# Review of Firewall Optimization Techniques

Shakirah Saidin
Faculty of Computer Systems & Software Engineering (FSKKP)
Universiti Malaysia Pahang
shakirah.saidin90@gmail.com

Mohamad Fadli Zolkipli
Faculty of Computer Systems & Software Engineering (FSKKP)
Universiti Malaysia Pahang
fadli@ump.edu.my

*Abstract*—Firewall is a vital component in network security. Changes in requirements due to the growth of the Internet and increasing types of attack lead to the larger size of firewall policies and consequently affect the firewall performance. Hence, the network security policy will also be jeopardized. Minimizing firewall rules by removing anomalies in the policy proven to be one of the solutions. Firewall performance can also be optimized using data mining technique and optimization based on traffic awareness. This paper review firewall optimization techniques such as data mining, anomaly detection, and traffic awareness, that have been done throughout time. Suggestion to combine the removing anomalies technique and data mining technique to enhance firewall performance further is also mention in this paper. As a result, this paper will be useful for researchers who are interested in learning more about firewall optimization techniques.

*Keywords—Firewall optimization; Data mining; Anomaly detection; Packet filtering*

## 1. INTRODUCTION

Firewall may be considered as a security guard that inspect every incoming and outgoing packet which placed at the entrance of a private network. The traffics need a firewall permission to access available open services such as hypertext transfer protocols and domain name servers [1]. It also acts as a defense mechanism to protect the local network from unauthorized access. The need for providing secure and safe information security systems through the use of firewalls increase as the rapid growth and widespread use of electronic data processing and electronic business conducted through the massive use of the wired and wireless communication networks, Internet, cloud computing alongside numerous occurrences of international terrorism [2].

Currently, the most common type of firewall used in networks is packet filtering firewall. Firewall policy is a set of filtering rules that were made based on security policy requirements. It defines an action that should be taken on every matching packet [3]. The action could either be allow or deny. For every firewall, there would be a default action. In this paper, we would assume that the default action is deny.

Throughout time, the requirements might change or increase. This is resulting in rules increment and larger size of firewall policies. Consequently, the time required for packet matching will increase. Every packet that entered the private network will be sequentially compared with every filtering rules in the policy until a match is found [4, 5]. This process is time consuming when the size of rules is large or when there are anomalies present in the rules. The large size of firewall policies might also contain redundant rules that are called anomalies.

To regard all these problems, many researches has been done in the field of firewall optimization. However, there is still no ultimate solution to resolve all the problems related to firewall optimization. Each technique that has been proposed usually resolve one or two problem and need to compromise other problem. For example, FIREMAN by Yuan *et al*. [6] might resolve pair-wise limitation in Hameed and Shaer [3] but failed to handle dynamic firewalls as it adapts a static analysis approach.

The rest of this paper is organized as follows: Section II presents the literature review of previous works related to firewall optimization techniques; Section III describes the implications of this review paper; and Section IV as the conclusion.

## 2. OPTIMIZATION TECHNIQUES

The most common type of firewall used in the current network is packet filtering firewall which is also known as stateless firewall or network layer firewall. According to Krit and Haimoud [7], there are five types of firewall; i. Packet-filtering firewall, ii. Circuit-level gateways, iii. Stateful inspection firewall, iv. Application layer firewall (Proxies), and v. Multilayer inspection firewall. Packet filtering firewall works by matching incoming packets to the firewall's rule set and take the predefined action; allow or deny. All open connections were recorded in a table in stateful inspection firewall. A packet that has the similar characteristic with the previous connection will be allowed without further analysis while a new packet will be compared with firewall's rule set. A direct connection between sender and recipient is restricted in application layer firewall. The session needs to be conducted through the proxy firewall.

Table 1 shows the example of firewall rule sets that found in a standard packet filtering firewall. The rules usually consist of seven fields including an action.

<order><protocol><src_ip><src_port><dst_ip><dst_port> <action>

The first field is the order number of the rule in the firewall policy. Rule order is essential in policy as it might also affect the performance of the firewall. For example, when a most hit rule is placed in the middle or at the bottom of the order, the time taken for packet filtering process will be lengthened as a packet will sequentially match with the rule. The protocol field specifies the type of protocol in the packet whether it is TCP or UDP. The source IP is the address of the packet origin, and the source port is the port number of the packet. Some of the well-known port numbers are 80 for HTTP connection, 53 for DNS, 25 for SMTP and 21 for FTP. Destination IP specifies the target address for the packet and destination port as the target port for the packet. The last field is the action field where an administrator determines the action that needs to be taken on the incoming and outgoing packet based on network requirements. If the action is allow, the packet will go through the network, while if the action is deny, the packet connection will be dropped.

Table 1: Example of firewall rule set.

| order | protocol | src_ip | src_port | dst_ip | dst_port | action |
|---|---|---|---|---|---|---|
| R1 | TCP | 172.17.0.0 | 80 | any | 80 | allow |
| R2 | TCP | any | 80 | any | 80 | allow |
| R3 | UDP | 172.17.0.2 | any | any | 53 | deny |
| R4 | TCP | 172.17.0.2 | 21 | any | 21 | allow |
| R5 | UDP | any | any | 172.17.3.2 | 53 | deny |
| R6 | TCP | any | any | 172.17.0.2 | any | deny |
| R7 | TCP | 172.17.3.4 | any | 172.17.0.2 | any | allow |

Previous works on firewall optimization has been done through various techniques such as anomaly detection techniques [3, 8, 9], optimization of firewall rules [4], optimization based on traffic awareness [10, 11], data mining [12-14], dynamic rule ordering technique [15, 16], and ACL partitioning [17, 18]. Table 2 shows the list of previous works and research that had been done in firewall optimization area. From the table, we could see the trend of using data mining technique, anomaly detection technique, and traffic awareness technique. Furthermore, other techniques such as optimizing firewalls with changing designs, integer programming language and cross-domain firewall optimization had been explained in Kadam and Bhusari [19]. Therefore, we further explained about those works; data mining, anomaly detection, and traffic awareness, in the next paragraph.

Table 2: List of previous works applying the various techniques for firewall optimization.

| Techniques | Title | Results |
|---|---|---|
| Data Mining | Firewall Performance Optimization Using Data Mining Techniques [12] | 40% less processing time compared to the standard firewall |
| | Analysis of Firewall Policy Rules Using Data Mining Techniques [13] | Automated tool to identify frequent traffic patterns and filtering rules, using Log Mining Frequency and Filtering-Rule Generalization, to discover and generate a valid and anomaly-free firewall policy rules |
| Anomaly Detection | Firewall Policy Advisor for Anomaly Discovery and Rule Editing [3] | A firewall policy advisor capable of rule insertion, removal, and modification |
| | Towards a New Design of Firewall: Anomaly Elimination and Fast Verifying of Firewall Rule [9] | Proposed firewall managed to verify 16384 rules within 28 milliseconds |
| Traffic Awareness | Optimized Firewall with Traffic Awareness [11] | 700 packets compared in 8 milliseconds |
| | Dynamic rule-ordering optimization for high-speed firewall filtering [15] | 60% reduction in packet matching processing time |

### *Data Mining Technique*

Data mining can be considered as knowledge discovery. Grossman defines data mining as being “concerned with uncovering patterns, associations, changes, anomalies, and statistically significant structures and events in data” [20]. To improve the firewall processing time and performance, an approach using Data Mining Classification techniques was proposed by Mustafa *et al.* [12]. The classification technique adopts decision tree classifier due to its fast classification time and overall satisfactory performance. Their enhanced firewall namely Data mining Boosted Firewall (DBF) is a combination of a standard firewall with a data mining module. In their approach, the authors trained a classifier to predict a matching rule instead of comparing it with the rules. The speed of the packet filtering has increased at least 40% more compared to a standard firewall.

In Mustafa *et al*. [12], their approach however only predicts the first rule, not the second and third filtering rule. In the case where the classifier predicted a wrong rule, the packet will go back to match the rule one by one like in the traditional firewall. This means that the DBF performance relies on the correctness of their classifier prediction. The experiment done was only using 13 filtering rules. This is not really portraying the real firewall where the rules could accumulate up to thousands of rules. Since the rules were generated, there is no consideration for anomalies in the filtering rule if this approach is to be implemented in the real network.

Another notable work applying data mining technique to make the firewall more efficient is proposed by Golnabi *et al.* [13]. They presented MLF which is Mining firewall Log using Frequency to create a practical firewall policy rules by mining its network traffic log based on its frequency. In the approach, they also decrease the number of policy generalization by using Filtering-Rule Generalization (FRG). To create a new set of efficient firewall policy rules, they proposed a technique to identify decaying rule and dominant rule. Consequently, by using Log Mining Frequency and Filtering-Rule Generalization, an automated tool capable of finding familiar traffic patterns and filtering rules has been provided by the authors to generate an anomaly-free firewall policy rules that effective. The authors concluded that data mining proved to be a practical, useful, and critical approach in firewall policy rules analysis and optimization in real time.

Approach [12] focused more on the filtering processes while approach [13] focused on the filtering rules. In the future, both approaches could be combined to get a better firewall performance. This is because by using approach [13], rule anomalies could be resolved and a new firewall policy which is anomaly-free could be generated. Then, by using approach [12], the filtering process could speed up. The combination of these two approaches resolved two main issues in firewall optimization which is to reduce the size of filtering rules and to decrease the time taken for the filtering process. However, a solution to enhance approach [12], which is to be able to predict the second and third filtering rules is still to be found.

### *Anomaly Detection Technique*

Al-Shaer and Hamed describe an approach which provides us Firewall Policy Advisor [3]. This advisor provides tools that purify and protect firewalls from anomalies of rules. They formally define the anomalies in firewall policies in both centralized and distributed firewalls. Anomaly classification is performed as shown in Fig. 1.

The anomaly in firewall rule set can be classified into five types.

a) Shadowing anomaly
A rule is shadowed when the preceding rule already matches all the packets that the said rule supposed to match. The shadowed rule never takes effect in the firewall policy and became a critical error. This error might be corrected by reordering or remove the rule.

b) Correlation anomaly
A rule is correlated with other rules if the rule intersects with others but having a different action. This anomaly could be rectified by prompting the network administrator to choose the proper order that conforms with security policy requirements.

c) Generalization anomaly
A rule is a generalization of another rule if this general rule can match all the packets that match a specific rule that precedes it. Therefore, the particular rule will make an exception to the general rule. For confirmation, an action should be taken by the network administrator.

d) Redundancy anomaly

A redundant rule is two rules that have the same action on the same packet. If one of the rules is removed, the security policy will not be affected. The redundant rule will contribute to the larger size of firewall and decrease the firewall performance. Therefore, it should be removed.

e) Irrelevance anomaly
A rule is considered irrelevance when it cannot match any packet that goes through the firewall. Therefore, the rule has no effect on the filtering outcome. It adds to unnecessary overhead and should be removed.

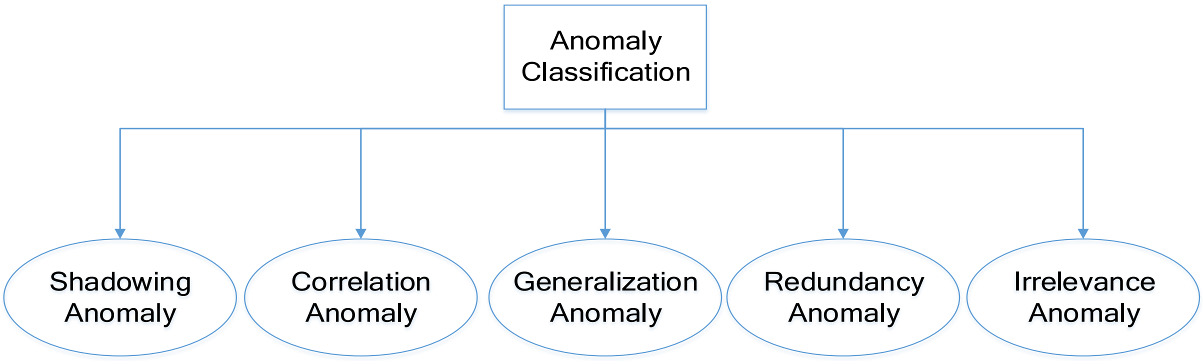

Figure 1: Anomaly classification.

They also proved that these types of anomalies are the only conflicts that could exist in the policies. However, Golnabi *et al.* [13] managed to find other two types of anomalies; blocking existing service anomaly, and allowing traffic to non-existing services anomaly, by using their data mining approach. Nevertheless, the work made by [3] served an excellent purpose for other succeeding works. Researchers are now aware that solving anomalies became a high priority before applying other enhancing firewall performance techniques.

The algorithm proposed by them detects rules anomalies in inter and intra firewall in the network. The Firewall Policy Advisor performed in two management tools; Policy Anomaly Detector and Policy Editor. The detector works to identify anomalies and prompt administrator for solutions while the editing tool is for rule insertion, removal, and modification. Although the authors went through a great length detailing about the function of the tools, apparently the experiment was done using only eleven rules. There is no indication that the tools are implemented in the real network. This raised a question of the effectiveness of the tools handling real network data.

Another work adopting the anomalies detection technique is from Khummanee, Khumseela, and Puangpronpitag [9]. They proposed a new firewall design with two parts; Single Domain Decision firewall (SDD) and Binary Tree Firewall (BTF). The SDD primary function is to solve anomalies. It is also able to address other restrictions such as re-order rules freely without changing its meaning. It can also reduce the rules redundancy by merging several rules into a single rule. The BTF rules work as a data structure of SDD firewall to increase the speed of checking and verifying the firewall rules. It used the properties of the binary tree for sorting and searching. As a result, the proposed firewall managed to verify 16384 rules within 28 milliseconds.

Approach by [9] focused on removing anomalies in the filtering rules. Although they managed to execute the idea, however, they encountered a problem where the size of the filtering rules is increased more than the standard firewall. They focused only on the verifying time while neglecting the issue of time for building structure and space complexity. For example, for 16384 rules, the build time is 253 milliseconds in standard firewall while 27964 milliseconds in SDD firewall and the space complexity is 204800 milliseconds and 397300 milliseconds respectively. There was also no mentioning of the number of packets that were used when verifying the SDD firewall. This is important since we need to know if the proposed firewall can handle real-time network and network attacks such as denial of service (DoS) attack. The space complexity problem has been addressed by [11] and [15] in their approach and explained in traffic awareness section.

*Traffic Awareness*

One of the drawbacks with current solutions for firewall optimization issues is the least awareness of packet traffic. Cherian and Chatterjee proposed an approach to creating a dynamic firewall access rule set by considering dynamic packet traffic scenarios [11]. The approach involves converting rules into binary format using BDD data structure. The optimized firewall proved to be more efficient and better in performance by reducing computation overhead for packet and rule comparison. Fig. 2 shows the architecture of proposed optimized firewall by the authors.

The architecture consists of three modules; i. Pattern matching engine, ii. Rule optimization engine, and iii. BDD engine. The first engine function to filter the incoming packets based on the pre-optimized initial rule set. Packets will be accepted and denied accordingly. The second engine maintained a counter that recorded number of times that a rule is compared. The counter helps to put the most compared rule on top of the optimized rule set when reordering process takes place. The third engine takes the optimized rule as an input and convert it into binary and produced a .blif file. The Optimized Rule Set is tested in four test scenarios; ICMP, TCP, UDP, and HTTP. In average, the firewall managed to filter 700 packets in 8 milliseconds.

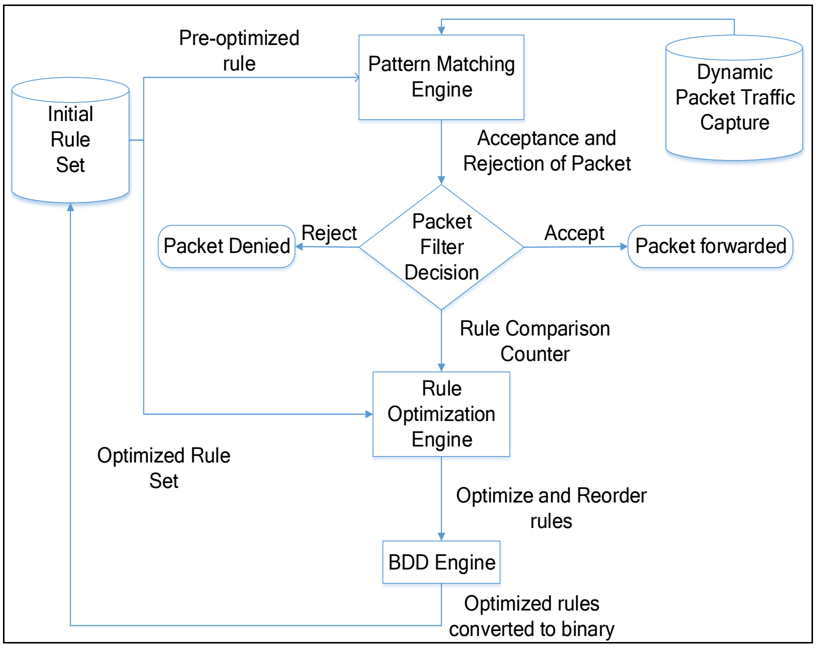

Figure 2: Architecture of Optimized Firewall [11].

Since approach by Cherian [11] is the latest in firewall optimization field, they managed to address many current issues such as least awareness of packet traffic, anomalies in firewall policy, the large size of firewall rules, and space complexity. They created a dynamic optimized rule set based on the pattern of packet traffic that managed to accept or deny packets at the earliest with minimal computation overhead. The authors also performed rule reduction and rule reorder in their approach which decreases the large size of the firewall. The conversion of the optimized firewall into binary solved the space complexity issues.

Packet filtering plays a vital role in firewall technology. Better firewall performance will result in the better speed of packet filtration. Current packet filtering techniques that did not consider traffic behavior in optimizing their search data structures will lead to impractically high space complexity and undermines the performance gain offered by these techniques [15]. Therefore, Hamed and Shaer [3] proposed an approach which utilizes traffic characteristics to enhance firewall's performance in filtering the firewall policies. They also developed an adaptive mechanism to dynamically update the rule list so that the matching performance is always optimal. The authors explained in detail the techniques, algorithms and evaluation study of their proposed approach. The technique managed to achieve 60% reduction in packet matching processing time.

## 3. Implication Of This Survey

This paper reviews previous works that had been done in firewall optimization and contribute literature in specific techniques such as data mining, anomaly detection, and traffic awareness. The data mining technique managed to increase the speed of packet matching process by 40%. However, this technique only predicted the first rule. If the predicted rule is wrong, then the packet needs to undergo another circle of packet matching process. Evidently, if the primary problem which is the anomalies in firewall rule set has not been resolved, the complexity of the process would be higher. This proved that the first step that needs to be taken in enhancing firewall performance is to remove the anomalies in the firewall policy. This step has been explained extensively by Hamed and Shaer [3]. We could see that many approaches [9, 11, 13] has applied this technique in their work as the first step. Next, the firewall's rule could be restructured where the rule with many hits should be placed on top in the policy.

Only then, the data mining technique, traffic awareness technique, or other proposed techniques could be performed and improved the performance of the firewall.

## 4. Conclusion

Conclusively, many algorithms and tools had been developed such as Firewall Policy Advisor by Hamed and Shaer [3], Firewall Compressor by Liu *et al.* [21], FIREMAN by Yuan *et al.* [6] and FIRO by Katic and Pale [4]; for the purpose of optimizing firewall. Detecting and removing anomalies in firewall's rules is one of the most critical steps to enhance firewall performance. Other than that, the size of the firewall policy should be minimized to reduce packet matching time and boost firewall efficiency. Future work might involve reviewing other techniques such as using integer programming techniques and changing the design of firewall. Hopefully, this review paper will help other researcher who are interested in pursuing in firewall optimization finding their path and research directions.

## Acknowledgment

Acknowledgments to my supervisor Dr. Mohamad Fadli Zolkipli for his input and feedbacks towards publishing this paper. I would also like to thank the members of FSK 9 Research Lab for their support and contributions. This paper has been funded under a grant : UIC170802 Next Generation Security Operation Center (NGSOC) Framework.

## References

[1] S. Anwar *et al.*, "From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions," *Algorithms*, vol. 10, no. 2, 2017.

[2] M.H. Ali and M.F. Zolkipli, "Review on hybrid extreme learning machine and genetic algorithm to work as intrusion detection system in cloud computing," *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 1, pp. 460–464, 2016.

[3] H.H. Hamed and E.S.A-. Shaer, "Firewall Policy Advisor For Anomaly Discovery And Rule Editing," *IEEE/IFIP Adv. Inf. Commun. Technol.*, vol. 118, pp. 17–30, 2003.

[4] T. Katic and P. Pale, "Optimization of Firewall Rules," *Proc. Int. Conf. Inf. Technol. Interfaces, ITI*, pp. 685–690, 2007.

[5] S. Syurahbil, N. Ahmad, M.F. Zolkipli, and A.N. Abdalla, "Intrusion Preventing System using Intrusion Detection System Decision Tree Data Mining," *Am. J. Eng. Appl. Sci.*, vol. 2, no. 4, pp. 721–725, 2009.

[6] L. Yuan, H. Chen, J. Mai, C.N. Chuah, Z. Su, and P. Mohapatra, "FIREMAN: A toolkit for firewall modeling and analysis," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2006, pp. 199–213, 2006.

[7] S. Krit and E. Haimoud, "Overview of firewalls: Types and policies: Managing windows embedded firewall programmatically," *2017 Int. Conf. Eng. MIS*, pp. 1–7, 2017.

[8] H. Hu, G.J. Ahn, and K. Kulkarni, "Detecting and resolving firewall policy anomalies," *IEEE Trans. Dependable Secur. Comput.*, vol. 9, no. 3, pp. 318–331, 2012.

[9] S. Khummanee, A. Khumseela, and S. Puangpronpitag, "Towards a new design of firewall: Anomaly elimination and fast verifying of firewall rules," in: *Proceedings of the 2013 10th International Joint Conference on Computer Science and Software Engineering, JCSSE 2013*, 2013, pp. 93–98.

[10] S. Acharya, J. Wang, Z. Ge, T.F. Znati, and A. Greenberg, "Traffic-aware firewall optimization strategies," in *IEEE International Conference on Communications*, 2006, vol. 5, pp. 2225–2230.

[11] M. Cherian, "Optimized Firewall with Traffic Awareness," *Int. J. Comput. Networks Appl.*, vol. 3, no. 2, pp. 32–37, 2016.

[12] U. Mustafa, T. Wood, M.M. Masud, Z. Trabelsi, and Z.A. Harthi, "Firewall Performance Optimization Using Data Mining Techniques," *World*, 2013.

[13] K. Golnabi, R.K. Min, L. Khan, and E.A-. Shaer, "Analysis of Firewall Policy Rules Using Data Mining Techniques," in: *2006 IEEE/IFIP Netw. Oper. Manag. Symp. NOMS 2006*, pp. 305–315, 2006.

[14] M.M. Masud, U. Mustafa, and Z. Trabelsi, "A data driven firewall for faster packet filtering," in: *4th Int. Conf. Commun. Networking, ComNet 2014 - Proc.*, 2014.

[15] H. Hamed and E.A-. Shaer, "Dynamic rule-ordering optimization for high-speed firewall filtering," in: *Proc. 2006 ACM Symp. Information, Comput. Commun. Secur. - ASIACCS '06*, p. 332, 2006.

[16] L. Zhang, Z. Trabelsi, and S. Zeidan, "Dynamic rule and rule-field optimisation for improving firewall performance and security," *IET Inf. Secur.*, vol. 8, no. 4, pp. 250–257, 2014.

[17] A. Dange and G. Sonune, "Optimization of Firewall," *J. Eng. Comput. Appl. Sci.*, vol. 2, no. 7, pp. 38–42, 2013.

[18] G. Misherghi, L. Yuan, Z. Su, C.N. Chuah, and H. Chen, "A general framework for benchmarking firewall optimization techniques," *IEEE Trans. Netw. Serv. Manag.*, vol. 5, no. 4, pp. 227–238, 2008.

[19] P.R. Kadam and V.K. Bhusari, "Review on Redundancy Removal of Rules for Optimizing Firewall," *Int. J. Res. Eng. Technol.*, vol. 3, no. 9, pp. 397–401, 2014.

[20] R.L. Grossman, Data Mining: Challenges and Opportunities for Data Mining During the Next Decade. 1997.

[21] A.X. Liu, E. Torng, and C.R. Meiners, “Firewall compressor: An algorithm for minimizing rewall policies,” INFOCOM, Phoenix, Arizona, April, vol. 1, pp. 691–699, 2008.