Maintaining Privacy Protection though firewall optimization

J.Naveen[1]&P.Praveen Kumar[2]

1. M-Tech CSE ,Vignana Bharathi Institute of Technology, Ghatkesar,Hyderabad . Telangana.

2. Associate Professor, Vignana Bharathi Institute of Technology, Ghatkesar , Hyderabad. Telangana.

[1]naveen.joseph2906@gmail.com

Abstract—

In Present days Firewalls have been broadly deployed on the Internet for protecting private networks. A firewall will checks each every incoming and outgoing packet to decide whether to accept or reject these packet is based on its policy. Optimizing firewall policies is essential for improving overall network performance. Prior work on firewall optimization should be focused on either intra firewall or inter firewall optimization within one managerial domain where the privacy protection of firewall policies is not a distress. In this paper we explores inter firewall optimization across the administrative domains for the first time .The key technical confront is that firewall policies cannot be shared across domains because a firewall policy contains secret information and even potential privacy security holes, that are be exploited by attackers. For this purpose, we propose a method cross domain privacy protection in cooperative firewalls policy optimization protocol. Mainly, for any of two adjacent firewalls belonging to two dissimilar managerial domains, our procedure can identify in each and every firewall the rules that can be detached because of the some other firewall. These optimization process involves in cooperative calculation between the two different firewalls without any third party disclosing its policy to the other party. We implemented our procedure and conducted widespread experiments.Our procedure incurs no extra online container processing overhead, and the offline handing out time is less than a few hundred seconds.

KEYWORDS—

Cross Domains; Intra and Inter Firewalls; VPN; Redundancy Detection; Efficiency

I.INTRODUCTION

1.1. Background and Motivation

Firewalls are important in securing private networks of organizations, institutions, and personal home networks. A firewall is frequently placed at the entry between a private network and the outside network so that it can check each and every incoming or outgoing packet and make a decision whether to accept or reject the packet based on its policy. A firewall policy is typically specified as a sequence of protocols, called Access Control List, and each rule has a separate predicate over multiple packet header fields and a decision making field for the packets that match the predicate value. These rules in a firewall policy in general follow the first match semantics, where the result for a packet is the result of the first rule that the packet matches in the policy. Each substantial interface of a router or a firewall is configured with two ACLs, one for filtering sending packets and the for filtering receiving packets. In this paper ,we use firewalls, firewall policies, and ACL, interchangeably.

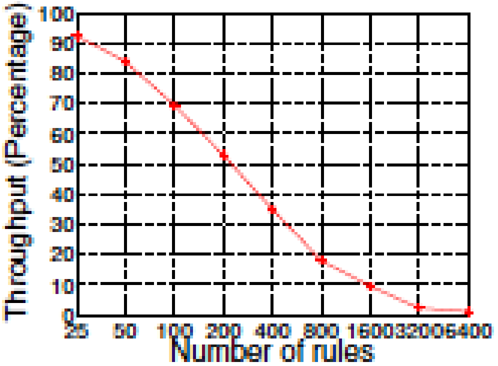


Fig 1. The effect of the number of rules on the throughput with frame size 128 bytes in

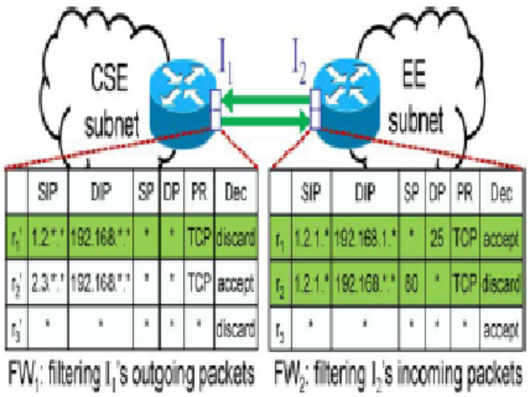
These various number of rules in a firewall significantly affects its throughput. These result of the performance test of iptables conducted by HPAC. That shows that increasing the number of rules in a firewall policy considerably reduces the firewall performance. Unfortunately, with the unstable growth of services deployed on the Internet ,firewall policies are growing quickly in size of networks.Thus, optimizing the firewall policies is critical for improving network performance.

1.2. Limitation of Prior Work

previous work on firewall optimization targets on either intrafirewall optimization, or interfrewall optimization, within the one managerial domain where the privacy protection of firewall policies is not a distress.Intrafirewall optimization means optimizing the single firewall. It is improved by either removing redundant rules, or rewriting policy rules. previous work on interfirewall optimization needs two firewall policies without any firewall privacy protection, and thus can only be used within one managerial domain. However, in actuality, it is frequent that two different firewalls that are belong to different managerial domains where that firewall policies are cannot be shared with each other firewall. Keeping firewall policies secret is important for two reasons. First one is a firewall policy may have more security holes that can be exploited by attackers. Quantitative study have shown that the most firewalls are misconfigured and have security holes. And the second one is a firewall policy often that contains private information, e.g.,the IP addresses of a servers, which can be used by third party attackers to launch more exact and targeted attacks.

1.3. Cross-Domain firewall Optimization

As we know our best knowledge, no previous work focuses on cross domain privacy protection interfirewall optimization techniques. In this paper represents the first step in explore this unknown space. exclusively, we spotlight on removing interfirewall policy duplicates in a privacy preserving manner. Let us Consider two adjacent firewalls1 and firewall2 that are belongs to different managerial domains and . Let denote the policy on firewall1's outgoing interface to firewall2 and denote the policy on firewall2's receiving interface from firewall1. For these rule in , if all these packets that are match but do not match any policy rule above in are discarded by ,rule that can be removed because those packets never come to .



	SIP	DIP	SP	DP	PR	Dec
r_1'	1.2.*.*	192.168.*.*	*	*	TCP	discard
r_2'	2.3.*.*	192.168.*.*	*	*	TCP	accept
r_3'	*	*	*	*	*	discard

FW_1: filtering I_1's outgoing packets

	SIP	DIP	SP	DP	PR	Dec
r_1	1.2.1.*	192.168.1.*	*	25	TCP	accept
r_2	1.2.1.*	192.168.*.*	80	*	TCP	discard
r_3	*	*	*	*	*	accept

FW_2: filtering I_2's incoming packets

Fig 2.Example for Interfirewall Redundant rules

We call policy rule an interfirewall redundant policy rule with respect to other firewall. Note that only filter the transfer from one to firewall2's sending interface to firewall1's receiving interface is guarded by other two different separate policies. For these simplicity, we suppose that and have no other intrafirewall redundancy as such redundancy that can be removed using the proposed solution.This can be illustrates interfirewall redundancy, where these two adjacent routers are belong to different managerial domains Cse and EE. The physical interfaces connecting two different routers are denoted as and ,respectively. The rules of the two different firewall policies and , that are used to filter the traffic flowing from Cse to EE, that are listed in two different tables following the format used in Cisco Access Control Lists.

1.4. Technical Challenges and Our Approach

The key aspect is to propose a protocol that allows two different adjacent firewalls to identify the interfirewall redundancy with respect to the other without knowing the policy rule of the other firewall. While through intrafirewall redundancy removal is so complex, interfirewall redundancy subtraction with the privacy protection requirement is even so hard. To conclude interfirewall redundant with respect to whether the rule in to positively needs some data about, cannot reveal from such data. yet,A straight forward solution is to perform some privacy protection comparison between two rules from two different adjacent firewalls. this solution have to checks whether particularly, for each polocy rule in this match a rule all available possible packets that match rule in with the abandon decision in. However, because all these redundants with respect to the firewalls follow the first match semantics rules and these rules in a firewall typically overlap, this explanation is not only incorrect and also incomplete. Incorrect means that incorrect redundant rules could. Suppose this solution clarifies as a re identified in with respect to in different firewall. However, if redundant policy rule in some packets that match rule also match rule with the accept decision in, these packet will pass, and then wants to rectify them with . In this case, is actually not redundant data. Incomplete means that, If all the portion of redundant policy rules could be identified in are not needed by not possible packets that match rule in , is also redundant. only one rule but multiple rules in However, the direct comparison solution cannot identify such duplicate results.

II. RELATED WORK

2.1. *Removal of Firewall Redundancy*

Previous work on intrafirewall redundancy removal aims to notice redundant policy rules within a single firewall. Mr. Gupta identified backward as well as forward redundant policy rules in a firewall. Later, Mr. Liu *et al.* barbed out that the redundant policy rules identified by Mr. Gupta are incomplete and proposed two different methods for detecting all redundant policy rules .previous work on inter firewall redundancy removal needs the knowledge of two various firewall policies and therefore this only applicable within a administrative domain.

2.2 Firewall Enforcement in Virtual Private Networks

Previous research on firewall enforcement in VPN's enforces firewall policies over all encrypted VPN tunnels without leaking the privacy information of the remote network's policy. This problem of collaborative firewall enforcement in VPN's and privacy protection interfirewall optimization are essentially different. First, of all their proposed a different model. The former on is focused on enforcing a firewall policy rule over VPN tunnels in a privacy preserving method, whereas the latter one is focused on removing interfirewall redundant policy rules without disclosing their policies to one other. And the second is, their requirements are different. The former one preserves the privacy protection of the remote network's policy, whereas the latter one preserves the privacy in both policies.

III. SYSTEM AND THREAT MODELS

3.1. System Model

Basically a firewall is an ordered list of semantic rules. These each rule has a decision for the packets predicate over the fields of that match the predicate value. Firewalls usually check five different fields, source IP address, destination IP address, source port number, destination port number, and protocol type. The length of these fields are 32b, 32b, 16b, 16b, and 8b bits, respectively. A packet over where each packet is a subset of domain results is a tupple where each the value is an element of the packet. A packet matches a policy rule if and holds. characteristic only if the condition firewall decisions include accept, reject, accept with logging, and reject with logging. Without loss of originality, we have to consider whether to accept or reject. In this paper. We identify a rule with the accept decision an accepting rule and a rule with the reject decision a discarding rule. In a firewall policy rule, a packet may match multiple rules whose decisions are dissimilar. To determine these problems, firewalls typically spend a first match semantics where the decision for a packet is the decision of ,is the first policy rule that matches. A matching set , is a set of all possible packets that match the rule. A result of a rule is the set of packets that match but solving set of conflicts ,do not match with any other any rule above, and is equal. And based on the above concepts, we declare interfirewall redundancy and, where redundant policy rules.

3.2. Threat Reproduction

We have to approve the semi honest model , For two adjacent different firewalls. And let we assume that they are semi honest, .that is each firewall follows our protocol exactly, but each and every firewall may not try to reveal the policy rule of the other firewall. These semi honest model is practical and well adopted. For example, In this model is suitable for large organizations and industries that have many independent branches as well as for loosely connected alliances composed by multiple different parties. While we are confident that all manageable domains follow permission protocols, we may not provide guarantee for that no corrupted employees are trying to disclose the private firewall policy rules of other parties. Also, it may be possible for one party to issue the sequence of inputs to try and reveal the other party policy.

IV.PRIVACY PRESERVING INTERFIREWALL

4.1. Privacy-Preserving Range Comparison

In this segment, we present our privacy preserving protocol for detecting the interfirewall redundant rules in with respect to . To do this,we first convert each firewall to an equivalent sequence of non overlapping rules. Because for any non overlapping rule , the matching set is equal to the resolving set i.e., ,we have only need to contrast non overlapping rules generated from the two firewalls for detecting interfirewall redundancy. And the Second, we partition this problem into two individual sub problems, single rule coverage redundancy detection and the multirule coverage redundancy rule detection, and then propose our privacy preserving scheme for solving each individual sub problem. A rule is covered by one or more multiple rules if and only if . The first sub problem will checks whether a non overlapping rule is covered by a non overlapping discarding rule. The second individual sub problem will checks whether a non overlapping rule in is covered by multiple non overlapping removal rules in , i.e., And Finally, after adding redundant non overlapping rules generated from are identified rules, we map them back to original rules of the polocies in and then identify the redundant ones.

The problem of checking whether the boils down to the problem of verifying whether one range in is contained by some another range in , which further boils down by the problem of checking whether each protocol for comparing a number and the range.Each number to the other firewall party, can first encrypt using secret key and sends to ; similarly, can first encrypt using key and sends to . Then, ether party checks. Note that if and only if . neither a party can learn anything about the numbers are being compared. It illustrates the process of checking whether to form is in estimated range.

4.2 A Multirule Coverage Redundancy Detection

To notice multiple rule coverage redundancy, our basic scheme is to merge all the non overlapping discarding policy rules from to a set of new policy rules so that for any subjective rule from , if it is covered by multiple non overlapping removal rules from , it is covered by a policy rule from these new rules. More formally, let we denote the non overlapping disposal rules from and denote the set f new policy rules generated. For any other rule from , if a non overlapping policy rule from is multirole coverage redundant, i.e., where from , there is a rule that covers, i.e.Thus, after compute the set of new policy rules from , we reduce the problem of multirule coverage redundancy detection to single rule coverage redundancy detection. Then, other two parties and can considerately run our protocol proposed in this section to identify the non overlapping single rule and multirule coverage redundant polocy rules from at the same time.

V. FIREWALL UPDATE AFTER PTIMIZATION

If or changes after in the inter firewall optimization, the interfirewall disused rules identified by the optimization may not be interfirewall deserted anymore. In this section, we discuss our solution to address firewall update. There are Main five possible cases under this scenario.

1) changes the decisions of some rules in . In this case, neither party desires to take actions because the interfirewall redundancy recognition does not consider the decisions of the rules in .

2) changes the decisions of some rules from reject to accept in . In this case, needs to notify which non rules indices of these rules from are changed. Using this in order, check if there were any rules in that were removed due to these rules, and then adds the affected rules back into .

3) changes the decision of some rules from accept to get rid of in . In this case, can run our supportive inter firewall unneeded rules in .

4) Adds or removes some rules in . In this case, since the resolving sets of some rules in may modify, a rule in that used to be inter firewall redundant maybe not redundant anymore. It is some main solution for to run our optimization protocol again.

5) Adds or removes a few rules in . Similar to the fourth case, since the resolving sets of some rules in may modify, it is significant for to run our protocol again.

VI. EXPERIMENTAL RESULTS

We estimate the effectiveness of our protocol on real firewalls and evaluate the capability of our protocol on both actual and synthetic firewalls. Our experiments were carried out on a PC running Linux with two Intel Xeon cores and 16 GB of memory.

6.1 *Evaluation Setup*

We conducted experiments over five groups of two actual adjacent firewalls. Each firewall examines five fields, source IP, destination IP, source port, destination port, and protocol. The number of rules values from dozens to thousands In implementing the commutative encryption, we used the developing Pohlig Hellman algorithm with a 1024-bit prime modulus and 160-bit encryption keys. To consider the effectiveness, we conducted our experiments Result above these five groups of adjacent firewalls. To evaluate the efficiency, for two firewalls in each group, we considered the processing time, the comparison time, and the statement cost of both parties. Due to protection concerns, it is complex to obtain a large number of real adjacent firewalls. To further evaluate the efficiency, we managed a large number of synthetic firewalls based on Singh et al.'s. The synthetic firewalls also observe the same five fields as real firewalls. The number of rules in the synthetic firewalls ranges from 200 to 2000,and for every number, we generate 10 synthetic firewalls.To quantify the efficiency, we first processed each synthetic firewall as and then measured the processing time and communication cost of two parties. Second, for two firewalls in each group, we considered the processing time, the comparison time, and the statement cost of both parties. Third, we measured the comparison time for every two synthetic firewalls. We didn't evaluate the effectiveness of our protocol on synthetic firewalls because they are generated at the random and independently without considering whether two firewalls are contiguous or not.

6.2. Methodology

In this section, we describe the metrics to determine the effectiveness of our protocol. Given our firewall optimization algorithm, and two adjacent firewalls and, we use to represent a set of interfirewall redundant rules in. Let denote the number of rules in and indicate the number of interfirewall redundant rules in. To assess the effectiveness, we define a redundancy ratio. This ratio measures what proportion of rules are interfirewall redundant in.

VII. CONCLUSION AND FUTURE SCOPE

In this paper, we identified an important problem, Minting privacy protection though firewall optimiation.We propose a novel privacy-preserving protocol for detecting such redundancy. We developed our protocol in Java and conducted extensive estimate. The results on real privacy protection though firewall optimiation policies show that our protocol can remove as many as 49% of the rules in a firewall whereas the average is 19.4%. Our protocol is related for identify the inter firewall redundancy of firewalls with a few thousands of rules, e.g. 3000 rules. However, it is still expensive to compare two firewalls with many thousands of rules, e.g. 5000 rules. Reducing the complexity of our protocol desires to be further studied. In our work, we have demonstrated rule firewall optimization, from to , and we note that a related rule optimization is possible in the opposite direction, i.e., In the first situation, to , it is that is improving the performance load of , and in return is improving the performance of in a vice-versa manner.

All this is being achieved the without or informative each other's Policies thus allowing for a proper executive partition. Our protocol is most useful if both parties are willing to benefit from it and can cooperate with each other in a mutual manner. There are many uses special cases that could be explored based on our current protocol. For example, there may be hosts or Network Address Translation (NAT) devices between two adjacent firewalls. Our present protocol cannot be directly applied to such cases. Extending our protocol to these cases could be an interesting topic and requires further analysis.

REFERENCES

[1] nf-HiPAC, "Firewall throughput test," 2012 [Online]. Available: http://www.hipac.org/performance_tests/results.html

[2] R. Agrawal, A. Evfimievski, and R. Srikant, across private databases," in *Proc. ACM SIGMOD*, 2003, pp. 86–97.

[3] E. Al-Shaer and H. Hamed, "Discovery of policy anomalies in distributed firewalls," in *Proc. IEEE INFOCOM*, 2004, pp. 2605–2616.

[4] J. Brickell and V. Shmatikov, "Privacy-preserving graph algorithms in the semi-honest model," in *Proc. ASIACRYPT*, 2010.

[5] Y.-K. Chang, "Fast binary and multiway prefix searches for packet forwarding," *Comput. Netw.*, vol. 51, no. 3, pp. 588–605, 2007.

[6] J. Cheng, H. Yang, S. H.Wong, and S. Lu, "Design and implementation of cross-domain cooperative firewall," in *Proc. IEEE ICNP*, 2007, pp. 284–293. [7] Q. Dong, S. Banerjee, J. Wang, D. Agrawal, and A. Shukla, "Packet classifiers in ternary CAMs can be smaller," in *Proc. ACM SIGMETRICS*,
2006,.

[8] O. Goldreich, "Secure multi-party computations," Working draft, Ver.1.4, 2002.

[9] O. Goldreich, *Foundations of Cryptography: Volume II (Basic Applications)*.Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] M. G. Gouda and A. X. Liu, "Firewall design: Consistency, completeness and compactness," in *Proc. IEEE ICDCS*, 2004, pp. 320–327.

[11] M. G. Gouda and A. X. Liu, "Structured firewall design," *Comput. Netw.*, vol. 51, no. 4, pp., 2007. [12] P. Gupta, "Algorithms for routing lookups and packet classification," Ph.D. dissertation, Stanford Univ., Stanford, CA, 2000.

[13] A. X. Liu and F. Chen, "Collaborative enforcement of
firewall policies in virtual private networks," in *Proc. ACM PODC*, 2008, pp. 95–104.

[14] A. X. Liu and M. G. Gouda, "Diverse firewall design," *IEEE Trans.Parallel Distrib. Syst.*, vol. 19, no. 8, pp. 1237–1251, Sep. 2008.

[15] A. X. Liu and M. G. Gouda, "Complete redundancy removal for packet classifiers in TCAMs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no.4, pp. 424–437, Apr. 2010.

[16] A. X. Liu, C. R. Meiners, and E., "TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs,"*IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 490–500, Apr. 2010.

[17] A. X. Liu, C. R. Meiners, and Y. Zhou, "All-match based completeredundancy removal for packet classifiers in TCAMs," in *Proc.*.

[18] A. X. Liu, E. Torng, and C. Meiners, "Firewall compressor: An algorithm for minimizing firewall policies 2008.

[19] C. R. Meiners, A. X. Liu, and E. Torng, "TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs," in *Proc. IEEE ICNP*, 2007, pp. 266–275.

[20] C. R. Meiners, A. X. Liu, and E. Torng, "Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs," in *Proc. IEEE*
ICNP, 2009, pp. 93–102.