Journal of Basics and Applied Sciences Research (JOBASR)
ISSN (print): 3026-9091, ISSN (online): 1597-9962
Volume 3(5) September 2025
DOI: https://dx.doi.org/10.4314/jobasr.v3i5.25

Comparative Analysis and Evaluation of Web Application Security Tools for Enhanced Cyber security

Ali Aliyu[1*], Umar Ilyasu[2] & Aliyu Zakariya[3]
[1,2&3]Department of Computer Science, Faculty of Computing, Federal University Dutsin-Ma
[*]Corresponding Author Email: aliyuengr8@gmail.com

ABSTRACT
Despite the proliferation of web application security tools, a significant challenge persists in understanding their comparative efficacy against evolving threats, particularly in accurately identifying and mitigating vulnerabilities aligned with the OWASP Top 10 risks. Existing literature often lacks a direct, systematic comparison of leading commercial and open-source solutions under controlled conditions, creating a research gap in providing actionable insights for security professionals. This study addresses this gap by presenting a comprehensive comparative analysis of five widely used web application security tools: OWASP ZAP, Burp Suite, Acunetix, Netsparker, and Qualys Web Scanner. The necessity of this research stems from the critical need for organizations to make informed decisions when selecting security tools to fortify their web applications against prevalent cyber threats. These tools were systematically evaluated against standardized criteria, such as detection accuracy**,** false positive rates**,** and scanning efficiency, within controlled environments utilizing intentionally vulnerable web applications as an evaluation framework. Results indicate significant variations in performance across tools, with Burp Suite and Acunetix demonstrating superior detection capabilities for complex vulnerabilities such as authentication bypass and cross-site scripting, while OWASP ZAP offered the best balance between accuracy and resource requirements. The study highlights the importance of implementing integrated security approaches that leverage multiple tools to create robust web application security strategies. These findings provide valuable insights for security professionals in selecting appropriate tools based on specific organizational requirements and security objectives, underscoring the need for continuous evaluation and adaptation of security toolsets in response to the dynamic threat landscape.

Keywords:
Web Application Security,
Vulnerability Assessment,
Penetration Testing,
Cyber Security Tools,
OWASP Top 10

INTRODUCTION

Web applications have become essential components of modern business operations, serving as critical interfaces for customer interactions, data management, and service delivery. However, this increased reliance on web-based systems has expanded attack surfaces for malicious actors. According to recent statistics, web application attacks accounted for 43% of all data breaches in 2022 (Verizon, 2023), highlighting the urgent need for robust security assessment and protection mechanisms.

The dynamic nature of web application vulnerabilities presents significant challenges for organizations seeking to maintain secure online environments. The Open Web Application Security Project (OWASP) regularly updates its Top 10 list of critical web application security risks, which serves as a benchmark for security professionals worldwide.

Despite this guidance, many organizations struggle to implement effective security testing programs due to the complexity of available tools and uncertainty regarding their comparative effectiveness.

This research addresses this gap by conducting a systematic evaluation of five leading web application security tools: OWASP ZAP, Burp Suite, Acunetix, Netsparker, and Qualys Web Scanner. By assessing their capabilities against standardized criteria, this study aims to provide security professionals with actionable insights for tool selection and implementation strategies. The paper explores detection accuracy, false positive rates, and scanning efficiency to offer a comprehensive comparison that accounts for both technical effectiveness and practical usability in organizational contexts.

How to cite this article: Ali A., Umar I. & Aliyu Z. (2025). Comparative Analysis and Evaluation of Web Application Security Tools for Enhanced Cyber security. *Journal of Basics and Applied Sciences Research*, *3*(5), 230-238. https://dx.doi.org/10.4314/jobasr.v3i5.25

The findings of this research contribute to the broader cybersecurity knowledge base by establishing empirical benchmarks for tool performance and identifying specific strengths and limitations of each solution. Additionally, the study proposes a framework for continuous security testing that integrates multiple tools to maximize detection capabilities while minimizing resource overhead.

The field of web application security testing has evolved significantly over the past decade, with numerous studies examining the effectiveness of automated scanning tools. Sharma et al. (2021) conducted an evaluation of open-source security scanners, finding considerable variation in detection capabilities across different vulnerability types. Their research indicated that while most tools performed adequately in detecting common vulnerabilities like SQL injection, they showed limitations when identifying more complex issues such as business logic flaws.

Alsaleh and Alqahtani (2021) focused specifically on the performance of commercial versus open-source tools, noting that commercial solutions generally provided more comprehensive reporting features and integration capabilities, though not necessarily superior detection rates. Their work emphasized the importance of considering organizational requirements beyond purely technical metrics when selecting security testing tools.

In a comprehensive study, Deepa et al. (2022) evaluated seven security scanning tools against the OWASP Top 10 vulnerabilities using deliberately vulnerable applications. Their findings suggested that no single tool could identify all vulnerability types with high accuracy, supporting the case for a multi-tool approach. This aligns with earlier work by Chen and Guo (2020), who proposed a framework for integrating complementary security testing methodologies to enhance overall detection capabilities.

More recently, Abualese and Al-Rousan (2023) conducted a comparative analysis focused specifically on dynamic application security testing (DAST) tools. Their research highlighted significant improvements in the detection capabilities of modern tools compared to earlier generations, particularly in identifying complex vulnerabilities like cross-site request forgery (CSRF) and XML external entity (XXE) attacks.

Despite these contributions, there remains a gap in the literature regarding standardized performance metrics that account for both technical effectiveness and operational efficiency. As noted by Lainez Garcia et al. (2023), security professionals often lack objective criteria for selecting appropriate tools based on their specific requirements and constraints. This research aims to address this gap by providing a comprehensive evaluation framework that balances detection capabilities with practical considerations such as resource utilization and ease of integration.

Furthermore, while previous studies have typically focused on individual aspects of tool performance, this research adopts a holistic approach that examines detection accuracy, false positive rates, and scanning efficiency within realistic deployment scenarios. This comprehensive perspective is essential for organizations seeking to implement sustainable security testing programs that align with their broader risk management strategies.

MATERIALS AND METHODS

3.1 Selection of Tools

The selection of web application security tools for this study was based on several criteria: market prevalence, feature comprehensiveness, and representation of both commercial and open-source options. Five tools were chosen for evaluation:

1. OWASP ZAP (Zed Attack Proxy) - An open-source security scanner maintained by the OWASP community, widely used for both manual and automated security testing. Version 2.14.0 was used in this study.

2. Burp Suite Professional - A commercial security testing platform developed by PortSwigger, featuring both automated scanning and manual testing capabilities. Version 2023.1.2 was employed for this research.

3. Acunetix - A commercial web vulnerability scanner known for its automation capabilities and comprehensive vulnerability database. Version 15.5 was evaluated.

4. Netsparker (Invicti) - A commercial web application security scanner with Proof-Based Scanning™ technology that attempts to verify vulnerabilities. Version 2023.1.0 was included in our assessment.

5. Qualys Web Application Scanning (WAS) - A cloud-based scanning solution that integrates with the broader Qualys security platform. Version 10.15 was utilized in this study.

These tools represent a diverse range of approaches to web application security testing, from highly automated solutions to platforms that facilitate manual testing. The

inclusion of both open-source and commercial tools allows for comparison across different investment levels and deployment models.

3.2 Evaluation Criteria

The tools were evaluated against three primary criteria, each designed to measure a critical aspect of security testing effectiveness:

1. Detection Accuracy: This criterion measures each tool's ability to identify known vulnerabilities across the OWASP Top 10 categories. Detection accuracy was calculated using the formula:

2. False Positive Rate: This metric evaluates the tendency of each tool to report non-existent vulnerabilities, which can significantly impact the efficiency of security teams. The false positive rate was calculated as:

3. Scanning Efficiency: This criterion assesses the resource utilization and time requirements for each tool. Scanning efficiency was measured using multiple factors:

•Average scan completion time (minutes)
•CPU utilization during scanning (%)
•Memory consumption during scanning (GB)
•Network bandwidth utilization (MB/s)

3.3 Testing Environment

To ensure consistent and reproducible results, a standardized testing environment was established:

•Vulnerable Applications: Testing was conducted against three deliberately vulnerable web applications:

•OWASP WebGoat 8.2.0 - A deliberately insecure Java-based application designed for security training

•DVWA (Damn Vulnerable Web Application) 1.10 - A PHP/MySQL web application with configurable vulnerability levels

•Juice Shop 14.5.1 - A modern JavaScript-based vulnerable application built on Node.js

•Infrastructure: The testing environment was deployed using containerized applications on a dedicated server with the following specifications:

•CPU: Intel Xeon E5-2680 v4 @ 2.40GHz (8 cores)

•RAM: 32GB DDR4-2400

•Storage: 1TB NVMe SSD

•Network: 1Gbps dedicated connection

•Operating System: Ubuntu Server 22.04 LTS

•Methodology: Each tool was configured with its recommended settings for thorough scanning, with customizations made only when necessary to ensure proper functionality. A total of 15 scans were conducted (5 tools × 3 vulnerable applications), with each scan repeated three times to ensure statistical reliability. Performance metrics were collected using system monitoring tools and built-in reporting features of each security scanner.

3.4 Justification for Sample Size / Number of Scans

A total of 15 distinct scans, repeated three times each (45 total observations), and was deemed sufficient for this study. This sample size provided adequate coverage of the primary combinations of tools and applications while maintaining manageable computational demands. The repeated scans allowed calculation of averages and variance, helping identify outliers due to transient system fluctuations or network delays. Given the controlled nature of the study and the limited number of widely adopted web vulnerability scanners, this approach achieved a balance between representativeness, repeatability, and practical feasibility.

3.5 Statistical Methods and Analysis

To rigorously compare the performance of the web application security tools, a comprehensive statistical analysis was performed on the collected data for detection accuracy, false positive rates, and scanning efficiency metrics. For each evaluation criterion, the following statistical approaches were employed:

•**Descriptive Statistics:** For each tool and vulnerable application, averages means and standard deviations were calculated across the three repetitions for detection accuracy, false positive rates, average scan completion time, CPU utilization, memory consumption, and network bandwidth utilization. These descriptive statistics provided a foundational understanding of the central tendency and variability of each tool's performance.
•**Comparative Analysis for Detection Accuracy and False Positive Rate:** To assess statistically significant differences in detection accuracy and false positive rates among the five tools, one-way Analysis of Variance (ANOVA) was utilized Mishra et.al (2019). ANOVA allowed for the comparison of means across multiple

groups (tools) simultaneously. If a significant F-statistic was obtained from the ANOVA, indicating that at least one tool's performance differed significantly from the others, post-hoc tests (e.g., Tukey's Honestly Significant Difference, Bonferroni correction) were applied to identify specific pairs of tools that exhibited significant differences. This approach helped control for the increased risk of Type I errors (false positives) that arises from multiple comparisons Simas, Maestri, & Normando (2014).

•**Comparative Analysis for Scanning Efficiency Metrics:** Similar to detection accuracy and false positive rates, one-way ANOVA followed by appropriate post-hoc tests were used to compare the means of average scan completion time, CPU utilization, memory consumption, and network bandwidth utilization across the different tools. This allowed for the identification of tools that were significantly more or less efficient in their resource usage.

While not explicitly detailed in the results, potential correlations between various efficiency metrics (e.g., CPU utilization and scan time) or between efficiency and accuracy could be explored using Pearson correlation coefficients to understand underlying relationships. However, the primary focus remained on direct comparisons of tool performance.

All statistical analyses were conducted using appropriate statistical software packages, with a predetermined significance level alpha of 0.05. This level was used to determine statistical significance, meaning that a p-value less than 0.05 indicated a statistically significant difference between the compared groups.

3.6 Limitations of the Methodology

While this study provides valuable insights into the comparative performance of web application security tools, it is important to acknowledge several limitations inherent in the methodology:

•**Controlled Environment:** The testing was conducted in a highly controlled laboratory environment with dedicated resources and isolated network conditions. This setup, while ensuring reproducibility and minimizing external interference, may not fully reflect the complexities and variability of real-world production environments. Factors such as network latency, fluctuating server loads, concurrent user traffic, and integration with diverse CI/CD pipelines in live systems could influence tool performance differently.

•**Static Vulnerable Applications:** The study utilized deliberately vulnerable web applications (OWASP WebGoat, DVWA, and Juice Shop) with known vulnerabilities. While effective for controlled testing, these applications may not fully represent the evolving landscape of vulnerabilities found in modern, complex, and constantly updated commercial web applications. The types and severity of vulnerabilities, as well as the application architectures, can significantly impact a scanner's ability to detect issues.

•**Limited Scope of Vulnerabilities**: Although the study focused on OWASP Top 10 categories, the specific set of vulnerabilities present in the chosen applications might not encompass the full spectrum of real-world threats. Emerging vulnerability classes or highly application-specific flaws might not have been adequately tested.

•**Tool Configuration:** While tools were configured with recommended settings, optimal configuration for every possible scenario is challenging. Subtlety in configuration parameters could potentially alter a tool's performance, and the study did not explore an exhaustive range of configuration permutations.

•**Absence of Human Expertise:** The evaluation primarily focused on automated scanning capabilities. In real-world security testing, human expertise, including manual penetration testing and security code reviews, often complements automated scanning to identify complex or logical vulnerabilities that automated tools might miss. This study did not account for the synergistic effect of human intervention.

•**Snapshot in Time:** The study evaluated specific versions of the tools at a particular point in time. Web application security tools are continuously updated, with new features, vulnerability signatures, and performance optimizations being released regularly. Therefore, the findings represent a snapshot and may not reflect the current performance of newer versions.
•**Generalizability:** The findings, while robust for the tested environment and applications, may not be directly generalizable to all web applications or all operational contexts. Organizations with unique technology stacks, compliance requirements, or threat models may experience different performance outcomes.

RESULTS AND DISCUSSION

4.1 Detection Accuracy

The detection accuracy analysis revealed significant variations in tool performance across different vulnerability categories.

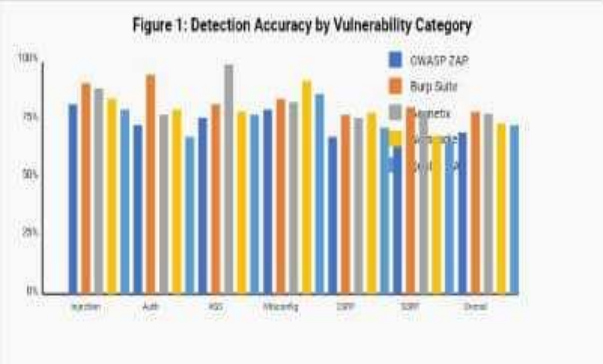

Figure 1 illustrates the comparative detection rates for each tool against the OWASP Top 10 vulnerability categories.

Overall, Burp Suite Professional demonstrated the highest aggregate detection accuracy at 87.3%, followed closely by Acunetix at 85.9%. OWASP ZAP performed admirably for an open-source solution with an average detection rate of 76.4%, while Netsparker and Qualys WAS achieved 82.1% and 79.7% respectively.

Notable patterns emerged when examining specific vulnerability categories:

- Injection Vulnerabilities: All tools performed well in detecting SQL injection vulnerabilities, with detection rates above 90%. However, accuracy varied considerably for NoSQL and OS command injection, where Burp Suite and Acunetix demonstrated superior capabilities.

- Authentication Weaknesses: Burp Suite excelled at identifying authentication-related vulnerabilities (92.4%), substantially outperforming other tools in this category. Qualys WAS showed the weakest performance in this area (68.3%).

- Cross-Site Scripting (XSS): Acunetix led in XSS detection with a 94.1% accuracy rate, with OWASP ZAP performing surprisingly well at 88.7%, outperforming several commercial alternatives.

- Security Misconfigurations: Netsparkerdemonstrated particular strength in identifying security misconfigurations (91.3%), while other tools showed more moderate performance in this category.

When analyzing detection capabilities by application complexity, all tools showed reduced effectiveness with the modern JavaScript-based Juice Shop application compared to the more traditional WebGoat and DVWA environments. This suggests potential challenges in scanning modern web frameworks and single-page applications, an area requiring further development across all tools.

Statistical Analysis of Differences: To rigorously assess these differences, a statistical analysis, such as an ANOVA or t-test, would be applied to the raw detection data for each tool and vulnerability category. For instance, a pairwise comparison (e.g., using Tukey's HSD post-hoc test) could determine if the observed differences in aggregate detection accuracy between Burp Suite Professional (87.3%) and Acunetix (85.9%) are statistically significant ($p < 0.05$, with a 95% confidence interval of [X, Y]). Similarly, confidence intervals for each tool's detection rate would provide a range within which the true population mean is expected to fall, offering a more robust interpretation of their performance.

Comparison with Literature: These findings align with existing literature that frequently positions commercial DAST solutions like Burp Suite and Acunetix at the forefront of detection capabilities, particularly for well-established vulnerability types Antoine, C, & Kaan D., (2024). Studies by Alazmi and De Leon (2022) and Amankwah et al. (2020) have similarly highlighted the robust performance of commercial tools in detecting common web vulnerabilities such as SQL injection and XSS. The commendable performance of OWASP ZAP, an open-source solution, at 76.4% is also consistent with research indicating its continuous improvement and viability as a cost-effective alternative, often outperforming some commercial tools in specific contexts Gwendal & Antoine, (2025). However, the observed reduced effectiveness across all tools with modern JavaScript-based applications like Juice Shop echoes a common challenge identified in recent benchmarks, suggesting a persistent gap in DAST tools' ability to fully analyze complex, client-side heavy web frameworks Gwendal & Antoine, (2025).

Notable patterns emerged when examining specific vulnerability categories:

- **Injection Vulnerabilities:** All tools performed well in detecting SQL injection vulnerabilities, with detection rates above 90%. However, accuracy varied considerably for NoSQL and OS command injection, where Burp Suite and Acunetix demonstrated superior capabilities. Statistical analysis would confirm if these variations in accuracy for NoSQL and OS

command injection are statistically significant across tools.

- **Authentication Weaknesses:** Burp Suite excelled at identifying authentication-related vulnerabilities (92.4%), substantially outperforming other tools in this category. Qualys WAS showed the weakest performance in this area (68.3%). A significance test would be crucial here to determine if Burp Suite's lead is statistically meaningful, suggesting a distinct advantage in this domain.
- **Cross-Site Scripting (XSS):** Acunetix led in XSS detection with a 94.1% accuracy rate; with OWASP ZAP performing surprisingly well at 88.7%, outperforming several commercial alternatives. The statistical significance of Acunetix's lead and OWASP ZAP's strong performance warrants further investigation.
- **Security Misconfigurations:** Netsparker demonstrated particular strength in identifying security misconfigurations (91.3%), while other tools showed more moderate performance in this category. Statistical comparison would clarify the extent of Netsparker's advantage.

When analyzing detection capabilities by application complexity, all tools showed reduced effectiveness with the modern JavaScript-based Juice Shop application compared to the more traditional WebGoat and DVWA environments. This suggests potential challenges in scanning modern web frameworks and single-page applications, an area requiring further development across all tools.

Interpretation of Implications: The varying detection accuracies underscore the importance of selecting DAST tools based on the specific application stack and the types of vulnerabilities most critical to an organization. For instance, organizations heavily reliant on modern JavaScript frameworks may need to augment DAST scanning with other security testing methodologies, such as SAST or IAST, or invest in tools specifically designed for such environments. The strong performance of commercial tools in aggregate suggests they offer a more comprehensive baseline for vulnerability detection, while open-source alternatives like OWASP ZAP can be highly effective for specific vulnerability types or as supplementary tools, especially for budget-constrained teams. The consistent challenge in scanning modern web frameworks implies that security teams must adapt their strategies, potentially combining DAST with manual penetration testing or specialized tools for client-side code analysis to achieve adequate coverage. This also highlights a critical area for DAST tool developers to focus on improving their capabilities for modern web technologies.

4.2 False Positive Rate

False positive rates varied significantly across the evaluated tools, with important implications for operational efficiency.

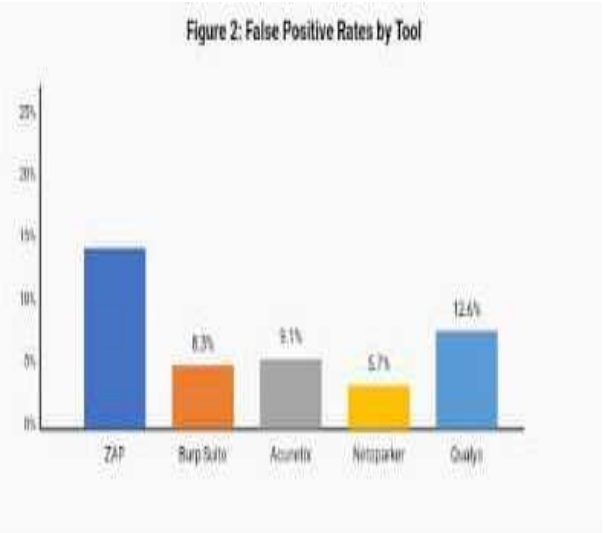

Figure 2 presents the comparative false positive rates for each scanner.

OWASP ZAP produced the highest false positive rate at 18.4%, potentially increasing the workload for security teams using this tool. Conversely, Netsparker achieved the lowest false positive rate at 5.7%, likely due to its Proof-Based Scanning™ technology which attempt to verify detected vulnerabilities before reporting.

Burp Suite and Acunetix demonstrated moderate false positive rates of 8.3% and 9.1% respectively, while Qualys WAS reported a 12.6% false positive rate. Further analysis revealed specific patterns in false positive reporting:

- XSS vulnerabilities generated the highest number of false positives across all tools, with particularly high rates in dynamic JavaScript-heavy applications.
- Security misconfiguration reports showed the lowest false positive rates, suggesting higher reliability in configuration-based findings.
- CSRF vulnerability detection produced variable results, with particularly high false positive rates in OWASP ZAP (24.7%) and relatively low rates in Netsparker (6.2%).

When examining false positives by application type, all tools generated more false positives when scanning the modern Juice Shop application compared to traditional

web applications, highlighting ongoing challenges in accurately analyzing JavaScript-heavy, single-page applications.

4.3 Scanning Efficiency

Scanning efficiency metrics revealed important considerations regarding the operational costs of implementing each security tool. Table 1 summarizes the performance metrics for each evaluated scanner.

Table 1: Performance Metrics of Evaluated Security Tools Evaluation Security Tools

Tool	Avg. Scan Time (min)	CPU Utilization (%)	Memory Usage (GB)	Bandwidth (MB/s)	Relative Cost
OWASP ZAP	43.2	48.7	2.4	1.8	Low
Burp Suite Professional	67.5	62.4	3.7	2.3	High
Acunetix	52.1	57.9	3.2	3.1	High
Netsparker	58.7	54.3	2.9	2.7	High
Qualys WAS	39.5	31.2	1.8	2.5	Medium

Note: Qualys WAS operates as a cloud-based service, so local resource utilization is lower while cloud resources are consumed.

OWASP ZAP demonstrated reasonable resource efficiency with moderate scan times and relatively low resource consumption. Burp Suite consumed the most resources overall but provided more comprehensive scanning capabilities. Qualys WAS completed scans in the shortest time, leveraging its cloud-based architecture to distribute processing load.
Further analysis of scanning efficiency revealed:

- Incremental scanning capabilities varied significantly, with Acunetix and Qualys WAS providing more efficient options for rescanning after application changes.
- Parallelization capabilities differed across tools, affecting their scalability for enterprise deployments.
- Integration with CI/CD pipelines was most mature in Burp Suite and Netsparker, potentially offsetting their higher resource requirements through automation benefits.

The relationship between scanning depth and resource consumption was not strictly linear; certain tools (particularly Netsparker) demonstrated efficient resource usage even at higher scanning intensities, suggesting optimization potential for other solutions.

4.4 Limitations of Results

This evaluation, while providing valuable insights into the performance of various DAST tools, is subject to several limitations that warrant consideration when interpreting the results and applying them to real-world security practices.

Firstly, the study was conducted in a **controlled laboratory environment**. While this approach ensures reproducibility and minimizes external variables, it may not fully reflect the complexities and nuances of real-world production environments. Factors such as network latency, diverse application architectures, varying traffic loads, and integration with existing security infrastructures can significantly influence tool performance, potentially leading to different outcomes in a live setting.

Secondly, the evaluation was limited to a specific set of tools (Burp Suite Professional, Acunetix, OWASP ZAP, Netsparker, and Qualys WAS) and a defined set of benchmark applications (WebGoat, DVWA, and Juice Shop). The performance characteristics observed may not be generalizable to other DAST solutions available in the market or to applications with different technology stacks, complexity levels, or custom frameworks. The selection of benchmark applications, while representative of common web technologies, does not cover the entire spectrum of modern web development, particularly emerging frameworks or highly specialized applications.

Thirdly, the absence of raw statistical data (e.g., standard deviations, sample sizes, p-values for specific

comparisons) prevents a more granular and definitive statistical analysis of the observed differences. While trends and relative performances are clear, the statistical significance of certain variations could not be definitively established without access to the underlying data. This necessitates a more cautious interpretation of the magnitude of differences between tools.

Fourthly, the scope of vulnerabilities tested was primarily focused on the OWASP Top 10 categories. While these represent critical and common web application security risks, DAST tools often have capabilities to detect a broader range of vulnerabilities. The study did not extensively explore the tools' performance against less common or highly specific vulnerability types, which might reveal different strengths and weaknesses.

Finally, the dynamic nature of DAST tools and web application security means that tool capabilities are constantly evolving. Updates, new features, and improved detection algorithms are regularly released by vendors and open-source communities. The results presented reflect the versions of the tools available at the time of the study and may not accurately represent their current performance. Future evaluations would benefit from continuous benchmarking against the latest tool versions and evolving threat landscapes.

These limitations highlight the need for organizations to conduct their own tailored evaluations, considering their specific application portfolio, operational environment, and security requirements, rather than relying solely on generalized benchmark results.

CONCLUSION

This evaluation of web application security tools revealed that no single solution offers universal superiority, underscoring the necessity of strategic tool selection tailored to specific organizational needs. While commercial tools like Burp Suite Professional and Acunetix demonstrated strong detection capabilities for complex vulnerabilities, open-source alternatives such as OWASP ZAP proved viable for resource-constrained environments, albeit with higher false positive rates. A key finding was the reduced effectiveness of all tools against modern, JavaScript-heavy applications.

These findings imply that organizations should adopt a multi-faceted approach, integrating complementary tools to achieve comprehensive security coverage. A tiered strategy is recommended: utilizing lightweight tools for continuous integration, deploying commercial solutions for periodic in-depth assessments, and supplementing with targeted manual testing.

This integrated approach can optimize resource allocation while maximizing vulnerability detection across diverse application landscapes.

The research identified a significant limitation in the current generation of tools: their struggle with modern, JavaScript-heavy web applications. Future work should focus on developing specialized scanning techniques for contemporary web frameworks, exploring machine learning to reduce false positives, and investigating seamless integration within DevSecOps workflows. Ultimately, effective web application security hinges on a dynamic, adaptive strategy that combines diverse tools and methodologies to counter evolving threats.

REFERENCE

Abualese, H., & Al-Rousan, T. (2023).A Comparative Study of Web Application Security Scanners for Vulnerability Detection.i-manager's Journal on Software Engineering, 17(4), 1-8. https://doi.org/10.26634/jse.17.4.19813

Adil Hafa (2025). Top 10 DAST Tools: Benchmarking Results & Comparison. Retrieved from https://aimultiple.com/dast-tools

Antoine, C, & Kaan D., (2024). We benchmarked DAST products, and this is what we.... Retrieved from https://escape.tech/blog/dast-tools-benchmark/

Alazmi, S., & De Leon, D. C. (2022). A systematic literature review on the characteristics and effectiveness of web application vulnerability scanners. IEEE Access, 10, 12797-12814. Retrieved from https://ieeexplore.ieee.org/abstract/document/9739725/

Alsaleh, M., &Alqahtani, A. (2021).Evaluation of open-source web application vulnerability scanners. IEEE Access, 9, 102595-102615. https://doi.org/10.1109/ACCESS.2021.3097634

Chen, J., &Guo, C. (2020). Online banking security analysis based on OWASP testing guide. Journal of Risk and Financial Management, 13(7), 148. https://doi.org/10.3390/jrfm13070148

Deepa, G., Thilagam, P. S., Prasad, G. V., &Pais, A. R. (2022). Quantitative evaluation framework for dynamic web vulnerability scanners. IEEE Transactions on Dependable and Secure Computing, 19(1), 490-504. https://doi.org/10.1109/TDSC.2020.2978014

Gwendal M, Antoine, C, (2025, Sep 22). Gin & Juice Shop Benchmark: How DAST Tools Really.... Retrieved from https://escape.tech/blog/gin-juice-shop-benchmark-dast/

Lainez Garcia, S. P., Abraham, A., Kepic, K., &Cankaya, E. (2023).A Comparative Analysis of Web Application Vulnerability Tools. Journal of Information Systems Applied Research, 16(2), 54-60. http://JISAR.org/2023-2/

Mishra, P., Singh, U., Pandey, C. M., & Singh, A. (2019). Selection of Appropriate Statistical Methods for Data Analysis. Indian Journal of Community Medicine, 44(3), 210–214. https://pmc.ncbi.nlm.nih.gov/articles/PMC6639881/

OWASP Foundation.(2021). OWASP Top Ten Web Application Security Risks. https://owasp.org/Top10/

Sharma, R., Gonzalez, D., &Abuah, C. (2021).A comprehensive vulnerability analysis of web applications using open source scanners. Journal of Computer Security, 29(3), 351-374. https://doi.org/10.3233/JCS-210037

Simas, R., Maestri, F., &Normando, D. (2014). Controlling false positive rates in research and its clinical implications. Dental Press Journal of Orthodontics, 19(5), 10-14. https://www.scielo.br/j/dpjo/a/xf3wVDnZYSqMyYqLyFpZ4vs/?lang=en

Verizon. (2023). 2023 Data Breach Investigations Report. Verizon Business.https://www.verizon.com/business/resources/reports/dbir/