AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 3(2026)

WEB APPLICATION TESTING USING OWSAP METHODOLOGY AND BURPSUITE

[1] T Tejaswi, [2]A Manoj, [3]K Kishore, [4]P Vikas
[1]Assistant Professor, [234]Students
Department of Computer Science and Engineering,
Siddhartha Institute of Technology and Sciences, Narapally, Korremula Road,
Ghatkesar, Medchal-Malkajgiri, Hyderabad, Telangana - 500088. INDIA
thadaka.tejaswi@siddhartha.org.in, 23TQ1A6232@siddhartha.co.in,
23TQ1A6231@siddhartha.co.in, 23TQ1A6204@siddhartha.co.in

Abstract

Web applications are widely used for delivering digital services such as e-commerce, banking, education, healthcare, and business management. Since these applications process sensitive information and communicate with databases, APIs, and authentication systems, security weaknesses can create serious risks. Regular security testing is therefore necessary to identify vulnerabilities and improve the overall security posture of web applications.

The proposed Web Application Testing Using OWASP Methodology and Burp Suite project provides a systematic approach for assessing the security of web applications within an authorized testing environment. The OWASP methodology provides a structured security-testing approach, while Burp Suite supports the inspection and analysis of web application traffic. Together, they help security testers identify and document potential security weaknesses.

The assessment focuses on important security areas such as authentication, authorization, session management, input validation, security configuration, access control, and protection of sensitive information. Identified observations are validated and classified according to appropriate OWASP categories. Each finding can be assigned a severity level and documented with relevant evidence and remediation recommendations.

The proposed system can provide a centralized dashboard for displaying assessment progress, vulnerability categories, severity distribution, affected components, and remediation status. Assessment results can be stored for historical comparison, allowing security teams to monitor whether previously identified issues have been addressed. The system can also generate structured reports for developers, testers, and management.

The main objective of the project is to improve the efficiency, consistency, and documentation of web application security testing. By combining the OWASP methodology with Burp Suite-assisted testing, vulnerability classification, risk assessment, visualization, and reporting, the platform provides a comprehensive defensive security workflow. All testing activities should be conducted only against applications for which explicit authorization has been obtained.

I. Introduction

Web applications have become an important part of modern digital infrastructure and are used by organizations to provide services to customers, employees, and other users. These applications frequently process confidential information and interact with databases, APIs, authentication services, and external components. As application complexity increases, the number of potential security weaknesses can also increase. Security testing is therefore an essential part of developing and maintaining reliable web applications.

Web application testing involves examining an application's functionality and security controls to identify weaknesses. Security testers may evaluate authentication mechanisms, authorization controls, session handling, input validation, encryption, security headers, and application configurations. Proper testing helps organizations discover security problems before they can be exploited and provides developers with information needed to improve the application.

The **OWASP methodology** provides a structured foundation for web application security testing. OWASP resources help security professionals understand common application-security risks and recommended testing practices. Using a recognized methodology allows testers to follow a consistent assessment process and organize findings according to established security concepts.

Burp Suite is a widely used platform for authorized web application security testing. It provides capabilities for intercepting and analyzing HTTP/HTTPS traffic and supports different stages of security assessment. Testers can use it to understand how an application processes requests and responses and to investigate potential security weaknesses within an approved scope.

The proposed project integrates OWASP-based testing methodology with Burp Suite-assisted assessment and centralized vulnerability management. The platform organizes findings, assigns risk levels, maintains assessment history, tracks remediation, and generates reports. Its main goal is to provide a structured and practical approach to web application security testing while supporting secure software development and responsible security practices.

II. Literature Survey

Web application security testing has been extensively researched because web applications are exposed to external users and frequently handle sensitive information. Research has identified vulnerabilities related to authentication, authorization, input validation, session management, cryptography, configuration, and application logic. These studies demonstrate that security should be considered throughout the application development lifecycle rather than only after deployment.

The OWASP methodology and security resources provide widely recognized guidance for assessing web application security. OWASP organizes important application-security risks and provides testing guidance that can be used by developers and security professionals. Using such a methodology helps create a consistent testing process and provides common terminology for communicating security findings.

Security testing generally combines automated tools with manual assessment. Automated tools can efficiently identify common security indicators and configuration weaknesses, while manual testing can investigate application-specific behavior and business logic. Research and industry practices suggest that combining automated assistance with expert validation can provide more meaningful assessment results.

Burp Suite is commonly used in web application security testing because it allows testers to inspect application traffic and understand interactions between browsers and web servers. Its proxy-based functionality can assist with request and response analysis, while other components can support authorized security testing. However, tool output must be interpreted carefully because automated findings may require manual validation.

Risk assessment is an important component of application security testing. Security findings can vary significantly in severity and potential impact, making it inefficient to treat every issue equally. Risk-based prioritization allows organizations to focus remediation efforts on vulnerabilities that may have greater consequences. Severity, impact, affected functionality, and application context can all contribute to prioritization.

Existing web application security tools provide scanning, traffic inspection, vulnerability identification, and reporting capabilities. However, testers may still need to manually organize findings, map them to OWASP categories, maintain assessment history, and prepare reports. The proposed system builds upon these concepts by combining OWASP methodology, Burp Suite-assisted testing, vulnerability management, risk analysis, remediation tracking, visualization, and reporting within a centralized platform.

III. System Analysis

System analysis for the proposed platform focuses on identifying the requirements necessary for systematic web application security testing. The system should provide authenticated access for authorized security testers and administrators. It should allow users to register applications and define the approved testing scope. The platform should support recording observations obtained through authorized OWASP-based testing activities. Burp Suite can be used to inspect application traffic and assist with security analysis within the approved scope. Findings should be validated and stored using a structured format. Each applicable finding should be mapped to an appropriate OWASP category or security-testing area. A risk-assessment component should classify findings according to severity and potential impact. The database should maintain applications, assessments, findings, evidence references, and remediation status. A dashboard should display vulnerability statistics, testing progress, and severity distribution. Historical assessment records should support comparison of security results over time. Finally, the system should generate structured reports containing findings, risk classifications, and recommended remediation actions.

Existing System

Traditional web application testing commonly uses a combination of Burp Suite, vulnerability scanners, browser tools, manual testing techniques, and separate documentation systems. Testers inspect application requests and responses and manually investigate potential security weaknesses. Different tools may produce different formats and levels of technical information. Findings therefore need to be manually collected, validated, and organized. OWASP categorization may also be performed separately after vulnerabilities are identified. Risk assessment is frequently documented using spreadsheets or separate reports. Evidence such as screenshots and request-response information may be maintained independently. Historical comparison between assessments can be difficult when previous results are stored as separate documents. Large applications can generate many findings that require considerable manual effort to prioritize. Report preparation can also take significant time after the assessment is complete. These limitations can make testing workflows less consistent and difficult to manage. A centralized platform can improve the organization, analysis, and reporting of assessment results.

Disadvantages of Existing System

- Requires multiple tools for complete security testing.
- Manual organization of findings is time-consuming.
- Different tools may generate inconsistent output.
- OWASP categorization may require manual effort.
- Risk classification can vary between testers.
- Evidence may be stored separately from findings.
- Difficult to maintain centralized assessment history.
- Large applications may generate many findings.
- Manual report generation requires additional effort.
- Limited centralized vulnerability visualization.
- Difficult to track remediation consistently.
- Repeated assessments can require considerable manual work.

Proposed System

The proposed system provides a centralized platform for conducting authorized web application testing using OWASP methodology and Burp Suite. The system allows security testers to register target applications and define the approved assessment scope. Testers can use Burp Suite to inspect application traffic and support security testing within the defined scope. Security observations are recorded and converted into structured vulnerability findings. Each applicable finding is mapped to an appropriate OWASP security category or testing area. The platform stores information such as vulnerability description, affected component, severity, evidence reference, and remediation status. A risk-assessment engine evaluates the potential severity and impact of identified findings. The database maintains assessment records, findings, application information, and historical results. A dashboard provides visual summaries of vulnerabilities, severity levels, OWASP categories, and assessment progress. Security teams can track findings through their remediation lifecycle and update their status after verification. The reporting module generates structured security assessment reports for technical and management stakeholders. Overall, the proposed system provides a consistent workflow from authorized testing through

vulnerability analysis, classification, risk assessment, remediation tracking, and reporting.

Advantages of Proposed System

- Provides a centralized web application testing platform.
- Supports OWASP-based security assessment.
- Supports Burp Suite-assisted testing.
- Organizes vulnerabilities in a structured format.
- Provides standardized vulnerability classification.
- Supports risk-based severity prioritization.
- Maintains assessment history.
- Provides remediation-status tracking.
- Improves visibility through dashboards and charts.
- Simplifies security report generation.
- Reduces repetitive documentation work.
- Helps developers prioritize security improvements.

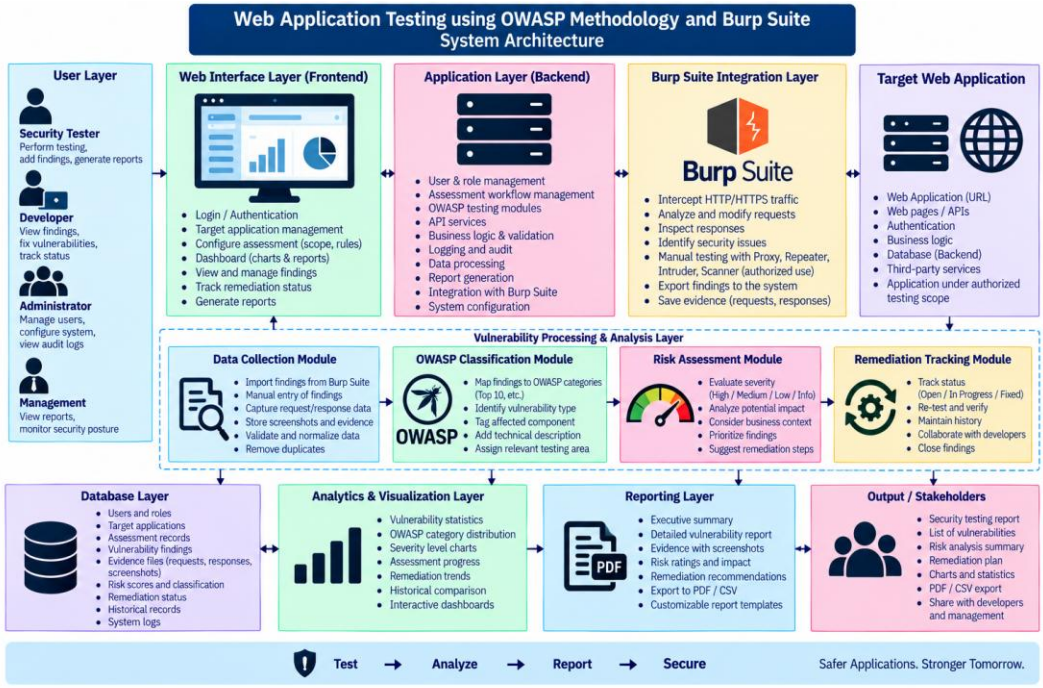
IV. Methodology

The methodology begins with authentication of the authorized security tester and registration of the target web application. The tester defines the permitted application scope and selects the appropriate OWASP-based testing areas. Burp Suite is configured for authorized traffic inspection and security testing within the defined scope. The tester examines application requests, responses, authentication behavior, authorization controls, and other relevant security characteristics. Potential security observations are recorded and reviewed for validity. Confirmed observations are converted into structured vulnerability findings. Each finding is mapped to an appropriate OWASP category or testing area. The risk engine evaluates severity, potential impact, and remediation priority. Findings and supporting evidence references are stored in the centralized database. The dashboard displays vulnerability counts, severity distribution, OWASP categories, and assessment progress. Security teams can update remediation status and verify corrected findings. Finally, the reporting module generates a structured security assessment report containing findings, risk levels, evidence references, and remediation recommendations.

System Architecture

The system architecture consists of several interconnected layers that support the complete authorized web application testing workflow. The User Layer contains security testers, developers, administrators, and authorized management users. The Web Interface Layer provides authentication, application management, assessment configuration, dashboards, vulnerability management, and report access. The Application Layer manages users, roles, testing workflows, validation, business logic, logging, and communication between system components. The Burp Suite Integration Layer supports authorized HTTP/HTTPS traffic inspection and security testing activities. The Testing Management Layer organizes OWASP-based testing areas and assessment observations. The Vulnerability Processing Layer validates, normalizes, and converts observations into structured security findings. The OWASP Classification Layer maps applicable findings to relevant OWASP categories and

testing areas. The Risk Assessment Layer evaluates severity, potential impact, and remediation priority. The Database Layer stores applications, assessments, findings, evidence references, risk scores, remediation status, and historical records. The Analytics and Visualization Layer presents vulnerability statistics, OWASP distributions, severity trends, and remediation progress. The Reporting Layer generates structured security assessment reports containing findings, evidence, risk ratings, and recommendations. The Remediation Tracking Layer monitors findings from discovery through correction and verification. Authentication, authorization, logging, and secure data-handling controls protect assessment information.



V. Result and Output

VI. Conclusion

The Web Application Testing Using OWASP Methodology and Burp Suite system provides a structured and systematic approach for identifying and analyzing security vulnerabilities in web applications. By combining the OWASP methodology with Burp Suite-assisted testing, the system helps security testers examine important areas such as authentication, authorization, input validation, session management, and security configuration.

The proposed system improves the traditional testing process by organizing identified vulnerabilities into standardized categories and assigning appropriate severity levels. It also provides centralized storage for assessment findings, evidence references, remediation status, and historical results. The dashboard makes it easier to understand vulnerability distribution and overall application security posture.

The reporting functionality further improves the usefulness of the system by generating structured security assessment reports for developers, security teams, and management. These reports can help stakeholders understand identified issues, prioritize remediation activities, and verify improvements during subsequent assessments.

Overall, the system can serve as a useful platform for authorized web application security testing, cybersecurity education, vulnerability assessment, and secure software development. Future enhancements can include integration with additional security-testing tools, automated vulnerability correlation, continuous security testing, advanced risk analysis, CI/CD integration, and automated remediation tracking. All testing should be performed only on applications for which explicit authorization has been obtained.

References

[1] Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.
[2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In The International Conference on Artificial Intelligence and Smart Environment (pp. 557-564). Cham: Springer Nature Switzerland.
[3] Sv satyakrishna, shirisha rangu ,bhargavi nalacheruve.(2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm
[4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.
[5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, “Real-Time Object Detection in Drone Surveillance Using YOLOv5,” in Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.
[6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, “Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks,” in

Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.
[7] R. D. Kumar, V. N. S.Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in Blockchain for Smart Cities, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.
[8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.
[9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," Journal of Machine and Computing, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.
[10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), Apr. 2025.
[11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.