Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.gaexcellence.com/jistm

PERFORMANCE EVALUATION OF OPEN-SOURCE WEB VULNERABILITY SCANNERS FOR SQL INJECTION DETECTION IN WEB APPLICATIONS

Raihana Md Saidi[1], Noraqilah Azlan[2], Anwar Farhan Zolkeplay[3]*

[1]Faculty of Computer and Mathematical Sciences, UiTM Cawangan Melaka Kampus Jasin, Melaka, Malaysia
raihana@uitm.edu.my https://orcid.org/0009-0007-8701-2408
[2]Faculty of Computer and Mathematical Sciences, UiTM Cawangan Melaka Kampus Jasin, Melaka, Malaysia
noraqilahazlan@gmail.com https://orcid.org/0009-0008-6115-216X
[3]Faculty of Computer and Mathematical Sciences, UiTM Cawangan Melaka Kampus Jasin, Melaka, Malaysia
anwarfarhan@uitm.edu.my https://orcid.org/0009-0009-2699-844X
*Corresponding Author

Article Info:

Article history:
Received date: 20.01.2026
Revised date: 15.02.2026
Accepted date: 26.03.2026
Published date: 31.03.2026

To cite this document:

Md Saidi, R., Azlan, N., & Zolkeply, A. F. (2026). Performance Evaluation of Open-Source Web Vulnerability Scanners for SQL Injection Detection in Web Applications. *Journal of Information System and Technology Management, 11* (42), 453-466.

DOI: 10.35631/JISTM.1142027

Abstract:

Small and Medium Enterprises (SMEs) are embracing online platforms which results in becoming more exposed to cybersecurity threats, especially SQL Injection (SQLi) attacks. SQLi attack is among the most common web application vulnerabilities listed by OWASP. To address these risks, this research examines the performance of several open-source Web Vulnerability Scanners (WVS) in detecting SQLi threats. Four popular open-source scanners: OWASP ZAP, SQLmap, Skipfish and Wapiti have been tested across five types of web applications. Vulnerable testbeds such as DVWA and OWASP Mutillidae II and the real e-commerce sites have been tested to see how the tools perform in practical situations. All the scanners have been evaluated based on few parameters: detection rate, response time, resource usage on CPU and memory and ease of use. The scanners show variations of results during the experiment. These findings offer useful guidance for SMEs in choosing cost-effective and efficient scanning tools that match their cybersecurity needs.

Keyword:

Black Box Testing, Penetration Testing, SQL Injection, Web Application Security, Web Vulnerability Scanner

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

Introduction

Online platforms have transformed how Small and Medium Enterprises (SMEs) industries conduct their businesses. The platform allows them to reach customers worldwide. Although it embraced on technology-based method, it also brings cybersecurity as main challenges. SMEs face growing risks from cyberattacks because most of the transactions and customer data are done on the net. One of the most threats is SQL injection (SQLi). It takes advantage of weaknesses in how web applications manage database queries. Open Worldwide Application Security Project (OWASP) has listed SQLi among its top ten most dangerous vulnerabilities for over a decade showing how serious this issue is.

SQLi attacks allow the hackers to manipulate the database queries in web applications (Ammagunta et al., 2025). When successful, attackers can gain unauthorized access to confidential information such as customer details, payment information and financial data. In some cases, they can even modify or delete database content. This sometimes will lead to service disruptions. These types of threat grew during the COVID-19 pandemic. During that time, many SMEs moved their operations to online transaction without sufficient security measures. This led to creating easy targets for cybercriminals.

SMEs are usually vulnerable because they often have limited budgets and do not have dedicated cybersecurity teams to support their operations (Curtin et al., 2025). The available commercial security tools can be expensive. The options they have are open-source tools. These tools help to identify vulnerabilities like SQLi and provide reports that guide businesses in fixing the problems. However, with so many tools available, SMEs are also unable to choose which ones are reliable and suitable for their business.

There are four popular open-source web vulnerability scanners that have been identified to be tested in this research. The scanners are OWASP ZAP, SQLmap, Skipfish and Wapiti. This test was done to solve the problem found for this research. DVWA and OWASP Mutillidae II were selected to be as vulnerable testbeds. In order to identify whether each of the scanners can perform on selected websites to prove all the scanners can be used in real-world e-commerce environments (Izabela, 2025). From the experiment, SMEs can have options for selecting an effective and affordable scanner that is suitable for their cybersecurity needs. As part of the contribution to informatics and web engineering fields, it uses data analysis and system testing to evaluate how well different security tools work on web applications. By doing this research, it helps to improve secured web systems during development and deployment.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

Literature Review

Web application vulnerability assessment has been one of the important issues in cybersecurity. This is because SQLi are happening quite often nowadays. A lot of open-source Web Vulnerability Scanners (WVS) have been available recently. On top of that, it is important to understand how well these tools work. Several studies were conducted before to discuss many aspects such as accuracy, detection rate and how these tools perform in real environments. Thus, this section reviewed what has been found in earlier work on web vulnerability scanners, how they compare with each other and what their strengths and weaknesses are valuable for SMEs.

Web Vulnerability Scanners

Many studies have investigated open-source web vulnerability scanners can help in identifying security problems in websites. These scanners are often used to find common threats such as SQLi and Cross-Site Scripting (XSS). Both of this are still listed among the OWASP Top Ten vulnerabilities. One early study by Felício et al. (2023) pointed out that automated black-box testing can help reduce human mistakes and save time compared to manual testing. It also explained how fuzzing can be used to simulate attacks and detect weak points in how web applications handle user input. This research later became the guideline for many improvements in automated web security testing.

Comparative Performance Studies

Few studies have analysed on how different open-source scanners work when it used in the experiments. For example, Shahid et al. (2024) tested a few popular scanners and discovered significant differences in detection accuracy. These differences mainly came from how each scanner was built and how the internal processes worked. The study also found that not all scanners were equally good at finding SQLi or XSS problems. This because the performance depends on how each tool handles responses and checks user input.

A study by Riepponen (2022), tested several scanners such as ZAP, Skipfish, Arachni, IronWASP and Vega using different test environments on DVWA and WebGoat. The results reveal that Skipfish performed quite well in finding vulnerabilities without giving too many false positives result. Whereas ZAP presents its clear reports and good documentation that made it easier for users to understand the results. These studies show that each scanner has its own advantages and weaknesses. Because of this reason, many organizations need to carefully choose tools that match their technical needs and available resources. This also to avoid relying on one solution that may fit for every problem.

Detection Capabilities and Limitations

A review by Alazmi and De Leon (2022) focuses on thirty scanners and compares each of the tools. The comparison is based on frequency of update, interface, vulnerability coverage and documentation availability. The review found that scanners that were not updated often or lacked on user support usually would perform worse, even if the scanners were reliable before. Research by Gandikota et al. (2023) and Tliahun et al. (2025) was tested on other scanners which are W3aF, Vega and Uniscan. This scanner was used on several vulnerable applications. The test found that each of the scanners had its own mismatched detection. The finding requires

better algorithms and more diverse test cases as suggestions in order to improve accuracy. Other than that, Nasereddin et al. (2021) and inspected the SQLi detection method. The study found that although the increment of awareness is high, the vulnerability of web applications still exists. It found that the weak input validation and poor filtering have become the main reasons for this problem. To get useful results, the scanners must be maintained and tested regularly to ensure high performance of the scanning process. In addition, the scanners must be updated to verify the accuracy of the data.

Scanner Architecture and Functional Design

The performance of the scanners also reflected by the development engine. Research by Kumar et al. (2024) shows that improper input handling can caused data breach. This weakness led to data leaks. This to indicate the design and development process of the scanners is important to avoid any risk. A study by Faiz et al. (2025) has identified several categories of SQLi attacks which are tautologies, union queries, piggybacked statements and timing-based inference. These categories can be a benchmark when testing the functionality of the scanners. The aim is to detect different attack pattern. Research by Abdulghaffar et al. (2023) and Lavens et al. (2022) investigated another way to improve accuracy of the scanners. The research found by adding machine learning into vulnerability scanners may contribute significant accuracy. The other way is to embed adaptive learning so that the scanners can identify new attack behaviours that the older tools may not have the features. This point shows brighter improvement for next generation of scanning tools to be more responsive in dealing with cybersecurity threats.

Practical Implications for SMEs

There are studies that focused on how the vulnerability scanners can help the same problem faced by SMEs. Bhawsar and Singh (2025) found that Wapiti's modular structure and simple reporting made it a good option for smaller organizations that might not have dedicated security staff. The non-technical users require easy-to-read reports to understand security issues in a better way and take some quick action if something happened in the organization. A study conducted by Christos et al (2022) highlighted the need for SMEs to implement basic security measures such as administering HTTPS, validating user inputs and monitoring the application layer. The research stated that to make the users can make decision during solving the vulnerabilities issues, the scanners need to show the results clearly. These findings revealed that open-source scanners can a practical choice for SMEs if the if the scanners are properly selected, used regularly and maintained over the time.

As a conclusion, the literature shows that web vulnerability scanners are required in helping organizations strengthen their cybersecurity mechanism. Taking the important criteria, consistent evaluation, proper tool selection and continuous updates to ensure long-term effectiveness.

Methodology

There are several processes involved in evaluating the performance of open-source web vulnerability scanners. The was done in detecting SQLi issues across different types of web applications. The entire process has been divided into five main phases which are information gathering, environment setup, testing and analysis, comparing the results and the last one is the

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

final documentation. This structured plan is to make sure the evaluation is consistent and reliable.

Environment Setup

The environmental setup phase focused on preparing all the tools and systems. VMware was used to create isolated virtual machines to ensuring that all scans ran in a controlled environment without affecting the real systems. Kali Linux was chosen as the testing platform because it already includes many penetration testing tools and supports the scanners well.

Four open-source scanners were used in this study that are OWASP ZAP, SQLmap, Skipfish and Wapiti. Each was installed inside the virtual environment. Five web applications were selected for testing namely DVWA, OWASP Mutillidae II, Acunetix Art, Altoro Mutual and the Syabab Bookstore website. These applications were chosen because they provided a mix of intentionally vulnerable platforms and realistic e-commerce systems.

All scans were performed in a non-production environment to keep testing safe. The scanners were also updated to their latest versions before the tests started to make sure the results were accurate and fair.

Web Vulnerability Scanners and Parameters

Each scanner was set up according to its official user guide to keep the testing consistent. There are six main parameters which are detection rate, response time, ease of use, CPU usage, Resident Set Size (RSS) and memory utilization has been evaluated. These parameters are the most suitable from the perspective of SMEs to give whole view of scanner performance. This is because of SMEs need the tools that are accurate but not too complex or expensive.

Testing and Analysis

Each of the scanner was run on all five web applications during testing phase. Each of scan phase was monitored closely to record what vulnerabilities were detected. At the same time, amount of system resources used during the process also recorded. If there are SQLi vulnerabilities were found during this step, the detection rate is counted. Response time was measured in minutes by the required duration for each scan. The graphs and tables were generated by the collected data to show the comparisons performance across the different parameters. This straightforward step keeps the results consistent and made it easier to repeat the tests if needed.

Data Analysis

Each scanner's strengths and weaknesses were identified based on the collected data. There are several ways to present the data which are detection rate was shown as a percentage, response time as the average scan duration, and resource usage as the average CPU, RSS and memory consumed during scans. Other than that, some of the data analysed qualitatively for ease of use. This done by observing how users interacted with the tools and reviewing how clear and useful the generated reports were. By combining the qualitative and quantitative results, it provided a clearer view of each scanner's performance and practicality especially for SMEs with limited technical resources.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 453-466

System Architecture Design

A system architecture diagram was created to illustrate the testing setup. Figure 1 depicted the components interacted to each other in this experiment. The data flowed and the connected part during the scanning process also presented in the diagram. This architecture diagram represents both the hardware and virtual environment used in the testing phase. This to help the process clearer and easier. It also can be use as references to be replicated for future studies or same experiments.

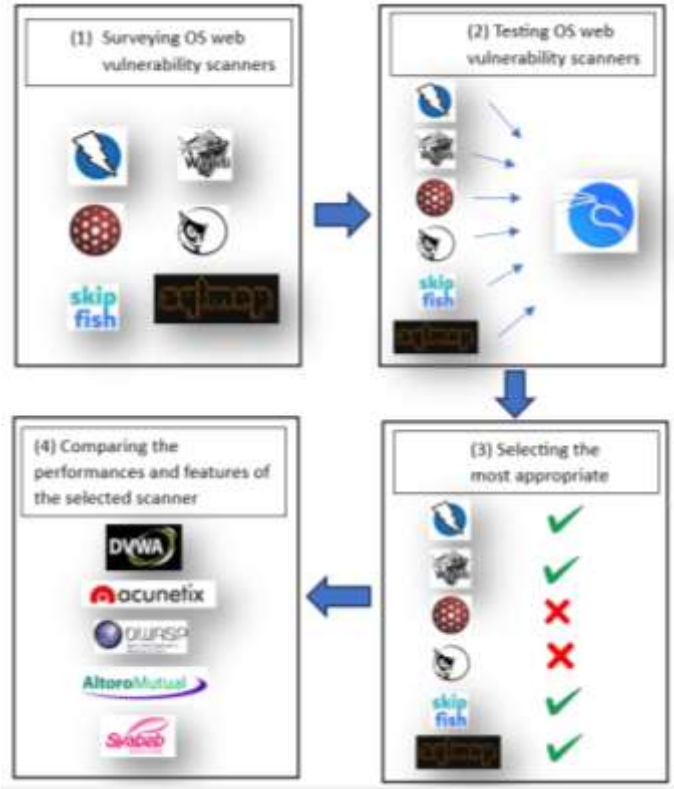

Figure 1: Project Architecture

Step 1: Surveying OS Web Vulnerability Scanners

The first step involved identifying a range of open-source web vulnerability scanners. Several tools were identified including Wapiti, Skipfish, SQLmap, OWASP ZAP and Vega. This is to make sure the research focused only on scanners that were widely recognized. By narrowing down the list, the study avoided wasting time on outdated or unsupported tools.

Step 2: Testing OS Web Vulnerability Scanners

Next, the shortlisted scanners were installed and tested in a controlled environment to observe their initial performance. As illustrated in Figure 1, each scanner was pointed toward a set to generate preliminary data. This initial testing is to make sure the scanners can effectively detect the vulnerabilities and can produce any notification on performance issues. Overall, this step is required to give a big picture of how each of the scanners works. The output for this stage can provide a general overview of the testing phase before continuing to the detailed comparison of each scanner.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

Step 3: Selecting The Most Appropriate

One of the important steps is to choose the most appropriate scanners for the evaluation phased. There are four tools has been selected on the initial testing results are OWASP ZAP, SQLmap, Skipfish and Wapiti. Some of the selection criteria are consistent detection capabilities, updated by developers and function well in different testing environment. The chosen tools are marked with check symbols, while others were excluded due to problems such as outdated features, weak detection performance or limited user support as depicted in Figure 1. The most reliable and relevant tools were decided for the final evaluation during this selection process.

Step 4: Comparing Performances And Features Of The Selected Scanners

As the last step for this experiment requires five identified web application to be running on the selected scanners. The web applications are DVWA, Acunetix Art, OWASP Mutillidae II, Altoro Mutual and the Syabab Bookstore website. There are six parameters used to compare each of the scanner's parameter as decided earlier. This step present better understanding of how each scanner performed under similar conditions. The details comparative analysis discussed later in the result section indicating the strengths and weaknesses of each scanner.

Results and Discussions

Evaluating the scanners' performance requires six parameters which are detection rate, response time, ease of use, CPU usage, RSS usage and memory utilization. To keep the evaluation fair, every scanner was tested on five different web applications that included both intentionally vulnerable sites which are DVWA and OWASP Mutillidae II to the real e-commerce platforms. The results from these tests were then compared to identifying how well each tool performed and where its limitations are present.

Figure 2 shows the detection rate results. OWASP Mutillidae II make the SQLmap and Skipfish achieved the highest detection rates with 8 percent. SQLmap consistently delivered strong detection accuracy, proving to be the most dependable tool for finding SQLi vulnerabilities across all the test environments.

Skipfish also performed well in some cases. Although its results were not as steady as SQLmap. Meanwhile, OWASP ZAP produced mixed outcomes as it detected several vulnerabilities effectively in certain applications, but its accuracy decreased in others. Wapiti recorded the weakest detection rate overall, showing that it might not be as efficient as the other tools in identifying SQLi threats. These findings suggest that although SQLmap performs best in terms of detection accuracy, SMEs should still consider other practical aspects such as how easy a tool is to use and how many system resources it requires before deciding which scanner suits the best needs.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

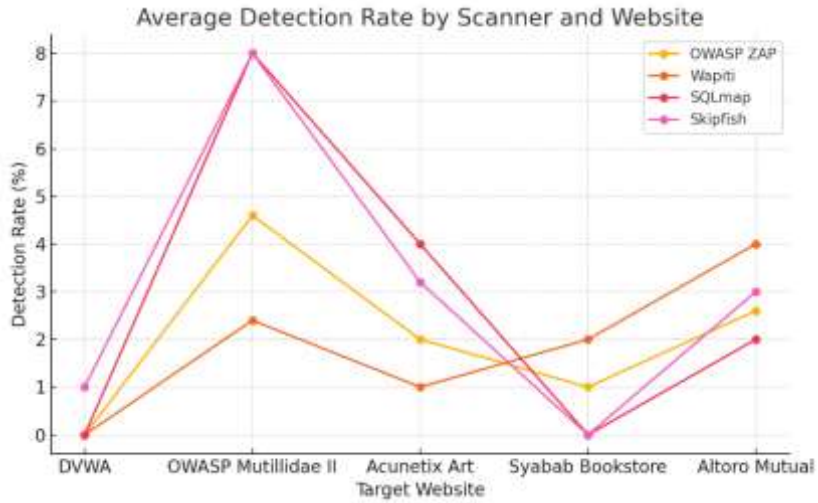

Figure 2: Average Detection Rate for Each Scanner

The result of response time for each scanner to complete its vulnerability scan depicted in Figure 3. Among all tools tested, SQLmap recorded the fastest time in 2.36 minutes per scan. This was proven that SQLmap is the quickest scanner among the others. Skipfish ranked as second fastest scan. Wapiti needed slightly longer, with an average of around 5 minutes. As OWASP ZAP had the slowest response time, taking about 16.51 minutes. However, OWASP ZAP's longer scan duration is not necessarily a disadvantage. The detailed scanning process affects the slower time for the scanner to finish the process. This is because of the scanner explores applications deeper and generates more comprehensive reports as found earlier. It provides SMEs with a broader and more informative view of their system vulnerabilities because of this approach takes extra time during completion of the process.

It is important for SMEs to rely on quick feedback to maintain smooth operations based on the results. Vulnerabilities can be detected and fixed sooner if it has shorter response time that led to slower the chance for attacker to exploit the transaction. Frequent testing on dynamic web applications with higher speed shows by SQLmap and Skipfish. This requires SMEs to find a balance between speed and depth when choosing a scanner. This indicate that SQLmap and Skipfish are suitable for quick checks and regular assessments. The scanner can be more helpful when detailed reporting and full coverage are require even though OWASP ZAP taking longer time.

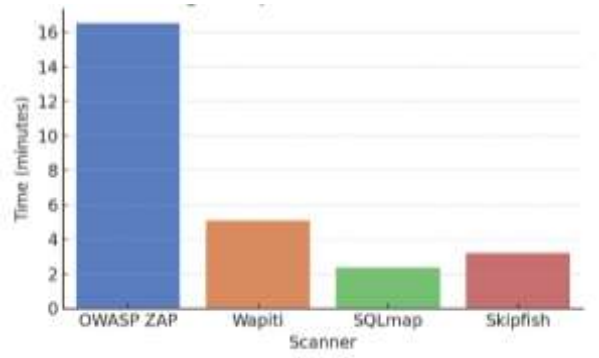

Figure 3: Average Response Time of Each Scanner

Figure 4 shows the ease-of-use score for each scanner, rated on a scale from 1 to 5. The evaluation focused on how simple the interface was, whether the documentation was understandable and the quality of reports was produced. Among the four scanners, OWASP

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

ZAP received the highest score of 5, making it the most user-friendly option. Its graphical interface, easy navigation and detailed reporting features make it suitable for SMEs or users who are not highly technical. SQLmap followed with a score of 4. Even though it mainly runs through the command line, the tool is supported by clear documentation, strong community help and decent reporting functions. These features make it easier to use for users familiar with technical setups. Wapiti and Skipfish both scored 3, showing moderate usability. These scanners work well but their interfaces are less natural, and their reporting features are limited. This can make navigation more challenging for beginners.

This finding indicates the importance of usability when choosing security tools, especially for SMEs. A higher score means less time spent on learning and interpreting results. While OWASP ZAP clearly is the best one, the other scanners such as Wapiti and Skipfish may still perform effectively. However, it may require additional training to be used efficiently.

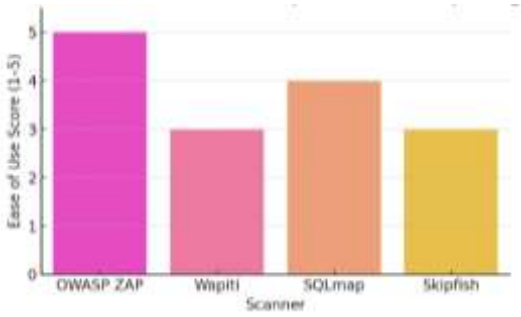

Figure 4: Ease Of Use Score Based on User Experience and Reporting Features

Figure 5 presents the average CPU usage recorded for each open-source scanner during SQLi testing. The graph indicates clear differences on processing power each of the tool consumers. OWASP ZAP had the highest CPU usage at around 45%, while Wapiti and Skipfish showed moderate usage of 23% and 20% respectively. The least CPU usage and most efficient system resources done by SQLmap is just under 8%. While the higher CPU usage in OWASP ZAP is expected because it performs deep scans and runs multiple tests simultaneously. OWASP ZAP also has an interactive interface and live reporting that also requires additional processing power. This complete scanning function makes extra usage for the scanners. Wapiti and Skipfish maintain between performance and efficiency. The scanners provide reasonable scan coverage without overloading the system. On the other hand, SQLmap consumes minimal CPU since it runs on a simple command-line interface and focuses directly on SQLi testing. This makes it useful when examining machines with limited processing capacity.

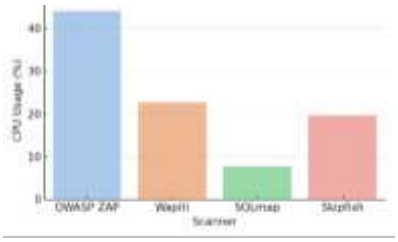

Figure 5: Average CPU Usage During Scanning

Average Resident Set Size (RSS) memory usage was measured by calculating the amount of RAM was used during scanning process as depicted in Figure 6. The most memory was consumed by OWASP ZAP with the size over 400 MB. In contrast, Wapiti used the least memory with the value of below 25 MB. And again, the SQLmap and Skipfish was maintained

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

on low memory usage under 50 MB. OWASP ZAP has strong reason on higher memory usage where it comes from the detailed scanning features and multiple built-in features on the scanner. It requires a larger memory footprint to process and store data because of monitoring all web traffic closely. Other than that, OWASP ZAP also has user-friendly interface and multiple functionalities. simpler command-line designs without heavy interfaces contribute to the less memory usage owned by Wapiti, SQLmap and Skipfish scanners. This makes the scanners more lightweight and efficient. SQLmap performs well due to its focus on SQLi testing. The most practical scanners are the tools used for systems with limited resources and low memory usage. Although OWASP ZAP proven as powerful tools, it requires stronger hardware to run the scanning

.

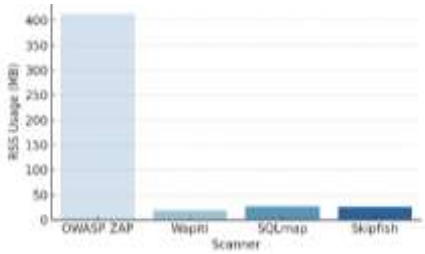

Figure 6: Average RSS Memory Usage During Scanning

Figure 7 compares the total memory usage of the four scanners. This parameter also shows significant differences with other scanners. OWASP ZAP consumed the most memory which is to 800 MB. This is because it is a full-featured proxy-based scanner that handles a lot of web traffic and generates detailed reports. The demand also come from extensive graphical interface and various plugins. However, both Skipfish and Wapiti use lighter memory for 50 MB respectively. SQLmap also remained efficient by consuming memory below 100 MB. Without require heavy graphical interfaces and running command-line tools, the scanners are suitable to be used in lower spec environments as for SMEs.

The overall comparison shows that there is always a trade-off between how powerful the scanner is and how many system resources are required. A complete and detailed analysis produced by OWASP ZAP but the scanner requires more CPU and memory to run properly. In contrast, the simpler and faster tools for SQLmap, Wapiti and Skipfish make it the practical choices for smaller setups or regular automated scans. System capability and the level of detailed security test are among the factors to be considered when making the right choice.

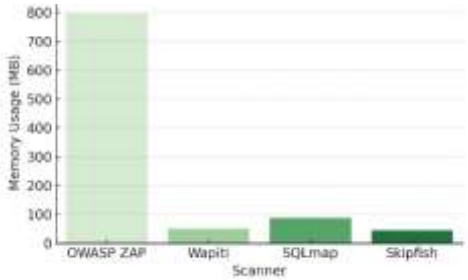

Figure 7: Total Memory Usage of Each Scanner

Future Work

It would be good to test more web applications that differ in complexity in the future. This to to get a clearer idea of how scanners perform in different situations. Another factor to consider

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466
is to include commercial scanners in future tests and compare the tools with open-source scanners. The aim is to see whether the extra cost gives better results or not. It also proves whether open-source tools are better than expensive ones. OWASP Benchmark or WAVSEP could also improve the reliability of future evaluations because it allows researchers to measure accuracy and consistency. Some of the limitations from this study would also improve future research. Examples of these limitations are outdated vulnerability databases, small testing budgets, and issues with system stability.

SQLmap and Skipfish showed strong detection accuracy, while OWASP ZAP was the easiest to use. Though, OWASP ZAP also used the most CPU and memory during the testing. The results show Wapiti was lightweight but less accurate. These results indicate that there is no single best tool for overall environment. And the choice of the tools depends on what an organization values most in terms of speed, accuracy, ease of use or system efficiency.

This study shows how structured data collection and performance analysis can give useful insights into how cybersecurity tools work to contribute to the informatics field. While for web engineering side, this research helps deepen understanding of how scanners were designed and the resources used that affect performance in real-world applications. The study connects theoretical knowledge with hands-on practice for vulnerabilities and real-world application. Future developers and researchers may design more secure, efficient and reliable web systems based on the proven experiment.

Conclusion

This study set out to examine the relationship between digital marketing strategies and the performance of small and medium enterprises (SMEs) in Malaysia. The findings reveal that the effective use of social media, search engine optimization, and customer relationship management tools significantly enhances business visibility, customer engagement, and ultimately financial performance. Importantly, the results highlight that SMEs with proactive adoption of digital tools tend to outperform those that rely solely on traditional methods. The implications of this study are twofold. From a theoretical perspective, it contributes to the growing body of literature on SME digitalization by demonstrating the measurable impact of online strategies on organizational outcomes. From a practical standpoint, it provides entrepreneurs and policymakers with valuable insights to design training programs, incentives, and policies that encourage digital transformation among SMEs.

Acknowledgements: The authors would like to express their gratitude to the Faculty of Computer and Mathematical Sciences (FSKM) for providing the necessary facilities and support for this research. We also acknowledge the International Conference on Mathematical Sciences and Statistics (MIC3ST 2025) for the platform to present and refine the preliminary findings of this work.

Funding Statement: No funding.

Conflict of Interest Statement: The authors declare that there is no conflict of interest regarding the publication of this paper. All authors have contributed to this work and approved the final version of the manuscript for submission to the International Journal of Information System and Technology Management (JISTM)

Ethics Statement: This study did not involve any human participants, animals, or sensitive data requiring ethical approval. The authors confirm that the research was conducted in accordance with accepted academic integrity and ethical publishing standards.

Author Contribution Statement: All authors contributed significantly to the development of this manuscript. Anwar Farhan Zolkeplay and Noraqilah Azlan were responsible for the conceptualization, methodology, data curation and overall supervision of the study. They also handled data collection, analysis, and interpretation of results. Anwar Farhan Zolkeplay and Raihana Md Saidi contributed to the literature review, drafting, and critical revision of the manuscript. All authors read and approved the final version of the manuscript prior to submission.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

References

Abdulghaffar, K., Elmrabit, N., & Yousefi, M. (2023). Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners. Computers, 12(11), 235. https://doi.org/10.3390/computers12110235.

Alazmi, S., & De Leon, D. C. (2022). A Systematic Literature Review on the Characteristics and Effectiveness of Web Application Vulnerability Scanners. IEEE Access, 10, 33200–33219. https://doi.org/10.1109/access.2022.3161522.

Ammagunta, S., Akula, A., Pottipati, C. S., Avula, L. M., & Reddy, Y. (2025). Defending against SQL injection: Practical application with open-source tools for improved cyber security. AIP Conference Proceedings, 3281, Article 020036. https://doi.org/10.1063/5.0260769.

Bhawsar, P., & Singh, S. (2025). A Comprehensive Framework for Web Application Penetration Testing: Leveraging Automated and Manual Methods for Enhanced Security. 2025 13th International Conference on Intelligent Systems and Embedded Design (ISED), 125–130. https://doi.org/10.1109/ised67359.2025.11405427.

Christos Tselios, Politis, I., & Xenakis, C. (2022). Improving Network, Data and Application Security for SMEs. Proceedings of the 17th International Conference on Availability, Reliability and Security. https://doi.org/10.1145/3538969.3544426.

Curtin, M., Sheehan, B., Gruben, M., O'Carroll, G., & Murray, H. (2025). Enhancing Cybersecurity Awareness in Small and Medium Enterprises Through a User-Friendly Risk Assessment Tool. 209–226. https://doi.org/10.1109/eurosp63326.2025.00021.

Duarte Felício, José Simão, & Nuno Datia. (2023). RapiTest: Continuous Black-Box Testing of RESTful Web APIs. Procedia Computer Science, 219, 537–545. https://doi.org/10.1016/j.procs.2023.01.322.

Faiz, A., Muhammad, Faiz, B., Hassan, Hassan, & Fatima, A. (2025). Assessment of SQL Injection Attacks and Defense Mechanisms in Stored Procedures. Journal for Current Sign, 3(3), 1471–1484. https://currentsignreview.com/index.php/JCS/article/view/338.

Gandikota, P. S. S. K., Valluri, D., Mundru, S. B., Yanala, G. K., & Sushaini, S. (2023). Web Application Security through Comprehensive Vulnerability Assessment. Procedia Computer Science, 230, 168–182. https://doi.org/10.1016/j.procs.2023.12.072.

Izabela Kaźmierak. (2025). Comparison of the effectiveness of tools for testing the security of web applications. Journal of Computer Sciences Institute, 34, 36–43. https://doi.org/10.35784/jcsi.6613.

Kumar, A., Dutta, S., & Prashant Pranav. (2024). Analysis of SQL injection attacks in the cloud and in WEB applications. Security and Privacy. https://doi.org/10.1002/spy2.370.

Lavens, E., Philippaerts, P., & Joosen, W. (2022). A Quantitative Assessment of the Detection Performance of Web Vulnerability Scanners. Proceedings of the 17th International Conference on Availability, Reliability and Security. https://doi.org/10.1145/3538969.3544416.

Nasereddin, M., ALKhamaiseh, A., Qasaimeh, M., & Al-Qassas, R. (2021). A systematic review of detection and prevention techniques of SQL injection attacks. Information Security Journal: A Global Perspective, 32(4), 1–14. https://doi.org/10.1080/19393555.2021.1995537.

Riepponen, M. (2024). Selection of open-source web vulnerability scanner as testing tool in continuous software development. Jyx.jyu.fi. https://jyx.jyu.fi/jyx/Record/jyx_123456789_94465.

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 453-466

Future Directions. Applied Sciences, 12(8), 4077. https://doi.org/10.3390/app12084077.

Tliahun E. B., Shalu G., Eshetu B., Perform Scanning and Comparison of Open Source Web Application Testing Tools: Using Strategic Holistic Approach. (2025). Journal of Posthumanism, 5(2), 1377–1402. https://www.ceeol.com/search/article-detail?id=1353363.