I. J. Computer Network and Information Security, **2013, 12, 60-67**
Published Online October 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2013.12.08

Performance Evaluation and Comparison of Network Firewalls under DDoS Attack

Chirag Sheth
Tata Consultancy Services Limited, Garima Park, Gandhinagar – 382009, India
chirag.sheth@tcs.com

Rajesh Thakker
Electronics & Commu Dept, Govt. Engg. College, Bhavnagar – 364002, India
rathakker2008@gmail.com

Abstract**—** Network firewalls act as the first line of defense against unwanted and malicious traffic and also represent critical point of failure during DDoS attack. Predicting the overall firewall performance is crucial to network security administrators and designers in assessing the strength and effectiveness of network firewalls against DDoS attacks. In this paper, authors have made a humble attempt to study and compare DDoS performance of various types of firewalls in operation as on today. Analysis and detailed comparison is performed on open source packet filter (PF) firewall, Checkpoint SPLAT and Cisco ASA in a testing environment with laboratory generated DDoS traffic. It is attempted to identify various firewall DDoS performance parameters which can be considered during DDoS attack. Further, experiments are carried out to study effect of varying TCP Opening Timers on performance of stateful inspection firewall during Sync Flood attack. Also, in order to improve performance, intelligence is applied in PF firewall rulebase to mitigate DDoS.

Index Terms**—** DDoS Attack, Network Security, Distributed Network Firewall, Checkpoint NGX, Cisco ASA, OpenBSD PF

I. Introduction

Most of the organizations want to be always connected and remain online 24 x 7. However, not enough focus is being put on analyzing network performance to defend or devising security solutions that will help to protect against attackers targeting to exhaust their network resources for personal or criminal gains. Distributed denial-of-service (DDoS) attacks are a major threat to the Internet. Good amount of research is being undertaken to detect, prevent, delay and trace back DDoS attacks. Most of researchers and network administrators are doing post attack forensics which comes after the attack has taken place. However, no system is currently in place which can totally mitigate or tolerate DDoS attacks. The frequency, size, duration and volume of Distributed Denial Services (DDoS) attacks have significantly increased. According to Quarterly Global DDoS attack report published by Prolexic during Q1 2013, average DDoS attack bandwidth has increased by 718% in Q1 2013 with the peak attack breaking 300 Gbps barrier for the first time [1]. Firewall deployments are critical considering magnitude and volume of DDoS attacks. Hence the firewall needs to show robust performance along with application intelligence in order to withstand against DDoS attack. Historically, DDoS attacks were carried out for extortion, but now they are even used for terrorist activities and by unscrupulous companies to take out their competitors' web presence. Most of the companies are paying large amount of money on annual basis to buy specialized DDoS mitigation and protection gear to protect their web applications during DDoS attack, which they may never use it. Also, most network providers and managed services hosting providers have no real operational solution to stop DDoS attacks.

Today, a significant portion of Internet traffic comprises of senseless data and illegitimate packets which consume lots of bandwidth and network resources. According to World Network Infrastructure Security report released by Arbor Networks, around 3% of total internet traffic is DDoS traffic. Stateful firewalls, IPS and load-balancer devices continue to fall short on DDoS protection capabilities [2]. More and more companies have been deploying intrusion detection systems (IDS) in their network. IDS can be effective addition to firewall considering better logging the contents. However, major issue with IDS is that they are not much effective for signature-based detection. Also, they are not intelligent enough and hence, they create huge number of false alerts.

A denial-of-service (DoS) attack or distributed denial-of-service (DDoS) attack is an attempt to make a computer resource or web services unavailable to its legitimate users. Although generation, targets and motives of a DDoS attack vary, it is generally created by concerted efforts of a person or group to prevent an internet site or service from functioning efficiently. It is an attempt to consume finite resources, exploit weaknesses in software design or implementation, or exploit lack of infrastructure capacity. The collateral damage caused by an attack can be very huge. DDoS attacks can also lead to problems in the network segments around the actual computer being attacked. For example, the bandwidth of a router between the Internet and a LAN may be consumed by an attack, compromising not only the intended computer, but also the entire network. If the attack conducted on a sufficiently large scale, entire geographical regions of Internet connectivity can be compromised without the attacker's knowledge or intent due to incorrectly configured network infrastructure equipment.

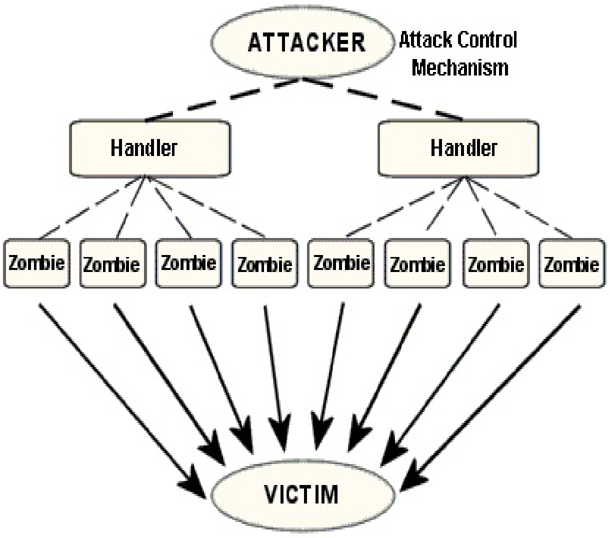

Figure 1: Typical DDoS attack

In DoS attack, one computer and one internet connection is used to flood a server with packets (TCP / UDP). The point of such a denial of service attack is to overload the targeted server’s bandwidth and other resources. This will make the server inaccessible to others, thereby blocking the website hosted there. In DDoS attack, instead of one computer and one internet connection it utilizes many computers and many connections. Attacker runs a malicious process in compromised systems which are called Zombies. They are under his control and generate enormous number of requests, which in turn can easily exhaust the computing resources of a victim web server within a short period of time. The attack is "distributed" because the attacker is using multiple computers, to launch the denial-of-service attack. Fig. 1 describes basic architecture of DDoS attack.

The paper is organized as follows: In Section II, various reported literature already undertaken in this area have been highlighted. In Section III, authors have attempted to briefly compare various DDoS attack types. Section IV provides detail of performance testing setup, tools used and experiment carried out to compare performance of some of the major firewalls in operation today. Section V deals with firewall performance improvement by tweaking TCP timers as well as by controlling firewall state table entries. Finally, conclusions are drawn in Section VI.

II. Related works

In their previous work, authors have carried out performance evaluation and comparative analysis of most widely used network firewalls by identifying various key performance indicators [3]. In this paper, further extension is carried out by analyzing performance during DDoS attack. Majority of the work carried out in literature is focused on detection of DDoS attack and identification of source of DDoS attack.

Various reported defense and response mechanisms have been suggested in literature about DDoS attacks. Hussain et al. made a notable contribution by presenting framework for classification of DDoS attacks into single-source or multi-source [4]. Mirkovic and Reiher presented a comprehensive taxonomy of DDoS attacks and defense mechanisms [5]. Many DDoS detection approaches, such as “IP traceback” [6], “traffic pattern and statistic” [7], “pushback” [8, 9], “packet filtering” [10] and “wavelet analysis” [11] have been proposed in literature. All of them try to find the identities of real attack sources and defend against attacks. It is evident from literature study that if we expect to prevent DDoS attacks significantly we need to first handle two critical issues – (a) Accurately identifying the machines participating in forwarding malicious flows and (b) Forcefully cutting off the malicious flows at those machines.

Significant work has also been done by Bi and Zhengstudy [12] and by Kumar et al. [14] on developing strategy against DDoS attack. Mirkovic et al. [15] also set forth benchmarks for DDoS defense evaluation. Several adaptive approaches to defend DDoS attack are also suggested in many of the literature. Salah and Elbadawi presented performance modeling of firewalls [17]. Singh and Verma came up with dynamic bandwidth assignment during DDoS [18]. Apart from these, there has been significant work done in the direction of DDoS mitigation. However, not much importance has been laid on analyzing network firewall performance during DDoS attack.

Authors have observed that during majority of DDoS attacks, firewalls are first point of failure. Hence, focus is made on identifying DDoS performance parameters of firewalls and attempt is made to improve the same.

III. DDoS Attack Types

There are basically two types of DDoS attacks.

(a) Bandwidth Depletion Attacks

It is designed to flood the victim network with unwanted traffic that prevents legitimate traffic from reaching primary victim system. There are two main types of bandwidth depletion attacks. First one is flood attack which involves the secondary victim systems for sending large volumes of traffic to a victim system. Eventually, it will congest the victim system's bandwidth. Second one is amplification attack which involves either the attacker's or the secondary victim system to send messages to a broadcast IP address. Eventually, this will cause all systems in the subnet reached by the broadcast address to send a message to the victim system.

(b) Resource Depletion Attacks

In DDoS resource depletion attacks the attacker sends a malformed packet that ties up the network resources or exhausts the system resources, so that no resources are left for legitimate users.

Listed in the TABLE I are some of the DDOS attack types along with brief description of them.

TABLE I. MAJOR DDOS ATTACK TYPES

DDoS Attack	Details
Generic flood attacks	Flood of traffic for one or more protocols or ports. UDP flood and Sync Flood are common types. It can be spoofed or non-spoofed.
Fragmentation attacks	A flood of TCP or UDP fragments are sent to overwhelm the victim's ability to reassemble the streams and severely reducing performance. It may also be a result of misconfiguration.
Connection attacks	Connection attacks maintain a large number of half-open or fully open idle TCP connections. Resource exhaustion in the TCP stack or application connection tables prevents the victim host from allowing new TCP connections to be opened to the victim.
Application-level floods attacks	Application attacks are designed to overwhelm components of specific applications. Buffer Overflow can consume all available memory or CPU time.
Vulnerability exploit attacks	Vulnerability exploit attacks are designed to exploit a software flaw in the victim's operating system or application.

IV. DDOS PERFORMANCE COMPARISON OF VARIOUS FIREWALLS

In recent times, there is strong demand to analyse the performance of network firewalls when subjected to DDoS attacks. If network firewalls are poorly designed to withstand DDoS attacks, the overall security of the protected network will be on high risk. Specifically, there is an increasing demand for analysing, modelling and simulating performance of network firewalls to predict how effective and efficient network firewall is under DDoS attacks. This will help Firewall designers and system administrators to identify bottlenecks and key parameters that impact its performance, and then perform the necessary tuning for optimal performance. Performance analysis can provide quick answers to numerous design and operational questions. This will help firewall designers to carry out a first cut design to reduce the set of design alternatives and then use simulations or experiments to assess performance of few good designs before building and deploying the system into their own network environment.

In spite of firewall representing one of the critical point of failure at the time of DDoS attack, no standard method of firewall performance evaluation during DDoS is prevalent in market as per author's knowledge. The primary reason for the same is that firewall implementations vary widely making it difficult to carry out direct performance comparisons. As more and more organizations deploy firewalls on their networks, question arises whether the products they buy will stand up and sustain to relatively heavy loads. All the three firewalls used in this setup are Stateful, i.e., they keep track of the state network of connections (such as CP streams) travelling across it. By keeping track of the connection state, stateful firewalls provide added efficiency in terms of packet inspection. This is because for existing connections the firewall need only check the state table, instead of checking the packet against the firewall's rule set, which can be extensive. In order to prevent the state table from filling up, sessions will time out if no traffic has passed for a certain period. These stale connections are removed from the state table.

Although the firewalls are stateful, during DDoS attack, each set of packets traversing a stateful firewall consumes state-table resources within those firewalls, creating a DDoS chokepoint. As firewalls have limited amount of state-table resources it is quite easy for attackers to programmatically generate sufficient well-formed traffic which will satisfy and pass the firewall policy rules. Eventually, this will choke up bandwidth for legitimate traffic from real users which will lead to denial of services of the servers and applications behind the firewall. Additionally, in most cases, sufficient firewall state-table exhaustion due to attack traffic will cause stateful firewalls to essentially fall over and fail to forward traffic. Hence, stateful firewalls almost invariably surrender to DDoS attacks even far more rapidly than the servers themselves would without the firewalls there at all.

A. Laboratory DDoS attack Generation – Open Source Tools Comparision

One of the major challenge to study firewall performance was to generate and replicate DDoS in laboratory environment. Study and implementation of many of the open source tools which generates traffic are done in order to generate as distributed traffic as possible. Below is the comparison of some of the Open Source Tools used along with their limitations.

(a) Apache JMeter

JMeter is an Apache Jakarta project [19] that can be used as a load testing tool for analysing and measuring the performance of a variety of services, with a focus on web applications. Its limitation was its inability to scale well as it can only send a maximum of 2500 requests per second using single system used in setup. Moreover, it is not able to tune the request rate (rps) and consequently, its variance is more during the test.

(b) FWPTT - Fast Web Performance Test Tool

FWPTT (Fast web performance test tool) [20] is an open source web application testing tool written in C#.net for load testing web applications Its limitation is whatever may be the input combination to this tool, this tool is unable to send more than 500 requests per second using single system used in setup, and hence it is not scalable. Moreover, it is not having an option to tune the number of request per second nor it is having graphical viewer.

(c) JCrawler – Stress Testing Tool

JCrawler [21] is an open-source Stress-Testing Tool written in Java for web-applications. The limitation of the same is that it is like a web portal system and not suitable to use as load-testing tool. It is not able to scale well since it is searching for the URLs to redirect in each web page.

(d) Curl-Loader

Curl-loader 22] is an open-source tool written in C-language. It is capable of simulating application load and application behaviour of thousands and tens of thousands of HTTP/HTTPS and FTP/FTPS clients, each with its own source IP-address. It runs under Linux platform. We observed only one major limitation that is not scriptable and hence it cannot be used for dynamic requests. However, inspite of this limitation, authors found Curl-Loader better as compared with other tools for setup described in the next section in Fig. 2. Hence, it was used for laboratory DDoS Traffic generation.

B. Performance Testing Setup

In order to characterize performance of firewall, the testing environment setup shown in Fig. 2 is used to compare performance of three most operational firewalls in market.

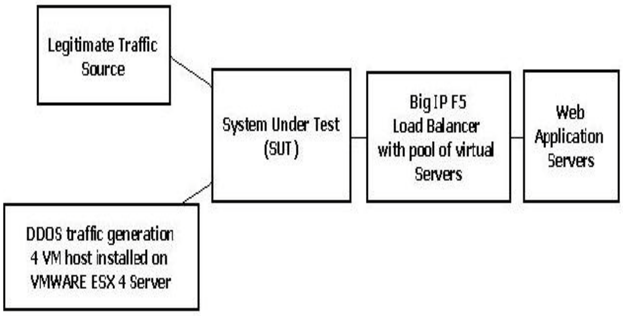
Legitimate Traffic Source
System Under Test (SUT)
Big IP F5 Load Balancer with pool of virtual Servers
Web Application Servers
DDOS traffic generation 4 VM host installed on VMWARE ESX 4 Server

Figure 2: Setup diagram for performance testing

Test traffic is generated using Open-Source tool Curl-Loader. Virtual machine setup is used in order to generate traffic as distributed and as higher in magnitude as possible. VMWARE ESX 4 server is deployed and 4 virtual machines hosts are installed on the same. The traffic is targeted towards a web application hosted on web application servers at the other side. The firewall policy is set to allow all the requests on port http and https towards the targeted IP where web services are hosted, hence firewall job is to establish state and forward the packet. Packets and states are observed on the firewall using various tools and CLI commands. The tool runs 2500-100000 and more simultaneously loading clients, all from a single curl-loader process. Big-IP F5 Load Balancer is also used which has virtual servers pool containing inside web servers. The traffic going to web servers is observed from Load Balancer. The Load Balancer is used so as to make the environment as close replica of live environment as possible.

The firewall configurations, operating system and hardware details of three firewall products under test are mentioned in Table II. The configurations used are similar to that used in [3] with some upgrades in OS. Cisco uses its own hardware. Checkpoint and PF are configured on HP Servers. Attempt is made to keep hardware as similar as possible for all three firewalls under test, in order to have conditions as close as possible to real world. Compatibility of hardware and network interfaces with firewall operating system is tested beforehand after referring to firewall product website [23-25].

TABLE II. FIREWALL CONFIGURATIONS

Firewall Configurations	Firewall Products		
	Cisco ASA	Checkpoint (CP) SPLAT	OpenBSD PF
Platform	Cisco ASA - 5580	HP DL 380	HP DL 380
Operating System	ASA V 8.2.2 ASDM 6.2.5	SPLAT 2.4 Checkpoint NGX R70	Open BSD 4.7
Product Architecture	Multi-processor, Multi-core		
Processing Cores	8		
Gigabit Ethernet Interfaces	0		
10 Gigabit Ethernet Interfaces	4		

C. Performance Testing Results

Some of the DDoS performance parameters are measured in Table III. These are explored in order to compare performance of three of the most widely used firewall products in market as on today.

TABLE III. DDOS PERFORMANCE TEST RESULTS

DDoS Performance Parameters	System Under Test – Firewall Products		
	Cisco ASA	CP SPLAT	OpenB SD PF
HTTP Throughput (Gbps)	10.6	5.6	4.5
Legitimate Traffic allowed till percentage of DDoS traffic	80%	82%	76%
Firewall CPU Utilization at 50% DDoS	40%	45%	43%
Firewall CPU Utilization at 75% DDoS	60%	80%	65%

Time for complete failure (unreachable) at full DDoS	12 min	15 min	9 min
Capacity limits (Percentage of other traffic blocked except TCP)	100%	100%	100%

1) *HTTP Throughput:* It is the maximum offered HTTP load, expressed in either bits per second or packets per second, at which no packet loss is detected. The goal of this test is to characterize the performance of the SUT when deployed to protect a high performance web-based application. Cisco outperformed other firewalls in real-world HTTP performance tests.

2) Legitimate Traffic allowed till percentage of DDoS traffic: Since we have laboratory generated DDoS traffic, we know about the IP Range used as legitimate traffic and IP Range used for DDoS. Checkpoint showed initial resistance and allowance of legitimate traffic at percentage of full DDOS more than other two firewalls.

3) *Firewall CPU Utilization at percentage of DDoS*: Firewall CPU Utilization is one of the important parameter under DDoS Attack. We have used 50% and 75% of DDoS traffic as reference point for checking CPU Utilization. The higher the CPU Utilization on firewall, it will take more time to process and forward packets and eventually more time to accept new requests. Cisco showed high processing power and eventually lower CPU utilization compared to its peers.

4) *Time for complete failure (unreachable) at full DDoS:* None of the firewall proved being capable of withstanding DDoS for longer time. At full DDoS, eventually firewall became completely utilized and lost connectivity. The only option left is to restart the system to make it flush its state table entries and eventually start accepting again. Checkpoint does outperform its peers to withstand DDoS for more time before crashing.

5) Capacity limits (% of other traffic blocked except TCP): This parameter determines capability of firewall to prioritize traffic based on application intelligence. The TCP Traffic (http/https) should be given priority to ping/UDP traffic and this will help to prioritize legitimate traffic. As expected, all three firewalls showed application intelligence and gave priority to TCP traffic than ping and UDP during DDoS.

D. Observations

To the best of author's knowledge, none of the firewalls used in our setup mentions anything about discussed DDoS performance parameters. The DDoS performance test results obtained are specific to environment used in the setup. The best course of action to test firewall performance is to replicate network conditions as close as possible to the conditions that actual firewall is supposed to experience. Hence, authors have tried to keep firewall hardware configurations as similar as possible. There could be variation in firewall performance on different make and model of hardware.

V. Firewall Performance Improvements

After analysing firewall performance during DDoS attack, we suggest performance improvements by varying some of the parameters and by controlling state table entries.

A. Performance Improvement by tweaking TCP Opening Timer during SYN Flood Attack

Most of the times, intruder can perform DDoS attack either as brute force or as logical attack. In brute force DDoS attack, legitimate looking, but actually error data packets are sent continuously targeting victim's services. It will in turn reduce legitimate user bandwidth and resources and prevent access to the desirable service. Logical attack exploits a specific feature or implementation bug of some protocol or application installed at the target machine in order to consume an excess amount of its resources.

All TCP communication is connection oriented. A TCP session must be established before hosts in the connection exchange data. The three-way handshake is shown in Fig. 3. At first, the initial request is acknowledged, then the data is sent and after that, at last the data is acknowledged. Today, majority of DDoS attacks are performed using TCP and large portion of them are targeted to flooding attacks.

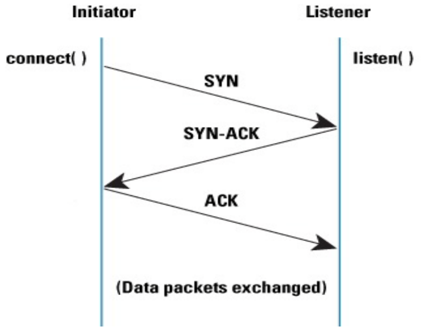

Figure 3: Normal TCP Handshake

Any system providing TCP-based network services is potentially subject to this attack. In normal case, TCP 3-way handshaking is performed. The attacker sends a flood of TCP/SYN packets, most of the times with a fake sender address. Each of these packets is handled like a connection request, causing the server to issue a half-open connection by sending back a TCP/SYN-ACK packet and waiting for an TCP/ACK packet in response from the sender address. However, because the sender address is fake, the response never comes. These half-open connections consume resources on the server and as the number increases, resources utilized increases to a level that will limit the number of connections the server is able to make. This will in turn reduce the server's ability to respond legitimate requests until the attack ends.

The result would be system crash and turning non responsive.

As shown in Fig. 4, an attacker initiates a SYN flooding attack by sending many connection requests with spoofed source addresses to the listener machine. That causes listener to allocate resources, and once the limit of half-open connections is reached, it refuses all successive connection establishment attempts.

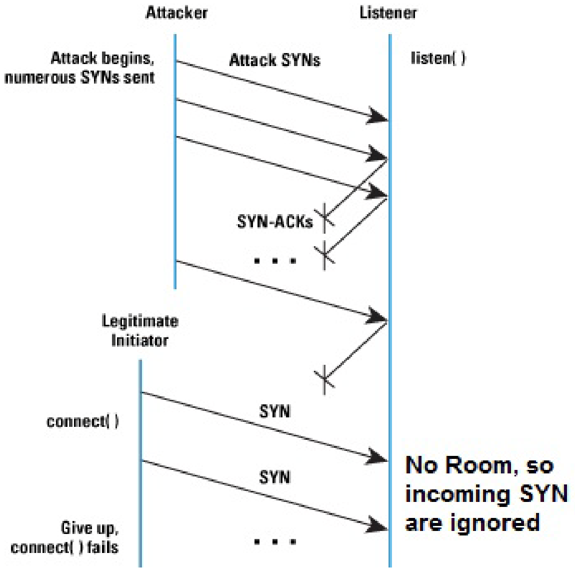
Attacker
Listener
Attack begins, numerous SYNs sent
Attack SYNs
listen( )
SYN-ACKs
. . .
Legitimate Initiator
connect( )
SYN
SYN
No Room, so incoming SYN are ignored
Give up, connect( ) fails
. . .

Figure 4: SYN Flood Attack

The basis of the SYN flooding attack lies in the design of the 3-way handshake that begins a TCP connection. In this handshake, the third packet verifies the initiator's ability to receive packets at the IP address it used as the source in its initial request, or its return reachability. Experiments are carried out by tweaking TCP.opening timer value from default 30 sec to 1 sec. Testing is carried out in same setup as used earlier (Fig 2). OpenBSD PF firewall is chosen considering it being open source and flexibility to change parameters from source code.

Different intensity of laboratory generated traffic is used to test performance. CPU Utilization is taken as key performance indicator along with firewall state table with half closed states. Below are the results obtained which shows consistent improvement in CPU utilization of firewall hardware when we set the TCP.opening value as 1 second during SYN Flood attack in which only SYN packets are send for denial of service. Changing TCP.opening value to 1 second might pose disadvantage that firewall will not keep states more than 1 sec for established connection. However during DDoS, lowering this value proves to be helpful in improving firewall performance.

TABLE IV. TEST RESULTS BY TWEAKING TCP TIMER IN OPENBSD PF FIREWALL

Laboratory Generated Traffic HTTP conn./sec	Tcp.opening = 30 s		Tcp.opening = 1 s	
	No. of half closed States	CPU Utilization	No. of half closed States	CPU Utilization
5 K	145 K	25%	4.7 K	8%
50 K	1.3 M	59%	48 K	16%
100 K	2.1 M	89%	93 K	19%

Results indicate that lowering the value of TCP timers for stateful firewalls helps improving firewall performance during DDoS Attack. In any setup, optimal value of timer should be chosen after taking into consideration web application and network environment.

B. Performance Improvement by DDoS Identification and Mitigation by controlling states in Firewall State Table

Most of the firewalls used today are stateful inspection firewalls. They perform the same function as packet filter firewalls, but with the ability to keep track of the state of connections in addition to the packet filtering abilities. By dynamically keeping track of whether a session is being initiated, currently transmitting data (in either direction), or being closed, the firewall can apply stronger security to the transmission of data. A stateful inspection firewall is capable of understanding the opening, communication, and closing of sessions. Stateful inspection firewalls usually have a fail-close default configuration, meaning that they will not allow a packet to pass if they do not know how to handle the packet. Overall, stateful inspection firewalls give high performance and provide more security features than packet filtering. Such features can provide extra control of common and popular services. Stateful inspection firewalls support most (if not all) services transparently, just like packet filters, and there is no need to modify client configurations or add any extra software for them to work. However, during DDoS attack, keeping state table entries results in exhaustion of firewall state table and not able to accept any more states.

Intelligence can be induced in firewall to identify hosts which are source of DDoS. Keeping the same setup for OpenBSD PF firewall as used throughout the paper, authors have introduced below in the rules in OpenBSD PF Firewall –

keep state (\ max 2000000, \ max-src-conn 10000, max-src-conn-rate 1000/10, overload <DDOSTable>)

The above will only keep states in the firewall state table which will satisfy specified conditions. Maximum state in the state table is set to 2 million. Once it reaches, it will start discarding older state table entries. Also, particular host can have maximum of 10000 concurrent connections or state table entries. This will ensure protection against DoS attacks. Apart from that source connection rate is kept as 1000 connections per 10 seconds. If any host is making requests faster, then it will be discarded and state table will be cleaned up. Also all such hosts who will meet this criterion will be further blocked and entries for that host will be made in DDoSTable. The maximum limits set are optimal for author’s laboratory setup and has proved to be effective in mitigating laboratory generated DDoS in setup used.

The parameters can be different for different setup and live environment.

VI. Conclusions

Security flaws in most firewalls do not appear until the network encounters a heavy load. Attacks can hide more easily within large amounts of traffic, potentially causing problems right when network downtime is most harmful. Firewalls often exhibit different behaviours as they encounter increasing loads. In the paper, we have attempted to evaluate performance of major operational firewalls in the market today under DDoS attack. To the best of author's knowledge, most currently undertaken and reported research work on DDoS focus on other parameters and firewall performance is not given due importance. We have attempted to compare performance of various firewalls based on practical implementation. Test results will help in identifying pre-deployment capacity planning and testing network performance to ensure that the increased security does not degrade performance beyond the levels acceptable for the business. The suite of tests performed will help determine the performance and behaviour of the firewall under various levels of DDoS attacks. Generation of DDoS attack was a challenge and authors compared various open source traffic generation tools and found Curl Loader to be suitable for the setup.

The performance testing results indicated that no firewall proved to be capable of withstanding DDoS for longer time. Checkpoint showed initial resistance and allowance of legitimate traffic at percentage DDoS more than Cisco and PF. However, CPU utilization of Checkpoint was higher as compared with Cisco ASA and PF firewalls. Time before all three firewall becomes unreachable is very less even to react to DDoS. It appears that although the firewalls are stateful, during DDoS Attack, each set of packets traversing a stateful firewall consumes state-table resources within those firewalls, creating a DDoS chokepoint. Hence, at high intensity DDoS state tables resources of all three firewalls get consumed making them unreachable in short time. Towards the end, authors also performed experiments to show improvement in firewall performance by tweaking TCP Opening timer during SYN Flood attack. Various parameters were changed in order to control state table entries to mitigate DDoS attack and attempt was made to introduce intelligence in firewall.

Authors also recommend that network environment and application should be thoroughly studied before making any changes in firewall state table parameters as changes may adversely affect overall traffic flow.

References

[1] Quarterly Global DDoS Attack Report released by Prolexic, Apr 2013, http://www.prolexic.com/knowledge-center-ddos-attack-report-2013-q1/pr.html.

[2] World Network Infrastructure Security Report Volume VI, released by Arbor Networks, Feb 2011, http://www.arbornetworks.com/report.

[3] C. Sheth and R. Thakker, "Performance Evaluation and Comparative Analysis of Network Firewalls," Proc. *IEEE Int'l Conf. on Devices and Communications (ICDeCom), pp. – 1-5,* Feb 2011.

[4] A. Hussain, J. Heidemann, and C.Papadopoulos, "A framework for classifying denial of service attacks," in *Proceedings of the ACM Conference on Internet Measurement (SIGCOMM '03), pp. 99-110,* Karlsruhe, Germany, August 2003.

[5] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *Computer Communication Review, vol. 34, no. 2. pp. 39–53*, 2004.

[6] H. Aljifri, "IP traceback: a new denial-of-service deterrent?" *IEEE Security and Privacy, vol. 1, no. 3, pp. 24–31*, 2003.

[7] M. Li, M. Li, and X. Jiang, "DDoS attacks detection model and its application," *WSEAS Transactions on Computers, vol. 7, no. 8, pp. 1159–1168*, 2008.

[8] M. Cai, Y. Chen, Y. K. Kwok, and K. Hwang, "A scalable set-union counting approach to pushingback DDoS attacks," *Tech. Rep. TR-2004-21, USC GridSec*, Oct 2004.

[9] C. C. Zou, N. Duffield, D. Towsley and W. Gong, "Adaptive defense against various network attacks," *US patent no. US7,587,761 b2*, September 2009.

[10] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service, *IEEE Transactions on Dependable and Secure Computin, Volume 3, pp – 141-155*, 2006.

[11] A. Dainotti, A. Pescap´e, and G. Ventre, "Wavelet-based detection of DoS attacks," *in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '06), pp. 1–6, San Francisco, Calif, USA*, November 2006.

[12] X Bi and Q ZhengStudy, "On network safety strategy against DDoS attack", *Proc. IEEE Int'l Conf on Advanced Management Science (ICAMS), pp. 623 – 627,* Aug 2010.

[13] D. Newman, Benchmarking Terminology for Firewall Performance, *IETF RFC2647*, August 1999.

[14] Kumar, R.; Karanam, R.; Bobba, R.C.; Raghunath, S., "DDoS Defense Mechanism," Proc. *IEEE Int'l Conf. on Future Networks, 2009 pp. 254 - 257,* Mar. 2009.

[15] Mirkovic, J.; Arikan, E.; Songjie Wei; Fahmy, S.; Thomas, R.; Reiher, P., "Benchmarks for DDoS Defense Evaluation," Proc. *IEEE Int'l Conf. Military Communications MILCOM 2006, pp. 1 - 10,* Oct. 2006.

[16] Ming Li; Jun Li; Wei Zhao, "Simulation Study of Flood Attacking of DDoS", Proc. *IEEE Int'l Conf Internet Computing in Science and Engineering, ICICSE '08, pp. 286 - 293,* Jan. 2008.

[17] K Salah and K Elbadawi, “Performance Modeling and Analysis of Network Firewalls,” *Proc. IEEE Transactions on Network and Service Management, Vol. 9, No. 1*, March 2012.

[18] R.Singh and A.Verma, “A Dynamic Bandwidth Assignment Approach Under DDoS Flood Attack, ” *Journal of Advances in Information Technology, Vol.3, No.2*, May 2012.

[19] Apache Jakarta Project, *"Apache JMeter"*, http://jakarta.apache.org/jmeter.

[20] Bogdan Damian, *“Fast Web Performance Test Tool - fwptt"*, http://fwptt.sourceforge.net.

[21] Idumali, Under the CPL, *"JCrawler – A Perfect Load Testing Toll"*, http://jcrawler.sourceforge.net.

[22] Robert Iakobashvili, Michael Moser, under the licensed GPLv2, *"curl-loader"*, http://curl-loader.sourceforge.net.

[23] Cisco Systems Inc., Cisco ASA 5500 Series Security Appliances http://www.cisco.com/en/US/products/ps6120/prod_models_comparison.html.

[24] Check Point Technologies Ltd., NGX R70 Release Notes http://dl3.checkpoint.com/paid/41/CheckPoint_R70_ReleaseNotes.pdf?HashKey=1315749093_c1565e4d15313c53c997f11107cf6ae3&xtn=.pdf.

[25] PF: The OpenBSD Packet Filter http://www.openbsd.org/faq/pf/.

[26] RFC 4732, M.J. Handley, The IETF Trust (2006), “Internet Denial-of-Service Considerations”, http://tools.ietf.org/html/rfc4732 .

[27] RFC 4987, W. Eddy, The IETF Trust (2007),"TCP SYN Flooding Attacks and Common Mitigations", http://tools.ietf.org/html/rfc4987.

[28] RFC 793, B.Postel, The IETF Trust(1981), “Transmission Control Protocol”, http://tools.ietf.org/html/rfc793 .

Chirag Sheth, born in 1980, is pursuing his PhD in Kadi Sarva Vishwavidyalaya, Gandhinagar, Gujarat from India. He has completed his Masters of Engineering (ME) degree in Electronics and Communication from BIT, Mesra. He is currently working with Tata Consultancy Services Limited and has more than 10 years of experience in the domain of Network Security. His main research interests include Network Firewalls, Packet Filtering, OpenSource Technologies etc.

Rajesh A Thakker received B.E. degree in Electronics & Communication Engineering from Gujarat University in 1993, M.Tech & PhD both in Electrical Engineering from IIT Bombay in the year 2002 and 2009 respectively. His major area of research includes applications of evolutionary algorithms in the field of VLSI.