Towards an Automated Defense: A Novel Framework for Log-Based Threat Detection and IP Remediation in AWS

Suhaib Kashif
Dept. of Computer and Software Engineering
Information Technology University (ITU)
Lahore, Pakistan
suhaibkashif03@gmail.com

Ali Sher Afzal
Dept. of Computer and Software Engineering
Information Technology University (ITU)
Lahore, Pakistan
bsse23007@itu.edu.pk

Project Technical Report

Department of Computer and Software Engineering
Information Technology University (ITU)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

***Abstract*—In the contemporary cloud threat landscape, manual responses to web application attacks are insufficient due to the volume and velocity of incoming threats. This project introduces an automated defense framework deployed on Amazon Web Services (AWS) designed to detect and remediate common web threats, such as SQL Injection (SQLi) and Cross-Site Scripting (XSS), in near real-time. The architecture leverages AWS cloud-native services to create an event-driven pipeline. Application Load Balancer (ALB) access logs serve as the source of truth for incoming traffic. An event-driven detection pipeline, utilizing Amazon S3, CloudWatch Events, and AWS Lambda, parses these logs to identify malicious signatures. Subsequently, a response pipeline automatically remediates the threat by updating DynamoDB state tables and modifying VPC Security Groups and Network Access Control Lists (NACLs) to block offending IP addresses. This report details the architectural components, data flow, and security configurations of the proposed framework, demonstrating a significant reduction in Mean Time to Respond (MTTR).**

***Index Terms*—AWS, Cloud Security, Automated Response, Threat Detection, Lambda, Serverless, DevSecOps.**

I. Introduction

The migration of enterprise applications to cloud environments like AWS has expanded the attack surface for malicious actors. Web applications remain a primary target, with attacks such as SQL Injection (SQLi) and Cross-Site Scripting (XSS) consistently appearing in the OWASP Top 10 vulnerabilities. These attacks pose significant risks to data integrity, confidentiality, and service availability. Traditional intrusion detection systems (IDS) and manual log analysis often rely on human intervention to analyze alerts and implement blocks. This manual workflow results in a critical time gap between detection and remediation—the "mean time to respond" (MTTR)—during which an attacker can successfully exfiltrate data or compromise backend systems.

To address this latency, modern security postures must shift "left" towards automated, event-driven security operations (DevSecOps). This project proposes a novel framework titled "Towards an Automated Defense," which utilizes a serverless architecture on AWS to automate the entire lifecycle of threat detection and remediation. By removing the human bottleneck, the system can react to threats at machine speed.

The proposed solution leverages the elasticity of cloud resources. By ingesting Access Logs from an Application Load Balancer (ALB) and processing them through AWS Lambda functions, the system can identify malicious request patterns instantaneously. The framework further closes the security loop by automatically updating network-level defenses—specifically Security Groups and Network ACLs—to isolate attackers without human intervention. This report outlines the technical design, component interactions, and implementation details of this automated defense mechanism, demonstrating a scalable and cost-effective alternative to commercial security appliances.

II. Related Work

The domain of automated cloud security is an active area of research, driven by the limitations of manual security monitoring and the increasing sophistication of automated botnets.

Research by Ullah et al. [1] emphasizes the necessity of moving beyond signature-based detection in legacy systems towards anomaly-based detection in cloud environments. Their work highlights the critical role of centralized logging and deep learning in identifying sophisticated web attacks that evade simple firewall rules. Our project builds on this by implementing a lightweight, signature-based detection mechanism suitable for high-throughput environments.

The utilization of serverless computing for security automation is explored by Yan et al. [2]. Their work demonstrates how event-driven architectures, like AWS Lambda, can significantly reduce remediation latency compared to traditional server-based security tools. Unlike server-based tools that require constant maintenance and scaling, serverless functions instantly react to state changes within the cloud environment, providing a more responsive security posture.

Furthermore, the National Institute of Standards and Technology (NIST) guidelines on incident response [3] emphasize the importance of containment strategies during cyber incidents. Specifically, NIST recommends automated containment for high-confidence attack vectors to prevent lateral movement. Our framework directly addresses the containment phase by automating IP blocking at the network perimeter, aligning with established industry best practices for rapid incident response and adhering to the "Defense in Depth" principle.

III. Proposed Architecture Overview

The proposed framework is designed as a highly available, event-driven architecture on AWS. It is segregated into four distinct logical flows: The Attack Path, the Detection Pipeline, the Response Pipeline, and the Monitoring Layer.

The core principle is to monitor traffic hitting the public-facing Application Load Balancer (ALB), push logs asynchronously to storage, trigger analysis functions upon log arrival, and execute remediation functions if threats are confirmed. The use of serverless components (S3, Lambda, DynamoDB) ensures the security infrastructure scales automatically with traffic volume, decoupling the security processing from the main application logic to prevent performance degradation.

Table I summarizes the key AWS services utilized in this architecture.

TABLE I: AWS Services Utilized and Their Functions

AWS Service	Function in Framework
EC2	Hosts Web App & Attacker simulations
VPC (SGs/NACLs)	Network isolation & blocking mechanism
ALB	Entry point & source of access logs
S3	Log storage & event trigger source
AWS Lambda	Serverless code for Detection & Response
DynamoDB	NoSQL state store for malicious IPs
CloudWatch	Event orchestration & monitoring

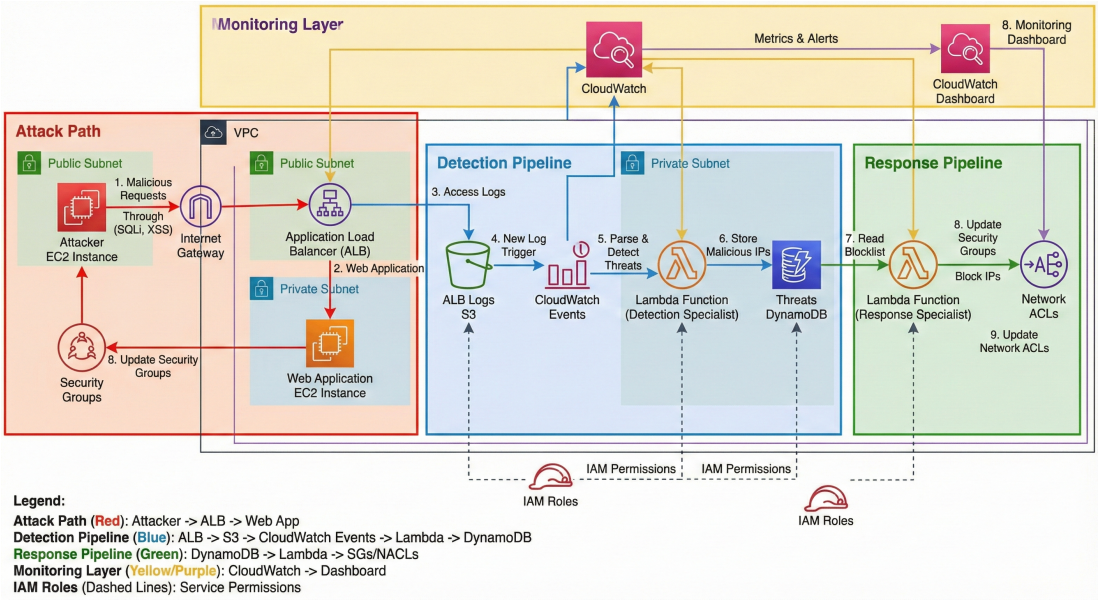

Fig. 1: Detailed Architecture Diagram of the Automated Defense Framework showing the interaction between the Attack Path, Detection Pipeline, Response Pipeline, and Monitoring Layer.

IV. Detailed System Design

This section details the step-by-step workflow of the automated defense framework, explaining the specific mechanisms used for detection and blocking.

A. The Attack Path (Red Flow)

The attack path represents the flow of both legitimate and malicious traffic entering the system.

1) **Infiltration:** An attacker, operating from an external EC2 instance (representing a compromised host or botnet), sends malicious requests. These requests typically contain SQLi payloads (e.g., `' OR 1=1--`) or XSS vectors (e.g., `<script>alert(1)</script>`) embedded in query parameters.
2) **Ingress:** The traffic passes through the Internet Gateway and reaches the Application Load Balancer (ALB) located in the Public Subnet. The ALB terminates the SSL/TLS connection and inspects the request headers.
3) **Forwarding:** The ALB forwards the request to the Web Application EC2 instance residing in the Private Subnet. If the application is vulnerable, this request initiates the exploit.

B. The Detection Pipeline (Blue Flow)

The detection pipeline operates asynchronously to analyze traffic. Crucially, this design ensures that security analysis does not add latency to the user experience (UX).

1) **Log Generation:** The ALB is configured to generate detailed access logs. These logs capture the requester IP, request URI, user agent, and response codes. The logs are delivered to a centralized Amazon S3 bucket in 5-minute intervals or when the log file size reaches a threshold.
2) **Event Trigger:** The act of writing a new log file to S3 generates an `s3:ObjectCreated:Put` event. This event is intercepted by CloudWatch Events (EventBridge), which serves as the glue between the storage and compute layers.
3) **Detection Execution:** CloudWatch Events triggers the "Detection Specialist" AWS Lambda function. This function retrieves the log file from S3, decompresses it (if gzipped), and iterates through log entries. It utilizes Regular Expressions (Regex) to match request URIs against a database of known attack signatures.
4) **State Update:** If a malicious pattern is identified, the Lambda function extracts the source IP address and writes it to the "Threats DynamoDB" table. This table acts as the single source of truth for active threats.

C. The Response Pipeline (Green Flow)

The response pipeline acts upon confirmed threats stored in the database, executing the remediation logic.

1) **Stream Processing:** A second AWS Lambda function ("Response Specialist") is connected to the DynamoDB table via DynamoDB Streams. This ensures the function is triggered only when a new item (threat) is successfully committed to the database.
2) **Update Security Groups (Stateful):** The function modifies the Security Group attached to the Web Application EC2 instance. It adds an inbound rule to explicitly deny traffic from the malicious IP. Security Groups are stateful, meaning response traffic is automatically handled, but they are applied at the instance level.
3) **Update Network ACLs (Stateless):** Simultaneously, the function updates the Network Access Control List (NACL) associated with the Public Subnet. It adds

a DENY rule with a priority number lower than the ALLOW ALL rule. This provides a stronger, subnet-level defense that blocks traffic before it even reaches the instance.

D. The Monitoring Layer (Yellow/Purple Flow)

Visibility into the security automation is crucial for auditability and debugging. The Lambda functions and the ALB report operational metrics (invocation counts, error rates, throttle events) to Amazon CloudWatch Logs. A custom CloudWatch Dashboard aggregates these metrics, providing a visual overview of detected threats, successful remediations, and overall system health to the security operations center (SOC).

V. Security and IAM Configurations

The integrity of this automation relies heavily on robust Identity and Access Management (IAM) roles. Adhering to the principle of least privilege is essential to prevent the security tool itself from becoming a vulnerability:

- **Detection Lambda Role:** This role is granted `s3:GetObject` permissions strictly for the log bucket and `dynamodb:PutItem` permissions for the Threats table. It has no network access permissions.
- **Response Lambda Role:** This role requires elevated privileges, including `dynamodb:GetItem` and `dynamodb:Scan`. Crucially, it requires `ec2:AuthorizeSecurityGroupIngress` and `ec2:CreateNetworkAclEntry` to modify network components. These permissions are scoped to specific Resource Tags to prevent accidental modification of critical infrastructure not related to the project.

VI. Conclusion

This project demonstrates a robust, scalable framework for automated threat detection and remediation on AWS. By coupling log analysis (ALB and S3) with event-driven compute (Lambda) and network enforcement points (Security Groups and NACLs), the system significantly reduces the window of opportunity for attackers. The move from manual log review to instantaneous, code-driven remediation represents a critical maturation in cloud security posture. Future work will focus on integrating AWS WAF for real-time Layer 7 filtering and implementing machine learning models within the detection pipeline to identify zero-day anomalies that standard signature matching may miss.

References

[1] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. A. Latif, F. Al-Turjman, and L. Mostarda, "Cyber security threats detection in internet of things using deep learning approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019.

[2] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602-622, Firstquarter 2016.

[3] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer security incident handling guide," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 800-61 Rev. 2, 2012.