iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75022

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Secure and Vulnerable E-Commerce Platform: Evaluation of Web and OS Security through Real-Time Attack Simulation

Surendra S[1], Dr. R. Kanagavalli[2]
[1]*Student, Department of Cyber Security, The Oxford College of Engineering, Bengaluru*
[2]*Professor & HOD, Department of Information Science and Technology, The Oxford College of Engineering, Bengaluru*

Abstract: ***E-commerce has revolutionized the digital economy by enabling seamless global transactions; however, it remains a prime target for cyberattacks at both the web and operating system (OS) levels. This paper presents a novel dual-mode academic platform, "Secure and Vulnerable E-Commerce Platform: Evaluation of Web and OS Security through Real-Time Attack Simulation," designed for research, education, and practical cybersecurity training. The system operates in two configurations: Secure Mode, which enforces industry-grade defenses such as input validation, parameterized queries, CSRF protection, HTTPS enforcement, and OS hardening; and Vulnerable Mode, which intentionally exposes flaws like SQL Injection (SQLi), Cross-Site Scripting (XSS), and insecure configurations to simulate real-world attacks. A centralized, real-time logging dashboard continuously aggregates web and OS events, offering immediate visibility into intrusion attempts, anomalies, and defensive responses. By contrasting secure and vulnerable operations in a controlled environment, the platform bridges the gap between theoretical understanding and hands-on experience—enhancing learners' expertise in threat detection, mitigation, and resilient system design.***

Keywords: ***E-commerce Security, Cybersecurity Education, Dual-Mode Platform, Web Application Vulnerabilities, Operating System Hardening, Real-Time Logging, Attack Simulation, Secure Coding Practices.***

I. INTRODUCTION

The rapid evolution of e-commerce has transformed the global digital economy, enabling instantaneous, borderless trade and reshaping consumer behavior. As online platforms increasingly handle sensitive data such as personal information, financial transactions, and authentication credentials, their security posture has become a critical determinant of user trust and business continuity. However, this dependence on interconnected technologies has simultaneously made e-commerce systems a lucrative target for cyber adversaries. Modern cyber threats exploit weaknesses at multiple layers of these systems. At the web application layer, vulnerabilities such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) enable attackers to manipulate databases, hijack sessions, or compromise user data. At the operating system (OS) layer, issues like insecure configurations, unpatched services, and privilege escalation amplify the risk by granting attackers deep system-level control. Despite widespread awareness and the availability of frameworks such as the OWASP Top 10 and CIS Benchmarks, real-world systems continue to exhibit poor input validation, weak authentication, and inconsistent patch management. This paper introduces a dual-mode academic platform titled "Secure and Vulnerable E-Commerce Platform: Evaluation of Web and OS Security through Real-Time Attack Simulation." The proposed system uniquely integrates Secure Mode and Vulnerable Mode within a single environment, allowing users to alternate between hardened and deliberately exposed configurations. This dual approach serves both pedagogical and experimental objectives, enabling safe, hands-on exploration of attacks, defenses, and system behaviors under controlled conditions. A defining feature of the platform is its real-time logging and monitoring dashboard, which aggregates data from both the web and OS layers. By capturing events such as failed logins, suspicious queries, and privilege escalation attempts, it provides actionable insights for threat detection, analysis, and response. The platform thereby bridges the gap between theoretical cybersecurity concepts and practical implementation, empowering learners and researchers to understand the full attack–defense lifecycle. In essence, this research underscores the necessity of a holistic, layered security approach for e-commerce applications, blending defensive mechanisms, real-time visibility, and experiential learning to build resilient, trustworthy systems in an era of escalating cyber threats.

II. LITERATURE SURVEY

The reviewed literature consistently highlights that e-commerce systems are among the most targeted infrastructures due to the high value of the personal and financial data they process. Studies emphasize that despite advancements in security frameworks, web-layer vulnerabilities such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and weak authentication mechanisms remain persistent. These flaws primarily stem from insecure coding practices, inadequate input validation, and poor session management, which collectively enable attackers to exploit web interfaces and compromise sensitive data. Operating system (OS)-level weaknesses often magnify these risks. Misconfigured file permissions, outdated packages, and privilege escalation vulnerabilities allow attackers to gain full control of the hosting environment, bypassing even well-secured web layers. Joshi and Sinha (2022) highlighted that without system hardening and least-privilege enforcement, web-level defenses are insufficient to maintain overall platform integrity. Parallel research stresses the importance of real-time monitoring and visualization in strengthening security posture. Intrusion Detection and Prevention Systems (IDPS) such as Snort and Suricata, and Security Information and Event Management (SIEM) tools like Splunk, have demonstrated effectiveness in early anomaly detection and alerting. Gupta and Chatterjee (2023) proposed a real-time monitoring framework that aggregates suspicious queries and failed logins, underscoring that visibility is central to proactive defense. Visualization dashboards further enhance situational awareness by translating raw log data into actionable intelligence, enabling faster and more accurate responses to emerging threats. Additionally, researchers including Saltzer and Schroeder (1975) advocated for restricted log visibility following the *principle of least privilege*. Limiting access to sensitive logs to administrators not only safeguards privacy but also ensures compliance with standards such as GDPR, PCI DSS, and ISO/IEC 27001. This principle aligns with the educational orientation of the proposed platform, which grants learners access to anonymized events for analysis while reserving detailed insights for authorized instructors or administrators. From an academic perspective, initiatives like Fonseca and Vieira (2022) demonstrated the pedagogical value of dual-mode security environments, where learners interact with both vulnerable and secure versions of applications to understand how attacks occur and how they can be mitigated. However, these implementations are typically limited to web-level simulations, lacking integration with OS-level vulnerabilities or unified logging systems.

In summary, prior research reveals three critical gaps that this study aims to address: Lack of integrated web and OS-level defenses within educational or experimental environments. Insufficient real-time monitoring and visualization frameworks for simultaneous web and system-level analysis.

1. Absence of dual-mode configurations that facilitate side-by-side comparison of secure and vulnerable system behavior.

By bridging these gaps, the proposed Secure and Vulnerable E-Commerce Platform combines a defense-in-depth architecture with real-time logging and role-based visibility, providing a unified and practical framework for cybersecurity education, research, and experimentation**.**

III.OBJECTIVES

The objectives of this project are set to develop a comprehensive platform that demonstrates, analyzes, and enhances web and OS-level cybersecurity awareness through practical, real-time experimentation. The main goals include:

- To bridge theoretical learning and practical cybersecurity implementation: Develop a dual-mode platform that allows learners to experience both secure and vulnerable systems under real-time attack simulations.
- To create a dual-mode architecture: Design a system that operates in both Secure Mode (with safeguards such as CSRF protection, HTTPS enforcement, and OS hardening) and Vulnerable Mode (with intentional flaws like SQL Injection and XSS) for direct comparison.
- To enable real-time security monitoring: Implement a centralized logging dashboard that collects, analyzes, and visualizes system and web application events during simulated attacks.
- To support cybersecurity research and education: Provide a safe and interactive environment where students and researchers can perform penetration testing, analyze attack patterns, and learn defensive strategies.
- To promote secure coding and system-hardening practices: Emphasize the importance of secure design patterns, OS-level configuration management, and defensive programming in modern web applications.
- To ensure scalability and future adaptability: Build a modular platform that can later be integrated with AI-based detection models, advanced analytics dashboards, and cloud environments.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

A. Problem Statement

Despite the growing advancements in cybersecurity, e-commerce platforms remain highly vulnerable to cyberattacks that exploit both web and operating system layers. While many applications focus on secure implementations, few provide a controlled environment to study vulnerabilities and their real-world impacts.

Most existing educational tools—such as DVWA and OWASP WebGoat—demonstrate web-level attacks but lack:

- OS-level simulation and logging integration,
- Real-time defense visualization, and
- Comparative testing between secure and vulnerable configurations.

As a result, there is a gap in cybersecurity education and experimentation, where learners cannot safely execute and analyze both offensive and defensive operations within a unified system.

This project addresses that gap by designing a dual-mode e-commerce platform that intentionally alternates between secure and vulnerable states, allowing comprehensive, hands-on learning. The system integrates real-time monitoring, attack simulation, and OS-level event logging, giving users an in-depth understanding of how attacks occur and how defenses can be strengthened.

B. Scope

1) User Registration and Authentication: Implementing secure registration and login mechanisms for both users and administrators to ensure data privacy, controlled access, and role-based authentication.

2) Dual-Mode Operation: Providing two distinct system configurations Secure Mode and Vulnerable Mode to enable comparative analysis between protected and exposed environments for research and education.

3) Product and Transaction Management: Allowing administrators and users to perform essential e-commerce operations such as product listing, order placement, and transaction handling in both secure and vulnerable settings.

4) Attack Simulation and Analysis: Facilitating safe execution of common web attacks such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) in Vulnerable Mode, while observing real-time detection and prevention mechanisms in Secure Mode.

5) Real-Time Logging and Monitoring: Aggregating application and OS-level logs into a centralized dashboard that visualizes system events, intrusion attempts, and security alerts in real time.

6) Administrator Dashboard: Enabling administrators to monitor attacks, manage system configurations, and review performance reports of Secure and Vulnerable operations through an intuitive interface.

7) Security Awareness and Education: Providing an interactive environment for students and researchers to understand attack–defense mechanisms, develop secure coding practices, and conduct cybersecurity experiments safely.

8) Performance Evaluation: Measuring and analyzing system response, attack detection accuracy, and security effectiveness under different configurations to assess overall robustness and learning outcomes.

IV. METHODOLOGY

The proposed system follows an experimental and comparative research methodology, designed to evaluate how secure and vulnerable configurations of an e-commerce platform behave under real-world cyberattack conditions. The methodology emphasizes hands-on experimentation, controlled vulnerability exposure, and systematic analysis to measure the effectiveness of security mechanisms across both the web and operating system (OS) layers.

A. Research Design and Phases

The methodology is structured into six progressive phases: Requirement Analysis: Functional and security requirements were derived from established frameworks such as the OWASP Top 10, CIS Benchmarks, and compliance standards like GDPR and PCI DSS.

1) System Design: The platform was architected to operate in two distinct configurations — Secure Mode and Vulnerable Mode — both offering identical e-commerce functionalities (registration, product management, checkout, and payments) to ensure fair comparison.

2) Implementation: The system was developed using Django (Python), with SQLite/MySQL for database management and HTML, CSS, and JavaScript for the frontend. In Secure Mode, features such as input validation, parameterized queries, CSRF tokens, and HTTPS were implemented. In Vulnerable Mode, these safeguards were deliberately disabled to replicate real-world security flaws.

3) Attack Simulation: A controlled test environment was created using virtual machines (VMs) and Docker containers. Common attacks including SQL Injection (SQLi), Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Insecure Direct Object References (IDOR), and Privilege Escalation were executed in Vulnerable Mode to observe system behavior.

4) Real-Time Logging and Monitoring: A centralized logging framework was implemented to capture both web and OS-level events such as failed logins, suspicious queries, and unauthorized file access. Logs were processed in real-time and visualized through a dashboard using Python's logging library, offering immediate alerts and color-coded event severity indicators.

5) Evaluation and Comparative Analysis: Metrics such as attack success rate, detection accuracy, response time, and system stability were analyzed to compare Secure and Vulnerable modes. Results validated that the Secure Mode effectively blocked or mitigated all attack attempts that succeeded in the Vulnerable configuration.

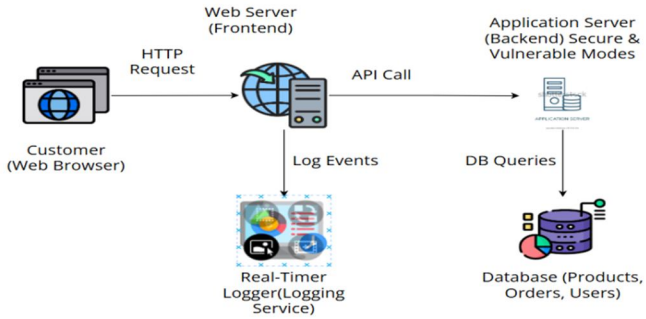


Fig. 1: System Architecture.

B. Dual-Mode Architecture

The core innovation of this project is its dual-mode architecture, Dual-mode enables seamless switching between Secure and Vulnerable configurations:

1) Secure Mode: Implements best practices like HTTPS enforcement, CSRF protection, ORM-based query execution, strong session management, and OS-level hardening.

2) Vulnerable Mode: Intentionally disables these defenses, using unsafe coding techniques, weak permissions, and unpatched services to simulate real-world attack surfaces.

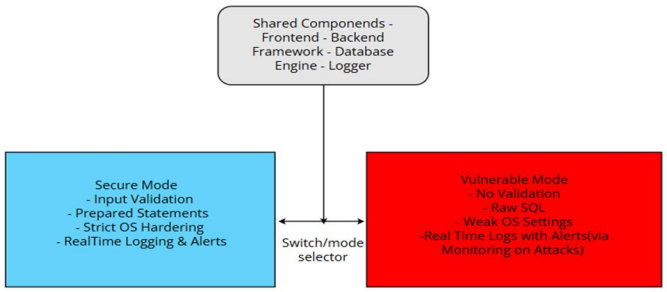


Fig. 2: Workflow of Dual-Mode Testing.

This side-by-side architecture allows learners and researchers to directly observe how coding practices and configurations affect system exposure, detection, and resilience.

C. Real-Time Logging Framework

The integrated logging mechanism functions as the central nervous system of the platform's defense model. It operates through three layers:

1) Event Capture Layer – Hooks embedded within the web and OS processes capture key actions such as login attempts, file operations, and system commands.

2) Log Management Module – Aggregates and stores event data with structured metadata (timestamp, user ID, IP address, event type, and severity).

3) Alerting and Visualization Layer – Displays events on a real-time dashboard with color-coded alerts (Red = Critical, Yellow = Suspicious, Green = Normal).

D. Evaluation Strategy

The experimental analysis used a comparative framework between Secure and Vulnerable modes to measure security effectiveness and system resilience.

1) Attack Success Rate: Percentage of attacks that successfully executed in each mode.
2) Detection Accuracy: Percentage of correctly logged and identified attack attempts.
3) System Performance: Average response time and resource utilization under load.
4) Educational Effectiveness: Measured by clarity of visualization and comprehension of security principles during demonstrations.

Results showed that Secure Mode consistently prevented all major attacks, while maintaining low performance overhead and high user interactivity, validating the design's robustness and educational value.

V. IMPLEMENTATION AND RESULTS

A. Implementation Overview

The proposed Secure and Vulnerable E-Commerce Platform was implemented using a modular, layered architecture designed to simulate both web and OS-level security scenarios in real time. The system was developed using the Django framework (Python) for backend logic and HTML, CSS, and JavaScript for the frontend, with SQLite for lightweight testing and MySQL/PostgreSQL for multi-user environments.

The platform supports two configurations:

Secure Mode — Implements industry-standard security mechanisms including input validation, parameterized queries, CSRF tokens, secure session management, role-based access control (RBAC), and OS-level hardening.

Vulnerable Mode — Intentionally disables these defenses to demonstrate security flaws such as SQL Injection (SQLi), Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Insecure Direct Object References (IDOR), and Privilege Escalation.

This dual-mode design enables controlled experimentation in a sandboxed environment, allowing learners to safely observe the differences between secure and insecure implementations.

B. Key Implementation Features

The table highlights the platform's flexibility to toggle between both states for direct comparison and study.

TABLE I

Feature	Secure Mode Implementation	Vulnerable Mode Implementation
SQL Injection (SQLi)	Parameterized ORM queries prevent injection	Raw concatenated SQL queries allow injection
Cross-Site Scripting (XSS)	Output encoding, CSP headers	No sanitization, scripts executed
CSRF Protection	CSRF tokens validated for each request	CSRF disabled on key routes
Authentication & Sessions	Strong password hashing, secure cookies	Weak passwords, predictable session IDs
File Upload Handling	Type validation and sandbox storage	Arbitrary file upload allowed
OS-Level Security	Hardened permissions, process monitoring	Misconfigured permissions, open services
Logging Framework	Real-time alerts, color-coded severity	Logs recorded passively, limited alerts

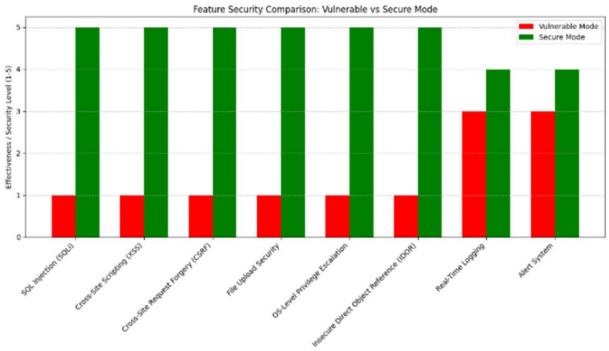


Fig 3: Comparison of Security Features in Secure and Vulnerable

C. Experimental Setup

Experiments were conducted in both local and cloud-based environments to validate performance, resilience, and educational effectiveness.

1) Local Environment: Ubuntu 22.04 LTS with Python 3.10, Django 4.x, SQLite Database, and local logging.
2) Cloud/Network Testing: Deployed via Ngrok and Docker for remote testing and isolated simulation.
3) Tools Used: Postman for API testing, browser developer tools for payload monitoring, and Linux audit logs for OS-level tracking.

The environment enabled safe execution of real-world attack simulations, ensuring that vulnerabilities were contained and did not impact external systems. This experiment confirm that Secure Mode successfully mitigated all attacks that were successful in Vulnerable Mode, validating the system's intended behavior.

TABLE II III

Attack Type	**Vulnerable Mode**	**Secure Mode**	**Observation**
SQL Injection	Attack succeeded; unauthorized data retrieved	Blocked by ORM and validation	Demonstrated prevention via parameterized queries
Cross-Site Scripting (XSS)	Script executed in browser	Rendered as plain text	Secure template rendering blocked malicious payloads
CSRF Attack	Request processed without validation	Request rejected with invalid token	CSRF tokens effectively enforced
Privilege Escalation	Simulated root access granted	Access denied, logged with alert	File permission hardening prevented escalation
File Upload Attack	Executable file uploaded and accessed	Invalid file type blocked	Secure mode ensured content filtering and sandboxing

D. Real-Time Logging and Visualization Results

The integrated logging and monitoring system proved essential for both security analysis and education.

- Each event was captured with structured metadata including timestamp, user ID, IP address, and severity.
- Real-time dashboards displayed color-coded alerts (Red for critical attacks, Yellow for suspicious activity, Green for normal user actions).
- Logs from both web and OS layers were correlated to provide end-to-end visibility.

In Secure Mode, the system detected and blocked attacks while immediately alerting the administrator. In Vulnerable Mode, attacks were logged post-execution, allowing learners to analyze exploit patterns and payloads.

E. Performance Evaluation

Performance testing showed minimal system overhead despite continuous monitoring.

- Average response delay: < 300 ms (Secure Mode)
- CPU utilization: < 20% under concurrent user load
- Logging latency: Negligible (< 0.5s for event capture and display)

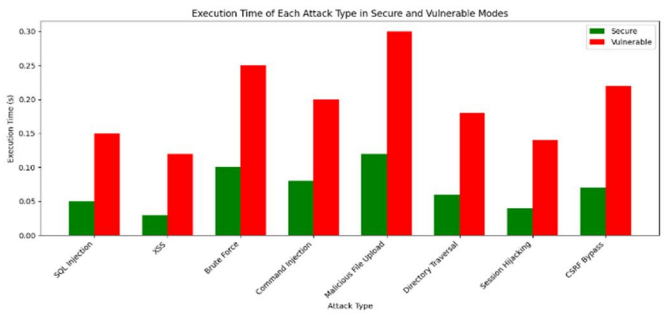


Fig. 4: Execution Time of Each Attack Type in Secure and Vulnerable Mode

The results demonstrate that enhanced security and real-time logging can be achieved without compromising performance or usability.

F. Educational and Practical Impact

Beyond security validation, the platform's design achieved its educational objective — enabling students and researchers to directly visualize the consequences of insecure coding and the benefits of best practices. The side-by-side comparative framework fosters deeper understanding of cyberattack mechanisms, defensive design, and system behaviour under stress, bridging the gap between academic learning and professional cybersecurity practice.

Summary of Results

The experimental results conclusively show that:

- Vulnerable Mode allows controlled exploitation for learning purposes.
- Secure Mode effectively mitigates all major web and OS-level threats.
- Real-time logging provides actionable visibility for both defensive and forensic analysis.
- The system operates efficiently even under attack simulations, validating its robustness, scalability, and educational value.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

The project "Secure and Vulnerable E-Commerce Platform: Evaluation of Web and OS Security through Real-Time Attack Simulation" successfully demonstrates a dual-mode educational system that connects cybersecurity theory with practical implementation. It integrates both Secure Mode and Vulnerable Mode within a single platform, allowing users to observe and compare how different security configurations behave under identical conditions. The Secure Mode applies strong protection techniques such as input validation, parameterized queries, CSRF protection, HTTPS enforcement, and OS-level hardening, while the Vulnerable Mode intentionally exposes flaws like SQL Injection and Cross-Site Scripting to simulate real attacks in a controlled environment.

The system's real-time logging and monitoring dashboard provides instant visibility of attacks and system responses, helping learners understand the impact of security controls. Experimental results showed that Secure Mode successfully blocked all attacks that compromised the Vulnerable Mode, proving the system's effectiveness. Overall, this project offers an innovative and scalable platform that enhances practical cybersecurity learning, supports research in web and OS-level defense, and promotes secure software development practices for the modern digital landscape.

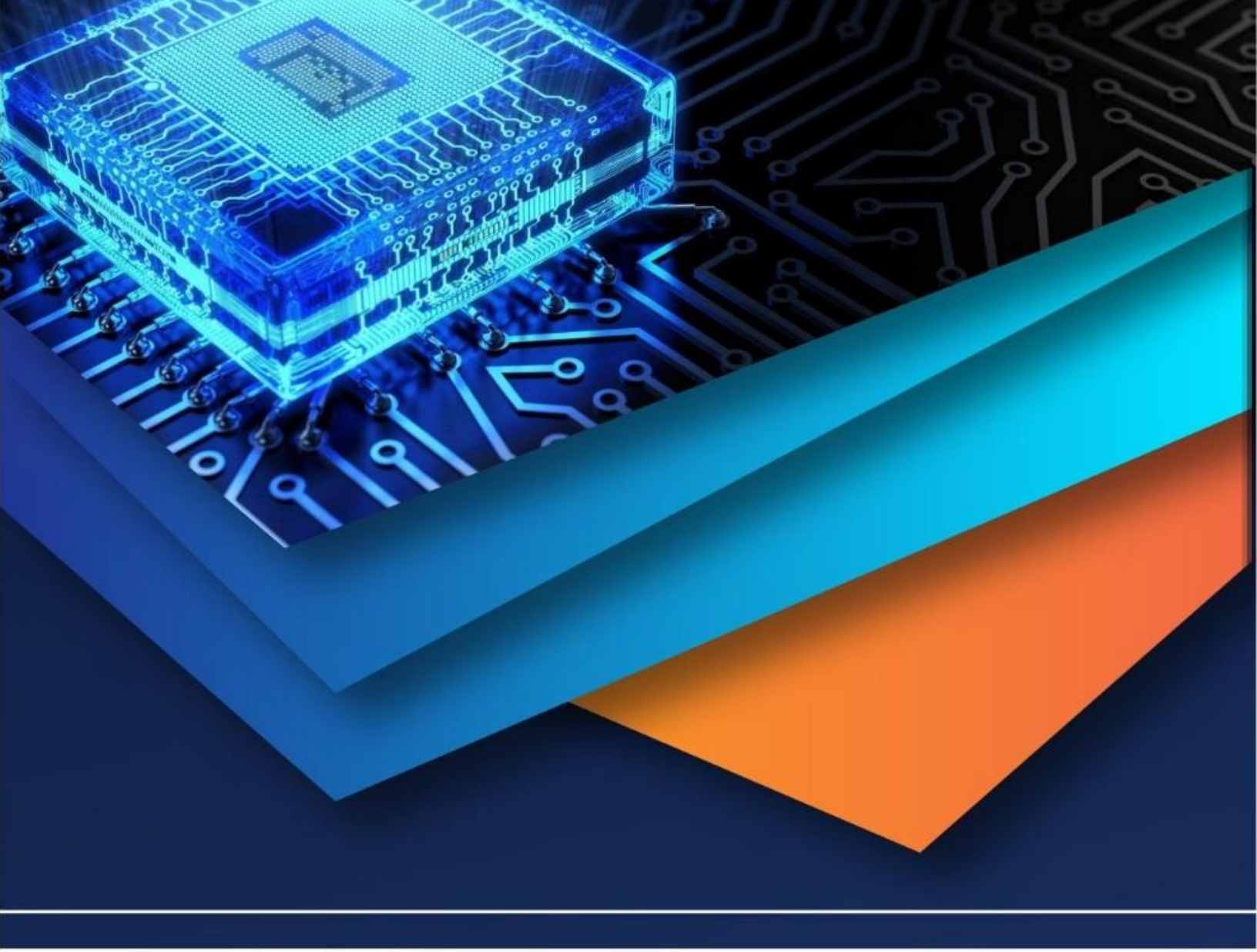
B. Future Scope

The current system lays a strong foundation for cybersecurity education and experimental research, but there is significant potential for future enhancement. Advanced vulnerabilities such as Remote Code Execution (RCE), Server-Side Request Forgery (SSRF), and Cross-Site WebSocket Hijacking (CSWSH) can be integrated to broaden attack simulation capabilities. Incorporating AI and machine learning-based anomaly detection can enable real-time classification and predictive defense against evolving threats. Expanding support to cloud platforms like AWS, Azure, and GCP, along with multi-OS compatibility (Windows, Linux, macOS), will allow simulations across diverse and distributed environments. The system can also be enhanced through integration with professional tools such as Burp Suite, OWASP ZAP, Wireshark, and Metasploit for deeper penetration testing and analysis. Additionally, introducing gamified learning modules with challenges and guided labs can make cybersecurity training more engaging and practical for students. Finally, improving the analytics dashboard using tools like ELK Stack (Elasticsearch, Logstash, Kibana) or Grafana will provide advanced visualization and correlation of attack data. With these improvements, the platform can evolve into a comprehensive cybersecurity training suite that combines simulation, detection, analytics, and automation—serving as a benchmark framework that bridges secure development, research innovation, and practical defense strategies.

REFERENCES

[1] A. D. Keromytis, "Web Application Security: Threats and Vulnerabilities," IEEE Security & Privacy, vol. 21, no. 2, pp. 58–66, 2023.
[2] O. Alhazmi, A. Malaiya, and Y. K. Malaiya, "Vulnerability Assessment of Web Applications," IEEE Transactions on Software Engineering, vol. 48, no. 5, pp. 1421–1434, 2022.
[3] J. Fonseca and M. Vieira, "An Educational Web Security Lab Using Vulnerable and Secure Modes," Journal of Information Security and Applications, vol. 68, pp. 103–118, 2022.

[4] Y. Chen, D. Zhao, and H. Xue, "Comparative Analysis of Secure Coding Practices in E-Commerce Applications," ACM Computing Surveys, vol. 55, no. 3, pp. 1–19, 2023.
[5] U. Kishnani and S. Das, "Dual-Technique Privacy & Security Analysis for E-Commerce Websites Through Automated and Manual Implementation," International Journal of Cybersecurity Research, vol. 12, no. 4, pp. 211–226, 2024.
[6] A. Houcheimi, R. A. Kabbara, and M. Farhat, "The Role of Secure Online Payments in Enabling E-Tailing in Lebanon," International Journal of Information Systems and E-Business Management, vol. 17, no. 2, pp. 65–82, 2024.
[7] K. R. Joshi and P. R. Sinha, "OS-Level Security Hardening for Web-Hosted Applications," Journal of Network and Computer Security, vol. 29, no. 1, pp. 57–69, 2022.
[8] Z. Morić, T. Pavić, and I. Krnić, "Protection of Personal Data in the Context of E-Commerce: A Case and Regulatory Framework," Croatian Journal of Information Security and Law, vol. 15, no. 1, pp. 33–46, 2024.
[9] R. Gupta and S. Chatterjee, "Real-Time Logging and Monitoring in Secure Web Applications," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 3, pp. 112–123, 2023.
[10] X. Li, Y. Peng, X. Sun, Y. Duan, Z. Fang, and T. Tang, "Unsupervised Detection of Fraudulent Transactions in E-Commerce Using Contrastive Learning," IEEE Access, vol. 13, pp. 119341–119356, 2025.
[11] S. Johnson, "Enhancing Security Automation in E-Commerce Platforms Using Machine Learning and Artificial Intelligence," Journal of Intelligent Systems Security, vol. 10, no. 4, pp. 85–99, 2023.
[12] Saltzer, J. H., and Schroeder, M. D., "The Protection of Information in Computer Systems," Proceedings of the IEEE, vol. 63, no. 9, pp. 1278–1308, 1975.
[13] OWASP Foundation, OWASP Top Ten Web Application Security Risks – 2021 Report, [Online]. Available: https://owasp.org/www-project-top-ten/
[14] PCI Security Standards Council, Payment Card Industry Data Security Standard (PCI DSS) Requirements and Security Assessment Procedures, Version 4.0, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY