Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

Analysis and Review of TCP SYN Flood Attack on Network with Its Detection and Performance Metrics

Hrishikesh Shriram Salunkhe
(Dept of Comp Engineering)
MGMCET, Kamothe, Navi Mumbai.

Prof. Sanjay Jadhav
(Dept of Comp Engineering)
SCOE, Kharghar,
Navi Mumbai.

Prof. Vijay Bhosale
(Dept of Comp Engineering)
MGMCET, Kamothe, Navi Mumbai.

Abstract **- The Denial of Service (DOS) attack is most widely employed technique used by attackers on the network in order to disrupt the network functionality. The intention is clearly to pull down the service of the victimized network by making it busy for legitimate users to be accessed and get the desired service; thus ultimately resulting in the poor performance. Among various DOS attacks the SYN flood attack is mostly implemented by attackers. The attack is implemented by focusing and targeting on the TCP's 'three-way handshake mechanism', as there is limitation on maintaining half opened connections. In this the attacker attempts to exploit all the available resources by bogus half connections and thus there may not be resources left to establish new legitimate connection with host. Due to this attack the server may get hang, it may crash or may be occupied fully with the large volume of traffic. In order to check whether the system is under influence of attack, its behavior is compared with normal system on the basis of different parameters. The Adaptive threshold algorithm and the cumulative sum (CUSUM) algorithm are the algorithms for detection which can serve as detection mechanism on the basis of some logical and mathematical model.**

Keywords**: *DDoS, TCP SYN, Adaptive threshold, CUSUM, Hping3, Wireshark***

I. INTRODUCTION

In DDoS attack the network availability is mainly affected by occupying or blocking the victim's resources[11]. It is carried out by sending huge amount of packets to the target node with the well planned arrangement of huge number of hosts which are distributed all over the network. All these host continuously send the packets to the target, thus causing the traffic at the link to the target. This result in the unavailability of the node to the legitimate users. Hence it is clear that DoS attacks are intended to prevent the server to deliver the service to its legitimate user. To perform the denial of service attack, the attacker consumes all resources of that system, hence these are not remain available for other users[11]. The DoS attack, generally attacks the main server which is responsible for providing services to the computer network. Fig. 1 gives the classification of DoS attack

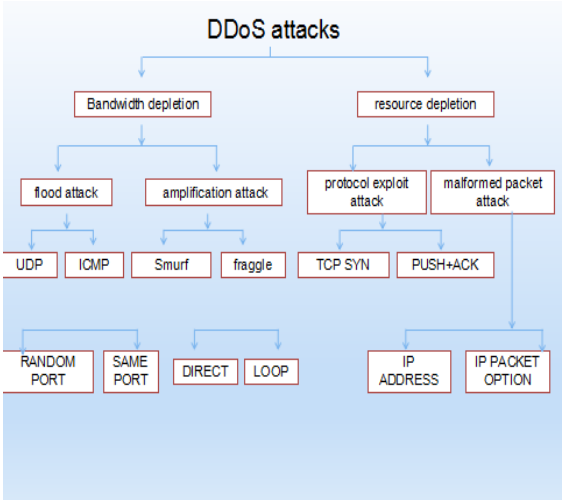

Fig 1 : The DDoS attack taxonomy

Attack scenarios statistics [12]: The fig. 2 shows In January 2016 reported DDoS attacks leads the chart of the known techniques with 22.3%, ahead of Account Hijacks (13.8%) and Defacements (10.6%). Targeted attacks are immediately behind with a remarkable 7.4%.

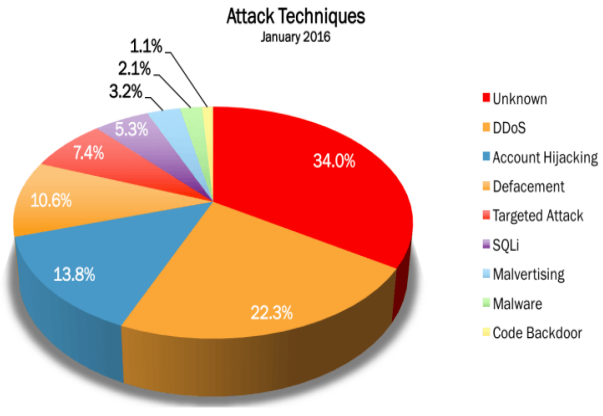

Fig 2 : The Attack scenarios statistics [12]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

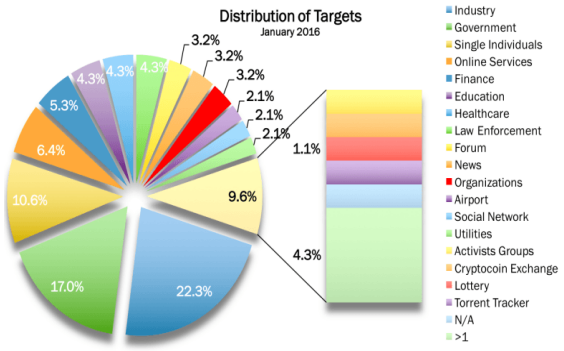

Fig 3 : Distribution of targets [12]

Fig. 3 depicts that the Industries lead the Distribution of Targets chart with 22.3%, ahead of Governments (17.0%) and Single Individuals (10.6%). Online Services (6.4%) and Financial Targets (5.3%) emerge from the other targets.[12]

The ranking of the most popular attack methods shown in fig. 4 remained constant from quarter to quarter[2]. Those used most often were the SYN DDoS method, although its share fell compared to the previous quarter (57.0% vs 54.9%), and TCP DDoS which fell by 0.7 percentage point. The proportion of ICMP DDoS attacks grew significantly, rising to 9%; however, it did not affect the order of the Top 5.

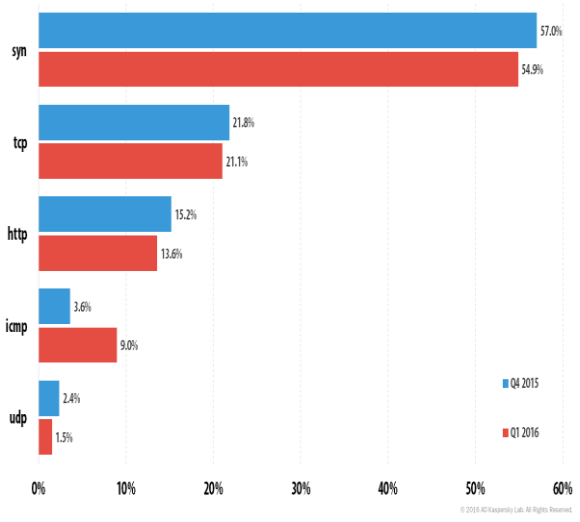

Fig 4 : Distribution of DDoS attacks by type [2]

A. Distributed Denial of service attack scenario:

Multiple attackers attempts to bombard the victim node by sending the request at the same time. This bombardment of fake requests causes the computer resources to be fully occupied and results in the denial of service to the legitimate users. This is achieved by the attacker by commanding large number of remotely controlled computers to follow the network traffic at the victim.[10][11] as in fig 5.

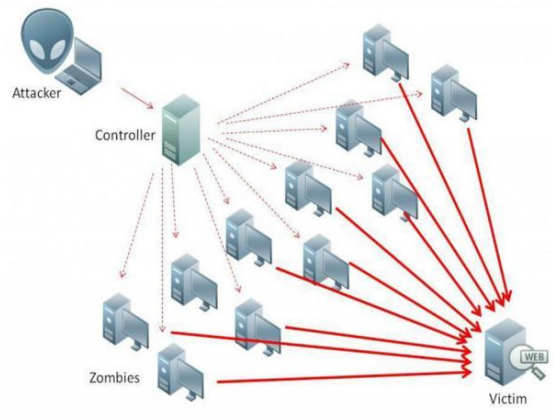

Fig 5 : The Architecture of DDoS attack

The victim node becomes busy with handling request from these host systems and unable to respond legitimate users' requests. Such remotely controlled systems that are involved as attacking agents known as 'Zombie' and a large network of zombie computers is called a 'Robot network', or 'Botnet'.

B. SYN Flood

The SYN flood attack is based on its design of the 3-way handshake which initiates the TCP connection.[8][10][11][13]

- *Normal TCP three-way handshake:*

Steps of normal TCP communication :

1. The client hosts initiates the connection by sending a SYN / Synchronize.
2. The server replies back to the client by sending SYN-ACK and thus responds / acknowledges the original SYN request. Then after a client sends the ACK as a response back to the server, and the connection is established.

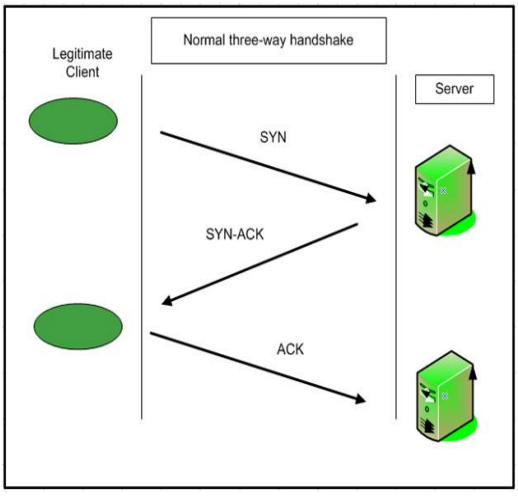

Fig 6 : The 3-way TCP Handshake

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

The Fig 6. depicts the TCP three-way handshake process.

- TCP SYN Flood:

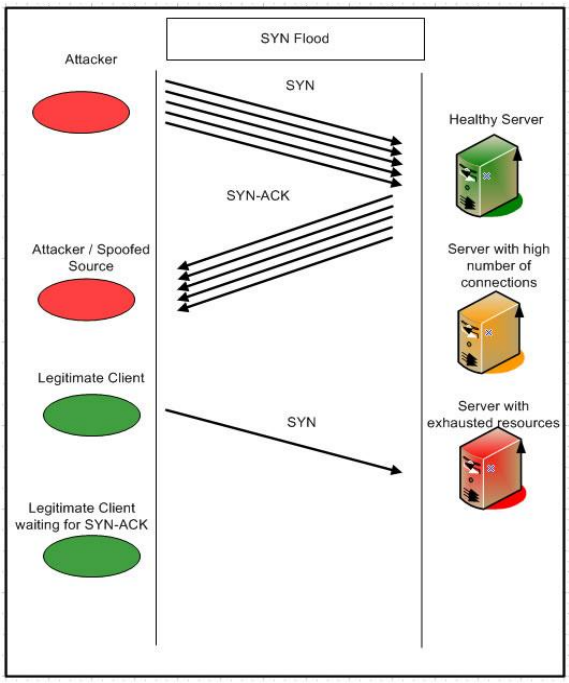

Fig 7 : SYN Flood Attack

An attacker client sends the TCP SYN connections at a high rate to the victim machine, more than what the victim can process. Fig 7 This is a form of resource exhausting denial of service attack. The attacker client can do the effective SYN attack using two methods.[11]

1. The malicious client will keep sending the SYN packets which are usually of 64 bytes. And when the victim server responses by sending the SYN-ACK messages as an acknowledgment to the malicious client, the malicious attacker client just ignores the SYN-ACK message and continue to send the new SYN messages.

2. In other method, the malicious client spoof the source IP address and sends the SYN packets to the victim server. The Victim server sends the SYN-ACK to the spoofed source address and waits for the ACK response. Since the spoofed sources did not originally send the SYN packets, they never respond back. And the Victim server holds up the connection..

In this way many number of connections are initiated to the server, the server waits for the response back from the client and keeps the connections in its connection table. This exhausts the resources of the server. Resulting in the denial of service. As the legitimate users will not be responded or get the service from server due to unavailability of the resources.

II. COMMON DEFENSES [11]

The TCP SYN common defense techniques are explained in the [11] viz. Filtering, Increasing backlog, Reducing SYN-RECEIVED timer Recycling the older half open TCB, SYN Cache, SYN cookies etc.

A. Filtering

This attack require the efficiency to send the packets with spoofed source IP addresses, in the absence of group of the controlled hosts. Thus by targeting and blocking an attacker's ability to send the spoofed IP packets can be thought as the effective solution which does not require any modification in the TCP. The Filtering is done in Unix based system by changing or modifying the '**Sysctl.config file'** in **'/etc'** directory**.** This file is configuration file where we can set the system variables. Shown in fig.8. The code

"net.ipv4.conf.default.rp_filter=1"

"net.ipv4.conf.all.rp_filter=1" is added to the file.

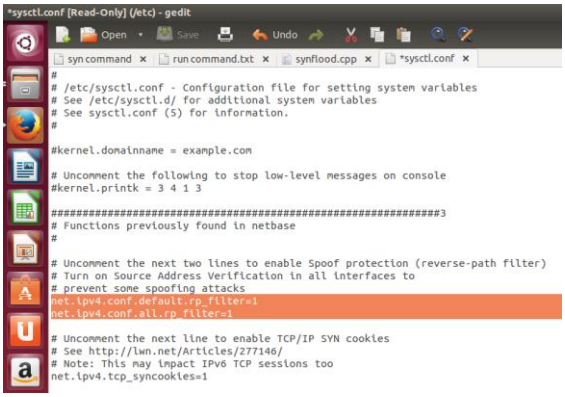

Fig. 8 : setting Syn Filter in sysctl.conf file

B. Increasing Backlog

It is one of the simplest and logical technique to pull down the intention of denial of service due to resource occupancy. But the technique is inefficient for the larger backlogs since the implementation is not designed to handle the larger past backlogs

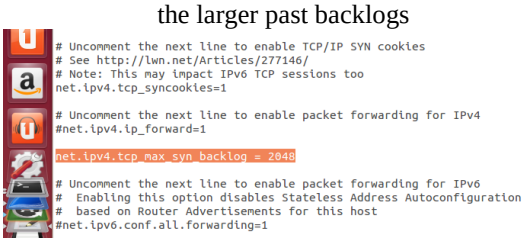

Fig. 9 : setting Backlog in sysctl.config file

Same case applies to TCP implementations which have similar design factors that have limitations in their performance with large backlogs, and also there are other solution present which can be applied without changing the data structure. In Unix based system the **'Sysctl.config'** file in **'/etc'** directory. As shown in fig. 9. The code

"net.ipv4.tcp_max_syn_backlog = 2048"

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

is added to the file. By default the backlog value is 256. that means 256 half opened connection can be handled.

C. Reducing SYN-RECEIVED Timer

This few of the quickly applied techniques of defense. The timeout period is reduced between SYN receive and removing the generated TCB for lack of progress. Due to this the lifetime of TCBs in SYN-RECEIVED is reduced hence bogus connection will be in the backlog for shorter time and space is free for legitimate user for establishment of connection. But sometimes it may prevent the connections of legitimate connection o be fully established. The time reduction may be implemented for trapping the occupancy in the backlog by putting the threshold value, and to also to trap the rate of SYN reception.

D. SYN Cache

In the SYN cache technique the amount of state is minimized which SYN allocates instead of allocating full TCB. Unless and untill the connection is fully established the full state allocation is delayed. In this case the incoming SYN SEGMENT consist of some secret bits, these bits are selected by the host. The hashing is performed on these secret bits along with the IP addresses and TCP ports of a segment, and the hash value which is generated decides the location in a global hash table where the incomplete TCB is stored. The bucket limit is set for each hash value, and the oldest entry is dropped after this limit is reached. This is effective because the secret bits prevent an attacker from being able to target specific hash values for overflowing the bucket limit, and it bounds both the CPU time and memory requirements. The data will require re transmission, if it accompanies the SYN segment, then this data is not acknowledged or stored by the receiver.this technique has limitation that it affects the system performance by affecting data transfer rate. But system is still reliable.

E. SYN Cookies

SYN cookies is stateless method. It does not allocate the state in SYN-RECEIVED. The states are encoded and the sequence number is assigned to them on the basis of transmission on the SYN-ACK.

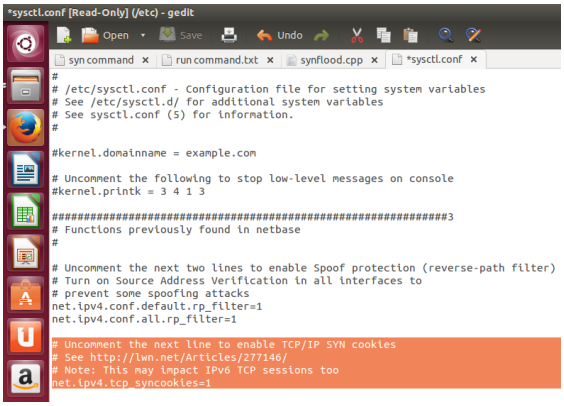

Fig. 10 : setting SYN cookies in sysctl.config file

For the unspoofed SYN the acknowledgment number along with other credentials in the ACK which require for handshake can be used for reconstruction of TCB state. The SYN cookies enabled in Unix based system by changing or modifying the '**Sysctl.config file'** in **'/etc'** directory. As shown in fig.10. The code

"net.ipv4.tcp_syncookies=1"

A technique responsible to prevent the replay, some time dependent random bits embedded in the sequence number. One of the techniques used 7 bits for these bits and 25 bits as data bits. To encode these bits is to XOR the initial sequence number received with the following parameters : truncated cryptographic hash of the IP address and TCP port number pairs, and secret bits. SYN cookies are enabled only under high stress condition.

III. RELATED WORK

Many different ideas have been put forward for the detection of SYN flooding attacks. In [1] the authors have explained the method to detect the SYN flooding attacks at leaf router. The leaf router is connecting the end host to the internet, In a given time interval the normalized difference between the number of SYNs packets and the FIN(RST) is utilized. If at any point of time if the difference occurs such that SYN is at much higher rate, the router detects the attacking condition. In [3], it has considered a non-parametric cumulative sum algorithm; the authors have applied it to measure the number of only SYN packets, and once the change of packets occur; in order to obtained the recent estimate of the mean rate an exponential weighted moving average(EWMA) is used. In [4] three counters algorithms for SYN flooding defense attacks are given. The three schemes include detection and mitigation. The detection scheme utilizes the inherent TCP valid SYN–FIN pairs behavior, hence is capable of detecting various SYN flooding attacks with high accuracy and short response time. The mitigation scheme works in high reliable manner for victim to detect the SYN packets of SYN flooding attack. Although the given schemes are stateless and required low computation overhead, making itself immune to SYN flooding attacks, the attackers may retransmit every SYN packet more than one time to destroy the mitigation function. It is necessary to make it more robust and adaptive. In [5] another method is explained which has comparitively faster execution. For DoS SYN flood attack detection the linear prediction analysis is applied. The technique exploits the exponential backoff property of TCP used during timeouts. Here the difference of SYN and SYN-ACK packets is calculated, and attack can be detected in short delays. In this method the attack can be detected without maintaining any state at leaf routers and firewall. The detection of flooding agent is proposed by the authors of [6] by the consideration of all kinds of the possible IP spoofing, which is based on the SYN/SYN-ACK protocol pair with the packet header information. In [9], The standard model has been generated by authors by observations from the characteristic between the SYN packet and the SYN+ACK response packet from the server by a program for the activity of the server.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

IV. ALGORITHMS

The two very well known algorithms for SYN flood detection are as follows.[10][3]

A. *Adaptive Threshold Algorithm [10]:*

This algorithm is used for the measurement of network traffic in this case SYN packets and compare it with the already set value in the system called as threshold. Every time the value is adaptively altered in certain period of time and the alteration is based on calculated mean number of SYN packets. If incoming traffic crosses the set limit i.e. threshold is exceeded then it indicates the anomalous condition and alarm will be activated.

Let us assume that the number of SYN packet in the p-th time interval is kp , and measured mean rate prior to p is σp-1. In this case the alarm condition is as following:

If,

kp >= (x+1)σp-1 (1)

Then,

ALARM signaled at time p,

where x > 0 parameter indicates the percentage above mean value which is considered as a threshold.

The mean σp can be computed over reference of some past time window or using an exponentially weighted moving average (EWMA) of previous measurements

σn = λσp-1 + (1-λ) kp (2)

{Where λ is EWMA factor.}

On the arrival of packets the algorithm checks if the packets are TCP, and if this is True it checks if TCP packets are SYN or other type of TCP. If the SYN bit is on than it will update the record. Calculating the **σ** and **x** rate than it will check the condition for **kp** either it is greater than **(x + 1) σn-1** or not. If it is greater it will generate the alarm.

B. *CUSUM Algorithm[3]:*

Cumulative sum (CUSUM) algorithm used for checking a measuring system in operation for any departure from some target or specified values and have been widely used for detecting the small and moderate mean shifts.

The non-parametric CUSUM [3] is used to detect TCP SYN flooding attacks. CUSUM monitors a set of n SYN packet sample interval {p1… pn} where pn is the sum of all SYN packets in n-th sample interval. If the change SYN traffic {pi} is distribution of known variance σ^2 and A_0 and A_1 are the mean SYN traffic before and after the change.

Then CUSUM can be written as:

$$C_n = \left[C_{n-1} + \frac{A_1 - A_2}{\sigma^2}\left(p_n - \frac{A_1 + A_0}{2} \right) \right]^+ \quad (3)$$

For complex and time consuming calculations we consider a simple approach to apply CUSUM to

$\overline{S_n}$ with $\overline{S_n} = S_n - \overline{A}_{n-1}$ (4)

$\overline{S_n}$ is sum of all SYN packets in the n-th sample interval and $\overline{A_n}$ is estimate mean number of SYN of SYN packets at sample n which is computed using EWMA as

$$A_n = E\overline{A_{n-1}} + (1-E)S_n \quad (5)$$

The mean number of SYN packets after change A_1 given by, $\alpha \overline{A}_n$ (6)

where α is amplitude percentage parameter.

The CUSUM written as:

$$C_n = \left[C_{n-1} + \frac{\alpha\overline{A_{n-1}}}{\sigma^2}\left(S_n - \overline{A_{n-1}} - \frac{\alpha\overline{A_{n-1}}}{2} \right) \right]^+ \quad (7)$$

V. PERFORMANCE AND ANALYSIS

- TCP SYN general scenario:

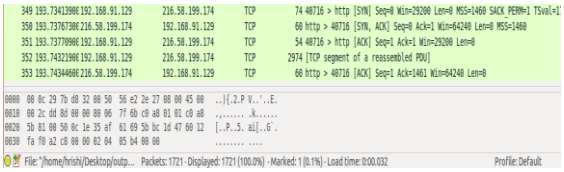

Fig. 11 : TCP SYN packets (Wireshark)

In the Fig. 11, The successful TCP connectionn based on three way handshake is shown, the SYN request packets are sent to the web server from the public IP address and SYN flag is set to 1, then after receiving the Acknowledgement from the server [SYN ACK], the ACK set to 1 but seq is still 0 indicating final Acknowledgement is pending, and finally it is seen that:

'40716 > [ACK] seq=1 Ack=1' indicates the connection establishment.

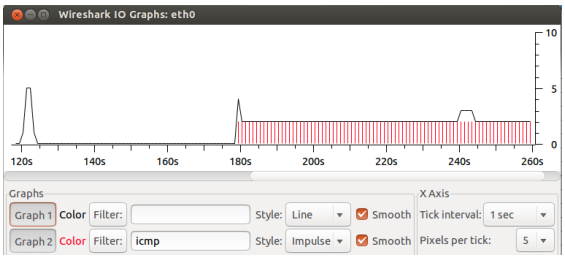

Fig. 12 : Normal traffic IO graph in the absence of attack

The above fig. 12 shows the IO Graph of the network traffic.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

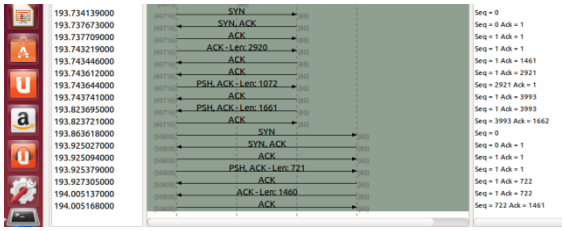

Fig 13 : flow line graph in the absence of attack

In fig. 13 we can observe the successful TCP SYN connection requests. The 'three way handshake mechanism' is smoothly working. And all the packets viz. SYN , SYN-ACK and ACK are sequentially being sent and received between source and destination. Also we can observe PSH packet i.e Push function, it causes the TCP sender to push all unsent data to the receiver rather than sends segments when the buffer is full.

- The implementation of TCP SYN flood attack in the Ubuntu:

The TCP SYN flood DOS attack is implemented in the Ubuntu. The Hping3 is used for sending the TCP SYN packets, which are captured by network analyzer tool, Wireshark.
The command used in the terminal is as follows:

$ sudo hping3 -i u1 -S -p 80 -c 1000 192.168.1.1

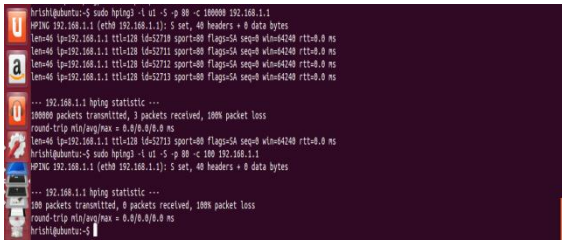

Fig. 14 : Initiating the attack by Hping3

The command sends TCP SYN packets to 192.168.1.1

sudo is necessary as the hping3 creates raw packets for the task

S - indicates SYN flag
p 80 - Target port 80
i u1 - Wait for 1 micro second between each packet

c - indicates the number of packets to send/receive

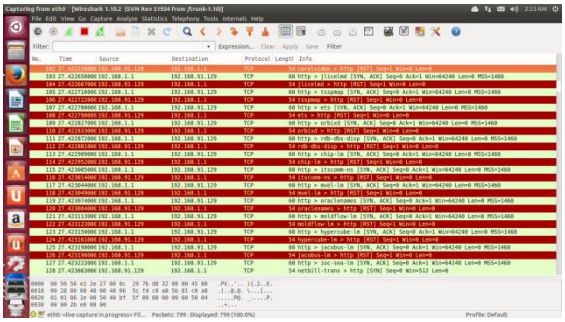

Fig. 15 : Unsuccessful half opened connection

As in fig. 15 huge number of packets come in very short time interval, it is the flooding condition, where the TCP packets could not establish the connection. Thus source 192.168.91.129 sends syn packets to 192.168.1.1 which replies with [SYN ACK], but the source could not send ACK, instead it sends [RST] indicates the abnormal condition. and it declines the connection.

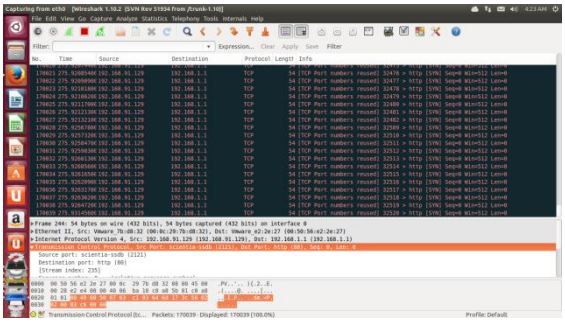

Fig. 16 : Unacknowledged failed connection requests

The fig. 16 depicts the huge number of TCP SYN packets which are coming to the victim 192.168.91.129 in very short time interval. The black colored packet information indicates the bad TCP packets and the packets of SYN flood which we are trying to trace.

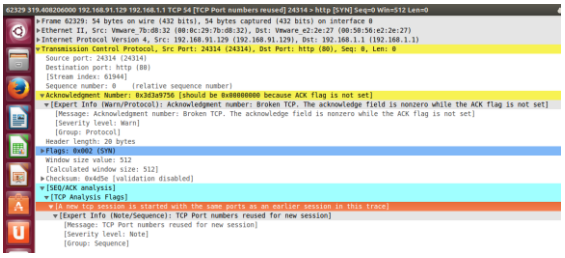

Fig. 17 : packet and flags information

If we click any packet under attack we could get the information as shown in fig. 17, here the information of packet number 62329 is given in details. The source port is 24314 and destination port is 80; in Expert-Info block the 'Severity level:warn' indicates fast restransmission suspected and warning, e.g. application returned an "unusual" error code like a connection problem.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 01, January-2017

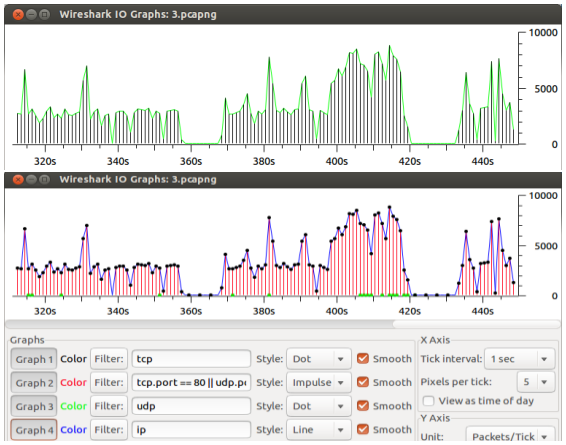

Fig. 18 : IO graph of the ongoing TCP SYN attack

Fig. 18 shows the IO graphs for the flooding network capture file. The continuous TCP packets are being bombarded on the victim 192.168.91.129.

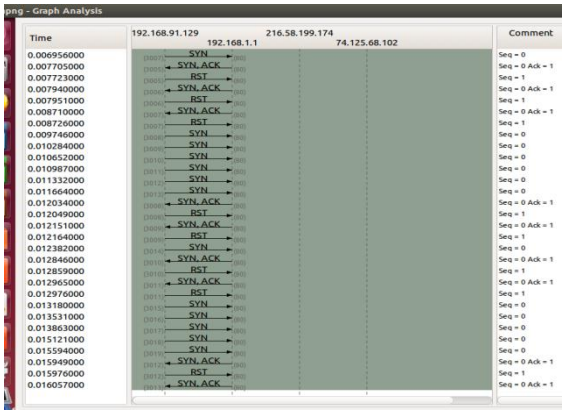

Fig. 19 : Flow graph under attack

The fig 19 depicts the flow graph. Here we can see clearly that TCP SYN packets are being transmitted between 192.168.91.129 and 192.168.1.1. at times SYN and then SYN ACK is set but final ACK is not set result in half open connection. In few cases continuous SYN packets are sent which are not acknowledged at all. In some cases the RST packet is sent to terminate the connection.

- Effect of attack:

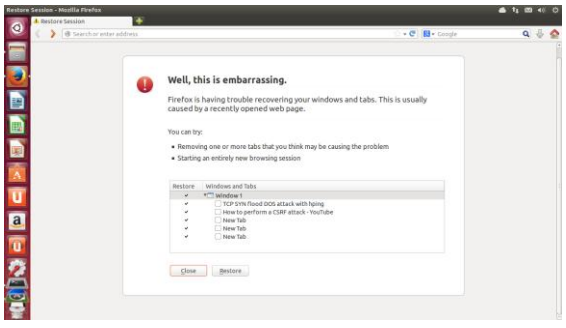

Fig. 20 : Effect of attack on server

The fig. 20 is showing the ultimate presence of the flood attack. and its effect. Here when we try to open the web browser it gives the above screen. Saying that Firefox is having trouble in recovering the session. This clearly indicates that the victim server is affected and system performance is brought down by the TCP SYN flood.

CONCLUSION

In this paper we have implemented the TCP SYN flood attack and by capturing the network packets The DoS attack is detected. The main intention of denial of service attack is to pull down the availability and even in some cases the failure of the server by occupying the resources and trafficking the network link to the victim. There arise many possibilities for detecting and blocking the attack by various methods, few of which are efficient with certain limitations and few are efficient with certain overheads to the system resources. Adaptive threshold and CUSUM algorithm holds good and work well under set up of the limiting threshold of the SYN receive packets. Betterment of the techniques and inventing new method to counterattack will be thought as the future work.

REFERENCES

[1] H. Wang, D. Zhang, and K. G. Shin, “Detecting SYN flooding attacks”, in Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM), volume 3, pp. 1530-1539, June 23-27, 2002.

[2] https://securelist.com/analysis/quarterly-malware-reports/74550/kaspersky-ddos-intelligence-report-for-q1-2016/

[3] V. A. Siris and P. Fotini, “Application of nomaly Detect Algorithms for Detecting SYN Flooding Attack” Elsevier Computer Communications, pp. 1433-1442, 2006.

[4] S.Gavaskar, R.Surendiran and Dr.E.Ramaraj, “Three Counter Defense Mechanism for SYN Flooding Attacks”, International Journal of Computer Applications, Volume 6–No.6, pp.12-15, Sep. 2010.

[5] D. M. Divakaran, H. A. Murthy and T. A. Gonsalves, “Detection of SYN Flooding Attacks Using Linear Prediction Analysis”, 14th IEEE International Conference on Networks, ICON 2006, pp. 218-223, Sep. 2006.

[6] D. Nashat, X. Jiang and S. Horiguchi, “Detecting SYN Flooding Agents under Any Type of IP Spoofing”, IEEE International Conference on e-Business Engineering table of contents, 2008.

[7] W. Chen and D.-Y. Yeung, “Defending Against SYN Flooding Attacks Under Different Types of IP Spoofing”, ICN/ICONS/MCL ’06, IEEE Computer Society, pp. 38-44, April 2006.

[8] Subramani Rao “Denial service attacks mitigation techniques real time implementation detailed analysis” -33764 the SANS institute.

[9] T. Nakashima and S. Oshima, “A detective method for SYN flood attacks”, First International Conference on Innovative Computing, Information and Control, 2006.

[10] Mitko Bogdanoski, Tomislav Shuminoski and Aleksandar Risteski, “Analysis of SYN flood DOS attack”, I.J. Computer network and Information Security 2013, 8, 1-11.

[11] W. M. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," RFC 4987, August 2007.[Online].Available:http://tools.ietf.org/html/rfc4987.

[12] http://www.hackmageddon.com/2016/02/16/january-2016-cyber-attacks-statistics/

[13] “Detection and Performance Evaluation of DoS/DDoS Attacks using SYN Flooding Attacks”, (ICICT- 2014)

[14] http://www.rhyshaden.com/tcp.html