SAPI-G
Secure API Gateway with AI Threat Detection

Secure Software Development
CY-321

Project Team:

Umar Tariq (2022604)
M Zeeshan (2022644)
Ayela Israr (2022130)
Ahmad Amjad (2022063)

1 Introduction

With the increasing reliance on APIs for modern applications, ensuring API security has become a critical challenge. Cyber threats such as DDoS attacks, SQL injection, and unauthorized access pose significant risks to data integrity and system reliability. This project aims to develop a Secure API Gateway with AI-driven Threat Detection, offering real-time monitoring, authentication, and protection against malicious activities.

2 Problem Statement

Existing API security solutions primarily rely on static rules and predefined filters, which are ineffective against adaptive cyber threats. Hackers exploit API vulnerabilities, leading to data breaches, unauthorized access, and service disruptions. Traditional security mechanisms lack intelligent threat detection capabilities to proactively identify and mitigate evolving attack patterns.

3 Objectives

The primary goals of this project include:

- Developing a secure API gateway with authentication and traffic filtering.
- Implementing rate-limiting, IP blacklisting, and SQL injection prevention.
- Utilizing AI/ML models to detect and mitigate security threats.
- Providing a real-time monitoring dashboard for security administrators.
- Ensuring secure authentication via JWT & OAuth 2.0.

4 Methodology

4.1 API Gateway Development (Node.js)

- Implement traffic filtering, authentication, and request validation.
- Enforce rate-limiting, logging, and access control policies.

4.2 AI-Based Threat Detection (Python, Machine Learning)

- Analyze API traffic using anomaly detection algorithms (Isolation Forest, Logistic Regression).
- Identify unusual activity patterns and block malicious requests dynamically.

4.3 Admin Dashboard (React.js)

- Provide a real-time security analytics dashboard for API traffic monitoring.
- Enable manual intervention (blacklisting/unblacklisting IPs).

4.4 Database (MongoDB/PostgreSQL)

- Store API logs, detected threats, and blacklisted IPs.
- Enable historical data analysis for security improvement.

5 Expected Outcomes

Upon completion, the project will deliver:

- A fully functional Secure API Gateway.
- AI-driven threat detection for proactive security.
- A real-time monitoring dashboard for security administrators.
- Improved API security via authentication, encryption, and logging.

6 Timeline & Milestones

Phase	Task	Duration
Phase 1	API Gateway Development (Node.js)	1 Week
Phase 2	Implement Security Features (Rate-Limiting, IP Blacklisting)	1 Week
Phase 3	Develop AI Threat Detection (Python)	2 Weeks
Phase 4	Create Admin Dashboard (React.js)	1 Week
Testing	Load Testing & Security Audits	1 Week
Deployment	Cloud Hosting & Finalization	1 Week

7 Resources Required

- **Software:** Node.js, Express.js, Python (Scikit-learn, Pandas), React.js, MongoDB/PostgreSQL.
- **Hardware:** Cloud-based server (AWS/GCP).
- **Development Tools:** VS Code, Postman, Docker, GitHub.

8 Conclusion

This project provides a proactive security solution by integrating AI-driven threat detection with traditional security mechanisms. The Secure API Gateway will offer real-time monitoring, intelligent attack prevention, and robust authentication, enhancing API security for modern web applications.