International Journal of Computer and Internet Security.
ISSN 0974-2247 Volume 8, Number 1 (2016), pp. 9-14

http://www.irphouse.com

Firewall Optimization with Traffic Awareness Using Binary Decision Diagram

Mimi Mariam Cherian and Madhumita Chatterjee

Department of Computer Engineering, PIIT, New Panvel, Mumbai.
Madhumita Chatterjee is with the Department of Computer Engineering, PIIT, New Panvel, Mumbai.

Abstract

Firewalls are the most popular network-based security devices and have been widely deployed since the early days of computer networks. They are designed to permit or deny network traffic based on a firewall policy that specifies what types of packets should be allowed from/into the protected network. The growth of network complexity, it is very common to find firewall policies with thousands of rules. Firewall policy contains a list of rules that are usually checked in a sequential order. This implies that the higher the order of the matching rule, the more costly the firewall filtering overhead will be. Thus, to reduce the filtering overhead, it is crucial to have the appropriate rule ordering in the firewall policy
Firewall checks every incoming and outgoing packets by analyzing the data packets and then by using many policies defines whether to accept or discard the traffic. It is vital to improve the firewall policies to increase performance of network.

Index Terms—Firewall, Rules, Binary Decision Diagram, Traffic Awareness.

1 INTRODUCTION

Performance of network is highly depen-dent on efficiency of firewall as the packet which passes in or moves out of the network, decision has to be made whether to accept or reject it [4]. The existing solutions and tech-niques suffer with various drawbacks. The ma-jor drawback is less traffic awareness leading to sequential comparison and computational overhead.

1. 1 Literature Survey

Gopal Pault et. al, [1] presents an approach in which packet filtering is done based on a binary decision diagram as it improves the stor-age and look up time for access rule compar-ison. BDD approach also improves the packet comparisons compared to list based packet fil-ter Subrata Acharya et. al, [2] propose an ap-proach to consider traffic based factors for optimizing firewall. They provide a technique to reorder the access rules based on pattern of traffic of packets and then do the packet filter thus reducing comparison computation overhead.

Zouheir Trabelsi et. al, [3] provides a method to reduce time of firewall packet filtering by improving the rearrangement of policy filtering fields to have early packet rejection. The pro-vided method is based on the optimization of filtering fields order based on traffic statistics

P. R. Kadam et. al, [4] provides few techniques used for removal of rule redundancy and lead-ing this work to the searching of malicious user in the system Hazem Hamed et. al, [5] proposes the techniques that adapt to timely changes in the traffic scenarios by undergoing basic calculations for optimizing search data structure.

Anssi Kolehmainen, [6] provides few algo-rithms used for firewall optimization that em-phasize minimizing memory usage and others minimizing execution time. Existing research considers two scenarios. First scenario they consider dynamic traffic pattern to create opti-mized rule set, but binary conversion of these rules are not taken into consideration. In the above scenario comparison between incoming packets and access rule is reduced but binary conversion of rule is time consuming. In second scenario they have considered a static access rule set which has logic of binary conversion that reduces binary conversion overhead, but dynamic traffic pattern is not considered to create optimized rule set. Therefore integrate traffic awareness and binary conversion on ac-cess rule list to have an optimized rule set might be a better solution.

The current research considers scenario of packet traffic leading to dynamic access rule set which leads to overhead on binary conversion computation during comparison. Hence inte-gration of traffic awareness leading to dynamic access rule set along with binary conversion of access rule list leads to better firewall optimiza-tion.

Proposed system tries to improve firewall by accepting or rejecting the packet as early as possible. The approach considers packet traffic scenarios and create a dynamic firewall access rule set which converts them into binary for-mat using BDD data structure. The proposed system relies on the packet traffic awareness and BDD tree data structure for rejecting and accepting packet as early as possible. BDD optimizes the firewall access rule set created by dynamic packets traffic-aware and improves the operational amount of firewalls. Our main contribution: The proposed method will dy-namically detect traffic behavior pattern and adaptively modifies the firewall rules to avoid critical performance degradation due to the traffic pattern and binary conversion.

2 PROPOSED METHODOLOGY

We need to have an optimized firewall access rule set that allow rejection and acceptance of packets as early as possible with less computa-tional overhead. The idea is to create optimized dynamic access rules set based on the traffic pattern of packets, later this optimized rule set undergoes Binary Decision Diagram for binary conversion of optimized rule set to enhance early packet rejection and reduce computation overhead.

The technique of optimization used to opti-mize the initial list of firewall rules. This opti-mization is done through three filtering levels in Figure 1:

Rules Set Optimization Traffic Optimization
BDD Approach

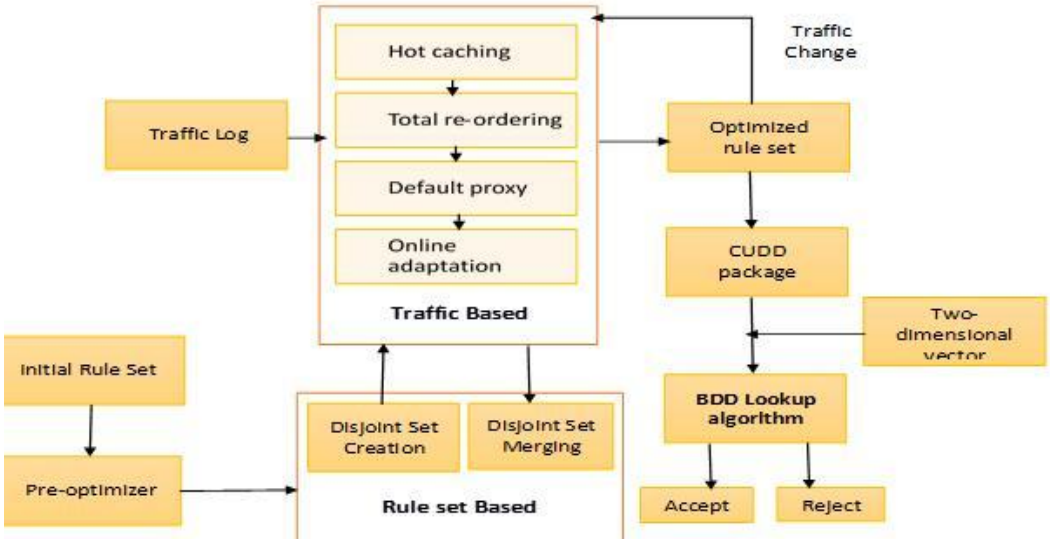

Fig 1: Proposed system

2.1 Module 1: Initial Rule Set

The raw set of rules which are static and not optimized are taken as input to further mod-ules. The rule consists of source address, desti-nation address, service type, protocol number, port number and action. These rules are later optimized based on traffic of packets and un-dergoes BDD conversion. In Fig 2 we have pre-optimized rule set.

Rule	Src Ip	Dst Ip	Src Port	Dst Port	Action
R1	S1, S2, S3	D1, D2, D3	SP1	DP1	DENY
R2	S2, S3, S4	D2, D3, D4	SP1	DP1	ACCEPT
R3	S5	D4	SP1	DP1	ACCEPT

Fig 2. TABLE I: Pre-optimized rule set

2.2 Module 2: Rule Set Optimization

The pre optimized rules are taken into this module. There are two sub modules in this module Disjoint Set Creation (DSC) and Dis-joint Set Merging (DSM) [2]. These modules helps in reducing the comparisons of rules by minimizing the rule interdependency and providing independent rules set [2]. DSC: In current rule set identify and eliminate redun-dancy. Thus creating a new rule set to create the entire rule set disjoint.

DSM: Combine the rules in disjoint rules provided by DSC in to improve the rule set representation, Merging occurs between rules having similar action held, to retain semantic integrity

In below Fig 3 we have the optimised rule set that have eliminated redundancy by DSC and also merging based on similar action done by DSM.

Rule	Src Ip	Dst Ip	Src Port	Dst Port	Action
R1	S1, S2, S3	D1, D2, D3	SP1	DP1	DENY
R4	S2, S3, S5	D4	SP1	DP1	ACCEPT
R23	S4	D2, D3, D4	SP1	DP1	ACCEPT

Fig 3. TABLE II: Final rule set

2. 3 Module 3: Traffic Optimization:

We take the rules from Rule Set Optimization as input to this module. In this module there are four sub modules Hot Caching, Total Re-ordering, Default Proxy, Online Adaptation [1].

Hot Caching: Reorder the rules based on most frequently accessed or most hit rules. It focuses only on small set of rules that can help in reordering the access rule list. Assuming R23 as most accessed rule please refer table in Figure 4

Rule	Src Ip	Dst Ip	Src Port	Dst Port	Action
R23	S4	D2, D3, D4	SP1	DP1	ACCEPT
R1	S1, S2, S3	D1, D2, D3	SP1	DP1	DENY
R4	S2, S3, S5	D4	SP1	DP1	ACCEPT

Fig 4. TABLE III: Hot Cache

Total Reordering: Reorder the whole access rule list based on small set of hot caching rules and also size of rules. The ordering of rules based on below calculation reduces the computational cost The table in figure 5 is assuming the Pr(Ri) value of R23 is highest while R4 and R1 values are low

$$Pr(Ri) = Hitcount(Ri)=Size(Ri) \quad (1)$$

Rule	Src Ip	Dst Ip	Src Port	Dst Port	Action
R23	S4	D2, D3, D4	SP1	DP1	ACCEPT
R4	S2, S3, S5	D4	SP1	DP1	ACCEPT
R1	S1, S2, S3	D1, D2, D3	SP1	DP1	DENY

Fig 5. TABLE IV: Total Reordering

Default Proxy: Relates to the hit rate of drop rule. Each newly created drop rule is prioritized based on its hit rate and its size, similar to total re-ordering.
Online Adaptation: Re-ordering based on profile uses trafc scenarios to build a long-term rule hit prole. This method is used to build the prole that exploits trafc vari-ability. It also helps in detecting long term and short term anomalies and adapt rule set accordingly.

2.4 Module 4: BDD based Approach

The optimized rules we got from rule set op-timization and traffic optimization are given as input to BDD module. In this module it automatically takes the optimised access rule list as input and gives the corresponding. blif file (convert into binary) as output. The algo-rithm developed for the blif file generation is as follows:
Input: Optimized Rule List
Output: containing binary form of the rule list.

1) Take each rule from the rule list.
2) Extract the protocol number, source ad-dress and port, and destination address and port.
3) Convert each part in to its binary format in required number of bits.
4) Store the each part of the converted bi-nary form in to the file

This blif file is given as input to CUDD pack-age to get the optimized form of BDD output files BDD algorithm decides packet rejection and acceptance. Thus we get the optimized access rule set which helps in early acceptance and rejection of packets with lesser comparison computation.

3 STRATEGY:

Each module of this architecture works inde-pendently. The interaction of each module is with the databases or source codes of the web pages. The architecture of this system is planned to be in the form of a web browser plugin, which will be compatible with all major browsers. The implementation is planned to be in ASP. NET and MS Access database

4 CONCLUSION:

All the above discussion clearly indicates that optimization of firewall rules is very

important.
The proposed framework optimizes the fire-wall rules based on dynamic network packet traffic pattern also BDD conversion helps in reducing computation overhead.

REFERENCES

1) Gopal Pault, Amaresh Pothnal, C. R. Mandalt, Bhargab B. Bhattacharya, Design and Implementation of Packet Filter Firewall using Binary Decision Diagram, IEEE Students Technology Symposium, 2011.
2) Subrata Acharya, Jia Wang, Zihui Ge, Taieb F. Znati and Albert Greenberg,, Traffic-Aware Firewall Optimization Strategies, 2010.
3) Zouheir Trabelsi and Safaa Zeidan, Multilevel Early Packet Filtering Technique based on Traffic Statistics and Splay Trees for Firewall Performance Improvement, Communication and Information Systems Security Symposium 2012.
4) P. R. Kadam, V. K. Bhusari, Review on Redundancy removal of rules for Optimizing Firewalls, International Journal of Research in Engineering and Technology, Sep-2014.
5) Hazem Hamed, Adel El-Atawy, and Ehab Al-Shaer, On Dynamic Optimization of Packet Matching in High-Speed Firewall, IEEE Journal, Oct-2006.
6) Anssi Kolehmainen, Optimizing Firewall Performance, 2008.
7) Ghassan Misherghi, Lihua Yuan, General framework for benchmarking firewall techniques, IEEE Transactions, Dec-2008.