CHAPTER 3
SYSTEM ARCHITECTURE AND DESIGN

Architecture and design of the proposed DDoS Protection System for Cloud using AWS and Machine Learning involves details on the layered system architecture, integration of anomaly detection mechanisms, and the deployment of cloud-based mitigation strategies. Its ability to mimic real-world traffic patterns while it identifies anomalies, and respond dynamically to DDoS attacks minimizes service disruption.

3.1 Layered Architecture Overview

The architecture of the DDoS Protection System is designed as modular and layered, hence scalable, maintainable, and adaptable. Each layer takes on certain roles to detect, analyze, and mitigate DDoS attacks so that it can easily manage traffic and resolve anomalies.

3.1.1 Traffic Data Generation Layer

This layer is used to mimic realistic traffic patterns. Python scripts generate synthetic traffic data that replicates realistic behavior of real-world web traffic. Such simulation helps create controlled environments in which to test anomaly detection methods.

3.1.2 Anomaly Detection Layer

The Anomaly Detection Layer utilizes the Isolation Forest algorithm from the scikit-learn library to point out anomalies in traffic pattern. This algorithm models what normal traffic is and sets apart anomalies that significantly deviate, indicating possible DDoS attacks. The anomaly detected is plotted using the matplotlib library for further thorough analysis.

3.1.3 Load Simulation and Testing Layer

Load testing is performed using Apache JMeter to simulate high traffic loads and test the resilience of the system. This layer configures the user load parameters, like concurrent users and request rates, to stress-test the server under various conditions that can be considered as DDoS conditions.

3.1.4 Mitigation Layer

The Mitigation Layer uses dynamic strategies against detected threats to ensure system stability. Key components such as Auto-Scaling allow the system to automatically launch or terminate EC2 instances in real-time based on the traffic load, thus providing adequate resources during traffic spikes, including DDoS attacks. The Load Balancer also plays a crucial role by spreading incoming traffic evenly across multiple instances, thus preventing one server from becoming overwhelmed. Together, these mechanisms ensure that legitimate traffic flows uninterrupted while malicious traffic is efficiently filtered, thus maintaining integrity and availability of the system.

3.1.5 Monitoring and Reporting Layer

AWS CloudWatch ties into the system and monitors key metrics, such as NetworkIn and CPU, and is configured to send alerts when thresholds are exceeded for real-time intervention during attacks.

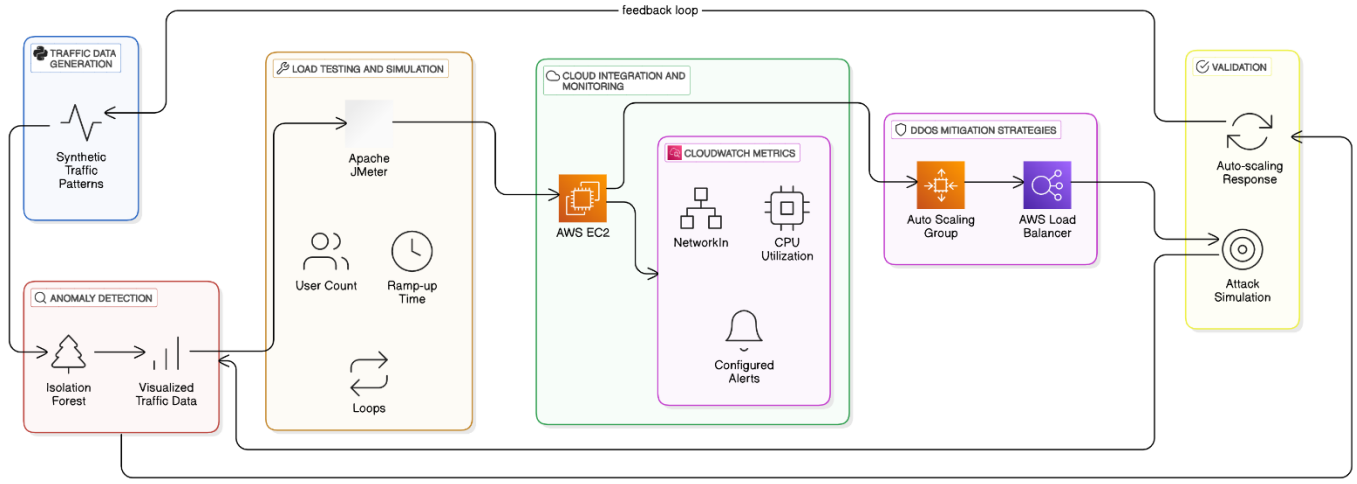

Fig. 3.1 Architecture Diagram

3.2 Core Components

3.2.1 Traffic Data Simulation and Anomaly Detection

The core component of the system, which is the anomaly detection model, is trained with synthetic traffic data. Employing the Isolation Forest algorithm, it detects outliers that are significantly different from learned baselines, indicating that they may be a DDOS attack. Visualization tool-supported analysis and validation enhance its true positive rate.

3.2.2 Load Testing Framework

Load testing is primarily done with the help of Apache JMeter. It produces different kinds of traffic patterns based on parameters such as user concurrency, ramp-up periods, and request intervals. It tests the system's response to stress and how well it can handle a variation in traffic loads.

3.2.3 Cloud-Based Mitigation

The system relies on the AWS services for scalability and resilience in traffic surges, especially DDoS attacks. Auto Scaling Groups automatically adjust the number of server instances based on traffic load, thereby ensuring there are enough resources available when traffic is high. Further, the Elastic Load Balancer (ELB) spreads incoming traffic across several instances, thus optimizing server performance and preventing bottlenecks. These features allow the system to adapt dynamically to varying traffic conditions and prevent server overload in the case of a DDoS attack, maintaining consistency in availability.

3.3 Scalability and Performance Considerations

3.3.1 Scalability

The solution is designed to handle the large-scale traffic with utmost ease. AWS Auto Scaling Groups ensure that additional resources are provisioned during the high-demand scenarios, thus not disrupting the service.

3.3.2 Low Latency Operations

Low latency is critical to real-time anomaly detection and mitigation. Using AWS edge services and optimized instance configurations, the system minimizes its response times to ensure on-time countermeasures for DDoS attacks.

3.3.3 Adaptability

The modular architecture is designed to allow easy incorporation of new detection algorithms and mitigation strategies. This would ensure that the system continues to be robust against the evolving attack vectors and support advancements in cloud-based technologies.

3.4 Integration with AWS Infrastructure

3.4.1 CloudWatch Monitoring and Alerts

AWS CloudWatch monitors all the essential metrics, such as traffic volume and server utilization. All the alerts are set so that whenever an anomaly or high traffic load is found, it sends a notification allowing for swift administrative action.

3.4.2 EC2 Deployment and Management

Scaling, reliability, and other requirements of cloud infrastructure are followed by the EC2 instance in which anomaly detection monitoring scripts are deployed. Due to this reason, in real-time, traffic can also be scaled up or scaled down automatically.

3.4.3 Legacy System Compatibility

This system has compatibility with any legacy communication protocol and also with previous cloud infrastructures. In this way, it could be deployed into any deployment environment without making much of architecture changes.

3.5 Validation of System Design

The validation of the proposed system is comprised of comprehensive testing to evaluate its effectiveness against DDoS attacks. High traffic loads are simulated using Apache JMeter to mimic real-world DDoS scenarios, giving a controlled environment to analyze the performance of the system. The Isolation Forest algorithm is tested for the accuracy of anomaly detection as its ability to identify any unusual patterns in the simulated traffic may indicate potential attempts of DDoS attacks.

Moreover, the mitigation strategy's effectiveness is further verified by monitoring the behavior of the AWS Auto Scaling and Load Balancing mechanisms within such scenarios. Such testing assures that the system adapts dynamically to the changing conditions of the attack and keeps consistent availability and service quality intact.

3.6 Summary

The proposed DDoS Protection System for Cloud utilizes AWS and Machine Learning to present an adaptive, scalable, and efficient mitigation of DDoS attacks. This system is built with layered architecture, combining traffic simulation, anomaly detection, and real-time mitigation strategies in order to protect cloud-based services. The modular design ensures compatibility with existing infrastructures and maintains resilience against evolving threats.