International Journal of Cyber-Security and Digital Forensics (IJCSDF) 6(1): 14-22
The Society of Digital Information and Wireless Communications (SDIWC), 2017 ISSN: 2305-0012

HTTP Flood Mitigation Using Gateway Inspection and Second Chance Approach Analysis

Ismaila Idris, Ogidiagba Jones Tobi, Victor O. Waziri, John K. Alhassan, Ojeniyi Joseph.

ismi.idris@futminna.edu.ng, tobipriest@gmail.com, Victor.waziri@futminna.edu.ng, jkalhassan@futminna.edu.ng, ojeniyija@futminna.edu.ng

Department of Cyber Security Science,
Federal University of Technology, Minna, Niger State, Nigeria

Abstract— Nowadays, Internet is the main standard for communication which is being used by quantity of users through the world. At the same time, its profitable nature is producing increasing exposure to heighten cyber-attacks and there has been a massive escalation in the amount of DDOS (distributed denial of service attack) especially the HTTP Flood attacks on web application servers over the years. Network assets such as web servers, ftp servers, email servers, network bandwidth and network assets are frequently the targets of DoS attacks. In this research we focus on the HTTP Flood DoS attack, its characteristics, and the propose approach to mitigate by creating a model which we used in analysing http packets and we came up with appropriate results.

Keywords - DDOS attack; Mitigation.

I. INTRODUCTION

The Internet was created to ensure stress-free distribution of data between various connected system and networks nevertheless, it was created not knowing the security flaws present in it [1]. For example, digital viruses and related threats has been present ever since the internet was created. In the year 1988, when the internet was created (formerly called ARPANET) it consisted sixty thousand systems linked together and a malicious program called Morris Worm was created by attackers which made over ten per cent of those systems to breakdown by consuming their processing bandwidth [2]. Having this known, a lot of organizations still do not take proper measures to protect their infrastructures. However, with more than one billion users today, the Internet has become a channel for persons and companies towards frequently accessing information, perform tasks and services such as banking, online shopping, and social media therefore making the Internet vital area for business operation where a lot of revenue are generated [3]. One of the shortcomings of the internet is exposure to denial of service for users by attackers. An attacker would either steal data or attempt to end regular computer process which is motivated by business espionage, monetary achievement or political aims [4].

A Denial of service attack is an attack by which a malicious persons targets to interrupt a systems server operation via wide collection of various attack routes like the TCP, or HTTP attacks. Denial of service attacks is among the highest security threats affecting web application. Attack orchestrated by one system is called denial-of-service while attack orchestrated by many attacking machine, it is referred to as distributed denial-of-service [5].

In this research we center our focus on HTTP-Flood DoS attack type, its characteristics, how it carried out and the approach to mitigate in order to decrease its effects.

A. WHAT IS A DOS/DDOS ATTACK?

In this attack, the target system is bombard with a gigantic quantity requests (or data) thereby draining the targets process resource and denying authentic users from normal operations. Basically, in Denial of Service attack an attacker employ just one system resources in performing denial of service of its targets, in a purpose to prevent it from working normally [6]. Large web servers have high capacity to prevent the Denial of Service attack from one system. Nevertheless, the attackers frequently carry out distributed denial of service attacks, which use many systems for amplified success [7]. Therefore special defences are essential to notice and counter such massive attacks [8]. Furthermore, attackers often control the attacking system illegally by infecting large amount of systems via the internet using malicious software in an aim to have illegal access to the systems [9]. A group of thousands of hacked systems acting as a mass with the control of one attacker can be referred to as a botnet. Most time the real owners of machines which are among the botnet mass are ignorant of the fact of their systems are being used to coordinate distributed denial of service attacks.

B. THE BOTNET

The attackers succeeds to build large collection of systems (called zombies) having control over them, there are two ways to hijack control over peoples system, is either by indirectly planting malware to hack the systems of ignorant users or by soliciting for volunteers accepting to use Denial of Service software.

Taking about the former instance, attackers will invent or buy from numerous secretive cybercrime forums experts on malicious software; they then distribute them to a lot of weak systems available. The numbers of users tricked into activating such software will repeatedly deactivate their antivirus program, which makes it easy to create an entrance point. By this the infected system can get commands from the attacker to send a massive distributed denial of service requests to the targets. In summary all the attacker do is to send instructions to its collection of infected systems telling them the exact targets to send their malicious requests.

In this aspect in which various systems are willingly acting in agreement, the hackers supporting an attack will broadcasts their intents on social media website or an Internet Relay chat network, containing full information of how the attack would be coordinated in order to solicits for volunteers [2].

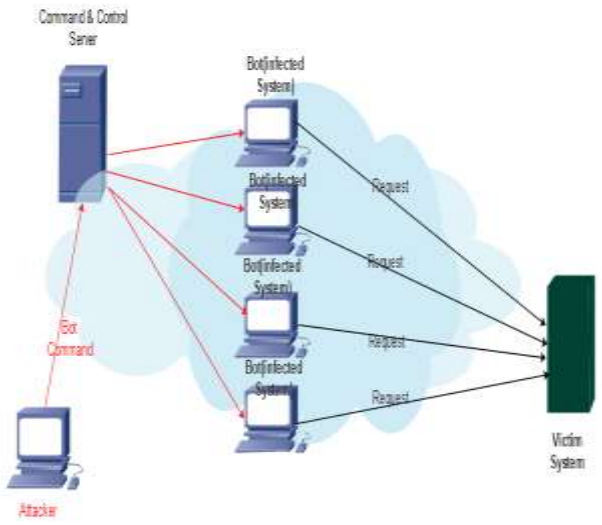

Figure 1: How a Botnet Operates

C. THE TYPES OF DDOS ATTACK

Note that, attackers never take the threat of missing their objects whenever they have dedicated to it; they always adjust their attack vectors so they can attempt to dodge Defense procedures that are available. Most recent attacks usually use numerous vectors in just one attack campaign, aiming various organisations cyber infrastructures. Recent statistics shows that 56% of cyber-attacks were directed at applications with HTTP protocol having the highest (21%); 46 % at the network. The rates keep increasing daily with attackers using more than 5 various vectors in just one campaign [2].

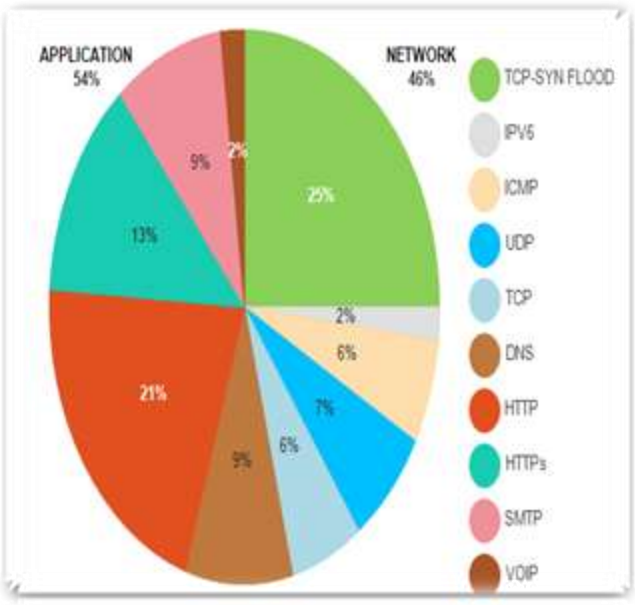

Figure 2: Statistics Showing Recent DDoS Attack Percentage

Note Attacks does not consume network bandwidth only, nevertheless in certain cases s application process bandwidth as well. Therefore denial of service and distributed denial of service attack can be categorizing into several kinds depending on the unique features that may possess. In nut shell, the categories of attacks comprise those that consumes network bandwidth, those that consumes server, and those that consumes application bandwidth. But our main focus is those that consume application bandwidth. Specifically that of HTTP Flood attack [10].

1) ATTACKS THAT CONSUMES NETWORK BANDWIDTH

The attack that consumes network bandwidth tries to drain the whole of the victim network resource using an enormous volume of malicious packets to flood the corporation's network links. Such an attack is referred to as network floods. In this saturating attack, the attacker sends command to huge amount of voluntary or compromised computers that send a gigantic numbers of requests to the victim's site, crushing down the network. Though packets of this attack could appear genuine in lesser quantities but in large quantities they are extremely harmful. Genuine user attempting to access the network would find very difficult. Examples of these attacks are UDP Flood, ICMP Flood, and IGMP Flood [10].

2) ATTACKS THAT CONSUMES SERVER BANDWIDTH

These Attacks tends to drain server bandwidth so as to disrupt a server's operation power, possibly triggering a service denial situation. In this category the attacker take advantage of the present vulnerability on the victim's server or flaws in the seating protocols in an aim to cause the victims server to become busy settling malicious requests which leads to it not having resources to respond to legitimate ones. Common examples of these kinds of attacks are TCP/IP SYN flood attack, TCP/IP RST attack, Low and Slow attack, Sockstress attack, SSL-Based attacks

3) ATTACKS THAT CONSUMES APPLICATION BANDWIDTHS

In recent years, Denial of Service attacks that attacks application bandwidth keeps growing enormously. It is extensively employed by a lot of attackers [11]. The Hypertext Transfer Protocol is the most common protocol attacked (and other application protocols).

a) HTTP FLOOD

In recent years, the Hype Text Transfer Protocol flood is the most widespread application layer distributed denial of service attack.

The HTTP flood attack is principally the most common non-vulnerability attack disturbing web servers today which is actually difficult for security parameter devices to differentiate authentic hypertext transfer protocol traffic and malicious ones [12]. For this reason, it is required to use a number of security parameters. Denial of Service attack on web services are called Hyper Text Transfer Protocol flood. Typically, in this type of attacks, mischievous attacker sends a huge amount of requests to the victim's server spontaneously. Meanwhile this attack requests have genuine formats making it difficult for other security device like intrusion prevention system to identify them.

Therefore, since this protocol functions at the seventh layer of the Open System Interconnection (OSI) Model, then only security parameters like Firewalls, IDS/IPS can inspect and analyse or examine the packets.

The security parameters that do not function at the seventh layer have no capability to inspect and analyse this flood attacks but they can only prevents or be blocked on these different layers of OSI model aside from the seventh layer.

Presently it has been observed that in the actual world that this attack is not mitigated appropriately. Of most of them is as a results of the security settings flaw of the devices while some is because of no security device. Employing security measures at different level of data which data passes through would be a lasting solution to this type of attack.

Now, this is where the seventh layer phase protection becomes essential because application seventh layer solution operates in the application it is defending.

The web service is what makes the web application tool to function that is, without the web service; the web application cannot be accessed. So before an attack is called hypertext transfer protocol flood, the Transport Control Protocol packet which transfers http request data must have been interpreted by the web service. The flood attack begins starts at the web service and its backend set-ups or assets. Therefore any flood attacks that did not make it to the web service level, is nothing but a Transport Control Protocol denial of service attack that consumes the network bandwidth.

II. RELATED WORK

Different models, algorithm, methods and techniques have been proposed and used by authors to detect HTTP flood attacks at the application layer.

[13] The Chi- square method uses the distribution of comparison measurements of HTTP packets size values. The rule here is that the expected rate of packets in a sample has possible value of at least seven. This is done by creating a set of range of the possible values. In the Chi-square method an attack is identified if it has an unusual high chi square statistics.

Firstly, the problem of this method is in the assignment of values to the different ranges which normally changes at each new chi square computation. Secondly, this method can lead to many false-positives.

[14] propose three ideas which causes a blockage at the website phase to counter the "HTTP" flood attacks, the fundamental concept are (1) Detect Internet Protocol (IP) address of malicious requests using a standard rule (2) in-order to lessen attack effects, reply the malicious requests with a little resource reply like an empty page or Page not found) (3) Dis-allow malicious IP addresses through other security parameters at the other mitigation phases like Firewall, web-server and services. The first could have an issue as some legitimate IP addresses may have a slightly abnormal formats and when blocked would lead to a lot of False-Positives.

A different mitigation method against application-layer phase denial of service attacks is called the CAPTCHA; this approach is in form of task response test used in figuring if a request and reply is made by a real person and not by a robot [15].
An operator must effectively answer a CAPTCHA text in-order to creating a connection with the host system. Nevertheless, this approach takes the resulting setbacks. Firstly, patience of the users as numerous information has shown that these quizzes irritate the users because they often not user-friendly [16]. Because a lot of users have low endurance to answer the CAPTCHA quiz and also wait for reply therefore a system which uses the CAPTCHA possibly will chase away genuine users. Secondly, cracking procedures: nowadays, several techniques have been invented to crackdown this approach [17]. Thirdly Uncertain implementation meaning that certain CAPTCHA security systems can be dodged without using Optical-character-recognition basically by reusing the session Identification of a famous images of CAPTCHA. Fourthly, the Labour attack which some information specifies that there are free or low-cost third-party human labour to crack CAPTCHAs. As a results of the setbacks which we estimated for the CAPTCHA methods, scholars have make an effort to resolve application phase layer DDoS attack minus CAPTCHA [18].
[19] Recommends two basic approaches. Firstly, once there exist attacks from hacked clients with bot then the server would detect the surfing order of pages repetitively at the server. Secondly, attackers surf a web page for a smaller timing than regular users; in this manner if a user surfs a web-page in less timing than the threshold timing then it is identified as malicious one. The first approach will have an as the attacker can make compromised systems to direct requests for different pages. Also the second approach will also have problems in which the attacker can surf a web page for a lengthier time and exceeds the threshold.
[20] Categorizes attacking speed into two classes: probable speed and no probable speed. Probable speed includes constant rate, monotonically increasing rate and periodical rate. However, Non-predictable rate has no classification.

The writer further recommends the Pearson-correlation coefficient theorem in order detect probable rates for the three classes. However, this has no fulfilment once the attackers direct requests at different non predictable ranges. [21] Suggests that regular users constantly access web pages serially on the hyperlinks arrangements, whereas most attackers always would not follow this organization and access different web pages using their links directly. So, the authors distinguish attackers over the entropy-test.

Although, the first assumption is true, but the second assumption which is the base of the procedure is not always true. Therefore we identify that an attacker can effortlessly build a tool and ask compromised systems to visit web pages using the hyperlink organization. In this case, the entropy value of attackers and normal users locate in the same range and the server cannot detect zombie machines.

III. RESEARCH METHODOLOGY

A. INTRODUCTION

This research is aim at developing a HTTP flood attack mitigation approach using gateway inspection, and analysis of packets at the application level of the open system interconnection level. This research will study the characteristics of packets that comes into the web application and to differentiate between the normal packets, the abnormal packets and the most dangerous packets thereby creating a packets status indicator which gives the range each categories of packets falls into ;also to create suitable rule that prohibits the entrance of abnormal and danger packets; design a structure that defines the path at which every user requests must follow; to reduce the amount of False-positives in the mitigation approach and lastly, to use this approach to study the packets that goes into the web application of other enterprise.

B. MITIGATION LEVEL AGAINST HTTP FLOOD

The key mitigation levels against the HTTP-flood attacks are [14]:

- Phase one is the Cloud Services Level (Internet Service Provider)
- Phase two is the Network level (web application firewall)
- Phase three is the Web Server Level (host Intrusion Prevention System)
- Phase four is the Web Service Level (Dynamic IP Restrictions),
- Phase five is the Web Application Level (DDoS mitigation against HTTP Flood.

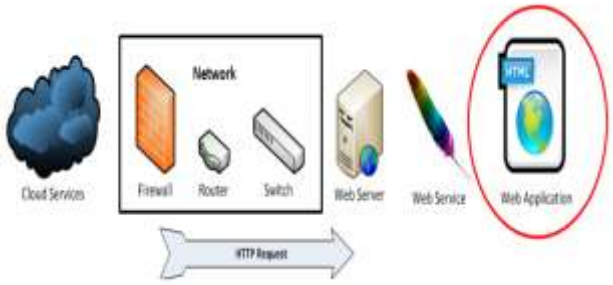

Figure 3: Mitigation phases against HTTP-flood Attack.

C. CHARACTERISTICS OF PACKETS FOR CREATING RULES

The first step to take to achieve the appropriate rule base against HTTP flood attacks is the defining of normal traffic. You can know normal traffic by studying or analysing different traffic that enters your network on normal situations and getting the average appearance of them.

The normal network traffic values on the web application are given below:

- Maximum request count from a single IP address: five requests per second,
- The time between the closest two requests from one IP address is 0.2 seconds.

Note: The above normal network values will be our standard values for creating our rule.

These are the standard traffic values for forming a rule against the HTTP-flood attacks.

The dangerous point for the web application phase HTTP flood attack mitigation is the false positives. To avoid false positives, the detection rules must be properly defined and be confirmed with the real world traffic usage scenarios. Also a good understanding for the rule creation concept is highly suggested.

D. MATHEMATICAL MODELS

1) PARAMETER DEFINITION

- Flood Rate (Maximum) = Fmax (byte/sec)
- Number of Request per second from an IP Address = X (byte)
- Time Taken between two closest request = t (second) set to be 0.2seconds.

2) MODEL EQUATION

For every IP request (no of requests (X)) at time (t) in seconds, the Maximum Http flood is defined as:

$$F_{max} = \frac{X\ (byte)}{t\ (second)}$$

3) LEGITIMATE PACKETS

A model of "No Alarm State" that is Legitimate Traffic given that:

$$F_{max} = \frac{X\ (byte)}{t\ (second)}$$

Therefore;

For all Fmax ≤ 25 byte/seconds, Raise no alarm that is consider traffic legitimate and grant access to request.

4) ABNORMAL PACKETS

For all 25 < F <50 byte/seconds, Raise alarm and redirect traffic for further analysis.

5) DANGER PACKETS

For Fmax ≥ 50 byte/seconds terminate connection and raise alarm for intense analysis.

In summary;

$$Packets = \int_{if\ F>25\ then\ its\ Abnormal\ or\ Danger\ If\ F>50}^{if\ F<25\ then\ it\ is\ legitimate}.$$

6) PACKET STATUS INDICATOR

The packet status indicator show the three categories every packets that makes a requests can fall into. The first category is the "Legitimate" packets; the second category is "Attention" packets while the third category is the "D anger" packets.

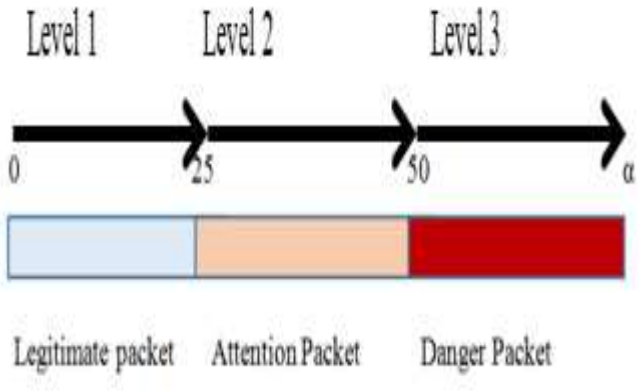

Figure 4 Packet indicators.

E. MITIGATION FLOW-CHART

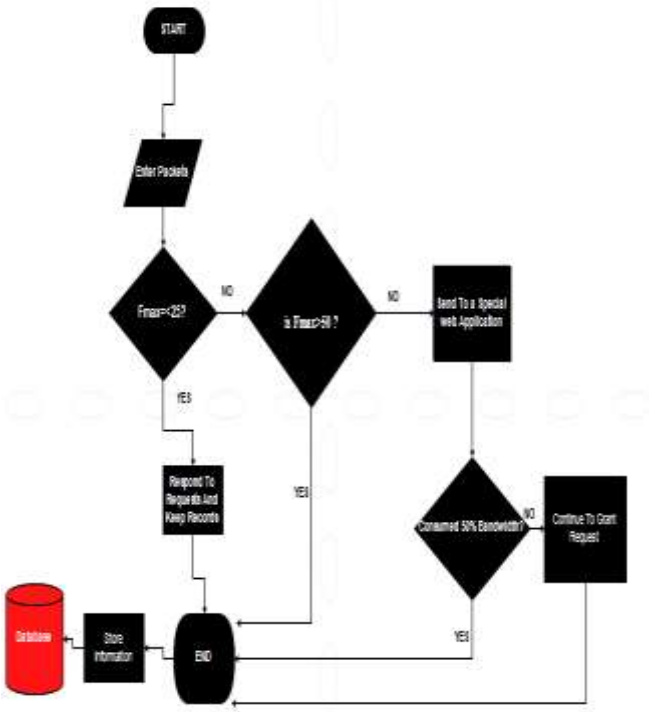

Figure 5: Flow Chart illustrating the inspection/Analysis Process

F. MITIGATION STEPS

- Enter Packets
- Inspection of the packets
- Does the packet meet the pre-defined rules?
- If yes, then respond to request
- If No, then what range on the packets category value does the packets fits?
- Does it fall between 51 and above (that is 50 < Fmax≥ infinity) if Yes then Terminate IP address OR Does the packet falls between the value of 26 and 50 (that is 25< Fmax≤ 50) then send the suspicious packet to a different web application and raise an alarm to the administrator to observe the behaviour effect of the packets on the special web application
- Is the suspicious packets consuming above 50% of the system resource?
- If yes then terminate the IP addresses sending such packets and if not then continue to respond to requests.

G. ANALYSIS OF PACKETS THAT FALLS IN LEVEL 2 AND LEVEL 3 (SECOND CHANCE APPROACH)

This approach will further help reduce False-positives as some legitimate packets might have characteristics of an abnormal traffic.

Firstly, any packets from an IP address that falls in level two is linked to another web application to respond to the request having raise an alarm to the administrator(s) to analyse the health of that application. The health of this special application is measured according to the resource or bandwidth spent (from 0 to 100%).

If the health of the web application goes above the 50% it clearly states the packets is a HTTP flood packets, therefore any request from such IP address should be terminated.

Secondly, information of such packets should be recorded in database for future prohibition.

Thirdly, any packets that falls in level 3 is terminated immediately as it signals “Danger” and every data (that is, the number of packets per seconds, the time between two closets packets and type of requests) is recorded in the database.

IV. RESULT AND DISCUSSION

A. PACKETS ANALYSIS

We were able to gather and analyse the HTTP packets of different IP address on the internet accessing a web site and we were able to categorize the packets into the legitimate packets, the abnormal packets and the danger packets using our standard rule base model. We got our packets capture from http://chrissanders.org/packet-captures/

Also in order to analyse the abnormal packets we used cloud shark on line tool to further analyse suspicious (remember no packets is tagged malicious until they portrays the characteristics of a malicious packets). We checked the rate of the packets (increase with time) and the size of the packets. The following is our findings from the analysis;

B. POSSIBLE TABLE FOR FINDINGS USING OUR MODEL

For time t = 0.2 seconds and a rate of 5 packets per seconds the following was collected

Table 4.1 Showing Information of Packets

Time	Address	Request/s	F_{MAX}	Action
0.071616	172.16.0.122	3	41.8	Abnormal
0.219233	172.16.0.122	5	22.8	Legitimate
0.358288	172.16.0.122	7	19.2	Legitimate
0.391966	205.234.218.129	6	15.3	Legitimate
0.413159	205.234.218.129	2	4.8	Legitimate
0.479647	68.71.208.11	5	10.4	Legitimate
0.590030	172.16.0.122	8	13.5	Legitimate
0.358303	172.16.0.122	4	11.2	Legitimate
0.590044	172.16.0.122	5	8.5	Legitimate
0.390053	172.16.0.122	7	17.9	Legitimate
0.590061	172.16.0.122	10	16.9	Legitimate
0.140994	199.181.132.250	8	56.7	Danger

From the table above we were able to study the packets and using our standard rule to identify the legitimate packets and the abnormal packets. The colour of each cell indicates the status of the packets from the IP addresses.

*C. ANALYSIS OF "*172.16.0.122*" PACKETS THAT FALLS IN LEVEL 2 (SECOND CHANCE APPROACH*

From the table above, the IP address "172.16.0.122" has abnormal characteristics because it violates our rule standard (that is Fmax is 48.1), so since it falls within the range of "25 < F <50" according to our "Abnormal Packets model" we therefore proceed for further analysis.

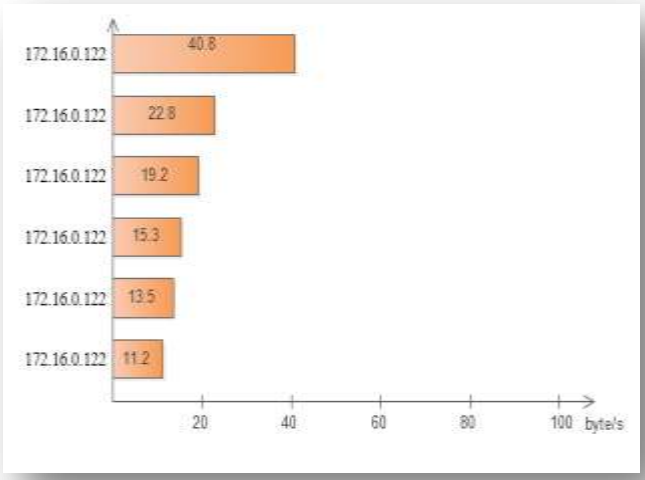

Figure 6: Showing the gradual change of the flood rate of a 172.16.0.122.

From Figure 6 above we noticed from analysis that with time, the time between closest packets from the same IP address "172.16.0.122" reduced with time subsequently.

D. POSSIBLE TABLE FOR FINDINGS USING CHI-SQUARE MODEL

Table 4.2 Showing Findings Using Chi- Square Model

Time	Address	Request/s	X^2	Action
0.071616	172.16.0.122	3	5.30	Legitimate
0.219233	172.16.0.122	5	0.80	Legitimate
0.358288	172.16.0.122	7	0.00	Legitimate
0.391966	205.234.218.129	6	0.20	Legitimate
0.413159	205.234.218.129	2	12.5	Legitimate
0.479647	68.71.208.11	5	0.80	Legitimate
0.590030	172.16.0.122	8	0.13	Legitimate
0.358303	172.16.0.122	4	2.25	Legitimate
0.590044	172.16.0.122	5	0.8	Legitimate
0.390053	172.16.0.122	7	0.00	Legitimate
0.590061	172.16.0.122	10	0.30	Legitimate
0.140994	199.181.132.250	8	0.13	Legitimate

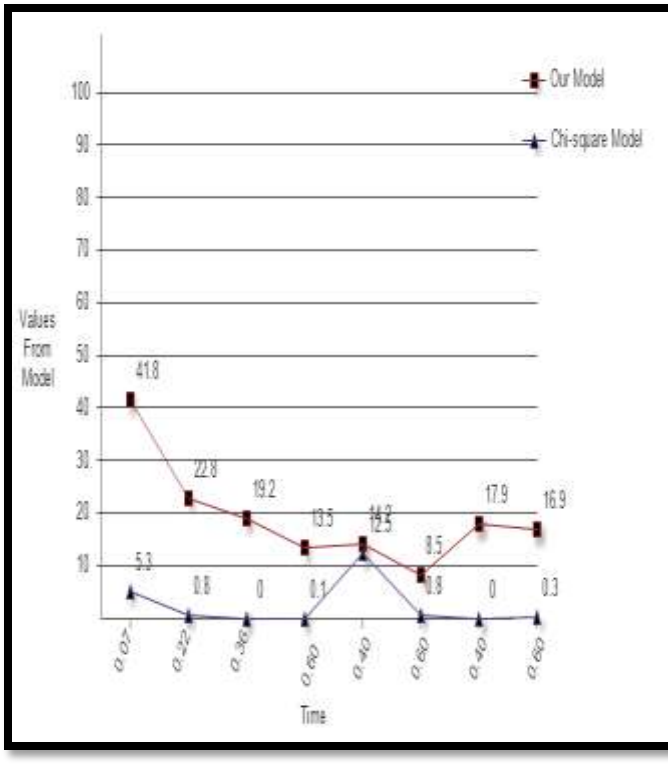

Figure 7: A line graph showing the Comparison between our method and Chi-square method.

E. MODEL PERFORMANCE AND EVALUATION ANALYSIS TABLE

Table 4.3 Model Performance and Evaluation Analysis

Approach	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)
Our Method	**92.31**	**100.0**	**100.0**	**100**
Chi-Square	**91.66**	**91.67**	**91.66**	**NA**
SNORT	**93.00**	**92.00**	**NA**	**NA**

V. CONCLUSION AND RECOMMENDATION

A. CONCLUSION

In this research we have been able to develop a solution, a mathematical model that solves HTTP Flood rule definition technique. The outcomes of our approach are promising. Our mitigation approach focused on studying the characteristics, features or behaviour of packets, so as to differentiate between the normal packets and the malicious or abnormal packets. We went further creating a standard rule that determines which packets is allowed entrance to the either of both web applications depending on the category which an incoming packets fits in-terms of their values. With this we were able to reduce false-positives. To wrap things up we tested this approach using real life scenarios.

B. RECOMMENDATION

Considering all that has been researched, I would like to emphasize that the adoption of this new mitigation approach will not be regretted as it would rather be a help to reduce the HTTP flood attack to a minimal level. Also this approach can be automated and embedded with a web applications system.

REFERENCES

[1]. Shinder, L. and M. Cross (2008). Chapter 2 - The Evolution of Cybercrime. Scene of the Cybercrime (Second Edition). Burlington, Syngress**:** 41-75.

[2]. Ronen Kenig, (2013). DDOS Survival Handbook. Pp 5-25

[3]. Apăvăloaie and Elena-Iulia (2014). "The Impact of the Internet on the Business Environment." Procedia Economics and Finance **15**: 951-958.

[4]. Silva, S. S. C. (2013). "Botnets: A survey." Computer Networks **57**(2): 378-403.

[5]. Mansfield-Devine, S. (2015). "The growth and evolution of DDoS." Network Security **2015**(10): 13-20.

[6]. Kalkan, K. and F. Alagöz (2016). "A distributed filtering mechanism against DDoS attacks: ScoreForCore." Computer Networks **108**: 199-209.

[7]. Chen, C.-M. and H.-C. Lin (2015). "Detecting botnet by anomalous traffic." Journal of Information Security and Applications **21**: 42-51

[8]. Gross and Garrett (2016). "Detecting and destroying botnets." Network Security **2016**(3): 7-10.

[9]. Contos and B. T. (2007). Chapter 1 - Cyber Crime and Cyber Criminals 101. Enemy at the Water Cooler. Burlington, Syngress**:** 3-47.

[10]. Qijun Gu, P. and P. Peng Liu (2011). Denial of service Attack. Pp 1-24

[11]. Kostadinov, D. (2013). "Layer Seven DDoS Attack." from http://resources.infosecinstitute.com/layer-seven-ddos-attacks/

[12]. Muharremoğlu, G. (2014). "Web Application Level Approach against the HTTP Flood Attacks IOSEC HTTP Anti Flood/DoS Security Gateway Module." IOSEC HTTP Anti Flood/DoS **1**: 5.

[13]. Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, Darrell Kindred (2012) “Statistical Approaches to DDoS Attack Detection and Response" Proceedings of the DARPA

Information Survivability Conference and Exposition (DISCEX'03)

[14]. Hally Khatri, A. G., Dheeraj Pal (2014). "Mitigation of HTTP-GET flood Attack." International Journal for Research in Applied Science & Engineering Technology **2**(Issue XI, November 2014): 450-453

[15]. W. G. Morein, A. S., D. L. Cook, A. D. Keromytis, V. Misra, D. Rubensteiny (2013). "Using Graphic Turing Tests To Counter Automated DDoS Attacks AgainstWeb Servers" Proceedings of the 10th ACM conference on Computer and communications security, Washington, DC, USA.

[16]. Caum, L. O. (2011). "L. O. Caum, Why is CAPTCHA so annoying?". from http://lorenzocaum.com/blog/why-is-captcha-so-fing-annoying

[17]. Mori, J. M. (2003). "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA." Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin.

[18]. D. Truong, C. F. T. a. C. C. Z. (2011). "iCAPTCHA: The Next Generation of CAPTCHA Designed to Defend Against 3rd Party Human Attacks." Proceedings of IEEE International Conference on Communications, Kyoto, Japan.

[19]. T. Yatagai, T. I., I. Sasase (2007). "HTTP-GET flood Attack Based on Analysis of Page Access Behavior." Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing: pp. 232–235.

[20]. Thapngam, T. (2011). " Discriminating DDoS Attack Traffic from Flash Crowd through Packet Arrival Patterns" Proceedings of 2011 IEEE Conference on Computer Communications Workshops: pp. 969–974.

[21]. Yu, X. a. (2009). "A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors." IEEE/ACM Transactions on Networking 17 **1**: 54–65.