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

WEB APPLICATION SECURITY ASSESSMENT USING OWASP TOP 10 AND BURP SUITE

[1]G Ravinder, [2]C Jitender, [3]V Bhargav Sharma, [4]N Harsha Vardhan Reddy
[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
shirisha.ravi@siddhartha.co.in, 23TQ1A6223@siddhartha.co.in,
23TQ1A6224@siddhartha.co.in, 23TQ1A6202@siddhartha.co.in

Abstract

Web applications have become an essential part of modern digital services, but their increasing complexity also introduces significant security risks. Applications communicate with databases, APIs, authentication systems, and external services, creating multiple areas that require security evaluation. Vulnerabilities in these components can result in unauthorized access, data exposure, account compromise, or disruption of services. Therefore, systematic web application security assessment is essential for identifying and addressing potential weaknesses.

The proposed Web Application Security Assessment Using OWASP Top 10 and Burp Suite project provides a structured methodology for evaluating web application security within an authorized testing environment. The system uses the OWASP Top 10 as a reference framework for organizing common categories of web application security risks. Burp Suite can be used as an assessment tool for inspecting application traffic, analyzing requests and responses, and supporting authorized security testing.

The assessment process examines security-related aspects such as authentication, authorization, input validation, session management, security configuration, sensitive-data protection, and other relevant application behaviors. Findings are documented and mapped to appropriate OWASP Top 10 categories. This structured approach makes it easier for developers and security analysts to understand the nature and potential impact of identified security issues.

The proposed platform can organize assessment findings according to severity and priority. A centralized dashboard can display the number of findings, OWASP categories, severity distribution, affected application components, and assessment progress. The system can also maintain assessment history and generate structured reports containing findings, evidence, risk levels, and recommended remediation actions.

The primary objective of the project is to improve the effectiveness and consistency of web application security assessment. By combining the OWASP Top 10 framework with Burp Suite-assisted testing, structured analysis, risk classification, visualization, and reporting, the system can help organizations identify security weaknesses and improve application security.

I. Introduction

Web applications are widely used to provide services such as online shopping, banking, education, healthcare, communication, and business management. These applications process large amounts of information and frequently interact with databases, APIs, authentication systems, and third-party services. As the number of application components increases, the possibility of security weaknesses also increases. A vulnerability in a web application can potentially affect confidentiality, integrity, and availability of information.

Web application security assessment is the process of systematically examining an application to identify potential security weaknesses. Security professionals may evaluate authentication mechanisms, access controls, input validation, session management, security configurations, encryption, and application behavior. Regular assessment allows organizations to identify security issues before they are exploited and provides developers with information needed for remediation.

The **OWASP Top 10** is widely used as an awareness and risk-classification reference for common web application security risks. It provides security teams and developers with a structured way to discuss important application-security categories. Mapping assessment findings to OWASP categories can make technical security issues easier to understand and can help organizations prioritize security improvements.

Burp Suite provides a collection of tools that can support authorized web application security testing. It can assist security professionals in observing and analyzing HTTP traffic and investigating application behavior. When used within an approved testing scope, Burp Suite can support both manual and automated aspects of a security assessment. However, interpreting testing results still requires appropriate technical knowledge and validation.

The proposed project combines the OWASP Top 10 framework with Burp Suite-assisted assessment and centralized result management. The system is designed to organize findings, classify their severity, maintain assessment records, visualize security results, and generate reports.

II. Literature Survey

Web application security has been extensively studied because web-based systems are exposed to users and external networks and often handle sensitive information. Research has identified several recurring classes of application vulnerabilities involving authentication, access control, input processing, security configuration, cryptographic protection, and application logic. These studies demonstrate the importance of incorporating security assessment throughout the software development and deployment lifecycle.

The OWASP Top 10 has become an important reference for understanding and communicating common web application security risks. It provides security professionals and developers with a common classification framework for discussing vulnerabilities and security weaknesses. Mapping assessment findings to standardized categories can improve communication between development, security, and management teams and can support risk-based remediation planning.

Security testing methodologies commonly combine automated scanning with manual testing. Automated tools can efficiently identify common configuration problems and potential security indicators, while manual testing can provide deeper analysis of application-specific behavior. Research and industry practices indicate that neither approach alone is sufficient for every application, making a combination of automated assistance and expert analysis valuable.

Burp Suite is widely used as a web application security testing platform for examining application traffic and behavior. Its proxy-based workflow allows authorized testers to observe requests and responses between clients and web applications. Additional testing capabilities can support investigation of application functionality and security controls. The effectiveness of such tools depends on proper configuration, testing methodology, and analyst interpretation.

Risk assessment is another important area in application security research. Identifying a vulnerability is only one part of the security process; organizations must also determine its severity, potential impact, likelihood, and remediation priority. Risk-based classification allows security teams to focus their resources on findings that may have greater consequences. Structured severity levels can also make assessment reports easier for stakeholders to understand.

Existing web application security platforms provide scanning, traffic analysis, vulnerability identification, dashboards, and reporting capabilities. However, assessment workflows may still require analysts to manually map findings to security categories, organize evidence, compare historical results, and prepare reports. The proposed system builds on these concepts by integrating OWASP Top 10 classification, Burp Suite-assisted assessment, finding management, risk prioritization, visualization, historical tracking, and reporting within a unified defensive workflow.

III. System Analysis

System analysis for the proposed platform focuses on identifying the requirements for conducting a structured web application security assessment. The system should provide authenticated access to authorized security analysts and testers. It should allow users to define the approved application scope before beginning an assessment. Burp Suite can be used within the authorized scope to inspect application traffic and support security testing activities. Assessment observations should be collected and organized into structured findings. Each finding should be mapped to a relevant OWASP Top 10 category where applicable. The system should record appropriate evidence and contextual information without unnecessarily storing sensitive application data. A risk assessment component should classify findings according to predefined severity criteria. The database should maintain applications, assessments, findings, and historical records. A dashboard should display vulnerability categories, severity distribution, and assessment progress. The system should provide remediation recommendations to assist developers and security teams. Finally, the reporting module should generate structured assessment reports for authorized stakeholders.

Existing System

Traditional web application security assessment commonly depends on a combination of manual testing, Burp Suite, vulnerability scanners, browser developer tools, and other security utilities. Analysts may inspect application traffic and manually investigate suspicious behavior. Different tools can generate results in different formats, requiring additional effort to organize the findings. Security teams may manually map identified issues to OWASP Top 10 categories. Severity classification may also depend on individual analyst judgment or separate risk-assessment processes. Evidence and screenshots may be stored separately from technical findings. Historical comparison between assessments can be difficult when results are maintained as individual documents. Large applications can generate many observations that require considerable manual review. Report preparation can also consume significant time after the technical assessment has been completed. Basic workflows may provide limited centralized visualization of vulnerability trends. These limitations can make security assessment less consistent and more difficult to manage across multiple applications. A centralized platform can improve organization, analysis, tracking, and reporting.

Disadvantages of Existing System

- Requires multiple tools for different assessment activities.
- Significant manual effort is required to organize findings.
- Findings may be stored in different formats.
- Manual OWASP Top 10 classification can be time-consuming.
- Severity assessment may vary between analysts.
- Evidence may be stored separately from findings.
- Historical assessment comparison can be difficult.
- Large applications can produce extensive assessment results.
- Manual report preparation requires additional effort.
- Limited centralized visualization in basic workflows.
- Difficult to maintain a unified vulnerability database.
- Repeated assessments may require considerable manual work.

Proposed System

The proposed system provides a centralized platform for conducting authorized web application security assessments using the OWASP Top 10 framework and Burp Suite-assisted testing. The system allows authenticated security analysts to define the application and testing scope before beginning an assessment. Burp Suite can support authorized traffic inspection and security testing within the defined scope. Assessment observations are collected and converted into structured security findings. The system maps relevant findings to corresponding OWASP Top 10 categories. A finding-management module records descriptions, affected components, evidence, severity, and assessment status. A risk engine evaluates findings using predefined criteria and helps prioritize remediation. The database stores application information, assessment records, findings, and historical results. A dashboard presents vulnerability categories, severity distribution, open findings, and assessment progress. Analysts can review individual findings and update their remediation status. The reporting module generates structured security assessment reports for developers, security teams, and management. Historical assessment information can be used to monitor improvements and recurring issues. Overall, the proposed system provides a consistent workflow

from authorized testing through vulnerability classification, risk analysis, visualization, and reporting.

Advantages of Proposed System

- Provides a centralized web application security assessment workflow.
- Uses OWASP Top 10 for structured vulnerability classification.
- Supports Burp Suite-assisted authorized testing.
- Organizes findings in a consistent format.
- Provides risk-based severity classification.
- Improves visibility into application security posture.
- Maintains assessment and vulnerability history.
- Supports finding-status and remediation tracking.
- Provides centralized dashboards and visual summaries.
- Simplifies security report generation.
- Reduces repetitive manual documentation.
- Helps development teams prioritize security remediation.

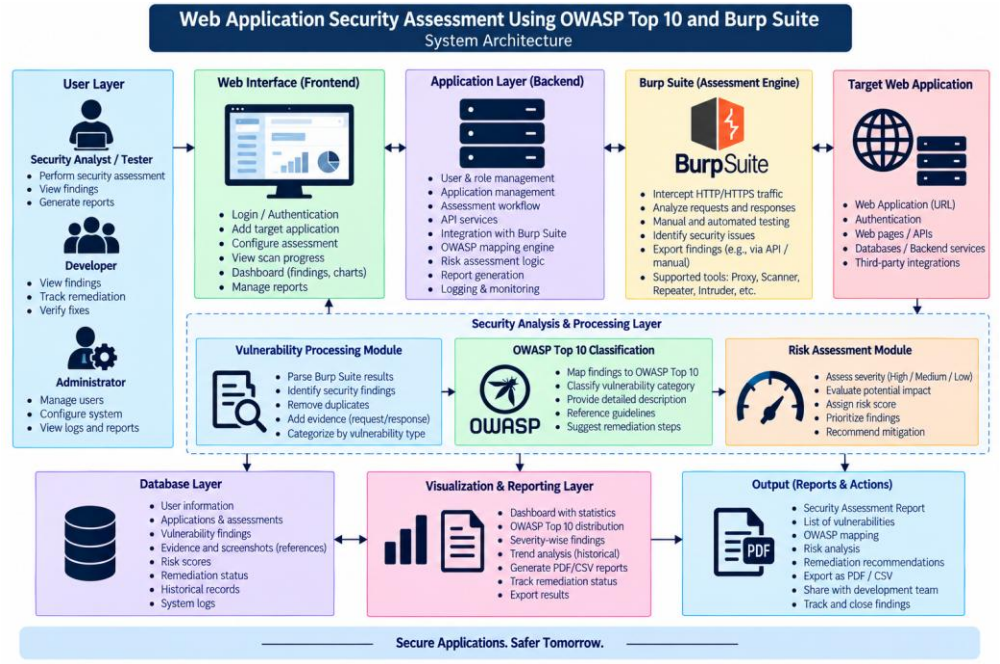
IV. Methodology

The methodology begins with authentication of the authorized security analyst and definition of the application testing scope. The analyst configures the assessment according to the approved application and testing requirements. Burp Suite is then used within the authorized scope to inspect application traffic and support security testing. Relevant application responses and security observations are reviewed by the analyst. Potential security issues are documented as structured findings within the platform. Each finding is mapped to an appropriate OWASP Top 10 category where applicable. The system records supporting evidence and contextual information required for validation. A risk engine evaluates each finding according to predefined severity and impact criteria. Findings are prioritized to help security teams determine which issues require greater attention. The database stores assessment results, findings, statuses, and historical information. The dashboard presents vulnerability statistics, category distribution, severity levels, and remediation progress. Finally, the reporting module generates a structured report containing findings, risk classifications, evidence, and recommended remediation actions.

System Architecture

The Web Application Security Assessment architecture consists of several interconnected layers that support the complete authorized assessment workflow. The User Layer contains security analysts, testers, developers, and authorized administrators who interact with the platform according to their assigned roles. The Web Interface Layer provides authentication, application management, assessment configuration, dashboards, finding management, and report access. The Application Layer manages user requests, authorization, assessment workflows, business logic, logging, and communication with assessment components. The Burp Suite Assessment Layer supports authorized HTTP traffic inspection and security testing within the defined application scope. The Assessment Processing Layer organizes assessment observations and converts them into structured security findings. The OWASP Classification Layer maps applicable findings to relevant OWASP Top 10

categories and security classifications. The Risk Assessment Layer evaluates severity, potential impact, and remediation priority. The Database Layer stores users, applications, assessment configurations, findings, evidence references, remediation status, and historical assessment records. The Analytics and Visualization Layer presents vulnerability distributions, OWASP categories, severity trends, and remediation statistics. The Reporting Layer generates structured PDF or other authorized assessment reports for stakeholders. Access control and logging mechanisms help ensure that only authorized users can perform or review assessments.



V. Result and Output

(a) Assessment Configuration (b) Scan Result – Identified Vulnerabilities (c) Output – Generated Report (PDF)

VI. Conclusion

The Web Application Security Assessment Using OWASP Top 10 and Burp Suite system provides a structured approach for identifying, analyzing, and managing security weaknesses in web applications. By using the OWASP Top 10 as a standardized classification framework and Burp Suite for authorized security testing, the system helps security analysts understand potential vulnerabilities and their impact.

The proposed system reduces manual effort by organizing assessment findings, mapping vulnerabilities to OWASP categories, assigning severity levels, and maintaining assessment history. The dashboard provides clear visualization of vulnerability distribution and risk levels, while the reporting module makes it easier to communicate findings and recommended remediation actions to development and security teams.

Overall, the system can serve as a useful solution for authorized web application security testing, cybersecurity education, secure software development, and vulnerability assessment. Future enhancements can include automated vulnerability correlation, improved risk-scoring mechanisms, continuous security monitoring, integration with CI/CD pipelines, and automated remediation tracking. All assessments should be conducted 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.