Mapana – Journal of Sciences
2023, Vol. 22, No. 1, 121-145
ISSN 0975-3303 | https://doi.org/10.12723/mjs.64.7

Distributed Denial of Services Attacks on Cloud Servers: Detection, Analysis, and Mitigation

Sudesh Pahal[*] & Anjana

Abstract

Today, most IT companies are moving towards Cloud infrastructure and technology due to its flexibility, scalability, and cost-effective features. Nevertheless, security is still the main hindrance to accepting cloud computing on a large scale. There are many security issues related to cloud implementation, and one of the major threats is Distributed Denial of Services (DDoS) attack on cloud servers and applications. The DDoS attack is a most prevalent security issue where the attacker intends to make all victim's resources, like cloud servers, storage, bandwidth, etc., unavailable to a general user, which results in dissatisfactory outcomes in related business. This paper emphasizes understanding issues related to DDoS attacks, such as server outages, asset theft, and resource losses, followed by their detection and analysis. The paper also explores the possible mitigation strategies to reduce the impact of DDoS.

Keywords: Security, Availability, Distributed Denial of Services, Botnet-based DDoS, Flood attacks, Detection, and Mitigation

I. Introduction

Many IT companies are still reluctant to use cloud infrastructure due to security issues. It is because Cloud computing architecture

[*] Department of Electronics and Communication, Maharaja Surajmal Institute of Technology, New Delhi; Email: pahal.sudesh@gmail.com

has multiple vulnerabilities where security is on sake. Based on Cloud computing services models like IaaS, PaaS, and SaaS, we also have many loopholes in their security architecture [1]. DDoS attacks are one of the biggest problems of these security threats. The DDoS attack is a disruptive attempt to hit the traffic of the victim server to make all its related resources unavailable to the legitimate user [2]. DDoS attacks enhance their impact by using various compromised devices available in the network. The main target of DDoS attacks is to clog the network bandwidth with multiple fraudulent requests, preventing authorized user's requests from reaching the required destination server [3].

DDoS attacks use the most prominent mechanism where the attacker always tries to hide its identity so that at destination server's firewall setting and intrusion detection systems present cannot identify and block it. To this end, attackers always use intermediate compromised devices available on the internet, which are controlled and asked to raise millions of fake requests to clog the network. [4]. These intermediate devices are called bots, and a group of such devices is called a botnet, as shown in Figure [1].One system called the command and control (C&C) server is an intermediate between the attacker's host and the other compromised hosts (bots). Once it receives an order from the attacker's host, it triggers the botnet to attack the victim's host by sending malicious packets [5].

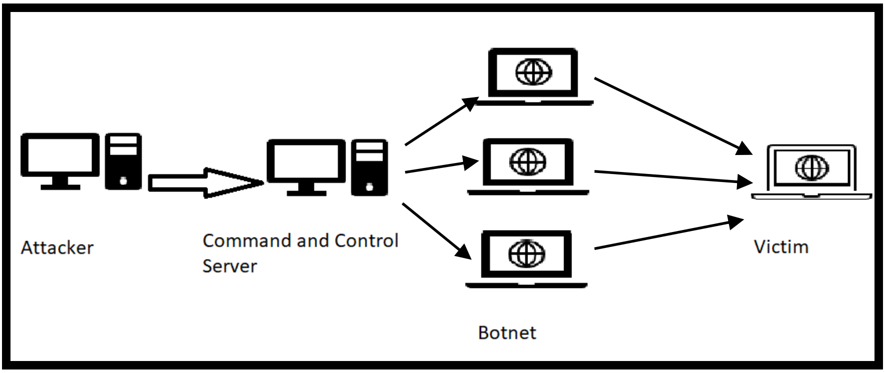

Figure 1: DDoS Attacks Architecture in Cloud Computing

This paper aims to explore the understanding of DDoS attacks, their mechanism, and in-depth study of various DDoS Detection and Analysis Techniques. Section I includes an introduction to the DDoS attacks with background and motivation. Section II covers a literature review, including types of DDoS attacks in cloud computing based on cloud components attacked and networking infrastructure. The paper emphasizes detection methods used to detect DDoS attacks where the main task of these methods is to differentiate fraudulent requests from the actual legitimate user's request. Furthermore, the last section investigates the different mitigation strategies to defend the DDoS attacks in cloud computing.

II. Literature Review

DDoS attacks are majorly based on botnet mechanisms [6]. Depending upon the components targeted to attack in cloud computing and networking infrastructure, we have different types of DDoS attacks. First, let us introduce Botnet-DDoS attacks networks models, which have been categorized into the following three categories:

(i) Agent handler Model: In this model, as shown in Figure 2, two main components are: handlers and agents, which are used by attackers to maximize the non-availability of the legitimate user's service. Handlers are controlled by the attacker, who further communicates with agents to give the instructions to attack the victim or upgrade the instructions. Agents are malicious code running through the internet, which are not aware that their machines have been compromised by the handlers [6]. Terms can be used interchangeably, ‘handlers’ as ‘masters’ and ‘agents’ as ‘demons’ respectively [7].

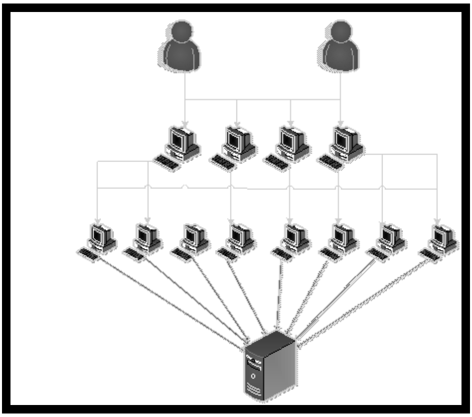

Figure 2: Agent-Handler Model

(ii) Internet Relay Chat (IRC) Model: IRC Model architecture is just like the Agent handler model except for handlers, which have been replaced by IRC communication channels, as shown in Figure 3. In the Agent–Handler model, handlers are a packet of malicious code that runs through the internet. However, in the IRC model, handlers have been replaced by IRC communication channels which act as an interface between attackers and agents. Also, attackers do not require any information regarding the agents, as once IRC is available to them, all information about all agents will also be available [8]. IRC is more beneficial to an attacker because it has a larger volume of traffic, due to which the attacker can easily hide his presence. IRC provides a legitimate port for the attacker to communicate with agents.

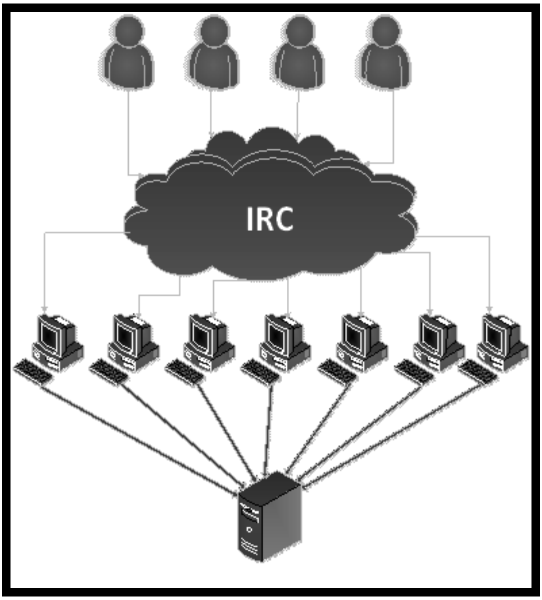

Figure 3: Internet Relay Chat (IRC) Model

(iii) Web-based Model: In this model, various bots are designed and configured using various PHP scripts, which are used to send statistics to a website. Encrypted Communication is done using HTTP or HTTPS protocols via the 80/440 port in the Web-based model. Though the IRC model is the best model in botnet-based DDoS attacks Web-based model has its own advantages over IRC like Easy Acquisition and setup configuration, use of lesser bandwidth with more distributed load, improved command functions, and reporting, coverage of traffic via ports, etc. [9]

There are many new kinds of attacks identified every day, and still, few remain undercover. Here, the focus is to explore botnet-based DDoS attacks that trouble the application layer on the cloud server. The vulnerability decides the type of DDoS attacks. So, on the basis of network and cloud components impacted, the following classification has been done for various DDoS attacks:

1) **Application DDoS Attacks**: These kinds of attacks fill all available bandwidth with illegitimate user requests and amplify the power of the attack by forcing expensive operations

on the victim's cloud server. Services are shattered by either offering malicious data or hampering the routing protocols [10].

a) **HTTP Flood Attacks**: The goal of these attacks is to exhaust all the resources of the cloud server by hitting many HTTP requests, which are called HTTP Flood attacks. An HTTP request is much more costly on the server because it requires the loading of many files and processing, and the main target of such attacks is at this layer of processing where an HTTP request is processed and the result generation of a webpage or packets to return to requester. Now, the attacker always instructs a bot to raise an HTTP request through a valid IP address which is processed by the target server and loaded into memory, and then packets are formed to be sent to the bot, as shown in Figure 4. So, the attacker gives such instructions in the loop or repeatedly so that all input/output devices, CPU, bandwidth, and memory are at maximum utilization. In a similar pattern, an HTTP request raised by the bot, again and again, becomes part of regular web traffic, which causes more problems in differentiating between a legitimate user's request and an illegitimate user's request while filtering HTTP requests [11].

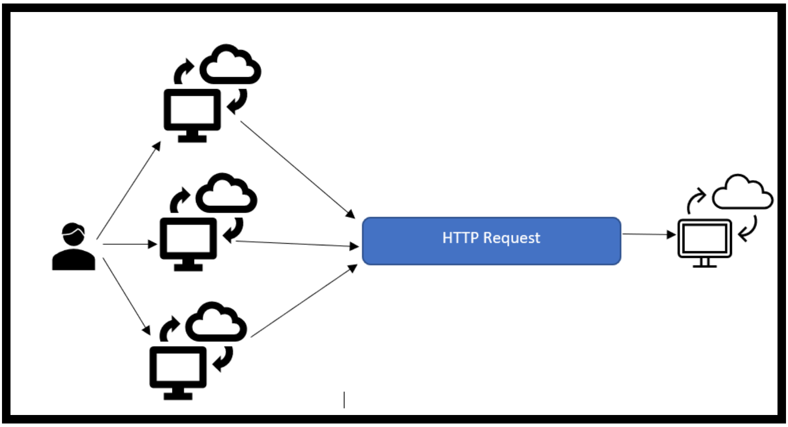

Figure 4: HTTP Flood Attacks

Based on the above architecture of HTTP Flood Attacks, the following is the further classification done [12]:

i) **HTTP Fragmentation attack:** The main aim of this attack is to bring down the cloud server by sending malicious HTTP requests recursively to keep the HTTP connection busy for a long time without any alarm [13].

ii) **Slow Request/Response Attacks:** It has two types of Slow Request and Slow Response attacks: a) Slow Header Attack, where the attacker never sends the complete information in the header of the packet and due to which cloud services will have as many as open connection as many half requests will exist which will result into an inaccessible cloud server.[14] Furthermore, b) Slow Response attack, in which the attacker reads the response from victims that much slow that again it makes the server unavailable for other legitimate users [15].

iii) **Slow Request Bodies or (RUDY) Attack**: The aim of this attack is to bring down the cloud server by sending an HTTP header that defines the content-length field of the message post the message body [16].

iv) High Workload Request Flood Attacks: First, identify the vulnerabilities in the cloud server architecture, then, depending on it, either send high workload request or SQL injections malicious code to harm the victim's cloud server by making all its CPU, memory, network bandwidth unavailable [17].

v) **Valid and Variant GET/POST Flooding Attacks:** In Valid attacks, to make the cloud server resources exhausted, the attacker sends multiple session open requests to the victim's cloud server[18]. In Variant attacks, the attacker uses a single session, but with the help of a botnet, within a single session attacker can send volumetric requests, due to which the cloud server is not able to process all the requests [19].

b) **SIP Flood Attacks:** Voice over IP address uses SIP (Session Initiation Protocol) standard for call set up using public internet access. SIP proxy servers are attacked using two methods: i) using a SIP Invite packet using a legitimate IP address or ii) using a botnet. A flood attack can be launched by an attacker to deplete the network available so genuine VoIP requests will never reach the SIP proxy server, and the call receiver gets a lot of fake VoIP calls, which makes it tough to reach legitimate callers respectively [20].

c) **Distributed Reflector (DRDoS) Attacks:** As shown in Figure 5, reflectors are used to hide the identity of the sources used in traffic attacks. Reflectors are third-party sources like routers, web servers, or cloud servers that help to relay the attack traffic to the victim by responding to an incoming malicious packet. There are three stages of a DRDoS attack: i) Attacker takes control over all bots (zombies/slaves), ii) Once all zombies are under the attacker's control, attackers send instructions to zombies to send attack traffic through reflectors using the victim's IP address as Source IP address and iii) reflectors send reply traffic to the victim who finally makes DDoS attack. In this way, DDoS attacks amplify the attack traffic by distributing it among various reflectors, which causes lots of damage to the services [21].

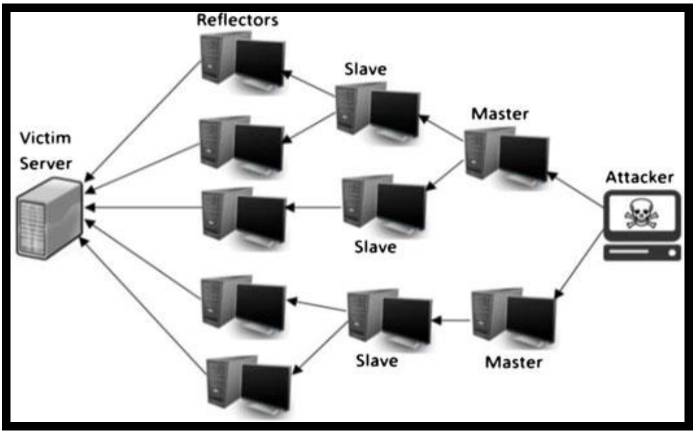

Figure 5: DRDoS Attacks

d) **DNS Amplification Attacks:** Domain Name System amplification attack is a kind of Distributed Reflector attack, but as its name speaks, it amplifies the attack and distributes it among various DNS servers in a recursive manner. First, the attacker compromises the DNS server by sending a signal, then using this compromised DNS server further sends instructions to botnets. Botnets then send spoofed information to a vast number of DNS servers in a recursive manner which results in an amplified DNS traffic in return, as shown in figure 6 below:

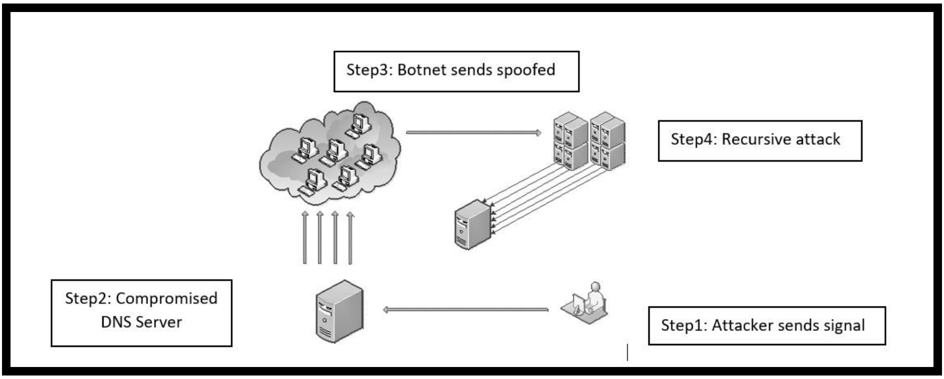
Step3: Botnet sends spoofed
Step4: Recursive attack
Step2: Compromised DNS Server
Step1: Attacker sends signal

Figure 6: DNS Amplification DDoS Attack

e) **Malformed packet:** To crash the victim's system, the attacker uses the malicious formed packets and sends them to the victim. These packets can be formed either based on the IP address or IP packet. In an IP address attack, the attacking packet contains the same IP address as the source and target IP address, due to which the victim's server gets confusion and resulting in a system crash. However, in an IP packet attack, within the packet, optional fields are randomized, and all other mandatory fields are set to true, which increases the processing and time consumption in packet handling by the victim [22].

f) **Protocol vulnerability exploitation:** These types of attacks are based on the vulnerabilities present in cloud computing. Like we have less secure APIs, an indefinite number of resources allocated, data storage-related vulnerabilities, and vulnerabilities in Virtual Machines, Hypervisors, and Virtual Networks. So, depending on any of the vulnerabilities, the attacker sends instructions to botnets to attack the victim's cloud server [23].

2) **Network DDoS Bandwidth attacks:** Network DDoS attacks look for IP weakness and then attack using only a single source. These types of Network DDoS attacks consume the bandwidth to the maximum. The prominent examples are:

a) **SYN Flood Attacks:** SYN Flood attack is based on the vulnerability of the TCP three-way handshake. In this attack, packets are sent with an unknown IP address by the attacker. So when the server receives such type of packet which does not have a correct IP address, then to complete the three-way handshake, it looks for IP address information from the client. But this packet has been sent by an attacker, not a legitimate user, so it will wait till the information is not complete. Thus, many incomplete requests fill the memory of the victim's server and result in a timeout. This accumulation of incomplete connections does not allow for the processing of any request further, and all related services are disabled completely [24].

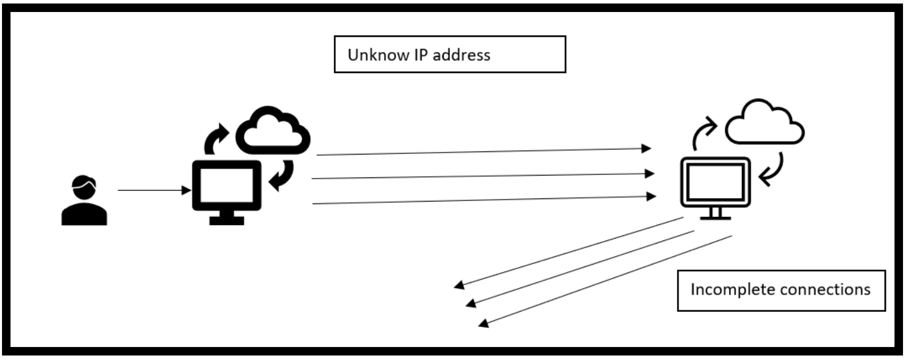

Figure 7: SYN Flood Attack

b) **ICMP Flood Attacks:** ICMP attack is a bandwidth attack based on IP protocol that determines the status of the network. In this attack, packets can be sent to a single machine or a complete network. When a packet is sent from one machine to a local network using an IP broadcast, then each machine on the network receives that packet. Similar way, when a packet is broadcasted to an outside WAN network, then each machine receives the packet in the target network. Ping of Death is an example of an ICMP attack [25].

III. DDoS Detection and Analysis Methods

Why do we need DDoS detection techniques in cloud computing? This question arises when possible; we do not have any other alternative to find out the difference between a legitimate user's request and a malicious attack's request. The main aim of the detection technique must be to identify the genuine user's request and to differentiate it from the fraudulent request because both can be confusing to each other for a cloud server. It is much more challenging to differentiate between the two when the legitimate user sends a volume of requests to the cloud server, called a flash event. So, service providers must be on alert to handle such a situation, and an attack detector should be installed which monitor the real-time traffic. Generally, most IT firms install monitors to scan the network traffic in their LAN, possibly near the firewall and most vulnerable resources or server. All detective methods identify a possible attack by scanning an abnormal traffic behavior than a usual load by using some statistical methods. Depending on these statistical results following detection techniques have been classified:

1) **Activity Profiling:** Activity profile is the study of the header information given in a network packet. It is an average rate of flowing similar packets with similar information, e.g., IP address, port, and network protocol. Activity level is measured by the total elapsed time taken by all consecutive packets to flow through the network. All inbound and outbound flows can be summed up to find the total network activity. To make this activity profiling easy and precise, network flows are divided into clusters. When there is an increase in activity logs of these clusters, then it clearly indicates an attack. Also, if there is a sudden increase in the total number of clusters, it also indicates an attack because the attacker can randomly increase its agent count to amplify the attack. So, we can detect the attack using activity profiling [26].

2) **Backscatter Analysis:** In Backscatter analysis, a large number of IP addresses are scanned to monitor the IP spoofing activity using backscatter packets. A backscatter packet is a response from the victim's server, which has the Source IP address as the victim's address but the destination address as various spoofed IP addresses. Probably, when uniform IP address distribution is done during an attack, there is a finite number of chances that the attacker receives

a backscatter packet which is monitored using the cluster's destination address distribution uniformity technique to detect the attack[27].

3) **Wavelet Analysis:** Wavelet analysis provides a global frequency distribution without time localization. So, at a given point in time, wavelets can identify the components at a specific high frequency by separating the anomalous signal from the background noise to facilitate detective applications. Now, by analyzing these separate signals and noise in their respective windows, abnormal behavior can be identified. Majorly, high and medium spectral windows are analyzed and then compared with the threshold value to identify the possibility of attack [28].

4) **Sequential Change-Point Detection:** Sequential Change-Point detection technique works by identifying the change in network statistics due to any attack. The main factors used to filter the target network are IP address, Port, and Network protocol, and then keep the result as a time series to represent the cluster's activity. Input for this time series is continuous sample data and low computational resources [29]. By taking an example here of cumulative sums algorithms, also known as Cusum algorithms. It works on the principle that if expected traffic and actual traffic time difference exceed a threshold value set before then, there is the probability of abnormal behavior and the possibility of a DDoS attack [30].

To explore more precisely the botnet-based DDoS attacks detection techniques, one more categorization has been done:

1) **Signature-Based Botnet-based DDoS Detection:** Signature-Based Botnet-based DDoS detection techniques usually work on the mechanism that they always try to find a signature or a known identity for each flooding attack. The accuracy and performance depend on the regular signature updates in the database to find out the match based on the signature. These detection techniques are like anti-virus software which scans for virus definitions already present in the database [31]. In these detection methods, the

network is searched for any malicious code or sequence of bytes that has some pattern related to a signature or some malicious activity.

a) **User's browsing pattern-based Detection Technique:** This detection technique is to identify the features of a user's Web-browsing behavior, which can help to differentiate between a human's valid request or Botnet malicious requests in the server traffic. There are three main elements of a user's browsing behavior: i) HTTP Request rate and ii) Page View time and requested sequence. Different patterns are identified in the traffic data collected on the server's side to identify legitimate user requests or illegitimate user requests. Xie and Yu have proposed a simulation technique in which HTTP request from normal web user is characterized to detect HTTP Flooding attacks [32]. This technique has the advantage that it is many efficient and accurate results. But it has a disadvantage, too, due to its computational complexity. Yatagai, Isohara, & Sasase also proposed two detection algorithms to identify page access behavior [33]. One is based on Page Browsing Order, and the other depends on the amount of information on the page and browsing time.

b) **Scheme-based HTTP GET Flood Detection Technique:** Lin et al. identified that all HTTP GET requests from a malicious user are in repetitive mode and with the exact same information continuously hitting to target server within a few milliseconds [34]. But a legitimate user's request behavior cannot be generated in such a pattern. So, using source IP, URI hash, URI size, timestamp, and matching information, an experiment was conducted on a test-bed of two servers and was able to identify the HTTP flooding attacks at a high rate and low rate attacking tools.

c) **Statistics-based Detection Techniques:** This technique works on the principle of the statistic approach that studies the behavior and its deviation from what is expected to be observed. Choi et al. suggested a threshold-based framework to detect HTTP flooding attacks by analyzing the monitoring time and period by calculating average HTTP requests[35]. The main advantage of this detection technique is that there is no need to analyze each HTTP request; hence fewer resources are required. However, it has the disadvantage that it overlooks the other features on the basis of which more attacks can be detected.

2) **Anomaly-Based Botnet DDoS Detection:** The main feature of this detection technique is to study network behavior. A network pattern is identified against the expected network behavior either set by a network administrator or learned by a heuristic approach, or both. If any deviation is found from acceptable network behavior, alarms are generated. Pimentel et al. suggested the attack when there is a threshold limit reached for deviating from acceptable network behavior to observed network behavior [36]. Rexroad et al. found an advantage of anomaly-based botnet DDoS detection technique over signature-based systems that it can still identify a new attack in which the signature is not updated in the database if it falls outside the normal, acceptable network pattern[37]. But Owezarski found that anomaly-based detection systems have a few disadvantages, too [38]: i) The rule-defining process has a dependency on the various classification of protocols used by cloud providers. To work it efficiently, there must be a vast knowledge base so that each new attack can be identified if it does not fall into acceptable network behavior. ii) If a few parameters fall under an acceptable network pattern, the attack will be missed and bypassed. iii) Any activity like directory traversal, if found within network protocol, will also result in missing the attack.

IV. Results and Discussion

In this section, DDoS Mitigation Strategies, and their comparative analysis are conducted as depicted in Table 1. The most challenging part of any DDoS Mitigation technique [40-52] is to differentiate between a genuine request from a normal user and a malicious request from a botnet or attacker. The attacker always tries to blend the traffic into normal traffic, so it creates more problems to identify among these. The complexity of the attack increases the complexity of the mitigation technique and differentiation between normal traffic and attacker traffic. Every mitigation technique works in four stages: i) Detection, ii) Response, iii) Routing iv) Adaptation. So, based on the functionality, mitigation techniques have been categorized into the following ways for HTTP flooding attacks:

i) **Software-based DDoS Mitigation Techniques:** Such types of techniques [48,49] use more memory and CPU usage as they always read the flow of information. To differentiate between a request initiated by a legitimate user and a request initiated by a botnet, these mitigation techniques first authenticate the users, and to authenticate the following methods have been proposed so far:

   a) CAPTCHA: Wen et al. proposed a technique in which, during user's authentication, an additional check is performed to differentiate between a human's request or request raised by any machine [38]. Nevertheless, the drawback of this additional check found that CAPTCHA itself can be attacked by DDoS attacks. CAPTCHA is just a random number, alphabet or alphanumeric, any graphic or audio, or video which is being used to cross-validate additionally that request has been raised by a machine or human being. But it was not so successful when the botnet increased its rate of request, and each time a new CAPTCHA was generated, and after some time, it started repeating. In that case, the attacker saves each CAPTCHA generated for future use if it comes again.

   b) Kill-bots: Kandula et al. proposed another technique based on CAPTCHA only in which if a user exceeds the threshold limit of incorrect CAPTCHA attempts, then it blacklists the IP address of that user and never allows to raise any request [40].

   c) Secure Overlay: Stavrou et al. proposed a DDoS mitigation technique in which each net server uses

CAPTCHA, and if the request is found genuine, then it allows to forward the traffic through a secure servlet which further sends traffic to selected beacon nodes which finally sends the traffic to its server [41]. In this technique, a certificate is generated for every genuine identified user and then allowed to send the traffic using redundant paths without re-authenticating the request.

ii) **Hardware-based DDoS Mitigation Techniques:** These mitigation systems [50,51] are based on three main components of the network: a) HTTP GET filter, b) URL extractor, and c) a hash-table-based URL counter. HTTP GET filter does parsing of HTTP GET packet. URL extractor separates the URL information from the HTTP GET packet, and hash table is used to store the Source IP address and its specific hits using a particular URL. If the count exceeds the threshold limit of allowed URL hits, then the particular IP address is marked as blacklisted. Following mitigation techniques have been proposed based on hardware:

➢ Not-a-Bot: Gummadi et al. proposed a Not-a-Bot mitigation technique in which Trusted Platform Modules were used, which are cryptographic processors used in laptops and desktops to study the behavior of keyboard and mouse activities to identify human activities. But this mitigation technique was not efficient in a smarter attack in which the bot used tricks to behave like a human and got the attestation of Not-a-Bot and sent the malicious request to the Web server [42].

➢ Sentinel: Djalaliev et al. suggested that a hardware token is a much more efficient way to mitigate the risk than any other CAPTCHA or puzzles. But we must configure the front-end monitoring device and Kerberos Federated authentication settings [43].

iii) Other important DDoS Mitigation Techniques:

a) Black Hole Routing: Black Hole routing is a technique in which malicious traffic is dropped to a null interface [44]. It is a filtering technique in which, after differentiating between legitimate traffic and malicious traffic, water is dumped into Black Hole at the router level. However, sometimes, we have to route both traffic to the null interface to mitigate the DDoS attack.

b) Rate Limiting: It is the most common technique to mitigate DDoS attacks in which a threshold is defined to accept the number of requests by a cloud server [45]. If it exceeds the threshold value, packets will be rejected summarily. There are two ways to perform rate limiting: i) Flow rate limiting and ii) Aggregate rate limiting. In the former, individual traffic is monitored and limited to flow to a server, and in the latter total traffic is monitored and limited to a server.

c) Web Application Firewall: To stop a DDoS attack, the firewall is installed between the cloud server and the whole internet. The firewall inspects each incoming request, and if any violation is found based on the security policy defined in it, then malicious requests are filtered [46]

d) Anycast Network Diffusion: Lua et al. suggested some network configuration in which malicious traffic is directed and scattered into various distributed servers, which prevents the original cloud server from DDoS attack. This is also a resilient approach in case of high volume and congested networks where network addressing and routing of traffic is done

Table 1: Comparative analysis of DDoS attack mitigation techniques

Category	Technique used	Strategies	Drawbacks
Software-based DDoS Mitigation Techniques	CAPTCHA	Authentication check	Multiple attempts can fail the system.
	Kill-bots	The threshold set for attempts	Can blacklist genuine users
	Secure Overlay	Certificate generated for	Time and computation

Category	Technique used	Strategies	Drawbacks
		authenticated users	overhead
Hardware-based DDoS Mitigation Techniques	Not-a-Bot	cryptographic processors used to study the behavior of human activities	The bot can mimic human behavior
	Sentinel	A hardware token is allotted.	Front-end support is also required.
Miscellaneous	Black Hole Routing	malicious traffic is dropped to a null interface	Dependency on filters to select legitimate traffic.
	Rate Limiting	accepts a fixed number of requests by a cloud server	Genuine requests can be ignored.
	Web Application Firewall	A firewall is installed between a cloud server and the whole internet.	
	Anycast Network Diffusion	Traffic is scattered into various distributed servers	Applicable to specific network configurations.

V. Conclusion and Future Scope

In this paper, we have examined various types of DDoS attacks based on different features. Then, we discussed analysis techniques like activity profiling, and scatter analysis. In a further section, DDoS attack detection based on Signature and Anomaly-based has been examined. The last section covers the different mitigation techniques based on network, hardware, and software, with various advantages and disadvantages of each. After studying all these details about DDoS attacks and their mitigation techniques following conclusion has been withdrawn, we must keep an alert throughout the flow of information from the source server to a destination server, which makes an end-to-end DDoS attack detection, analysis, and mitigation solution. We should take the following point into consideration to keep our cloud server safe from DDoS attacks:

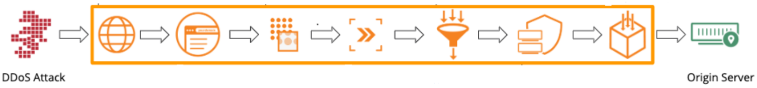

Figure 8: End-to-end DDoS Alert and Mitigation

i) Stay Online: Attack traffic should be absorbed so that customer is always online.

ii) Identify anomalous traffic: HTTP requests should be flagged for genuine or botnet-generated requests.

iii) Protect applications with control: Rate limiting can be applied at a granular level so that slow-rate attacks can be blocked.

iv) Block direct attacks: Protect the cloud servers from direct attacks by keeping a tunnel between the origin and the server.

v) Protect origin infrastructure: First, detect and then block the layers from attacks.

vi) Anticipate attacks: Proactive mitigation should be done by studying behaviour analysis of signatures and IP addresses.

vii) Protect all TCP ports: Protect all TCP ports by using proxy traffic from the attack traffic.

References

1) Anjana & Ajit Singh, Security concerns and countermeasures in cloud computing: a qualitative analysis International Journal of Information Technology volume 11, pages683–690(2019), 28 February 2018(Original)

2) G. Carl, G. Kesidis, R. R. Brooks and Suresh Rai, "Denial-of-service attack-detection techniques," in IEEE Internet Computing, vol. 10, no. 1, pp. 82-89, Jan.-Feb. 2006.

3) Hadeel S Obaid, International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181, Vol. 9 Issue 03, pp 631-636, March-2020

4) Mohammad Masdar* and Marzie Jalali, A survey and taxonomy of DoS attacks in cloud computing, SECURITY AND COMMUNICATION NETWORKS Security Comm. Networks 2016 in Wiley Online Library (wileyonlinelibrary.com); Vol. 9, pp 3724–3751, DOI: 10.1002/sec.1539, Published online 13 July 2016

5) Yu S. Distributed Denial of Service Attack and Defence. Springer: London, UK, 2014.

6) Esrra Alomari et al.Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art, International Journal of Computer Applications (0975 – 8887), Volume 49– No.7, July 2012

7) S. M. Specht and R. B. Lee, "Distributed Denial of service: Taxonomies of attacks, tools, and countermeasures," in the Proceedings of the International Workshop on Security in Parallel and Distributed Systems, 2004, pp. 543-550.

8) K. J. Houle, "Trends in Denial of Service Attack Technology," CERT Coordination Center, Carnegie Mellon Software Engineering Institute, Oct 2001.

9) Company, "Distributed Denial of Service (DDoS) and Botnet Attacks," An iDefense Security Report, 2006.

10) A Mishra, BB Gupta, RC Joshi, ―A Comparative Study of Distributed Denial of Service Attacks, Intrusion Tolerance and Mitigation Techniques," In the proc. of European Intelligence and Security Informatics Conference (EISIC-2011), pp. 286-289, 2011.

11) P. Bächer, et al., "Know your enemy: Tracking botnets," The Honeynet Project and Research Alliance, Tech. Rep,2005.

12) Esrra Alomari et al., A Survey of Botnet-Based DDoS Flooding Attacks of Application Layer: Detection and Mitigation Approaches,Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber SecurityChapter: 3Publisher: IGI GlobalEditors: Brij Gupta, Dharma P. Agrawal, Shingo Yamaguchi, DOI: 10.4018/978-1-5225-0105-3.ch003, May 2016

13) Suliman, A., Shankarapani, M. K., Mukkamala, S., & Sung, A. H. (2008). RFID malware fragmentation attacks. Paper presented at the collaborative technologies and systems, 2008. Cts 2008. International symposium on.

14) Aiello, M., Papaleo, G., & Cambiaso, E. (2014). Slowreq: A weapon for cyber warfare operations. Characteristics, limits, performance, remediations. Paper presented at the International Joint conference Soco'13-cisis'13-iceute'13.

15) Bethencourt, J., Franklin, J., & Vernon, M. (2005). Mapping internet sensors with probe response attacks.Paper presented at the Usenix security.

16) Damon, E., Dale, J., Laron, E., Mache, J., Land, N., & Weiss, R. (2012). Hands-on denial of service lab exercises using slowloris and rudy. Paper presented at the proceedings of the 2012 information security curriculum development conference.

17) Yu, J., Fang, C., Lu, L., & Li, Z. (2010). Mitigating application layer distributed denial of service attacks via

effective trust management. IET Communications, 4(16), 1952-1962.

18) Ali, S. T. (2009). Throttling DDoS attacks using integer factorization and its substantiation using enhanced web stress tool. National Institute of Technology Karnataka Surathkal.

19) Zhou, Y., & Jiang, X. (2012). Dissecting Android Malware: Characterization and evolution. Paper presented at the security and privacy (sp), 2012 IEEE symposium on.

20) J. Rosenberg, et al., "RFC 3261 SIP: Session initiation protocol", 2002. Available at: www.ietf.org

21) Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," ACM SIGCOMM Computer Communication Review, vol. 31, pp. 38-47,2001.

22) Douligeris C, Mitrokotsa A. DDoS attacks and defense mechanisms: classification and state-of-the-art.Computer Networks 2004; 44(5): 643–666.

23) Harrison K, White G. A taxonomy of cyber events affecting communities. In System Sciences (HICSS),2011 44th Hawaii International Conference on. IEEE, 2011.

24) D. C. Wyld, et al., "Trends in Network and Communications," International Conferences, NeCOM, 197: Springer, 2011.

25) M. Zelkowitz, "New programming paradigms," vol. 64, Academic Press, 2005.

26) Glenn Carl and George Kesidis et al., Denial-of-Service Attack-Detection Techniques, Published by the IEEE Computer Society, IEEE INTERNET COMPUTING, JANUARY • FEBRUARY 2006

27) D. Moore, G.M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," Proc. Usenix Security Symp., Usenix Assoc., 2001; http://citeseer.ist.psu.edu/moore01inferring.html.

28) P. Barford et al., "A Signal Analysis of Network Traffic Anomalies," Proc. ACM SIGCOMM Internet Measurement Workshop, ACM Press, 2002, pp. 71–82.

29) R.B. Blazek et al., "A Novel Approach to Detection of 'Denial-of-Service' Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods," Proc.IEEE Workshop Information Assurance and Security, IEEE CS Press, 2001, pp. 220–226.

30) H. Wang, D. Zhang, and K. Shin, "Detecting SYN Flooding Attacks," Proc. 21st Joint Conf. IEEE Computer and Comm. Societies (IEEE INFOCOM), IEEE Press, 2002, pp.1530–1539

31) Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 51(12), 3448-3470.

32) Xie, Y., & Yu, S.-Z. (2009). Monitoring the application-layer DDoS attacks for popular websites. Networking, IEEE/ACM Transactions on, 17(1), 15-25.

33) Yatagai, T., Isohara, T., & Sasase, I. (2007). Detection of HTTP-get flood attack based on analysis of page access behavior. Paper presented at the communications, computers and signal processing, 2007. Pacrim 2007. IEEE pacific rim conference on.

34) Lin, H., Lee, C.-Y., Liu, J.-C., Chen, C.-R., & Huang, S.-Y. (2010). A detection scheme for flooding attack on application layer based on semantic concept. Paper presented at the computer symposium (ics), 2010 international.

35) Choi, s., Kim, I.-K., Oh, J.-T., & Jang, J.-S. (2012). Aigg threshold based HTTP get flooding attack detection. In Information security applications (pp. 270-284). Springer.

36) Pimentel, A., Clifton, D. A., Clifton, L.,& Tarassenko, L. (2014). A review of novelty detection. Signal Processing, 99, 215–249. doi:10.1016/j.sigpro.2013.12.026

37) Rexroad, B., & van der Merwe, J. (2010). Network security–A service provider view. In Guide to reliable internet services and applications (pp. 447-515). Springer.

38) Owezarski, P. (2009). Implementation of adaptive traffic sampling and management, path performance. Academic Press.

39) Wen, S., Jia, W., Zhou, W., Zhou, W., & Xu, C. (2010). Cold: Surviving various application-layer DDoS attacks that mimic flash crowd. Paper presented at the network and system security (NSS), 2010 4th international conference on.

40) Kandula, S., Katabi, D., Jacob, M., & Berger, A. (2005). Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds. Paper presented at the 2nd conference on a symposium on networked systems design & implementation.

41) Stavrou, A., Cook, D. L., Morein, W. G., Keromytis, A. D., Misra, V., & Rubenstein, D. (2005). Web so: An overlay-based system for protecting web servers from denial of service attacks. Computer Networks, 48(5), 781-807.

42) Gummadi, R., Balakrishnan, H., Maniatis, P., & Ratnasamy, S. (2009). Not-a-bot: Improving service availability in the face of botnet attacks. Paper presented at the NSDI.

43) Djalaliev, P., Jamshed, M., Farnan, N., & Brustoloni, J. (2008). Sentinel: Hardware-accelerated mitigation of bot-based DDoS attacks. Paper presented at the computer communications and networks, 2008. Icccn'08. 17th international conference on.

44) M. Glenn, "A summary of dos/DDoS prevention, monitoring and mitigation techniques in a service provider environment," SANS Institute, 2003.

45) J. Molsa, "Effectiveness of rate-limiting in mitigating flooding DOS attacks," In International Conference on Communications, Internet, and Information Technology, pp. 155-160, 2004.

46) M. El-Soudani and M. A. Eissa, "Cooperative defense Firewall Protocol, "In Security and Privacy in the Age of Uncertainty, pp. 373-384, 2003.

47) Lua and K. C. Yow, "Mitigating DDoS attacks with transparent and intelligent fast-UX swarm network," Network, IEEE, vol. 25, no. 4, pp.28-33, 2011.

48) Zebari, Rizgar R., et al. "Distributed denial of service attack mitigation using high availability proxy and network load balancing." 2020 International Conference on Advanced Science and Engineering (ICOASE). IEEE, 2020.

49) Osanaiye, Opeyemi, Kim-Kwang Raymond Choo, and Mqhele Dlodlo. "Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework." Journal of Network and Computer Applications 67 (2016): 147-165.

50) Bhardwaj, Hanshi, et al. "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions." Computer Science Review 39 (2021): 100332.

51) Taylor, Omer Easier, and Muhammad Nazir Marson. "Collaborative detection and mitigation of distributed denial-of-service attacks on software-defined network." Mobile Networks and Applications 25.4 (2020): 1338-1347.

52) Devi, BS Kiruthika, and T. Subbulakshmi. "A comparative analysis of security methods for DDoS attacks in the cloud computing environment." Indian Journal of Science and Technology 9.34 (2016): 1-7.