An Experimental Study on Firewall Performance: Dive Into the Bottleneck for Firewall Effectiveness

Chenghong Wang[†]
Computer Science and Techology Dept.
Harbin Engineering University
Harbin, China
Email: wangchenghong@hrbeu.edu.cn

Donghong Zhang[†]
Computer Science and Techology Dept.
Harbin Engineering University
Harbin, China
Email: zhangdonghong@hrbeu.edu.cn

Hualin Lu
Computer Science and Techology Dept.
Harbin Engineering University
Harbin, China
Email: luhualin@hrbeu.edu.cn

Jing Zhao[*]
Computer Science and Techology Dept.
Harbin Engineering University
Harbin, China
Email: zhaoj@hrbeu.edu.cn

Zhenyu Zhang
Institute of Software
Chinese Academy of Sciences
Beijing, China
Email: zhangzy@ios.ac.cn

Zheng Zheng
Beijing University of
Aeronautics and Astronautics
Beijing, China
Email: zhengz@buaa.edu.cn

***Abstract*—Performance is an important indicator of firewalls effectiveness, which represents capability of firewalls handling network requests. ModSecurity and iptables, two representative firewalls of packet filtering and application firewall, are studied experimentally in this paper. Firstly, we develop the experiments to test the capacity of these two kinds of firewalls. Secondly, we locate the bottlenecks for system resources such as CPU and memory usage that affect the firewalls performance by analyzing the collecting data from firewalls experiments. Finally, with the same settings, we compare the performance of the two kinds of firewalls by varying the parameters such as request rate, packet length, and maximum concurrent connections.**

***Keywords*-packet filtering firewall; application firewall; performance bottleneck; network security; hardware resources**

I. Introduction

With the development of Internet, the threat of cyber attack has become more and more serious. A new pattern of cyber attack named APT *(Advanced Persistent Threat)* [1] attack comes into being, which marks the beginning of a new Internet era of sophisticated attacks. Firewall is a crucial facility for network security reinforcement. It protects private regions by filtering malicious network streams from outside. Firewalls are now classified into two kinds of products: packet filtering firewalls and application firewalls [2]. Each kind of firewall possesses a large family of mature products. Because of firewalls efficient capabilities of inspecting network streams safety, firewalls are now the cornerstones of the security infrastructure for most enterprises [3].

Firewall performance has become a major concern for the security defense, and its performance degradation may destroy the data integrity, even for the system confidential and availability. On the one hand, the above elementary attributes belong to the necessary elements of the system security [4]. On the other hand, capacity as an important indicator of firewalls performance represents the maximum capability that firewalls can reach when handling network packets, some kinds of factors such as CPU and memory usage for the system resource may become the bottlenecks of the firewalls capacity. The analysis of the firewalls bottlenecks can help the system administrator allocate the system resources effectively to tune the firewalls performance.

Application firewalls such as *ModSecurity* run on the transmission layer and application layer. Application firewalls inspect the contents of network packets while packet filter firewalls just inspect the head of each packet. Thus, packet filter firewalls usually have better performance than application firewalls. However, the specific performance difference between these two kinds of firewalls under equivalent configurations still remains unclear. Hence, it is necessary for us to compare the performance of application firewalls and packet filter firewalls.

We present a comparative experimental study of *iptables* and *ModSecurity* which are two representative firewalls for packet filter firewalls and application firewalls, respectively. The background and related works of testing firewalls are introduced in Section II. Experimental setups and specific experimental plans are discussed in Section III. For each firewall we set two testing scenarios to test the firewall performance dealing with high speed network packets and large size of packets. In Section IV, performance comparisons between two firewalls are presented based on experimental results. Then we discuss the experimental results in Section V. Finally, we make a conclusion and introduce the future

[*]Corresponding Author : Jing Zhao (zhaoj@hrbeu.edu.cn)
[†]Co-first Author : Chenghong Wang, Donghong Zhang

978-1-4799-8099-4/14/$31.00 ©2014 IEEE

work in Section VI.

II. Background and Related Work

Daniel Hoffman et al. [5] made an experiment on *iptables*, a kind of packet filter firewall running on the kernel level, in order to observe the performance of *iptables*. However, the authors just observed the performance data when *iptables* faces three kinds of request rate including 10Mbps, 100Mbps and 1000Mbps, and varying frame length. The observed data sets are not sufficient to analyze the firewalls capacity. Micheal et al. [6] reported the performance of three distributed packet filtering firewalls under different types of flood attack. They tested each firewall under the network flood with different request rate and different protocol type. Though this experiment contained more performance results under different situations, these experimental results were still not enough for precise capacity analysis. Some factors like maximum concurrent connections should be added into experiment design. Additionally, Micheal et al. didnt intend to take some measures to reduce the noises made by intermediate devices like routers in their experiments. Hence, their collecting data from experiments may not be reliable. Mohd et al. [7] set up experiments probing into application firewalls. An application firewall working at the 7th layer of the OSI model was tested in this experiment. Kostnick et al. [8] made an exhaustive performance test of application firewalls in their experiments. They focused on the firewalls performance dealing different kinds of application layer protocols like E-MAIL, FTP and HTTP. All these previous researchers just focused on either application firewall or packet filter firewall, but didnt make some comparisons between these two kind of firewalls. Moreover none of them did any study on the relationship between firewall performance and the hardware resources.

III. Experiment Plan

A. Experiment Setup

The experimental setups consist of a target host server running Apache under the protection of tested firewall, and a payload generator client connected via CAT5 Ethernet Cable directly. The two hosts have the same hardware configuration of 2.4 GHz Intel CPU, 6G DDR3 1600Mhz memory and 1000M Ethernet Adapter. We develop the experimental components including the payload generator and testing firewalls as shown in figure.1, where payload generator at client side is used to generate the client workloads, and the tested firewalls running on the server side are used to deal with the packets before transferring the packets to the Apache web server. The corresponding configurations of the components are explained in detail next.

In figure.1, a special CAT5 Ethernet cable called *Crossover Ethernet Cable* [9] is utilized to connect the payload generator client and the web server. We do not use the switches and routers to connect the server and the payload generator, since the intermediate devices may become the network traffic bottleneck, so that just the partial generated workloads can arrive at the firewall due to the bottlenecks introduced by intermediate devices. Thus, this direct link mode could significantly reduce the irrelevant noises made by some such as switches and routers when observing firewall performance. Our payload generator could load target host with stable network bandwidth at 110 Mbps, while it could only reach 30 Mbps through the forwarding of routers or switches.

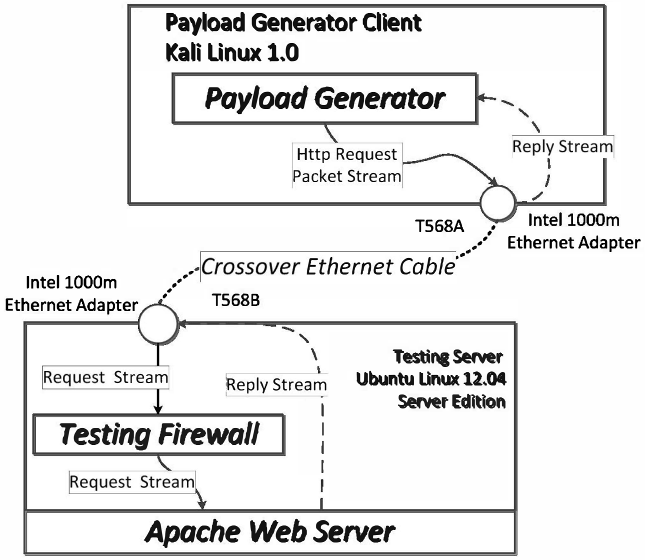


Figure 1. Experimental platform

We write the script called the *Firewallperf* as a payload generator based on *Httperf* [10]. The *Firewallperf* can invoke the *httperf* to generate the workloads [11] by setting the parameters such as request rate, packet size, and concurrent connections, respectively. The reply rate provided by *httperf* is an important indicator of the performance, and other measurements like errors and response time are also important for performance analysis [12].

We use Apache 2.22 server as our web service support platform. Tested firewall, running in front of the web server, hooks all the requests firstly and checks each request packet by comparing them with filter rules, then forwards safe requests to the Apache web server. We develop two monitors on the web server and the payload generator client, respectively. The web server side monitor is used to inspect the hardware resources usage (i.e. CPU usage and memory usage), and the client side monitor is responsible for logging the firewalls performance indicators in terms of response information, reply rate and frame loss ratio.

In order to ensure the equivalent testing conditions for *iptables* and *ModSecurity*, we import the same number of security rules into both firewalls. On the one hand, *OWASP-CRS ModSecurity core rules* set [13] is imported

Group	Experimental Var	Var Scope	Unit	Control Method
(S1)	Request Rate	500-3000	times/s	Increase request rate with 25 times/s
(S2)	Packet Size	256-7680	bytes	Increase frag size with 256 bytes

Table I
TEST CASES DESIGN DETAILS

as the filter rules for *ModSecurity*, which could block many typical offensive activities such as SQL Injection, Command Injection, XSS, and Burp Force requests, etc.. We remove the rule *modsecurity_crs_21_protocol_anomalies.conf* since it limits the maximum request rate of individual client. In this way, payload generator can create enough workloads on tested *ModSecurity*. On the other hand, we randomly generate 6700 security rules for *iptables* which equal the total number of *OWASP-CRS ModSecurity core rules*. The *iptables* rules have no maximum request rate limitation, so that all the requests made by the payload generator can reach web server through testing firewall.

B. Experiment Setup

We conduct experiments for each tested firewall by varying parameters request rate, packet size, respectively. In our experimental platform, the firewall firstly checks each request packet by comparing them with filter rules, then forwards safe requests to the Apache web server. Thus, the capacity of the Apache web server should be guaranteed to surpass far away from the capacity of the firewalls. In this way, the corresponding contrast experiment without firewall with the same settings is set up in our experimental plan. 5 replications for each experiment are used for statistically calculations for performance indicators. So that we have 30 replications for varying parameters request rate, packet size, and also we have 15 replications for contrast experiments. The monitored interval is 2 seconds for the CPU and memory usage, and we also collect the average response time, average reply rate and average frame loss ratio at each replication. Three experiment scenarios are discussed as follows:

1) Varying Request Rate: This experiment is designed for probing into the tested firewalls performance when handling network packets by varying request rate. The generators request rate increases gradually from 500 requests/s to the maximum requesting rate, 3500 requests/s, with the increment of 25 requests/s, and payload generator generates 60000 requests at each request rate. Meanwhile, we fix packet size to 16 bytes, without any concurrent connections.

2) Varying Packet size: In this experimental scenario we need to analyze firewalls performance when dealing with large size of network packets. Packet size increases gradually from 512 bytes to 7680 bytes with the increment of 256 bytes. Payload generator sends total 10000 packets to the web server through testing firewalls at each packet size. We also fix the request rate to 200 times/s and set just one request sessions.

TABLE I shows the details of the two testing scenarios of each experiment, including the scope of experimental variables, units of experiment variables and control methods for the two experiment scenarios, S1, S2, respectively.

IV. RESULTS ANALYSIS

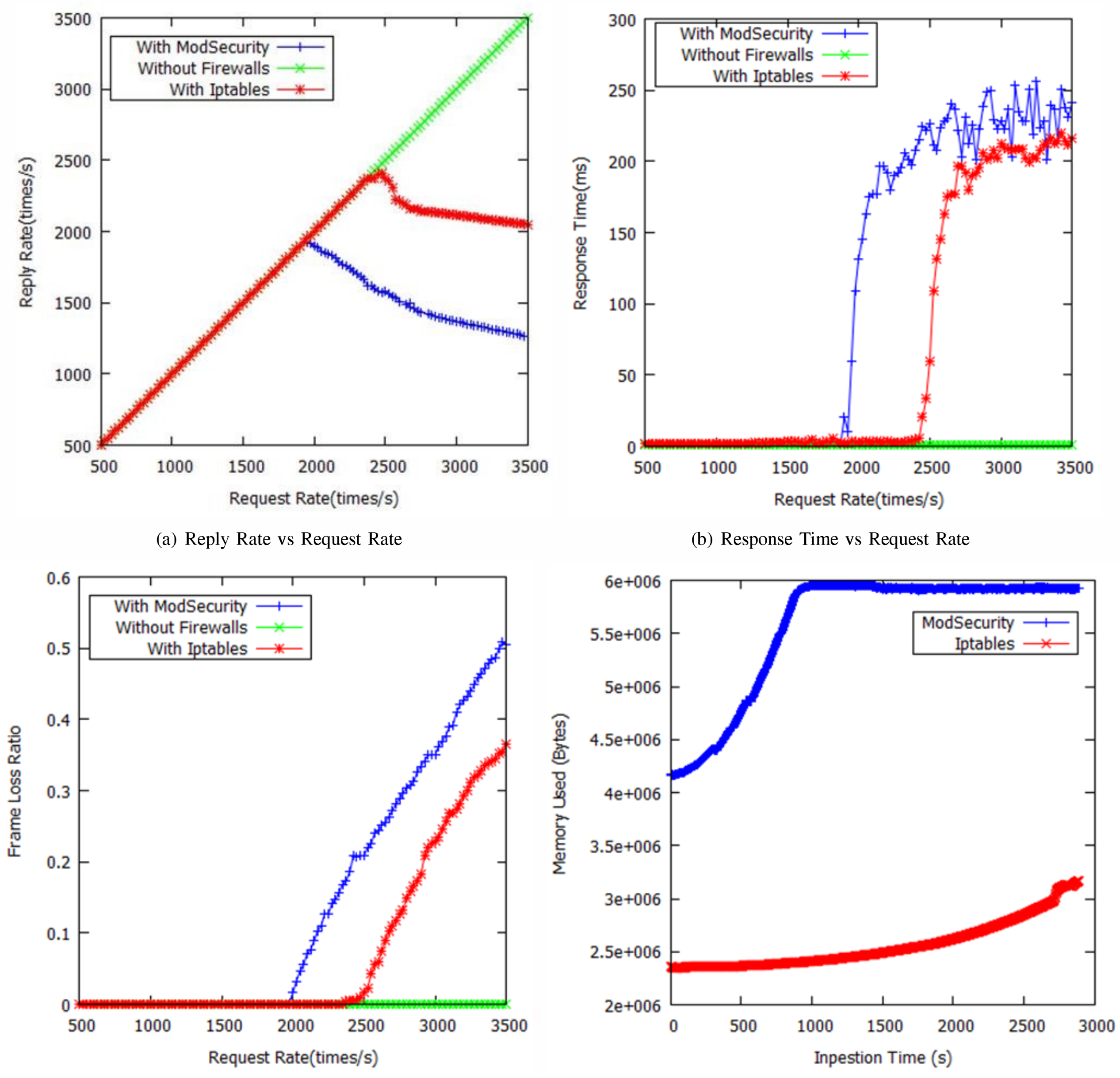
A. Varying Request rate Results and Bottleneck analysis

It is clear to see that the capacity of Apache web server is far beyond that of *iptables* and *ModSecurity* by contrast experiment without firewalls. Figure.2(a) shows that by varying the request rate, the reply rate for the *iptables* and *ModSecurity* reach maximum 2025 times/s, and 1925 times/s, respectively. After this value, the reply rates for both firewalls decrease with the varying request rate, meanwhile from figure.2(c), we can see that the frame loss ratios for *iptables* and *ModSecurity* increase sharply at the request rate of 2025 times/s, and 1925 times/s, respectively. Therefore, from figure.2(a),(c) , we can conclude that the capacity of *iptables* as well as *ModSecurity* is 2025 times/s, and 1925 times/s, respectively. From figure.2(b) we can see that the response time for both firewalls increase sharply when they reach their capacities.

Figure.2(d) shows that the memory usage for both firewalls. We can see that the CPU usage for the user mode and kernel mode of *iptables* and *ModSecurity* increases initially with experimental process, and then saturated at the capacity, respectively. It is clear to see that memory usage for *ModSecurity* starts at the high value compared with *iptables* and then exhausted at the full allocation memory. As a result, the bottlenecks for *ModSecurity* are due to effects of the combinations of CPU and memory usage, while the bottlenecks for the *iptables* are largely due to the effect of the CPU usage.

B. Varying Packet size Results and Bottleneck analysis

The experimental results of varying packet size are show in figure.3. Figure.3 (a) shows that by varying the packet size, both kinds of firewalls reply rate keep stable at the early period, but the reply rate for *ModSecurity* decreases sharply after the packet size increases to 1526 bytes. We also see that *iptables* reply rate declines slightly after the request packet size raises to 4096 bytes. Meanwhile, from figure.3(b), (c), we can see a significant increasing tendency of response time and frame loss ratio for *ModSecurity*, while the slow increasing trend for *iptables*. Furthermore, from figure.3(b), (c), when packet size of request requests increases to 6144



(a) Reply Rate vs Request Rate

(b) Response Time vs Request Rate

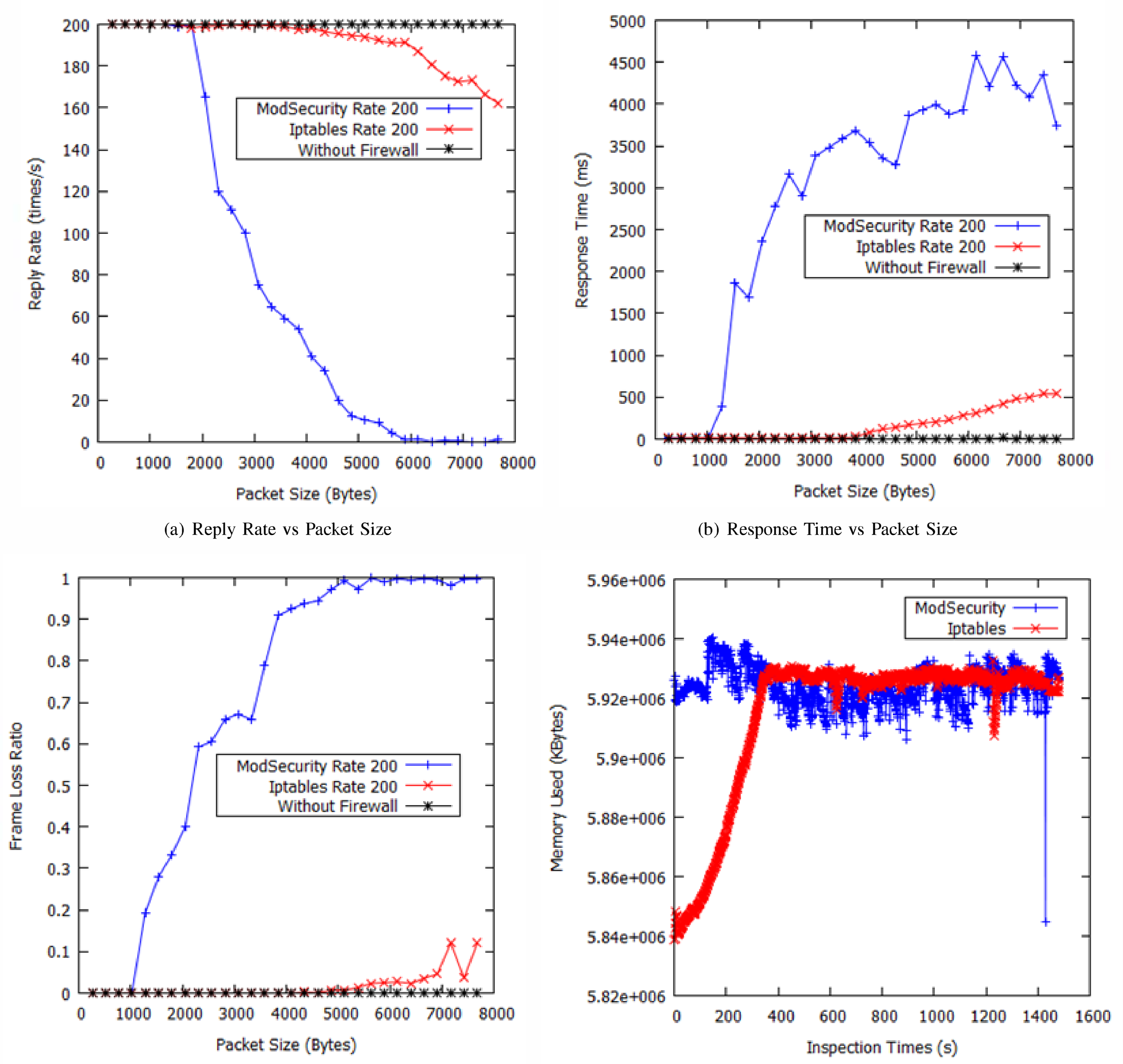
(c) Frame Loss Ratio vs Request Rate

(d) Memory Usage vs Request Rate

Figure 2. Experiment Result of Varying Request Rate

bytes, response time as well as frame loss ratio reaches a very high level, and no reply happens. It is clear to see that *iptables* performs much better than *ModSecurity* with low response time value, high reply rate and low frame loss ratio.

The huge gap between two firewalls performance could be explained by the difference between resources consumption rate. Server side monitor data is obtained by figure.3 (d) which could help us to analyze the resources consumption difference. Figure.3(c) shows that in this scenario, both firewalls have large requirements for memory resources. The memory usage for *ModSecurity* reach memory limit as soon as the experiment starts and the memory for *iptables* also increases quickly and soon reaches memory limit. However , firewalls performance of handling large size of data may not affect decisively by memory limits. Because even the worse one, which is *ModSecurity*, in this testing scenario can run with good performance in a short period at the beginning, not mention to the *iptables* which runs entire testing period with little performance loss. Meanwhile, Figure.3(a), (b), (c), (d) show us that memory limit would affect *iptables* performance slightly, so that *iptables* can still work well



(a) Reply Rate vs Packet Size

(b) Response Time vs Packet Size

(c) Frame Loss Ratio vs Packet Size

(d) Memory Usage vs Pakcet Size

Figure 3. Experiment Result of Large Packet Scenario

due to plenty of CPU resources.

V. Discussion

In this paper, we develop two groups of experiments to test *iptables* and *ModSecurity* by varying parameters such as request rate, packet size, respectively. We conclude the capacity of *iptables* and *ModSecurity* by varying the request rate and fixing the other kinds of parameters, and also we obtain that performance of *iptables* is better than *ModSecurity*. At the second group experiment, we fix the request rate and concurrent connections to study the performance of both firewalls with varying packet size, we find that the performance of *ModSecurity* is affected much more by packet size compared with *iptables* at this case. Experimental results for these two groups show that *iptables* performs better than *ModSecurity*.

VI. Conclusion and Future Work

With the experiments of two parameters, request rate and packet size, we get the conclusion that *iptables* works better than *ModSecurity* in most cases. The reason for *iptables* being better possibly dues to fewer hardware resources requirements for its just handling packet heads, so that leading to lower resources consumption rate. In the process of testing the firewalls, we find the concurrent connection may alsio affect the performance of both firewalls, so we will make a deep research on the relationship between firewalls and concurrent connections.

Acknowledgment

Thanks for the anonymous reviewer as well as the valuable advice. This research is supported in part by the Opening Fund of the State key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences (SYSKF1405).

References

[1] C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Network Security*, vol. 2011, no. 8, pp. 16 – 19, 2011.

[2] K. Ingham and S. Forrest, "A history and survey of network firewalls," *University of New Mexico, Tech. Rep*, 2002.

[3] A. Liu, "Firewall policy change-impact analysis," *ACM Trans. Internet Technol.*, vol. 11, no. 4, pp. 15:1–15:24, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/2109211.2109212

[4] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," *Nist special publication*, vol. 800, no. 30, pp. 800–30, 2002.

[5] D. Hoffman, D. Prabhakar, and P. Strooper, "Testing iptables," in *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2003, pp. 80–91.

[6] M. Ihde and W. H. Sanders, "Barbarians in the gate: An experimental validation of nic-based distributed firewall performance and flood tolerance," in *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*. IEEE, 2006, pp. 209–216.

[7] M. Z. A. Aziz, M. Y. Ibrahim, A. M. Omar, R. Ab Rahman, M. Md Zan, and M. I. Yusof, "Performance analysis of application layer firewall," in *Wireless Technology and Applications (ISWTA), 2012 IEEE Symposium on*. IEEE, 2012, pp. 182–186.

[8] C. Kostnick and M. Mancuso, "Firewall performance analysis report," *Computer Sciences Corporation-SSC-NSD, Hanover MD*, 1995.

[9] D. Jansen and H. Buttner, "Real-time ethernet: the ethercat solution," *Computing and Control Engineering*, vol. 15, no. 1, pp. 16–21, 2004.

[10] D. Mosberger and T. Jin, "httperfa tool for measuring web server performance," *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, no. 3, pp. 31–37, 1998.

[11] B. Hickman, D. Newman, S. Tadjudin, and T. Martin, "Benchmarking methodology for firewall performance," *RFC3511, April*, vol. 16, 2003.

[12] L. Li, K. Vaidyanathan, and K. S. Trivedi, "An approach for estimation of software aging in a web server," in *Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n*. IEEE, 2002, pp. 91–100.

[13] I. Ristic, *ModSecurity Handbook*. Feisty Duck, 2010.