Adaptive Reorientation Method for Performance Enhancement in Network Firewalls

M B Subrahmanyam
MVGR College of Engineering
Vizianagaram, AP

P Ravi Kiran Varma
MVGR College of Engineering
Vizianagaram, AP

ABSTRACT
Firewall plays a crucial role in network defense and perimeter security. The performance of such a firewall greatly depends on number of rules processed per packet and the order of the rules as well. In this paper an Adaptive Reorientation Method (ARM) was proposed, which will calculate the weight of each rule, after few cycles of traffic simulations. The rules are then reoriented according to their weights. The firewall is configured using several Access Control Lists (ACL) and using the ARM priority of the rules are calculated and are reoriented accordingly. The performance of the firewall is evaluated and compared before and after orientation.

Keywords
Network firewall, Performance enhancement, Adaptive Reorientation.

1. INTRODUCTION
Firewalls are used as a first-line barricade in a protected network. Network performance highly depends on effectualness of the firewall, and it is a crucial part of a perimeter security. It is the one which decides which traffic should enter in and out of the network. Firewalls are configured at starting point of a secured network and it interrogates incoming and outgoing traffic. Every firewall contains a set of rules. Every packet passes through the firewall rules with a top to bottom approach as shown in fig.1[14]. The network packet is processed by the firewall against each rule and when the packet matches a rule, the firewall will take a decision to either allow or drop it.

Firewall techniques are categorized into four types: Packet interrogation, Stateful packet interrogation, Application proxies and Dynamic packet interrogation [15].

Packet filtering firewalls are the most basic firewalls. Most of the layer-3 devices have packet filtering built-in but the problem with these routers is that they don't provide extensive logs. Stateless firewall does not remember information about the passing traffic. These firewalls are not smart and can be simply attacked or fooled by intruders. These firewalls cannot identify the packet that contains a malicious code.

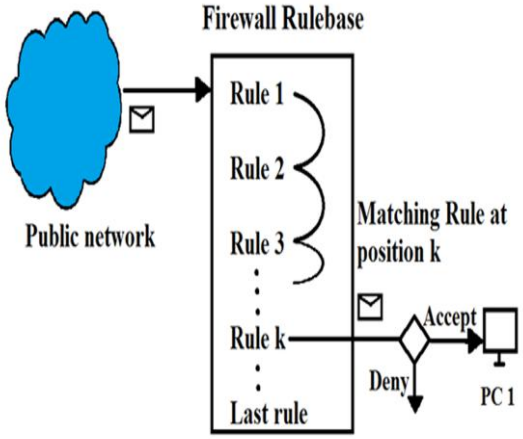


Fig 1: Interrogation of firewall rules

Stateful packet filtering firewalls are the second generation firewalls. They contain connection tables. They require a separate memory space for storing the connection table information. The dissection is based on rule set and connection tables. Stateful firewalls are smart firewalls as they interrogate attacks like IP-spoofing.

Application Firewalls or Application Proxies are the firewalls which function on the server and client model, acting as an agent between the systems which intend to communicate. Application firewalls have a relationship with layer 3 to 7 (Network, transport, session, presentation and application layer) of the OSI layers. These firewalls are more secure ones, but more complex and with the lowest performance. This type of firewalls make decisions based on the packets sent by each application and implement authentication for certain protocols.

Dynamic packet filtering firewalls can monitor the state of active connections and store this information. This information will help whether the packet is allowed or not (interrogation). These firewalls come under fourth generation. It records the information of both the IP address and the port numbers. These firewalls create a secured posture than a normal firewall.

In any Organization, the packet filters control the packet traffic across firewall according to a given filtering policy. That filtering policy is expressed in an Access control list. Access control list interrogate the packet using the packet header information. Packet header contains source and destination addresses, source and destination port addresses, protocol type, and other fields. The main problem of stateless firewalls is that the filtering decision is based on individual packets. Stateless packet filters have to face the problem while filtering packets of a multi-session application protocols like

FTP or SIP. Packet filters cannot relate the flows since they operate from the top layers (Network and Transport layers). The main challenges of firewall are performance, availability and complexity.

Distributed Denial of Service (DDoS) attack can create big problem to the networks and it is one of most serious attacks. DDoS attacks are conducted with primary intent of targeting a network resource. According to Arbor report, at the first quarter of 2012 the tolerable size of DDoS attacks was 1.77 Gbps, later it is increased by 19.5% in the same year. When Compared to 2011 the ddos attacks are increased by 200% in fourth quarter of 2012[13].

This paper extends work presented in [14]. The most worthy of the work includes a Adaptive Reorientation Method. In this model changes the rules order based on the priority base. What are the rules are most triggered then that rules will change the order to top position.

Firewall rules that provide access to a bunch of services in a network, and securing the valuable assets from incursion, tend to become very large in size. As rule bases become large, Administrators become indecisive to revise predefined rule-set and instead add new rules for the fear of causing a disadvantageous impact on existing service. Over the time, there will be negative impact on firewall performance due to increasing the size of fire wall bases and it also requires more effort for making changes. It is therefore extremely important to modify the order of rules by sending these most triggered rules to the top position base.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents Adaptive Reorientation Method. Section 4 is the experimental setup. Section 5 presents performance result of network firewall. Section 6 concludes the paper and describes the future work.

2. RELATED WORK

The literature consists of firewalls being used till day in many organizations and the techniques used for improving the firewall performance.

Every Firewall has a rule-based engine. Firewalls are of two types: Commercial and Open source. Firewalls like Cisco PIX fall under commercial type and in open-source type Linux Net-filter and Free BSD ipfw as reported in [1-5].

H.Hemed, A.El-Atawy and E.Al-shaer addresses two important problems related to packet filtering that are not yet thoroughly explored in research: first one is the early rejection of unwanted packets and second one is optimizing packet filtering based on traffic statistics. They presented techniques, algorithms and evaluation study to tackle each problem effectively [7].

K. Salah presented an analytical model to study and analyze the performance of rule based firewall. This model can be used to measure the performance when the firewall is subjected to normal and Dos attack flow targeting different rule positions [8].

Huirong Fu, Ming Zhang proposed an on-line adaptive firewall allocation schema that can simultaneously achieve high utilization, quality of service and security requirements with dynamic workload [9].

Ray Hunt, Theuns Verwoerd demonstrates some of the advanced technique with improved performance for conditional firewall [10]. Implementing a self protected system whose main characteristics are 1) to minimize the confusion on the managed system while providing a high reactivity, 2) to automate the configuration security components when the system components when the system evolves, 3) to keep the protection on the manager independent from the protected legacy system [11].

3. Adaptive Reorientation Method

This section presents an Adaptive Reorientation Method based on priority. The main objective of rules reorientation is to increase the performance of the host based firewall. It can decrease the packet interrogation time using the Adaptive Reorientation Method. When a high triggering rule's position change to the top position, the firewall can interrogate more number of packets compared to the normal condition.

Algorithm for Adaptive Reorientation Method**:**

Step 1:- Receiving the packets from source

Step 2:- Sending the packet to the Firewall Rule base

Step 3:-

```
IF (packet matches the all conditions of the rule Ri)
        Initially packet weight is W=0;
                IF (decision == ACCEPT)
                WR[i]=WR[i]+1
    // increase the W value for the matched rule R[i]
Sending packet to Destination
                Else
                IF (decisions == Deny)
                Drop the packet
    //Drop the packet without sending to Destination
```

Step 4:-

```
After certain intervals reorientation of the Rule using Weight
IF (certain time limit)
For (i=0; i<length.ruleset; k++)
  For (j=1; j< length.ruleset; j++)
                IF (R[k] < R[i])
                        Max =WR[i];
                        WR[i] = WR[j];
                        WR[j] = Max;
            Else
                    No change in Order of rules
                End IF
  End For
End For
Neutralize all the WR values to Zero
End IF
```

Note: WR [5]→ 5th Rule Wight (number of Hits)

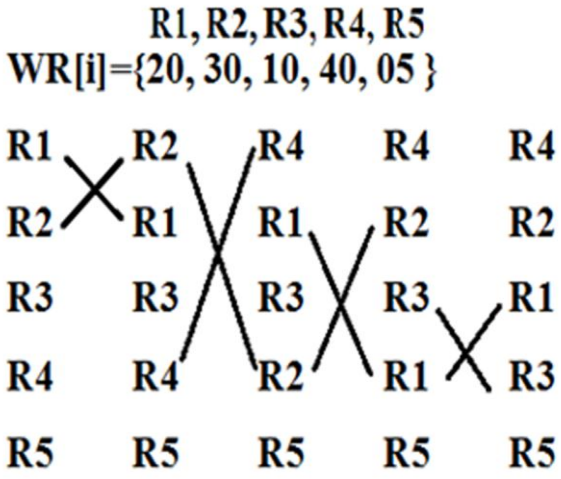
Weight is nothing but the number of hits generated in a certain time limit.

Ex: in rule-set a Rule K condition is matched N times the value of WR[K] is N. WR[K]= N

When a host sends packet to the destination that packet reaches the firewall then the packet is forwarded to the Rule-Base (Rule-set). The packet gets interrogated with the Rule-set through the top to down approach as show in the Figure 1. When a packet satisfies all the conditions of a Rule *n* then it will check with the decision: Accept || Deny.

If the condition is Accept, increment the value of WR[*n*] with 1, forward the packet to a respective destination IP address, otherwise Deny the packet without incrementing the value of WR[*n*] and then interrogate the other packets.

After a certain period of time, change the rules, get ordered using the value of WR[*i*]. Previously the W value to all rules are calculated when a packet matches all conditions of a rule. With the help of W values reorient the rules using the technique present in step 4. After the reorientation the rules-set neutralize the value of W.



In the above figure, the Rules are having some W values based on the values sorting takes place. At starting it is assumed that R1 has the Maximum value and then it is checked with other rule values. If the other rule is having the highest value, then the positions are swapped and one cycle completes. Continue the same task *n-1* time in this case n is the number of rules in the rules-set.

Using simulations Adaptive Reorientation Method applied to the rule-set based on the triggering value. This technique is constrained to some scope.

Scenario 1: firewall drops the packets from some X-network except one host. First write the accept rules flowed by deny rule. When a W value of the deny condition is greater than the allow condition, the rules changes the position. That specific host is not allowed to the network.

R1-access-list 101 permit ip 218.199.48.1 host 192.168.0.0 0.0.255.255

R2-access-list 101 deny ip 218.199.48.0 0.0.15.255 192.168.0.0 0.0.255.255

After reorientation of the rules

R2-access-list 101 deny ip 218.199.48.0 0.0.15.255 192.168.0.0 0.0.255.255

R1-access-list 101 permit ip 218.199.48.1 host 192.168.0.0 0.0.255.255

In above example the source ip 218.199.48.1 will be allowed by the firewall and remaining network must be dropped. After the reorientation of the rules-set, the source ip 218.199.48.1 of the traffic is not allowed by firewall.

To overcome this problem, the first step is to group the dependent rules in one place. Second, when the rule matches the condition, assign the priority values to the entire group. Third, reorder the rules based on the priority values.

4. EXPERIMENT SETUP

The experimental setup contains one Linux machine and three Windows machines connected using Gigabit Ethernet links as shown in figure 2. Linux machine is configured with CentOS 6.3. CentOS operating system is configured as a host-based firewall with the help of iptables.

Host based Firewall is placed at starting point of a secured network. Each network packet which enters or leaves the network a judgment has to be interrogated by host based firewall whether to accept it or reject it.

Other machines will be configured with the Windows XP operating system. For generating the DDoS attacks, install LOIC tool and supporting .net plug-in. The remaining machines are configured with Windows or Linux operating systems.

LOIC tool can generate a DDoS traffic targeting the destination machine ip-address. It can generate different type of traffic i.e. tcp, udp or http traffic.

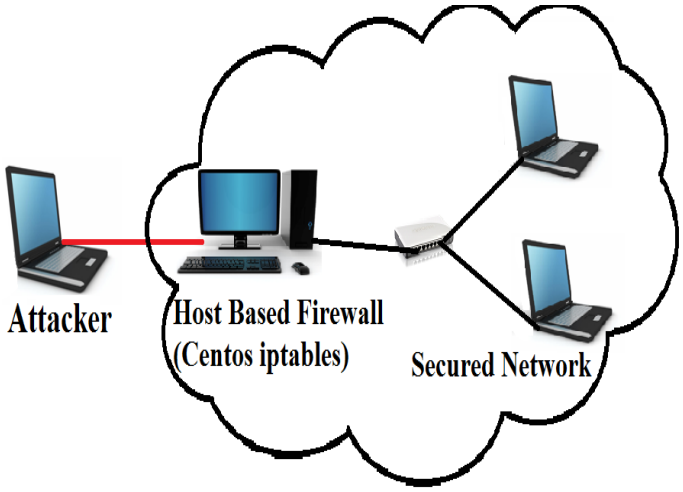


Fig 2:- Experimental Setup

In above figure the Firewall is configured with n number of iptable rules. The configuration file is stored in /etc/sysconfig/iptables path. iptables can start using with "service iptables start" command. After any change made in the configuration, the necessary task is to restart the iptables using "service iptables restart" command. Iptables can stop using with "service iptables stop" command and using "service iptables status" command check the status of the iptables.

```
[root@localhost sysconfig]# service iptables restart
iptables: Flushing firewall rules:                         [  OK
iptables: Setting chains to policy ACCEPT: filter          [  OK
iptables: Unloading modules:                               [  OK
iptables: Applying firewall rules:                         [  OK
[root@localhost sysconfig]# service iptables stop
iptables: Flushing firewall rules:                         [  OK
iptables: Setting chains to policy ACCEPT: filter          [  OK
iptables: Unloading modules:                               [  OK
[root@localhost sysconfig]# service iptables start
iptables: Applying firewall rules:                         [  OK
[root@localhost sysconfig]# service iptables status
Table: filter
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       tcp  --  192.168.5.50         0.0.0.0/0
2    DROP       tcp  --  192.168.5.50         0.0.0.0/0
3    DROP       all  --  192.168.5.50         0.0.0.0/0
4    DROP       icmp --  192.168.5.50         0.0.0.0/0
5    DROP       icmp --  0.0.0.0/0            0.0.0.0/0
6    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
7    ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
8    ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
9    ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0
10   REJECT     all  --  0.0.0.0/0            0.0.0.0/0
d

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination
1    REJECT     all  --  0.0.0.0/0            0.0.0.0/0
d

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination

[root@localhost sysconfig]#
```

Fig 3: commands

Monitoring tools can compute the performance of a firewall. Rules are interrogated using the top to down approach. In above figure the attacker would target the secured area and send the DDoS traffic to destination machine.

5. PERFORMANCE RESULTS

In this section, the experimental and analytical results of the firewall performance in terms of various key measures which include firewall's CPU history and Network history were presented. In particular the results when sending a DDoS traffic targeting a different rules set. As mentioned in the above experimental setup (Fig 2) three scenarios targeting 1000 rules, 2000 rules and 3000 rules were tested, with time on the X-axis.

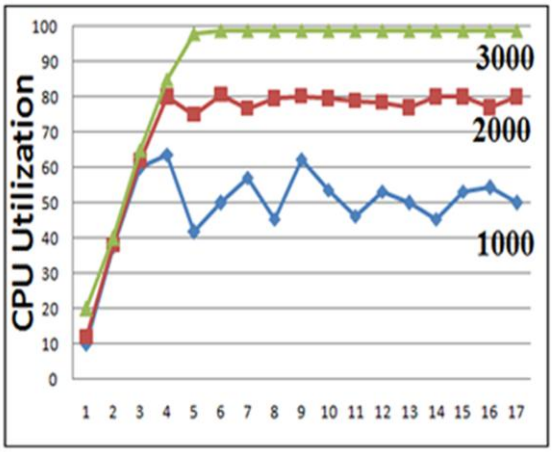


Fig 4: CPU Utilization

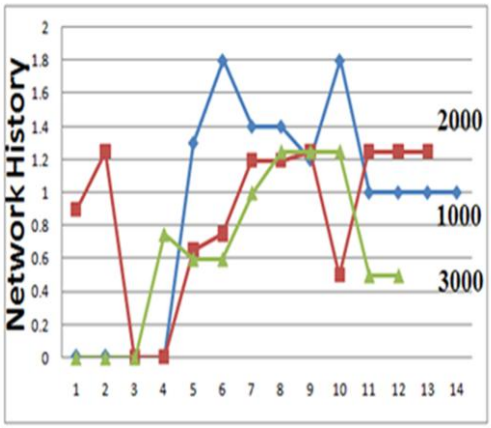


Fig 5: Network History

Figure 4 shows the CPU Utilization when a DDoS attacks targeting a set of 1000, 2000 and 3000 rules. The results were obtained and compared. When DDoS targeted 1000 rules the CPU utilization is between 40% to 60 %. The CPU utilization value become 70% to 80% when DDoS attacks targeting the 2000 rules. Coming to 3000 rules the value quickly reaches the 100%. The host is also idle or was not responding for few minutes.

Figure 5 shows the network history compare to 2000 and 3000 rules the network performance is better when DDoS attacks targeting the 1000 rules. The network history value reaches 1.8Mbps when targeting 1000 rules.

The firewall does not respond when DDoS attacks target bottom rules. To overcome this problem, change the position of the most triggered rules to top position in certain time intervals, using Adaptive Reorientation Method. After reorientation of the rule-set the performance results are shown below.

As shown in figure 6 and 7, firewall achieves the better performance. It can overcome the crashes from attacks. Using the priority values change the most triggered rules to top position. After reorientation the Cpu utilization maximum value is 40%. The network history maximum value is 2.5Mbps.

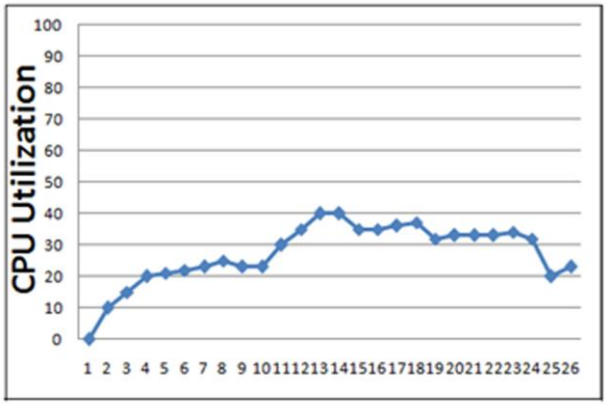


Fig 6: CPU Utilization after reorientation takes place

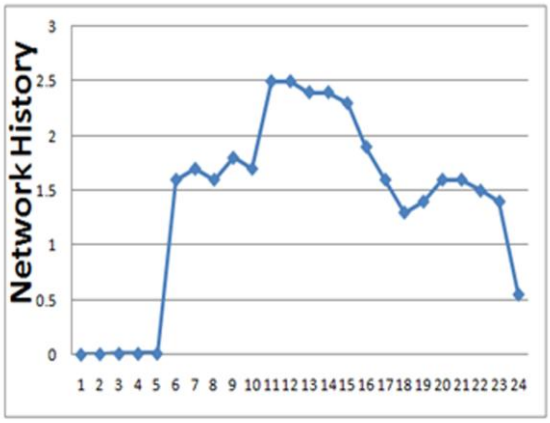


Fig 7: Network History after reorientation takes place

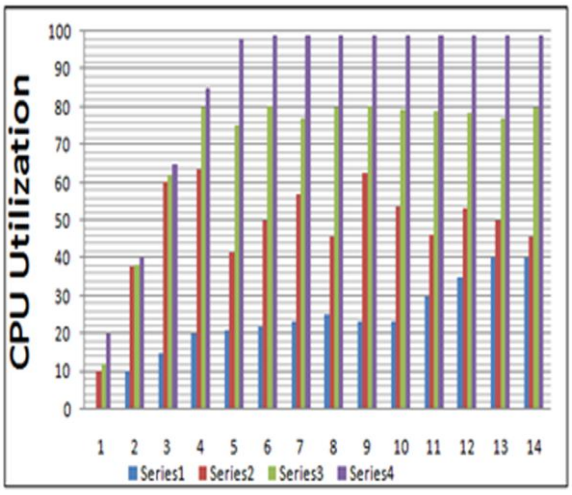


Fig 8: comparision between pre and post reorientation

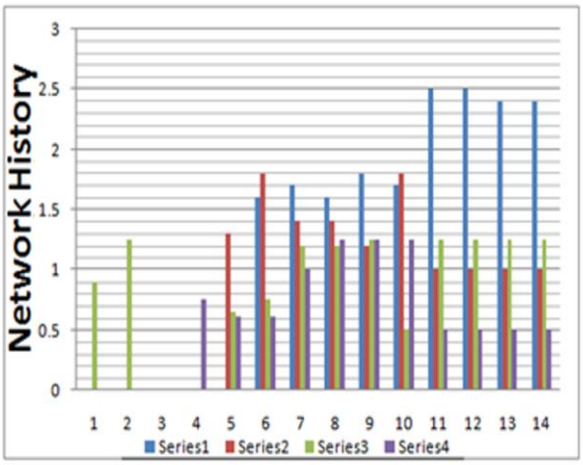


Fig 9: comparision between pre and post reorientation

Fig 8 and 9 illustrate the pre and post reorientation method applies on the rules-set. s1 is post reorientation. s2,s3,s4 represent the pre orientation states of the firewall with 1000 2000 and 3000 rules respectively. Finally achieve better performance compare to pre reorientation.

6. CONCLUSION

In any Organization, packet filtering controls are dependent on a number of services acquiesces. The high triggered rules are generally placed at the bottom position. Every time the packets interrogation value is high, using this Adaptive Reorientation Method, the firewall rules place the high triggered rules in top position. To achieve quality of service firewall successfully decreased the interrogation time increasing the internal network security thus creating an intelligent behavior for the firewall's functionality.

7. REFERENCES

[1] David W. Chapman Jr., Andy Fox, December 18, 2001. Cisco Secure PIX Firewalls, Cisco Press.

[2] Gregor N. Purdy., June 30, 2009**.** Linux iptables Pocket Reference (Pocket Reference (O'Reilly)) [Kindle Edition]

[3] Babak Farrokhi ., April 2008. Network Administration with FreeBSD 7, packt publishing.

[4] Americo J. Melara, June 2002. Performance analysis of the Linux firewall in a host," Master's thesis. California Polyphonic State University.

[5] Noe Nevarez and Huy Duc Vo. 2007. Linux Firewall Performance Analysis"University of Houstn

[6] Qing-Xiu Wu. 2012. The Research and Application of Firewall based on Netfilter. Science Direct, Physics Procedia 25.

[7] H. Hemed, A. El-Atawy, and E. Al-Shaer. Adaptive statistical optimization techniques for firewall packet filtering. School of Computer Science, DePaul University, Chicago, USA.

[8] K. Salah. proceedings 2010. Queuing Analysis of Network Firewalls. IEEE Globecom.

[9] Huirong Fu, Ming Zhang. proceedins 2006. Online adaptive firewall allocation in internet data canter. Science direct, computer communications.

[10] R.Hunt, T.verwoerd. 2003 Reactive Firewalls –a technique. Science Direct, Computer Communication 26.

[11] Noel De Palma, Daniel Hagimont, Fabienne Boyer, and Laurent Broto. proceedings 2012. Self-Protection in a Clustered Distributed System. IEEE Parallel and Distributed Systems. Page 330-336

[12] Liang Zhi-honga, Luo Jian-zhenb, Liang Zhi-qianga a*. 2011. System Recovery Testing of Hardware Firewall" Science Direct, Procedia Engineering 25.

[13] Sean Michael Kerner,.April 26, 2013. DDOS attack Report.

[14] Khaled Salah, Khalid Elbadawi, and Raouf Boutaba. 2012. Performance Modeling and Analysis of Network Firewalls. IEEE Transaction on Network and Service Management.

[15] Chris Roeckl. 2004. Stateful Inspection Firewalls. Juniper Networks, Inc.