DDOS PREVENTION SYSTEM USING AWS SERVERLESS-ARCHITECTURE

A Report submitted
in partial fulfilment for the Degree of

B. Tech
in
Computer Engineering
by

Chanakya Patil (07)
Aryan Deshmukh (08)
Anamay Brahme (12)
Asad Vathare (14)

pursued in

Department of Science and Technology
Vishwakarma University

To

VISHWAKARMA UNIVERSITY

PUNE

November, 2023

CERTIFICATE

This is to certify that the project report entitled **DDOS Prevention System Using AWS Serverless-Architecture** submitted by **Chanakya Patil, Aryan Deshmukh, Anamay Brahme, Asad Vathare** to the Department of Computer Engineering, Science Technology, Pune, in partial fulfilment for the award of the degree of **B. Tech in Computer Engineering** is a bona fide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

Prof. Noshir Tarapore
Supervisor
Department of Science and Technology

Pune, Counter signature of HOD with seal
November, 2023

DECLARATION

I declare that this project report titled **DDOS Prevention System Using AWS Serverless-Architecture** submitted in partial fulfilment of the degree of **B. Tech in Computer Engineering** is a record of original work carried out by me under the supervision of **Prof. Noshir Tarapore**, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Pune – 411056
22 Nov. 23

ACKNOWLEDGEMENTS

We are glad for the opportunity we got in this semester and are concluding Project Stage-2.

We want to convey our heartfelt gratitude to Prof. Noshir Z. Tarapore for his guidance and unwavering support in completing our project.

This project report could not have been completed without the constant and cooperative efforts of the group members Aryan Deshmukh, Anamay Brahme, Asad Vathare and Chanakya Patil

We also thank Vishwakarma University's Faculty of Science and Technology for providing us with this opportunity to present our study and research to the industry experts.

Last but not least, we would like to thank our connections from the industry who guided us, friends who supported us, and respondents for their active participation & encouragement and willingness to spend time on our project.

Chanakya Patil - 202000542
Aryan Deshmukh - 202000584
Anamay Brahme – 202000702
Asad Vathare – 202000966

ABSTRACT

The cloud is referred to as the servers or endpoints that can be accessed and controlled remotely. By using cloud computing, users and companies do not have to manage and take care of the physical servers themselves or even run the software applications on their own machines. The advantages of using cloud enables the users to access the same files and applications, because the computing and storage takes place on servers in large data centres, instead of storing locally on the user device itself, saving a lot of costs and manpower. Amazon Web Services is a leading cloud computing platform provided by Amazon. It's incredibly useful for individuals and businesses for several reasons. AWS offers scalability, which allows users to easily adjust resources to match their demands, which reduces costs and improves performance. Since AWS has a cost-effective pay-as-you-go model, you only pay for what you use. With data centres that are globally distributed, AWS ensures high reliability and availability of resources, ideal for building resilient applications. Tools like IAM (Identity Access Management) are used for resource access control and encryption for data protection, which increases security.

A Distributed Denial of Service Attack widely known as DDoS attack is a type of cyberattack that uses up the resources of the web server in order to disrupt the regular functioning of the website. DDoS is the malicious attempt to make resources unavailable for the intended users temporarily. In DDoS attacks, multiple compromised computers known as “bots” are used to increase the attack intensity. These computers are part of a botnet which is a network of computers that is controlled by hackers.

To mitigate DDoS attacks, organizations employ multiple strategies. Firewalls and Access Control Lists (ACLs) are used to control the traffic reaching applications, while rate limiting techniques set restrictions on incoming traffic. These measures collectively enhance network security and protect against malicious traffic.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS/ NOTATIONS/ NOMENCLATURE

Abbreviations:

DoS: Denial-of-Service

DDoS: Distributed Denial-of-Service

AWS: Amazon Web Services

FaaS: Functions as a Service

API: Application Programming Interface

VPC: Virtual Private Cloud

SDK: Software Development Kit

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

DNS: Domain Name System

Notations:

Lambda: Refers to AWS Lambda, a serverless computing service.

DynamoDB: Amazon's NoSQL database service.

CloudWatch: AWS service for monitoring and managing applications and resources.

API Gateway: AWS service that creates, manages, and secures APIs.

SNS: Simple Notification Service, an AWS messaging service.

REST: Representational State Transfer, a software architectural style for web services.

Nomenclature:

Serverless Architecture: A cloud computing execution model where the cloud provider dynamically manages the allocation of machine resources.

Microservice Architecture:

An architectural style where applications are structured as a collection of loosely coupled services.

Firewall:

A network security system that monitors and controls incoming and outgoing network traffic.

Intrusion Detection System (IDS):

A system that monitors network or system activities for malicious activities or policy violations.

Botnet:

A network of private computers infected with malicious software and controlled as a group without the owners' knowledge.

CHAPTER 1
INTRODUCTION

1. Introduction -

The digital world relies heavily on web applications and online services, which makes them highly vulnerable to Distributed Denial of Service (DDOS) attacks, if not configured properly. DDoS attacks can disrupt these services, leading to economic losses and damages to reputation. Traditional DDoS prevention methods are often costly and complex. To address this challenge, this research explores the potential of AWS Serverless Architecture as a cost-effective and scalable solution for DDoS attack prevention. By adopting this approach, organizations or companies can boost their ability to detect and mitigate DDoS attacks, in real-time.

This paper examines the practical implementation of AWS Serverless Architecture for DDoS prevention, offering insights and guidelines to bolster cybersecurity defences. The goal is to make online services more secure and resilient in the face of persistent threats.

1.1. Motivation -

DDoS attacks present a substantial risk to web servers, leading to service interruptions, downtime, and potential financial setbacks for businesses.

This project's drive is to investigate an improved and scalable approach in countering frequency-based DDoS attacks. Utilizing serverless architecture removes the necessity of server management and provisioning, enabling automatic scalability and a more economical deployment.

1.2 Aim and Scope Objectives -

Aim:

The aim of this project is to create a serverless architecture that can protect web servers from frequency-based distributed denial-of-service (DDoS) attacks.

Scope and Objectives:

1. Develop a serverless architecture using Amazon Web Services (AWS) resources, including Lambda serverless functions, API Gateway, CloudWatch, DynamoDB, and Simple Notification Service (SNS).
2. Monitor the frequency of incoming HTTP requests to identify potential DDoS attacks.
3. Trigger an alarm when the invocation frequency surpasses the threshold, indicating a potential DDoS attack.

The ultimate goal is to provide web server administrators with an efficient and reliable defence mechanism against frequency-based DDoS attacks, reducing the risk of service disruptions, minimizing downtime, and ensuring a smooth user experience.

CHAPTER 2

LITERATURE SURVEY

Existing systems for DDoS prevention typically involve the use of specialized hardware appliances or dedicated software solutions. These systems often rely on traditional network-level defences, such as firewalls and intrusion detection systems, to detect and mitigate DDoS attacks. They employ various techniques, such as rate limiting, traffic analysis, and IP blocking, to identify and block malicious traffic.

In contrast, DDoS prevention using AWS serverless architecture takes advantage of the cloud computing paradigm and AWS services, specifically AWS Lambda. With serverless architecture, the focus shifts from hardware-based solutions to a more scalable and cost-effective approach. AWS Lambda allows users to define and invoke small pieces of custom code, known as serverless functions, that are executed in a dedicated server environment without the need for managing infrastructure. This eliminates the overhead of launching virtual machines and provides the flexibility to scale up rapidly when an attack occurs.

The key difference lies in the scalability and resource management. Existing systems often require dedicated hardware or software resources that may need to be provisioned in advance or scaled manually during an attack. In contrast, AWS serverless architecture automatically handles resource allocation, scaling, and load balancing, allowing for rapid scaling by several orders of magnitude when needed. This dynamic scalability ensures that the system can handle the increased traffic during an attack without incurring excessive costs during normal operation.

By leveraging AWS serverless architecture, organizations can benefit from a highly scalable and cost-efficient approach to DDoS prevention. The on-demand nature of serverless computing allows for efficient resource utilization and reduces the operational burden associated with maintaining and managing dedicated hardware or software solutions.

CHAPTER 3

PROBLEM STATEMENT

Web servers are vulnerable to distributed denial-of-service (DDoS) attacks, where an overwhelming volume of HTTP requests are sent to the server, causing service disruptions and potential financial losses.

Traditional methods of mitigating DDoS attacks often involve resource-intensive infrastructure and complex configurations, making them costly and challenging to implement.

The problem addressed is the need for an efficient and scalable solution to protect web servers from HTTP-based DDoS attacks using a serverless architecture.

CHAPTER 4

PROJECT REQUIREMENT SPECIFICATION [HW/SW]

Hardware Requirement Specifications:

1. Laptop/Computer with 4GB RAM and Mobile Device or more
2. Chrome/Firefox/Edge Browser with latest version
3. Stable Internet Connection [min. 5Mbps]

Software Requirements Specification:

1. Amazon Web Services (AWS) Account:
2. Programming Language: Python 3.9
3. AWS Software Development Kit (SDK)
4. IDE/Text Editor

CHAPTER 5
SYSTEM ARCHITECTURE

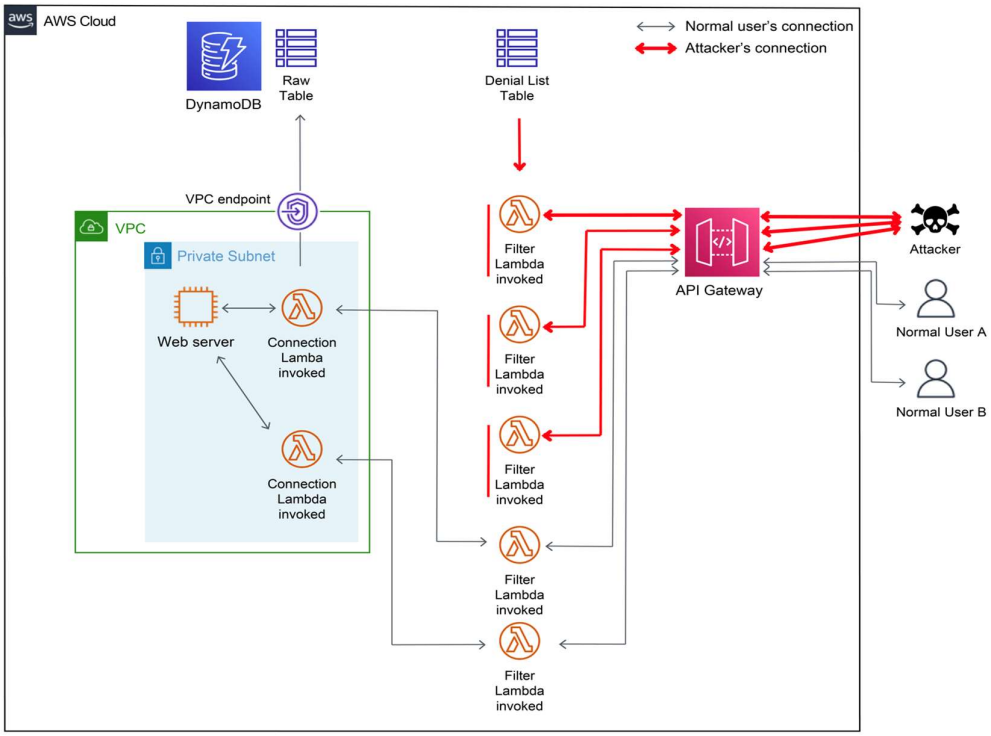

Figure 1

5.1. Summary of the design:

In this project, we are using AWS Serverless Architecture to prevent and mitigate a DDoS Attack. Here we will be writing three lambda functions: filter lambda, connection lambda and alarm lambda. Filter lambda will check and filter all the incoming traffic. Connection lambda will be used as a medium that connects the webserver after the filtration.

Alarm lambda will parse the Raw table in order to identify the high-frequency requesters. AWS CloudWatch is a monitoring and observability service provided by Amazon Web Services. We will be hosting the web server on VPC i.e., Virtual Private Cloud. We will be using AWS API Gateway Service to act as a communication medium between the Internet and the web browser.

Filter Lambda will be integrated into API Gateway, so that it monitors all the incoming requests. And there will be two DynamoDB tables: Raw Table and Denial List Table. Raw Table will be used to record all the incoming requests information and Denial List Table will act as a blacklist table, which stores the blacklisted/banned malicious IPs.

We will be using an AWS Simple Notification Service which will send an email notification to the admin when the alarm goes off.

CHAPTER 6

HIGH LEVEL DESIGN OF THE PROJECT

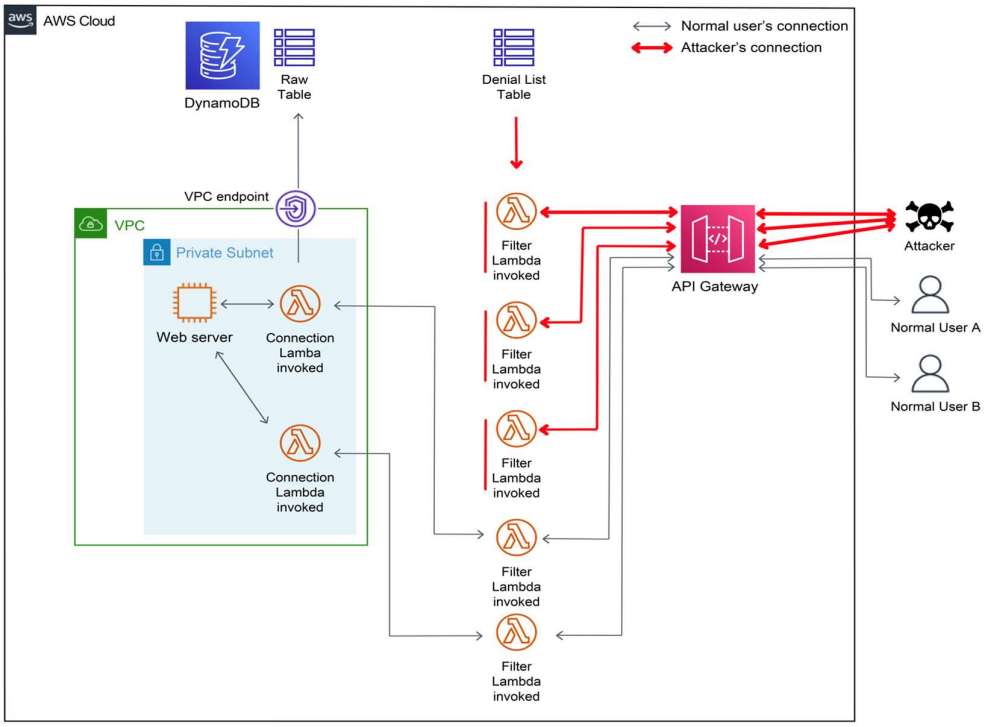

Figure 2

CHAPTER 7

SUMMARY OF IMPLEMENTATION

Differentiation between a Normal connection and an Attacker's connection to evaluate the system for the behavioural changes for possible outcomes to test the architecture.

7.1 Normal Connection:

1. Normal User Request: A legitimate user initiates an HTTP request from their device to access a web service.
2. API Gateway: The request is directed to an API endpoint created by the AWS API Gateway service. The API follows the REST style and supports HTTPS for secure communication.
3. Filter Lambda: For every request passing through the endpoint, the Filter Lambda function is invoked. It checks the request against a Denial List DynamoDB table to determine if the request should proceed.
4. Connection Lambda: After passing the filter check, the request is passed to the Connection Lambda function. Connection Lambda sends the request to the Web server instance located in a private subnet within a Virtual Private Cloud (VPC). Additionally, the request information is recorded in the Raw table of DynamoDB.
5. Response Flow: The response from the Web server travels back in the reverse direction. It passes through the Connection Lambda, Filter Lambda, API endpoint, and is finally returned to the user, ensuring a secure response flow.

7.2. Attacker’s Connection:

1. Attacker Request: An attacker sends an HTTP request, similar to a normal user, to the API endpoint.
2. Filter Lambda Check: The request is intercepted by the Filter Lambda function, which checks the request against the Denial List DynamoDB table.
3. Denial List Check: If the attacker's IP address and user agent are not listed in the Denial List, the request proceeds to the Connection Lambda.
4. Connection Lambda and Web Server: The Connection Lambda forwards the request to the Web server instance within the private subnet of the VPC, just like in the normal connection scenario.
5. Alarm Trigger: If the invocation frequency of the Connection Lambda exceeds a threshold, an AWS CloudWatch alarm is triggered.
6. Alarm Lambda Response: The triggered alarm invokes the Alarm Lambda function, which parses the Raw table to identify high-frequency requesters. The identified requesters are added to the Denial List table.
7. Denial of Attackers: As a result of the alarm response, requests associated with high-frequency attackers are blocked from reaching the Web server instance. However, normal user requests remain unaffected.
8. Continuous Monitoring: If the attacker continues the DDoS attack by bombarding the endpoint with different source IP addresses and/or user agents not yet listed in the Denial List, those requests can be captured in the next alarm cycle. The CloudWatch alarm typically takes a few minutes to trigger after high-frequency invocations start.
9. Iterative Blocking: The process of identifying and blocking high-frequency attackers through the Alarm Lambda and Denial List table may require several rounds, depending on the number of IP addresses and user agents used, their frequency of change, and how they are distributed over time.

CHAPTER 8
CONCLUSION

In conclusion, the described project implements a robust system architecture for protecting against Distributed Denial of Service (DDoS) attacks. By utilizing AWS services such as API Gateway, Lambda functions, DynamoDB, and CloudWatch, the system effectively filters and monitors incoming HTTP requests, allowing normal user requests to reach the web server while identifying and mitigating high-frequency attackers.

Through the continuous monitoring and iterative blocking process, the system can effectively mitigate DDoS attacks by blocking requests from known high-frequency attackers. The architecture distinguishes between normal user connections and attacker connections based on the combination of source IP address and user agent, enabling accurate identification and targeted blocking. Overall, this project demonstrates a comprehensive approach to protecting web services from DDoS attacks by leveraging AWS services and implementing an intelligent workflow. By promptly detecting and mitigating attacks, the system ensures the availability and reliability of the web service for legitimate users.

Existing DDoS solutions should definitely be considered for a fast and reliable strategy. For example, an AWS Web Application Firewall (WAF) could be attached to the API Gateway endpoint to block known attack sources and apply other filtering rules. AWS also provides a powerful DDoS prevention service called AWS Shield Advanced that protects against HTTP as well as UDP reflection flood, SYN flood and DNS query flood attacks (Priyam, 2018).

AWS Shield Advanced is expensive (e.g., $3,000 monthly fee plus data transfer), but its response time is less than seconds. Note that a full DDoS load test was not performed on the Lambda-based architecture described here. The purpose of this project was to come up with a design that works on a small scale as a proof of concept, and not intended to replace existing or recommended DDoS response services.

REFERENCES

[1]Soohyun Lee (May 10 2021), Conference details, date, year, volume. A serverless architecture for frequency-based HTTPrequest filtering against distributed denial-of-service(DDoS) attacks

[2]Ajay Singh Chauhan (2018) “DoS and DDoS attacks.” Practical network scanning. Packt publishing.

[3]Cade Metz (2009) “DDoS attack rains down on Amazon cloud” The Register. Retrieved from

https://www.theregister.com/2009/10/05/amazon_bitbucket_outage/

[4]Nick Galov (2021, Jan 16) "39 Jaw-Dropping DDoS

[5]Specht and Lee (2004) “Distributed Denial of Service: taxonomies of attacks, tools and countermeasures” ISCA PDCS. 543–550

[6]Ahmed Bakr ,Abd El-Aziz Ahmed ,Hesham A. Hefny (2019) ,”A Survey on Mitigation Techniques against DDoS Attacks on Cloud Computing Architecture “ Vol. 28, No. 12, (2019), pp. 187-200