A Distributed Firewall Optimization Protocol For Customized Policy Verification


Abstract—Firewalls have been widely deployed on the Internet for securing private networks. Optimizing firewall policies is crucial for improving network performance. Prior work on firewall optimization focuses on either intra-firewall or inter-firewall optimization within one administrative domain where the privacy of firewall policies is not a concern. This paper explores inter-firewall optimization across administrative domains for the first time. The key technical challenge is that firewall policies cannot be shared across domains because a firewall policy contains confidential information and even potential security holes, which can be exploited by attackers. In this paper, we propose the first cross-domain privacy-preserving cooperative firewall policy optimization protocol. Specifically, for any two adjacent firewalls belonging to two different administrative domains, our protocol can identify in each firewall the rules that can be removed because of the other firewall. The optimization process involves cooperative computation between the two firewalls without any party disclosing its policy to the other. We implemented our protocol and conducted extensive experiments. The results on real firewall policies show that our protocol can remove as many as 49% of the rules in a firewall whereas the average is 19.4%. The communication cost is less than a few hundred KBs. Our protocol incurs no extra online packet processing overhead and the offline processing time is less than a few hundred seconds.

Index Terms—Distributed firewall, Firewall optimization, Privacy control.

I. INTRODUCTION

A firewall is a system acting as an interface between a network and one or more external networks. It helps implementing the security policy of any network by deciding which packets to let pass through and which to block, based on the set of rules defined by the network administrator. Any error in defining the rules may compromise the system security by letting undesired traffic pass through or blocking the desired traffic. The rules when defined manually often results in a set that contains conflicting, redundant or overshadowed rules, which creates anomalies in the firewall policy. A network firewall protects a computer network from unauthorized access. Network firewalls may be hardware devices, software programs, or they may be a combination of the two. Network firewalls guard an internal computer network (home, school, business intranet) against malicious access from the outside. Network firewall may also be configured to limit access to the outside network of internal users.

II. RELATED WORKS

A. FIREWALL REDUNDANCY REMOVAL

Prior work on intrafirewall redundancy removal aims to detect redundant rules within a single firewall [12]. Later, Liu et al. pointed out that the redundant rules identified by Gupta are incomplete and proposed two methods for detecting all redundant rules. Prior work on interfirewall redundancy removal requires the knowledge of two firewall policies and therefore is only applicable within one administrative domain.

B. COLLABORATIVE FIREWALL ENFORCEMENT IN VIRTUAL PRIVATE NETWORKS (VPNS)

Prior work on collaborative firewall enforcement in VPNs enforces firewall policies over encrypted VPN tunnels without leaking the privacy of the remote network's policy.
The problems of collaborative firewall enforcement in VPNs and privacy-preserving interfirewall optimization are fundamentally different. First, their purposes are different. The former focuses on enforcing a firewall policy over VPN tunnels in a privacy-preserving manner, whereas the latter focuses on removing interfirewall redundant rules without disclosing their policies to each other. Second, their requirements are different. The former preserves the privacy of the remote network’s policy, whereas the latter preserves the privacy of both policies.

C. PRIVACY-PRESERVING INTERFIREWALL REDUNDANCY REMOVAL

To detect the redundant rules in FW1, converts its firewall to a set of non-overlapping rules. To preserve the privacy of its firewall, first converts each range of a non-overlapping discarding rules from to a set of prefixes. Second and encrypt these prefixes using commutative encryption

C. PROCESSING FIREWALL FW2

In order to compare two firewalls in a privacy-preserving manner NET1, and NET2 convert firewall FW2 to d sets of double encrypted numbers, where d is the number of fields. The conversion of FW2.

IV. PROPOSED APPROACH FOR CUSTOMIZED POLICY VERIFICATION IN DISTRIBUTED FIREWALL OPTIMIZATION

In this paper, we propose the first cross-domain privacy-preserving cooperative firewall policy optimization protocol. We implemented our protocol and conducted extensive experiments. The results on real firewall 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 incurs no extra online packet processing overhead and the offline processing time is less than a few hundred seconds.

V. SIMULATION SETUP AND RESULT DISCUSSION

We evaluate the effectiveness of our protocol on real firewalls and evaluate the efficiency of our protocol on both real and synthetic firewalls. We implemented our protocol using Visual Studio 2010 and SQL Server 2008. Our experiments were carried out on a PC running Windows 7 with two Intel Core i5 processors and 16GB of memory.

VI. CONCLUSION

In this paper, we identified an important problem, cross-domain privacy preserving interfirewall redundancy detection. We propose a novel privacy-preserving protocol for detecting such redundancy. We implemented our protocol in Java and conducted extensive evaluation. The
results on real firewall policies show that our protocol can remove as many as 60% of the rules in a firewall whereas the average is 20.4%. Our protocol is applicable for identifying the interfirewall redundancy of firewalls with a few thousands of rules, e.g. 2000 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 needs to be further studied. In our work, we have demonstrated rule optimization, from to , and we note that a similar rule optimization is possible in the opposite direction, i.e., to . In the first scenario, 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 without or revealing each other’s policies thus allowing for a proper administrative separation. Our protocol is most beneficial if both parties are willing to benefit from it and can collaborate in a mutual manner. There are many 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 current protocol cannot be directly applied to such cases. Extending our protocol to these cases could be an interesting topic and requires further investigation. We implemented our protocol and conducted extensive experiments. The results on real firewall 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 incurs no extra online packet processing overhead and the offline processing time is less than a few hundred seconds.

ACKNOWLEDGMENT

We take immense pleasure in expressing our humble note of gratitude to our project guide Mrs. S. Vaishnavi, Assistant Professor, Department of Computer Science and Engineering, SNS College of Technology, for her remarkable guidance and useful suggestions, which helped us in completing the paper before deadline.

REFERENCES

[6] Alex X. Liu, Member, IEEE, and Mohamed G. Gouda, Member, IEEE “Diverse Firewall Design”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 19, NO. 8, AUGUST 2008
[8] MyungKeun Yoon, Shigang Chen, and Zhan Zhang, ”Minimizing the Maximum Firewall Rule Set in a Network with Multiple Firewalls”, IEEE TRANSACTIONS ON COMPUTERS, VOL. 59, NO. 2, FEBRUARY 2010