

FINAL YEAR RESEARCH REPORT
EARLY PACKET REJECTION MULTI-LEVEL FIREWALL

Submitted By

MIR RIFAT HASAN ABRAR
ID: 201-15-13612

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Professor Dr. Touhid Bhuiyan
Head of Computer Science & Engineering
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 19th 2023

APPROVAL

This Research Project titled “Early Rejection Multi-Level Firewall”, submitted by Mir Rifat Hasan Abrar, ID No: 201-15-13612 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 19-01-2023.

BOARD OF EXAMINERS

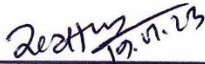


Dr. Touhid Bhuiyan

Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman

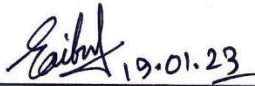


Dr. Md Zahid Hasan

Associate Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Saiful Islam

Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Sazzadur Rahman

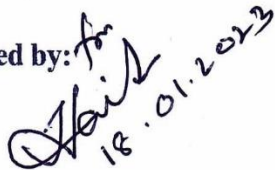
Associate Professor

Institute of Information Technology
Jahangirnagar University

External Examiner

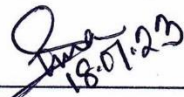
DECLARATION

I am Mir Rifat Hasan Abrar ID: 201-15-13612, student of Daffodil International University on Computer department. I am declaring that I have completed this research report on Early Rejection Firewall under the supervisor of **Professor Dr. Touhid Bhuiyan**, Department Head of Computer Science & Engineering, Department of CSE Daffodil International University. I hereby certify that I had a great support from my Supervisor while completing this research project. I further certify that this research project, nor any component of it, has ever been submitted to another institution for consideration of a degree or certificate.

Supervised by: 
18.01.2023

Professor Dr. Touhid Bhuiyan
Dept. Head of Computer Science & Engineering
Department of CSE
Daffodil International University

Co-Supervised by:


18.07.23

Ms. Taslima Ferdous Shuva
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:


18.01.23

Mir Rifat Hasan Abrar
ID: 201-15-13612
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

I want to start by expressing my sincere gratitude to Almighty ALLAH for the wonderful grace that has enabled me to successfully finish the final year Research.

I feel really fortunate to have a Supervisor like Professor Dr. Touhid Bhuiyan, Dept. Head of Computer Science & Engineering, Department of CSE Daffodil International University, Dhaka. I want to give my supervisor a heartfelt thanks for his continuing support of his endeavors. I am also grateful to my Co Supervisor Ms. Taslima Ferdous Shuva, Assistant Professor of Computer Science & Engineering, Department of CSE Daffodil International University, Dhaka.

This research project was made possible by our supervisor's extensive knowledge and deep interest in the topic of " **EARLY REJECTION MULTILEVEL-FIREWALL**" It was made possible to accomplish this research project thanks to his unending tolerance, academic guidance, persistent, vigorous supervision, helpful suggestions, and reviewing several drafts and editing them at every level.

I would like to extend our sincere appreciation to our Department Faculty members and the Head of the CSE Professor Dr. Touhid Bhuiyan, for their kind assistance in helping me complete my research project.

Finally, I must thank my parents and especially my mother & also others for always being there for me with their support, attention & blessings.

ABSTRACT

While performing research is a required element of a path, engineering students might use the Early Rejection Multilevel-Firewall for as a guidance. This Multilevel-Firewall for Early Rejection is intended to serve as a resource for instructors as well as engineering university and college students who are conducting research for reports or theses. Each day, whether or not we use the net for extended intervals of time or surf the internet at the same time as working, here we need to bypass a firewall. An essential component of a firewall's number one universal overall performance is packet filtering. A lot of studies and researches done to enhance the high-universal-performance packet filtering components of a firewall. In this Research Project, I have proposed a firewall that distinguishes incoming packets or facts according to protocols like UDP, TCP and others. Then, the use of the larger checking machine's passing device, and or the structuring of the rules according to the protocols. Now, we'll look at how effectively each technique performed on its own and using that data as support for our excellent Research that merging packets or data with the usage of some strategies might perhaps increase firewall performance. The report also offers conclusions and recommendations that, in my opinion, if put into practice would enhance the firewall system. Secondly, the report makes findings and suggestions that, in my opinion, would improve the firewall system if they were applied.

Table of Content for the Research Project

	Page No
Approval Page	ii
Declaration	iii
Acknowledgement	iv
Abstract	v
List of Tables	i-ii
List of Figures	iii-iv
List of Tables	v
Chapter 1: Introduction	
1.1 Introduction	1
1.2 Firewall Details	1
1.3 Motivation	2
1.4 Relational of Study Firewall	2
1.5 Research Questions	3
1.6 Expected Output	4
1.7 Project Management and Finance	4
1.8 Report Layout and Working Principle	5
Chapter 2: Background	
2.1 Preliminaries of early ejection firewall	6
2.2 Related Work	7
2.3 Background Study	7
2.4 Comparative Analysis and Summary	8
2.5 Scope of the Problem	8
2.6 Challenges	9
Chapter 3: Research Methodology	
3.1 Research on Early Traffic Rejection	10
3.2 Details of SA-BSPL Techniques	10-12
3.3 Host Based Firewall	13
3.4 Data Collection Procedure	14-18
3.5 Statistical Analysis on Optimizing Rejection	18
3.6 Regular Traditional Based Packet Filter	19
3.7 Proposed Methodology on Protocol Based Packet Filter	20-21
3.8 Implementation Requirements for research on firewall	22-23

Chapter 4: Research Result and Discussion	
4.1 Traditional Filter Firewall	24
4.2 Experiment on Protocol Based Packet Filter	25-26
4.3 Experimental Result	27
4.4 Research Outcome	27-28
4.5 Discussion on Firewall Rule	28
Chapter 5: Impact on Society, Environment and Sustainability	
5.1 Impact on Society of Protocol Based Filter Firewall	29
5.2 Impact on Environment of Matching Recency	30-31
5.3 Ethical Aspects on Protocol Based Firewall	32-33
5.4 Optimal Rules Ordering ORO procedure	33-35
5.5 Dynamic rule in order for Protocol	35
5.6 Integration with packet matching	36-37
5.7 Performance based Update System	38
5.8 Sustainability Plan for Protocol Based Firewall	39-40
Chapter 6: Summary, Solution, Conclusion and Implication for Future Research	
6.1 Summary of the Study	41
6.2 Solutions	41
6.3 Conclusions	42
6.4 Implication for Further Study on Packet Filter Firewall	42
References	43-44
Appendices of Research	45

LIST OF FIGURES

FIGURES	PAGE NO
Figure: 1.2.1 Firewall Structure	2
Figure: 1.8.2 Firewall Layout and Working Principle	5
Figure: 2.1.1 Early Rejection System Basic Diagram	6
Figure: 2.6.2 Firewall Disable Situation for packets	9
Figure: 3.2.2 Rejection Rules	12
Figure: 3.2.3 Rejection Rules in Standard Laws	12
Figure: 3.3.4 Host Based Firewall Working Procedure	13
Figure: 3.4.5 Dynamic rule formula	14
Figure: 3.4.6 Expected Result	15
Figure: 3.4.7 Formula of Algorithm-1	15
Figure: 3.4.8 Formula of Algorithm-2	15
Figure: 3.4.9 Formula of Determining the Average Number	16
Figure: 3.4.10 Portion of the total traffic analysis System.	16
Figure: 3.4.11 Following Condition for Early Rejection.	17
Figure: 3.5.12 Traditional Firewall Rule Matching	19
Figure: 3.6.13 Proposed Protocol Based Early Rejection Firewall Model.	20
Figure: 3.7.14 Packet Filtering Structure for Internal Network.	22
Figure: 3.8.15 Packet Filtering Structure for Internal Network.	23
Figure: 4.1.1 Traditional Filter Firewall	25
Figure: 4.2.2 Protocol Based Firewall Rule Matching	26
Figure: 4.2.3 Network Accessing Protocol analyzing System.	27

LIST OF TABLES

FIGURES	PAGE NO
Table: 3.2.1 Table of Early Filtering System	10-11
Table: 3.7.2 Proposed Protocol Filter Model Table	20
Table: 3.8.3 These are the approximate amount of generated through FLIP	24
Table: 4.4.4 Source of Traffic Incoming	28
Table: 4.4.5 Source of Traffic allowing & Disallowing.	29
Table: 5.1.6 Impact of Exception Rules for Early Rejection Protocol.	29

CHAPTER 1

INTRODUCTION

1.1 Introduction

The final year Research Project is the completion of the bachelor's degree program for students. The major goal of this Research is to motivate pupils to put the knowledge they have learned in university to use. It enables us to focus on one issue for a considerable amount of time and shows how to address issues that arise in everyday life.

1.2 Firewall Details

As in today's internet infrastructure, firewalls have emerged as one of the key safety additives. A firewall is a kind of network security system that monitors and regulates incoming and outgoing network traffic in accordance with the security rules that have been defined.

There aren't many different kinds of firewalls available nowadays. Although there are five different types of firewalls in multimedia, we often only know about four of them. for example,

- Packet filtering firewall
- Circuit-Level Gateway Firewall
- Application-Level Gateway Firewall
- State-full Inspection Firewall
- Next-Generation Firewall (NGFW)

I've examined firewall early packet rejection strategies in my research project. Firewalls are gradually becoming the most important safety component in the new internet architecture. The system filters out harmful packets in accordance with user protection system.

Step 1: Monitors and control traffic based on rules of firewall.

Step 2: Act like a barrier to secure the Network with multi firewall zones & IP addresses.

Step 3: Protocol-based Firewall working process.

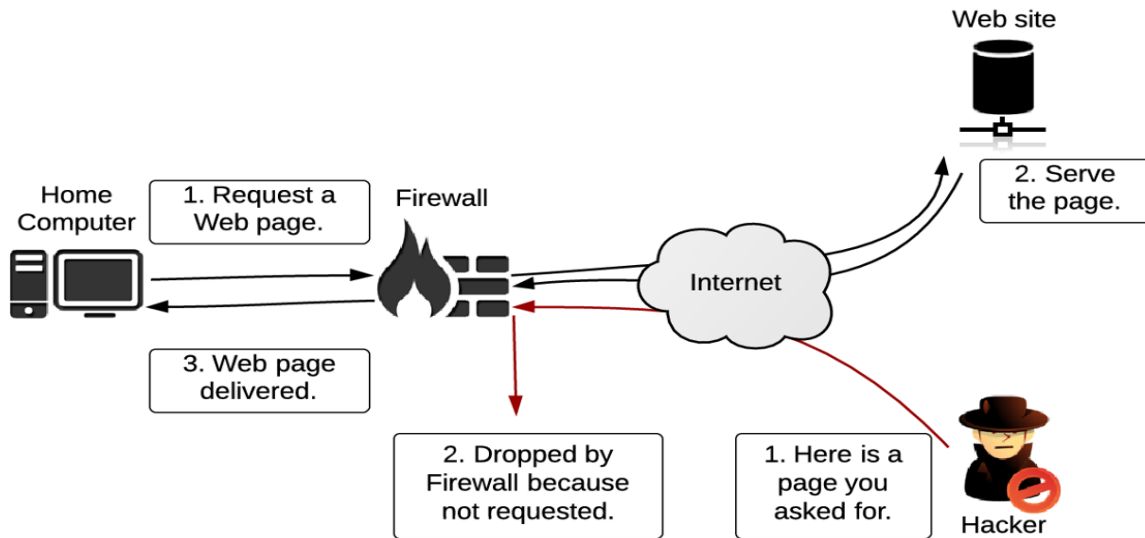


Figure: 1.2.1 Firewall Structure

1.3 Motivation

This paper is motivated through the algorithm proposed by Al Shaer known as “Early Traffic Rejection” & also Zouheir Trabelsi’s Researches. There, it was presented that, if any packet does not match any of the common criteria for “allow” it should be rejected at its earliest without any further checking. In order to cover all of the rules in the policy, they combined the common field values to make certain rules. They provided significant amount of proof to show that their method is feasible. The second part is proposed hypothesis; as the packet is at first separated according to the protocol for example ‘TCP or UDP’. Afterwards, the packet is checked by dynamic rule. The primary key is that the behavior of Internet traffic still retains a number of characteristics that may be used to improve packet filters.

1.4 Rational of Study Firewall

Here we can see the problem of classification of packets has been broadly finished lately. The fundamental approach to packet layout is to iteratively explore the rule list until a match is discovered. Although this approach is reasonably memory-efficient, its scalability is frequently subpar since the search time is connected with the length of the rule.

Al Shaer and H Hamed A method that uses traffic characteristics to optimize firewall filtering settings has been put forth by them [2]. Their technique adapts to the traffic conditions, timely, using actively calculated statistics to dynamically optimize the rules of packet filtering. In other words, to reduce the average packet matching time, they employed statistical search trees to exploit traffic features.

1.5 Research Questions

There are multiple questions around this globe about Firewall & Early Rejection Firewall. I have put few questions which are asked many times about Firewall Working System. As,

➤ **How do firewalls choose which packets to let through and which to block?**

The static packet filtering firewall does not detect between application protocols by looking at fields in the packet's IP & protocol headers instead, it mainly works at the network levels of the OSI model.

➤ **What are some methods for network traffic filtering by firewalls?**

Firewalls are designed to screen out the malicious software & other dangerous packets from getting enter the communications. The firewall blocks data packets from entering our PCs' network if it identifies them as security hazards.

➤ **How does a firewall choose which things to block and which things to allow?**

In order to defend against attacks, firewalls extensively assess incoming communication in line with pre-established criteria and prevent traffic from questionable or suspicious sources.

➤ **What takes place once a packet enters?**

Similar to this, a firewall analyzes each packet of data to figure out where it came from, where it is heading, and then it decides whether to accept it and allow it to go on its path, deny it, or drop it.

1.6 Expected Outcome

I am trying to show the difference between Traditional firewall principles & Protocol-based firewall principles. Actually, a firewall is installed for reduce the event of unwanted network faster than a regular firewall system communication [10]. This work describes the filtering Representation and the Early Rejection Firewall Rule. The suggested model can be grasped graphically. With this strategy, an algorithm for locating and alerting filtering rule mistakes may be created.

As,

- For each policy group P do
- $P, state \leftarrow$ unfinished
- end for
- for each P with $P, state =$ unfinished do
- rule Generation (P)
- end for

1.7 Project Management

Firewall management is the process of configuring and monitoring a firewall to maintain a secure network. Firewalls are an integral part of protecting private networks in both a personal and business setting. An organization may have many different firewalls protecting its devices and network as standard. In order to maintain a secure network, a firewall must be configured and management. In both a personal and professional context, firewalls are essential for securing private networks. A company may use a variety of firewalls to safeguard its network and devices on a regular basis.

Administrative Services

In order to manage and maintain networks, computers, users, and applications, a variety of protocols, are covered in this part. which also include booting protocols. Dynamic Host Configuration Protocols, timekeeping protocols like ICMP and NTP, informative protocols like Syslog and SNMP, routing protocols like RIP and OSPF, and (DHCP). As necessary, we also talk about the tools, such as ping and traceroute, that employ these protocols.

1.8 Report Layout and Working Principle

This section will compare and contrast the classical firewall concept with the protocol-based firewall philosophy. By filtering incoming and outgoing network traffic, a firewall is a system that offers network security based on a set of user-defined rules. In general, a firewall's job is to lessen or stop the occurrence of malicious network connections while allowing all valid communications to pass freely.

The Early Rejection Firewall Rule and its filtering Representation have been defined in this work. Using this method, an algorithm for locating and reporting filtering rule flaws might be created. Then, with the development of an anomaly-free firewall rule editing system, it will be much easier to create and amend rules in the 2/21 firewall policy [8]. With the assistance of my advisor, I want to incorporate these strategies into a future product using the Java programming language.

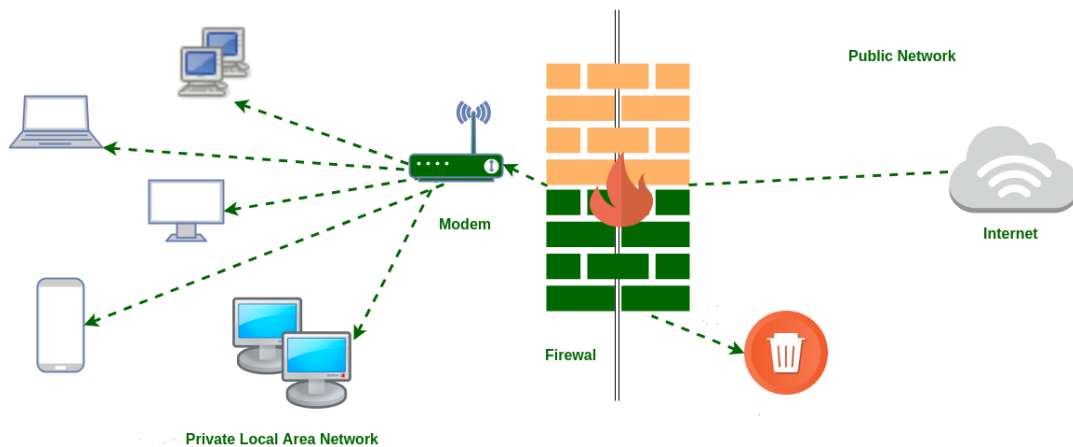


Figure: 1.8.2 Firewall Layout and Working Principle

Here, we can observe how firewalls inspect packets for harmful attack methods that have been recognized as known threats. If a data packet is identified as posing a security concern and is blocked from accessing the network or our machine by the firewall, it is marked and blocked.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Preliminaries of Early Rejection in Firewall

The topic of early packet filters has received a lot of attention for two main reasons. The first is to protect firewalls against DoS attacks that target the standard deny rule. In order to lower the filtering cost brought on by discarding unnecessary data, the second step is to establish approximation policies that may swiftly filter out discarded traffic.

Packet filtering is crucial to the operation of many network devices, such as firewalls, routers, and intrusion detection and prevention systems. Many studies on packet classification have been proposed in hopes of improving packet filtering. Unfortunately, the vast majority of research uses deterministic techniques and disregards traffic dynamics. The early packet rejection techniques employed by firewalls are the subject of our investigation in this research. Both the merits and downsides of the strategies are covered.

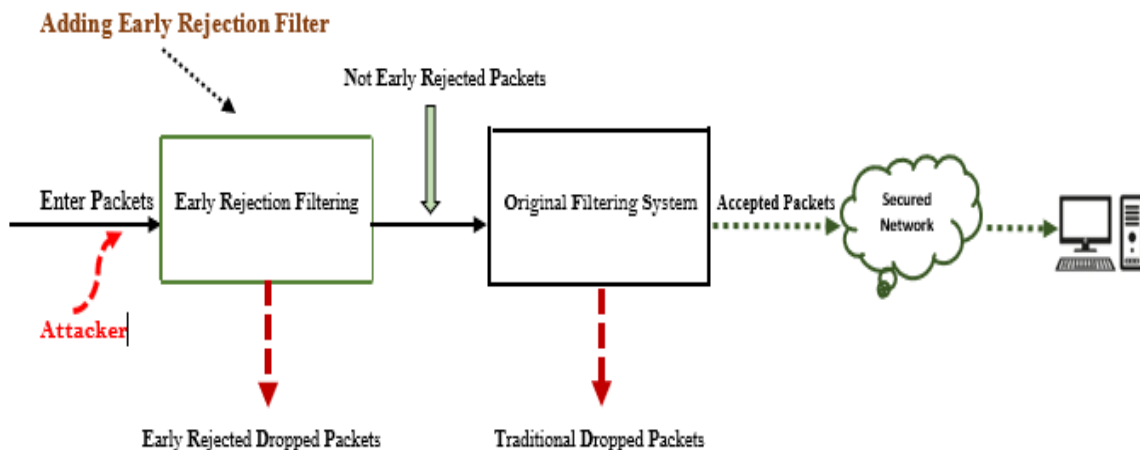


Figure: 2.1.1 Early Rejection System Basic Diagram

However, while early packet rejection is crucial for enhancing firewall performance, most packet classifiers rarely optimize it. A few changes have also been proposed. This study can serve as a basis for providing new ideas that enhance existing techniques.

2.2 Related Works

We can notice that most of the packet difficulties in the categorization have just been resolved. Searching the rule list iteratively until a match is discovered is the fundamental strategy used in packet layout. Due to the trade-off between search time and rule length, this strategy, although being very memory-efficient, frequently has limited flexibility.

E. Cohen and C. Lund had proposed and measured decision tree classifiers with common branches. These classifiers proved that their data had comparable average and worst-case time performance while having linear worst-case memory constraints and requiring substantially less memory than traditional decision tree classifiers [13]. They claim that "common branches resourcefully leverage the structure that is available in real-time data sets".

Al Shaer and H Hamed has proposed a method that makes use of traffic attributes to enhance firewall filtering rules. In their approach, statistics are actively computed for dynamic packet filtering rule optimization in response to the traffic conditions at hand [22]. To lower the average packet matching time, they employed statistical search trees to take advantage of traffic characteristics. Another method by Al Shaer & H Hamed maximizes early rejection of undesired flows with the least amount of influence on other flows.

2.3 Background Study

The main duty of packet filters, in security policies, is to sort packets based on a set of rules that characterizes the filtering policies. The protocol, source IP, source port, destination IP, and destination fields which are typically controlled in various header fields of the packet are the data employed for packet filtering.

In firewall, the packets are matched with rules and then forwarded or blocked as ("Allow" or "Deny") to a specific boundary.

A packet filtering rule set is arranged in a firewall policy. The packets are serially compared against the provided rules until a matching rule is discovered or matched [28].

If there is no match found, then the packet will be “Denied” by default system. The packet organization for routing is, purely, based on the destination IP address & location. The rules of firewall are stated as IP address formula.

2.4 Comparative Analysis and Summary

The performance of network firewalls against DDoS assaults has recently been under intense scrutiny. The general security of the protected network is at great risk if network firewalls are not built well enough to withstand DDoS attacks. In order to forecast how successful and efficient network firewalls are during DDoS attacks, there is a rising demand for evaluating, modeling, and simulating the performance of network firewalls.

In order to do the necessary tweaking for optimum performance, this will assist firewall designers and system administrators in locating bottlenecks and important variables that affect the operation of the firewall. Numerous design and operational concerns may be quickly resolved with performance analysis. This will assist firewall designers in doing a first-cut design to narrow the pool of design choices, and then using simulations or experiments to evaluate the performance of a few excellent ideas before constructing and deploying the system into their actual network environment.

2.5 Scope of a Problem in Early Rejection Firewall

Since the sequence of rules in the filtering rule list directly affects the semantics of the firewall security policy, a new rule must be added to the policy in the proper order to prevent shadowing or redundancy from being introduced. The user can insert new rules in the appropriate places by using the policy editor. Also, it locates anomalies that could develop as a result of properly inserting the new rule.

2.6 Challenges

According to research, source and destination IP addresses are frequently stated in terms of address ranges or system runs, or both, when defining firewall rules. There are a few problems or security advice that must be followed. Having to deal with issues of this nature is frequent.

The challenges rules can contain fixed matches, prefix matches, or range meets over various fields of the packet header. The most common problems and challenges for the Early Rejection Multi-Firewall comes in-front of us are.

- Build a firewall that satisfies our requirements
- Creating particular effective combinations.
- successfully updating the firewall.
- Following rules and regulations.
- Removing erroneous results when traffic is obstructed.
- Guaranteeing efficient setups.
- To observe the laws and regulations.
- To stop fake and harmful traffics while avoiding false positives.

For choosing the proper firewall we are up to stop fake and harmful traffics while avoiding false positives.

What if Firewall Fails?

All type of data packets can enter and leave the network without restriction when a firewall is disabled. This encompasses both legitimate traffic and malicious data, placing the network at danger. Unwanted access across a poorly built firewall can result in breaches, data loss, and IP that has been taken or demanded as ransom. Unexpected interruptions: A misconfiguration may prevent a customer from interacting with a business, and this downtime costs the company sales.



Figure: 2.6.2 Firewall Disable Situation for packets.

All we can see if the Firewall is Disable to process packets them, all type of wanted & Unwanted data packets can enter and leave the network without permission when a firewall is disabled.

CHAPTER 3

RESEARCH DETAILS

3.1 Research on Early Traffic Rejection

Firewall rules are written mainly to secure the internal network from the outside thread. These rules filter the unwelcome packets or thread which are found in the incoming node of traffic. The main target of firewall rules is to remain handle the exceptions and match the exception with the default deny rule. This problem costs highly matching overhead rather than other policies in the firewall rule-set. But the annoying data packets are dropped according to the denial policies. These midway denial rules are not optimized if these are optimized then it will have great impact on firewall act. In this segment, I have proposed a technique which reduce the matching time of firewall rules. So, we will pick the minimum early rejection rules which have got the maximum denial rate.

Obviously, the address space of the traffic matching the default deny rule is the addition the address space represented by all previous rules.

3.2 Details of the SA-BSPL Techniques

The packet will be rejected as soon as possible without more examination of the remaining data if the two lists do not share any common rules. Other than that, the following field is checked. The list of matched rules will cross the prior list if the pertinent packet field matches this field, and so on. Due to the early discarding of undesired packets brought about by these 2 major improvements, the packet processing time is expected to improve.

Table: 3.2.1 Table of Early Filtering System

Algorithm	FVSC	PBER	Trial Details
Policy Estimating	To get close to the policy, use the set cover algorithm.	Boolean expression can be used to estimate the policy.	There isn't any estimating done. In a other approach, the exact policy is stated.

Off-Line Phase	Creating RR rules	Create a BDD policy tree	Create policy hash tables, then splay
Traffic Flow Security	RR rules added/removed according to traffic statistics	Traffic statistics are used to alter the depth and layers of BDD tree traversal.	The use of splay trees makes the method adaptable to fluctuations in traffic. But no traffic numbers are mentioned.
Policy Execution	Packets are delivered to the original policy if no choice is made using RR rules.	If a certain BDD depth cannot produce a decision, the packet is routed to the initial policy.	The initial policy was represented using hash tables and splay
Filtering System	Early filtering for rejected packets	Early filtering for rejected and accepted packets	Early packet rejection and accept filtering.
Limitations	Appropriate for simpler security settings with little variation in field values.	Appropriate for extensive and complex security policies.	The data storage can be increased by the values of range fields into prefixes using prefix conversion technique.

Logically, when the rules were not satisfied the data packets then the data packets will be removed as early as possible to reduce the memory consumption according to the protocol. The early Rejection Rules can upgrade combinations of values from common fields. I have provided that the firewall laws are easy to find when the number of field values are fairly small. For illustration. If every positive rule uses a specific subnet or port

number as the destination, packets without this address or port can be safely rejected without more delivery.

As,

Definition 1: Let $\mathcal{V}(f_j) = \{\mathcal{V}_{\mathcal{K}} | \exists \text{ a policy provision that has the value } \mathcal{V}_{\mathcal{K}} \text{ for } f_j\}$, then $S_k^j = \{r_i | f_j^{r_i} = \mathcal{V}_{\mathcal{K}}, i = 1 \dots n, j = 1 \dots 5\}$, where $f_j^{r_i}$ represents main value of field j in the rule i.

Intuitively, S_k^j is the set of rule that has the same value as $\mathcal{V}_{\mathcal{K}}$ in field f_j .

A bunch of different sets (S_k^j)

in all fields of decision laws is alike to the number of different needs ($\sum_{j=1}^5 |V(f_j)|$).

Finding a subset of the S_k^j 's is the difficulty because every legal provision now belongs to at least one of them.

Definition 2: Let the A signifies the set of all possible S_k^j , and let $A' \subset A$ presents a selection of S_k^j 's such that $\bigcup_{S_k^j \in A'} S_k^j = S$.

This means that all rules in the regulation are addressed by the set of rules.

Formula 1: After NP-complete, there is a problem of identifying a collection of minimal size field values such that each law rule has at least one of these field values. The set cover issue with the component's low frequency serves as an instance of this (if the frequency is exactly 2) Refer to Using an approach, we will create an RR for the whole proof.

So, there would be a rejection term (RT) for each $S_k^j \in A'$ that will construct RR respectively. This paper, where $Pkt(f_j)$ is the value of field f_j in packets to be examined,

$$RR = \bigwedge_{S_k^j \in A'} (Pkt(f_j) \neq v_k)$$

Figure: 3.2.2 Rejection Rules

is used interchangeably by RR and A' . A standard law, for example,

$$RR = (DPort \neq 80) \wedge (DPort \neq 20) \\ \wedge (DAddr \neq 15.16.17.18) \wedge (Proto \neq UDP)$$

Figure: 3.2.3 Rejection Rules in Standard Laws

3.3 Host Based Firewall

A host-based firewall is a type of firewall software that is installed on a single computer or other networked device. These kinds of firewalls provide granular protection for individual hosts against viruses and malware while also limiting the spread of these damaging infections across the network.

The different conditions of this firewall is It's faithful to the Local Configuration & it Travels with the only Computer it's connected.

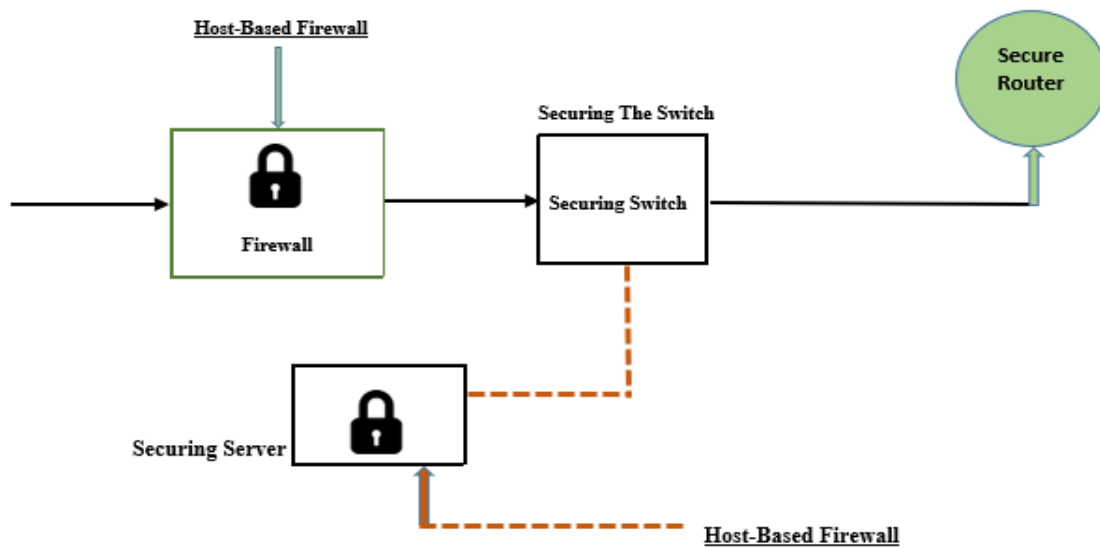


Figure: 3.3.4 Host Based Firewall Working Procedure

A firewall program that is installed on a single computer or other networked device is known as a host-based firewall. These kinds of firewalls prevent the network-wide spread of these harmful diseases while also providing granular protection against viruses and malware for specific hosts.

3.4 Data Collected Using Optimizing Rejection Address Space

Since it is nearly impossible to look for the minimum size solution as the size of the rule increases. In fact, we need other alternatives for detecting different types set of flows which have been rejected. The ratio for the first one is around $1+\ln(|S|)$, For the five fundamental firewall rules-protocol, source-IP, destination-IP, source Port, and destination Port-the first method employs strict integer programming and yields an f-estimate ratio, where f is the maximum number of subsets to which each element may belong.

For almost all policy size,rules the latter algorithm is better because it delivers better calculation result, but both results can help to generate more effectual solutions.

Dynamically Rule Selection

Here, I am proposing to discuss about a calculating technique that generates a group of "A" (i) to reject data packets. Then again, locating the perfect answer to denial is by no means easy, and this results in perplexity. Therefore, the phase's awareness is on designing each issue using arriving packet reports and policy information to establish the right collection for rejection regulations RRS. In the actual international setting, not one of the positive checking techniques is effective, yet all of them perform nicely in less controlled situations. This is usually the case because early rejection would benefit the most from linear search strategies. Assuming that each person has the equal risk of rejecting a packet and that simplest the default policies follow to rejected visitors

Take the proportion of site visitors that may be brushed off early average and the percentage of site visitors with a purpose to be early conserved the use of RRs. The range of rules for rejection need to be installed by using, with the intention to lower the standard number of comparisons within the early rejection regulations.

$$\frac{n}{2}(1 - \delta_{\text{inf}}) + n\delta_{\text{inf}} > \frac{r}{2}\delta_r + (r + \frac{n}{2})(1 - \delta_{\text{inf}}) + (r + n)(\delta_{\text{inf}} - \delta_r)$$

Figure: 3.4.5 Dynamic rule formula

Then it will take us to,

$$r < \frac{2n\delta_r}{2 - \delta_r}.$$

Figure: 3.4.6 Expected Result

Use of **Algorithm-1** here,

```

< S, A > ← Convert(Policy Rules).
rmax =  $\frac{2n\delta_{est}}{2 - \delta_{est}}$ 
i ← 0
repeat
    A' ← Approx_SetCover(S, A)
    RRSet ← Build_Rules(A')
    i ← i + 1
until i ≥ rmax or A' is empty
sort(RR_Set) by size, shorter first
r ← 1
Active_RR_list ← RR_Set(r)

```

Figure: 3.4.7 Formula of Algorithm-1

After this we reach to, Algorithm 2 Dynamic Rule Selection

```

IF  $\Delta\delta_r > \frac{c(1 - \delta_r + \gamma_r)}{2n}$  then
    Active_RR_list → rule r {Remove last rule, rule r}
    r ← r - 1
end if
if |RR_Set| > r then More rule to be added}
    r ← r + 1
    Active_RR_list ← RR_Set(r)
end if
sort Active_RR_list according to hit frequency

```

Figure: 3.4.8 Formula of Algorithm-2

The first, second, and third terms on the right side of the inequality reflect the average values of early rejection rules rejection, law rejection, and standard rules rejection, respectively. The left term in the inequality displays the average number of correlations per packet despite early rejection. As long as new rules continue to reject packets, we can see that the boundary for all values of increases. Therefore, as long as the bound is met, we may add more RRs to enhance filtering.

To judge the result of adding a particular RR, more careful analysis is required. Now let α be the traffic portion approved by the legislation, and after adding Υ early denial rules, we have β_Υ , δ_Υ and γ_Υ respectively, the traffic segment denied by the RRs, the law denies rules, and the default rule 3

Now, after adding the Υ RR, we can determine the typical quantity of comparisons per packet as follows:

$$A_r = c.r\left(\frac{\delta_r}{2} + \alpha + \beta_r + \gamma_r\right) + n\left(\frac{\alpha + \beta}{2} + \gamma_r\right)$$

Figure: 3.4.9 Formula of Determining the Average Number.

So where c is the overall cost of the RR, now it's usually related to the number of terms included with the statute. now, $\partial\delta/\partial\gamma > 0$, $\partial\beta/\partial\gamma > 0$, $\partial\Upsilon/\partial\gamma > 0$, $\alpha + \beta_\Upsilon + \gamma_\Upsilon + \delta_\Upsilon = 1$.

Now, Let $\Delta \delta_\gamma$ be the portion of the total traffic the r^{th} RR refuses. Then we can clearly see that,

$$A_r = c.r\left(\frac{\delta_r}{2} + \alpha + \beta_r + \gamma_r\right) + n\left(\frac{\alpha + \beta}{2} + \gamma_r\right)$$

Figure: 3.4.10 Portion of the total traffic analysis System.

It is meant to flood the victim network with unwanted traffic in order to prevent actual traffic from getting through to the primary victim system. There are mainly two types of attacks that use up bandwidth. The first type of assault is a flood attack when a victim system is chosen and additional victim systems are exploited to saturate it with traffic. Eventually, the bandwidth of the victim system would get congested.

Data plus a header make up a packet. By applying the rules successively, an IL program analyzes whether a certain packet is approved or rejected. The result depends on the first rule header that matches with the packet header (Accept or Deny).

After matching packets, Algorithm-3 For Early Rejection Filter,

```

Match Packet against Active_RR_list (w' shortcut evaluation)
if packet matched any RR then
    reject packet
    INCREMENT  $\Delta\delta_i$  of matched rule
    INCREMENT  $\delta_r$ 
else
    send packet to normal filtering process
    INCREMENT  $\gamma_r, \alpha_r$  or  $\beta_r$ 
end if
DECREMENT Window_Expired
if Window_Expired = 0 then
    Call Dynamic Rule Selection
    Window_Expired = Window
end if

```

Figure: 3.4.11 Following Condition for Early Rejection.

To explain the implementation of the law of γ^{th} RR: $A_\gamma - A_{\gamma-1} < 0$ must be in place. Therefore, we can derive the following condition from (3) and (4):

$$\frac{\Delta\delta_r}{c} > \frac{\frac{\alpha + \beta_r}{2} + \gamma_r}{n}$$

Additionally, it can also be translated as

$$\Delta\delta_r > \frac{c(1 - \delta_r + \gamma_r)}{2n}$$

To improve runtime evaluation based on the kinds of statistics stored at the firewall.

The added rule can be evaluated after each time window based on (5) or (6) to decide whether to use or delete the r th RR as defined in Algorithm 2. Because shorter rejection rules are easier to evaluate and cover more ground than longer ones, we should first add them in order of length.

As the traffic data demonstrate each RR's effectiveness, they will be utilized frequently to optimize the procedure by continually selecting the most crucial rules for early rejection.

In addition, RTs is sorted system according to its efficiency within each rule; to maximize running time by evaluating each RR's shortcut. The three algorithms demonstrate the early rejection module's main operations. In Algorithm 1, we use different solutions to the set cover problem to construct the candidate rejection rule list.

Algorithm 2 is responsible for the periodic removal of rules depending on the output loss of each rule.

Set of Algorithm-2 updates the records necessary for early rejection in addition to displaying the early rejection zone in compared to conventional packet filtering. For every packet window, it also triggers the dynamic rule selection method to update the active early rejection rule listing. To keep away from computing new answers at runtime, which is noticeably costly to utilize in actual time, an Algo-1 pre-processing step is used to generate a set cowl.

To avoid computing new solutions at runtime, which is exceedingly expensive to utilize in real time, an Algo-1 pre-processing step is used to generate a fixed cover.

3.5 Statistical Analysis on Optimizing Rejection

The firewall are designed to let valid packets flow. By comparing a packet's header information with the established regulations, it may be determined whether or not it is legitimate. A packet that is rejected by the default rule will consume significantly more computer resources and processing time than packets that are subject to several rules. If the attacker is aware of how the firewall works, it may get stuck and consume a lot of

resources handling these packets. On firewall devices, certain early packet rejection mechanisms for unwanted packets are suggested as defense against this attack.

In Figures 1 and 2, The differences between the default firewall and the firewall with an extra early packet rejection module are displayed. The basic firewall rules instantly filter the packets, as seen in Figure 1. Whereas Fig-2: The packet is filtered by a module that performs early packet rejection before filtered by the original firewall rules.

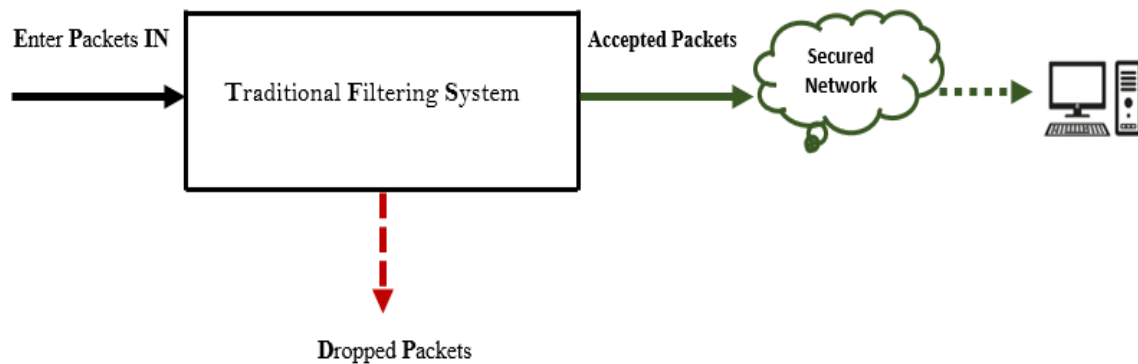


Figure: 3.5.12 Traditional Firewall Rule Matching

3.6 Regular Traditional Based Packet Filter

Traditional firewall has a rule-set to execute actions according to incoming and outgoing packets of network. The data packets are mainly checked in the listed rule of a firewall and then if the packets match with any rule, after that it will be passed according to the predetermined allow or deny action. It consists a huge number of rule-set which are well-ordered according to importance. How the original firewall and the firewall with an additional early packet rejection module differ from one another is seen in Figures 14 and 15. Fig. 14 shows that the first firewall rules directly screen the packets, but Fig. 15 shows that the early packet rejection module filters the packet first and then the initial firewall rules filter it.

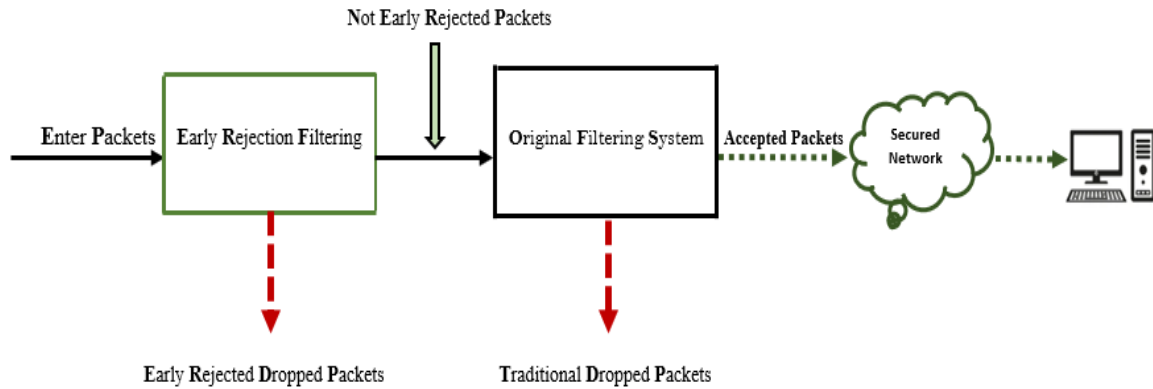


Figure: 3.6.13 Proposed Protocol Based Early Rejection Firewall Model.

3.7 Proposed Methodology on Protocol Based Packet Filter

Protocol-based firewall mainly works as a subsection of traditional firewall which means it divides the long traditional firewall into many sub-firewalls classified by protocol. Here incoming traffic is sorted against various types of protocol after that it goes into the exact firewall for further rule matching. In this model, firewalls are shorter and also effective for rule matching because in the protocol-based firewall the rules are classified by the protocol.

Table: 3.7.2 Proposed Protocol Filter Model Table

Direction	Source	Destination	Protocol	Source Port	Destination Port	ACK set UDP	Rules / Notes
In	Ext	Int	UDP	>1023	1812	[2]	Authentication query for external client to internal Radius server.
Out	Int	Ext	UDP	1812[1]	>1023	[2]	Authentication response internal Radius server to external client.
In	Ext	Int	UDP	>1023	1812[2]	[2]	Authentication notification for external client to internal Radius server.
Out	Int	Ext	UDP	1813[3]	>1023	[2]	Authentication response internal Radius server to

							external client.
Out	Int	Ext	UDP	>1023	1812[1]	[2]	Authentication query for internal client to external Radius server.
In	Ext	Int	UDP	1812[1]	>1023	[2]	Authentication response external Radius server to internal client.
Out	Int	Ext	UDP	>1023	1812[2]	[2]	Authentication notification for internal client to external Radius server.
In	Ext	Int	UDP	1813[3]	>1023	[2]	Authentication response external Radius server to internal client.

Packet Filtering Characteristics

UDP port 1812 and UDP port 1813 are used for RADIUS authentication and accounting, respectively. The ports 1645 and 1646 that were utilized in early RADIUS implementations are no longer in use.

Here we can see there is few equations as (1) UDP has no ACK equivalent, (2) Early operation or implements might use 1646 & (3) Early operation or implementations use 1645.

There is never a need for router discovery to pass via one. Router discovery is solely meant to communicate details about the local network. Therefore, it is secure and recommended to exclude it from all packet filtering routers.

Recent studies indicate greater awareness of and developments in firewall functioning and technology. asks for: The firewall is one of the most used network security although it has limits and can be abused by hackers. It may be able to protect the network from external threats, but it's not designed to do that if the threat or hacker occurs from within the network.

There has been an improvement in knowledge about firewall technology, operation, and improvements, according to firewall study. asks about: The firewall is one of the most widely used network security solutions, despite having limitations and being vulnerable to malicious hacking. However, if a threat or hacker comes from within the network, that threat or hacker is not designed to achieve that. The network may be safeguarded against external threats.

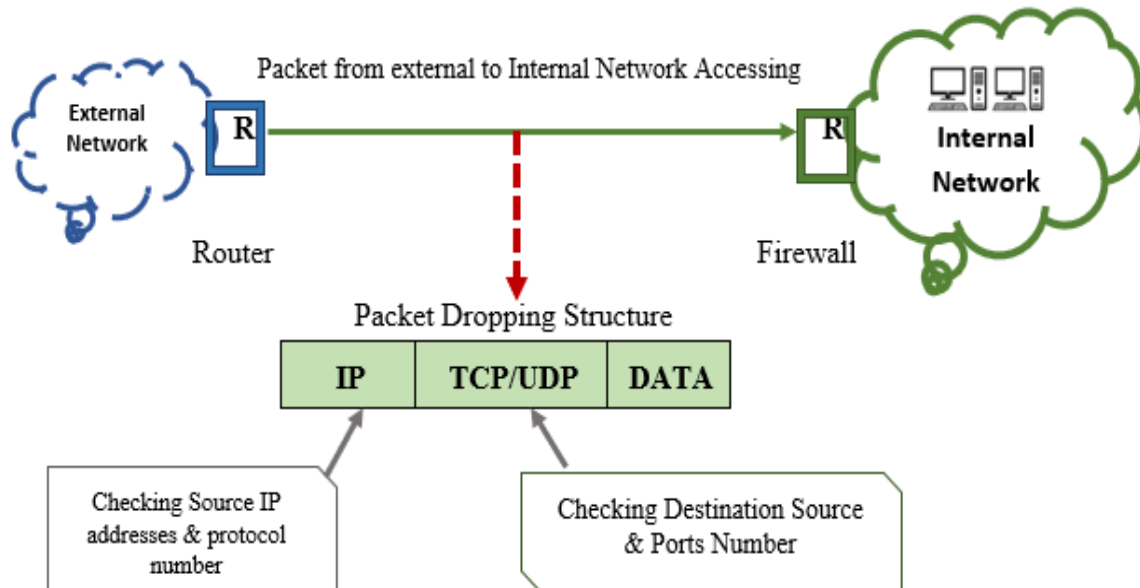


Figure: 3.7.14 Packet Filtering Structure for Internal Network.

3.8 Implementation Requirements for Protocol Based Firewall

I'm trying to use the FLIP tool to integrate the methods and algorithms discussed in this chapter into a software module. Conflict analysis and the generation of basic rules are performed using the implementation tool FLIP. Building the implementation in Java will be preferable. I'm trying to describe my evaluation method in this part for the analysis of the FLIP's usability and performance.

The FLIP tool has been used to try to incorporate the ideas and methods described in this chapter into a software module. Fundamental rule generation and conflict analysis are done using the implementation tool FLIP. The implementation should be created in Java. I'm aiming to lay out my evaluation process in this part as I look into FLIP's usability and usability.

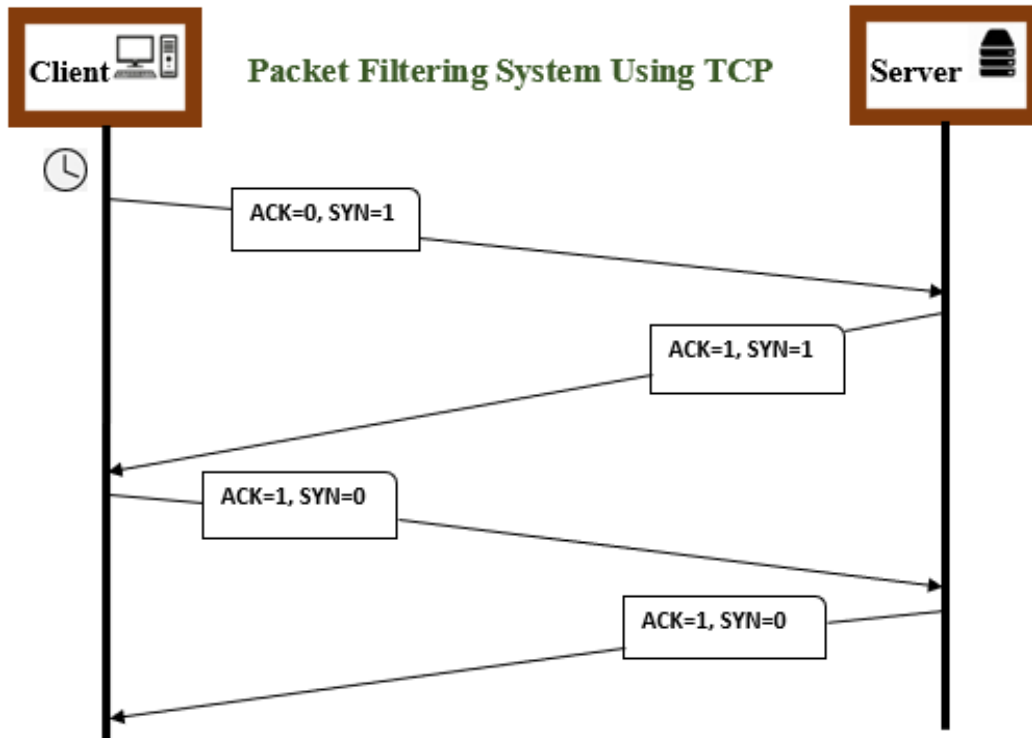


Figure: 3.8.15 Packet Filtering Structure for Internal Network.

SYN is activated during the first two packets of a connection to establish sequence numbers. In order to provide the second packet with a number to acknowledge, the initial packet of a connection must have SYN on and ACK off because it is not a response to anything. More information about sequence numbers is given in the section that follows.

In order to establish sequence numbers, SYN is engaged during the first two packets of a connection. The initial packet of a connection must have SYN on and ACK off as it is not a response to anything in order to provide the second packet a number to acknowledge.

This is because several policy groups are applied to each domain crossing; hence, different policy groups' rules must be reviewed independently in order to detect conflicts.

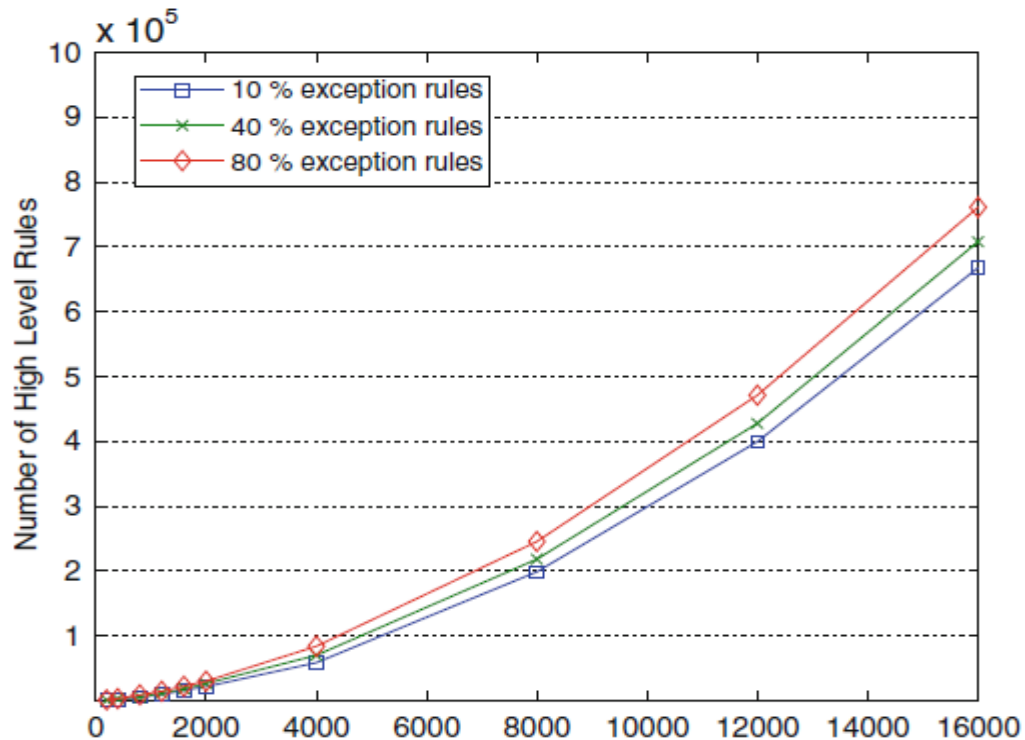


Figure: 3.8.16 These are the approximate amount of generated through FLIP

In an effort to assess FLIP's scalability, I'm trying to run some simulations. I've created three sets of definitions for policy groups and domains. The target domains on which the policy groups are used do not cross in the first set. In the second group, 20% of the target domains overlap, and in the third set, 40% of the target domains overlap.

The B-class IP network that my proposal simulates is the aim, and we may suppose that each policy group clearly defines 20 high-level rules and applies to a different domain. FLIP is the main tool I am trying to use to assess these policy groups and look for conflicts with other policy groups. The outcome makes it clearly apparent that the time required to identify conflicts grows as the number of intersecting domains rises.

This is the result of the fact that each domain intersection is applied to many policy groups, and that in order to identify conflicts, the rules from various policy groups must be examined together.

CHAPTER 4

PROPOSED PROTOCOL BASED FIREWALL

4.1 Traditional Filter Firewall

Traditional firewall has a rule-set to execute actions according to incoming and outgoing packets of network. The data packets are mainly checked in the listed rule of a firewall and then if the packets match with any rule, after that it will be passed according to the predetermined allow or deny action. It consists a huge number of rule-set which are well-ordered according to importance. In the character ‘Traditional Firewall Rule Matching’ it’s shown as that, there are many rules to execute. In a result “if any packet does not match with the rule then it will cross a long path to match the default discard rule”.

A traditional firewall is designed for the flow of traffic that goes in and out of a network.

As example, if one packet matches with the rules then it would cross a long path of rule list. It consumes more time and processing overhead to pass the data into the secured network. In traditional firewall there is no group based on any section of 5 tuples.

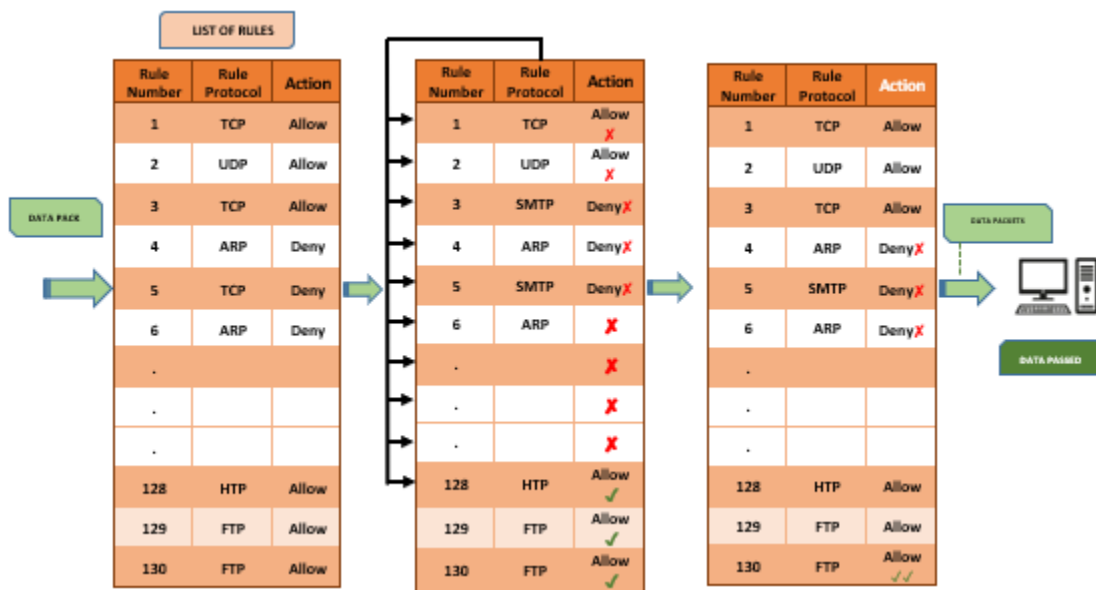


Figure: 4.1.1 Traditional Filter Firewall

4.2 Experiment on Protocol Based Packet Firewall

Protocol-based firewall mainly works as a subsection of traditional firewall which means it divides the long traditional firewall into many sub-firewalls classified by protocol. Here incoming traffic is sorted against various types of protocol after that it goes into the exact firewall for further rule matching. In this model, firewalls are shorter and also effective for rule matching because in the protocol-based firewall the rules are classified by the protocol. For example, (If a TCP packet goes for the TCP firewall rule set, the UDP packet will go for the UDP firewall rule set in the same way, then packets will match the rules of their specific firewall rule set. This method will help to reduce the matching time and processing procedures.

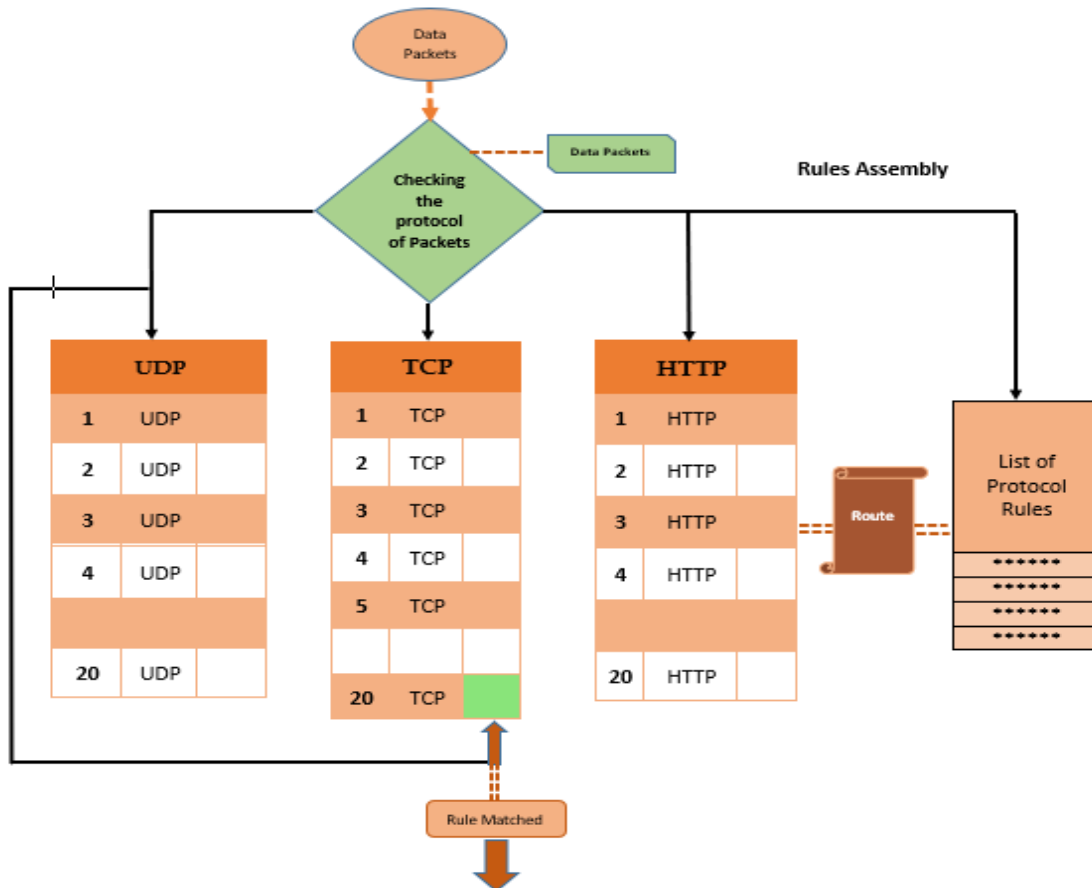


Figure: 4.2.2 Protocol Based Firewall Rule Matching

There are many types of firewalls have been used to secure the internal network from the outside world.

Firewall matches network traffic with the rule set in its table. Once the rule is met, the network traffic related action is applied. Handling out going traffic from source is little bit easy and less important than handling the incoming traffics. Incoming traffic can come from different end point among them some are dependable and some of them are false.

In the traditional firewall all protocol (UDP, TCP, ICMP, HTTP etc.) have same firewall rule-set. But the rules are not branded according to protocol before. As a result, when the firewall started matching the rules from the list for the different packets of different protocols, then the packets need to traverse a long path to match the rules because all the rules for different protocol are in the same list. In my proposed model I have mainly categorized the incoming data packets according to protocol and bypass them to the protocol-based firewall for matching the rules. In this theory, I have mainly divided the long firewall rule-set into many protocol-based firewall rule-set which will be shorter and real than the previous ones. Because now in my model the packets will navigate less path to match the firewall rule-set.

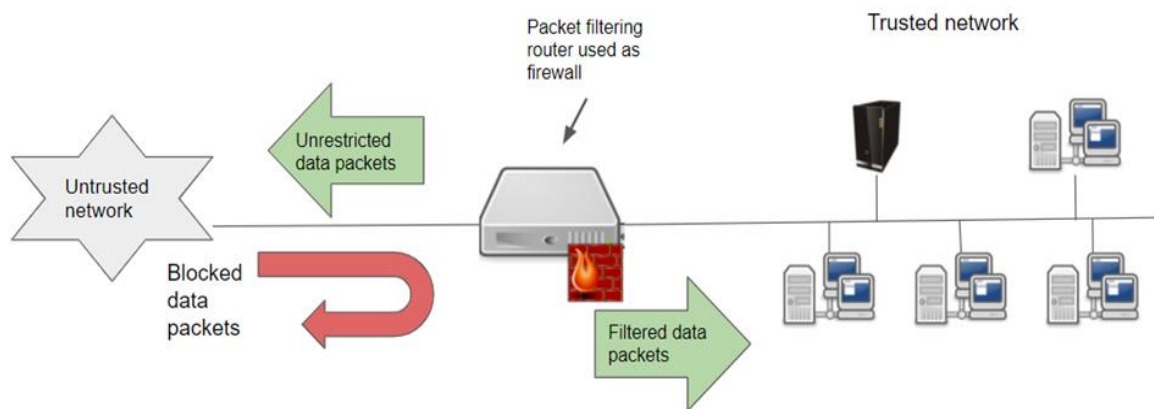


Figure: 4.2.3 Network Accessing Protocol analyzing System.

4.3 Experimental Result on Research

In this section, we analyze the Fire Wall using both simulations with TCP and UDP and real-world case studies. We initially conducted a number of simulations runs to evaluate the accuracy and scalability of our approximation algorithms in comparison to the best solutions. In order to ascertain how well Fire-Blanket will perform in a real-world network environment, we use Fire Wall to evaluate the campus network for an academic institution. From there, we may try to design security configurations for various budgets.

I'm trying to do a sizable number of simulations and packet trace analysis tests to investigate the performance gain of my proposed rule order optimization method in firewall filtering. Using my adaptive rule optimization approach instead of linear packet matching with the original, unoptimized rule list results in an average decrease in the number of packet header comparisons against the policy rules, that is how the performance gain is determined.

4.4 Research Outcome

In order to generate a set of rules that can swiftly reject a large number of unwanted packets, this approach looks at the firewall policy rules. It is an issue, and the answer is to use an approximation strategy that pre-processes the firewall policy offline to produce a number of almost perfect solutions.

In the research when the router arranged the rules according to the number of significant bits in the source address, the more specific rules are implemented first. So, regulations that apply to more precise IP source addresses would take precedence over those that apply to less particular IP source addresses. The regulations would then be implemented in the Early Rejection Rules order in this situation. If the rules are applied in order then,

Table: 4.4.3 Source of Traffic Incoming

Rule	Source Address	Destination Address	Action
Y	10.1.99.0/23	188.16.0.0/16	Deny
X	10.0.0.0/8	188.16.6.0/24	Allowed
Z	Any Other	Any Other	Deny

Here are a few examples of packets with the new results if the rules are applied in the sequence of Early Rejection rules; we point out the differences between the new results and the previous case.

Table: 4.4.4 Source of Traffic allowing & Disallowing.

Packet	Source Address	Destination Address	Desired Action	Actual Action
1	10.1.99.1	188.16.1.1	Deny	Deny(Y)
2	10.1.99.1	188.16.6.1	Allowed	Deny (Y)
3	10.1.0.1	188.16.1.1	Deny	Deny (Z)
4	190.168.4.3	188.16.1.1	Deny	Deny(Z)
5	10.1.1.1	188.16.6.1	Allowed	Allowed(X)

4.5 Discussion on Firewall Rule

To improve packet matching capabilities in firewalls, I am representing the issue of optimizing packet filtering rules. Here, the general problem in polynomial time is challenging to solve. We provide a heuristic method of estimation that runs in polynomial time and achieves results that are close to the best for the most well-known firewall rules. IP-Sec filtering rules will be the focus of this discussion in this part, but others may model and evaluate other filtering policies using the framework that is being provided.

Definition-1: An access policy, or $\{P = R_1, R_2, \dots, R_n\}$ is a list of n filtering rules that decide the proper course of action to be taken with regard to each incoming packet.

Definition-2. A filtering rule, R_i , is made up of a set of restrictions on a set of k filtering fields, $\{F = f_1, f_2, \dots, f_m\}$, and an action, chosen from the set of all actions. A. Each rule may be expressed. It's crucial to note that we see this method as an offline procedure that occurs before the actual deployment of the filtering rules in the firewall rule table. We place greater emphasis on accuracy and usability than computation complexity and algorithm optimization since the "Firewall Policy Advisor" is a tool available. When the

filtering rules are eliminated, the rules may be arranged in the best way possible by simply sorting them according to their weights in a non-increasing manner. It's not an option since weighted ordering of the rules could be incompatible with rule dependencies.

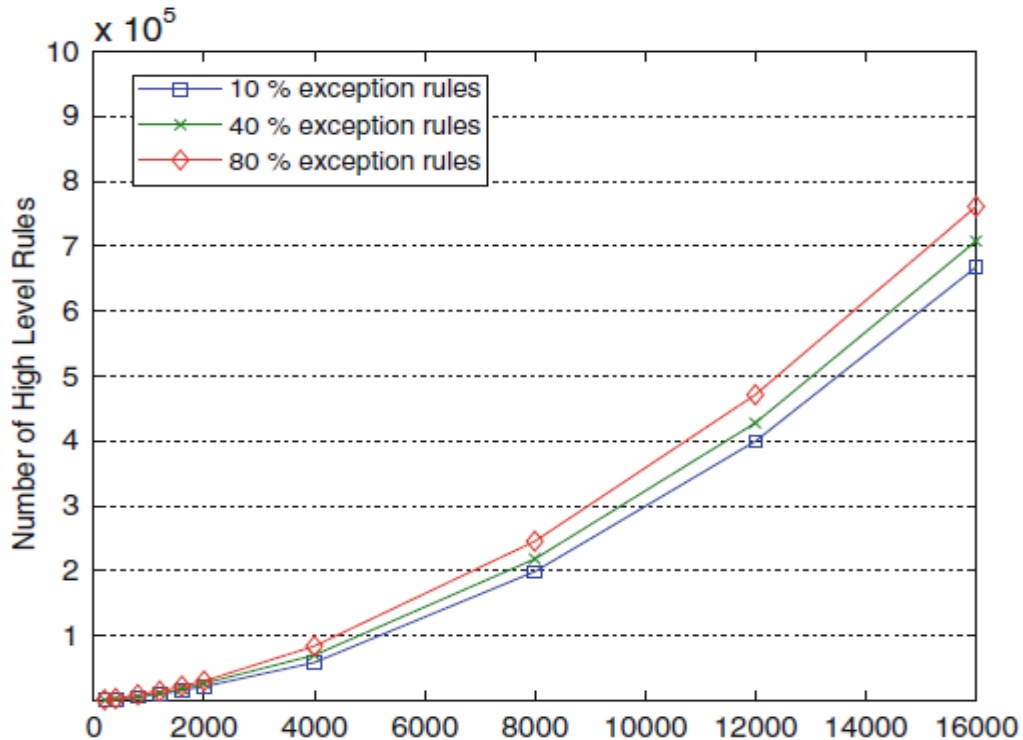
I've attempted to show how to improve packet filtering rules here so that we can see how firewalls may improve their packet-matching skills. The more general polynomial-time problem is challenging to resolve in this case. We provide a heuristic estimate method that yields results that are almost optimum and runs in polynomial time for the most popular firewall rules. Although other people may create and evaluate alternative filtering policies using the provided structure, the discussion in this part will be focused primarily on IP-Sec filtering rules.

CHAPTER 5 OPTIMIZING RULES IN ORDERS

5.1 Impact on Society of Protocol Based Filtering Firewall

The maximum number of children one parent is capable of having is how we regulate the inheritance links between groups (degree). Greater tree depth results from lower degrees. By extension, this means that kid groups receive additional regulations from parent groups. In this experiment, it is assumed that 20% of domains are IP-range-defined and 20% of high-level rules are exception-free. Both policy groups and domains are created by us. In order to execute the experiment with varied degree values, we fix the relationship between the policy group and domain. The outcome of the quantity of low-level regulations is depicted in Fig. 3.10. We might infer from this conclusion that the degree of change has little impact on the total number of rules.

Table: 5.1.5 Impact of Exception Rules for Early Rejection Protocol.



Since several rules are necessary to adequately represent the user's goals, it is crucial in my research to translate exception rules. By examining the number of rule modifications ©Daffodil International University

in proportion to the percentage of exception rule changes, I'm aiming to illustrate the level of 5, 20% of IP-defined domains. Figure 5.1.1 shows the result. This graph shows that the overall number of low-level rules does not increase much when exception rules increase.

5.2 Impact on Environment of Matching Recency

When the actual packet matched quality significantly deviates from the ideal matching expected, this form of update is instantly initiated. I am using the routine factor ϵ to measure the nonconformity from the optimal average number of matches measured in the last update of the rule list for the actual average number of matches. The period between two successive rule list update occurrences is referred to as the update interval. Therefore, ϵ is given using the equation below:

$$\epsilon = \frac{\sum_{i=1}^n p_i d_i}{\sum_{i=1}^n q_i d_i} - 1 \quad (12)$$

Here d_i is the R_i rule length, p_i and q_i are the R_i packet proportions, respectively, in the current and previous update pause. Although this equation efficiently measures It costs a lot of money to monitor the deviation from optimal matching for each packet that the firewall receives in real time. Therefore, I have used an exponential moving usual \bar{h} to calculate the average number of packets matches as follows:

$$\bar{h}_j = (1-\omega) \bar{h}_{j-1} + \omega h_j \quad (13)$$

Where h_j is the filtering rule depth consistent to packet j . The quality variance can therefore be valued at any time using \bar{h} as follows from the above equations:

$$\epsilon = \frac{\bar{h}}{\sum_{i=1}^n q_i d_i} - 1 = K\bar{h} - 1 \quad (14)$$

Where $q_i d_i$ is a constant when determined when optimizing the rule list. The deviation factor is calculated after each packet matches the rule list and a new optimized rule list is built if its value exceeds a certain deviation threshold. A user configurable limit to determine the maximum appropriate deviation from the optimum average matching is the deviation edge ε_{thr} .

The rule recentness T_i can be represented in the time break as the time t_i ratio in which rule R_i is finally matched in the same interval to the time t_{last} till the last rule is matching in firewall.

$$T_i = \frac{t_i}{t_{last}} \quad (7)$$

$$T_i = \frac{p_i}{P} \quad (9)$$

Due to the locality of matching property outlined in Section 3, rule frequency and regularity are crucial because they show how probable a rule is to match a packet in the future. According to our Section 6 evaluation research, the rule's frequency is more responsive to heavy-weight bulky flows than it is to long-lived burst flows. A weighted average of these two variables is the weight ω_i of rule R_i as follows:

$$\omega_i = (1 - \rho)F_i + \rho T_i \quad (10)$$

The recentness of ratio ρ specifies the weight of the rule would be dependent on the recency of the rule. The traffic types has an impact on the recency factor value as (Bulky vs Bursty) Section 6 further explores the impact of the recency ratio. I have removed the frequency and recency values found in Equations (8) and (9) to avoid the division operations in the calculation of law weights. ω'_i can be extracted from a computationally simplified weight ω'_i of rule R_i as follows:

$$\begin{aligned} \omega_i &= \rho \frac{p_i}{P} + (1 - \rho) \frac{f_i}{P} \\ &= \frac{1}{P} [\rho p_i + (1 - \rho) f_i] \end{aligned}$$

$$\omega'_i = P\omega_i = \rho p_i + (1 - \rho)f_i$$

Notice that ω'_i does not require any division activities, but still tests the value of rule R_i as per other policy regulations. Therefore, in Algorithm 1 ω'_i can be used directly to fix rule weights with much less overhead processing. Note also that the algorithm's weight optimization limit should be multiplied by the total number of P packets.

5.3 Ethical Aspects on Optimal Rules in Ordering Problem

The ideal rule ordering challenge in a protocol-based firewall is to fill a real rule order that receives the fewest packet matches due to an assembly of filtering rules with inter-rule trust. Only when the existing relationships between the rules are preserved is the ordering of rules acceptable. If all packets are in serially matched in order to the policy containing of 'n' filtering rules with d_i as the policies order of rule R_i and d_i is a weight that resembles R_i 's power in packet matching, we can then define **ORO** by the following minimization:

$$\min \sum_{i=1}^n \omega_i d_i \quad (1)$$

In order to solve the ORO problem, it can be formalized as a binary integer program (BIP) as follows.

$$\min \sum_{i=1}^n \sum_{k=1}^n k \omega_i x_{ik} \quad (2)$$

$$\text{Subject to } \forall_i \sum_{k=1}^n x_{ik} = 1 \quad (3)$$

$$\forall_k \sum_{i=1}^n x_{ik} = 1 \quad (4)$$

$$\sum_{k=1}^n k x_{ik} - \sum_{k=1}^n x_{jk} < 0 \text{ if } R_i \rightarrow R_j \quad (5)$$

$$x_{ik} \in \{0, 1\}, i \in \{1, \dots, n\}, k \in \{1, \dots, n\}$$

Where x_{ik} is a binary function, $x_{ik} = 1$ if rule R_i is set to position k in the law, and $x_{ik} = 0$ otherwise. The objective minimize function (2) is similar to the optimization problem described in (1) where the rule R_i depth d_i is given by the $\sum_{k=1}^n k x_{ik}$ expression and

ω_i is the rule weight. The rule weights calculation is defined later in Section 5.1. Constraint (3) ensures that each rule is located exactly at one place, while Constraint (4) ensures that exactly one rule applies to any specific location. Constraint (5) maintains the ordering precedence between dependent rules by certifying that if R_i has a higher precedence over R_j rule (referred to as $R_i \rightarrow R_j$), then R_i 's order in the law will precede R_j . The validation of the above-mentioned BIP allows iterative numeric ways to solve the ORO problem. It is possible to obtain a lower limit as the ORO problem by soothing the binary variables through the linear variables & helps to solve the **BIP** calculations like a issue with linear programming. To minimize the impact of creative explosion and find the best solution, the branch-and-bound technique can be used in connection with the gradient projection method. Nonetheless, a more effective approach is required to calculate the solution, due to the fact that the branch-and-bound technique cannot provide a polynomial computation time in the number of rules. In the section that follows, I outline my heuristic approach to the ORO issue, which can produce a close to optimum result in polynomial time. By simply sorting the rules in non-increasing order based on their weights, the best rule ordering may be discovered after the filtering rules have been eliminated. However, since most firewall policies have dependent rules, simple sorting is ineffective since rule dependencies could clash when rules are arranged in accordance with their weights. The ORO issue is comparable to the scheduling of work for a single machine with historical constraints.

5.4 Optimal Rules Ordering ORO Procedure

Although several approximation algorithms have been put forward to solve the

$1|\beta|n \sum_{i=1}^n \omega_i C_i$ problem of the best guess provides an optimal solution $(2 - \frac{2}{n+1})$,

There are different policies. The range of policies is n . This answer is two times larger than the appropriate one, even though there are between 100 and 1,000 restrictions. Numerous work scheduling theories also rely on simulating the issue as a linear system and solving it, which is a dynamic and assistance-extensive approach for actual firewall application. I have developed a special heuristic estimate technique for using the **ORO**

issue it is more effective and practical for use. in firewalls as a result of these factors. Real firewall filtering strategies consider three key characteristics. The dependency intensity is so profound because of the rule weight distribution, which is notably biased so that a few rules match the majority of internet users.

My heuristic's specifics are described in Algorithm 1. The rule list to be optimized and an optimization limit, also known as the active rules, are inputs to the algorithm. The optimization limit indicates an upper bound on the overall weight of the selected rules throughout the optimization process. Then a Max Heap of weight-based sorting criteria is generated.

The Max-Heap data structure keeps the item with the greatest weight at the top so that it may be restored in constant time.

Algorithm 1 Optimize Active Rules

```
1: weight  $\leftarrow$  0
2: H  $\leftarrow$  Build Max Heap (H, rule list)
3: while H is not empty do
4:    $R_b \leftarrow$  Heap Extract Max (H)
5:   for each  $R_d \in \{\text{rules dependent on } R_b\}$  do
6:     if  $R_d \notin$  active rules then
7:       current  $\leftarrow$  List Tail (active rules)
8:       while current  $\neq$  nil do
9:          $R_a \leftarrow$  List Get (current)
10:        If weight ( $R_a$ ) < Weight ( $R_d$ ) and  $R_a$  not reliant on  $R_d$  then
11:          List Remove (rule list,  $R_d$ )
12:        end if
13:      end for
14: List Insert Tail (active rules,  $R_b$ )
15: List Remove (rule list,  $R_b$ )
16: if weight  $\geq$  opt_limit then
17:   break
18: end if
19: end while
20: for each  $R_m \in$  rule list do
21: List Insert Tail (active rules,  $R_m$ )
22: end for
23: return to active rules
```

This is the policies are successively selected from the pile in descending order of their weights. The optimized energetic rule listing (traces 6–17) gets rid of each base rule from the heap together with all structured guidelines that have to come earlier than it, and then installs them in the right order. Every based rule is protected in the powerful rule set, which arranges the regulations inside the list in keeping with their weights in descending order. The optimized list is then up to date to encompass the simple rule that changed into eliminated from the heap. The algorithm optimizes a set of n rules in $O(n^2)$ running time, thus, theoretically. It leads to a decrease in the total number of iterations in the outer loop, as well as a reduction in the heaping time. Therefore, when considering only the most effective rules, using the optimization weight limit greatly reduces the number of iterations in the inner while loop. Here in this analysis in Section 6 shows that optimizing the most effective 25-45% of the policy rules is necessary. Clearly, the space complexity is bounded by $O(n)$ as the algorithm only carries two lists of filtering rules.

To decrease the time required to insert the rule, my approach employs a dual-linked rule list operation. Also, the collection of rules preceding each policy rule is easily accessible via a linked list of pointers established during the policy pre-processing step. Such pre-processing occurs only when the firewall is booted or a filtering rule is modified.

5.5 Dynamic Rule in Order for Protocol

I have described the basics of the way the guideline order optimization method is applied in real firewalls here. A version tool that activates the best optimization set of rules while it's miles important based on the maximum current visitor's systems is likewise blanketed inside the implementation.

The process for calculating filtering rule weights to represent the matching cost of each rule relative to others is discussed. I am trying the rule as like.

As,

- For each policy group **P** do
- **P, state** ← unfinished
- **end** for
- for each P with **P, state** = unfinish **do**
- rule Generation (P)
- **end** for

Computation of rule weights

Every rule in the filtering policy is assigned a weight that matches the traffic controlled by the firewall to indicate the authority of this rule. Two conditions are used to determine the rule weight: **(1)** Matching rate to determine how often the rule was triggered & **(2)** Using matching timeliness, you may find out when the rule was last triggered during the packet's matching phase.

5.6 Integration with packet matching

The optimized rule list is created to fit incoming packets to the firewall based on the chosen rule weights. When the future distribution of traffic over filtering rules is exactly in line with the distribution when the list is produced, the reduction in matching is at its greatest. The primary reason is that because Internet traffic flows across filtering rules are dynamic, it is necessary to dynamically alter rule weights in order to replicate the present distribution.

As a result, I'm suggesting two different types of rule list updates: (1) performance-based triggered updates, and (2) periodic updates based on time. By limiting these changes to prevent processing overhead, it achieves the objective of dynamically adjusting the rule weights to create an order that is as near to the ideal as feasible.

Algorithm 2 Matching Packet

```
1: The packet counts  $\leftarrow$  packet count + 1
2: time  $\leftarrow$  Get Current Time ()
3: H  $\leftarrow$  Get Packet Header(p)
4: rule  $\leftarrow$  Match Rule (H, rule list)
5: if rule  $\neq$  nil then
6: action  $\leftarrow$  rule action
7: rule frequency  $\leftarrow$  rule frequency + 1
8: rule recency  $\leftarrow$  packet count
9: action  $\leftarrow$  DEFAULT_ACTION
10: end if
```

11: if the solution is = ALLOW then
 12: Forward Packet(p)
 13: else
 13: if $\epsilon > \epsilon_{thr}$
 14: Calculate Rule Weights (rule list)
 15: Optimize Active Rules (rule list, OPT_THR)
 16: for every single rule \in list of rules
 17: rule frequency $\leftarrow 0$
 18: rule recency $\leftarrow 0$
 19: end for
 20: count of packet $\leftarrow 0$
 21: update \leftarrow time
 22: end if

5.7 Performance-Based Triggered Updates

If the observed packet matching quality differs noticeably from the desired matching, this form of update is rapidly initiated. The performance component is used to calculate the difference between the real average number of matches and the suitable common variety of fits as described via the most current rule listing update.

The update c programming language is the period of time among 2 successive rule listing update activities. Consequently, it the usage of the following equations:

$$\epsilon = \frac{\sum_{i=1}^n p_i d_i}{\sum_{i=1}^n q_i d_i} - 1$$

Where d_i is the R_i rule length, p_i and q_i are the R_i packet proportions, respectively, in the current and previous update pause. Despite measuring the deviation from ideal matching effectively, this equation is quite costly to use for each packet that the firewall receives in real-time. Therefore, we use an exponential moving usual \bar{h} to calculate the average number of packets matches as follows:

$$\bar{h}_j = (1-\omega) \overline{h_{j-1}} + \omega h_j$$

Where h_j is the filtering rule depth consistent to packet j . The quality variance can therefore be valued at any time using \bar{h} as follows from the above equation:

$$\varepsilon = \frac{\bar{h}}{\sum_{i=1}^n q_i d_i} - 1 = K\bar{h} - 1$$

Where $q_i d_i$ is a constant when determined when optimizing the rule list. The deviation factor is calculated after each packet matches the rule list and a new optimized rule list is built if its value exceeds a certain deviation threshold. A user configurable limit to determine the maximum appropriate deviation from the optimum average matching is the deviation edge ε_{thr} .

5.8 Sustainability Plan for Packet matching algorithm

Algorithm 2 specifies how adaptive rule order optimization is implemented in a typical firewall packet matching module. The algorithm executes the common packet matching procedure by comparing the packet header to the rule list and performing the associated filtering action. As packets are received, the global packet counter will increase, the rule frequency and recentness will change, and Equations 13 and 14 will be used to calculate the current average number of matches and output variance. The procedure for enhancing the rule order raised after computing the new rule weights is employed if the current variance exceeds the appropriate threshold or the most recent periodic update interval has passed. Then the global packet counter is reset to zero (Lines 22-26) as well as the frequency and recency of each law.

It's important that the other processing applied to the packet matching algorithm presents minimal overhead processing. On the one side, each packet's storage requires just six mathematics operations, when reviewing the rule list, caused and frequent updates are rarely performed.

CHAPTER 6

FUTURE RESEARCH FOR IMPLEMENTATIONS

6.1 Summary of the Study Research

The Firewall and Early Rejection summary is a type of first-level security procedure for the private network from outsiders and attackers set by the creation of a firewall. The sorts of attackers that are active online are briefly described in this section. There are many other ways to classify these assailants; however, we can't really do credit to the wide variety of assailants we've encountered over the years, and any rapid overview of this kind always provides a very stereotypical perspective. However, this description may be helpful in identifying the main types of attackers. There are traits that all attackers have in common. They make an effort to hide their name, true location, and physical characteristics because they don't want to be discovered.

If they manage to obtain access to our system, they will undoubtedly try to keep it, if at all feasible, by adding other access points. They believe we won't notice these access points even if you discover the attackers themselves. The majority of them interact with others who share their interests and will typically spread the knowledge they get through assaulting our system. It's possible that a second round of assailants won't be so kind.

6.2 Solutions

I have focused mostly protocol-compliant exclusive traffic and the firewall rule-set in my examine report. Because the packet would not should undergo a lengthy listing of rules like an anticipated firewall, my recommended method works in multilayer security as well as high pace testing. This approach yields matching reductions of nineteen% while the full rejected site visitors is as little as 25% of the overall traffic and 50% when it accounts for up to 75% of all traffic. Only 4% to 10% of the firewall policy is made up of early rejection criteria that have been covered.

I believe my projected model will reduce the rule checking time and processing above because the checking is shorter than ever. But we can also find some drawback, if a

different type of protocol randomly hits the firewall it may cost some time and processing above as I have not tried it in real world.

6.3 Conclusion

There are many various types of firewalls accessible nowadays, as we can see. Different firewalls operate at various network tiers to protect the internal network from internet threads. The performance of firewalls may, however, constantly be improved. So that the firewall system may be improved, I worked on firewall policy and a few other algorithms. Two algorithms were combined in my report paper. These algorithms operate discretely, and their performance is generally good.

Early Traffic Rejection is actually checking the data packets of incoming traffic, if the data packets are unauthentic then it discards the packets in the early stage as a result the threads cannot get the access into the internal network in the other side, the Rule of Re Ordering algorithm mainly changing the rule position of firewall rule-set giving the actively calculated weight of rules. I have also proposed a model of firewall which is considered by the packet protocol.

6.4 Future Related works on Multi Firewall

There are several notes of research arising in this work that should be followed carefully. This project has not been implemented physically, so that's why I am not fully sure about real-world impact. I hope there is a good chance to find a lot of new innovations or problems solving solutions during implementation.

In my project I have considered traffic data as per protocol but it can be characterized in different ways to check the result deviation. Selected attacks on probabilistic filtering strategies are very important in future work. In few algorithms there are few rules are reordered dynamically for actively calculated statistics so that it works better for the same type of data packets but in for random data packets it might work slight less.

That's why I believe here is also an opportunity to optimize it more professionally.

6.5 References

- [1] H. Hamed and E. Al-Shaer. Statistical optimization techniques for Firewall packet Filtering. Technical Report TR-05-012, DePaul University, 2005.
- [2] Website-<https://www.techopedia.com/definition/33097/host-based-firewall>
- [3] H. Hamed & E. Al-Shaer Adaptive statistical optimization techniques for Firewall packet Filtering. Technical Report TR-05-012, DePaul University, 2005.
- [4] "Adaptive Statistical Optimization Techniques for Firewall Packet Filtering," by H. Hamed, A. El-Ataawy , and E. Al-Shaer. In Proceeding of IEEE INFOCOM, pp. 1–12, 2006.
- [5] Website-<https://www.cloudflare.com/learning/network-layer/what-is-arouter/>>
- [6] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, and W. Willinger. A methodology for studying persistency aspects of internet flows in
- [7] Books - SIGCOMM Computer Communication Review, 35(2), 2005.
- [8] Zouheir Trabelsi A survey on firewall's early packet rejection techniques.2011
- [9] E. Al-Shaer & H. Hamed for “Dynamic rule-ordering optimization for high-speed firewall filtering” Proceedings of the 2006.
- [10] Performance Evaluation and Comparative Analysis of Network Firewalls, C. Sheth and R. Thakker, Proc. IEEE International Conference on Devices and Communications (ICDeCom), pp. - 1–5, February 2011.
- [11] Books - ACM Symposium on Information, computer and communications security.
- [12] C. Sheth, R. Thakker, and (2011, February). Network firewalls are evaluated for performance and compared. ICDeCom, the 2011 International Conference on Devices and Communications (pp. 1-5).
- [13] https://www.researchgate.net/publication/220526786_Analysis_of_firewall_policy_rules_using_traffic_mining_techniques
- [14] Books - In The 5th International IPCO Conference, 1996.
- [15] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, & W. Willinger. A methodology for studying persistency aspects of internet flows. SIGCOMM Computer Communication Review, 35(2), 2005.
- [16] file:///C:/Users/HP/Desktop/Fire%20Wall%20Defence/FINAL%20USE/Guide%20Books/AnOverviewofFirewallTypesTechnologiesandFunctionalities.pdf

- [17] Website- http://www.checkpoint.com/products/downloads/vpe_datasheet.pdf
- [18] http://www.cisco.com/warp/public/cc/pd/sqsw/sqppmn/prodlit/spmgr_ds.pdf
- [19] Learn about Firewall, available at << shorturl.at/KOZ47 >>
- [20] T. Woo. "A Modular Approach to Packet Classification: Algorithms and Results." In Proceedings of IEEE INFOCOM'00.
- [21] Website- http://www.checkpoint.com/products/downloads/vpe_datasheet.pdf
- [22] W. Cheswick and S Belovin, Firewalls and Internet Security. Addison-Wesley, 1995.
- [23] H. Hamed and E. Al-Shaer. Adaptive statistical optimization techniques for Firewall packet Filtering. Technical Report TR-05-012, DePaul University, 2005.
- [24] E. Al-Shaer and H. Hamed. Modeling and management of firewall policies. IEEE Transactions on Network and Service Management, 1(1):2–10, April 2004.
- [25] D. Taylor and J. Turner. Scalable packet classification using distributed cross producing of field labels. In IEEE INFOCOM, 2005.
- [26] H. Hamed, A. El-Atawy, and E. Al-Shaer. Adaptive statistical optimization techniques for firewall packet filtering. In IEEE INFOCOM'06, April 2006.
- [27] Website- http://www.cisco.com/warp/public/cc/pd/sqsw/sqppmn/prodlit/spmgr_ds.pdf
- [28] N. Neji, A. Bouhououla. "Dynamic Scheme for Packet Classification Using Splay trees". Information Assurance and Security, pp. 1-9, 2009.
- [29] E. Al-Shear, A. El-Atawy, T. Tran. "Adaptive Early Packet filtering for Defending firewalls against DoS Attack". In Proceeding of IEEE INFOCOM, pp. 1-9, 2009.
- [30] H. Hamed, A. El-Atawy, E. Al-Shaer. "Adaptive Statistical Optimization Techniques for Firewall Packet Filtering". In Proceeding of IEEE INFOCOM, pp. 1-12, 2006.
- [31] E. Cohen and C. Lund. Packet classification in large ISPs: Design and evaluation of decision tree classifiers. ACM SIGMETRICS Performance Evaluation Review, 33(1):73–84, 2005
- [32] B. Hari, S. Suri and G. Parulkar. "Detecting and Resolving Packet Filter Conflicts." In Proceedings of IEEE INFOCOM'00, March 2000.
- [33] D. Chapman and E. Zwicky. Building Internet Firewalls, Second Edition. Orielly & Associates Inc., 2000.

6.6 Appendices of Research

Most firewall security problems don't show up until the network is under a lot of stress. Large volumes of traffic make it easier for attacks to conceal themselves, possibly causing problems just when network outages are most destructive. As their workload increases, firewalls often have among of. In the report, we made an effort to assess how well the most popular firewalls currently in use on the market performed during a DDoS assault. To the best of the author's knowledge, firewall performance is not given the proper weight in the majority of currently conducted and publicized study work on DDoS and is instead focused on other factors. We have made an effort to compare the performance of various firewalls based on actual installation.

In order to ensure that the increased security does not result in performance declining beyond what is acceptable for the company, more research and test findings will be helpful in identifying pre-deployment capacity planning and testing network performance. The variety of tests run will help determine how well the firewall performs and behaves under diverse DDoS attacks. Because it was challenging to generate DDoS attacks, the authors examined a variety of open-source tools for creating traffic and came to the conclusion that Curl Loader was the best choice for the setup.

Through a variety of studies on firewall-based computer network security, I have tried to identify early rejection system solutions through a number of researches on firewall-based computer network security in this research. I believe I'll be able to apply one of these methods in the future.

Final Test

ORIGINALITY REPORT

25%	19%	16%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mnlab.cti.depaul.edu Internet Source	6%
2	Submitted to Daffodil International University Student Paper	4%
3	www.dtic.co.cu Internet Source	2%
4	link.springer.com Internet Source	1%
5	Automated Firewall Analytics, 2014. Publication	1%
6	www.arc.cs.depaul.edu Internet Source	1%
7	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
8	H. Hamed, A. El-Atawy, E. Al-Shaer. "Adaptive Statistical Optimization Techniques for Firewall Packet Filtering", Proceedings IEEE INFOCOM 2006. 25TH IEEE International	1%

Conference on Computer Communications,
2006

Publication

9	Zeidan, Safaa, and Zouheir Trabelsi. "A survey on firewall's early packet rejection techniques", 2011 International Conference on Innovations in Information Technology, 2011.	1%
Publication		
10	Ehab Al-Shaer. "Dynamic rule-ordering optimization for high-speed firewall filtering", Proceedings of the 2006 ACM Symposium on Information computer and communications security - ASIACCS 06 ASIACCS 06, 2006	1%
Publication		
11	www.mnlab.cs.depaul.edu	1%
Internet Source		
12	Chirag Sheth, Rajesh Thakker. "Performance Evaluation and Comparison of Network Firewalls under DDoS Attack", International Journal of Computer Network and Information Security, 2013	1%
Publication		
13	Nguyen Manh Hung, Vu Duy Nhat. "B-tree based two-dimensional early packet rejection technique against DoS traffic targeting firewall default security rule", the 2014 Seventh IEEE Symposium on Computational	<1%

Intelligence for Security and Defense Applications (CISDA), 2014

Publication

14	mafiadoc.com Internet Source	<1 %
15	www.titania.com Internet Source	<1 %
16	www.coursehero.com Internet Source	<1 %
17	www.cs.unc.edu Internet Source	<1 %
18	Submitted to National Tertiary Education Consortium Student Paper	<1 %
19	Submitted to North West University Student Paper	<1 %
20	Edith Cohen. "Packet classification in large ISPs", ACM SIGMETRICS Performance Evaluation Review, 6/6/2005 Publication	<1 %
21	Submitted to Asia Pacific Institute of Information Technology Student Paper	<1 %
22	Ehab Al-Shaer, Saeed Al-Haj. "FlowChecker", Proceedings of the 3rd ACM workshop on	<1 %

Assurable and usable security configuration, 2010

Publication

23	Submitted to American College of the Middle East Student Paper	<1 %
24	Zhitang Li, Xue Cui, Lin Chen. "Analysis And Classification of IPSec Security Policy Conflicts", 2006 Japan-China Joint Workshop on Frontier of Computer Science and Technology, 2006 Publication	<1 %
25	Submitted to University of Northampton Student Paper	<1 %
26	Submitted to University of Witwatersrand Student Paper	<1 %
27	Submitted to Colorado Technical University Student Paper	<1 %
28	Bin Zhang, Ehab Al-Shaer, Radha Jagadeesan, James Riely, Corin Pitcher. "Specifications of a high-level conflict-free firewall policy language for multi-domain networks", Proceedings of the 12th ACM symposium on Access control models and technologies - SACMAT '07, 2007 Publication	<1 %
29	Hazem Hamed, Ehab Al-Shaer. "Dynamic rule-ordering optimization for high-speed firewall	<1 %

filtering", Proceedings of the 2006 ACM Symposium on Information, computer and communications security, 2006

Publication

30	ir.hust.edu.tw Internet Source	<1 %
31	www.researchgate.net Internet Source	<1 %
32	Submitted to Cardiff University Student Paper	<1 %
33	H. Hamed. "On Dynamic Optimization of Packet Matching in High-Speed Firewalls", IEEE Journal on Selected Areas in Communications, 10/2006 Publication	<1 %
34	H. Hamed, A. El-Atawy, E. Al-Shaer. "On Dynamic Optimization of Packet Matching in High-Speed Firewalls", IEEE Journal on Selected Areas in Communications, 2006 Publication	<1 %
35	core.ac.uk Internet Source	<1 %
36	www.cisco.com Internet Source	<1 %
37	Sreelaja, N.K.. "Ant Colony Optimization based approach for efficient packet filtering in	<1 %

firewall", Applied Soft Computing Journal,
201009

Publication

38	dokumen.pub Internet Source	<1 %
39	L. Kencel, C. Schwarzer. "Traffic-Adaptive Packet Filtering of Denial of Service Attacks", 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM'06), 2006 Publication	<1 %
40	Trabelsi, Zouheir, Safaa zeidan, Mohammad M. Masud, and Kilani Ghoudi. "Statistical dynamic splay tree filters towards multilevel firewall packet filtering enhancement", Computers & Security, 2015. Publication	<1 %
41	csweb.cs.wfu.edu Internet Source	<1 %
42	krishikosh.egranth.ac.in Internet Source	<1 %
43	upcommons.upc.edu Internet Source	<1 %
44	Pozo, S.. "Model-Based Development of firewall rule sets: Diagnosing model inconsistencies", Information and Software Technology, 200905	<1 %

Publication

45 Sagar Ajay Rahalkar. "Chapter 13 IDSes, Firewalls, and Honeypots", Springer Science and Business Media LLC, 2016 <1 %
Publication

46 W. K. G. Seah. "Game-Theoretic Model for Collaborative Protocols in Selfish, Tariff-Free, Multihop Wireless Networks", 2008 IEEE INFOCOM - The 27th Conference on Computer Communications, 04/2008 <1 %
Publication

Exclude quotes Off

Exclude matches Off

Exclude bibliography On