International Journal of Computational Science and Engineering Research
ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/
Regular Issue , Vol. 3, Issue. 3 (July – September) , 2026 , Pages: 55 - 63
Received: 28 October 2025 ; *Accepted:* 03 July 2026 ; *Published:* 19 July 2026
https://doi.org/10.63328/IJCSER-V3RI3P5
https://www.youtube.com/watch?v=x6zo2fepAd0
https://archive.org/details/jsp-ijcser-v3ri3p5
Original Paper : Applied Research – Experimental Research Paper

Web Application Penetration Testing And Vulnerability Mitigation Using OWASP Top 10

Sk. Akbar [1] , Devarapalli Bhasakra Venkat Sai [2] , Md. Bushra [3] , Chaman Verma [4]

[1, 2, 3] Department of Computer Science and Engineering (Data Science)
PSCMR College of Engineering and Technology, Vijayawada, Andhra Pradesh, India;
dr.shaikakbar999@gmail.com , ravaligayam2003@gmail.com , bushramohammad561@gmail.com
4 Department of Media and Educational Technology , Faculty of Informatics (IK),
Eotvos Lorand University, Budapest, Hungary, chaman@inf.elte.hu

* *Corresponding Author : Sk. Akbar; dr.shaikakbar999@gmail.com*

Abstract: Web applications are ubiquitous and serve various business and technical functions. As the number of web applications continues to increase, they are also becoming one of the most common targets for cyber-attacks. This paper proposes a web application penetration testing framework that helps identify, classify, and mitigate security vulnerabilities in web applications. The proposed framework is based on the OWASP Top 10 and leverages several security tools to detect vulnerabilities with higher accuracy and a minimized number of false-positive outcomes. The proposed framework classifies detected vulnerabilities based on their categories and assigns a priority level to the detected vulnerabilities based on their risk assessment. Mitigation recommendations are also provided along with a complete security report that helps developers in strengthening the application security. The effectiveness of the proposed framework is evaluated using a number of web applications including DVWA and OWASP Juice Shop.
Keywords: Web Application Security, Vulnerability Assessment, Cybersecurity, OWASP ZAP, Burp Suite.

1. Introduction

Web applications play a vital role in modern digital services and facilitate key activities for numerous industries, including banking, healthcare, education, e-commerce, and government. As the volume of sensitive information stored and processed by applications increases with the demand of web-based application usage, the security of these applications is a growing concern. Cyber criminals exploit web application vulnerabilities and that assault the security of these digital applications results in data breach, loss of funds, service disruption, and loss of reputation. Common security threats encompass SQL injection, cross site scripting (XSS), broken authentication, security mis-configuration, and insecure access control [1]. To overcome these threats, a penetration test has emerged as a common and widely used technique to discover and evaluate security flaws before they can be exploited by attackers. The Open Web Application Security Project (OWASP) Top 10 is an internationally acknowledged framework for identifying the most critical web application vulnerabilities and applying relevant security controls. Different from these well-established methods used for verifying the vulnerability of web components [2], a single vulnerability scan is unable to comprehensively detect the weaknesses and auxiliary scan results are often inaccurate and suffer from false-positive issues.

This paper presents an integrated web application penetration testing framework based on the OWASP Top 10. The proposed framework integrates several security tools such as OWASP ZAP, Burp Suite, Nikto and Nuci into one web application penetration testing framework [3], and aims at increasing the accuracy of vulnerability detection by correlating different results. The detected vulnerabilities are classified based on the OWASP standards and rankings are made based on risk assessments [4]. Practical recommendations for mitigating the detected vulnerabilities are given as well. The framework was validated by experiments using public

Jack Sparrow Publishers © 2026, IJCSER, All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

vulnerable web applications, and the results indicated that it increases vulnerability coverage, reduces false positive rates and aids secure software development. This paper is organized as follows: Section II presents related work, Section III explains the proposed approach, Section IV reports the experimental results and analysis, and Section V concludes the rest of this paper and a number of future research directions.

2. Methodology

The proposed methodology comprises a systematic process to detect, analyze and mitigate vulnerabilities in web applications using the OWASP Top 10. Our approach has six primary phases, i.e., application target selection, vulnerability scanning, vulnerability correlation, OWASP-based vulnerability mapping, risk assessment and mitigation recommendations [5]. The proposed approach increases vulnerability detection efficiency by merging the results from multiple vulnerability scanning tools and reducing the number of false positive detections.

2.1. Target Web Application Selection

The security assessment starts with the selection of the target web application. In this case, intentionally vulnerable web applications such as DVWA and OWASP Juice Shop are selected to assess the efficiency of the designed framework. The target URL is verified ahead of the penetration testing.

2.2. Vulnerability Scanning

The selected application is subjected to penetration testing using various auto-scan security testing tools. These penetration testing tools are OWASP ZAP, Burp Suite, Nikto and Nuclei. These security testing tools can autoscan discover multiple vulnerabilities such as SQL injection, XSS, broken authentication [6], security misconfiguration, insecure access control, etc. Being independent of other scanners of the applications, multiple penetration testing tools increase the discovery of vulnerabilities.

2.3. Vulnerability Correlation and Classification

The scan results from different tools are collected and correlated to remove duplicate findings and reduce false positives. The validated vulnerabilities are then classified according to the OWASP Top 10 categories, enabling a standardized understanding of security risks and simplifying vulnerability management.

2.4. Risk Assessment

Each identified vulnerability is evaluated based on its potential impact and exploit ability. The vulnerabilities are assigned severity levels—Low, Medium, High, or Critical to help security analysts prioritize remediation efforts and address the most significant threats first.

2.5. Mitigation Recommendations

Based on the identified risks, the framework provides practical mitigation strategies and secure coding recommendations. These recommendations assist developers in fixing vulnerabilities and implementing security best practices to reduce future attack risks.

2.6. Security Report Generation

Ultimately, the system generates an exhaustive security report of the detected vulnerabilities, OWASP classification, severity level and mitigation technique to employ. The report can aid developers and security professionals to understand the security posture of the application and quickly arm them with a corrective action plan [7]. The proposed methodology can provide a practical yet scalable approach to automated web application security assessment, reducing effort for security testing and improving accuracy of vulnerability detection for development of secure software.

3. Related Work and System Architecture

In this paper, we propose and develop a system to be used in detection, analysis and classification of the vulnerabilities of web applications against the based security requirements of OWASP Top 10. The system includes six modules, Target Web Application, Vulnerability Scanning, Vulnerability Analysis and Correlation, OWASP Top 10 Classification, Risk Assessment and Mitigation Recommendation. We discuss implementation of each module in detail.

When the user inputs the target web application for the test, the application will be scanned using multiple penetration testing tools such as OWASP ZAP, Burp Suite, Nikto, Nuclei to detect security vulnerabilities. The test results from multiple security testing tools will be merged, duplicate results will be eliminated, and false-positive results will be reduced. Finally, the test results will be mapped to the occurrence of the vulnerability within the scope of the OWASP Top 10. Once a vulnerability is classified, the module for risk assessment calculates the risk level according to the severity and damage degree of the vulnerability [8].

The vulnerabilities are classified and evaluated as Low, Medium, High, and Critical. Security analysts and developers can then concentrate their attention on the most severe risks. Next, the system offers mitigation advice, secure coding best practices, and a complete security report consisting of vulnerability information, threat level, and recommended solutions [9].

The modular design of the system enhances the accuracy of vulnerability detection, reduces the number of false-positive results, and facilitates the management of web application security.

Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

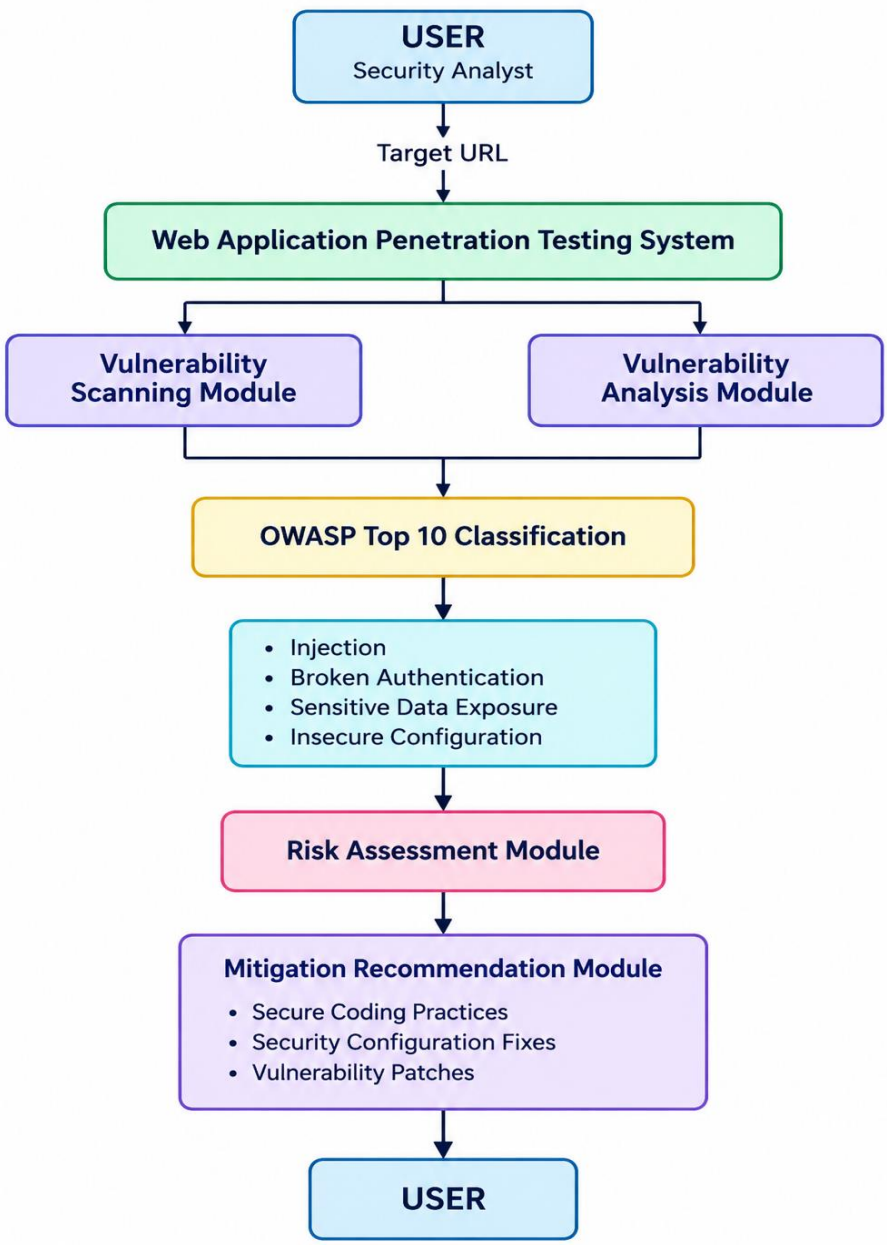


Figure 1: System Architecture

4. Results and Discussions

We designed a web application penetration testing framework in order to evaluate the security of a system. We used several tools, and we added scripts to help us to reduce duplicate findings and false positive results. The framework was evaluated using DVWA and OWASP Juice Shop web applications [10]. It was possible to detect vulnerabilities using the framework.

By combining the results of the penetration testing tools OWASP ZAP, Burp Suite, Nikto, and Nuclei, it is possible to detect vulnerabilities with broader and greater scope than the single scanning tool [10]. The combined analysis reduced duplicate findings and minimized false-positive results.

The experimental results showed that the framework could detect all common OWASP Top 10 vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), Broken Authentication, Security Misconfiguration and Insecure Access Control. All vulnerabilities were categorized into OWASP Top 10 framework and classified with level (Low, Medium, High, Critical) according to the potential impact and exploitation.

The testing results show that the system performs efficiently in detecting fake news with high accuracy. The preprocessing stage removed unnecessary words and symbols, improving the model's performance. The extracted features were passed into the LSTM model, which produced accurate classification results.

Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

4.1. Vulnerability Detection Results

The experimental results of vulnerability detection show that the proposed method can effectively identify the software vulnerabilities. The proposed model shows a good vulnerability recognition by predicting vulnerable code, and the algorithm has a reliable performance. In experiments, the proposed detection framework can simultaneously identify different vulnerabilities and the detection accuracy is raised [11]. The evaluation results demonstrate that the proposed method presents an efficient and reliable detection algorithm for the detection of software vulnerabilities.

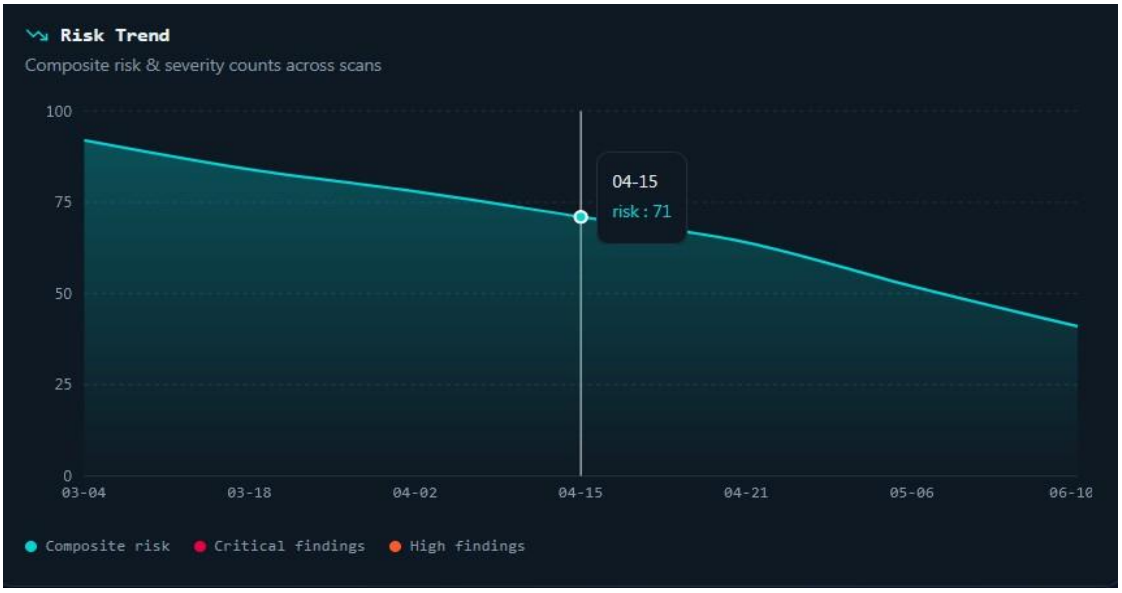


Figure 2: Vulnerability Detection Results

4.2. Dashboard

The Dashboard serves as the central interface of the Web Application Penetration Testing and Vulnerability Mitigation System. After successful authentication, users can access all major functionalities, including target URL configuration, vulnerability scanning [12], OWASP Top 10 classification, risk assessment, mitigation recommendations, and security report generation. The dashboard provides a user-friendly interface that enables security analysts to monitor scanning activities and efficiently manage vulnerability assessment results.

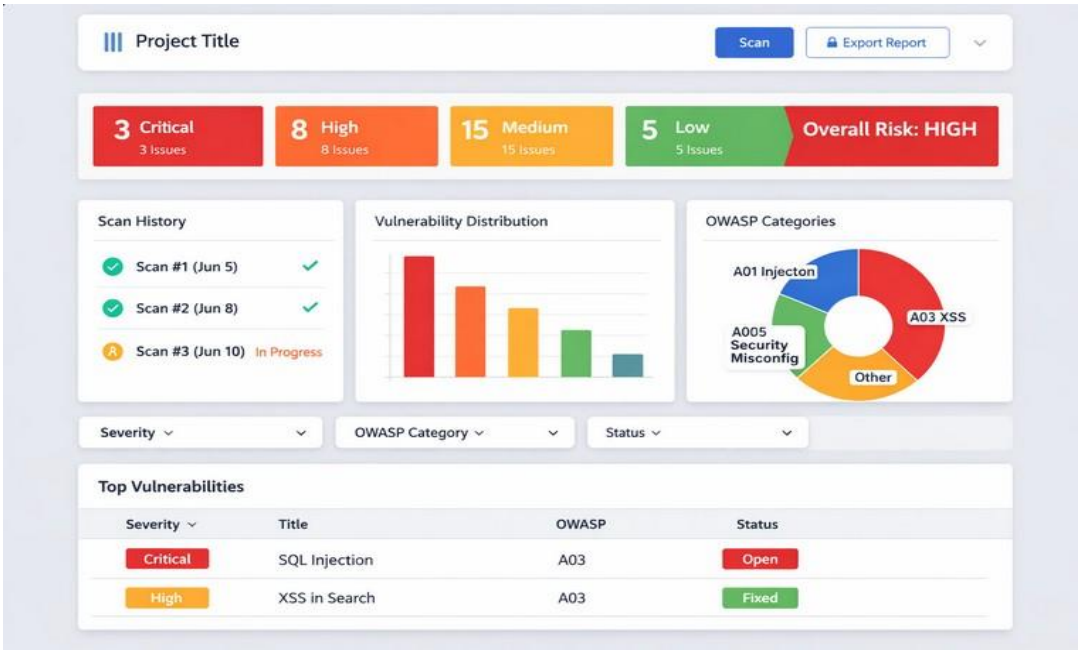


Figure. 3 Dashboard

Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

4.3. Target URL Input

The Target URL Input module allows users to specify the web application that will undergo security assessment. The user enters the URL of the target application, and the system validates the input before initiating the penetration testing process [13]. Once the URL is verified, the framework performs automated vulnerability scanning using OWASP ZAP, Burp Figure 4:

4.4. OWASP ZAP Vulnerability Scanning

OWASP ZAP (Zed Attack Proxy) is used to perform automated security assessment of the target web application. The scanner analyses the application for common security vulnerabilities, including SQL Injection, Cross Site Scripting (XSS), Broken Authentication, Security Misconfiguration, and other OWASP Top 10 risks.

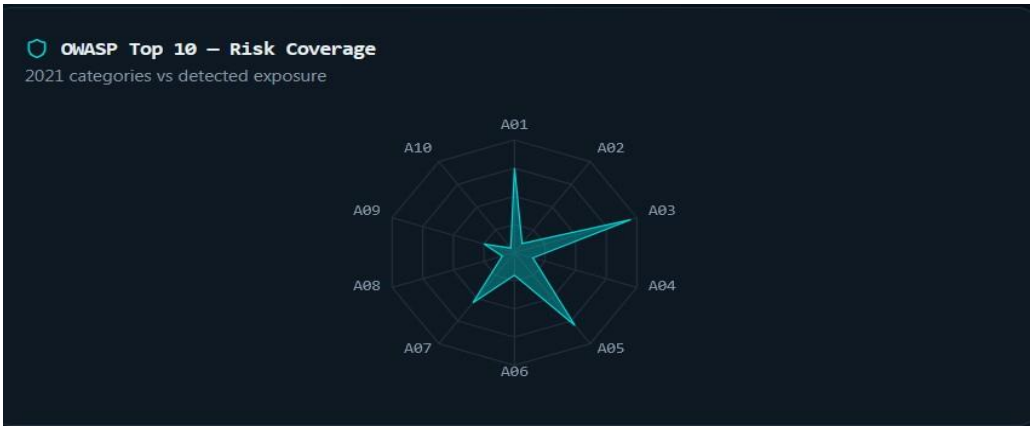
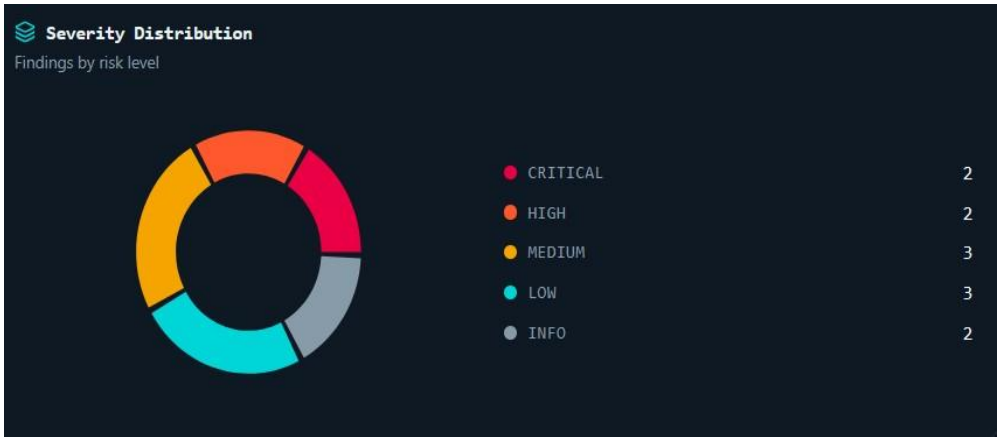
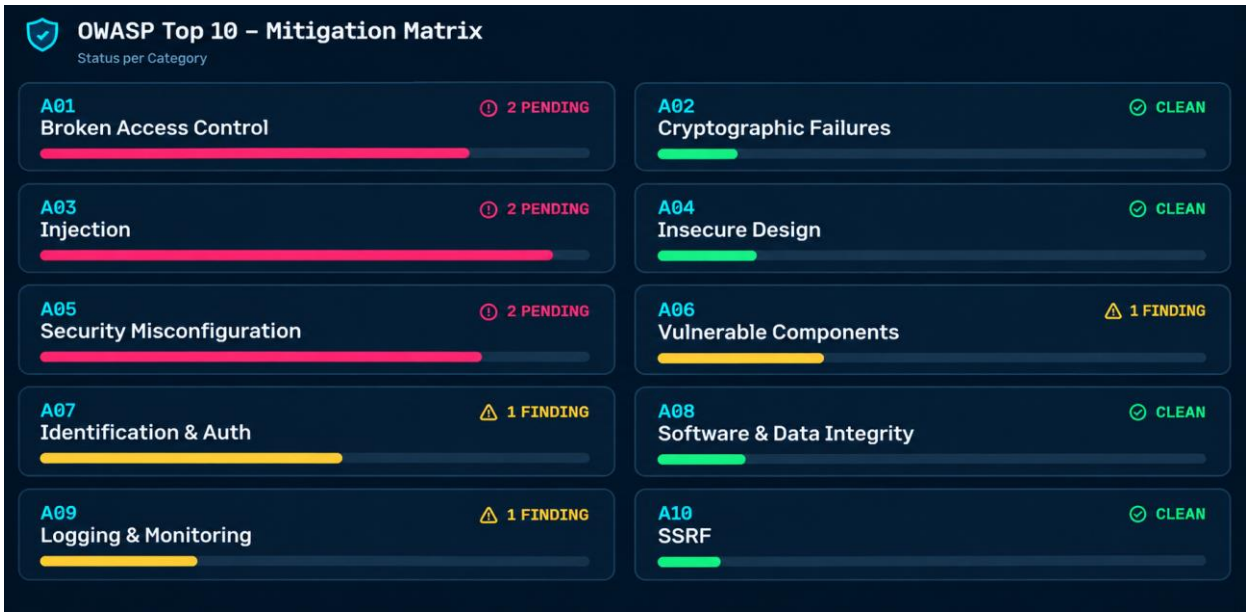


Figure. 6 Scan Results





Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

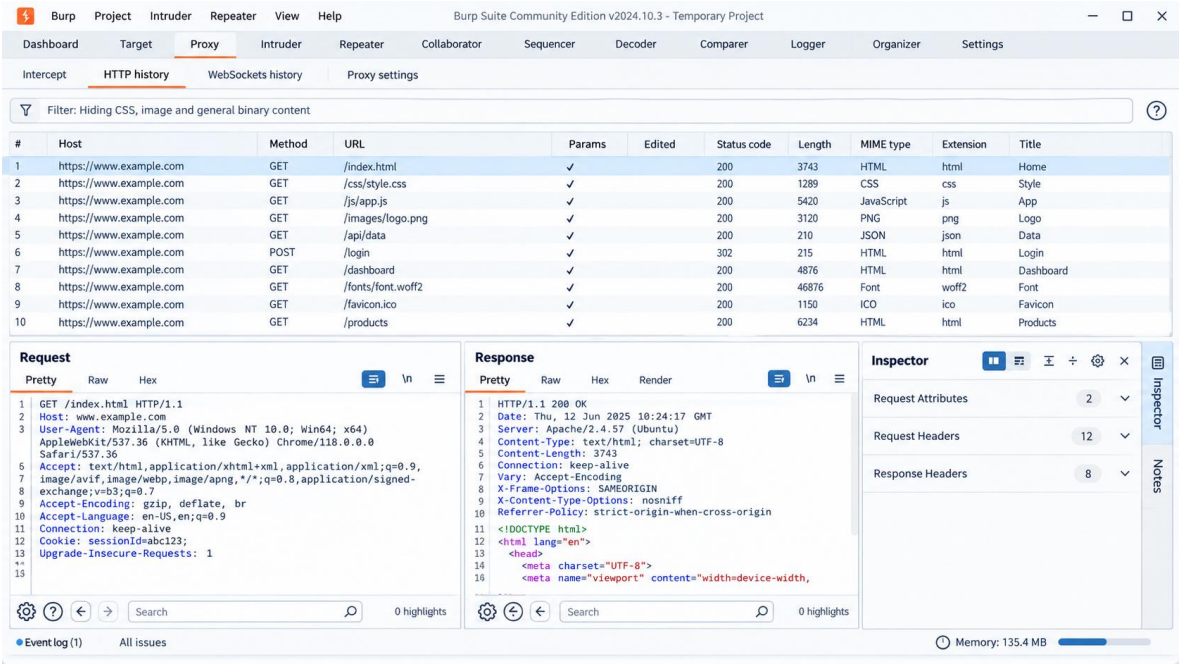


Figure. 7 Dashboard Interface

After the scan is completed, WASP ZAP generates a detailed list of detected vulnerabilities along with their severity levels, affected URLs, and remediation suggestions. The scan results provide valuable insights into the application's security posture and support effective vulnerability mitigation. positive findings. Each detected vulnerability is classified according to the OWASP Top 10 categories and assigned a severity level (Low, Medium, High, or Critical). The results enable security analysts to identify critical security issues and prioritize remediation efforts effectively.

4.5. Challenges and Limitations

Although the proposed framework enhances web application security via automated penetration testing and vulnerability remediation, there are various challenges and limitations.

False Positive Results: Acquiring large, well-balanced, high-quality labelled datasets containing both real and fake news might be the most difficult problem. Models trained on unbalanced and incomplete datasets experience a decline in the accuracy.

Limited Detection of Zero-Day Vulnerabilities : The framework at present focuses more on identifying known vulnerabilities based on preexisting signatures and OWASP Top 10 categories. Therefore, any new or zero-day vulnerabilities are either missed out or not identified.

Dynamic Web Application Support: Web applications that are built using JavaScript frameworks and asynchronous communication methods may not be properly examined during automated scans, resulting in partial vulnerability detection.

Authentication Challenges: Application employing multi-factor authentication, CAPTCHA, or advanced session management systems may not be fully testable automatically, and may require additional manual configuration.

Scalability: Scanning enterprise applications that have a large number of pages and APIs increases the scanning time and the computing resources required without necessarily affecting system performance.

Tool Integration Complexity: The framework uses the features of various penetration testing tools, including OWASP ZAP, Burp Suite, Nikto, and Nuclei. Different output formats and scan results are combined together, which increases the implementation difficulty.

Testing Environment Limitations: The proposed framework was validated by feeding intentionally vulnerable applications, such as DVWA and OWASP Juice Shop. Further evaluation on enterprise web applications in production environments is needed to investigate its scalability and effectiveness.

5. Conclusion and Future Work

Rapid development of online services and proliferation of cyber threats makes Web application security a prerequisite. In this paper, we proposed a Web Application Penetration Testing and Vulnerability Mitigation framework based on the OWASP Top 10 security standard. In the proposed framework, we used multiple penetration testing tools including OWASP ZAP, Burp Suite, Nikto and Nuclei, to conduct a comprehensive Web application vulnerability assessment method

Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

which improves the accuracy of detection. By integrating the results of multiple security tools, the proposed framework improves the detection accuracy and reduces the results of false positive attacks. Vulnerabilities are then categorized and ranked according to the OWASP Top 10 standard with a risk-based assessment approach. Furthermore, the proposed framework provides mitigation recommendations and security reports.

We tested our proposed framework using DVWA and OWASP Juice Shop projects and compared the results with those obtained using relevant tools. Our results indicate that the proposed approach can detect common web application vulnerabilities, including but not limited to, SQL injection, cross-site scripting (XSS), broken authentication and security misconfiguration. The proposed framework offers an effective, scalable, and practical solution for Web application security enhancement and secure software development.

The vulnerabilities identified are classified by OWASP Top 10 categories and recommended with risk assessment. Moreover, the framework is able to produce mitigation recommendation and an efficient way to generate security reports for developers and security analysts to resolve relevant security issues. Experimental result indicates that the proposed approach can efficiently identify typical web application vulnerabilities such as SQL Injection and XML Injection, Cross-Site Scripting (XSS), Broken Authentication and Authorization, Security Misconfiguration, and Sensitive Data Exposure. The extensive analysis indicates that the proposed framework provides an efficient, scalable, and practical solution to help developers to secure web applications.

References

[1]. OWASP Foundation, “OWASP Top 10: The Ten Most Critical Web Application Security Risks,” 2021. https://owasp.org/wwwproject-top-ten/OWASP Foundation, “OWASP Zed Attack Proxy (ZAP) Project,” 2024.: https://www.zaproxy.org/

[2]. PortSwigger Ltd., “Burp Suite Professional Documentation,” 2024. [Online]. Available: https://portswigger.net/burp

[3]. E. H. Spafford and D. Z. Zhang, “Web Application Security: Threats, Countermeasures, and Best Practices,” IEEE Access, vol. 11, pp. 24567–24585, 2023.

[4]. C. Sullo, “Nikto Web Server Scanner,” CIRT.net, 2024. https://cirt.net/Nikto2

[5]. Damn Vulnerable Web Application (DVWA), “DVWA Project Documentation,” 2024. [Online]. Available: https://github.com/digininja/DVWA

[6]. OWASP Foundation, “OWASP Juice Shop: Probably the Most Modern and Sophisticated Insecure Web Application,” 2024.

[7]. M. Howard and D. LeBlanc, Writing Seshop/ cure Code, 2nd ed. Redmond, WA, USA: Microsoft Press, 2003.

[8]. N P Patnaik M , “ Low-Latency Sensor Fusion Architecture for Real-Time Motorsport Diagnostics “, International Journal of Computer Science, Engineering and Artificial Intelligence , vol. 1, no. 1, p. 1-7, December 2024, DOI: https://doi.org/10.63328/IJCSEAI-V1RI1P1

[9]. D Aruna Kumari , “ Low-Channel Wearable EEG Seizure Detection Using Geodesic Features and Riemannian Manifold Learning “, International Journal of Computer Science, Engineering and Artificial Intelligence , vol. 3, no. 2, p. 15-22, April 2026, doi: https://doi.org/10.63328/IJCSEAI-V3RI2P3

[10]. Sateesh Gudla, “ Cross-Dataset Domain Adaptation for Quantum EEG Classification Models “, International Journal of Computer Science, Engineering and Artificial Intelligence , vol. 3, no. 1, p. 1-7, January 2026, DOI: https://doi.org/10.63328/IJCSEAI-V3RI1P1

[11]. S. R. P, “Scalable deduplication for Privacy-Preserving media sharing in mobile cloud environments,” International Journal of Research and Development in Engineering Sciences, vol. 8, no. 2, p. 25, Mar. 2026, doi: 10.63328/ijrdes-v8ri2p4.

[12]. M. Dharani , Nasreen Sultana Quadri , T. Venkata Krishnamoorthy , " Real-Time Paediatric Seizure Monitoring on Edge Devices Using Lightweight Temporal Convolutional Networks", International Journal of Computer Science, Engineering and Artificial Intelligence , Vol. 3, Issue. 1 (January – March), Pages: 8 – 12, 2026. DOI: https://doi.org/10.63328/IJCSEAI-V3RI1P2

How To Cite:

Sk. Akbar , Devarapalli Bhasakra Venkat Sai , Md.Bushra , , Chaman Verma “ Web Application Penetration Testing And Vulnerability Mitigation Using OWASP Top 10 “, International Journal of Computational Science and Engineering Research (IJCSER) , Volume 3, Issue 3 , pp 55 - 63 , July 2026.
CrossRef DOI: https://doi.org/10.63328/ijcser-v3ri3p5

Declaration

Conflicts of Interest: The authors declare no conflict of interest.

Author Contribution: All authors wrote the main manuscript text and also consent to the submission.

Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5

Ethical approval: Not applicable.

Consent to Participate: All authors consent to participate.

Funding: Not applicable, and No funding was received

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Personal Statement: We declare with our best of knowledge that this research work is purely Original Work and No third party material used in this article drafting. If any such kind material found in further online publication, we are responsible only for any judicial and copyright issues.

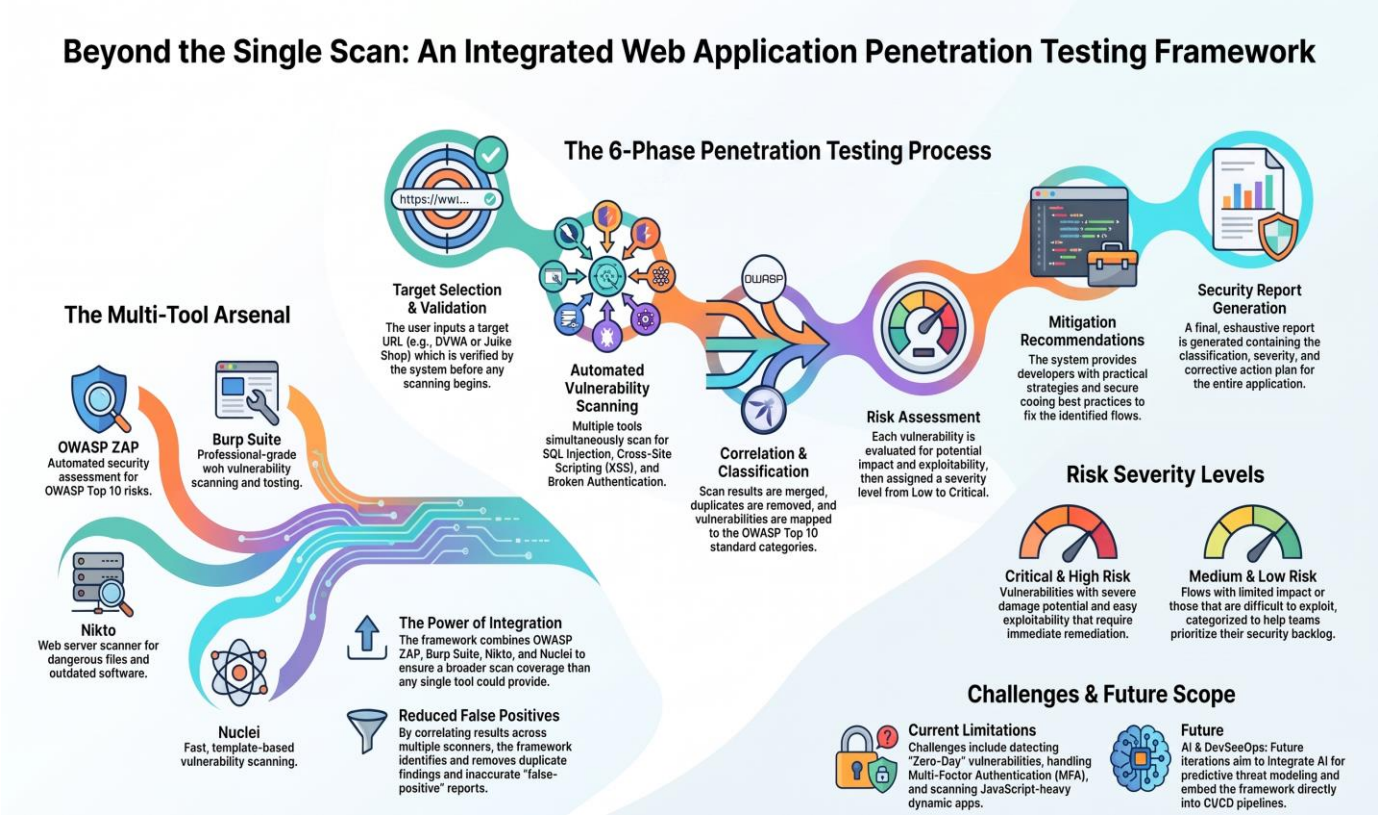
Generative AI Statement: The authors declare that generative AI was not used in the preparation or creation of this manuscript. The alternative text (alt text) accompanying the figures in this article was generated by Frontiers with the assistance of artificial intelligence. Reasonable efforts have been made to ensure the accuracy and appropriateness of the generated alt text, including review by the authors wherever possible. If you identify any inaccuracies or issues, please contact the editorial team.

Publisher's Note: The statements, opinions, and claims presented in this article are exclusively those of the authors and do not necessarily reflect the views of their affiliated institutions, the publisher, editors, or reviewers. Any products discussed or evaluated, as well as any claims made by their manufacturers, are not warranted, approved, or endorsed by the publisher.

Acknowledgements

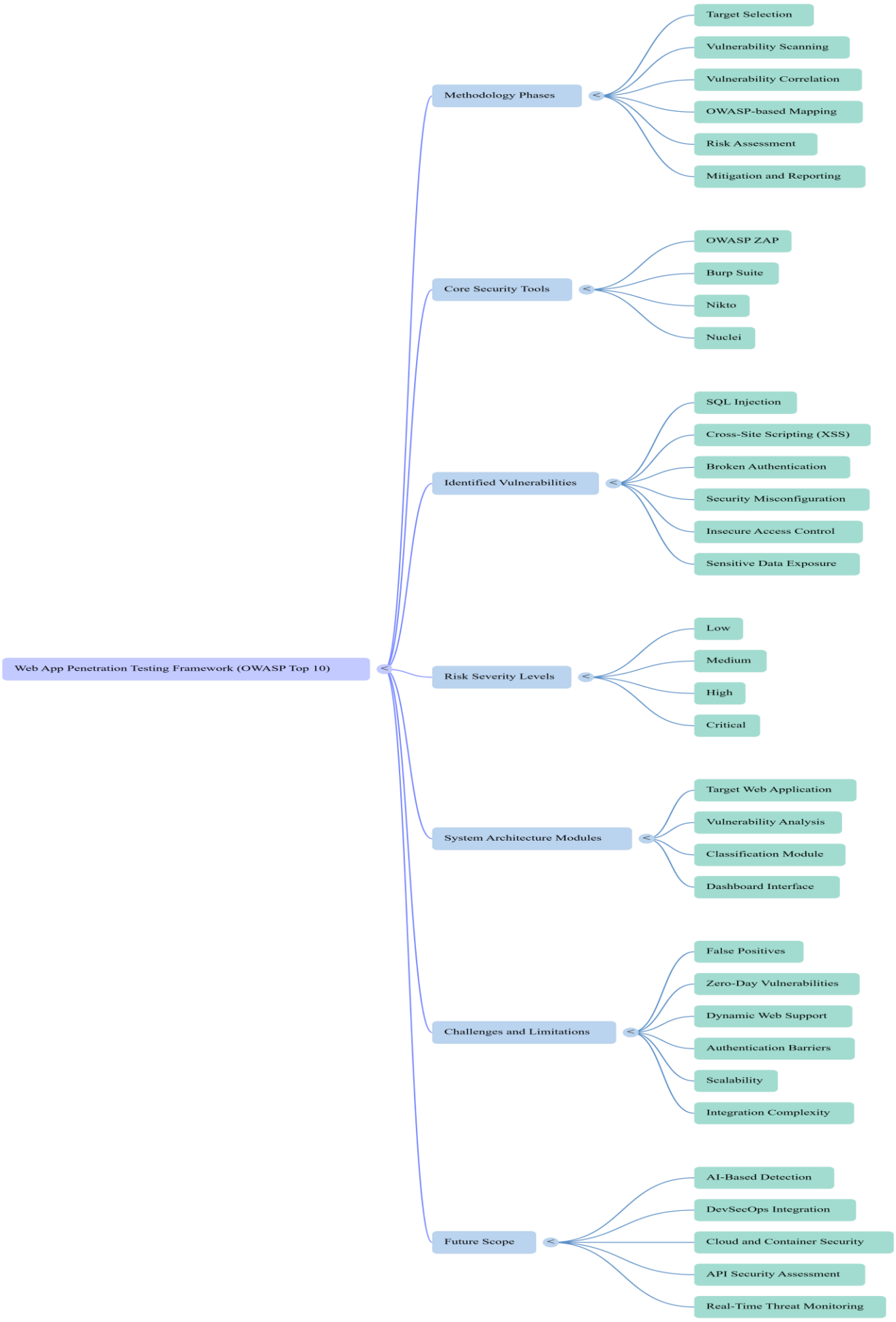
We thank everyone who inspired our work

: : Overview of the Paper for Researchers : :



Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com

ISSN: 3107 - 8605 (Online) , http://www.ijcser.com/ , Vol. 3, Issue 3 , 2026 , https://doi.org/10.63328/IJCSER-V3RI3P5



Jack Sparrow Publishers © 2026, IJCSER , All Rights Reserved
www.jacksparrowpublishers.com