DEVELOPMENT OF SMART FIREWALL LOAD BALANCING FRAMEWORK FOR MULTIPLE FIREWALLS WITH AN EFFICIENT HEURISTIC FIREWALL RULE SET

[1]R. BALA KRISHNAN, [2]Dr. N. K. SAKTHIVEL
[1] School of Computing, SASTRA University, TamilNadu, India.
[2] Professor, School of Computing, SASTRA University, TamilNadu, India.
Email : [1]mailofbala@gmail.com , [2] sakthi@cse.sastra.edu

ABSTRACT

Firewalls are the devices that we are used to protect data. It might be configured to allow certain devices or applications to access our network. The Firewalls are termed as stateful devices. Traditional firewalls typically need to inspect each packet to ensure that it adheres to the policy that has been configured or not, and then perform the necessary action associated to that particular rule. There are various actions and they are typically allow, deny, or even NAT the packet. Many Personal Computer Operating Systems include software-based Firewalls to protect against threats from the public Internet. Existing Firewall Technologies such as Cisco PIX Firewalls and Checkpoint FireWall-1 provide various software tools that allowing firewalls as Clustered or Grouped and these Configured Firewalls will share their loads. The main objectives of these existing technologies are to improve the Resource Utilization along with performance and Security. But however these techniques fail to achieve higher performance while focusing resource utilization. This is one of the serious issues. To address this problem, our research work is developed and implemented an efficient Adaptive Scheduler, which improves the network performance in terms of Resource Utilization, Delay and Throughput. This work also introduced Firewall Reduction Policy, which reduces firewall rules without compromising Security. From our Experimental results, it is established that this proposed technique reduces the computational cost, which leads to higher performance.

Keywords: *Firewall, Adaptive Scheduler, Resource Utilization*

1. INTRODUCTION

A **firewall** is a device which is designed to permit or deny network transmissions based upon a set of rules. It is used to protect networks from unauthorized access when permitting legitimate communications to pass [1,5,6,8,9]. Many Personal Computer operating systems include software-based firewalls are used to protect threats from Internet.

If a firewall decides the fate of some packets not only by examining the packet itself but also by examining the packets that the firewall has accepted previously, then the firewall is called a stateful firewall.

Using a stateful firewall[8,9,10] to protect a private network, one can achieve finer access control by tracking the communication state between the private network and the outside Internet. For example, a stateful firewall can refuse to accept any packet from a remote host to a local host unless the local host has previously sent a packet to the remote host. Although a variety of stateful firewall products have been available and deployed on the Internet, such as Cisco PIX Firewalls [7], CheckPoint FireWall-1 [7], its policies are more. From our implementation result, it is noted that the execution time to process all these rules are considerably high. The main objectives of these existing technologies are to improve the Resource Utilization along with performance and Security. But however these techniques fail to achieve higher performance. i.e. it is very important to keep a firewall’s rule set as small as possible in order to improve the throughput and lower the execution time.

This is one of the serious issues. To address this problem, our research work is developed and implemented an efficient Adaptive Scheduler. The existing techniques and their models have been discussed in the following Section 2.

2. RELATED WORK

In this section, this work is focused various Firewall Policies, Rule Graph & Topology Graphs, Configurations, Access Control Rules and Networks.

2.1. Introduction to Firewall Configurations and Its Access Control Rules

A firewall's configuration contains a large set of Access Control Rules, each specifying Source Addresses, Destination Addresses, Source Ports, Destination Ports, One or Multiple Protocol IDs, and an appropriate action. The action is typically *accept* or *deny*. Some firewalls can support other types of actions such as sending a log message, applying a proxy, and passing the matched packets into a VPN tunnel [2]. An incoming packet will be checked against the ordered list of rules. The rule that matches first decides how to process the packet. Other firewalls such as Cisco's PIX use the best-matching rule instead. Due to the multidimensional nature of the rules, the performance of a firewall degrades as the number of rules increases. Commercially deployed firewalls often carry tens of thousands of rules, creating performance bottlenecks in the network. More importantly, the empirical fact shows that the number of configuration errors on a firewall increases sharply in the size of the rule set.

2.2. Network Models

The Security-Sensitive Enterprise Network consisting of domains called subnets that are connected with each other through firewalls. This work assumed intra domain security is appropriately enforced. This work focuses on inter domain access control [1,2,5,6]. This work further assumes that dynamic routing is turned off on firewalls, while static routes are used to direct inter domain traffic, which is today's common practice in banks or other institutions that have high-level security requirements. In fact, some popular firewalls such as Cisco PIX models do not support dynamic routing protocols. With static routes, robustness is achieved by using dual firewalls. Using static routes on firewalls is a direct consequence of the high complexity in managing the security of a mesh network. It has a number of practical advantages.

First, it ensures that traffic flows are going through their designated firewalls where appropriate security policies are enforced. Second, predictable routing paths simplify the security analysis in a complex network environment, and consequently, reduce the chance of error in firewall configuration. Third, most existing dynamic routing protocols are not secure. Note that dynamic routing is still used inside each domain as long as it does not cross an inter domain firewall. The Figure 1 shows, there are many ways to connect a set of domains via a set of firewalls.

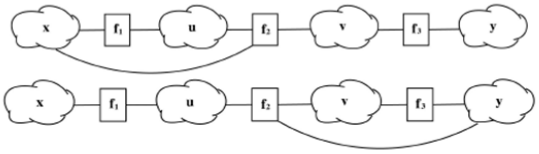
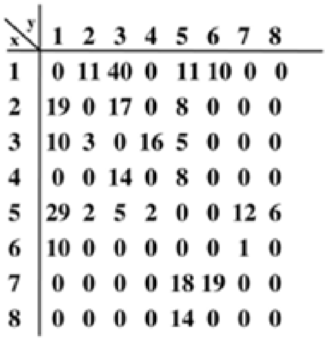


Figure 1. Two Topologies that connect domains x, u, v, and y via Firewalls f1, f2, and f3 set.

For any network topology, there are different ways to lay out the Routing paths. In general, the rule sets to be enforced on the firewalls will be different when we change the network topology or the routing paths.

2.3. Rule Graph and Topology Graph

This work uses the following figure to illustrate a few concepts [1]. The eight domains with IDs from 1 to 8 and the rule matrix (r(x,y), x, y ∈ N) for the eight domains are shown in the Figure 2a.



x \ y	1	2	3	4	5	6	7	8
1	0	11	40	0	11	10	0	0
2	19	0	17	0	8	0	0	0
3	10	3	0	16	5	0	0	0
4	0	0	14	0	8	0	0	0
5	29	2	5	2	0	0	12	6
6	10	0	0	0	0	0	1	0
7	0	0	0	0	18	19	0	0
8	0	0	0	0	14	0	0	0

Figure 2a Rule Matrix r(x,y)

We constructed a rule graph Gr, which is shown in Figure 2b, where each node is a domain and there is an undirected edge <x,y> if r(x,y)+r(y,x) > 0.

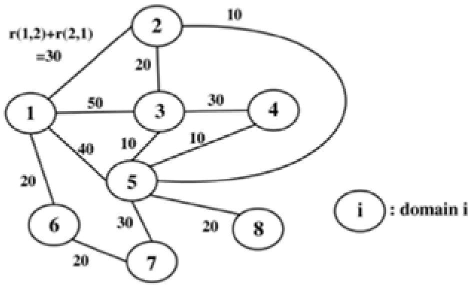


Figure 2b Rule graph Gr

The number of access control rules to be enforced between the two domains, i.e. r(x,y)+r(y,x), is shown beside the link. To get the output of the algorithm, we define a topology graph which is denoted as Gt. It consists of a network topology and a routing structure. A node in Gt is either a data source or domain. An undirected link (x,f) represents a physical connection between a domain x and a firewall f. This research work considers five firewalls, each having three network interfaces. The topology graph [1] is considered for this work, which is shown in Figure 2c. For this topology, the routing table is generated. A few interpretations are given below.

- rt(1,2) = f3 means that the routing table at domain1 has an entry for destination domain2 with the next hop being firewall f3.
- rt(f1,1) = 1 means that the routing table at f2 has an entry for destination domain1 with the next hop being domain1.

The complete routing interpretation for the Figure 2c is as follows.

rt(1,3) = f1, rt(1,5) = f2, rt(1,6) = f3, rt(1,7) = f2. rt(2,1) = f3, rt(2,3) = f4, rt(2,5) = f4, rt(3,1) = f1, rt(3,4) = f4, rt(3,2) = f4, rt(3,5) = f1 and etc.

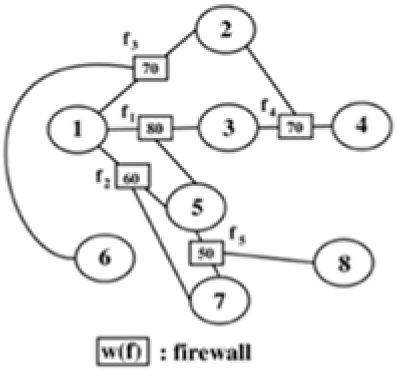


Figure 2c. Topology Graph Gt

2.4 Identified Problems

From the Previous Section, it is observed that various Mechanisms have been proposed to improve the performance of Firewalls. Existing Technologies provide software tool, which Clustered/Grouped the firewalls and therefore the load can be shared. From our experimental result, it is observed that the existing architecture [1,7], which is having multiple Firewalls, is suitable for low volume of users. If the volumes of users are increasing, the processing time is also increased, which leads to performance degradation. This work is also revealed that if the firewall usage level is beyond 75%, its processing time is also increasing, which leads to performance degradation. At the same time, the other parallel firewalls are idle, which leads to poor resource utilization. This is one of the major issues while implementing Multi-firewalls. To address this issue, i.e. to improve the Resource Utilization along with Performance and Security, this research work introduced an Adaptive Scheduler.

To minimize the vulnerability of firewalls, researchers have introducing more policies. As the number of rules increases, the processing time is also increases, which causes the performance degradation. This is the second identified problems. To address these two issues, this research work designed an efficient Adaptive Load Balancing Technique for Smart Firewalls.

3. PROPOSED TECHNIQUE

This Research Work has identified various problems of the existing Firewall Architecture [1], which are discussed in the previous section. To overcome these identified problems, this work is designed an efficient Architecture, which consists of Smart Firewalls with Adaptive Load Balancer, which is shown in the Figure 3.

The implemented software for this Firewall is named as FLUENT which means **F**irewall **L**oad balancing r**U**les r**E**ductio**N** **T**echnique. This Architecture has two techniques namely,

- *An efficient and effective Adaptive Load Balancing Technique, which is used to optimize the load to all replicated servers.*
- *Smart and Heuristic mechanism to minimize the maximum rule set.*

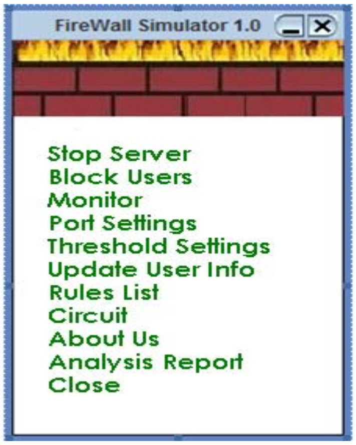


Figure 3. Initial screen of the **FLUENT** firewall software

3.1 Adaptive Load Balancing Technique

This Adaptive Load Balancing Technique is used to find the best firewall for an incoming request and this request will be forwarded identified Firewall. The procedure for finding the optimized firewall is given below. Here the Threshold Limit is denoted as $L_{Th.}$ As we know, the processing time is depends upon the Primary Memory availability and number of users requests. This research work calculates its processing time for various levels of Primary Memory availability and number of users requests [3,4]. Based on which, the optimized threshold value is calculated. This threshold value is differing for different System configurations.

```
Accept  new Call Requests

if (Utilization.FW1 < $L_{Th}$ )  then
          Forward request to FW1
If (Utilization.FW1 >= $L_{Th}$
      and Utilization.FW2 < $L_{Th}$) then
          Forward request to FW2
if (Utilization.FW2 > = $L_{Th}$ ) then
          Forward request to FW3
If (Utilization.FW1 > = $L_{Th}$
     and Utilization.FW2 > = $L_{Th}$
     and Utilization.FW3 < $L_{Th}$) then
          Forward request to FW3
if (Utilization.FW3 >=  $L_{Th}$ ) then
          Forward request to FW4
```

The proposed system and the Monitor Window of the Proposed FLUENT software is shown in the Figure 4a and Figure 4b respectively.

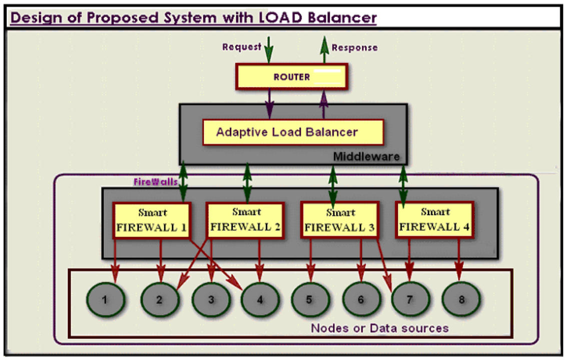


Figure 4a. Proposed System with Load Balancer and Smart FLUENT software

The proposed Adaptive Load Balancer is doing the following steps while forwarding the requests to Smart firewalls.

- Read the Load value of the firewalls
- Select a suitable firewall for handling the arrived request
- Update its load value and assign the task(request) to that firewall
- Get the response from the firewall for the request
- Send the response to Router

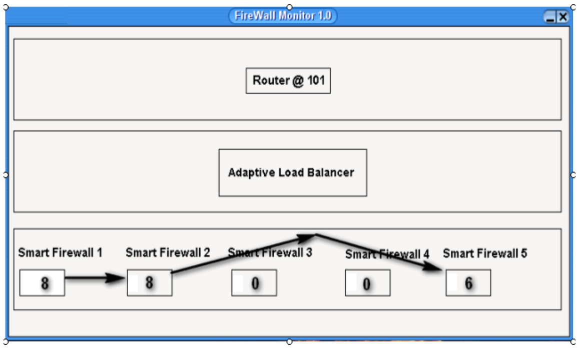
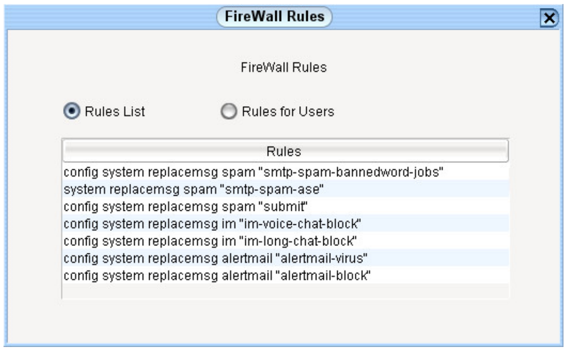


Figure 4b. Monitor window of the Proposed FLUENT Software

3.2 Smart And Heuristic Mechanism To Minimize The Maximum Rule Set

To address the second identified problem, this research work has designed Heuristic Mechanism to Minimize the Maximizing Rule Set. In this Heuristic Mechanism, an efficient Policy Optimizer is introduced to minimize the Maximizing Rule Sets of Firewall. Here various policies/rules have been generated, which are shown in the Figure 5. From the Figure 5, it is very clear that from the Rule Sets, a few rules can be assigned to various users based on their privileges.



FireWall Rules

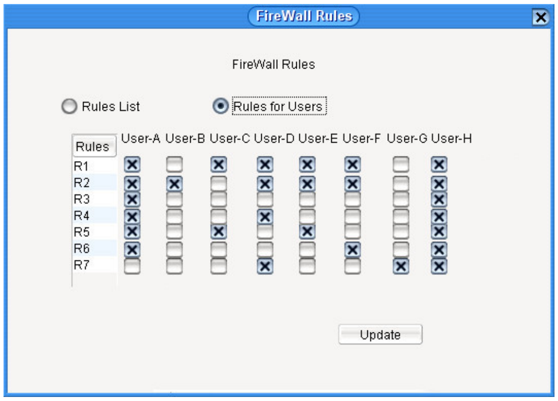
● Rules List ○ Rules for Users

Rules
config system replacemsg spam "smtp-spam-bannedword-jobs"
system replacemsg spam "smtp-spam-ase"
config system replacemsg spam "submit"
config system replacemsg im "im-voice-chat-block"
config system replacemsg im "im-long-chat-block"
config system replacemsg alertmail "alertmail-virus"
config system replacemsg alertmail "alertmail-block"

Figure 5. Shows the rules list of the proposed FLUENT software

This is shown in the Figure 6. The existing architecture is processing all the rules one by one and even if two rules are same, this system executing these two rules separately. It consumes more Processor Time to evaluate these Rules which leads Processor Performance Degradation.

For example, *consider* Rule 1: tcp, 140.192.37.10, any, 163.122.51.*,21 for User 1 and Rule 2: tcp, 140.192.37.10, any, 163.122.51.*, 21 for User 2. These two rules are same, which are assigned to User1 and User2. Our proposed Policy Optimizer will find these types of rules similarities earlier with Early Parallel Acceptance/Rejection Rule (EPRR and it is executing these two policies only once rather twice, which is improving the firewall performance.



FireWall Rules

○ Rules List ● Rules for Users

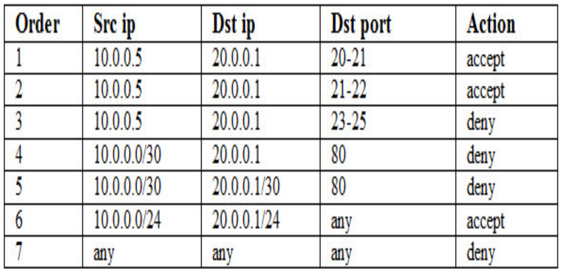
Rules	User-A	User-B	User-C	User-D	User-E	User-F	User-G	User-H
R1	☒	□	☒	☒	☒	☒	□	☒
R2	☒	☒	□	☒	☒	☒	□	☒
R3	☒	□	□	□	□	□	□	☒
R4	☒	□	□	☒	□	□	□	☒
R5	☒	□	☒	□	☒	□	□	☒
R6	☒	□	□	□	□	☒	□	☒
R7	□	□	□	☒	□	□	☒	☒

Figure 6. Shows the rules list of the software that are assigned to various users

4. PERFORMANCE ANALYSIS

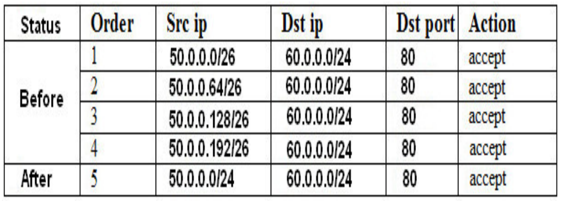
The firewall software FLUENT is thoroughly tested Real-Time environment and the performance of the system is analyzed. In the existing firewall systems Adaptive Load balancing feature doesn't exist but in the proposed system we have introduced the Adaptive Load balancing feature along with Firewall rules reduction facility. Rules reduction works in an effective way if more numbers of users are online. For example, if at a particular time UserA, UserB and UserD are requesting to the firewall application in order to access their authorized data and their usernames and password also correct but in the currently existing firewalls Rules list for UserA, UserB and UserD are checked or manipulated separately even if these users belong to same group.

This proposed technique has Firewall-Trust mechanism which will verify the pattern of user and if the pattern is same, for all users, the only one policy checking will be performed instead of three times, which will reduce the processing time and hence the throughput/response time increases, which will be improving the performance of the system. This application works efficiently not only in the situation where all the rules are common for users but also it works well if some of the rules between the users are different.
It check and finalizes the common rules applicable for all the users and the common rules will be checked once and all other user specific rule will be checked individually.



Order	Src ip	Dst ip	Dst port	Action
1	10.0.0.5	20.0.0.1	20-21	accept
2	10.0.0.5	20.0.0.1	21-22	accept
3	10.0.0.5	20.0.0.1	23-25	deny
4	10.0.0.0/30	20.0.0.1	80	deny
5	10.0.0.0/30	20.0.0.1/30	80	deny
6	10.0.0.0/24	20.0.0.1/24	any	accept
7	any	any	any	deny

Figure 7. Shows the list of arrived requests from various sources and the action that is either accepts or not



Status	Order	Src ip	Dst ip	Dst port	Action
Before	1	50.0.0.0/26	60.0.0.0/24	80	accept
	2	50.0.0.64/26	60.0.0.0/24	80	accept
	3	50.0.0.128/26	60.0.0.0/24	80	accept
	4	50.0.0.192/26	60.0.0.0/24	80	accept
After	5	50.0.0.0/24	60.0.0.0/24	80	accept

Figure 8. *Shows the concatenated requests arrivals*

The request will be either accepted or rejected on the basis of the rules, which is shown in Figure 7. The Analyzed report of the firewall process is shown in the Figure 8, which demonstrates the status of concatenated requests arrivals of both before and after the processing of firewall.

After receiving new requests calls from the input sources the system will analyze the requesting data, destination_ip address and destination port and corresponding action is performed. Threshold level plays a vital role in the process of handling requests by firewall.

That is for different Threshold value, the performance of the Firewall/Processor will vary, which demonstrated in the Figure 9. The threshold value is finalized/calculated by reading the time required for each individual request and the performance percentage.

This test has been performed and the output is monitored and it depends on the basis of the system configuration such as RAM Size, Cache Size, Processor Speed and etc. From the Figure 9, it is observed that the system achieves best performance if the number of request is very low and if the number of requests increases then memory utilization will also be increased and performance of the system is decreased.

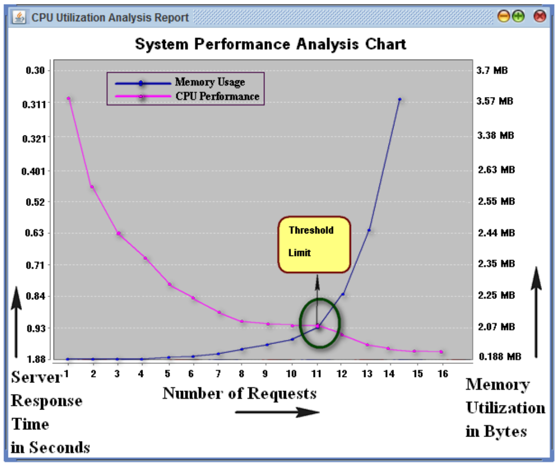


Figure 9. Arrived requests and their corresponding memory usage and CPU performance

The Figure 9 is generated from the tested data and it contains the details about the number of arrived processes and the memory utilization of each process and the performance of the CPU.

5. CONCLUSION

Firewalls are core elements in Network Security. However, managing firewall rules and effective Utilization in High Performance Networks are becoming more complex and error-prone. This paper has studied all the main objectives of the existing technologies such as Cisco PIX and Checkpoint FireWall-1. From our study, it is revealed that the existing firewall architecture need Smart Load Balancing Technique and Rule Reduction Technique. To address this, we have developed and implemented an efficient Load Balancer and Rule Optimizer for Smart Firewalls. These proposed works have been tested thoroughly and from our experimental results, it is established that our work improves Firewall performance in terms of Execution Time, Delay, Throughput and Resource Utilization without compromising Security.

However, we realized that this proposed work has more packet matching rules which lead to considerable Time Complexity and Space Complexity. This identified limitation could be improved in our future work.

REFRENCES

[1] MyungKeun Yoon, Shigang Chen, and Zhan Zhang, “Minimizing the Maximum Firewall Rule Set in a Network with Multiple Firewalls,” IEEE Transactions On Computers, Vol. 59, No. 2, pp. 218-230, February 2010.

[2] J. Wack, K. Cutler, and J. Pole, “Guidelines on Firewalls and Firewall Policy”. National Institute of Standards and Technology, Jan. 2002.

[3] Paul Dütting, Monika Henzinger, Ingmar Weber, “Offline file assignments for online load balancing”, Information Processing Letters 111 (2011) 178–183.

[4] A.X. Liu, E. Torng, and C. Meiners, “Firewall Compressor: An Algorithm for Minimizing Firewall Policies,” Proc. IEEE INFOCOM ’08, Apr. 2008.

[5] M.G. Gouda and A.X. Liu, “A Model of Stateful Firewalls and Its Properties,” Proc. IEEE Int’l Conf. Dependable Systems and Networks (DSN), June 2005.

[6] A.X. Liu, C.R. Meiners, and Y. Zhou, “All-Match Based Complete Redundancy Removal for Packet Classifiers in TCAMs,” Proc. IEEE INFOCOM, Apr. 2008.

[7] M. Gouda and A. Liu, “A model of stateful firewalls and its properties,” Proceedings of International Conference on Dependable Systems and Networks, 2005 (DSN), pp. 128–137, 28 June-1 July 2005.

[8] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In

IEEE INFOCOM'04, pages 2605–2616, March 2004.

[9] Y. Bartal, A. J. Mayer, K. Nissim, and A. Wool. Firmato: A novel firewall management toolkit. Technical Report EES2003-1, Dept. of Electrical Engineering Systems, TelAviv University, 2003.

[10] M. Frantzen, F. Kerschbaum, E. Schultz, and S. Fahmy. A framework for understanding vulnerabilities in firewalls using a dataflow model of firewall internals. Computers and Security, 20(3):263–270, 2001.

AUTHOR PROFILES:

BALA KRISHNAN. R. received the M.C.A., Degree in Computer Applications from SASTRA University, Tamil Nadu, India in 2006. He served as a Software Engineer in IT industry as a Java Programmer in Cloud Computing Domain and Objective-C Programmer in IPhone Platform from June 2006 to june 2009. He has then joined at SASTRA University as an Assistant Professor in 2009. He is currently doing his M.Tech. degree in Computer Science & Engineering in SASTRA University. His research interests include Component Object Modelling, Ethical HACKING, Network Monitoring, Network Security & Biometric Cryptosystems and Semantic Web Services.

SAKTHIVEL. N. K. received the Ph.D., Degree in Computer Science (Intelligent Routing Technique) from SASTRA University, Tamil Nadu, India in 2006. He is currently a Professor with School of Computing, SASTRA University, Tamil Nadu, India. His research interests include Next Generation Networks, High Performance QoS Routing, Wireless Sensor Networks, Bioinformatics and Computational Biology, Semantic Web Services, Data Mining and their Applications. He is the member of Computer Society of India(CSI), India, Advanced Computing and Communication Society,Bangalore. He has published more than 25 Technical Papers in International Journals and Conferences.