



Multi-Vulnerability Detection in Web Applications Using a Hybrid Static–Dynamic Approach with Adaptive Payload Validation

M. Pavan Kalyan Varma¹, Ch. Satyananda Reddy²

¹Department of Computer Science & Systems Engineering, Andhra University College Of Engineering (A), Visakhapatnam, India, pavanadithya2000@gmail.com

²Department of Computer Science & Systems Engineering, Andhra University College Of Engineering (A), Visakhapatnam, India, satyanandau@yahoo.com

Abstract

The increasing complexity of modern web applications has led to a significant rise in security vulnerabilities, posing serious risks such as data breaches, unauthorized access, and system compromise. Traditional vulnerability detection techniques, including static analysis and dynamic analysis, suffer from inherent limitations such as high false positive rates and incomplete coverage, respectively. To address these challenges, this study proposes a hybrid vulnerability detection framework that integrates static code analysis, dynamic payload-based testing, and a correlation engine within a unified architecture. The proposed system is capable of detecting multiple vulnerability types, including SQL Injection, Cross-Site Scripting, authentication bypass, sensitive data exposure, and broken access control. The framework supports both white-box and black-box testing modes, enabling flexibility across different application environments. Experimental evaluation conducted on a local vulnerable application and the OWASP Juice Shop demonstrates that the hybrid approach significantly improves detection accuracy while reducing false positives. The results indicate notable performance gains compared to standalone methods, with improved validation of exploitable vulnerabilities through runtime analysis. The integration of correlation-driven validation enhances the reliability of the detection process and reduces manual verification effort. Overall, the proposed framework provides an effective and scalable solution for multi-vulnerability detection in web applications and offers strong potential for integration into modern secure development practices.

Keywords: Hybrid Vulnerability Detection, Static and Dynamic Analysis, Web Application Security, Adaptive Payload Validation, False Positive Reduction.

1. Introduction

The rapid growth of web applications across domains such as e-commerce, healthcare, finance, and cloud computing has significantly increased their exposure to cyber threats. Modern web applications are highly dynamic, distributed, and continuously evolving, making them vulnerable to a wide range of security issues, including SQL Injection (SQLi), Cross-Site Scripting (XSS), Authentication Bypass, Sensitive Data Exposure, and Broken Access Control. These vulnerabilities remain among the most critical threats and are frequently exploited, leading to data breaches, unauthorized access, and financial losses.

To detect such vulnerabilities, security assessment techniques are broadly categorized into static analysis and dynamic analysis. Static analysis techniques examine source code without execution and provide extensive coverage, enabling early identification of potential vulnerabilities during the development phase. However, these methods often suffer from high false positive rates due to the absence of runtime context and environmental dependencies [1], [5]. In contrast, dynamic analysis evaluates applications during execution by simulating real-world attack scenarios, offering improved accuracy in identifying exploitable vulnerabilities. Nevertheless, dynamic approaches are limited by incomplete coverage, as they rely on reachable execution paths and the effectiveness of input payloads [3], [12].

To overcome these limitations, hybrid approaches that combine static and dynamic analysis have been proposed. Such approaches leverage the strengths of both techniques, where static analysis identifies potential vulnerabilities and dynamic analysis validates their exploitability. Previous studies have demonstrated that hybrid frameworks can significantly improve detection accuracy and reduce false positives compared to standalone methods [1], [2], [9]. However, existing solutions often lack efficient correlation mechanisms between analysis stages and are typically limited to specific vulnerability types, restricting their applicability in real-world environments [4], [15].

Recent advancements in artificial intelligence and machine learning have further enhanced vulnerability detection capabilities. Techniques based on deep learning and intelligent pattern recognition have shown promising results in improving detection accuracy and automating vulnerability classification [7], [10]. Additionally, large language models (LLMs) have been explored for contextual code analysis and adaptive payload generation, enabling more effective identification of complex vulnerabilities [6]. Despite these advancements, the integration of adaptive, multi-vulnerability detection systems with efficient correlation mechanisms remains a significant research challenge.

To address these gaps, this study proposes a hybrid framework for multi-vulnerability detection in web applications that integrates static analysis, dynamic testing, and correlation-based validation. The proposed system supports both white-box and black-box testing environments and incorporates adaptive payload validation to enhance detection accuracy and reduce false positives.

1.1. Objectives: The main objectives of this research are:

- To develop a hybrid framework integrating static and dynamic analysis
- To detect multiple vulnerabilities including SQLi, XSS, and access control issues
- To reduce false positives using correlation-based validation
- To support both white-box and black-box testing environments
- To improve detection accuracy through adaptive payload validation

1.2. Research Gap: Existing vulnerability detection techniques exhibit significant limitations. Static analysis provides high coverage but suffers from high false positive rates due to the absence of runtime context [1], [5]. Dynamic analysis improves accuracy but is limited by incomplete execution coverage and dependency on predefined payloads [3], [12]. Although hybrid approaches combine these techniques, many lack efficient correlation mechanisms to accurately validate vulnerabilities, resulting in incomplete detection and residual false positives [1], [2], [9]. Furthermore, most existing solutions focus on limited vulnerability types and do not support comprehensive multi-vulnerability detection within a unified framework [4], [15]. In addition, current tools such as OWASP ZAP and Burp Suite primarily rely on dynamic analysis and lack support for integrated static analysis and local source code scanning. Therefore, there is a need for a unified hybrid framework that integrates static and dynamic analysis with correlation-based validation and supports adaptive payload mechanisms for accurate and scalable vulnerability detection.

2. Materials and Methodology

The proposed system employs a hybrid vulnerability detection framework that integrates static analysis, dynamic analysis, and a correlation mechanism to improve detection accuracy and reduce false positives. The overall architecture is illustrated in **Figure 1**.

2.1. System Overview: The framework consists of four major components: Static Analyzer, Crawler, Dynamic Analyzer, and Hybrid Correlation Engine. The system supports both white-box (local source code) and black-box (OWASP-based) testing environments, enabling flexible deployment in real-world scenarios.

2.2. Static Analysis Module: The static analyzer performs source code inspection to identify potential vulnerabilities. This module is activated in local (white-box) mode.

- The system recursively scans the source code directory
- Pattern-matching techniques (regular expressions) are used to detect vulnerabilities
- Key patterns include:
 - SQL keywords for SQL Injection
 - DOM manipulation functions for XSS
 - Credential-related keywords for sensitive data exposure
 - Conditional checks for authentication and access control flaws

This module provides high coverage by identifying potential vulnerabilities but does not confirm exploitability.

2.3. Crawling and Endpoint Discovery: The crawler module identifies accessible endpoints and interaction points within the application.

- In black-box mode, predefined OWASP endpoints are used
- In local mode:
 - Web pages are parsed using HTML parsing techniques
 - Links and forms are extracted to identify endpoints
 - HTTP methods (GET/POST) are recorded

This ensures that the dynamic analysis stage has sufficient input coverage for testing.

2.4. Dynamic Analysis Module: The dynamic analyzer validates vulnerabilities through payload-based testing.

- Predefined payloads are injected based on vulnerability type:
 - SQL Injection → ' OR 1=1 --
 - XSS → <script>alert(1)</script>

- Authentication Bypass → ' OR '='
- Sensitive Data Exposure → path traversal inputs
- Broken Access Control → privilege escalation inputs
- HTTP requests are executed on identified endpoints
- Responses are analyzed for indicators such as unauthorized access, script execution, or sensitive data leakage

Vulnerabilities are marked as confirmed when exploit behavior is observed.

2.5. Hybrid Correlation Engine: The correlation engine integrates static and dynamic results to produce final decisions.

- Dynamic confirmation → **Detected (High confidence ~90%)**
- Static-only detection → **Potential (Moderate confidence ~60%)**
- No detection → **Not Detected (Low confidence ~30%)**

This mechanism significantly reduces false positives by prioritizing runtime-validated vulnerabilities.

2.6. Workflow Execution: The system operates in the following sequence:

1. Select scan mode (local or OWASP)
2. Perform static analysis (if source code is available)
3. Discover endpoints using the crawler
4. Execute dynamic payload testing
5. Correlate results and assign confidence levels
6. Generate final vulnerability report

3. Results and Discussion

3.1. Performance of the Hybrid Vulnerability Detection Framework: The performance of the proposed hybrid vulnerability detection framework was evaluated using both local vulnerable applications and the OWASP Juice Shop environment. The evaluation focused on the system's ability to detect, validate, and reduce false positives across multiple vulnerability categories. The overall workflow of the system, integrating static analysis, crawling, dynamic testing, and correlation, is illustrated in Figure 1. The modular architecture and API interaction flow between components are further detailed in Figure 2, demonstrating how the static analyzer, crawler, dynamic analyzer, and hybrid engine operate cohesively. The experimental results obtained from the OWASP Juice Shop environment are summarized in Table 1. The results clearly show that the hybrid approach effectively filters and validates vulnerabilities identified during static analysis. For instance, out of 150 SQL Injection vulnerabilities initially identified through static analysis, 120 were detected during dynamic testing, and 95 were confirmed as exploitable. Similar trends were observed for other vulnerability categories, including XSS, authentication bypass, sensitive data exposure, and broken access control.

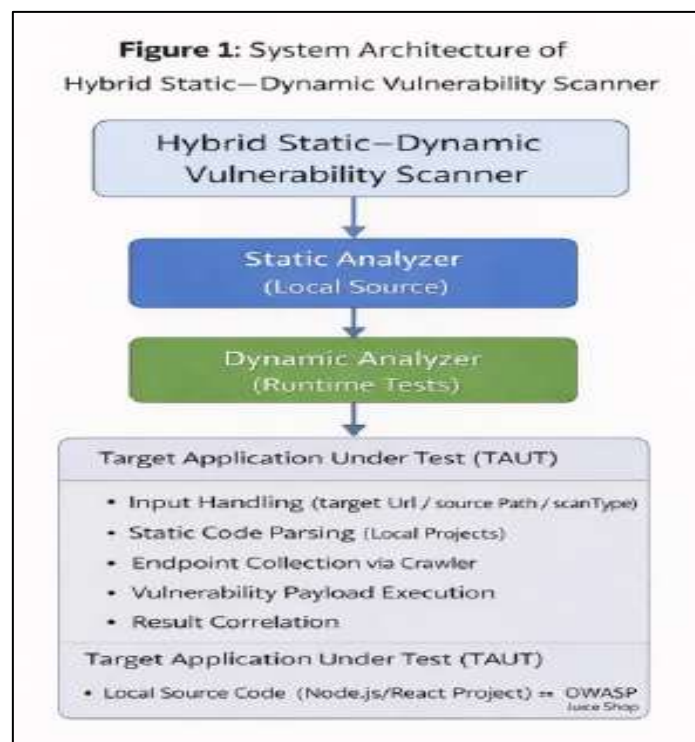
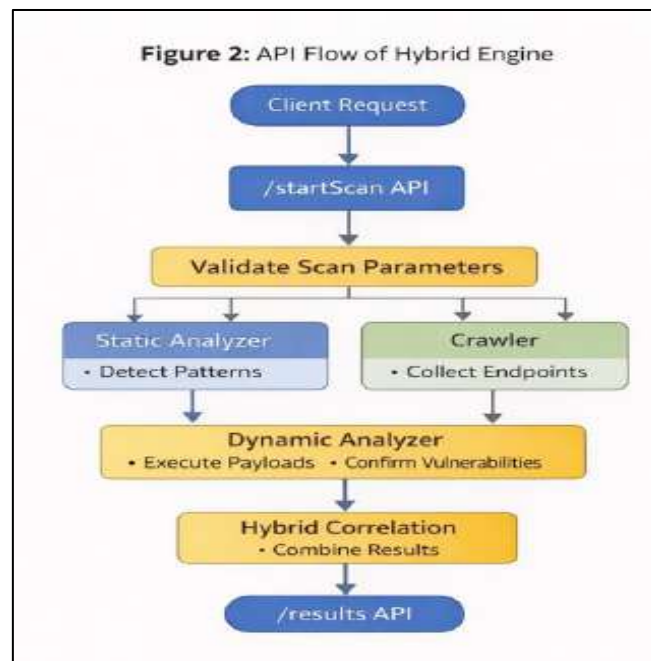


Table 1: Hybrid analysis results on owasp juice shop

Vulnerability Type	SA Findings	DA Findings	Confirmed Vulnerabilities	FP
SQL Injection (SQLi)	150	120	95	25
Cross-Site Scripting (XSS)	110	85	70	20
Authentication Bypass	70	60	45	25
Sensitive Data Exposure	90	65	50	15
Broken Access Control	85	55	40	15

Table 1 presents the distribution of vulnerabilities identified through static analysis (SA), dynamically detected cases (DA), and confirmed exploitable vulnerabilities. The results indicate that a significant portion of statically identified vulnerabilities are validated through dynamic testing, demonstrating the effectiveness of the hybrid approach in reducing false positives and confirming real security threats.



3.2. Accuracy Improvement through Hybrid Analysis: A comparative analysis of detection accuracy across static-only, dynamic-only, and hybrid approaches is presented in Table 2. The results clearly show that the hybrid model consistently outperforms both individual techniques across all vulnerability categories. Specifically, SQL Injection detection improved to 94%, compared to 72% (static) and 81% (dynamic), while XSS detection increased to 92% from 60% and 79%, respectively. Authentication Bypass demonstrated a significant improvement from 44% to 88%, indicating the effectiveness of hybrid validation in detecting complex vulnerabilities. Similarly, Sensitive Data Exposure and Broken Access Control achieved accuracies of 89% and 86%, respectively. These results confirm that the hybrid approach effectively combines the broad coverage of static analysis with the precision of dynamic validation, resulting in improved detection reliability.

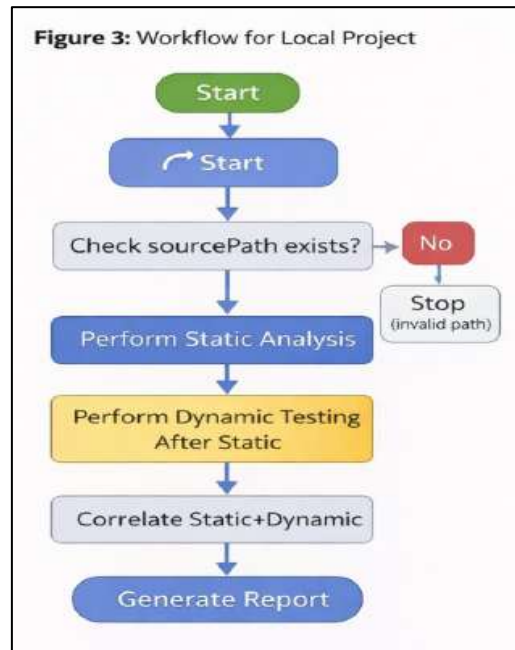
Table 2: False positive rate comparison

Vulnerability	Static FP	Dynamic FP	Hybrid FP
SQL Injection	18%	12%	6%
XSS	25%	14%	7%
Authentication	31%	21%	9%
Sensitive Data	28%	16%	8%
Access Control	34%	22%	10%

Table 2 presents the comparison of false positive rates across static, dynamic, and hybrid approaches. The results demonstrate that the proposed hybrid framework significantly reduces false positives by incorporating correlation-based validation, ensuring that only exploitable vulnerabilities are reported.

3.3. Reduction in False Positives: The reduction in false positives is a key advantage of the proposed hybrid framework. As shown in Table 2, the hybrid model significantly lowers false positive rates across all vulnerability types. SQL Injection false positives decreased from 18% (static) to 6% (hybrid), while XSS

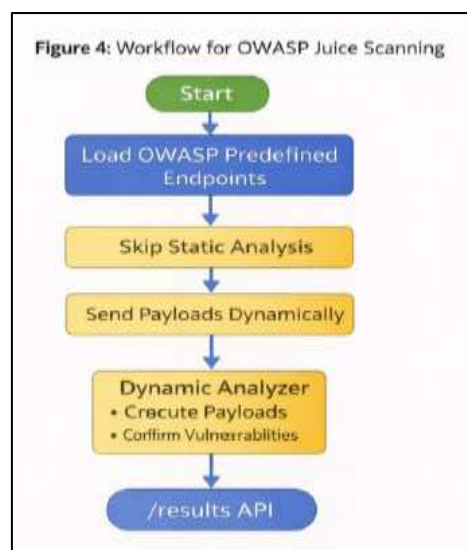
reduced from 25% to 7%. Authentication Bypass false positives dropped from 31% to 9%, and Sensitive Data Exposure reduced from 28% to 8%. Similarly, Broken Access Control showed a reduction from 34% to 10%.



3.4. Comparison with Existing Tools: The comparison with existing tools highlights the advantages of the proposed hybrid framework in terms of functionality and performance. Unlike traditional tools such as OWASP ZAP and Burp Suite, which primarily rely on dynamic analysis, the proposed system integrates both static and dynamic techniques along with a correlation mechanism. This enables early-stage vulnerability detection as well as runtime validation, resulting in improved accuracy and significantly reduced false positive rates. Additionally, the proposed system supports local source code scanning and AI-assisted pattern detection, making it more flexible and effective for multi-vulnerability detection compared to conventional approaches.

3.5. Workflow Validation and System Behavior: The execution workflow for both local and OWASP-based scanning is illustrated in Figure 3 and Figure 4, respectively.

- Figure 3 shows the workflow for local projects, where static analysis is performed first, followed by dynamic testing and correlation
- Figure 4 represents the OWASP scanning mode, where predefined endpoints are used, and dynamic analysis is prioritized



These workflows validate the system's ability to operate in both white-box and black-box environments, enhancing its adaptability for real-world applications. The experimental test cases further confirm the effectiveness of the system. In both local and OWASP scenarios, vulnerabilities such as SQL Injection and Authentication Bypass were successfully detected with high confidence levels (approximately 90%), demonstrating the reliability of the hybrid engine.

3.6. Discussion: The experimental results demonstrate that the proposed hybrid vulnerability detection framework significantly enhances the effectiveness of web application security testing. By integrating static analysis, dynamic testing, and a correlation mechanism, the system successfully addresses the limitations associated with standalone approaches.

One of the key observations is the improvement in detection accuracy across multiple vulnerability categories. As shown in Figure 3, the hybrid approach consistently outperforms both static-only and dynamic-only techniques. This finding aligns with previous studies that highlight the complementary strengths of static and dynamic analysis, where static methods provide broader coverage while dynamic analysis confirms exploitability [1], [3]. The improved performance is particularly evident in complex vulnerabilities such as authentication bypass and access control issues.

Another important outcome is the significant reduction in false positives. The False Positive Rate Comparison (Figure 3) indicates a notable decrease in false alarms when using the hybrid model. This result is consistent with earlier research, which emphasizes the importance of runtime validation in filtering non-exploitable vulnerabilities [1], [9]. The correlation engine plays a critical role by prioritizing dynamically confirmed vulnerabilities, thereby reducing redundant findings and improving overall reliability.

The results presented in Table 1 further demonstrate the effectiveness of the proposed approach in distinguishing between potential and exploitable vulnerabilities. Static analysis identifies a large number of candidate vulnerabilities; however, only a subset is validated during dynamic testing. This observation is consistent with existing literature, where static techniques tend to overestimate vulnerability presence due to conservative analysis models [5]. The hybrid framework addresses this limitation by incorporating payload-based validation, ensuring that only actionable vulnerabilities are reported.

In addition, the proposed system shows improved flexibility compared to traditional tools such as OWASP ZAP and Burp Suite, as illustrated in Figure 3. Unlike these tools, which primarily rely on dynamic testing, the proposed framework integrates static analysis and supports local source code scanning. This capability aligns with recent trends emphasizing the integration of security testing within the software development lifecycle (DevSecOps) [2], [15]. Furthermore, the incorporation of AI-assisted pattern detection reflects emerging advancements in intelligent vulnerability detection [6], [10].

Despite these advantages, certain limitations remain. The effectiveness of dynamic analysis depends on the quality of payloads and endpoint accessibility, as noted in prior studies [3], [12]. Additionally, pattern-based static analysis may not capture complex logical vulnerabilities. Future enhancements can incorporate machine learning techniques to improve detection accuracy and adaptability [7], [10].

3.7. Conclusion: This study presents a hybrid vulnerability detection framework that integrates static analysis, dynamic testing, and a correlation engine to improve the accuracy and reliability of web application security assessment. The proposed approach effectively addresses the limitations of standalone techniques by combining the broad coverage of static analysis with the precision of dynamic validation. Experimental results demonstrate that the hybrid model significantly enhances detection accuracy across multiple vulnerability categories, including SQL Injection, Cross-Site Scripting, authentication bypass, and access control issues, while also reducing false positive rates.

The framework supports both white-box and black-box testing environments, making it adaptable to diverse real-world scenarios. The incorporation of a correlation mechanism ensures that vulnerabilities are not only detected but also validated, thereby minimizing manual verification efforts and improving the overall efficiency of the assessment process. Despite these advantages, certain limitations remain, particularly in terms of dependency on predefined payloads and pattern-based detection mechanisms. Future work will focus on integrating machine learning and AI-driven techniques for adaptive payload generation and improved detection of complex vulnerabilities. Additionally, extending the framework for seamless integration into DevSecOps pipelines can further enhance its practical applicability. Overall, the proposed hybrid approach provides a scalable and effective solution for multi-vulnerability detection in modern web applications.

4. References

- [1.] Nunes, J. Fonseca, and M. Vieira, "Blending static and dynamic analysis for web application vulnerability detection," *IEEE Access*, vol. 13, pp. 3139–3151, 2025.
- [2.] R. Karthick, S. Balaji, M. Praveen Kumar, and V. Rajesh, "Streamlining vulnerability detection with hybrid static–dynamic analysis," *Preprints*, 2026. [Online]. Available: <https://www.preprints.org/manuscript/202601.0878>

- [3.] C. Liang, Y. Zhang, X. Wang, and H. Liu, "Survey of source code vulnerability analysis based on hybrid methods," *Computers & Security*, vol. 138, 2025.
- [4.] R. Correa, F. A. González, and J. M. Luna, "Hybrid security assessment methodology for web applications," *Journal of Systems and Software*, vol. 167, 2020.
- [5.] M. Shaon, A. Rahman, S. Ahmed, and M. Karim, "Modern approaches to software vulnerability detection," *Electronics*, vol. 14, no. 22, p. 4449, 2025.
- [6.] J. Wang, H. Li, Y. Chen, and X. Zhou, "A survey on large language models in program analysis," *Transactions on Artificial Intelligence*, 2025.
- [7.] M. Uddin, S. Islam, M. Rahman, and N. Hossain, "Deep learning aided software vulnerability detection: A survey," *arXiv preprint, arXiv:2503.04002*, 2025.
- [8.] M. H. Abbas, K. Iqbal, S. Rafique, and B. Hamid, "Machine learning approaches in software vulnerability detection: A systematic review and analysis of contemporary methods," 2025. [Online]. Available: <https://www.researchgate.net/publication/390053784>
- [9.] F. Mateo-Tudela, J. Bernal, and P. García, "On combining static, dynamic and interactive analysis," *Applied Sciences*, vol. 10, no. 24, p. 9119, 2020.
- [10.] Y. Chen, L. Zhao, X. Wu, and Q. Li, "SQL injection detection based on deep learning," *Computers, Materials & Continua*, vol. 74, no. 2, 2025.
- [11.] J. Ciavatta, M. Rossi, L. Bianchi, and F. Russo, "Integration of large language models with static analysis for security applications," *Computers & Security*, 2026.
- [12.] C. Cifuentes, M. van Emmerik, and D. Simon, "The role of program analysis in security vulnerability detection," *Computers & Security*, vol. 134, 2023.
- [13.] S. Kumar, R. Sharma, and A. Verma, "Real-time static and dynamic application security in DevSecOps," *International Journal of Future Machine Learning Research (IJFMR)*, 2025. [Online]. Available: <https://www.ijfmr.com>
- [14.] A. Singh, P. Mehta, and R. Gupta, "Design and implementation of an enhanced web application vulnerability scanner," *International Journal of Innovative Research in Computer Science and Technology (IJIRCST)*, 2025. [Online]. Available: <https://www.ijircst.org>
- [15.] V. Patel, N. Shah, and K. Desai, "Web application vulnerability detection using hybrid algorithm," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 2025. [Online]. Available: <https://www.ijraset.com>