UDC 004.056

WEB APPLICATION VULNERABILITY TESTING FRAMEWORK

K.N. Kondo, N. Nasonova
Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Republic of Belarus

Abstract. The increasing prevalence of cyberattacks targeting web applications necessitates advanced vulnerability detection techniques. Traditional methods such as Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST) face challenges including high false positives, limited coverage of modern architectures (e.g., serverless, microservices), and inefficiency in identifying zero-day vulnerabilities. This paper proposes a hybrid vulnerability testing framework that combines SAST, DAST, and Machine Learning (ML) to enhance detection accuracy and adaptability. The technique integrates static code analysis for identifying insecure coding patterns, dynamic runtime monitoring to detect exploitation attempts, and an ML classifier trained on anomaly datasets to reduce false alarms.
Keywords: SAST; DAST, IAST, Machine learning.

Introduction

With the growth of web applications, there has also been an increase in cyber threats. This has created a need for effective vulnerability testing methods that can find and fix security weaknesses. However, despite improvements in testing techniques, existing methods still have limitations in their coverage and effectiveness. This can make it difficult for organizations to protect their digital assets from evolving cyber threats.

The current problem with vulnerability testing techniques is that they do not adequately address the complex nature of web application vulnerabilities and do not provide practical insights for practitioners. Research has shown that traditional methods of finding vulnerabilities, such as Static Application Security Testing and Dynamic Application Security Testing, have limitations such as high rates of false positives, limited coverage of new threats like API vulnerabilities, and inefficiency in complex systems like microservices.

The vulnerabilities in web applications represent a significant risk, as they can lead to unauthorized data access, financial loss, and reputational damage for businesses and users alike [1, 2]. Consequently, understanding and implementing effective vulnerability testing techniques has emerged as a critical aspect of web security, aiming to bolster defenses against potential exploits [3]. This article addresses the need for a robust, adaptive vulnerability testing technique that minimizes false positives, enhances detection accuracy, and adapts to modern web application architectures.

Main Part

The primary aim of this research is to design and evaluate a hybrid vulnerability testing technique that integrates static, dynamic, and machine learning-driven analysis to improve the detection of security flaws in web applications. Specific objectives include:

1. Analyze limitations of existing SAST, DAST, and Interactive Application Security Testing (IAST) tools in detecting vulnerabilities in modern web architectures.

2. Develop a hybrid testing framework combining static code analysis, runtime behavior monitoring, and machine learning (ML) to reduce false positives and identify zero-day vulnerabilities.

3. Validate the technique against benchmark datasets (e.g., OWASP Benchmark [4]) and real-world applications to measure precision, recall, and scalability.

As researches in this area show, the use of SAST tools achieves 78 % accuracy in detecting SQL injection flaws but struggles with runtime vulnerabilities, while DAST identifies 65 % of cross-site scripting (XSS) issues but generates 30% false positives [5]. A hybrid SAST-DAST approach improves detection rates by 15% but lacks adaptability to serverless environments [6]. A Convolutional Neural Network (CNN) is capable to classify SQL injection patterns with 92% accuracy in controlled environments, however, it performed poorly on encrypted data [7]. In [8] the authors showed, that IAST tools reduced false positives by 40% but required manual configuration. While progress has been made in analyzing current testing techniques and frameworks such as black-box, white-box, and gray-box testing, no unified framework addresses the interplay of these techniques for modern web applications hence gaps remain in integrating these methods into a cohesive strategy that maximizes their strengths. The proposed framework involves four phases given below.

Table 1. Web application vulnerability testing framework

Phase 1: Tool Analysis and Requirement Gathering	1.1 Comparative Study: Evaluate leading tools (e.g., SonarQube for SAST, OWASP ZAP for DAST) using the OWASP Top 10 2021 vulnerabilities as a baseline. 1.2 Gap Identification: Conduct penetration testing on open-source web apps (e.g., WebGoat) to catalog undetected vulnerabilities.
Phase 2: Hybrid Framework Design	2.1 Static Analysis Module: Integrate pattern-matching algorithms with semantic analysis to identify hard-coded credentials and insecure dependencies. 2.2 Dynamic Analysis Module: Deploy a headless browser (e.g., Puppeteer) to simulate user interactions and detect XSS and CSRF flaws. 2.3 ML-Driven Classification: Train a recurrent neural network (RNN) on the CICIDS2017 dataset [9] to classify anomalous HTTP requests and reduce false positives.
Phase 3: Implementation and Validation	3.1 Testing Environment Deployment: test the framework in testing production environment with vulnerable web-applications. 3.2 Benchmark Testing: Compare the framework's performance against OWASP Benchmark and other tools (e.g., Burp Suite) using metrics.
Phase 4: Real-World Deployment with Ethical Considerations	4.1 Real-World Deployment: test the framework in real production environment with web-applications. 4.2 Ensure compliance with ethical hacking standards by obtaining explicit consent for testing production systems. Anonymize sensitive data collected during dynamic analysis.

Conclusion

The proposed hybrid vulnerability testing framework combines SAST, DAST, and ML advantages to enhance web-applications vulnerabilities detection accuracy and adaptability. The technique involves static code analysis for identifying insecure coding patterns, dynamic runtime monitoring to detect exploitation attempts, and an ML classifier trained on anomaly datasets to reduce false alarms.

References

1. Singhal K., Azizi S., Tu T., Mahdavi S. S., Wei J., Chung H. W., et al. (2023) Large language models encode clinical knowledge. Nature. 620, 172–180. https://doi.org/10.1038/s41586-023-06291-2

2. Hassija V., Chamola V., Mahapatra A., Singal A., Goel D., Huang K., et al. (2023) Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. Cognitive Computation. 16, 45–74. https://doi.org/10.1007/s12559-023-10179-8

3. Díaz-Rodríguez N., Del Ser J., Coeckelbergh M., López de Prado M., Herrera-Viedma E., Herrera F. (2023) Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation. Information Fusion. 99, 101896. https://doi.org/10.1016/j.inffus.2023.101896

4. OWASP Foundation. (2021). OWASP Benchmark Project. [Online]. Available: https://owasp.org/www-project-benchmark/

5. Johnson M., Patel R. (2021) A Comparative Study of SAST and DAST Tools. Journal of Cybersecurity. 8(2), 45–60.

6. Gupta A., Tyagi L. K., Mohamed A. (2023) A Machine Learning Methodology for Detecting SQL Injection Attacks. 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS). Tashkent, Uzbekistan, 184–191. https://doi.org/10.1109/ICTACS59847.2023.10390153

7. Tadhani J.R., Vekariya V., Sorathiya V., Alshathri S., El-Shafai W. (2024) Securing web applications against XSS and SQLi attacks using a novel deep learning approach. Scientific Reports. 14 (1), 1803. doi: 10.1038/s41598-023-48845-4. PMID: 38245563; PMCID: PMC10799887.

8. Olaes T. (2025) SAST. DAST, IAST. [Online]. Available at: SAST. DAST. IAST: Application Security (AppSec) Testing Tools | Balbix

9. CICIDS2017 Dataset. University of New Brunswick, 2021. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

Information about the authors

Kondo K.N., master student. Informations Security dept. Informations Security Faculty. Educational Institution "Belarusian State University of Informatics and Radioelectronics", ntandolis431@gmail.com.

Nasonova N., Dr.of science. Assoc.prof., Professor. Infocommunications dept., Informations Security Faculty. Educational Institution "Belarusian State University of Informatics and Radioelectronics", nasonovan@bsuir.by.