Tcp Syn Flood Attack Detection and Prevention System using Adaptive Thresholding Method

Ramkumar B N[1,*] *and Subbulakshmi* T[1]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Abstract. Transmission Control Protocol Synchronized (SYN) flooding contributes to a major part of the Denial of service attacks (Dos) because of the easy to exploit nature of the TCP three way handshake mechanism. Attackers use this weakness to overflow the TCP queue of the server and make its re-sources consumed resulting it to be unavailable for the requests of legitimate users. So we are in need of a quick and precise defence mechanism to detect the TCP-SYN Flood attack. The main objective of the paper is to propose a detection and prevention mechanism of the TCP-SYN flood attack using adaptive thresholding. Adaptive threshold algorithm (ATA) is used to calculate dynamic threshold .Thus this algorithm helps to overcome the limitations of static thresholding like high false positive ratio and also alert users after violation of the threshold calculated by adaptive thresholding algorithm. The result of the suggested mechanism is very effective in the detection and prevention of the TCP SYN flood attack using adaptive thresholding algorithm.

1 Introduction

The recent advancements in technology and the widespread use of the internet have resulted in the need of the internet security at an alarming rate. Denial of service attacks (Dos) one of the dangerous attacks towards the computer network. Dos compromise the availability of the service which is a very important aspect in today's business world. The widespread use of TCP protocol and easy to exploit nature of the TCP three way handshake mechanism has resulted in Dos becoming more common among the cyber-attacks. This paper proposes an effective way of preventing the TCP SYN flood attack using the adaptive thresholding algorithm. This method detects the anomalous TCP requests by monitoring the rate of TCP SYN packets from the attacker to the system.

The TCP three way handshake mechanism is responsible for creating a TCP connection between a client and server. To create a TCP connection a client must send a synchronize flag packet (SYN) to the server as shown in Fig.1. After receiving the SYN packet sent by the client the server sends an acknowledgement flag for the synchronize packet (SYN-ACK) to the client. Upon receiving SYN-ACK flag from the server the client sends an acknowledgement flag to server. After these three stages a connection between both client and server is created and the transformation of data can begin now.

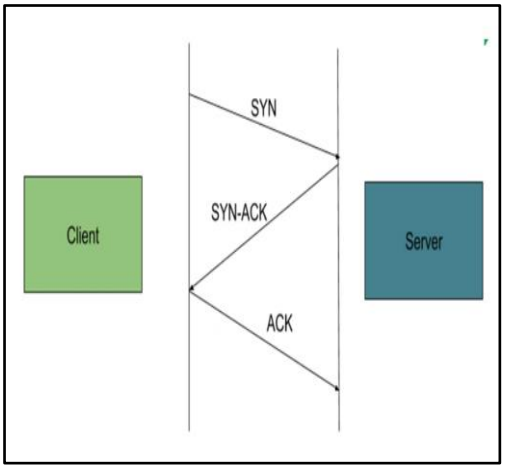

Fig.1 TCP three way handshake mechanism

To perform TCP SYN flood attack on servers, attackers exploit the half opened connection state of the server. This is the state where the server waits for the ACK flag from the client to create a connection. During this state the server would have already allocated memory resources to the client. For exploiting this behaviour attacker sends enormous amount of SYN flags to the server as shown in Fig.2 so that the system would allocate memory resources and wait for the ACK flag from the client which it would never receive. This results in opening of illegitimate half open connections and wastage of memory resources in this server until the session gets expired. During the attack if a legitimate user requests for a connection the server would not respond to the request

*Corresponding author: bnramkumar492@gmail.com

as all the memory resources are allocated to the illegitimate request from the attacker.

To attack a server with TCP SYN flood attack the attacker sends a large amount of TCP SYN flag from different spoofed IP address to the server. The server takes these requests as legitimate and allocates memory and resources to these IP sources and sends an SYN-ACK flag to the client. The server would now wait in half open-ended state expecting the ACK flag from the client. The large amount of illegitimate SYN requests send by the attacker results in the overflow of the TCP backlog queue and create half opened connection until all system resources are depleted.

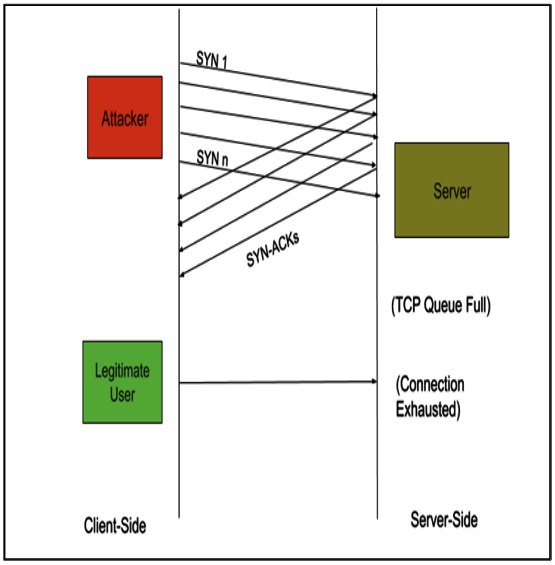

Fig.2 TCP SYN-Flood attack on a server by an attacker.

Due to the overflow of the TCP queue the request made by the legitimate user are not accepted by the server. The main motive of the TCP SYN flood attack is to affect the availability of the system which poses a major threat to the business aspect of the organization.

2 Related works

Several methods for detecting TCP SYN Flood attacks have been proposed. Some of the techniques proposed are the following:

A SYN flood detection system was proposed by Y.Ohista [1] considering the time variation of the incoming traffic. They modelled the arrival of the normal TCP SYN packets into normal distribution where the anomalous requests fail to follow the normal distribution. This method is quick to detect attacks but fails in the case of low time variation attacks because it also follows the normal distribution. H.wang [2] proposed a mechanism which detects the attack at routers instead of the victim's end. This detection mechanism uses a non-parametric CUSUM method to detect TCP SYN floods at a low computational cost. The mechanism considered the nature of the TCP FIN and RST flags to detect the Changepoint effectively. The proposed mechanism provides alerts during the detection of the attack and also reveals the flooding sources location. Schuba [3] proposed that the SYNKILL mechanism is capable for detecting the TCP SYN flood attacks. This mechanism classifies all the incoming packet's IP sources as good or bad. The addresses which are classified bad are sent a TCP-RST packets which would reveal whether the packet from the IP source is spoofed or not. Blazek [4] proposed a method which involves inspection of packets control bits during the observation period using the Cumulative Sum (CUSUM) mechanism for detecting the attack. This mechanism fails in case of a flash crowd where the number of requests over an observation period will be higher than the normal, resulting in a false positive result. Jin and Yeung [5] proposed a covariance analysis model to detect the SYN flooding attack .The mechanism stated that where the attack can be detected by inspecting the degree of correlation between the TCP SYN packets. They used the difference between the correlation of the normal traffic and the traffic during an attack to detect the on-going attack. Siris and Papagalou [6] explored the two statistical anomaly detection algorithms adaptive thresholding and Cumulative Sum algorithm for the detection of the TCP SYN flood attacks. They also provided the suggestion for the performance improvement of the above Changepoint detection algorithms. D. Kshirsagar [7] proposed a mechanism for detecting the TCP SYN flood attack by combining the thresholding and misuse detection approach. The results are measured in terms of CPU workload by comparing the CPU workload during and after detection of the attack.

K.Pai [8] proposed a system where the number of TCP SYN packets is taken as a metric to determine the flooding attack. The number of SYN packets in a particular interval of time from a source is greater than the threshold is considered as an attack. This system can detect the attack precisely but implementing the system for large networks is very difficult. The authors of [9] proposed an efficient methodology for preventing the TCP SYN flooding attacks using the iptables firewalls. They have also explained about various functionalities of the IPtables firewall and making firewall rules for preventing the attacks. The authors of [10] presented a detection mechanism where the SYN flood attack is detected by monitoring the anomalous TCP handshakes between the client and server. This method employs the Cumulative Sum (CUSUM) algorithm for detecting the change. Nakashima and Sueyoshi [11] proposed a method which uses the packet loss rate of the TCP packets as a metric to detect the difference between the normal and abnormal traffic flow during the attack. Lu et al. presented a new framework to effectively identify packets which are compromised. It uses a perimeter-security based Distributed Denial of Service (DDoS) prevention system to classify compromised packets at the router end [12]. The authors of [13] proposed real time architecture for DDos detection using cluster analysis. In this method, the authors extracted particular features from a DDoS architecture and selected variables based on the feature. Wei et al. [14] proposed a detection mechanism using the rank correlation (RCD) of the incoming traffic to detect the change between the normal and abnormal traffic

through the network. S.H.C.Haris et al. [15] proposed a method in which the network is monitored for anomalies in the payload of the incoming packets to detect TCP SYN flood attacks.

3 Environment Setup for the proposed architecture

Proposed TCP-SYN Flood attack detection and prevention system using adaptive thresholding method architecture is implemented in Linux operating systems.

3.1 Python

Python is a general purpose high level programming language which follows an object oriented programming approach. Python programming language is easy to learn and also provides the advantage of readability. Python is well known for its packages and modules which provides code reusability and program modularity. Python is coming pre-installed with operating systems like Linux and Mac. Since windows don't have Python pre-installed, it must be installed explicitly. In Windows, there's no universal library for installing Python, so it must be downloaded like all other GUI applications.

3.2 Scapy

Scapy is a python programming library supported by Python and its later versions. It is used for analysing the packets on the network. It has the ability to decode or forge packets, capture them, send them on the wire, and match requests and replies. It can also handle tasks like probing, unit tests, attacks, and network discovery, scanning, tracerouting [16]. This python library can be used to develop more advanced tools related to network security and ethical hacking. As scapy library is not included with Python libraries by default, scapy python libraries can be installed using pip for our proposed architecture.

3.3 IPtables

IPtables is user-space utility software that empowers a system administrator to configure the rules of IP packet filtering using the Linux kernel firewall, implemented as different Netfiltering modules. The filters are categorized in different tables, which consist of chains of rules for how to deal with network traffic packets [17].IPtables are pre-installed with newer versions of Linux operating systems. If not installed in the system, users can install it through apt-get from the repositories of their own flavours of Linux. IPtables can also be installed in Solaris operating system as a substitute for a firewall.

3.4 Hping3

Attack generation module uses hping3 tool for creating the TCP SYN Flood from the attack generation module to detection module.hping3 is a powerful to create Dos attack against systems [18]. It is one of the most common tools used for testing of the network security of an organization.hping3 tool should be executed for simulating the TCP SYN flood attack. Hping3 can be installed as a source tarball from the project website. Installation for Debian or Ubuntu operating system can be done either with apt-get or Synaptic Package Manager.

3.5 Smtplib

The architecture model is implemented in python so the proposed architecture requires the installation of python SMTP Libraries for sending mails to the administrator [19]. Python offers smtplib module, which creates an SMTP client session object which can be used to send emails. The smtplib module for sending mails can be installed through pip command. As it is supported with python it can be used in all types of operating systems including Windows and mac OS.

4 Proposed TCP-SYN Flood attack detection and prevention system.

Proposed TCP-SYN Flood attack detection and prevention system using adaptive thresholding method architecture is mainly based upon the anomalous TCP hand-shakes behaviour. The detection and prevention mechanism architecture is divided into five modules: the attack generation module, the detection module, prevention module, adaptive thresholding module and an alert module as shown in Fig. 3. The detection and prevention is done at the end of the victim's computer. The detection module in the architecture is responsible for detection and classification of the network traffic. The detection module collects all the necessary information for detection from the incoming network traffic and analyses the data and classifies whether traffic is normal or not using the detection rule. It analyses the information collected by the sniffing process and, according to the adaptive threshold value provided by the adaptive thresholding module, it makes the decision to classify a request into malicious or not.

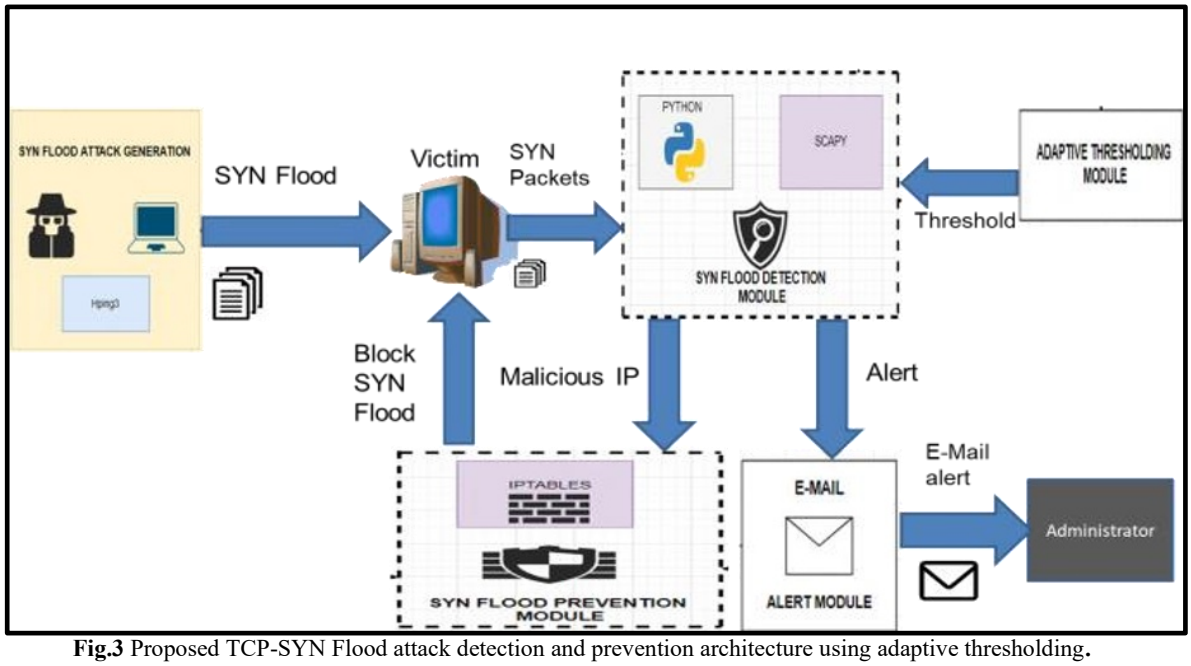

Fig.3 Proposed TCP-SYN Flood attack detection and prevention architecture using adaptive thresholding**.**

The prevention module is triggered when it gets the alert from the detection module. It uses the information from the alert messages to prevent the SYN flooding attack by using IPtables. The alert module uses Simple Mail Transfer protocol (SNMP) to alert the possibility of the on-going attack. Thus reducing the catastrophe of the flooding attack by deciding earlier whether to prevent the attack or to avoid false positive from the detection module.

4.1 Detection Module

As it is stated before, to detect Denial of Service (DOS) attacks, the module monitors the network traffic. So the proposed architecture needs information related to every network traffic from a IP source differentiate normal from abnormal conditions .The detection module needs two important metrics to detected whether a request is anomalous or not:

- The detection module requires an adaptive threshold for classifying the traffic.
- The number of TCP connections made by a source.

In this proposed detection method it uses the number of requests made by a single source within the specified time interval to classify whether a TCP SYN packet is malicious or not using the adaptive threshold generated by the algorithm. The detection module analyses the source IP address and number of requests made by the source. If the number of requests within a given interval of time exceeds than that of the threshold the detection mechanism classifies this as an anomalous request. As it is known, getting thousands of TCP-SYN requests within a minute from a single IP source is not usual.

4.1.1 Algorithm for detection of TCP-SYN flood attack.

The algorithm for detection of TCP SYN flood attack using the adaptive threshold value generated by adaptive thresholding algorithm and uses technologies like scapy and python to detect the ongoing attack is shown in Fig.4. It actively analyses the incoming TCP packets to the system to classify whether the packet is malicious or not.

1. Start the detection module.
2. Get the adaptive threshold value from the adaptive threshold algorithm.
3. Sniff the incoming packets using scapy.
4. Check for the packet protocol.
    1. If the packet has a TCP layer, proceed with the next step.
    2. Else ignore the packet.
5. Check for the number of packets sent by the source IP address within the time interval.
    1. If the number of requests exceeds the threshold, proceed with the next step.
    2. Else ignore the packet.
6. Trigger the alert module to send the details of the attack to the user.
7. Trigger the prevention module to stop the attack.
8. Stop if the program is closed.

Fig.4 TCP-SYN Flood attack detection algorithm

4.2 Adaptive thresholding module

The adaptive thresholding module provides the threshold value for the detection module based on adaptive thresholding algorithm. The threshold provided to the detection module should be very precise if not it would affect the performance of the system. The value of threshold is set adaptively based upon the seasonal, monthly or daily usage where an estimate of SYN packets mean is computed from

recent measurements of network traffic. The threshold can also be calculate using the EWMA(Exponentially Weighted Mean Average) method .In Equation 1 ,yn is the count of SYN packets in the n-th interval of time, and $\bar{x}_{n-1}$ is the mean rate of the traffic computed from measurements prior to n, then the anomalous traffic condition is

If $y_n \geq (\beta + 1)\, \bar{x}_{n-1}$ then the request is malicious (1)

Where $\beta > 0$ is the percentage of packets that can be allowed greater than the adaptive threshold value before classifying it as an anomalous traffic.

4.3 Prevention Module

The prevention module plays a vital role in the architecture as it prevents the attack from happening further. The prevention module is triggered by the detection module when it finds any anomalous TCP requests. The prevention module uses IPtables for blocking the requests from the malicious IP addresses. IPtables is user-space utility software that empowers a system administrator to configure the rules of IP packet filtering using the Linux kernel firewall, implemented as different Netfiltering modules. The filters are categorized in different tables, which consist of chains of rules for how to deal with network traffic packets [17].This detection module gets an IP address as an input from the detection module. This malicious address is blocked by the module from performing any request to the host. This is done by a OS system call by the prevention module to the operating system to add the drop action rule for the malicious IP address using IPtables.

4.3.1 Algorithm for prevention of TCP-SYN Flood attack.

The algorithm which is responsible for the prevention of TCP SYN flood attack is shown in Fig.5. It uses the detected malicious IP address from the detection module to prevent the attack. It uses technologies like python and IPtables to block the malicious requests.

1. Start the prevention module.
2. Check for triggers from the detection module.
    a) If malicious activity detected continue with next steps
    b) Else ignore the next steps.
3. Get the IP address sent by the detection module.
4. Block the IP address by adding it to the iptables rules by setting option DROP.
5. Check for a stop signal.
    a) If found, proceed with the next step.
    b) Else repeat from step-2.
6. Stop the prevention module.

Fig.5 Algorithm for alert module of the proposed architecture

4.4 Alert module

The alert module proposed in this architecture provides a real time solution to avoid catastrophic results from the actual flooding attack and also prevents from the business loss due to false positive results from the detection algorithm. It uses Simple Mail Transfer Protocol (SMTP) for sending the alert to the system administrator in real-time of the attack. So that administrators can act accordingly to prevent from effects of SYN flooding attack.

4.4.1 Algorithm for alerting the TCP-SYN flood attack

The algorithm for the alert module used in the proposed architecture is shown in Fig.6. It uses the detected malicious IP address from the detection module to alert the user about the attack. It uses common technologies like python, Simple Mail Transfer Protocol (SMTP) for sending email alerts to the administrators.

1. Start the alert mail.
2. Configure SMTP and relevant details like receiver's address.
3. Check for the trigger from the detection module.
    a) If triggered, execute following steps.
    b) Else ignore next steps.
4. Get the details of the attack from the detection module.
5. Wrap the details as a message with the protocol header of mail.
6. Send the mail to the system administrator about the attack.
7. Stop the alert module.

Fig.6 Algorithm for alert module of the proposed architecture

4.5 Attack generation module

The attack generation module in the proposed architecture is responsible for creating a real world attack scenario. This helps us in testing our detection, prevention and alert modules. The attack generation module sends enormous number of TCP SYN packets to the detection machine for testing the effectiveness of the proposed architectures algorithm. Attack generation module gets the IP address of the victim's machine to flood the machine with thousands of SYN packets per second. Attack generation module uses hping3 to generate TCP SYN flood to the detection machine for testing.

5 Implementation

To implement the proposed architecture, simulation of a real world scenario of TCP SYN flood attack is to be created. The attack generation module is launched to flood the detection machine. After launching the attack hping3 would have started sending numerous amounts of TCP SYN packets to the detection machine. The

main program launches the execution of the detection, prevention, adaptive thresholding and alert module. The adaptive thresholding module returns the threshold value to the detection module. Now the detection module analyses the packet flow in the network and stores the information and counts the number of requests made by a single source. The detection module could detect the anomalous request being made by the hping3 tool by verifying with the thresholding amount and the number of requests made by a single source IP. Now this detection will trigger the prevention module. The prevention module works based on IPtables. It gets the malicious IP address as a input from the detection by module. Then this IP address is blocked by the prevention module by adding drop IPtables rule by invoking a system call to the operating system. This prevents the TCP-SYN flood attack from happening again from the malicious IP address. This prevention module triggers the alert module that sends the details of the attack to the system administrator via Email in real time. Then the administrator can act accordingly to the attack to avoid prevention of false positive results. In this way the TCP-SYN flood could be detected and prevented.

6 Results

The proposed architecture's experimental results and analysis of the TCP SYN flood attack detection and prevention system is stated in the below section. From the above modules of the architecture we can infer that.

6.1 Attack generation

Attack generation phase simply involves running the hping3 tool residing in the attacking module. This hping3 takes in the IP and port of the target machine along with the number of packets which the target has to be flooded with. As soon as the attack was started, the packets started getting produced and sent to the target machine.

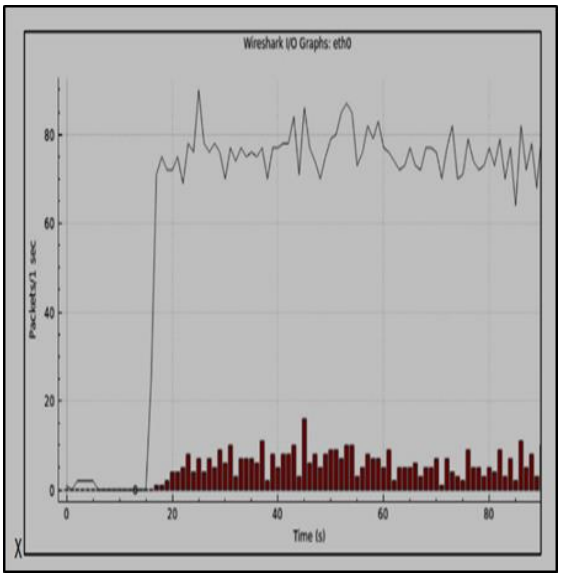

Fig.7 Attacker's network traffic graph during the SYN flood attack.

The I/O network graph of the attacker's machine during the TCP SYN flood attack is shown in Fig.7 which clearly states that the number of packets sent per second increases to an abnormally large value on the eth0 interface. Also from the rapid spike in the graph at around the 15 second mark as compared to the normal traffic which was captured before the attack was generated.

6.2 Detection and Prevention

The efficiency of this module is stated in terms of the CPU workload of the system before attack, in course of the attack and after detection of the attack. The CPU workload before attack, during attack and after TCP SYN flood detection and prevention by blocking the malicious IP address is shown in Fig 8 .The value of the of the CPU load under normal condition before the attack ranges from 7-12. The CPU load value observed during the attack ranges from 96-100 because of the malicious incoming requests. After detection of the attack the CPU load ranges from 7-13 which is similar to the CPU load of the system before the attack. Figure below clearly depicts that this system is capable of detecting and preventing TCP SYN Flood attack efficiently which also reduces the CPU's workload after detecting TCP SYN flood attack and restores the system back to the normal state prior to that of the attack.

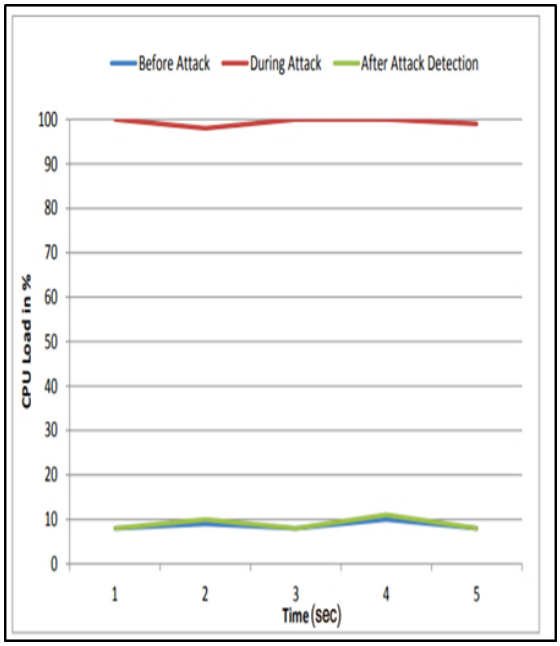

Fig.8 CPU Load of the detection module

The time taken by our proposed algorithm to detect and prevent TCP SYN flooding algorithm under variable SYN requests per seconds is shown in Table 1 when threshold is set to 300 SYN requests. So we can see the detection time reduces as the number of SYNs per second increases. This clearly states that our algorithm can effectively detect large SYN flood attacks.

Table.1 Time taken for detecting the SYN flood under different SYN rates

SYN s/ s	Detection Time
35	8.591
50	6.011
60	5.750
80	3.750
100	3.132

IPtables rules are updated after the detection of the SYN flood as shown in Table 2. The prevention module adds the malicious IP addresses to the IPtables which helps in preventing the attack happening further. The packets further after detection from the malicious source is dropped by the IPtables which prevents the TCP-SYN Flood attack.

Table.2 IPtables rules after prevention of the TCP SYN Flood attack.

Num	Target	Prot	Opt	Source	Destination
1	DROP	all	--	192.168.229.129 (attacker IP)	anywhere
2	DROP	all	--	192.168.229.234 (attacker IP)	anywhere

The results shows that our algorithm is very effective and efficient for detecting and preventing TCP-SYN Flood attacks at a higher rate also the performance of the algorithm can be increased to great extent by introducing and tuning various thresholding parameters according to the scenario.

7 Conclusion and Future Works

In this paper, the proposed architecture is capable of an effective and efficient detection and prevention of SYN flood attacks. It consists of four modules detection, adaptive thresholding, prevention and alert modules and the relevant metrics (IP source, number of requests made, time of previous request and SYN arrival rate). The detection module is efficient because it requires less computation and memory resources of the system .The adaptive thresholding module increases the efficiency of the detection module by providing a precise threshold. The prevention module works well in preventing the attack. Using this result, the proposed architecture described an attack detection and prevention method taking the IP address and number of requests made by a IP source as a metric to classify the incoming traffic .The results states that our proposed architecture can effectively detects and prevents SYN flooding attacks faster and precisely.

The future works are to optimize and add some additional parameters (e.g., variance, standard deviation and other statistical and network parameters) to the adaptive thresholding algorithm so the detection module can get more precise threshold values for detection hence improving the performance of the module. Also the proposed architecture can be extended for other denial of service attacks (HTTP Flooding, ICMP Flooding, UDP flooding) [20] using this adaptive threshold based detection algorithm.

8 References

1. Ohsita .Yuichi, Ata. Shingo, Murata, Masayuki, Detecting Distributed Denial-of-Service Attacks by Analyzing TCP SYN Packets Statistically, Ieice Transactions, Vol.4. (2006).
2. Wang, Haining, Zhang, Danlu, Detecting SYN flooding attacks. Proceedings of IEEE INFOCOM. Vol **3**, (2002).
3. Schuba. CL, Krsul. IV, Kuhn. MG, Spafford. EH, Sundaram. A, Zamboni D, Analysis of a denial of service attack on TCP. In Proceedings of the IEEE Symposium on Security and Privacy, (2017).
4. Blazek. RB, Kim. H, Rozovskii. B, Tartakovsky. A, A novel approach to detection of denial of service attacks via adaptive sequential and batch sequential change point detection methods. In Workshop on Information Assurance and Security, IEEE, June (2001).
5. Jin. S, Yeung. DS, A covariance analysis model for DDoS attack detection. In IEEE International Conference on Communications, Vol. **4**, 1882–1886, (2004).
6. Siris. VA, Papagalou. F, Application of anomaly detection algorithms for detecting SYN flooding attacks. In IEEE, GLOBECOM, December (2004)
7. D. Kshirsagar, S. Sawant, A. Rathod, S. Wathore, CPU Load Analysis & Minimization for TCP SYN Flood Detection ,Procedia Computer Science 85, (2006).
8. K. Pai, N. HR, A. Bhat, "Detection and Performance Evaluation of DoS/DDoS Attacks using SYN Flooding Attacks, International Journal of Computer Applications, (2014).
9. Mirzaie. Sara, Elyato. Alireza , Sarram. Mehdi Agha, Preventing of SYN Flood Attack with IPtables Firewall, ICCSN, (2010).
10. Bellaiche. Martine, Grégoire. Jean-Charles, SYN flooding attack detection by TCP handshake anomalies, Security and Communication Networks, Vol **5**, (2012).
11. Nakashima. T, Sueyoshi. T, Performance estimation of TCP under SYN flood attacks, First International Conference on Complex, Intelligent and Software Intensive Systems, (2007).
12. K. Lu, D. Wu, J. Fan, S. Todorovic, A. Nucci,Robust and efficient detection of DDoS attacks for large-scale Internet, Comput. Netw., vol. **51**,(2007).
13. K. Lee, J. Kim, K. H. Kwon, Y. Han, S. Kim, DDoS attack detection method using cluster analysis, Expert Syst. Appl, vol. **34**, (2008).

14. W. Wei, F. Chen, Y. Xia, G. Jin, A rank correlation based detection against distributed reflection DoS attacks, IEEE Commun. Lett, vol. **17**, Jan, (2013).
15. Haris. S, Ahmad, R. Badlishah, Abd. Ghani, Mohd, Waleed, Ghossoon, TCP SYN flood detection based on payload analysis, SCORED, (2010).
16. Rohith, R. Yadav, Rohith, Moharir, Minal, Shobha. G, SCAPY- A powerful interactive packet manIPulation program, (2018).
17. Othman, Mohamed, Kermanian, Mostafa, Detecting and preventing peer-to-peer connections by Linux IPtables, ITSIM, (2008).
18. Kaur. H, Behal. S, Kumar. K, Characterization and comparison of distributed denial of service attack tools (IEEE), (2015).
19. Banday, M. Tariq, Qadri, Jameel, Shah, A Practical Study of E-mail Communication through SMTP, (2010).
20. Zlomislic, Vinko, Fertalj, Krešimir, Sruk, Vlado, Denial of service attacks: An over-view. Iberian Conference on Information Systems and Technologies, CISTI, (2014).