DDOS Protection System for Cloud: Architecture and Tool

Venkat Deepak J[1], Ashish S[2], A Yuvaraja[3], Chitragar Rajesh[4], Siddharth Bej[5], Dr. Nagaraja S R[6]
[1,2,3,4,5]Department of CSE, SoCSE, Presidency University, Yelahanka, Bengaluru, Karnataka, India
[6]Associate Professor, Presidency University, Yelahanka, Bengaluru, Karnataka, India

Abstract-Currently, the highly interconnected world is plagued by service interruptions, economic loss and reputational harm stemming from DDoS attacks targeted at cloud infrastructures. A surge in malicious traffic is often left unmonitored until the last minute, making ancient defense structures incapable of responding in time.
Achieving DDoS detection and traffic monitoring is built around a DDoS attack recognition algorithm which focuses on a crucial range of traffic metrics such as Network In, Packets In and CPU Utilization and other parameters. Other than real-time monitoring, this system utilizes features Preemptively watching over systems and using customized Python scripts and Apache JMeter to simulate DDoS attacks ensures verification of detection capabilities. As well as including detection/APT, this project has response tactics activated. In the aftermath of detection, automation scripts will be developed in Python to thwart the IPs perpetrating DDoS attacks which will impede subsequent multiple entrances into the system that are attempted. Rate limiting algorithms will also be added to assist in traffic spiking control and increased mitigation towards automatically block IP Addresses on the infrastructure.

I. INTRODUCTION

In the last few years cloud-based infrastructures have become an integral part of services and businesses around the globe. As the usage of cloud systems increases, they become more susceptible to facing multiple security threats, the worst being the Distributed Denial of Service (DDoS) attacks which is highly disruptive. Flooding the servers with traffic to consume all available resources and block the processing of legitimate user requests is the objective of DDoS attacks. The service is interrupted and at the same time the organization suffers severe financial losses, reputational damage and customer satisfaction declines.

Due to the uncertainty and scale of threats of modern DDoS attacks, existing solutions have difficulty using them in real time with the attack identification and stopping. Manual action taken during the attack is slow and ineffective, which indicates the need for intelligent automated systems that offer real time detection and response.

This project proposes a solution to the problem of DDoS attacks by installing the visualization and monitoring tool known as Grafana on an AWS EC2 instance which enables real-time monitoring and detection of DDoS attacks. It aims to issue suspicious activity warning flags by collecting and analyzing traffic metrics such as "Network In," "Packets In," and CPU utilization. The system makes use of a Python script that simulates web traffic together with Apache JMeter which generates DDoS-like traffic to train the system on distinguishing between malicious and healthy traffic patterns.

Beyond mere detection, the project sets up a framework for automatic mitigation, which enables the system to automatically counteract detected attacks. To prevent the same DDoS attacks from recurring, we plan to implement scripts written in Python that will automatically and proactively IP block identified malicious addresses. Later phases will also incorporate rate-limiting strategies aimed at not only controlling the inflow of requests but also reinforcing defence against automated and bot-driven traffic surges.

Ensuring that services hosted in the cloud remain secure while remaining dependable, and resilient to changing cyber threats, the project aims to take an integrated scalable approach in attempts to automate the primitive defences and initiate early detection of DDoS attacks.

II. PROPOSED METHODOLOGY

This approach emphasizes the design of a self-contained mitigation mechanism and a real-time monitoring tool meant to mitigate the impact of Distributed Denial-of-Service (DDoS) attacks on cloud-hosted infrastructure.

The AWS account contains an EC2 instance with a Grafana service that provides real-time monitoring and visualization of network activity. Identifying features that could indicate the presence of DDoS attacks includes important server parameters such as Network In, Packets In, and CPU Utilization.

In order to generate realistic environments, authentic user traffic is emulated using a Python-based traffic generator, which simulates routine tasks on the website. Concurrently, Apache JMeter is configured to simulate DDoS attacks by issuing a disproportionate number of requests to the server. The intelligent detection and mitigation framework is fully operational under real-world conditions due to the combination of legitimate and malicious traffic.

With alert thresholds set for important performance indicators, any deviation from established safe operating limits indicates a possible attack and necessitates an investigation right away. The monitoring configuration makes sure that any odd increases in CPU usage, packet rates, or network traffic are quickly detected so that prompt actions can be undertaken. In this case, automated actions will be undertaken to stabilize the affected system.

The framework is developed for the future by combining rate-limiting strategies. Rate limiting is a process of reducing the manual effort by limiting the number of requests that can be sent from a single IP address, so that the SOC analyst does not need to block every single IP manually.

III. OBJECTIVES

This project's goal is to design a real-time scalable, cloud-based system capable of DDoS attack detection and scrubbing. The project works on the infrastructure of Amazon Web Services (AWS), which hosts EC2 instances, Elastic Load Balancer (ELB), Auto Scaling Groups – all of which are monitored via CloudWatch, ensuring availability and performance of services even under very high traffic conditions. To simulate attack behavior and system responsiveness, the system will mimic normal and malicious traffic with Python scripts and Apache JMeter, respectively.

The system is designed to include an anomaly detection system that tracks request rates and the associated network to check for any unusual activity that could signal an attack. The system undertakes mitigation measures once an attack is detected which includes IP-based blocking through firewall scripts written in Python. Also, auto-scaling is active which means that EC2 instances will be added or removed depending on the current load and system resources.

Normal and malicious traffic simulation can be done effectively with Python. Detection of abnormal traffic behavior coupled with resource utilization beyond a certain threshold will result in an alert being generated by AWS CloudWatch, which is actively monitoring the system.

The project investigation extends beyond core detection and mitigation features into future improvements like rate limiting for suspicious IPs and utilizing machine learning algorithms for more precise threat classification. As a whole, this project seeks to develop a malleable, cost-efficient, and secure cloud solution that defends against DDoS attacks which is suitable for modern web services infrastructures.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The system is proposed to use a blend of open-source applications and private scripts for simulation, detection, and mitigation of Distributed Denial of Service (DDoS) attacks on a cloud-based infrastructure. To robustly defend against attack scenarios, the architecture integrates traffic generation, real-time monitoring, traffic anomaly detection, and automated mitigation strategies all working together.

To create synthetic traffic data, two concurrent methods are set in place. A Python program is developed to generate normal traffic to an EC2 hosted website to simulate actual user activity. Simultaneously, large scale request flooding to the server is performed by Apache JMeter, which emulates DDoS attack traffic. The accuracy of monitoring and mitigation solutions is enhanced by the use of dual-traffic generation techniques which allow thorough testing under real-world conditions.

Monitoring real-time network metrics like CPU utilization, packets in, and network in are visualized on Grafana, which is integrated with the EC2 server. These metrics are crucial in identifying abnormal spikes that may indicate a DDoS attack. Custom alerting thresholds along with dashboards enable rapid event detection for set alerts that notify system admins when thresholds are crossed.

To effectively steer incoming traffic, an AWS load balancer is integrated with the EC2 instance. The Load balancer not only distributes incoming traffic, but also mitigates unforeseen spikes caused during DDoS simulation attacks from being absorbed. This guarantees that reliable service performance will be maintained even during extreme attacks, ensuring high availability and preventing a single server from becoming a performance strainer. The load balancer works in conjunction with backend mitigation scripts to systematically identify and eliminate the source of malicious traffic without affecting genuine user access.

An abuse detection system is configured on the IP pattern recognition task and data stream classifier that controls wherein logic means applied on them. The monitored traffic patterns on incoming via a Python script run in the background alongside a continuous IP checker are polished along with compiling the questionable behaviours being extracted. Firewall rules are modified on the fly to stop the identified IPs from accessing the server. This guarantees that an attacker will be stopped from carrying out the attack as soon as they are identified.

The system implements its mitigation strategy when an anomaly is recognized. A malicious IP address is collected by a traffic detection Python script that analyses traffic patterns for suspicious activity. Through automation, the suspicious IP addresses are instantly added to the firewall and are forbidden from accessing the server in the future. This encourages resource saving. This blocking approach reduces the effects of ongoing attacks by resource exhaustion.

Besides, the improvements and design scalability were addressed in the same system. A rate-limiting control is set to be added to the server; this will restrict and control the number of requests to be accepted from a single IP in a predefined time frame. This approach will automate control without manual action being required, thus reducing the attack surface. Further, controlling abusive IPs even before full attack patterns are exposed will strengthen proactive measures.

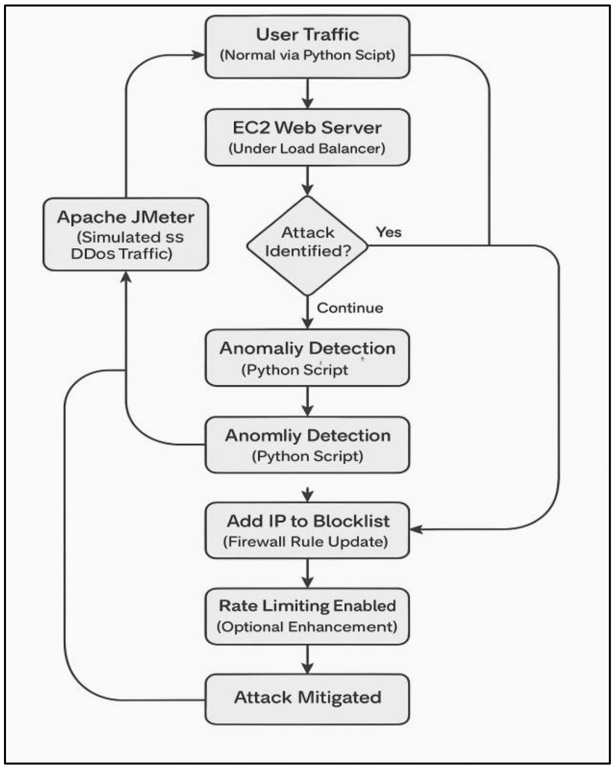


Fig. 1 – Project Architecture

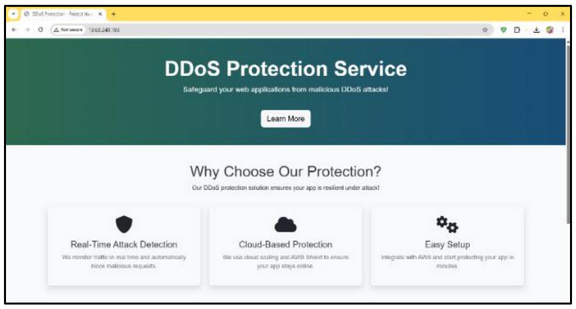
V. RESULTS AND DISCUSSIONS

The designed DDoS detection and mitigation system that needed to be implemented for an Apache JMeter test was successful in reclaiming service availability and ensuring minimal time to respond to an attack. In the course of controlled tests, Apache JMeter generated enormous volumes of unrealistic DDoS traffic aimed at the load balancer's AWS EC2 instance. The load balancer utilized ensured traffic balancing, which meant that no single server got overloaded and maximum uptime was assured despite intense attack patterns.
The traffic generator in Python-created streams of typical user traffic which segregated the genuine actions from the malevolent ones. The IPs that were classified to be of malicious nature due to the distinct request patterns were effectively filtered by the anomaly detection module that employed heuristic analysis and traffic rate thresholds. As soon as said IP addresses were flagged as malicious, there was a temporary dip in the identified threatening streams of information due to a wider net being casted through the automated python mitigation script, altering firewall rules to block said IP addresses.

The system proved its capabilities for extension by altering resources during times of peak traffic. During simulated attacks, auto-scaling policies executed across multiple layers, where CloudWatch monitored CPU and network usage and added new EC2 instances when necessary. Network input and CPU usage metrics served as triggers for CloudWatch Alarms, which in turn, prompted scaling policies to alleviate the system. That ensured, regardless of how many attacks were executed, the system's response times and stability remained unyielded.

In general, answers from the experiments indicate strong support for the architecture's flexibility. These attempts to defend the system against DDoS attacks suggest that the feedback loop associated with traffic generation, monitoring, detection, and mitigation functions effectively. There is hope that future advancements with adaptive mechanisms of rate control in real time, along with traffic identification algorithms utilizing machine learning, will bolster accuracy in detection and the speed of mitigation. Due to its modular approach, which ensures adaptability to evolving attack patterns, the system stands as a viable answer to real-world implementation issues.

RESULT:


DDoS Protection Service

Safeguard your web applications from malicious DDoS attacks!

Learn More

Why Choose Our Protection?

Our DDoS protection solution ensures your app is resilient under attack!

Real-Time Attack Detection — We monitor traffic in real time and automatically block malicious requests.

Cloud-Based Protection — We use cloud scaling and AWS Shield to ensure your app stays online.

Easy Setup — Integrate with AWS and start protecting your app in minutes.

Fig. 2 - Homepage

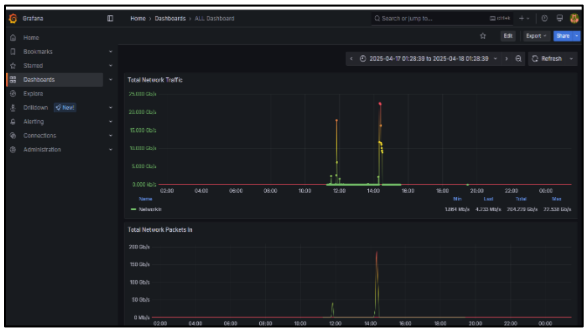
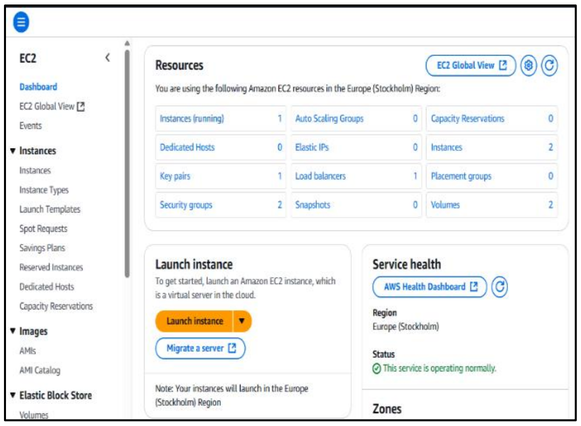


Fig. 3 – Grafana Dashboard


EC2

Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	1	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	2
Key pairs	1	Load balancers	1	Placement groups	0
Security groups	2	Snapshots	0	Volumes	2

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the Europe (Stockholm) Region

Service health

Region: Europe (Stockholm)

Status: This service is operating normally.

Fig. 4 – AWS Instance

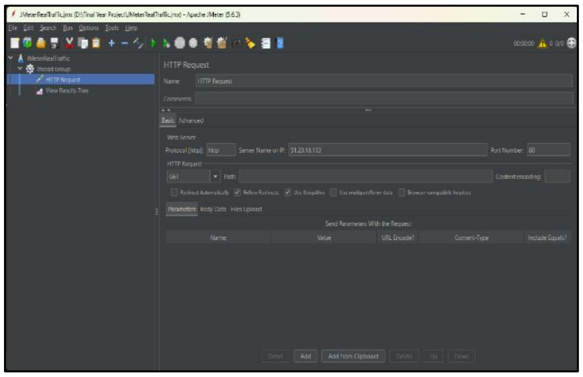


Fig. 5 – Apache Jmeter

VI. CONCLUSION

This was the case for us as well. Amazon web services were harnessed in the design and implementation of the system for identification and prevention of distributed denial of service attacks (DDos). We developed an architecture which employed the use of Apache JMeter with scripts written in Python to simulate masquerade and legit traffic respectively. These were aimed at showcasing how traffic changes in real life depending on load scenarios. The incoming traffic was successfully amalgamated to several EC2 instances aided by the Elastic Load Balancer (ELB) which maintained service availability even under a simulated attack being scrutinized.

To assist the system in distinguishing between legitimate and suspicious traffic, anomaly detection AI scripts were implemented to monitor changes in request rates. IP blocking attempts via firewalls written in Python were performed automatically during the identified attacks. Also, to enable prompt action against recurring or gradually escalating attacks, rate limiting was proposed as an improvement for the system. For example, launching extra EC2 instances to maintain a set level of service or to control resource allocation reduces response time ensured high availability. Managing traffic with auto scaling groups, CloudWatch metrics and alerts provided real-time support for monitoring, acting as catalysts for scaling and defensive actions.

The architecture allows adding new components without impacting existing ones due to its modular structure. This enables easy future upgrades and replacements to keep the system flexible and future-ready. In later versions, we intend to enhance detection accuracy with machine learning models as well as include AWS WAF to allow for filtering some traffic more intelligently. We are advancing our understanding of cybersecurity and cloud infrastructure, but this project also illustrates the importance of having proactive defences in place given the active threat landscape.

VII. ACKNOWLEDGMENT

We express our deepest gratitude towards Presidency University, Bengaluru, for providing the resources and support necessary to undertake this project. Special gratitude goes to our guide, Dr. Nagaraja S. R., Associate Professor, School of Computer Science Engineering & Information Science, whose invaluable guidance, constructive feedback, and encouragement guided us to feasible implementation during the building of this project. His insights enhanced the quality of our framework, making it possible to meet our objectives.

We also extend sincere thanks to the department faculty, particularly Dr. S. P. Anandaraj, Head of the Department, and the Capstone Project Coordinators Dr. Sampath A. K., Dr. Abdul Khadar A., and Mr. Md. Zia Ur Rahman, for their continued guidance and timely advice throughout the various stages of the project. Their knowledge and passion for the success of students have greatly enhanced our experience of learning.

We are grateful to our peers, friends, and families for their unwavering encouragement and moral support. They have motivated us in every way to advance our first project to completion. We are also thankful for the cooperation and assistance extended by the administrative staff in getting the various project-related activities executed smoothly.

Finally, we would like to appreciate all the efforts of the DDOS Protection system for cloud development team for their hard work and dedication towards transformation of this concept into a reality. Such type of project would not have been possible without these efforts of all people joined in.

REFERENCE

[1] R. M. H. Fouladi and M. G. Sarhan, "A Survey on DDoS Attacks and Defense Mechanisms in Cloud Computing," *Int. J. Comput. Netw. Inf. Secur.*, vol. 13, no. 6, pp. 1–12, 2021.

[2] H. Wang, D. Zhang, and K. G. Shin, "Detecting SYN Flooding Attacks," *IEEE INFOCOM 2002*, vol. 3, pp. 1530–1539, 2002.

[3] A. S. Hafeez, et al., "Cloud-Based DDoS Detection Using Machine Learning and SDN," *IEEE Access*, vol. 8, pp. 132850–132860, 2020.

[4] M. Hussain, A. Almogren, and M. A. Jan, "An Intelligent DDoS Detection Framework for Cloud Environment Using Hybrid Feature Selection and Ensemble Classifier," *Comput. Mater. Contin.*, vol. 65, no. 3, pp. 2703–2719, 2020.

[5] Y. Zhang, et al., "DDoS Detection and Mitigation for Cloud-Based Services Using Machine Learning Techniques," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 1, pp. 42–54, 2020.

[6] G. Liu, et al., "High-Performance DDoS Detection and Mitigation System in Cloud Environment," *Int. Conf. Cloud Comput. Technol. Sci.*, pp. 311–318, 2019.

[7] S. M. Mousavi and M. St-Hilaire, "Early Detection of DDoS Attacks Against SDN Controllers," *2015 Int. Conf. Comput. Netw. Commun. (ICNC)*, pp. 77–81, 2015.

[8] M. Aamir and M. A. Zaidi, "Clustering-Based Semi-Supervised Machine Learning for DDoS Attack Detection," *Comput. Secur.*, vol. 101, p. 102034, 2021.

[9] K. Bhushan and S. Gupta, "An Overview of Cloud-Based DDoS Attacks and Their Mitigation Techniques," *Int. J. Comput. Appl.*, vol. 179, no. 14, pp. 6–11, 2018.

[10] Singh and N. Choudhury, "Auto-Scaling in Cloud Computing: A Survey," *Int. J. Comput. Appl.*, vol. 178, no. 7, pp. 20–25, 2019.

[11] Al Eroud and G. Karabatis, "Detecting Insider Threats Using Data Mining Techniques," *J. Intell. Inf. Syst.*, vol. 48, no. 3, pp. 641–662, 2017.

[12] R. M. Noor and W. H. Hassan, "Current Research on Internet of Things (IoT) Security: A Survey," *Comput. Netw.*, vol. 148, pp. 283–294, 2019.

[13] D. Arif, H. A. Shah, and M. F. Qureshi, "Towards an Efficient DDoS Detection Approach Using Hybrid Features and Ensemble Classifier in Cloud Environment," *IEEE Access*, vol. 9, pp. 115128–115143, 2021.

[14] J. Singh and S. K. Pandey, "A Review of DDoS Attacks Detection Mechanisms in Cloud Computing," *Int. J. Comput. Sci. Inf. Technol.*, vol. 10, no. 2, pp. 93–97, 2018.

[15] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 4, pp. 2046–2069, 2013.