An Approach to Improve Performance of a Packet-Filtering Firewall

Urjita Thakar[1]
Department of Computer Engineering
Shri G. S. Institute of Technology & Science, 23, Park Road
Indore 452003 (MP) India
uthakar@sgsits.ac.in

Lalit Purohit[2]
Department of Information Technology
Shri G. S. Institute of Technology & Science, 23, Park Road
Indore 452003 (MP) India
lpurohit@sgsits.ac.in

Akhilesh Pahade[3]
Department of Computer Engineering
Shri G. S. Institute of Technology & Science, 23, Park Road
Indore 452003 (MP) India
akhi005@gmail.com

Abstract- **Firewalls are generally used to protect private network from unauthorized access. The software based firewalls commonly use large number of rules to govern acceptance or discarding of packets exchanged. Such rules can be implemented in access control lists (ACLs) in the firewalls commonly known as packet filters. These access control rules define contents of the fields like source IP address, destination IP address, source port number, destination port number and the protocol used. Based on the values contained in these fields, the packets are either permitted or denied entry into or out of a network. 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. In this paper, a method is proposed that reduces the packet matching time, thus improving the performance of the firewall. To achieve this, the ACL rules are rearranged using clustering index method. Also the rules are prioritized based on their usage. The results show that the proposed method results in significant reduction of packet matching time in packet filtering firewalls.**

Keywords- Access control lists (ACL); Firewalls; Classification; Clustering.

I. INTRODUCTION

With the increasing usage of Internet in various enterprises and organizations, network security is becoming a significantly important issue. Firewalls are an important frontline defence mechanism to secure the networks against attacks and unauthorized access by filtering out unwanted network traffic coming into or going from the secured network. The filtering decision is taken according to a set of filtering rules defined based on security policy defined by the network administrator [1]. ACLs are an essential implementation tool for packet-based traffic policies. The simplest use of ACLs is in the blocking of certain types of traffic to or from a network.

Conventionally, to filter the packets, the filtering rules are matched sequentially. Therefore as the number of filtering rules increases, the time required for packet matching increases. When the number of rules is very large, this packet matching time may be too high posing negative effects on the network performance. It is therefore important to arrange the rules specified in the ACLs that shall lead to quick packet matching and enhance the network performance.

The frequency at which these rules can be applied is an important aspect that needs to be considered for packet matching. It is an important parameter that affects performance of a packet filtering firewall. Hence the ACL rules should be arranged taking this issue into account.

Some work related to optimization of access control lists has been presented by Grout et al.[2][3][4]. In another work, a method to reduce latency in network packet filters has been discussed by Grout et al. considering hit-rate of the rules [5]. An optimization technique has been used taking in to consideration the inter dependence of rules. In the existing systems the packet matching time is linear.

In this paper, a method has been presented to rearrange the ACL rules using clustering and indexing techniques. The method shall enable faster packet matching, improving the network performance.

Rest of the paper is organized as follows- In section 2 the work related to the area has been discussed. Information related to access control lists and their structure is given in section 3. The proposed approach is presented in section 4. Experimental results are discussed in section 5 with concluding remarks given in section 6.

II. RELATED WORK

Some work related to the field is discussed in this section.

An approach to constitute a policy for firewalls is proposed in [6]. In this work, a method for verification of filtering rules is also proposed that shall also eliminate inconsistencies. A functional model of firewalls and an algebra for mathematical description of access policy and a formal tool for firewall configuration have been presented by Vladimir et al. in their work [7]. Another tool to write and modify firewall rules has been presented by Al-Shaer et al. The work also includes automatic anomaly detection for rule insertion and discovery [1]. The performance of the firewall with respect to transaction time and latency have been measured and analyzed for distributed systems by Lyu et al. [8]. In this work, various security levels have been considered for forming policies. It is shown that overhead for security adversely affects the performance of the firewall.

An algorithm to analyze access control lists for firewalls and routers has been presented by Scott Hazelhurst et al. [9]. The method is based on ordered binary decision diagrams (BDDs) to represent and manipulate rules.

An algorithm that uses binary decision diagrams to represent the access control rules has been presented by

Hazelhusrt [9]. The paper also describes how the rule set can be analyzed.

A method based on heuristic tree search that uses filter Buckets has been presented by Woo [10]. The proposed approach allows adaptation to variation in input traffic for which variable size filter buckets have been used. However, the method is very complex since many filter buckets are to be constructed.

Grout et al. have discussed the problem related to implementation of traffic policies on network routers, attempting to minimize the time taken to process a sequence of rules in an Access Control List (ACL). The paper describes heuristic approaches for searching the appropriate rules [11].

In the next section the background related to the work has been presented.

III. BACKGROUND

A. Access Control Lists

An ACL is generally prepared by the Network Administrator. Network administrator decides the access rights for different users and network resources. Network security covers both private network and public network, within in any institution or a company. From a network router, the packet should be passed or blocked according to set of rules that are listed in ACL.

Access Control List (ACL), is a sequence of rules designed to implement a given objective or set of objectives. ACLs can be used for security purposes simply to pass or block packets, or as filters for more sophisticated policies such as traffic shaping, address translation, queuing or encryption [12].

```
access-list 101 permit tcp 192.168.212.0 0.0.0.255 10.0.0.0 0.255.255.255 eq http
access-list 101 deny ip 192.168.212.0 0.0.0.255 10.0.0.0 0.255.255.255
access-list 101 permit icmp any 10.0.0.0 0.255.255.255 administratively-prohibited
access-list 101 permit icmp any 10.0.0.0 0.255.255.255 echo-reply
access-list 101 permit icmp any 10.0.0.0 0.255.255.255 packet-too-big
access-list 101 permit icmp any 10.0.0.0 0.255.255.255 time-exceeded
access-list 101 permit icmp any 10.0.0.0 0.255.255.255 unreachable
access-list 101 permit icmp 172.16.20.0 0.0.255.255
access-list 101 deny icmp any any
access-list 101 permit ip 202.33.42.0 0.0.0.255 any
access-list 101 permit ip 202.33.73.0 0.0.0.255 any
access-list 101 permit ip 202.33.48.0 0.0.0.255 any
access-list 101 permit ip 202.33.75.0 0.0.0.255 any
access-list 101 deny ip 202.33.0.0 0.0.255.255 any
access-list 101 deny tcp 210.120.122.0 0.0.0.255 10.2.2.0 0.255.255.255 eq www
access-list 101 deny tcp 210.120.183.0 0.0.0.255 10.2.2.0 0.255.255.255 eq www
access-list 101 deny tcp 210.120.114.0 0.0.0.255 10.2.2.0 0.255.255.255 eq www
access-list 101 permit tcp any 10.2.2.0 0.255.255.255 eq www
```

Figure 1: A typical Access Control List (ACL).

B. ACL Structure

The figure1 shows is typical access control list in which many filtering rules are defined. The rules have five fields: the permit or deny type, the protocol, a source address, destination address and a flag function for fine-tuning.

Each parameter may be a single value or a range of allowable matches. If the absence of any field such as a address, protocol or flag function the rules will match a packet with any such values fields which are present.

The interpretation of an ACL is that its rules are considered as being processed in sequential order from the top. That is, 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.

Access list 2(a)

```
access-list 102 permit ip 192.168.16.0 0.0.0.255 any
: :
: :
access-list 102 deny ip any 10.0.0.0 0.255.255.255
: :
: :
{access-list 102 deny all} {implicit}
```

Access list 2(b)

```
access-list 102 deny ip any 10.0.0.0 0.255.255.255
: :
: :
access-list 102 permit ip 192.168.16.0 0.0.0.255 any
: :
: :
{access-list 102 deny all} {implicit}
```

Figure 2: Dependent rules

A typical blocking rules in access control list of the Cisco Internetwork Operating System (IOS) [13], is as shown below-
access-list 101 deny icmp any 10.0.0.00.255.255.255 echo-reply

In this rule the ICMP echo-reply packets from any source to the network 10.0.0.0 are to be blocked. The first part of the rule simply assigns it to access list 101.

C. Elements of ACL

An ACL rule contains various types of access elements. Some of them are explained below-

- Src

It is used to define the client IP address. There are three ways to define source IP address.

(1) Single IP address.
e.g. acl singleip src 192.168.12.21
(2) IP address with netmask.
e.g. acl netmaskip src 192.168.1.0/24
(3) Range of IP address.
e.g. acl rangeip src 192.168.11.10-192.168.11.50

- Dst

It is used to define the destination (domain) IP address. Same as src ACL but looks for destination IP address.
e.g.: acl destination dst 192.194.81.67/32 myip
Local IP address on which the client connection exists can also be specified.
e.g.: acl clientip myip 168.126.1.53/32

- Srcdomain

We can match client domain names. Here "." is more important. It must perform an address to name lookup for each request.
e.g.: acl client domain srcdomain .team.com

- dstdomain

To match destination server domain. Here "." is more important.

e.g: acl dstserver dstdomain .linux.com

- srcdom regex

Match against client domain name. Here the source pattern name can be given.

e.g.: acl srcregex srcdom_regex tyre #this looks for word 'tyre' in the client domain name.

- dstdom_regex

Same as srcdom_regex but looks for destination domain based on regular expression.

e.g.: acl dstregex dstdom_regex linux. #this looks for word 'Linux' in the client's request.

- url_regex

Matches using regular expression on the compiler request URL. There are three ways to define url_regex acl.

e.g.:

(1) Contains:

acl contains url_regex http//www.google.co.in

(2) Start with:

acl contains url_regex ^http://www.google.com

(3) End with:

acl contains url_regex in$

- urlpath_regex

This acl matches any protocol, port, and host name information.

e.g.: acl urlpathregex urlpath_regex index.com

IV. PROPOSED METHOD

In the proposed method, a rule matching module is envisaged that rearranges the rules to improve the packet matching time. Clustering index method is used to form clusters of the ACL rules according to the subnet IP addresses. Further, the rules within the cluster are prioritized based on the usage frequency. Architecture of the proposed system is as shown in Figure 3.

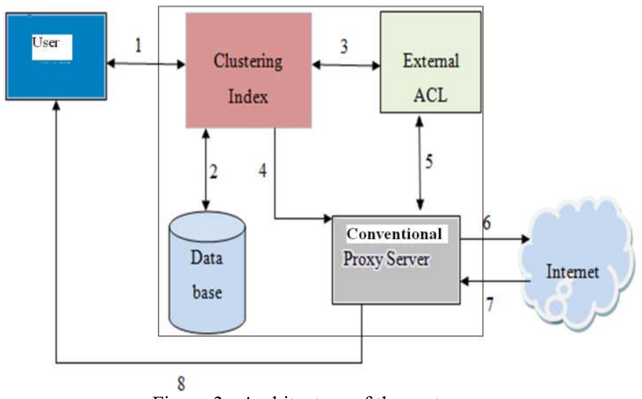


Figure 3: Architecture of the system

A clustering index module is added to the conventional proxy server. This module prepares the ACLs clustered on the basis of IP addresses that are used as indexes. The user requests for the resources to this module.

The flow of activities is as given in the Figure 4.

Step 1: User sends a request to the proxy server for a web resource.

Step 2: After taking request in the clustering index module uses database which already contains different ACL rules. If the requested resource has matching rule defined in the present ACL, then that rule count is incremented by 1.

Step3: The clustering index module decides which request will match in the external ACL according to sub-net IP address.

Step 4: The clustering index module forwards the request to proxy server.

Step 5: Proxy server decides whether the request is to be granted or rejected by using External ACL.

Step 6: If request is accepted then resource is accessed from the Internet.

Step 7: Response is received.

Step 8: Proxy server sends response to user.

A. Clustering of ACL

The process of organizing object into groups whose members are similar in some way is called clustering. Here different ACL clusters are formed according to subnet addresses. The clustering index has two fields: the subnet address and a block pointer for different ACL clusters. The number of ACL clusters depends on the number of subnets in the network.

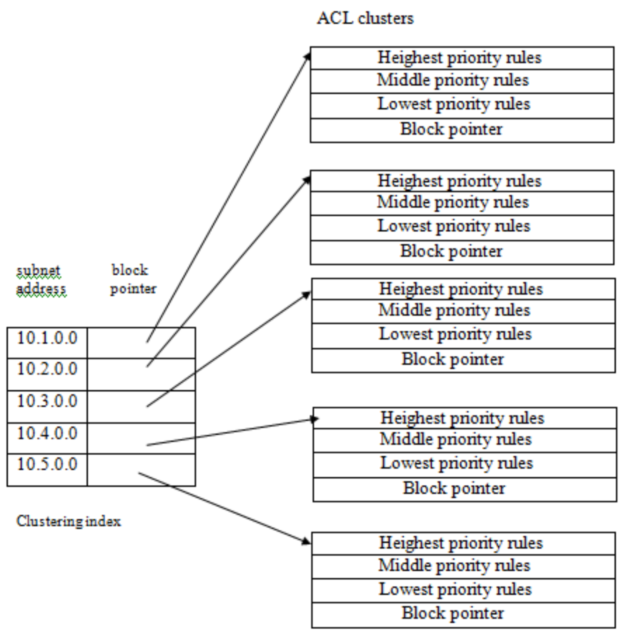


Figure 4: Clustering index with a separate block cluster for each group

In Figure 4 block pointer point to ACL clusters. Each ACL cluster holds ACL rules defined for the individual subnets.

B. Arranging rules according to priority

In the Access Control List rules are defined by the network administrator. Many ACLs contain access control rules that are very large in number. All the rules may not match very often. Therefore the ACL rules are arranged as per the frequency of their usage. Since if a rule is accessed more number of times it should be assigned higher priority. With each rule, a match count is associated. Whenever the ACL rule is matched, its count value is increment by 1. The rule with higher count is placed higher in the list since these rules are matched more often. Thus the ACL is arranged in the order of usage frequency. Figure 5 shows an example of cluster having ACL rules arranged according to their priority.

Highest priority rules
Middle priority rules
Lowest priority rules
Block pointer

Figure 5: ACL Cluster

For any request received from the client the clustering module checks ACL rules.

Algorithm for finding access frequency of rules is as follows:

```
// Ri= incoming request
// Rp= present rule in database
While (true)
{
        get response from the client
        match with the blocked acl rules Rp
        If  (Ri == Rp)
        {
        Increment  Rp by 1
        }
        else
        {
        Insert Ri into blocked acl rules
        }
}
```

The proposed method was tested on a network in an educational institute. The results and test cases are discussed in the next section.

V. TESTING AND RESULTS

The testing was done taking ACL for a private network having 5 subnetworks.

In this work, the fields source address; destination address and type of the ACL were considered. The subnets were with following ids-

1. Subnet -1: 10.1.0.0
2. Subnet-2:10.2.0.0
3. Subnet-3:10.3.0.0
4. Subnet -4:10.4.0.0
5. Subnet -5:10.5.0.0

First on the existing ACL, clustering was applied to obtain various clusters based on the subnet id.

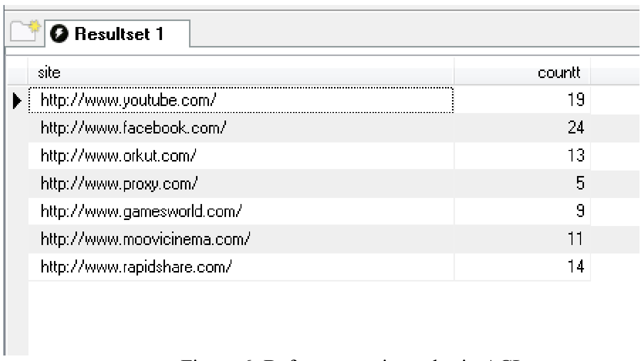
The rules in each ACL cluster were then rearranged based on the matching frequency. Table 1 shows the ACL1 after arranging the rules.

Table 1: After arranging rules in ACL 1

rules	IP Address	permission	count
http_access	10.1.33.20	allow	99
http_access	10.1.33.18	allow	89
http_access	10.1.33.9	allow	84
http_access	10.1.33.13	allow	82
http_access	10.1.33.14	allow	81
http_access	10.1.33.19	allow	79
http_access	10.1.33.5	allow	78
http_access	10.1.33.15	allow	74
http_access	10.1.33.10	deny	72
http_access	10.1.33.1	allow	70
http_access	10.1.33.2	allow	69
http_access	10.1.33.12	allow	69
http_access	10.1.33.16	deny	66
http_access	10.1.33.11	allow	65
http_access	10.1.33.4	allow	60
http_access	10.1.33.6	deny	50
http_access	10.1.33.17	deny	47
http_access	10.1.33.3	deny	40
http_access	10.1.33.7	deny	39
http_access	10.1.33.8	allow	38

Similarly, the rules were rearranged for ACL2, ACL 3, ACL 4 and ACL 5.

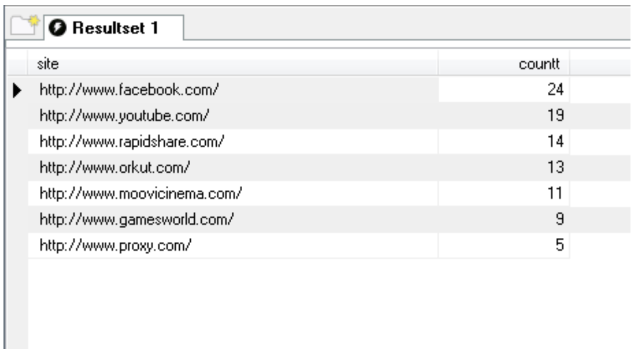
In one ACL many urls are defined that the network administrator wishes to block. Figure 6 shows two fields, 'site' and 'count' showing the number of times the blocked sites have been accessed by the user.



site	countt
http://www.youtube.com/	19
http://www.facebook.com/	24
http://www.orkut.com/	13
http://www.proxy.com/	5
http://www.gamesworld.com/	9
http://www.moovicinema.com/	11
http://www.rapidshare.com/	14

Figure 6: Before arranging rules in ACL

Figure 7 shows the access control list after arranging rules in descending order of usage frequency. Highly accessed rules appear at the top and the least used rules appear at the end of the list.



site	countt
http://www.facebook.com/	24
http://www.youtube.com/	19
http://www.rapidshare.com/	14
http://www.orkut.com/	13
http://www.moovicinema.com/	11
http://www.gamesworld.com/	9
http://www.proxy.com/	5

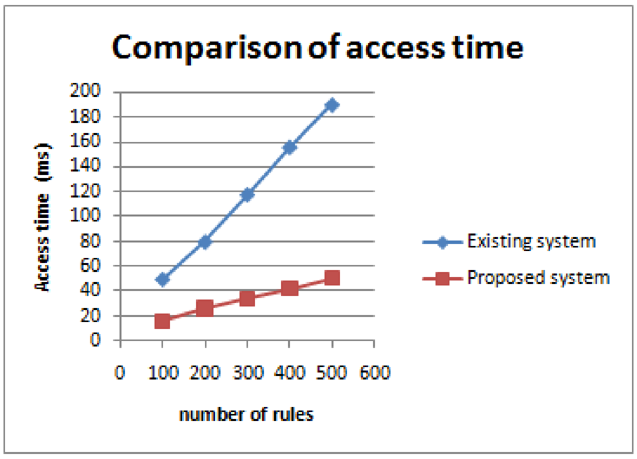
Figure 7: After arranging rules in ACL

With the proposed system, packet matching time shows significant improvement as given in Table 2.

Table 2: Compare time between existing system and proposed system for 500 number of rules

rules	Time taken by existing system	Time taken by proposed system	% improvement
100	50 ms	16 ms	68 %
200	80 ms	26 ms	67 %
300	118 ms	34 ms	71 %
400	156 ms	42 ms	73 %
500	190 ms	50 ms	73 %

When the ACL contains 500 rules it is observed that the average packet matching time improves by about 73%. For ACLs containing 100, 200, 300, 400 and 500 rules, the average packet time is as graphically shown in figure 8.



Graph 1: Comparison of existing and proposed systems

VI. CONCLUSIONS AND FUTURE WORKS

In this paper the problem of large amount of time required to match the rule for the request in the ACL has been addressed. The method proposed considers the usage frequency of various rules present in the ACL. The rules have been rearranged based on this important parameter using clustering and indexing technique. It is observed that the proposed method results into significant improvement in packet matching time in packet filters. For an ACL containing 100 rules, the improvement in packet matching time is about 68% while for ACL containing 500 rules the improvement is about 73%. Results show that more are the number of rules in the ACL, better is the performance.

In future the usage frequency can be calculated based on the requirements of different users such as administrator staff, client etc.

REFERENCES

[1] Ehab S. Al-Shaer and Hazem H. Hamed, "Firewall Policy Advisor for anomaly Detection Rules Editing and Translation". Integrated Network Management 2003, pp 17-30.
[2] Vic Grout and John N. Davies, "A Simplified Method for Optimising Sequentially Processed Access Control Lists", IEEE 2010, pp 347-352.
[3] V. Grout, J. McGinn & J. Davies, "Real-Time Optimisation of Access Control Lists for Efficient Internet Packet Filtering, Journal of Heuristics", Vol. 13, No. 5, October 2007, pp435-454.
[4] V. Grout, J. Davies & J. McGinn, "An Argument for Simple Embedded ACL Optimisation", Computer Communications, Vol. 30,No. 2, January 2007, pp280-287.
[5] V. Grout, J. McGinn & J. Davies, Reducing Processing Latency in Network Traffic Filters, Proceedings of the 5th International Network Conference (INC 2005) Samos Island, Greece, 5th-7th July 2005, pp.3-10.
[6] Kotenko, I. Polubelova, "Verification of security policy Filtering Rules by Model Checking", IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS) Prague, 15-17 Sept.2011,Volume: 2 pp. 706 – 710 ISBN: 978-1-4577-1426-9 DoI: 10.1109/ IDAACS.2011.
[7] Vladimir Zaborovsky, Vladimir Mulukha, Alexander Silinenko, Sergey Kupreenko, "Dynamic Firewall Configuration Security System Architecture and Algebra of the Filtering Rules", The Third International Conference on Evolving Internet, IARIA, 2011. ISBN: 978-1-61208-1410, pp. 40-45.
[8] Michael R. Lyu and Lorrien K. Y. Lau, "Firewall Security Policies, Testing and Performance Evaluation". Proceeding on COMPSAC '00 24th International Computer Software and Applications Conference, pp 116-121. IEEE Computer Society Washington, DC, USA , 2000.
[9] Scott. Hazelhusrt. "Algorithms for Analyzing Firewall and Router Access Lists." In Technical Report TR-WitsCS-1999, Department of Computer Science, University of the Witwatersrand, South Africa, July 1999.
[10] T. Woo. "A Modular Approach to Packet Classification: Algorithms and Results." In Proceedings of IEEE INFOCOM', March 2000.
[11] V. Grout, & J. McGinn, "Optimisation of Policy-Based Internet Routing using Access Control Lists", Proceedings of IFIP/IEEE International Symposium on Integrated Network Management, Nice, France, 16-19 May 2005.
[12] Syngress, Building Cisco Remote Access Networks, Syngress Media, 2002. ISBN: 1-928994-13-X.
[13] A. Colton, Cisco IOS for IP Routing, Rocket Science Press Inc.2002.