplest form, a firewall acts as a security barrier to control traffic and manage connections between internal and external network hosts. Connections and service requests are either accepted or rejected based on a set of rules defined by the network administration security policy [2].

Firewall have many abilities like defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, provides a location for monitoring security-related events, audits and alerts action also can be implemented by firewall system [3]. Furthermore, it can serve as the platform for IP security, and can be used to implement virtual private networks. However firewall has number of limitations, including inability to protect against attacks that bypass the firewall, may not fully protect against internal threats, such as a disgruntled employee, also an improperly secured wireless LAN may be accessed from outside the organization [4]. Generally the continuous growth of the Internet, coupled with the increasing sophistication of attacks is placing further demands and complexity on firewalls design and management.

A firewall type varies and ranges from a packet filtering and circuit-level to a proxy service [3]. Proxies or application-level firewalls are CPU intensive and considered to be the most secure type of firewalls, but they incur a significant performance penalty. The penalty arises because a new process must be started each time a user starts a new session. These types of firewalls work at the application layer and are protocol specific [5]. Circuit-level firewalls are another type of firewalls work at the session layer and provide more general type of security. These firewalls act as relays for TCP connections. They intercept TCP connections being made to a host behind them and complete the handshake on behalf of that host and determine the legitimacy of a requested session by monitoring the handshake between packets. The circuit level firewalls can hide the network from the outside world and also restrict the session rules to known computers. Typically, cost of circuit-level gateways less than other forms of firewalls protection. Disadvantage of this kind is that many packets can go undetected due to more general things in consideration of filtering the packets [6].

To start with the network security, a packet filtering based firewall is the way to go. It works in the network layer and considers the common core component of any network monitoring tool, which processes every packet header and passes those packets according to filter's rules present in the access control list [3]. In this type of firewall there is no concern of applications, therefore focus is on the individual packets [7]. The information used for classifying packets is usually contained in distinct header fields in the packet, which are protocol field, source IP, source port, destination IP, destination port, TCP flags, Internet Control Message Protocol (ICMP) type and ICMP code [5].

A Packet filtering firewall is known as less secured one when compared with its counterparts because of its operation in low level devices. A weakness of packet filtering is that it

Manuscript received July 5, 2014; revised November 24, 2014.

Ali A. Ali is with Iraqi Commission for Computers and Informatics, Department of Computer, Ministry of Higher education and Scientific Research, Iraq (e-mail: ali_rq88@yahoo.com).

Saad M. Darwish and Shawkat K. Guirguis are with the Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, 163 Horreya Avenue, El-Shatby 21526, P.O. Box 832, Alexandria, Egypt (e-mail: saad.saad@alexu.edu.eg, Shawkat_g@yahoo.com).

DOI: 10.7763/JACN.2015.V3.144

pretty much trusts that the packets themselves are telling the truth when they say who they're from and who they're going to. Another weakness of packet filtering is that it examines each packet in isolation without considering what packets have gone through the firewall before and what packets may follow, in other words, although network traffic contains huge of events and lot of useful information, most of the current packets filtering techniques exploit the characteristics of filtering rules but they do not consider the traffic behavior in their optimization schemes [3], [8]. This is the main problem of this paper and our proposal suggests a solution to cope with this weakness point.

In spite of these weaknesses, packet filtering firewall have several advantages that explain why they are commonly used. Packet filtering is very efficient, it holds up each inbound and outbound packet for only a few milliseconds while they look inside the packet to determine the (source, destination) ports and addresses. In contrast, other firewall techniques have a more noticeable performance overhead [4]. Packet filtering is almost completely transparent to users and applications. The only time a user will be aware that a packet filter's firewall is being used is when the firewall rejects packets. Other firewall techniques require that clients and/or servers be specially configured to work with the firewall. In general, packet filtering technologies are fast, inexpensive compared to the other technologies and development cost is very less [7]. Most routers include built-in packet filtering. These reasons are the motivation of this research.

Since firewalls need to filter all the traffic crossing the network perimeter, they should be able to sustain a very high throughput, or risk becomes a bottleneck. Firewall packet matching can be viewed as a point location problem: Each packet (point) has five fields (dimensions), which need to be checked against every firewall rule in order to find the first matching rule. Generally, packets are various and variety, so there is large and different number of filtering rules which are ridding these packets. Most rules not aware about traffic behaviors of these packets, so the ability of optimize filtering and threat detection needs to improve. This makes the trend to offers approaches and tools that have the ability to cope with uncertainty of packet filtering cases with the aim of building a model of an "intelligent" and adaptive firewall that changes its behavior depending on the dynamical conditions of networks and computational resources [9].

Fuzzy logic technique has emerged as one of an essential methodology to deal with uncertainty and working with vague, imprecise or ill-defined systems. The central idea of fuzzy logic is to model the human way of reasoning in situation of uncertainty and imprecise concepts. In our viewpoint, present firewall technology will be greatly improved by the application of fuzzy logic control techniques. Some scholars [10] exploit a fuzzy control approach for improving present firewall architectures. Taking into account not only network packets' contents but also connection status information to gain filtering decisions, it's possible to make firewall's behavior adaptive to increase operating systems' immune to attacks.

It is well known that any model in network system community needs a tool to represent it. However, UML model is difficult to reflect the concurrency and consistency of constraint rules between objects; therefore impossible to completely demonstrate the dynamic behavior characteristics of a system. Further, UML cannot be directly expressed and analyzed by mathematical tools. On the other hand, Petri Net (PN) modeling technology uses graphical element place and transition to clearly describe the internal interactions of a system. Furthermore, Petri Net has mature mathematical analysis methods [11].

It has shown that Petri Net theory can be used to improve the representation and analysis of the dynamic model of a system that is specified using UML, making the design engineer more confident that the model accurately represents the system. Due to Petri Nets inability to face with systems working on obscure data and continues events, the trend to adopt an extension of PN called a Fuzzy Petri Nets (FPN) that is a combination of fuzzy set theory and Petri Net concept. FPN is a tool for the representation of uncertain knowledge about a system state and has the ability to describe fuzzy event-condition-action rule in a good vision [12].

This paper presents a new packet filtering firewall approach that utilizes traffic behavior of incoming packets to assess the threat from the outside and invest this assessment to enhance packet filtering process. The major contribution is to improve the firewall efficiency and performance by adapting the FPN concept as a technique employed to cope with uncertainty filtering states; this lead to deal with various and variety types of packet to increase the ability of firewall to detect and prevent illegal usage or intruder.

The rest of the paper is organized as follows: Section II describes some of recent related works. Section III presents some background and theory used in our approach. The detailed description of the proposed system has been made in Section IV. In Section V, the simulation results and discussions on the syntactic dataset are given. Finally conclusions are drawn in Section VI.

II. Literature Survey

Many attempts have been done to prompt packet filtering technologies in order to meet objective such as increasing security, performance, and quality of services. Filtering of packets mainly based on compare and match processes with a number of rules that are prior determined by rule designer. This seems to be as main filtering problem, which is hard to solve due to its variety and dynamically change of packets and rules [13]. Current studies tend to find appropriate solutions to solve this problem. Some authors [14] adopted the BDD (Binary decision diagrams) as an approach to improve packet filtering. Implementation of such packet filter using BDD gives more advantages in terms of memory usage and look up time. Researchers in [9] described an early filtering decision technique that reduces the packet matching cost. Their technique utilized Internet traffic characteristics coupled with a special carefully tuned representation of the policy to generate early defense policies. They used Boolean expressions built as BDD to represent relaxed versions of the policy that are faster to evaluate. Moreover, it is guaranteed that the technique will not add an overhead in filtering time.

A rule in ACL (Access Control List) has been addressed in as another packet filtering approach. Conventionally, the rules are applied in the sequential order on the packets arriving at the firewall. This results in poor efficiency due to

delay incurred in forwarding the packets through the firewall. The method suggested in [15] considers the usage frequency of various rules present in the ACL; these rules have been rearranged based on important parameter using clustering and indexing technique. Furthermore, the rules are prioritized based on their usage. It is observed that their method results into significant improvement in packet matching time in packet filters.

Authors in [16] introduced a contribution consists of designing and developing an intelligent system in order to help the security administrator to exploit, manage and analyze the firewall log files content. The firewall log files trace all incoming and outgoing events in a network. Their content can include details about network penetration attempts and attacks. P. Eronen and J. Zitting [17] presented a tool that helps administrators in analyzing firewall rules. Their presented tool is based on constraint logic programming (CLP) that allows the user to write higher level operations for, e.g., detecting common configuration mistakes.

The work in [18] proposed a protocol-independent Distributed Denial of Service (DDoS) defense scheme that is able to dramatically improve the throughput of legitimate traffic during DDoS attacks. The algorithm leverages on the "attack graph" information (such as whether or not a network edge is on the path from an attacker) obtained through IP trace back, and uses such information to preferentially filter out packets that are more likely to come from attackers. The system works by performing "smart filtering": dropping DDoS traffic with high probability while allowing most of the legitimate traffic to go through. This clearly requires the system to be able to statistically distinguish legitimate traffic from DDoS traffic.

In addition a new method based on Hierarchical Colored Petri Nets (HCPN) was proposed in [19] that concentrated on one technological aspect of providing communications security, firewall technology. HCPNs are well suited for modeling concurrent, distributed systems in which regulated flows of information are significant, such as firewall systems that enforce access control policies on network packets. The authors demonstrated with several examples how firewalls can be modeled. They outlined how simulations of such models can facilitate testing, performance analysis, and interactive design exploration. This tool can serve as the basis for formal analysis techniques available through applied Petri Net theory.

Swarm intelligence has been studied extensively in recent years and a number of developments have been proposed. In [20] an Ant Colony Optimization (ACO) approach for filtering the incoming packets in a network by matching the rules in a rule set is proposed to build intelligent firewall. The ant agent makes a decision about the rule position in the rule set matching with the compared field of the incoming packet based on its attractiveness (energy value) towards the solution. Another method includes a preprocessing algorithm for the rule organizing and removing of redundant rules. This method gives the capability of auto-generation of rules for packets that do not match with any rule by the use of a rule-based expert system that employs the previously defined rules in deducing new rules [21]. A short summary of the existing literature on packet filtering can be found in [3].

Despite of the variety of existing packet filtering techniques, we believe that there are still many paths to be explored in the field of intelligence firewalls. This research presents an intelligent packet filtering model that invests FPN properties for building a system with 2-level of filtering stage. First level of packet filtering aims to enhance filtering protection by using risk level to detect and prevent the attacker' packets, while the objective of the second level is to increase filtering performance depending on rating of accept and reject packets. FPN theory provides means to manipulate imprecise and vague information within packets and is used for fuzzy knowledge representation and reasoning. In fact, by implementing the FPN model, major features such as correctness, circular rules, consistency, and completeness checking, can also be employed.

III. Background

This section defines terminology used throughout the paper and gives a working definition of the term firewall technology

A. Access Control Lists (ACL)

An ACL, a sequence of rules intended to implement set of security objectives, is commonly setting by the network administrator that decides the access rights for different users and network resources. From an outside the network, the packet should be accepted or rejected according to set of ACL' rules [15]. The main task of packet filters is to categorize packets based on the filtering rules. A typical access control list in which many filtering rules are defined is shows in Table I.

TABLE I: A Typical Access Control List

No.	Source IP	Destination IP	Source port	Destination port	Type of protocol	Action
R1	202.118.101.11	192.168.90.1	*	*	ICMP	Allow
R2	202.118.176.2	192.168.78.3	*	*	UDP	Allow
R3	202.118.176.5	192.168.60.23	*	*	TCP	Block

Each rule has five fields or may be more and each field contains a single value or a range of allowable matches. IF the absence of any field such as address or protocol type, the rules will match a packet with any such values fields that are present. The interpretation of ACL is that the rules are considered as being processed in sequential order from the top, each incoming packet is tested against the first rule; if it matches, it passed or blocked accordingly and no further rules are considered; otherwise it is tested against the second rule, and so on. There is an implicit deny all rules at the end of each ACL to block all packets [8].

B. Fuzzy Petri Net

The classical Petri Net is a kind of directed graphs' which consist of places, transitions, directed arcs, tokens. Directed arcs connect places to transitions or transitions to places [11].

A transition is activated when every place in the precondition of the transition is fulfilled. Tokens, which reside in the places of a Petri Net, are used to define the execution of a Petri Net. The presence or absence of a token in a place can indicate whether a condition associated with this place is true or false, and the number and position of tokens may change during the execution of a PN. In general, a PN can be represented by a transition together with an input place and an output place. The notations of PN are shown in Fig. 1.

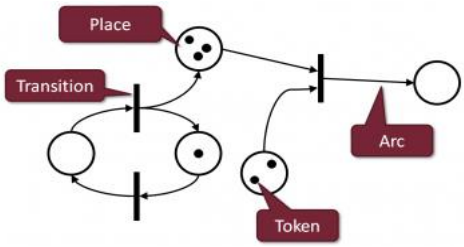


Fig. 1. Petri net notation.

FPN is a combination of Fuzzy logic and Petri Net. It described as a Petri Net that uses fuzzy logic rather than Boolean logic. FPN are used for fuzzy knowledge representation and reasoning. The fuzziness concept can be incorporated in Petri Nets by applying a fuzzy reasoning mechanism over the Petri Nets structure [12]. Generally a FPN is capable of modeling fuzzy production rules (of type if d_j then d_k with Certainty Factor (CF) μ_i). Each place may contain a token associated with a truth value of a proposition, which are quantified as numbers in the unit interval. Each transition is associated with a certainty factor taking values from the unit interval. Formally a model of FPN is defined as a 7-tuple $N_f(P, T, D, I, \alpha, \beta)$ where [22]:

- $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places;
- $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions;
- $D = \{d_1, d_2, \ldots, d_j\}$ is a finite set of propositions, where I: $P \rightarrow T$ is an input mapping, $O : T \rightarrow P$ is an output mapping, and $f = \rightarrow [0, 1]$ is an association mapping. Furthermore $\alpha : P \rightarrow [0, 1]$ and $\beta : P \rightarrow D$. $P \cap T \cap D = \phi$, $|P|=|D|$.
- A token value in place $p_i \in P$ *is* denoted by $\alpha(p_i) \in [0,1]$ If $\alpha(p_i) = y_i, y_i \in [0,1]$ and $\beta(p_i) = d_i$ then this states that the degree of truth of proposition d_i is y_i. A transition t_i is enabled if for all $p_i \in I(t_i), \alpha(p_i) \geq \lambda$, where λ is a threshold value in the unit interval. If this transition is fired, then token are removed from its input places and a token is deposited to each of its output places. The truth value of the output tokens are generally computed through some aggregation function τ i.e.

$$y_K = y_J \; \tau \; \mu_i, or \; y_k = \tau(I(t_j), \mu_i), y_k \in O(t_j) \quad (1)$$

In theory, PNs and FPN are the same computational power, but FPN have much more modeling power because they have better structuring facilities. Logic expressions and functions can be built using fuzzy logic for all PN objects (Transitions, places or arcs). FPN can effectively analyze concurrent systems by verifying the safety rules and standards for traffic operations, also it uses a graphical presentation that is easy to understand, easy to modify because of its modularity. Reader looking for more information about FPN can refer to [12]. In our approach we adapt the fuzzy logic for modeling PN transition object that concentrates on if–then fuzzy production rules realized by truth values to describe ambiguity of packet's movement.

IV. Methodology

Fig. 2 shows the general data flow diagram of the proposed 2-level fuzzy packet filtering system, which comes with better security filtering performance. The system utilizes FPN as graphical method to describe the fuzzy logic control of packets movement through the firewall. Two levels of fuzzy have been applied to filter packets, first level led to determine the level of threat that is embedded with packets from Internet, and the second level used to rearrange ACL by determining the rating of acceptance and rejection of packets. The following subsections discuss each level in details.

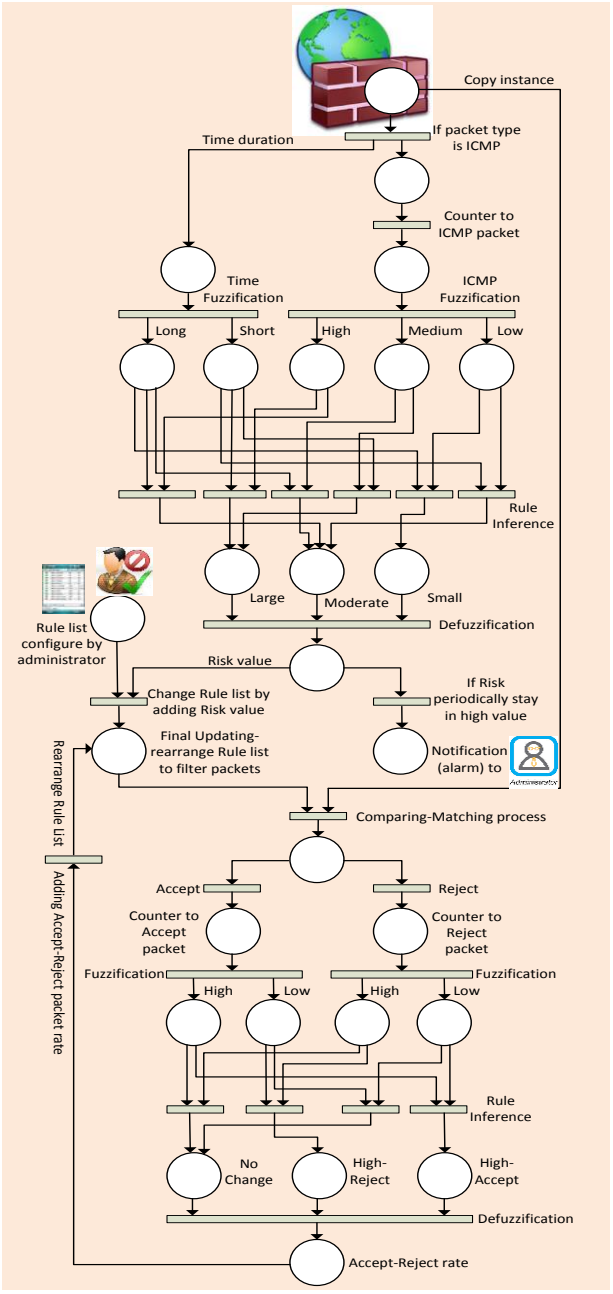


Fig. 2. Proposed model of 2- level fuzzy packet filtering.

A. Level 1- Fuzzy Filtering

This level relies on capturing and classifying all arrival packets depending on the information related with each packet, such as IP address, packet time and protocol type to simulate and trace the packet movement. In our approach, packet is represented by a token in FPN place, and packet's operation is exemplified by FPN transition that is responsible to move packet from one place to another [12], [22]. Once the packet is captured by a gate; it moves to place where checking and matching with ACL is done, in addition an instant copy of this packet is moved to the traffic analysis part to extract the packet's parameters (features) like number of packets of ICMP protocol coming through time period. These two parameters (packets counting and time period) embody the inputs to fuzzy logic engine that is used to produce the risk level. This risk level represents threats coming through movement of packets from untrusted sources.

The proposed system deals with ICMP protocol because ICMP has been used in many phases of an attacker's advance in a system compromise. ICMP flood attacks (with characteristics similar to that of the Acknowledge flood) are traffic-based attacks that use heavy traffic to bring high loads to the servers, which will affect the server's normal services [23]. Furthermore ICMP has been used for exploiting systems as well as in certain instances as a covert channel for attacker's communication. This system can also deal with attack's techniques that use other protocols like TCP SYN and UDP flood [24]. User Datagram Protocol (UDP) flooding occurs when an attacker sends IP packets containing UDP datagrams with the purpose of slowing down the victim to the point that it can no longer handle valid connections. While, the striking feature of SYN-Flood attacks is that the attackers send a large number of TCP SYN request packets with forged source IP addresses. This results in the server side consuming large amounts of resources in order to maintain a very large list of half-open connections, eventually leading to the server running out of resources and becoming unable to provide normal services.

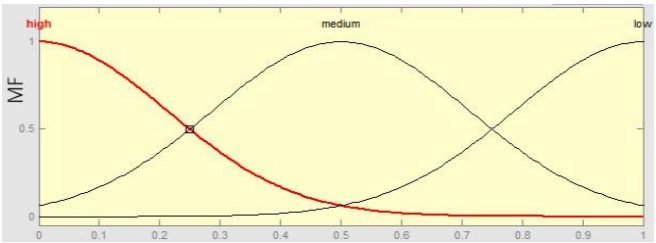
The rationale of chosen the number of ICMP echo-request packets p_{no} and packets arrival time interval p_t is that they are simple and suitable in most of the attacks defense cases, and especially when we have a large number of entire packets. To satisfy the requirement of the membership grade function (MF) used in proposed fuzzy system, measures for feature vectors need to be transformed into range [0, 1] with Gaussian normalization method as stated in [10]. Fuzzy Logic (FL) is probably the most efficient and flexible method available for packet filtering to manage the combination of measurements through their degrees of uncertainty. FL is a theory that allows the natural descriptions, in linguistic terms, of problems to be solved rather than using numerical values. The FL system consists of [25] (i) fuzzifier that takes input values and determines the degree to which they belong to each of the fuzzy sets via MFs; (ii) fuzzy inference system that defines a non-linear mapping of the input data vector into a scalar output, using fuzzy rules and (iii) defuzzifier that maps output fuzzy sets into a crisp number.

In this research, the attempt is to model the uncertainty of packets features through a fuzzy model. Here, a two-input single-output fuzzy system is used, which is given by $f: u \subset R^n \rightarrow V \rightarrow R^n \; where \; U = U_1 \times U_2$ is the input space and V is the output space. Three fuzzy variables including 'low', 'medium' and 'high' are used to describe feature of p_{no} and two fuzzy variables including 'Long ' and 'Short ' are used to describe the feature of p_t. Their respective MFs (μ) are Gaussian function (see Fig. 3a and Fig. 3b) [12]. All of membership function' parameters are numerically specified based on the experiences to estimate the risk level impending through packet traffic.

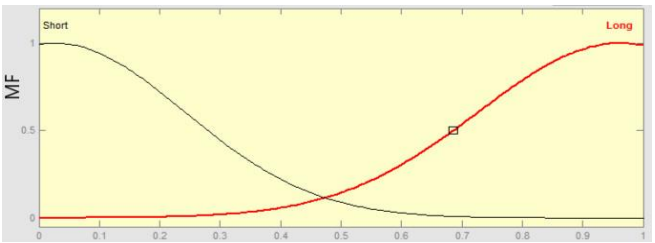
Once the system acquires the fuzzy descriptions of the packets' features, the rule base (fuzzy reasoning) can be built to make an inference of their similarity. Fuzzy reasoning, which is formulated by group of fuzzy IF–THEN rules, presents a degree of presence or absence of association or interaction between the elements of two or more sets. In the proposed system, reasoning is carried out through the following rules:

- **Rule 1 IF** ICMP-echo-rate = high and time duration = long; **Then** Risk = moderate
- **Rule 2 IF** ICMP-echo-rate= medium and time duration **= long; Then Risk = moderate**
- **Rule 3 IF** ICMP-echo-rate = low and time duration = long; **Then** Risk = small
- **Rule 4 IF** ICMP-echo-rate = high and time duration = short; **Then** Risk = large
- **Rule 5 IF** ICMP-echo-rate = medium and time duration = short; **Then** Risk = large
- **Rule 6 IF** ICMP-echo-rate = low and time duration = short; **Then** Risk = moderate

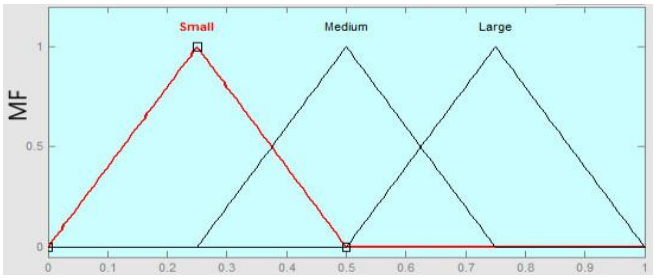
The six rules altogether deal with the weight assignments impliedly in the same way as what humans think. The fuzzy inference processes all of the six cases in a parallel manner, which makes the decision more reasonable. The output of the fuzzy system is the risk level r_l that characterizes the risk embedded in packets traffic and it is also described by three fuzzy variables, including 'small', 'moderate' and 'large' with triangular MFs illustrated in Fig. 3c. The outputs of fuzzy values are then defuzzified to generate a crisp value for the variable. The most popular defuzzification method is the centroid, which calculates and returns the center of gravity of the aggregated fuzzy set. Reader looking for more information regarding fuzzy logic can refer to [22].



(a) Input variable "number of ICMP-request" packets"



(b) Input variable "packets arrival time"



(c) Output function "risk level"

Fig. 3. Membership function of (a) p_{no} (b) p_t (c) r_l.

B. ACL Configuration and Updating

Packet filtering makes it possible to control the traffic, monitor bad behavior users or programs, and filter ill dataflow to protect the whole LAN. As mentioned earlier that the network administrator is responsible for configuring and writing the first setting of the ACL. Each rule has an action, if a packet matching a rule. The actions indicate whether to block ("deny") or forward ("allow") the packet to a particular interface [14]. To improve the performance of the packet filtering, the proposed system utilizes obtainable risk level to change the action of ACL rules to reflect the valid and trustable address list and determine the authentication level of all IP address. If the risk value is more than the critical degree (threshold $\delta \geq 0.75$) that is determined by the administrator then the rules' action of black list change from accept to deny; therefore packets that supposed to be accepted will be rejected. While the actions of white list not change even the risk value beyond the critical degree, see Table II for demonstration.

TABLE II: ACCESS CONTROL LIST AFTER RISK VALUE

No.	Source IP	Destination IP	Source port	Destination port	Type of protocol	Risk Value	Action
R1	202.118.101.11	192.168.90.1	*	*	ICMP	0.80	Block
R2	202.118.176.2	192.168.78.3	*	*	UDP	0.85	Block
R3	202.118.176.5	192.168.60.23	*	*	TCP	0.20	Block

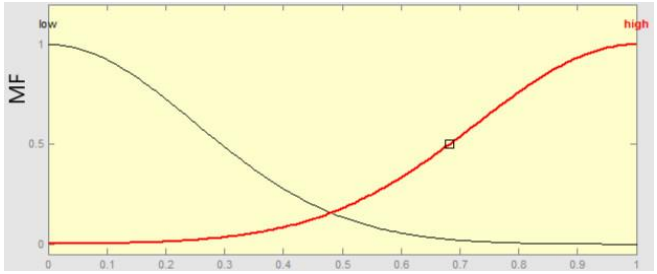
If the risk value periodically stays in increasing manner then a notification must be send to the network administrator to check the incoming sites and make a risk assessment. Typically, a black list must be updating to avoid consideration of some of trusted sites as harmless sites.

C. Level 2- Fuzzy Filtering

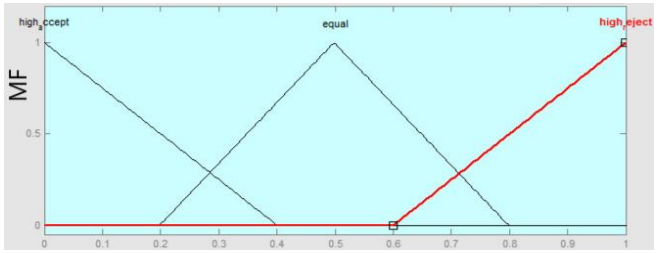
Typically, there are two sets of packets that associated with each firewall: the set of packets that are accepted by the firewall, and the set of packets that are discarded by the firewall [15]. Our system invests this fact to enhance packet filtering performance by adopting level -2 fuzzy filtering to monitor rate of packets' acceptance or rejection to reorder ACL rules with the aim of minimizing the time of rule matching [26]. Here, the attempt is to model the uncertainty of the rate of acceptance or rejection of packets through a fuzzy model. In this case, a two-input single-output fuzzy system is used. Two fuzzy variables including 'low', and 'high' are used to describe both counter of acceptance rate A_r and rejection rate R_r. Their respective MFs (μ) are Gaussian function (see Fig. 4a). The output of the fuzzy system is the calculated rate C_r that characterizes the rejection and acceptance rates in packets traffic and it is described by three fuzzy variables, including 'High reject', 'Equal' and 'High accept' with triangular MFs illustrated in Fig. 4b. All of membership function' parameters are numerically specified based on the experiences and experimental results to adjust ACL rules ordering. In the same manner, the proposed system reasoning is carried out through the following rules:

- **Rule 1 IF** A_r= high and R_r= low; **Then** C_r= high accept
- **Rule 2 IF** A_r = high and R_r = high; **Then** C_r = equal
- **Rule 3 IF** A_r = low and R_r = low; **Then** C_r = equal
- **Rule 4 IF** A_r = low and R_r = high; **Then** C_r = high reject

The outputs of fuzzy values are then defuzzified to generate a crisp value for the variable. Here, if C_r = high accept then all rules that have an action with 'allow' are rearranged and are moved at the beginning of the ACL, taking the highest priority to execute. Otherwise if C_r = high reject then all rules that have an action with 'block' are reshuffled and relocated at the beginning of the ACL, taking the highest precedence to fulfill, and as a result the rules that have an action with 'accept' are stabilized at the end of ACL.



(a) Input variable "acceptance rate, rejection rate".



(b) Output function "calculated rate".

Fig. 4. Membership functions of (a), A_r, R_r, (b) C_r.

In summary the proposed system has the ability to change rules' activities through two phases, at the first phase the rule's action may be changed according to the traffic behavior based on risk level; in the second phase sequence of rules may be altered dynamically to reflect the rules' highest priority depending on acceptance and rejection rates of packets. The system utilizes fuzzy logic to handle uncertainty associated with system's variables like risk level and accept rate to build a smart model that has the ability to cope with network traffic attacks. In the proposed model, a fuzzy technique was used to connect expert opinions with linguistic variables. These linguistic variables reflect the expert opinions more precisely.

V. Experimental Results

In order to test the efficiency and validity of the proposed algorithm, we implement the algorithm by MATLAB language and deposit the firewall rules in C# language. We built the model in a modular fashion and demonstrated how the hierarchical concepts of FPNs can be used to combine several mechanisms into a comprehensive firewall system.

A. Experiment Design

We adopt our firewall rules by editing the rule tables. The filtering fields consist of 5 categories: Source IP, destination IP, source port, destination port and protocol. The proposed system has been simulated in a (DELL) PC machine which has the following features: Intel (R) Core (TM) i5-2450M CPU @ 2.50GHz, and 4.00GB of RAM, 64-bit Windows 8 Pro. In the simulation, the proposed firewall packet filtering has been tested with the standard TCP, UPD, ICMP protocols. Different scenarios have been implemented to evaluate the proposed approach. In addition, this includes comparison between the proposed approach and traditional non-fuzzy filtering technique. The results of this evaluation will be discussed below.

B. Results and Analysis

In Fig. 5, the time needed for a set of data packets to pass through the firewall is compared before and after the optimization (reordering) of firewall rules table. Line 1 represents the time needed when rules are sequenced randomly and filtering fields are sequenced by our filtering technique in ascending order(from beginning of the ACL); Line 2 represents the time needed when rules are sequenced in an optimum (reordering) manner and filtering fields are sequenced by our filtering in ascending order. From the comparison between line 1 and line 2, it is obvious that the time needed for data packets going through the firewall after the optimization of filtering fields (Line 2) is about 1/3 of the time needed before the optimization (Line 1). Also the more the rules are, the more significant the result performs related to response time. The finding proves that the proposed algorithm decreases the number of rule-comparisons and improves the efficiency of firewall remarkably. The outcomes also demonstrate the system's ability to deal with large scale of rules set.

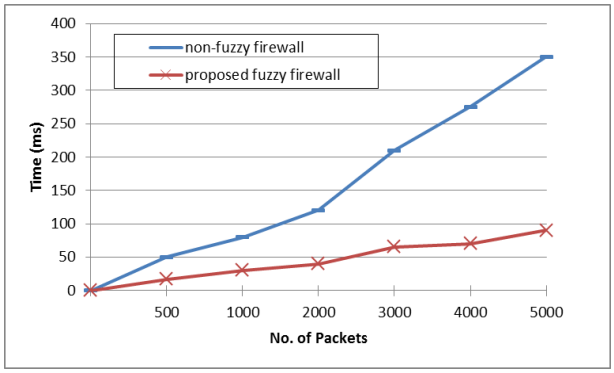


Fig. 5. Experiment results.

In experiment 2, the proposed solution's has been compared with the non-fuzzy firewall filtering technique. This comparison focuses on the packet's acceptance and rejection rates. Each user sends 1 legitimate packet (first case) and sends 3 malicious packets (second case). The victim will define the action (reject or accept) of the filtering rules based on the behaviors. In the first scene of the first case, Fig. 6 and Fig. 7 shows the comparison results of accept and reject rate according to different thresholds. In the second case we tested the reject rate for malicious packets and the results are shown in Fig. 8. According to these figures, the proposed system has high ability to accept legitimate packets and reject malicious packets. The experiments are entirely reasonable because the effects of fuzzy logic-based filtering is always better comparing with other filtering policies regarding uncertainty associated with the packet traffic behavior.

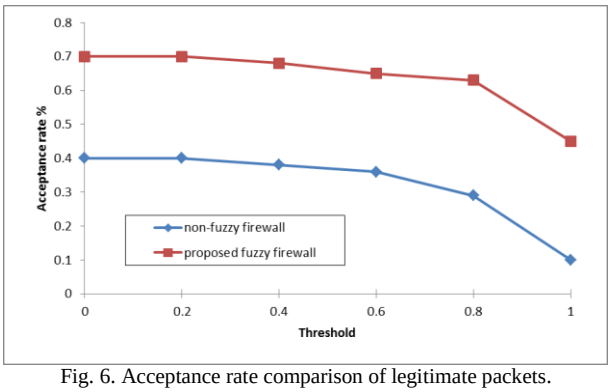


Fig. 6. Acceptance rate comparison of legitimate packets.

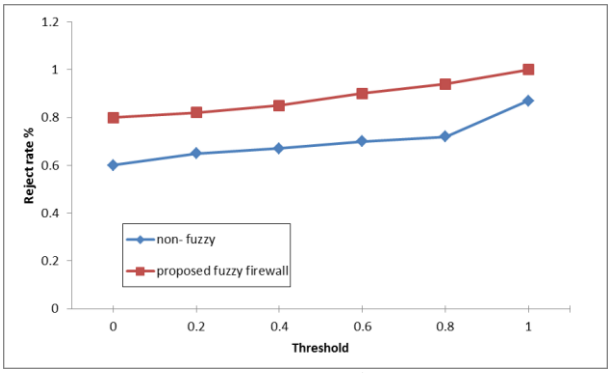


Fig. 7. Reject rate comparison of legitimate packets.

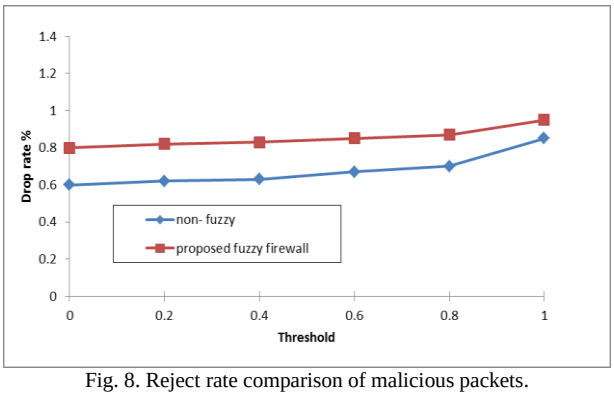


Fig. 8. Reject rate comparison of malicious packets.

VI. Conclusions and Discussions

Firewalls are a very important component of system security. Recently, the packet filtering optimization problem has received the attention of the research community for many years. Nevertheless, there is a manifested need for new innovative directions to enable filtering devices such as firewalls to keep up with high capability for preventing network traffic attacks. This paper presents a new approach for optimizing packet filtering in network security policies

based on online traffic statistics.

In our viewpoint, present firewall technology will be greatly improved by the application of fuzzy logic control techniques. It is the first effort in using Fuzzy Petri Net to design and optimize firewall rules sets. Both rule set based and traffic based optimizations are integrated in our firewall accelerating tool. The system is based on a formalism that uses Fuzzy Petri Nets to describe the functionality of mechanisms exploited by firewall technology. FPN provide us with a theoretical framework and means of description, composition, simulation, and analysis of firewall systems.

The novelty of proposed system is the use of two level of fuzzy filtering suitable for network traffic behavior to handle different levels of uncertainty related to packet contents in order to increase security and to improve the efficiency of firewalls through rules optimizations. Experts present their opinions with respect to specific criterions leading us to increase accuracy and reliability of the results. Our prototype implementation and experimental results have shown us a decent firewall system which can be deployed to the practical networking. It proves that our research results are promising when combined with other achievements of the rules optimization to further improve the efficiency of firewalls

In the future, it may be beneficial to investigate the question of which desirable properties of firewall systems can be expressed as dynamic properties, which in turn can be verified mechanically by FPNs. We conjecture that concentrating on invariants as an analysis technique is likely to be a rewarding strategy.

REFERENCES

[1] S. Acharya, J. Wang, and Z. Ge, "Simulation study of firewalls to aid improved performance," in *Proc. 39th Annual Symposium on Simulation*, USA, Apr. 2-6, 2006, pp. 18-26.

[2] A. El-Atawy, K. Ibrahim, H. Hamed, and E. Al-Shaer, "Policy segmentation for intelligent firewall testing," in *Proc. 1st IEEE International Conference on Network Protocols*, 2005, pp. 67–72.

[3] O. Nurika, A. Aminz, A. Rahman, and M. Zakaria, "Review of various firewall deployment models," in *Proc. International Conference on Computer and Information Science*, 2012, pp. 825–829.

[4] F. Bin-Hamid, "A study of technology in firewall system," in *Proc. Business, Engineering and Industrial Applications*, 2011, pp. 232-236.

[5] H. Mao, L. Zhu, and M. Li, "Current state and future development trend of firewall technology," in *Proc. 8th Int. Conf. of Wireless Communications, Networking and Mobile Computing*, 2012, pp. 1-4.

[6] D. Singh, R. Sharma, and T. Singh, "Enhancement of firewall filtering techniques," *International Journal of Emerging Trends and Technology in Computer Science*, vol. 2, issue 4, pp. 258-261, 2013.

[7] M. Alhabeeb *et al.*, "Preventing denial of service attacks in government E-services using a new efficient packet filtering technique," in *Proc. Int. Conf. of Parallel and Distributed Processing with Applications*, 2011, pp. 262–269.

[8] X. Yue *et al.*, "The research of firewall technology in computer network security," in *Proc. Int. Asia-Pacific Conf. of Computational Intelligence and Industrial Applications*, 2009, pp. 421-424.

[9] A. El-Atawy, E. Al-Shaer, T. Tran, and R. Boutaba, "Adaptive early packet filtering for defending firewalls against DoS attacks," in *Proc. Int. Conf. on Computer Communications*, 2009, pp. 2437–2445.

[10] E. Dadzie and A. Veselý, "User perception of security on social networking sites using fuzzy logic," *British Journal of Applied Science and Technology*, vol. 3, no. 4, pp. 714-734, 2013.

[11] J. Capek, M. Hub, and R. Myskova, "Basic authentication procedure modelled by petri nets," *International Journal of Computers and Communications*, vol. 4, no. 4, pp. 101-108, 2010.

[12] X. Li and F. Rosano, "Adaptive fuzzy petri nets for dynamic knowledge representation and inference," *International Journal of Expert Systems With Applications*, vol. 19, no. 3, pp. 235–241, 2000.

[13] Z. Liu and P. Chen, "Improved method of packet filtering," in *Proc. International Conference of Web Information Systems and Applications*, China, May 22-24, 2009, pp. 294-296.

[14] G. Pault, A. Pothnal, C. Mandalt, and B. Bhattacharya, "Design and implementation of packet filter firewall using binary decision diagram," in *Proc. the IEEE Students' Technology Symposium (TechSym)*, India, Jan. 14-16, 2011, pp. 17–22.

[15] U. Thakar, L. Purohit, and A. Pahade, "An approach to improve performance of a packet-filtering Firewall," in *Proc. IEEE International Conference of Wireless and Optical Communications Networks*, India, Sep. 20-22, 2012, pp. 1-5.

[16] H. Bensefia and N. Ghoualmi, "An intelligent system for decision making in firewall forensics", *International Journal of Digital Information and Communication Technology and Its Applications*, vol. 166, pp. 470-484, 2011.

[17] P. Eronen and J. Zitting. "An expert system for analyzing firewall rules", in *Proc. the 6th Nordic Workshop on Secure IT Systems*, November 2001, pp. 100–107.

[18] M. Sung and J. Xu, "IP trace back-based intelligent packet filtering: a novel technique for defending against internet DDoS attacks," in *Proc. 10th IEEE International Conference on Network Protocols*, USA, Sept. 2003, pp. 861-872.

[19] C. Schuba and E. Spafford, "Modeling firewalls using hierarchical colored petri nets," in *Proc. NATO Symposium on Protecting Information Systems in the 21st Century*, 1999, pp. 1-15.

[20] N. Sreelaja and G. Vijayalakshmi, "Ant colony optimization based approach for efficient packet filtering in firewall," *International J. of Applied Soft Computing*, vol. 10, no. 4, pp. 1222-1236, 2010.

[21] M. Hashemi *et al.*, "Intelligent IP packet filtering," *IFIP Advances in Information and Communication Technology*, vol. 86, pp. 521-533, 2002.

[22] H. Virtanen, "A study in fuzzy petri nets and the relationship to fuzzy logic programming," *Turku Centre for Computer Science*, vol. 162, pp. 1-30, 1995.

[23] Choudhary *et al.*, "Smurf attacks: attacks using ICMP," *Int. Journal of Computer Science and Technology*, vol. 2, no. 1, pp. 75-77, 2011.

[24] M. Bogdanoski, T. Shuminoski, and A. Risteski, "Analysis of the SYN flood DoS attack," *Int. J. Computer Network and Information Security*, vol. 8, pp. 1-11, 2013.

[25] Y. Rodin and I. Snitsyn, "Using fuzzy logic system for making decisions about information security in grid infrastructure," in *Proc. 3rd International Conference on High Performance Computing*, pp. 336-339, Ukraine, October 7-11, 2013.

[26] Z. Trabelsi, H. El-Sayed, and S. Zeidan, "Firewall packet matching optimization using network traffic behavior and packet matching statistics," *3rd IEEE International Conference of Communications and Networking*, Tunisia, Mar. 29-Apr. 1, 2012, pp. 1-7.

Shawkat K. Guirguis obtained the B.Sc. and M.Sc. degrees in computer science & automatic control, from Faculty of Engineering, Alexandria University, in 1981 and 1984 respectively with Grade: "Distinction with the degree of honor". In 1988, he obtained a Ph.D. degree in electronics & communication, from Cairo University, Co-Supervised by Imperial College of Science & Technology, University of London, U.K. Currently he is a professor of computer science and informatics, Department of Information Technology, Institute of Graduate Studies & Research (IGSR), Alexandria University, Egypt. His current research interests include network and information security, data mining and cloud computing.

Saad M. Darwish received his Ph.D. degree from the Alexandria University, Egypt. His research and professional interests include image processing, optimization techniques, security technologies, and machine learning. He has published in journals and conferences and severed as TPC of many international conferences. Since Feb. 2012, he has been an associate professor in the Department of Information Technology, Institute of Graduate Studies and Research, Egypt.

Ali A. Ali received his M.Sc. degree in computer science from Iraqi Commission for Computers and Informatics, Iraq in 2006. Currently he is a Ph.D. student in the Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Egypt. His research and professional interests include network communication and security technologies.