INSTITUT SUPÉRIEUR D'ÉLECTRONIQUE DE PARIS (ISEP)

RESEARCH PROJECT PROPOSAL FOR IR. 2406
NETWORK SECURITY COURSE

DISTRIBUTED DENIAL OF SERVICE (DDoS): ATTACK (HTTP FLOODING ATTACKS AND COUNTER MEASURES) AND DEFENSE

BY

ABUBAKAR UMAR ELNAFATY (62717)
YAO YUAN (62848)
BUHARI ALIYU (62788)
LEIGHA IFIYEMI (62769)

SUPERVISOR: IDOWU AJAYI

MAY 2023

Contents

LIST OF FIGURES ii
1.0 INTRODUCTION 2
1.1 OBJECTIVE 5
1.2 MOTIVATION 6
1.3 CONTRIBUTION 6
2.0 REVIEW OF RELATED LITERATURES 7
3.0 EXPERIMENTS AND RESULTS 11
3.1 Network Environment 11
3.2 Attack Simulation 11
3.2.1 The Process 11
3.3 Defense Mechanisms 12
3.3.1 The Process 12
4.0 RESULTS DISCUSSION AND ANALYSIS 14
4.1 Attack Impact Analysis 14
4.2 Defense Mechanism Evaluation 14
5.0 CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK 15
REFERENCES 16

LIST OF FIGURES

Figure 1: Schematics of a DDoS Attack 3
Figure 2: HTTP GET/POST Request Attack 5

ABSTRACT

HTTP application layer Distributed Denial of Service (DDoS) flooding attacks pose a significant threat to the availability and performance of web applications and services. These attacks exploit vulnerabilities in the Hypertext Transfer Protocol (HTTP) to overwhelm the targeted servers with a massive influx of malicious requests, rendering them unresponsive to legitimate users. As the dependency on web-based services continues to grow, understanding the nature of these attacks and developing effective defense strategies becomes paramount.

The paper investigates the nature and impact of HTTP application layer DDoS flooding attacks and proposes effective defense mechanisms to mitigate their effects. This research aims to enhance the understanding of these attacks and contribute to the development of robust defense strategies. The study encompasses an experiment that simulates DDoS flooding attacks and evaluates the effectiveness of various defense mechanisms. The results highlight the severity of the attacks and the efficacy of the proposed defense mechanisms, providing valuable insights for network administrators and cybersecurity practitioners.

Furthermore, this paper investigates the GET/POST floods attack vector that is commonly employed in HTTP application layer flooding attacks. It explores the techniques used by attackers to evade detection and mitigation, including the utilization of legitimate user agents, IP spoofing, and encryption. The consequences of these attacks on organizations and end-users are also discussed, highlighting the financial losses, reputational damage, and potential regulatory implications.

Overall, this paper provides a comprehensive understanding of the nature of HTTP application layer DDoS flooding attacks, their impact on web applications, and the defense strategies employed to mitigate them. It serves as a valuable resource for researchers, security practitioners, and system administrators seeking to bolster the resilience of their web-based services. By fostering a deeper understanding of these attacks and their countermeasures, this paper aims to contribute to the ongoing efforts to combat the ever-evolving threat landscape of HTTP application layer DDoS flooding attacks.

1.0 INTRODUCTION

The Distributed Denial-of-Service (DDoS) attack is one of the age long Cyber attacks that has been around, with its first occurrence in 1996 when Panix (an Internet Service Provider [ISP] company) was knocked offline for several days by SYN flood [1]. DDoS attacks are a major threat to computer networks and hence all applications, hardware and other resources that use the network and its services.

These attacks attempt to make a machine or network resource unavailable to its authorized users. These attacks, including its threat actors, are increasing daily in leaps and bounds, even with the creation of bigger Botnets (the armies of hacked devices that are used to generate DDoS traffic).

Some of the most notable attacks include:

1. The Google Attack, 2020 [19]
   this attack saw a record -breaking User Datagram Protocol (UDP) amplification attack source out of several Chinee Internet Service Providers [ISPs]. This remains the largest bandwidth attack of which Google was aware of as of that time.

2. The Amazon Web Service (AWS) DDoS Attack, 2020. [19]
   The AWS was hit by a gigantic DDoS attack in February 2020, targeting an unidentified AWS customer. The attack lasted for three (3) and peaked at an astounding 2.3 tetra bytes per second.

3. The MIRAI KREBS Attack, 2016. [19]
   in 2016, the blog of Cybersecurity expert Brian Krebs was assaulted by a DDoS attack more than 620 Giga bits per second (Gbps) denoting the magnitude of the attack. The source of the attack was the Mirai Botnet which at its peak later that year, consisted of more than 600,000 compromised Internet of Things (IoT) devices such as Internet Protocol (IP) cameras, home routers and video players.

[18] DDoS attack is a coordinated attack that involves a multitude of compromised systems, called Botnets (many infected computers that form a network and are controlled by a master system remotely), attacking a single or more systems thereby denying service to users of the target system (s). The flood of incoming messages to the target system (s) essentially forces it to shut down, thereby denying service to legitimate users. The impact of DDoS attack include:

1. Loss of goodwill.
2. Disables network functions and services.
3. Results in financial losses.
4. It disables organizations.

How does DDoS attack work using Botnets?

The attacker initiates the DDoS attack by sending a command to Zombie agents, which are Internet-connected computers compromised by an attacker through Malware programs to perform various malicious activities through a command and control [C & C] server. These Zombie agents send a connection request to a large number of reflector systems with the spoofed IP address of the victim, which causes the reflector systems to presume that these requests originate from the victim's machine instead of the Zombie agents. Hence the reflector systems send the requested information to the victim. Consequently, the victim's machine is flooded with unsolicited responses from several reflector computers simultaneously, which may either reduce the performance or cause the victim's machine to shut down [19].

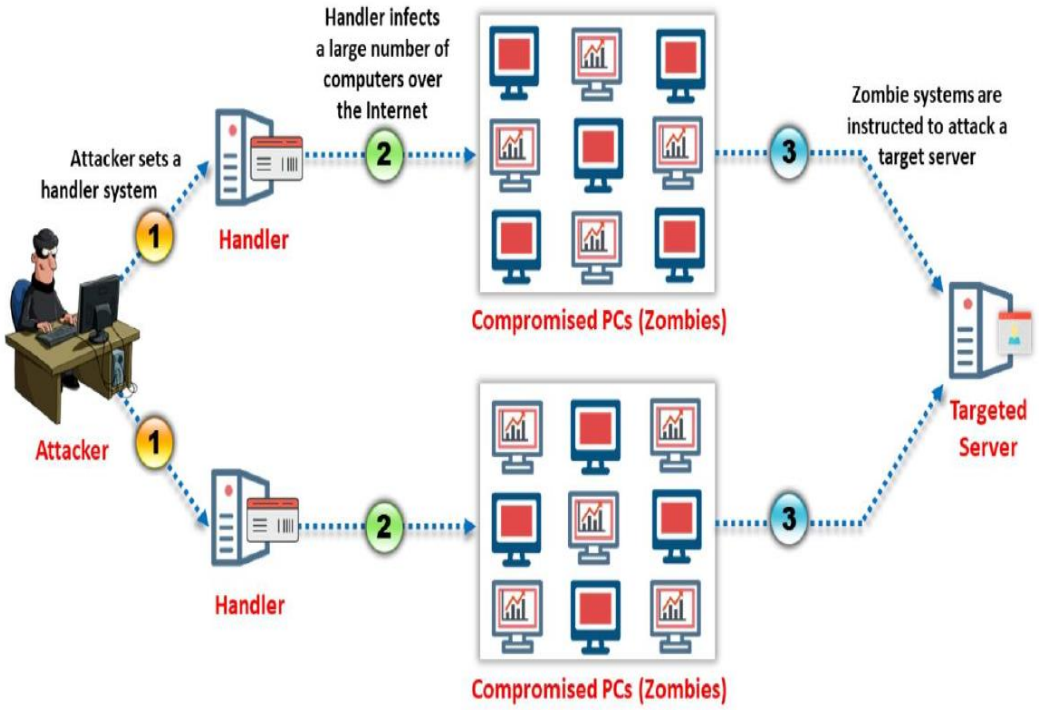

Figure 1: Schematics of a DDoS Attack

The different categories of DDoS attack vectors and their attacking techniques include [18]:

1. Volumetric Attacks
   These attacks exhaust the bandwidth either within the target network/service or between the target network/service and the rest of the Internet to cause traffic blockage, preventing access to legitimate users. The attack magnitude is measured in bits per second.

   Volumetric attack techniques include:

   * UDP flood attack.
   * Internet Control Message Protocol [ICMP] flood attack.
   * Ping of Death (PoD) attack.
   * Smurf Attack and so on.

2. Protocol Attacks
   Attackers can also prevent access to a target by consuming other types of resources other than bandwidth. These resources could be a connection state tables present in load balancers, firewalls, and application servers.
   These attacks exhaust resources available on the target or on a specific device between the target and the Internet.
   The attack magnitude is measured in packets per second (pps) or connections per second (cps).

   Protocol attack techniques include:

   * Synchronize (SYN) flood attacks.
   * ACK & PUSH ACK flood attack.
   * SYN-ACK flood attacks.
   * ACK flood attacks.
   * Spoofed session flood attacks and so on.

3. Application Layer Attacks
   In these attacks, the attacker attempts to exploit vulnerabilities in the application layer protocol or in the application itself to prevent legitimate users from accessing the application. In these attacks, the application layer or application resources are consumed by opening connections and leaving them open until no new connections can be made. The magnitude of the attack is measured in requests per second (rps). These attacks result in the loss of services of a particular network, such as emails and network resources or the temporary shutdown of applications and services.

   Application Layer attack techniques include:

* Hypertext Transfer Protocol (HTTP) flood attacks.
* Slowloris attacks.
* UDP application layer flood attacks.
* DDoS extortion attacks.

1.1 OBJECTIVE

The objective of our study is to practically demonstrate HTTP application layer DDoS flood attacks and mitigation measures, as well as present adequate knowledge of it so that our readers will be well informed about these attacks and the best ways and tools to defend against them.

Our major focus will be on HTTP flood attacks which occur in the application layer of the Open Systems Interconnection (OSI) model. In this case, HTTP clients, such as web browsers, connect to a web server through HTTP to send HTTP requests, which can either be HTTP GET or POST. Attackers exploit these requests to perform DDoS attacks.

In the HTTP GET attack, the attacker sends an incomplete header to the server and hence the server retains the HTTP connection and waits, making it inaccessible for legitimate users. In the HTTP POST attack, the attacker sends HTTP request with complete heard but an incomplete message body to the target web server or application. Because the message body is incomplete, the server waits for the rest of the message body, making the server unavailable to legitimate users.

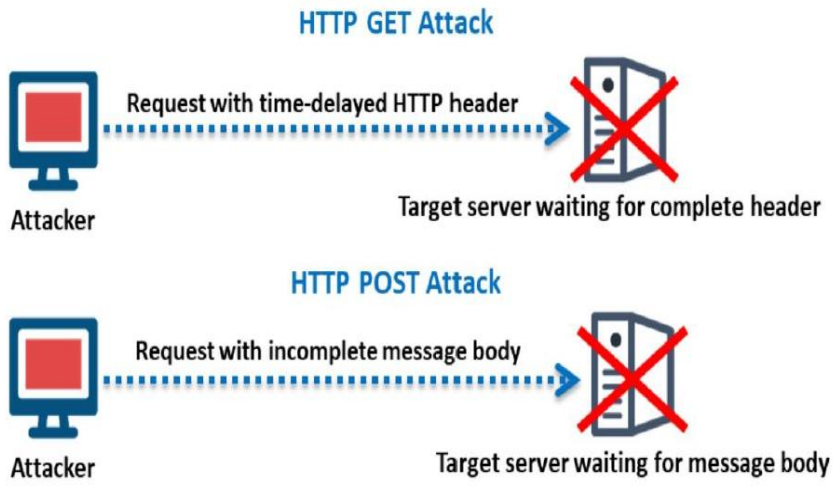

Figure 2: HTTP GET/POST Request Attack

In addition to the HTTP GET/POST attacks, attackers can employ the following HTTP flood attacks to exhaust the target network's bandwidth:

Single-session HTTP flood attack, single-request HTTP flood attack, recursive HTTP GET flood attack and random recursive HTTP GET flood attack [18].

1.2 MOTIVATION

Our motivation for this study is that as Cybersecurity Engineers in the making, it is our responsibility to proffer practical solutions to problems and to be of service to the larger society by providing relevant and practical information to the public about the current issues on security which affects all and sundry so that they can better be informed and protect their digital assets from loss.

1.3 CONTRIBUTION

Our contribution is to practically demonstrate HTTP application layer DDoS flood attacks and mitigation measures.

2.0 REVIEW OF RELATED LITERATURES

Hypertext Transfer Protocol (HTTP) flooding attack is a type of application layer distributed denial of service (DDoS) attack designed to overwhelm a targeted server, network bandwidth or other resources of the target system.

In this type of attack, the attacker floods the server with many illegitimate HTTP requests, making the server overwhelmed and unresponsive to legitimate requests. In this literature review, we will see the different ways that can be used to perpetrate HTTP flooding DDoS attacks and the counter measures that can be used to mitigate them.

According to research work done by [1], HTTP flooding application layer DDoS attack can be launched using techniques which include HTTP GET flood, HTTP POST flood, and Slowloris attack.

In an HTTP POST Flood attack, the attacker injects a large number of POST requests (requests that are used to send data to the server for processing and may modify the server's state) to the server, while in an HTTP GET Flood requests, the attacker sends a large number of GET requests (requests that are intended to retrieve data from a server and don't modify the server's state) in such a way that the server is unable to handle the number of requests sent to it. In a Slowloris attack, the attacker sends many incomplete HTTP requests, which keeps the server busy, processing them and thus preventing the server from processing legitimate requests.

Several counter measures have been proposed to mitigate HTTP flooding DDoS attacks. According to [2], one of such measures is Rate Limiting, which limits the number of requests that a server can handle from a particular IP address within a specific time.

Another counter measure that can be used according to [3] is Content Delivery Networks (CDNs) which are used to distribute the traffic across multiple servers (a form of load balancing), thus reducing the load on individual servers.

Yet another counter measure proposed by [4] is to handle all incoming requests (both legitimate and illegitimate) in three (3) stages.

Stage one, also called the Resource Request Monitor stage, sets a threshold on each request from a client. So, if the rate of request for a particular resource by a client is below a threshold limit, it is assumed that the request is not coming from a bot and hence the server is not overloaded so it can grant the request. Otherwise, the system assumes that it is a malicious request coming from a bot. To prove this, stage two is invoked.

Stage two, also called invisible challenge (a moving cursor challenge which only humans can solve and not bots) and is used to pose a challenge to the suspected client. If the client can solve the challenge, then it is considered a legitimate client and his request is granted otherwise, it is considered a bot. Whatever the situation is, stage three is invoked by the system.

Stage three, also called the Release Request Process, serves requests when the number of requests is limited, that is, below a particular threshold and also when the client solves the invisible challenge.

Somani et al. proposed a HTTP- GET flood DDoS mitigation technique called DARAC which is based on human behavior analysis and source IP blacklisting. The system is a DDoS aware resource allocation system designed for cloud computing. It includes three key components that make it a successful DDoS mitigation solution: intelligent auto-scaling for actual users, attacker and benign traffic segregation based on human behavior analysis, and quality services to benign users throughout the attack. DARAC divides IP addresses into good and bad ones and distributes resources to lawful users according to their needs and the quality of their service. It uses a capacity planner module to track the resources needed to provide the appropriate level of service. In addition, DARAC contains an auto-scaling mechanism that determines whether to provide resources depending on the characteristics of valid traffic observed over the previous three minutes. Without any downtime, DARAC can successfully mitigate DDoS attacks and quickly determine whether more resources are actually necessary or not. The average amount of time needed to consider all options and make a decision is between 10 and 15 seconds. [5], Singh and De presented a defense model named DDoS attack detection and mitigation technique based on HTTP count and verification using CAPTCHA. It uses IP blacklisting, http counter, and CAPTCHA to distinguish normal users from zombies [6]. Devi and Yogesh proposed a DDoS detection method based on an access matrix that captures access information from legitimate clients of a web server. The method uses a score to decide whether to accept or drop the packet. However, if the proposed counter miscalculates, it drops legitimate traffic [7].

Ahamed Aljuhani et al. proposed framework that mitigates application-layer Distributed Denial of Service (App-DDoS) attacks that can be applied against all types of DDoS attacks which includes a detection and mitigation model that operates in three modes: normal, screening, and suspicious. When a server is overloaded, the model sends alert messages and modifies the server load using a resource monitoring protocol. To assess whether a specific user is a normal user or an attacker (0 or 1), machine learning techniques are used in the screening mode. The document specifies

that the logistic regression approach is used for the binary classification. If the traffic cannot be discriminated against, the system switches to suspicious mode and each user must pass the CAPTCHA test to connect to the server. The system is designed to defeat App-DDoS attacks, and every action is recorded into a reporting module for security evaluation [8]. The proposed framework aims to address the limitations of current mechanisms for defending against App-DDoS attacks, such as slow/delayed attack detection, increased computational complexity, and reduced computational capacity of dedicated hardware.

Research by Dhanapal A. et al. discussed various real time data sets available in the internet for Application Layer HTTP Flooding DDoS attack which is beneficial to researchers in this field. Several work had been carried out in worldcup'98 dataset and exhibited their lack of details on how to regenerate the HTTP Flooding from the dataset.[9] Firstly they discusses the available real time data sets, which are The FIFA World Cup 1998 data set[10], Environmental Protection Agency (EPA) HTTP dataset [11], Dataset from web server running in the San Diego Supercomputer Center (SDSC) [12], Dataset from University of Calgary's Department of Computer Science web server [13], Dataset from ClarkNet web server [14], NASA Kennedy Space Center Florida web server dataset [15], and Dataset from University of Saskatchewan's located in Canada [16]. The overview of the dataset of World Cup 1998 to show the content and structure of it. They next examine the datasets from earlier studies to identify any gaps, which is no clear mechanism on how to convert such a massive world cup 1998 real time dataset for HTTP flooding to validate DDoS solutions and practically researchers face difficulties to make use of real time dataset for their works.[9]

Dhanapal A. et al. then proposed their solutions of the problem by designing a set of modules which is HTTP Request Filtering Module, Client Identifier to IP Address Mapping Module, HTTP Request Formatter and Flooding Module[9]. And then do the evaluation the performance of the proposed method by using the tool WireShark.

Krishan Kumar Saluja et al. provides a comprehensive overview of the research landscape related to HTTP-GET flood DDoS attacks, as well as the challenges that need to be addressed in order to effectively detect and mitigate these types of attacks. They further discuss several techniques that have been used for detecting and mitigating HTTP-GET flood attacks. One such technique is the signature-based detection technique, which involves creating a signature of the attack traffic and then using it to detect future attacks. The authors also discuss behavioral-based detection techniques, which involve analyzing the behavior of the traffic and identifying

anomalies that may indicate an attack. Additionally, the paper discusses the use of rate limiting, traffic filtering, and traffic shaping techniques as mitigation strategies. Rate limiting involves limiting the amount of traffic that is allowed to enter the network, while traffic filtering involves filtering out traffic that is suspected to be malicious. Traffic shaping involves prioritizing traffic flows to ensure that critical traffic is not affected by the attack[17]. In all these reviews, what is lacking is the actual demonstration of the attack and its possible countermeasures so that the readers can have better understanding of how the attack is performed.

3.0 EXPERIMENTS AND RESULTS

Experimental setup

3.1 Network Environment

The experiment was conducted using a variety of virtual machines running on Oracle Virtual Box and all connected. Such virtual machines include Windows 11 Virtual Machine, Windows Server 2022 Virtual Machine, Windows Server 2019 Virtual Machine, and Parrot Security Virtual Machine. Other tools like Web Browsers, and Wireshark were also used. Each of these virtual machines run HOIC (High Orbit Ion Cannon) software, which is a network stress and DDoS attack application written in the BASIC language. It can attack up to 256 Uniform Resource Locators (URLs) simultaneously. It sends HTTP (Hypertext Transfer Protocol), POST and GET requests to a computer that uses an inspired Graphical User Interface (GUI). It offers a high-speed mutli-threaded HTTP Flood; a built-in scripting system allows the deployment of "boosters", which are scripts designed to thwart DDoS countermeasures and increase DDoS output. So, all other virtual machines apart from the Parrot Security are used to generate the flood of HTTP requests. The monitoring system (Wireshark) captured and analyzed network traffic during the experiment.

3.2 Attack Simulation

In this task, we used the Windows 11 VM, Windows Server 2019 and Windows Server 2022 machines to launch a DDoS attack on the Parrot Security machine. Below are the steps we took.

3.2.1 The Process

1. Turn on Windows 11 and Windows Server 2022 Virtual Machines.
2. Switch on the Parrot Security VM. Click APPLICATIONS in the top-left corner of the DESKTOP and navigate to PENTESTING → INFORMATION GATHERING →WIRESHARK.
3. Log into the VM by entering the password.
4. This brings up the Wireshark Network Analyzer window; double-click on the PRIMARY NETWORK INTERFACE [eth0] to start capturing the network traffic.
5. Switch to the Windows 11 VM. Download High Orbit Ion Cannon (HOIC) and install it. Note that to perform the DDoS attack, run this HOIC tool from the various VMs at once.
6. Similarly, follow the previous step (step#6) on the Windows Server 2019 and Windows Server 2012 VMs. Now run the HOIC software on all the VMs except Parrot Security.
7. The HOIC GUI main window appears; click the "+" button below the TARGETS Section.
8. The HOIC –[Target] pop-up appears. Type in the target URL such as http://[Target IP Address] in the URL field.

9. Slide the power bar to high. Under the BOOSTER Section, select GENERICBOOST.HOIC from the drop-down list, and click add.
10. Set the THREADS value to 20 by clicking the > button until the value is reached.
11. Now switch to the Windows Server 2019 and Windows Server 2022 VMs and follow the previous steps to configure HOIC.
12. Once HOIC is configured on all machines, switch to each machine (Windows 11, Windows Server 2019, and Windows Server 2022) and click the FIRE THE LAZER! Button to initiate the DDoS attack on the target [Parrot Security] VM.
13. Observe that the status changes from READY to ENGAGING.
14. Switch to the Parrot Security VM and observe that WIRESHARK starts capturing a large volume of packets, which means that the machine is experiencing a huge number of incoming packets. These packets come from Windows 11, Windows Server 2019, and Windows Server 2022 machines.
15. You can observe that the performance of the machine is slightly affected and that its response is slowing down.
16. Stop all processes.
17. This concludes the demonstration of how to perform a DDoS attack using HOIC.

3.3 Defense Mechanisms

In this lab, we attempt to demonstrate how to detect and protect against DDoS attacks using Anti DDoS Guardian. We used other tools such as Windows 11 VM, Windows Server 2022 VM, Windows Server 2019, HOIC and Web Browsers with Internet connection. Anti DDoS Guardian is a DDoS attack protection tool. It protects utilities like Internet Information Services (IIS) Servers, Apache Servers and so on. It monitors incoming and outgoing packets in real-time. It displays the local address, remote address, and other information of each network flow. It limits network flow number, client bandwidth, client concurrent TCP connection number and TCP connection rate. It also limits the User Datagram Protocol (UDP) bandwidth, UDP connection rate, and UDP packet rate.
In the task below, we use Windows Server 2019 and Windows Server 2022 machines to perform a DDoS attack on the target system Windows 11.

3.3.1 The Process

1. Turn on Windows 11, Windows Server 2019, and Windows Server 2022 Virtual Machines.
2. In the Windows 11 VM, run DDoS protection tool (Anti DDoS Guardian). Follow the onscreen procedure to install the tool.
3. Run the Anti DDoS Guardian windows which show information about incoming and outgoing traffic.

4. Log into the Windows Server 2019 VM to activate it.
5. On Windows Server 2019 and Windows Server 2022, repeat the previous procedures on how to set up an attack using HOIC.
6. Switch back to the Windows 11 VM and observe the packets captured by Anti DDoS Guardian.
7. Observe the huge number of packets coming from the host machines [Windows Server 2019 and Windows Server 2022].
8. Double click any of the sessions.
9. The Anti DDoS Guardian traffic Detail Viewer window appears, displaying the content of the selected session in the form of raw data. You can observe the high number of incoming bytes from the machines.
10. You can use various options from Anti DDoS Guardian Traffic Program such as Clear, Stop Listing, Block IP and Allow IP. Using Block Ip option blocks the IP address sending the huge number of packets. You can click this option.
11. Observe that the blocked machine turns red in the Action Taken Column.
12. This concludes the demonstration of how to detect and protect against a DDoS attack using Anti DDoS Guardian.

4.0 RESULTS DISCUSSION AND ANALYSIS

4.1 Attack Impact Analysis

During the experiment, the HTTP application layer DDoS flooding attacks were launched with varying intensity, ranging from moderate to high request rates. The analysis of the attack impact revealed that because of the large volume of packets sent to the target, its resources are flooded and as such, its performance is affected and that its response to request slows down. a severe degradation in the performance and availability of the target host [Parrot Security] became inevitable. At one point, the target machine became unresponsive under high attack volumes.

4.2 Defense Mechanism Evaluation

The defense mechanisms implemented were evaluated in terms of their effectiveness in mitigating the DDoS flooding attacks. The rate limiting mechanism demonstrated substantial success in filtering out excessive requests, reducing the impact of the attacks. Traffic filtering techniques effectively identified and blocked a significant portion of malicious traffic, further enhancing the defense posture.

The experiment results validate the severity of HTTP application layer DDoS flooding attacks and highlight the importance of implementing robust defense mechanisms. The rate limiting, and traffic filtering, techniques proved to be effective in mitigating the impact of the attacks. However, it is crucial to note that a multi-layered defense approach, combining various mechanisms, is necessary for comprehensive protection against evolving DDoS threats.

The results showed a significant improvement in the target's performance and resilience to flooding attacks when the defense mechanism was employed. The defense system successfully mitigated the impact of the attacks by intelligently identifying and blocking malicious traffic while allowing legitimate requests to be processed.

5.0 CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

The study presented a comprehensive experiment investigating HTTP application layer DDoS flooding attacks. From the result of our experiments, we conclude that the best approach to detecting, mitigating, and protecting against HTTP Flooding DDoS attacks is a multilayered approach, where we employ multiple defense mechanisms against this kind of attack.

In this paper, we investigated HTTP application layer DDoS flooding attacks and proposed a defense mechanism to mitigate their impact. Through experimental evaluation, we demonstrated the vulnerability of HTTP protocols.

In the future, we will work on improving defense against DDoS attacks by utilizing the techniques of IPv6 Unique Local Addresses (ULAs).

REFERENCES

[1] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher. 2004. Internet Denial of Service: Attack and Defense Mechanisms. Prentice Hall.

[2] M. Liao, S. Chakraborty, and C. Guan. 2017. A Survey of DDoS Attack and Defense Mechanisms. Journal of Network and Computer Applications 88, 27-45. S.

[3] Sinha, D. Liu, and Y. Xiang. 2014. DDoS Attack Detection and Mitigation Using IP Flow Records. Journal of Network and Computer Applications 45, 140-147.

[4] Durga Naga Mallesware Rao Varre, and Jayanag Bayana 2022. A Secured Botnet Prevention Mechanism for HTTP Flooding Based DDoS Attack.

[5] G. Somani, A. Johri, M. Taneja, U. Pyne, M. S. Gaur and D. Sanghi, "DARAC: DDoS Mitigation Using DDoS Aware Resource Allocation in Cloud," in Int. Conf. Inf. Syst. Secur., in Information System Security, in Lecture Notes in Computer Science, vol. 9478, pp. 263-282, doi: 10.1007/978-3-319-26961-0_16.

[6] K. J. Singh and T. De, "DDOS Attack Detection and Mitigation Technique Based on Http Count and Verification Using CAPTCHA," 2015 Int. Conf. Comput. Intell. Netw., Bhubaneshwar, 2015, pp. 196-197, doi: 10.1109/CINE.2015.47.

[7] S. R. Devi and P. Yogesh, "An effective approach to counter application layer DDoS attacks,' in 2012 Third Conf. Comput. Commun. Netw. Technol. (ICCCNT 2012), Coimbatore, 2012, pp. 1-4, doi: 10.1109/ICCCNT.2012.6395941.

[8] Ahamed Aljuhani 1,3 *, Talal Alharbi 2,3, Bradley Taylor 3, "Mitigation of Application Layer DDoS Flood Attack Against Web Servers" in Journal of Information Security & Cybercrimes Research 2019; Volume 2 Issue (1), 83-95, doi: 10.26735/16587790.2019.002.

[9] A. Dhanapal and P. Nithyanandam, "An effective mechanism to regenerate HTTP flooding ddos attack using real time data set," 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017.

[10] World Cup 1998 Data Set, May 2017, [Online]. Available: http://ita.ee.lbl.gov/html/contrib/WorldCup.html

[11] EPA-HTTP Data Set, May 2017, [Online]. Available: http://ita.ee.lbl.gov/html/contrib/EPA-HTTP.html

[12] SDSC-HTTP Data Set, May 2017, [Online]. Available: http://ita.ee.lbl.gov/html/contrib/SDSC-HTTP.html

[13] Calgary-HTTP Data Set, May 2017, [Online]. Available: http://ita.ee.lbl.gov/html/contrib/Calgary-HTTP.html

[14] ClarkNet-HTTP Data Set, May 2017, [Online]. Available:

http://ita.ee.lbl.gov/html/contrib/ClarkNet-HTTP.html

[15] NASA-HTTP Data Set, May 2017, [Online]. Available:

http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

[16] Saskatchewan-HTTP Data Set, May 2017, [Online]. Available:

http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html

[17] Krishan K. S., Paramvir S. and Karanpreet S. 2016 " Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges " DOI: 10.1016/j.cose.2016.10.005

[18] EC-Council Ethical Hacking and Countermeasures CEHv12 Module 8-13.

[19] Paul Nicholson – Five Most Famous DDoS Attacks, May 4, 2022, [Online]. Available:

https://www.a10networks.com/blog/5-most-famous-ddos-attacks/#:~:text=A%20Brief%20History%20of%20DDoS,become%20a%20classic%20DDoS%20attack.