INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATION ENGINEERING & TECHNOLOGY (IJECET)

ISSN 0976 – 6464(Print)
ISSN 0976 – 6472(Online)
Volume 5, Issue 11, November (2014), pp. 37-46
© IAEME: http://www.iaeme.com/IJECET.asp
Journal Impact Factor (2014): 7.2836 (Calculated by GISI)
www.jifactor.com

IJECET

©IAEME

PROTECTION OF SERVER FROM SYN FLOOD ATTACK

Shaila R Ghanti, G.M. Naik

Department of Electronics, Goa University, Goa, India

ABSTRACT

Due to advances in Internet Technology applications, the clients at remote places require constant services from a server. Internet services can be denied by malicious attacks on the server. One such attack is SYN flood attack which is a type of DDoS attack. This manuscript demonstrates the protection of server against the SYN flood attack. The algorithm uses a continuous self detecting method for identifying and updating genuine client information in the presence of spoofed packet and thereby protecting the server from SYN flood attack. During this process the algorithm builds the repository of genuine client information. This repository of genuine client can be used by other security systems like IDS, Packet filtering etc. for protecting the server. The performance of SYN flood attack protector can be further improved by implementing this algorithm in hardware such as FPGA.

Keywords: DDoS, Server Protection, SYN Flood Attack and Prevention.

1. INTRODUCTION

Due to advances in Internet Technology applications, the clients at remote places require constant services from a server. Internet services can be denied by malicious attacks on the server. Distributed Denial of Service (DDoS) is one of the most popular types of attack as reported by Worldwide Infrastructure Security Report [1]. DDoS attack blocks legitimate clients from accessing the server, and is generated from many compromised machines acting as zombies. Such attacks incur huge monetary loss to the respective companies. Vast number of recent DDoS attacks (e.g. attack was launched in support of Wikie leaks and its founder and it lasted for about 16 hours.) are reported [2] [6].

SYN flood attack is one of the most common types of DDoS attack. This attack deals with exploiting the standard TCP 3 way handshake protocol required for setting up of TCP connection between the client and the server. In a normal TCP 3 way handshake protocol (as shown in fig. 1), the client sends the SYN packet to the server. The server replies with SYN-ACK packet to the client by reserving the memory of server for maintaining the half open connection. In reply to this, the client sends back the ACK. Only when the server gets the ACK packet, the TCP connection is set up between the client and the server.

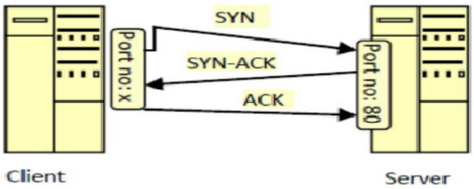

Figure 1: Normal TCP 3 way handshake

If the attacker sends the SYN packet with spoofed source IP address, the server sets the half open connection and sends SYN-ACK to the spoofed IP address, which may not exist. Now the server waits for corresponding ACK, which will not be received as the IP address is spoofed. The connection remains in half open state for a period of up to the TCP connection establishment timeout, which is typically 45 seconds [3].

During SYN flood attack an exceptionally large number of spoofed SYN packets are received by the server, the server replies with SYN-ACK and all the server resources are reserved for maintaining the half open connections as the respective ACKs will not be received from client. At such times, if there is a genuine request to connect to the server, the request is dropped as the resources of server are exhausted [4] [5].

Hence to provide service to all genuine clients, every server needs to be protected from SYN flood attack. We propose an improved SYN flood attack protector which protects the server during SYN flood attack.

2. BACKGROUND STUDY

Presently most of the servers are protected from SYN flood attack by implementing victim end protection methods. Some of the popular victim end protection methods are as below.

SYN Caches*:* In this case the amount of state allocated initially for a TCB generated by a received SYN is reduced, and full state is not initialised [7]. In a host that uses a SYN cache, a hash table with a limited amount of space in each hash bucket is used to store a subset of the data that would normally go into an allocated TCB. If an ACK is received, then it can be moved into a full TCB; otherwise the oldest bucket at a particular hash value can be reaped when needed.

SYN defender*:* In this case the SYN request from the client is received by the firewall and then it sends the SYN-ACK packet to the client. After the firewall receives the ACK, the request is then sent to the server. In this case the server does not maintain the half-open states and so does not deplete its resources [8]. SYN defender needs to repeat the same process for every incoming SYN packet irrespective of attack is generated or not.

Synkill: The source IP addresses are classified in a database as good or bad based on observed network traffic and administratively supplied input [9]. RST packet is generated in response to a request from Bad source address to terminate their request, while good ones are allowed to carry on with the handshaking. Once the packets are identified as they are in good state. It remains in good state till staleness period i.e. no TCP traffic was observed from that address for a period of time. In this method if within the staleness period any spoofed IP packets are received then it treats as, it is in good state.

Probing: L.Kavisankar and et. al. uses specific probing where the client is requested to change the window size/cause packet retransmission while sending the ACK in the three way handshake [10]. This is very useful to find the Spoofed IP packets /TCP SYN flood and preventing them. In this case probing is used for every request irrespective of attack.

SYNCookies: In this method when the server receives the SYN packet, it does not reserve any memory for maintaining half open connections instead it calculates cookie value based on the parameters of the SYN packet and sends the SYN-ACK [11]. Once the ACK is received the server checks the legitimacy of the ACK in accordance with the cookie value. If legitimate then the connection will be set up. Cookie calculations are done in [12]. Many researchers proposed some improved SYN cookie method [13-16].

Firewall/Router based SYN flood protection methods are employed on firewall or router generally as purely software but sometimes as hardware. The advantage of this type of SYN flood protection is, server processor is not at all used for processing the SYN flood protection methods, instead main processor of the server is used only for providing service to the clients.

FortiGate: It's a consolidated security platform that helps in blocking DOS attacks. In the event of TCP SYN Flood attack, FortiOS examine the SYN packet rate of new TCP connections, including retransmission, to one destination IP address. If this rate exceeds the configured threshold value (measured in packets per second), the FortiGate platform will block the traffic. [17].

Netscreen 5GT: Sanjeev Kumar and et.al. evaluates effectiveness of a security device Netscreen 5GT (Firewall) from Juniper networks under different attack loads. This device uses SYN proxy protection i.e. SYN defender is used [18].

In this method the server is prevented from SYN flood attack by maintaining good registry and bad registry, and updating these registries dynamically [19].

Daniel J Bernstein says that most popular methods that are used for SYN flood attack defence are SYN Cookie technology and state based monitoring of the source address technology [11]. According to CSI Computer Security Survey [20], firewall type of security technology was used by 94% of the organizations.

3. SYN FLOOD ATTACK PROTECTION SYSTEM

The "SYN flood attack protector system" is connected between the server and Internet as shown in fig. 2. It could be any type of server, but for our experimental purpose we have used Web Server.

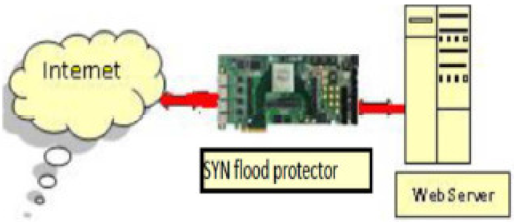

Figure 2: Server with SYN flood attack protector system

The essential task of the "SYN flood attack protector system" is to block an attack and allow only genuine client requests to the server. The proposed improved algorithm is based on [8], [9] and [19]. The algorithm uses a continuous self detecting method for identifying and updating genuine client information in the presence of spoofed packets and thereby protecting the server from SYN flood attack.

3.1. Model of "SYN flood attack protector system" as shown in fig.3 consists of the following main blocks:

3.1.1. Packet identifier   3.1.2. Good registry
3.1.3. Packet verifier   3.1.4. Half open connection detector

3.1.1. Packet identifier: Every incoming packet from internet is intercepted by Packet Identifier. It allows only SYN and ACK packets to the packet verifier block and other packets are sent to the server.

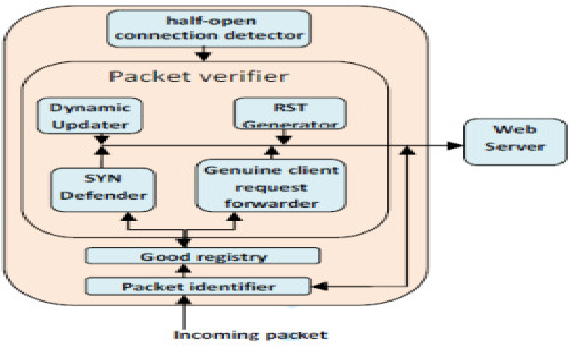

Figure 3: Block diagram of "SYN flood attack protector system"

3.1.2. Good registry: At any given time good registry should maintain the information about the genuine clients. The straight forward method to find genuine client information (IP address) is based on the previous history of the connections set up with the server. But as the spoofed IP addresses are used during SYN flood attack, it is impossible to identify the genuine client's IP addresses, only based on previous history of the connections. Hence, in the proposed method the good registry is used to maintain the genuine client IP addresses and is continuously updated by "dynamic updater", by verifying if the packet is from attacker or from genuine client as explained below in 3.1.3c.

3.1.3. Packet verifier: It verifies whether the incoming request is from genuine client, spoofed/attack client and accordingly the good registry is continuously appended. Based on the IPs stored in the good registry, only the genuine client requests are allowed to set up the connection with the server. Other packets whose IPs are not found in good registry are first verified if it is from genuine client, only then the genuine packets are forwarded to the server. The Packet verifier consists of the following blocks and its functions are as below.
3.1.3a) Genuine Client Requests Forwarder (GCRF)
3.1.3b) SYN defender
3.1.3c) Dynamic updater
3.1.3d) RST generator

3.1.3a) Genuine Client Request Forwarder (GCRF): The GCRF as shown in fig. 4, forwards the genuine client requests (whose IP addresses are found in good registry) to the server. The server sends back SYN-ACK to the client. If there is a corresponding ACK received from the client, then the connection is set up on the server.

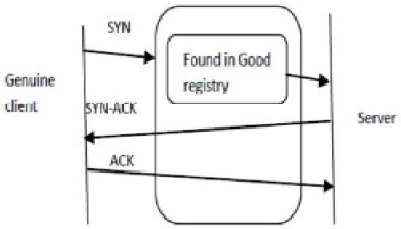

Figure 4: Genuine Client Request Forwarder

3.1.3b) SYN defender: In case the IP address of incoming client request is not found in good registry the request is not forwarded to the server, instead the SYN defender sends the SYN-ACK back to the client on behalf of the server. If the corresponding ACK is received then the SYN defender [8] in turn sets up the connection with the server as shown in fig. 5.

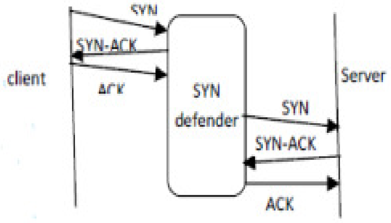

Figure 5: SYN defender

3.1.3c) Dynamic updater: Dynamic updater keeps on updating the genuine client IP addresses in good registry based on the following rules.

RULE 1: If the source IP address of incoming SYN packet is not found in the good registry then that client is considered as bad/new client. This packet request is not forwarded to the server, instead SYN defender is activated.

o *RULE 1.1:* If the three way handshake is completed between the client and the SYN defender then this client IP is updated in the good registry as the client is genuine client.

o *RULE 1.2:* If the three way handshake is *not* completed between the client and SYN defender then the SYN packet is dropped as the client has not send ACK and is an attack packet.

RULE 2: If the source IP address of incoming SYN packet is found in the good registry, then that client is considered as genuine client. But it is possible that it could be spoofed packet. In such case

o *RULE 2.1:* Initially it is assumed that incoming SYN packet is from *genuine client*. The packet is sent to the GCRF which forwards the request to the server and waits for the corresponding ACK till TCP connection-establishment timeout. However in case of large number of requests/SYN flood attack the server waits for ACK for a *stipulated amount of time which is varied* depending upon number of half open connections stored in the backlog queue of the server. This time is very less during attack as number of requests generated is large. In our case we have considered this time as the time taken to set up 75% half open connections of the maximum number of half open connections that can be stored on the backlog queue of the server.

o *RULE 2.2:* If the 75% of half open connections of backlog queue is filled, then the RST generator is activated and transfers the status of all half open connections of the server to the SYN defender. RST generator now generates RST signals to release all half open connections from the server. Also the respective client IP addresses are removed from good registry as they may be *spoofed IP*.

o *RULE 2.3:* The SYN defender in turn waits for the corresponding ACK. If the request is from genuine client, the corresponding ACK is received by SYN defender and SYN defender in turn sets up the connections to the server, at the same time good registry will be updated. With this the service to the genuine clients will not be denied even during the attack.

o *RULE 2.4:* If no ACK is received by SYN defender then the SYN packet is dropped from the SYN defender as it will be a spoofed/attack.

3.1.3d) RST generator: It generates RST packets so as to release half open connections on the server.

3.1.4. Half open connection detector: It detects the maximum number of half open connections set up in the backlog queue of the server. The maximum number of half open connections set up on server depends upon total memory available on server and the memory size required to store single half open connection.

3.2. Working Principle of "SYN flood attack protector system"

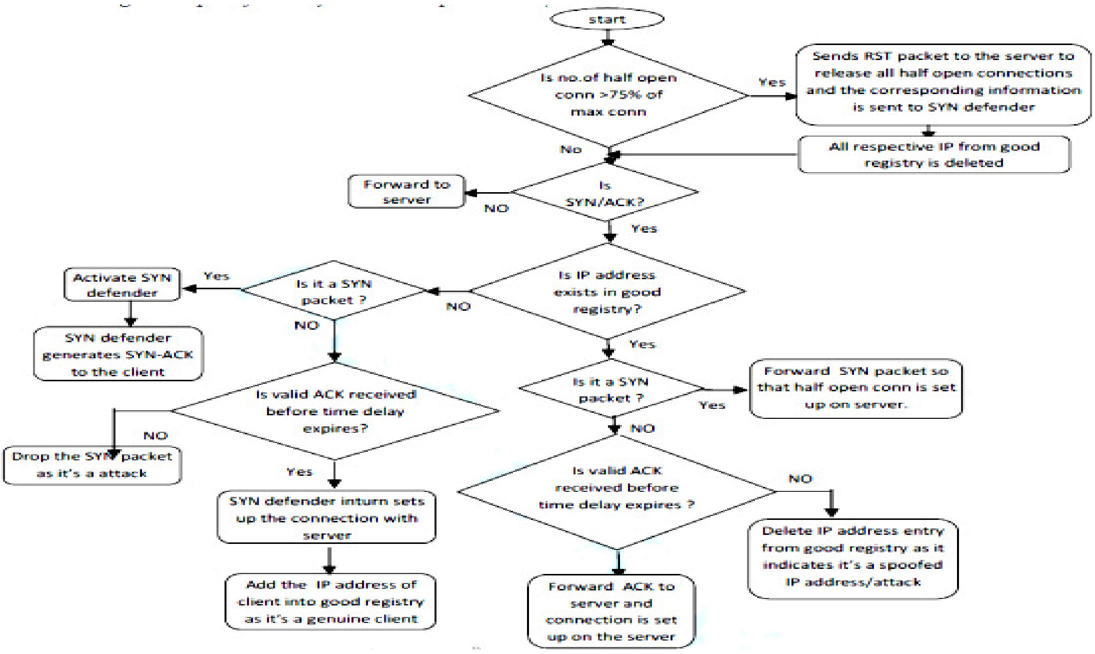
start
Is no.of half open conn >75% of max conn
Yes
Sends RST packet to the server to release all half open connections and the corresponding information is sent to SYN defender
All respective IP from good registry is deleted
No
Is SYN/ACK?
NO
Forward to server
Yes
Is IP address exists in good registry?
NO
Is it a SYN packet ?
Yes
Activate SYN defender
SYN defender generates SYN-ACK to the client
NO
Is valid ACK received before time delay expires?
NO
Drop the SYN packet as it's a attack
Yes
SYN defender inturn sets up the connection with server
Add the IP address of client into good registry as it's a genuine client
Yes
Is it a SYN packet ?
Yes
Forward SYN packet so that half open conn is set up on server.
NO
Is valid ACK received before time delay expires ?
NO
Delete IP address entry from good registry as it indicates it's a spoofed IP address/attack
Forward ACK to server and connection is set up on the server

Figure 6: Flow chart of "SYN flood attack protector system"

The flow chart of "*SYN flood attack protector system"* is as shown in fig. 6 and is explained below.

If the number of half open connections set up on the server is greater than 75% of maximum number of connections that can be stored in the back log queue of the server, then the RST generator is activated. Also SYN flood attack protector system transfers the status of half open connections to the SYN defender as though SYN defender has sent SYN_ACK to client and is waiting for the corresponding ACK. RST generator sends RST packet to the server to release all half open connections of the server and respective IP's from good registry is then deleted. Here all half open connections are released so that server resources are made available for further client requests. Even though half open connections on the server are deleted, the status information is still available in the SYN defender, and if any corresponding ACK is received then the SYN defender will set up the connection on behalf of client. Thus during the SYN flood attack all the genuine clients receive the service from the server**.**

For every incoming SYN packet, the packet verifier compares the source IP address of packet with the good registry.

3.2.1. If the source IP address is found in the good registry then the SYN packet is forwarded to the server by GCRF. The same information is also maintained in the SYN-wait-table by packet verifier. The SYN-wait-table maintains the number of half open connections set up on the server. The forwarded SYN packet sets up the half open connection on the server then the server sends the SYN-ACK to the client and waits for the corresponding ACK from client. Though the client is considered as genuine client, it is always possible that it could be spoofed packet. In such case there are two possibilities as given below.

a. Request is from genuine client: In this case the packet verifier receives the corresponding ACK, it is then forwarded to the server and the connection is set up.
b. Request is from attacker/spoofed IP: In this case the packet verifier does not receive the corresponding ACK (within specified time period), that means the SYN which has been received and sent to server is spoofed.
At this stage dynamic updater removes the corresponding packet information from the good registry and the packet information from SYN-wait table is removed. Thus the good registry is continuously updated and always it maintains the information about genuine clients.

3.2.2. If the source IP address of incoming packet is not found in good registry then, it indicates that the packet may be from an attacker or it could be from new client. In such case packet verifier activates the SYN defender, which in turn sends the SYN-ACK to the client and waits for the corresponding ACK.

a. If the corresponding ACK is received, the SYN defender in turn sets up the connection with the server on behalf of the client. Since ACK is received, it means that the client is a genuine client and hence the Dynamic updater will append client information in good registry.
b. If ACK is not received**,** that means the SYN which is received may be spoofed and hence the SYN packet will be dropped.

Thus by updating the good registry continuously the “SYN flood attack protector” provides service to all genuine clients and blocks the attack clients**.**

4 EXPERIMENTAL SET UP

For this study we had set up the network using three Linux based machines as shown in below fig. 7.

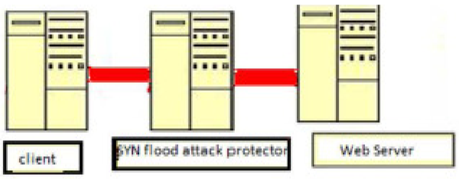

Figure 7: Experimental set up

One machine was configured as Web Server; second machine is used as client and the third machine is connected between the Web Server and client .The third machine acts as s SYN flood attack protector/router. For our experiment we generated the genuine requests from client using httperf tool and SYN attacks using the hping3 tool. The proposed algorithm is implemented on SYN flood attack protector for protecting the Web Server using libipq [21]. Libipq is a development library for iptables user space packet queuing. Libipq provides an API for communicating with ip_queue. On the third machine iptables rule is set to send tcp packets to IPQ as below

```
iptables -A FORWARD -p tcp -j QUEUE
```

Case 1 (The server is not protected using “SYN flood attack protector system”): The client generates the large number of genuine requests to server and the number of connections set up on the server is recorded.

Case 2 (The server is not protected using “SYN flood attack protector system”): The client generates the large number of genuine requests to server in the presence of SYN flood attack and similarly the number of connections set up on the server is recorded.

Case 3 (The server is protected from SYN flood attack using the “SYN flood attack protector system”)**:** In this case the third machine acts as “SYN flood attack protector”. The client generates the genuine client requests in the presence of SYN flood attack and the number of connections set up on the server is recorded similar to above two cases.

5. RESULTS AND DISCUSSION

5.1. In Case1 where the server is not protected from “SYN flood attack protector system” and the SYN flood attack is not generated. All the client requests are allowed to set up connection on the server. Hence all the genuine client requests are served by the server and are shown in fig. 8.

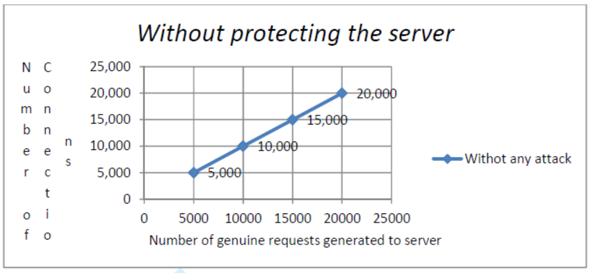

Figure 8: Service to the genuine clients without SYN attack

5.2. In Case2 where the server is still not protected using “SYN flood attack protector system” but the SYN flood attack is generated. The services to almost all the genuine client requests are denied as shown in fig. 9.

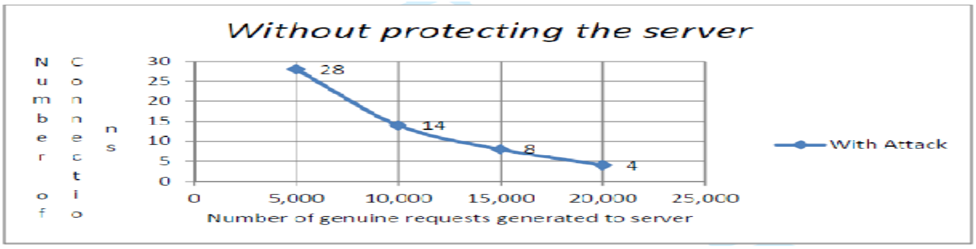

Figure 9: Service to the genuine clients during SYN attack

5.3. In case 3 where the server is protected from SYN flood attack using the "SYN flood attack protector system" and the genuine clients request is generated in the presence of SYN flood attack. All the attack packets are blocked by SYN flood attack protector and all the genuine clients receive service from the server and are shown in fig.10.

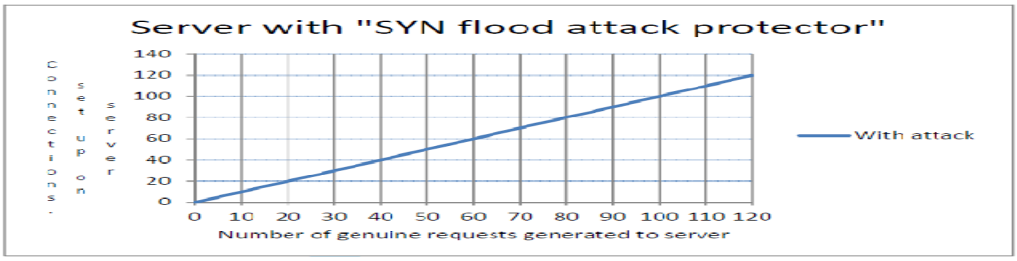

Figure 10: Service to the genuine clients during SYN attack with server protected

This particular case was tested up to 120 number of genuine requests only as there was a limitation due to libipq buffer space available. However in the scaled up model the request reaches the figures of 5,000 to 20,000 we hope that the characteristic shown in fig. 10 will be maintained.

5.4 Advantages of "SYN flood attack protector": In the proposed method when there is no SYN flood attack the "SYN flood attack protector system" checks for the incoming packet's source IP address in good registry and the service is provided to all the genuine clients. Thus when there is no attack, not much processing is done.

During SYN flood attack all the incoming packets IP addresses are compared with good registry and accordingly the packets are either forwarded to server or it is sent to the SYN defender as explained above. Thus "SYN flood attack protector system" needs to do lot of processing to protect the server from the attack. But during SYN flood attack apart from blocking the attack packets, it provides uninterrupted service to all the genuine clients.

5.5 Future Scope: The proposed method at present is implemented purely as software simulation. To further improve the speed of processing we propose to use Field Programmable Gate arrays. FPGA provides the speed, reliability because many of the algorithm blocks can be hardwired and also, has the same flexibility of software based approach running on a general purpose processor. Further improvement can be done to overcome the existing limitations of the libipq buffer, by which large number of genuine requests can be handled by the SYN flood attack protector. This repository of genuine client can be used by other security systems like IDS, Packet filtering etc. for protecting the server.

REFERENCES

[1] Arbor Networks 7th Annual *Worldwide Infrastructure Security Report* VII, 2011 http://www.arbornetworks.com/report.

[2] Ketki Arora et al. Impact analysis of recent DDoS attacks, I.Journal on Computer Science and Engineering (IJCSE) Vol. 3 No. 2 Feb 2011 pp 877-884.

[3] Martine Bellaiche and jean-Charles Gregore, Avoiding DDoS with active Management of Backlog queues, Network and System security, 2011 5th I. Conference, pp 310-315.

[4] Mariusz Burdach, Hardening the TCP/IP stack to SYN attacks, SecurityFocus.com, 2010, http://www.symantec.com/connect/articles/hardening-tcpipstack-syn-attacks.

[5] D. Nashat and X Jiang, Detecting SYN flooding agents under any type of IP spoofing, in IEEE I. Conference on e-business Engineering 2008, pp 409-505.

[6] WikiLeaks Supporters Tear down VISA in DDoS Attack, December 9, 2010. http://www.digitaltrends.com/computing/wikileaks-supporters-tear-down-visa-in-DDoSattack.

[7] Lemon, J., Resisting SYN Flood DoS Attacks with a SYN Cache, BSDCON 2002, February 2002, people.freebsd.org/~jlemon/papers/syncache.pdf.

[8] Check Point Software Technologies Ltd., SynDefender, http://www.checkpoint.com.

[9] Christoph L. Schuba, et al., Analysis of a Denial of Service Attack on TCP, IEEE Symposium on Security and Privacy, May 1997, http://docs.lib.purdue.edu/cstech/1327.

[10] L.Kavisankar, C.Chellappan, A Mitigation model for TCP SYN flooding with IP spoofing, IEEE I. Conference on Recent Trends in Information Technology, ICRTIT 2011, pp 251-256.

[11] Daniel J Bernstein, SYN Cookies , http://cr.yp.to/syncookies.html,1997.

[12] Xianmin Wei, Analysis and Protection of SYN Flood Attack, Advances in Intelligent and Soft Computing, Volume:106, chapter 30, Publisher: Springer Berlin Heidelberg Location: Berlin, Heidelberg DOI: 10.1007/978-3-642-23753-9_30, , 2011 – Springer, pp 183-187.

[13] Han Jianying, Wang Jing, Wang Wei. SYNCookie Implementation and Improvement. China's new communications, 2007, 13:pp 44-46.

[14] Peng Di, Wang Wensheng. The DDoS Defense Technology ResearchBased on SYN Cookie. Information security and confidentiality of communications 2007,2: pp 125-127.

[15] Jian Xiaochun, Wu Zhenqiang, Huo Chengyi, Zhang Jie., Homologous SYN packet twice reception method defence against SYN Flood attacks, Computer Engineering and Design, 2008,6 (29):pp1440-1442.

[16] Bo Hangl , Ruimin Hu, A Novel SYN Cookie Method for TCP Layer DDoS Attack, 2009 I. Conference on Future BioMedical Information Engineering, pp 445-448.

[17] FortiGate DoS Protection, Block Malicious Traffic Before It Affects Critical Applications and Systems. http://www.fortinet.com/sites/default/files/whitepapers/WPDOS.pdf

[18] Sanjeev Kumar, Raja Shekhar Reddy Gade, Article Title: Experimental Evaluation of Juniper Network & apos; s Netscreen-5GT Security Device against Layer4 Flood Attacks Volume: 02, Issue: 01. DOI: 10.4236/jis.2011.21005

[19] Shaila Ghanti, G.M.Naik , "FPGA System for preventing TCP SYN flood attack", International journal of VLSI design , ISSN no 2229-3167, vol 3 no,1 , 2012, Pages: 39-43

[20] R. Richardson, 2008 CSI Computer Crime and Security Survey,CSI, 2008 https: //www.hlncc.com /docs/CSIsurvey2008.pdf

[21] libipq(3) - Linux man page http://linux.die.net/man/3/libipq.

[22] Prof. S.B. Javheri and Shwetambari Ramesh Patil, "Attacks Classification in Network", International Journal of Information Technology and Management Information Systems (IJITMIS), Volume 4, Issue 3, 2013, pp. 1 - 11, ISSN Print: 0976 – 6405, ISSN Online: 0976 – 6413.

[23] Dr. Sandip Nemade, Prof. Manish Kumar Gurjar, Zareena Jamaluddin and Prof. Nishanth N, "Early Detection of SYN Flooding Attack by Adaptive Thresholding (EDSAT): A Novel Method for Detecting SYN Flooding Based Dos Attack in Mobile Ad Hoc Network", International Journal of Advanced Research in Engineering & Technology (IJARET), Volume 5, Issue 2, 2014, pp. 79 - 86, ISSN Print: 0976-6480, ISSN Online: 0976-6499.

[24] Sharada Valiveti, Swati R Sharma and Dr. K Kotecha, "Performance Evaluation of Byzantine Flood Rushing Attack in Ad Hoc Network", International Journal of Electronics and Communication Engineering &Technology (IJECET), Volume 5, Issue 2, 2014, pp. 1 - 9, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.