DOI: 10.2478/seeur-2025-0021

STATISTICAL ANALYSIS OF UNIQUE WEB APPLICATION VULNERABILITIES: A QUANTITATIVE ASSESSMENT OF SCANNING TOOL EFFICIENCY

Gani Zogaj, PhD (c)
Faculty of Contemporary Sciences and Technologies,
South East European University,
Tetovo, North Macedonia
gz30455@seeu.edu.mk

Florie Ismaili
Faculty of Contemporary Sciences and Technologies,
South East European University,
Tetovo, North Macedonia
f.ismaili@seeu.edu.mk

Ermira Idrizi
Faculty of Contemporary Sciences and Technologies,
South East European University,
Tetovo, North Macedonia
e.idrizi@seeu.edu.mk

Artan Luma
Faculty of Contemporary Sciences and Technologies,
South East European University,
Tetovo, North Macedonia
a.luma@seeu.edu.mk

ABSTRACT

Web application security is a critical aspect of modern cybersecurity, necessitating efficient and reliable vulnerability detection mechanisms. This study presents a quantitative analysis of unique web application vulnerabilities detected by four automated scanning tools: Nessus, Acunetix, OWASP ZAP, and BeSECURE. We scanned 67 web applications and sorted

the vulnerabilities we found into four categories: Critical, High, Medium, and Low. This study evaluates each tool's effectiveness and reliability using mean and standard deviation, providing key insights into their performance consistency. Using straightforward statistical methods, we aim to determine which scanning tool performs best in finding vulnerabilities while maintaining consistent results across different web applications. Additionally, the analysis offers comparative insights into the performance variations among these tools, highlighting their strengths and limitations. The study paper contributes to strategic decision-making in cybersecurity, enabling organizations to select the most effective tools for vulnerability assessment. The findings demonstrate that OWASP ZAP exhibits superior detection capabilities and consistency across various severity levels, while integrating tools like Nessus, BeSECURE, and Acunetix enhances vulnerability detection, with Nessus excelling in identifying critical and high-severity vulnerabilities.

Key words: Vulnerability Scanning, Nessus, Acunetix, OWASP ZAP, BeSECURE, Web Application, Vulnerability Detection Tools, Comparative Analysis and Cybersecurity.

INTRODUCTION

In today's interconnected world, web applications have emerged as vital components in facilitating business operations, social interactions, and the delivery of public services. However, as businesses rely more and more on these technologies, they unintentionally expose themselves to a wider range of cybersecurity risks. Web application vulnerabilities are a primary target for malevolent actors among these threats, highlighting the urgent need for strong and efficient vulnerability detection and mitigation techniques. Thus, the need to find and fix vulnerabilities before they can be exploited defines the field of web application security.

This study seeks to critically evaluate the effectiveness of four widely used vulnerability scanning tools Nessus, Acunetix, OWASP ZAP, and BeSECURE by conducting a comparative analysis of their performance in detecting unique vulnerabilities across a sample of 67 web applications. Through this rigorous evaluation, we explore how each tool identifies vulnerabilities, the accuracy of their results, and the extent to which they overlap in their findings. By focusing on the unique vulnerabilities uncovered by each tool, this study provides valuable insights into the strengths and limitations of these widely used scanners.

This article goes beyond raw detection capabilities, contextualizing these tools' effectiveness in real-world security. A comparative analysis of vulnerability identification, accuracy, and reporting aids security experts in selecting the best tools to protect web applications. The findings contribute to the evolving web security landscape, fostering data-driven decisions for stronger digital protection.

Existing research highlights the significance of web application vulnerabilities and the necessity of robust detection mechanisms. For example, Ali et al. [1] conducted a comparative analysis of protection techniques against Structured Query Language Injection Attacks (SQLIAs), identifying strengths and weaknesses of various approaches in mitigating specific attack vectors. This study showed that creating all-encompassing security solutions requires an awareness of the similarities and differences in protective measures. In a similar vein, Kejiou and Bekaroo [2] examined vulnerability scanning tools for Wireless Local Area Networks (WLANs), highlighting the ways in which various programs specialize in identifying certain vulnerabilities yet differ in terms of accuracy and output granularity. These studies underline the importance of comparative evaluations in identifying the most effective tools for varying contexts.

Recently deep learning (DL) approaches for vulnerability detection have been learned on historical data, which opened the door for better accuracy. For example, Lamrani Alaoui and Nfaoui presented a systematic review of the research works based on DL algorithms for

the detection of web attacks, highlighting the need for unified datasets and frameworks to bridge the gap between research endeavors and practical implementations in the field [3]. Although DL techniques are promising, traditional vulnerability scanning tools cover a foundational role in identifying baseline issues and then informing higher-level detection models.

This paper builds upon these insights by focusing on the comparative capabilities of traditional vulnerability scanning tools in identifying unique web application vulnerabilities. By analyzing the types of vulnerabilities identified, their overlap across tools, and the deduplication of results, this study offers a nuanced understanding of each tool's strengths and limitations. The findings contribute to the broader field of web application security by providing actionable insights into tool selection and usage for practitioners and researchers alike.

The increasing reliance on web applications for critical functions in business, social, and public domains has heightened exposure to cybersecurity threats. Vulnerabilities in these applications have become prominent attack vectors, underscoring the urgent need for robust security measures. This study addresses this challenge by evaluating the effectiveness of four widely-used vulnerability scanning tools—Nessus, Acunetix, OWASP ZAP, and BeSECURE. By analyzing their performance across 67 web applications, the research seeks to identify the unique vulnerabilities detected by each tool. Through this comparative analysis, the study intends to provide actionable insights into the capabilities and limitations of these tools, ultimately contributing to the development of more secure web applications.

The quick growth of web apps has changed industries, allowing new business ways, easy service giving, and better social talks. But this surge has brought big security problems, with web app weaknesses becoming a main target for attackers. Cyber threats like SQL Injection Attacks that use database gaps to harm whole systems show the urgent need for strong weakness finding and fixing methods. Tools like Nessus, Acunetix, OWASP ZAP, and BeSECURE have emerged as critical assets in addressing these challenges by systematically identifying security gaps in web applications.

The role of vulnerability scanning tools in securing web applications parallels frameworks that evaluate other security technologies, such as Web Single Sign-On (SSO) systems [5]. Similar to how SSO systems are analyzed for their tradeoffs in usability, privacy, and security, vulnerability scanning tools require careful evaluation to determine their effectiveness in identifying and categorizing threats. These tools operate as a critical first line of defense by detecting exploitable weaknesses before attackers can exploit them, complementing other protective measures like intrusion prevention systems and secure authentication frameworks.

This study builds upon previous works by Ali [4], which emphasized the significance of understanding attack vectors like SQLIA, and Alaca and Van Oorschot [5], who highlighted the necessity of comprehensive evaluation frameworks for cybersecurity solutions. The need for rigorous vulnerability assessment aligns with broader themes in cybersecurity and software development methodologies. Yeng et al. [6] underscore in their analysis of software development practices the critical role of incorporating security considerations during the development lifecycle, particularly in sensitive domains like healthcare. Vulnerability scanning tools contribute significantly to this proactive approach, offering targeted insights into security flaws that may otherwise go undetected. By identifying and analyzing the unique vulnerabilities flagged by these tools, the present study complements this methodological emphasis, demonstrating their practical application in enhancing secure development practices.

According to Hamza and Hammad [7], there are a variety of testing methods which can be employed in web and mobile applications, such as black-box, white-box, and gray-box testing. These tools are often utilized in one or more of the previously mentioned paradigms.

Also, their relative performance offers some insightful information on their advantages and disadvantages. This contribution builds on the previous work by concentrating on how tools of this type of deal with security issues of web applications with respect to the different contexts they use for detection.

Aslan et al. [8] further emphasizes the dynamic and multifaceted nature of cybersecurity threats, detailing how emerging technologies like machine learning and cloud computing create new vulnerabilities. They argue that traditional security systems are increasingly inadequate for modern, sophisticated attacks. The current study resonates with these findings, offering an empirical evaluation of vulnerability scanning tools' ability to adapt to such evolving threats.

This paper takes a comparative approach, not only pushing the conversation on vulnerability assessment forward but also connecting theory with real-world insights. This link underscores how crucial it is to keep evaluating and improving cybersecurity tools to better protect our digital ecosystems. It is a perspective that resonates with the work of Yohanandhan et al. [9] and Ahmad et al. [10], who also stress the need for innovative strategies to tackle cyber risks in complex areas like cloud computing and cyber-physical systems. The importance of IT Security literacy is crucial and needs to be revisited by major organizations around the world, specifically by governments soon [11].

BACKGROUND AND THEORETICAL FRAMEWORK

The study provides a detailed analysis of the effectiveness of web application vulnerability scanning tools using statistical methods. The study will try to make the most in-depth connection with the existing theories and models in information security, which are:

1) Threat and Vulnerability Management (TVM): The study aligns with TVM by evaluating tools that identify and assess vulnerabilities, helping organizations prioritize and manage security risks effectively [12].

2) Threat Control Theory and Security Risk Model: The analysis of scanning tools supports this theory by offering insights into which tools act as effective control mechanisms for detecting and mitigating threats [13]. The findings help assess security risks by identifying which tools detect the most critical vulnerabilities, enabling better risk prioritization and mitigation strategies.

3) Intrusion Detection Theory and Incident Response Theory: The study enhances intrusion detection systems by highlighting tools that effectively identify vulnerabilities, which are critical for preventing potential intrusions. The study aids incident response by improving the initial vulnerability detection phase, ensuring quicker and more effective responses to security incidents [14].

4) Layered Defense Model: The research supports a multi-layered defense approach by demonstrating that combining tools like OWASP ZAP, Nessus, and Acunetix provides comprehensive vulnerability coverage.

By linking the study's empirical findings to these established theories, it strengthens its relevance and applicability in improving information security practices.

Overview of the selected tools

The vulnerability scanning tools play a critical role in identifying and mitigating security risks within web applications. This study evaluates four widely used tools Nessus, Acunetix, OWASP ZAP, and BeSECURE, each offering unique capabilities and methodologies for detecting vulnerabilities. These tools were chosen for their complementary scanning approaches, industry recognition, and academic credibility.

1) Diverse Capabilities: Nessus targets network vulnerabilities, Acunetix focuses on web flaws, OWASP ZAP is a recognized industry-standard open-source tool [15], and BeSECURE provides automated scanning.

2) Proven Reliability & Standards Compliance: Widely used in cybersecurity research and industry, ensuring credible vulnerability detection. Nessus and Acunetix align with ISO 27001, NIST, and PCI DSS, while OWASP ZAP is a recognized industry standard.

3) *Scientific Justification*: Selection is backed by prior studies on scanner efficiency and effectiveness.

The features and limitations of these tools as reported in prior studies are shown in Table 1:

Tool	Key Features	Focus Areas	Known Strengths	Known Weaknesses
Nessus	Extensive plugin library, Network and web application scanning, Scalable and user-friendly, Powerful reporting tools	Network security, compliance auditing	Reliability and versatility, broad vulnerability detection, Industry trust and widespread adoption	High false positives, Limited web app testing, Resource-intensive, Expensive for enterprises
Acunetix	Automated scanning, SQLi/XSS detection, modern tech support, reporting	Web app security, vulnerability scanning, penetration testing	Fast and efficient scanning, good coverage for modern web technologies, easy-to-use interface	Expensive licensing, Limited network scanning capabilities, can miss logic flaws
BeSECURE	Unified vulnerability management, continuous monitoring, risk scoring & prioritization	Vulnerability management, web/network security audits	Provides detailed remediation steps, web and network security focus, easy to use	Limited advanced features for enterprises, fewer configuration options
OWASP ZAP	Automated/manual testing, active/passive scanning, API testing	Web application security, penetration testing, automation of security checks	Open-source, active community, extensive plugins	Slow performance, steep learning curve, limited automation

Table 1: Summarizes the key features, focus areas, strengths, and weaknesses of popular vulnerability scanning tools, as referenced in studies [16-22].

4)Study Limitations: This study provides a structured and data-driven evaluation of web application vulnerabilities using four well-known scanning tools across 67 diverse web platforms. While the methodology is sound and the findings valuable, several limitations should be acknowledged to support a more balanced interpretation and to guide future research. First, the one-month scanning period limits the temporal scope of the analysis. Vulnerabilities often appear or evolve over time, and a longer study window may capture a more accurate and dynamic picture of security risks.

Second, although duplicate vulnerabilities were removed, the tools used Nessus, Acunetix, OWASP ZAP, and BeSecure differ in detection logic, coverage, and false positive handling. These tool-specific variations may introduce bias, affecting the comparison of their performance.

Third, scanning was conducted through ports 80 and 443 only, within a controlled firewall environment. While this setup reflects common deployment scenarios, it may restrict the tools from detecting certain vulnerabilities that emerge under broader or more complex network conditions.Additionally, the process for validating the detected vulnerabilities—whether manual, automated, or hybrid, is not clearly detailed. This lack of clarity may impact the reliability of the reported results, especially in distinguishing true positives from false alarms.
Finally, while the study includes a wide range of web applications from various sectors and platforms, its findings may not fully generalize to all systems, particularly those with uncommon configurations or security practices.
By recognizing these limitations, the study maintains transparency and encourages further research to build on these findings. Future work could explore longer evaluation periods, broader network configurations, and combined tool approaches to enhance the robustness of vulnerability assessment models.

5)Theoretical Foundation and Alignment with Cybersecurity Models

This study is guided by established cybersecurity frameworks that support effective vulnerability detection, classification, and risk analysis. The use of four scanning tools: Nessus, Acunetix, OWASP ZAP, and BeSecure across 67 diverse web applications reflects the best practices drawn from these models.
The Common Vulnerability Scoring System (CVSS) is used as the main reference for categorizing vulnerabilities by severity: critical, high, medium, and low. Our statistical analysis follows CVSS standards, ensuring consistency and comparability across tools.
The research also aligns with the NIST Cybersecurity Framework (CSF) and Risk Management Framework (RMF), both of which emphasize continuous monitoring and measurable risk evaluation. By analyzing the average, standard deviation, and Pearson correlation of unique vulnerabilities, the study supports the CSF's "Identify" and "Detect" functions and encourages data-driven prioritization.
The use of multiple tools supports the Defense-in-Depth model. Relying on a single scanner may produce incomplete results, while a multi-tool approach reveals differences in detection performance and improves overall coverage.
Focusing on non-duplicated vulnerabilities also reflects the principles of Attack Surface Reduction (ASR). This strategy improves the accuracy of risk profiling and enables more targeted security responses.
By linking our findings to these frameworks, the study provides a practical and theory-informed approach to evaluating vulnerability scanning tools. It offers a replicable and evidence-based method for improving web application security assessments.

OBJECTIVES OF THE STUDY

The primary objectives of this study are as follows:

1) Compare the vulnerability detection effectiveness and consistency of Nessus, Acunetix, OWASP ZAP, and BeSECURE across multiple web applications: The study aims to assess the effectiveness of four widely used vulnerability scanning tools Nessus, Acunetix, OWASP, and BeSECURE, in detecting various vulnerabilities in web applications. By conducting a thorough comparative analysis, the study seeks to understand how each tool identifies and reports security weaknesses in web applications.

2) Identify Overlapping and Unique Vulnerabilities to Assess Tool Complementarity. This study evaluates the accuracy and reliability of four vulnerability scanners—Nessus, Acunetix, OWASP ZAP, and BeSECURE by analyzing overlapping and unique vulnerabilities across 67 web applications. Overlapping findings, such as SQL Injection reported by multiple tools, strengthen confidence in detection accuracy. In contrast, unique findings highlight each tool's specialized capabilities.

To ensure accuracy, duplicate vulnerabilities were removed. This reduction process improves clarity and reflects the true detection scope of each tool. For instance, Nessus may focus on network-level vulnerabilities, while OWASP ZAP excels in application-layer assessments. Understanding these strengths helps security professionals select complementary tools with minimal overlap for broader coverage.

Duplicate removal relied on a three-step AI-driven process:

- Set-Based Deduplication – Converts findings into a Set data structure to eliminate exact duplicates efficiently.
- Text Similarity Analysis - Uses AI models like BERT and fuzzy matching to detect near-duplicates in report descriptions.
- Clustering and Metadata Comparison Applies DBSCAN clustering, similarity hashing, and decision trees to group similar findings. High-confidence duplicates are merged automatically, while ambiguous cases are flagged for manual review.

3) Derive insights and recommendations for security practitioners on tool selection and combined use: The study also seeks to evaluate how well these tools perform in real-world scenarios, reflecting practical application security challenges. By analyzing the precision of their vulnerability detection, the study will assess how effectively these tools can be integrated into security frameworks to protect against emerging web application threats. This study offers practical insights and recommendations for security professionals and developers, helping them make informed decisions when selecting and using vulnerability scanning tools. Additionally, it seeks to contribute to the broader conversation on web application security by demonstrating how these tools can be used more effectively to detect vulnerabilities and strengthen defenses, ultimately promoting the creation of more secure digital systems.

Research Questions

This paper will address these three research questions:

1) *How do the detection capabilities of different web vulnerability scanning tools compare across varying severity levels (Critical, High, Medium, Low)?*

2) *What is the relationship between the variability in detection performance and the reliability of each web vulnerability scanning tool over multiple scans*?

3) *What impact does the overlap in vulnerability detection (duplicate vulnerabilities) have on the overall effectiveness of different scanning tools in identifying unique web application vulnerabilities*?

The structure of this paper is organized as follows. Research Methodology outlines the methodological framework used in the study. Descriptive Statistics Analysis of Web Vulnerability Scanning Tools presents an empirical evaluation of different tools across various severity levels. It begins with An Empirical Perspective on Total Vulnerability Detection, examining the overall detection capabilities of the tools. Finally, Conclusion and Future Work summarizes the key findings, discusses their implications, and provides recommendations for future research aimed at improving web application vulnerability detection methodologies.

RESEARCH METHODOLOGY

This study employs a quantitative, comparative, and correlational research approach to assess the effectiveness, consistency, and interdependence of four widely used web application vulnerability scanning tools: OWASP ZAP, BeSECURE, Nessus, and Acunetix. The methodology is structured into multiple phases, ensuring a rigorous and replicable assessment of the tools' vulnerability detection capabilities. All tools use criteria (CVSS) for categorizing vulnerabilities into high, medium and low levels. CVSS or Common Vulnerability Scoring System is an open framework used to assess and quantify the severity of software vulnerabilities. The high severity score range is 7.0 - 10.0, medium severity is 4.0 - 6.9 and

Low severity 0.1 - 3.9. Except for Nessus, which divides the high category into critical (9.0-10) and high (7.0-9.0), but which will be treated based on the CVSS values. So, a vulnerability with CVSS ≥9.0 is counted as "Critical" for Nessus, whereas other tools might lump those into High.

Research design and approach

This study follows a comparative analysis framework that systematically evaluates the performance of each tool based on empirical data collected from 67 web applications. The research is both descriptive and inferential, incorporating statistical measures to assess detect efficacy and reliability. For Descriptive Statistics, mean and standard deviation are used to measure the overall detection capability and consistency of each tool.

Data Collection and Scanning Procedure

To ensure standardization and validity, the data collection was conducted under controlled conditions, so each tool scanned identical targets with no interference.

Analyzed Web Applications

To address the reviewer's request for greater clarity, we provide a more detailed description of the 67 web applications examined in this study, because not every website offers security [24]. A high-quality dataset will enhance the model in automating the process in a proper manner [25]. The sample was purposefully selected to reflect a broad and realistic spectrum of web environments, ensuring diversity in architecture, functionality, and sector. These applications span multiple domains, including Government portals for citizen services and public data access, Healthcare systems supporting scheduling and record management, Financial platforms for online transactions and client services, Content Management Systems (e.g., WordPress, Drupal), and Custom-built business applications, including registries and enterprise tools.

The systems range from large-scale national platforms to medium-sized local services, varying in complexity and security posture. Their functionalities include public information delivery, interactive services, and backend data processing—offering a representative array of modern cybersecurity challenges.

All applications were tested within ethical and legal boundaries. No unauthorized or intrusive activity was conducted, and all targets fell within the approved scope of analysis.
This clarification strengthens the context of our findings and affirms the practical relevance of the dataset. By capturing a cross-section of real-world systems, the study offers meaningful insight into the performance of vulnerability scanning tools across varied environments.
Vulnerability scans were performed on each web application separately using the four tools. The four tools have been installed in the same computer machine. All 67 web applications are hosted in the same ICT infrastructure with the same network computer IP addressing. Between the computer system and the ICT infrastructure, there is a firewall configured uniformly for all web applications, permitting traffic exclusively through ports 80 and 443. The scans were performed using a black-box approach, meaning the tools had no access to the internal code or authentication mechanisms of the web applications. All tools were assumed to have the same level of external access. The tools operated with default configurations, with no custom tuning for specific applications. The scans were conducted over a 1-month period, ensuring consistency in the testing timeframe for all applications. After conducting scans on all 67 web applications to identify their vulnerabilities, each web application was scanned using four different tools: Nessus, Acunetix, OWASP ZAP, and BeSecure. This process generated a total of 268 reports with the results (67 applications × 4 tools). Each report was carefully analyzed, extracting relevant vulnerability data (see Figure 1,2 and 3 for sample report results from all tools). The results from these reports were compiled into a Microsoft Excel dataset for further analysis. Duplicate vulnerabilities detected across multiple tools were then removed using AI-assisted filtering. Finally, a refined dataset containing unique vulnerabilities was created and

stored in Microsoft Excel format for subsequent evaluation. As an example, the tables below display screenshots of the report results generated from four tools utilized in web application number 67. Due to data confidentiality, the application's name has been removed from these images.

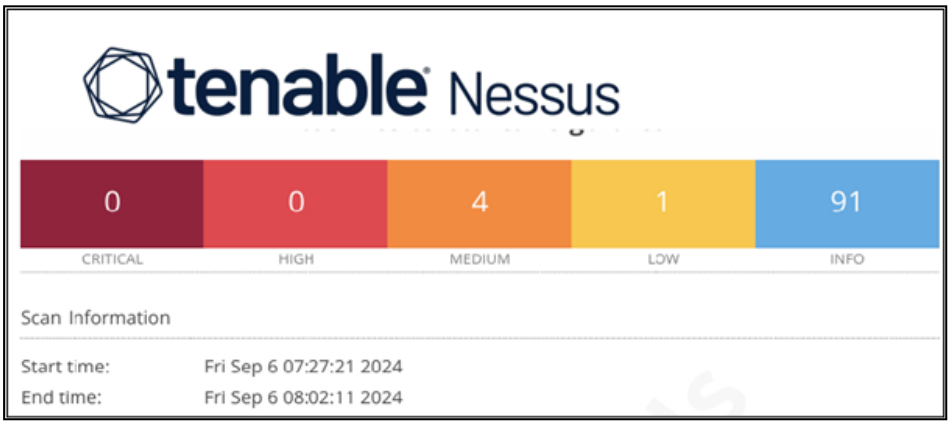

Figure 1. Presents the Nessus scan results for web application number 67

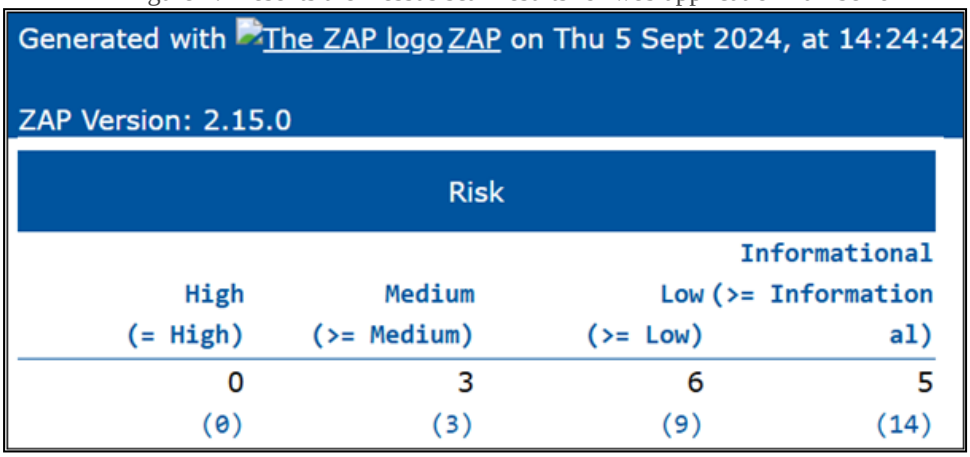

Figure 2. Presents the OWASP ZAP scan results for web application number 67

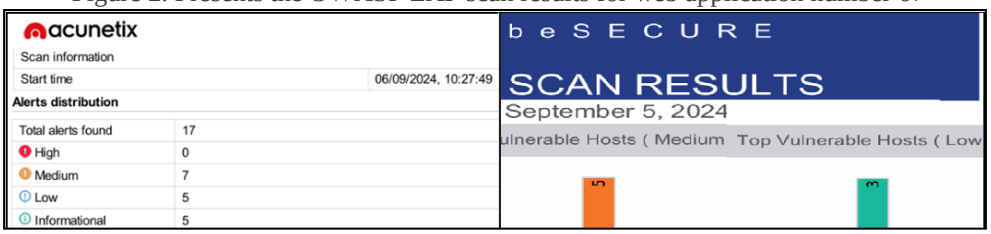

Figure 3. Presents the Acunetix and beSECURE scan results for web application number 67

This study will focus on vulnerabilities categorized as critical, high, medium, and low. False positives, false negatives, and informational vulnerabilities will not be covered.

Metrics and Analysis

The study evaluates the tools based on two primary statistical metrics Mean (Effectiveness Measure) and Standard Deviation (Consistency Measure) [26].

The mean $\mu = (\sum X_i)/N$, measures the average number of vulnerabilities detected by each tool. Indicates how well a tool identifies critical threats in a web application environment. When, μ is average security scan result which represents the average number of vulnerabilities across all web applications analyzed, $\sum Xi$ represents summation of all security scan results from the 67 web applications, while each Xi represents the result from one web application.

The Standard Deviation $\sigma = \text{sqrt}(\sum(X_i - \mu)^2 / N)$, reflects the stability of detection rates across multiple applications. A higher standard deviation indicates inconsistent

performance, while a lower standard deviation suggests reliable and repeatable detection [27]. σ (Standard Deviation of Security Analysis Results) represents the variation in the security scanning results across different web applications. A higher σ means significant differences in detected vulnerabilities, while a lower σ means more consistent results. ∑ is Summation of Indicates that calculations are performed over all web applications in the study, which in this case are 67.

Xi is security scan result for a specific web application which represents the security vulnerability assessment score, number of detected threats, or another measurable outcome for each web application. The term (X_i – μ) ^2 represents the measures how far each web application's scan result deviates from the average while squaring ensures all deviations contribute positively.

N (Number of Web Applications) represents the total number of samples used in the study. In this case, N = 67, meaning that each scanning tool was used to analyze 67 different web applications, and the results were average for comparison.

The processed data was analyzed using descriptive statistics and comparisons were made to rank the tools on effectiveness (mean) and consistency (SD). No inferential tests were applied (since the focus is descriptive), but the results were interpreted with the research hypotheses in mind

Descriptive Statistics Analysis of Web Vulnerability Scanning Tools

This study presents an in-depth descriptive statistical analysis of vulnerabilities identified across 67 web applications using four industry-standard scanning tools: Nessus, Acunetix, OWASP ZAP, and BeSECURE. The findings are classified into four severity levels: Critical, High, Medium, and Low, providing a structured assessment of the vulnerabilities detected.

To ensure precision and analytical rigor, the following key statistical measures have been calculated for each category of vulnerabilities. To refine the accuracy of the dataset, redundant vulnerabilities were systematically removed, ensuring that only unique vulnerabilities were considered in the final analysis. This approach eliminates duplication bias and provides a more reliable representation of the scanning tools' effectiveness. This study is centered on the quantification and evaluation of web security vulnerabilities by analyzing mean values and standard deviations.

Comparative Analysis of Detection Capabilities Across Threat Levels

The evaluation of web vulnerability scanning tools is a critical endeavor in cybersecurity, ensuring robust defenses against malicious exploitation [28]. This study presents a comparative analysis of four widely used tools OWASP ZAP, BeSECURE, Nessus, and Acunetix based on their effectiveness in identifying security vulnerabilities across 67 web applications.

Tool	Total Vulnerabilities Mean	Total Std.D	High Mean	High Std.D	Medium Mean	Medium Std.D	Low Mean	Low Std.D	Sample Size (N)
OWASP ZAP	10.61	2.881	0.28	0.517	3.45	1.091	6.88	2.164	67
BeSECURE	8.54	3.569	0.15	0.5	3.87	2.296	4.52	2.331	67
Nessus	4.46	5.827	0.69	1.948	2.61	3.000	0.85	0.982	67
Acunetix	5.57	3.439	0.18	0.49	2.13	1.850	3.25	1.878	67

Table 2: The data shows the mean and standard deviation (Std.D) of vulnerabilities detected by different security tools, categorized by severity levels (High, Medium, Low). The sample size (N) is 67 for each tool.

1)Empirical Evaluation of Detection Effectiveness: The average number of vulnerabilities found per online application is a key indicator when evaluating vulnerability scanners. OWASP ZAP is the most widely utilized vulnerability scanner and is considered the optimal choice for comprehensive vulnerability assessments, with an average of 10.61 vulnerabilities identified per online application. BeSECURE detects an average of 8.54 vulnerabilities, with 20 of those categorized as critical. In contrast, Acunetix identifies an average of 5.57 vulnerabilities, primarily of lower severity. Nessus has the lowest number of vulnerabilities found, with only 4.46. The reliability of the instrument is as important as the detection rates. According to it, OWASP ZAP has the least variability (2.881), thus being the most reliable option, whereas Nessus has the largest variability (5.827) and thus raises questions about it reliability across situations. With the standard deviations of 3.569 and 3.439 for BeSECURE and Acunetix respectively, a significant variability is indicated.

These findings suggest strategic takeaways for security teams. OWASP ZAP's high effectiveness and consistency make it the preferred tool for organizations prioritizing comprehensive vulnerability detection. BeSECURE, with strong detection but moderate variability, should be considered a complementary tool. Acunetix offers stable performance and is useful in specific testing scenarios, while Nessus, with its high variability and low detection rate, is best suited as a supplementary scanner rather than a primary tool.

2) Critical-level vulnerabilities: These severities represent the most severe security risks, potentially leading to unauthorized system control, data breaches, and remote code execution [29]. The detection of such vulnerabilities is essential for maintaining a robust cybersecurity posture. The investigation found that Nessus stands out because it is the only tool with a specific category for critical vulnerabilities. During the study, it identified 21 major vulnerabilities across 67 online applications, classifying them as critical. On the other hand, tools like OWASP ZAP, BeSECURE, and Acunetix, when identifying the same vulnerabilities, categorize them as high-level, not critical. This shows that Nessus provides a more detailed classification and more accurate prioritization of risks, especially for the most severe vulnerabilities that could cause significant damage if exploited. However, other tools still play an important role in detecting and managing high, medium, and low-level vulnerabilities. This highlights the need to use a combination of different scanning tools to achieve a comprehensive security assessment.

3) High-level vulnerabilities: Pose serious threats to systems and networks, allowing attackers to exploit weaknesses for unauthorized access, malicious activities, or data theft [30]. This study evaluates four vulnerability scanning tools—Nessus, Acunetix, OWASP ZAP, and BeSECURE—based on their effectiveness (average detection rates) and consistency (standard

deviation) in detecting high-level vulnerabilities across 67 web applications. Nessus (mean = 0.69) leads with the highest detection rate, identifying nearly twice as many vulnerabilities as OWASP ZAP (mean = 0.28). Acunetix (mean = 0.18) detects fewer vulnerabilities than both OWASP ZAP and Nessus, while BeSECURE (mean = 0.15) is the least effective. In terms of consistency, Acunetix (SD = 0.49) and BeSECURE (SD = 0.5) show the lowest variability, suggesting stable performance in predictable environments. OWASP ZAP (SD = 0.517) also offers low variability, while Nessus (SD = 1.948) has the highest variability, indicating fluctuating performance across different applications. Nessus (1.948) exhibits a much higher standard deviation, signifying significant variability in its detection rates. While it has the highest mean, its inconsistency suggests it may perform unevenly depending on the context or configuration of the web applications. Both tools Acunetix and BeSECURE prioritize consistency over detection efficacy, as reflected by their low standard deviations. Their performance may be more predictable in specialized or smaller-scale applications, despite their relatively low vulnerability detection rates.

Regarding recommendations for practical use, Nessus is the best for detecting a wide range of vulnerabilities, while OWASP ZAP is ideal for validating findings and ensuring consistent results. Acunetix or BeSECURE are the Best for environments requiring stable, predictable performance.

Balancing Cost and Performance: For resource-constrained organizations, OWASP ZAP offers an optimal balance of cost, efficacy, and reliability. Acunetix and BeSECURE are better for specific threats requiring consistent performance.

4) Medium-level vulnerabilities: These severity levels, such as misconfigurations and outdated software, pose moderate risks and can escalate to severe attacks if ignored. This analysis compares the effectiveness of Nessus, Acunetix, OWASP, and BeSECURE in detecting these vulnerabilities across 67 web applications, focusing on average detection rates (mean) and variability (standard deviation). The findings reveal varying performance across the tools. OWASP shows the least variability, indicating consistent results. BeSECURE exhibits higher variability, detecting a broader range of medium vulnerabilities across different scans. Nessus has the highest variability, with detection rates fluctuating significantly between scans. These differences underscore the importance of tool choice depending on the need for consistency versus broader detection coverage. Acunetix has moderate variability, suggesting that while the tool is consistent, there is still some variability in the number of medium vulnerabilities it identifies. The BeSECURE tool outperforms others in detecting medium-severity vulnerabilities, with a mean of 3.87 findings per scan, significantly higher than OWASP ZAP 3.45, Nessus 2.61, and Acunetix 2.13. This suggests it is more effective for comprehensive medium-risk vulnerability detection. Acunetix lags detecting only 2.13 medium vulnerabilities on average, the lowest among the group. OWASP ZAP provides a reliable middle ground, detecting 3.45 medium vulnerabilities on average, offering a balance between performance and reliability. Nessus shows the highest variability (SD = 3.030), meaning its results fluctuate significantly, making it less predictable. BeSECURE (SD = 2.296) and Acunetix (SD = 1.850) offer more consistency, with Acunetix being the most stable. Regarding Vulnerability Coverage, the BeSECURE's higher detection rate makes it more comprehensive. OWASP and Acunetix are more consistent (lower SD), providing stable results. Nessus's high variability may require additional validation tools.

Tool Selection: Organizations seeking broad coverage should prioritize BeSECURE, while those needing stable and predictable results should lean towards OWASP or Acunetix.

5) Low-level vulnerabilities: The effectiveness of vulnerability scanning tools in detecting low-level security weaknesses is a crucial aspect of web application security [31]. Low-level vulnerabilities may not pose an immediate threat but can be exploited in multi-stage attacks. This study provides an empirical analysis of four widely used vulnerability scanners

OWASP, BeSECURE, Nessus, and Acunetix based on their detection rates and correlation in scanning 67 web applications. OWASP ZAP detects the highest number of low-level vulnerabilities (Mean = 6.88), suggesting a broad detection strategy that focuses on comprehensive scanning of minor security flaws. BeSECURE takes a balanced Approach which finds a moderate number of low-level vulnerabilities (Mean = 4.52). It does not go as deep as OWASP ZAP, but it strikes a nice balance between catching minor issues and focusing on more serious threats. Think of it as the multitasker of the group. Acunetix detects fewer low-level vulnerabilities (mean = 3.25), suggesting it prioritizes higher-severity issues. This indicates a more selective scanning focus on bigger issues rather than exhaustive low-level findings. It still catches some minor weaknesses, but it is not its main priority. It is like the sniper precise but selective. Nessus is excellent at detecting high-level and critical vulnerabilities but is not effective in identifying low-level ones.

Comparative analysis of unique and total vulnerabilities

On Table 3 statistics show unique vulnerabilities and total vulnerabilities including duplicates.

Category	Mean	Standard Deviation	N
Total Unique Vulnerabilities	18.43	6.514	67
Total Vulnerabilities (Including Duplicates)	29.18	10.748	67
Difference (Duplicates Found Across Tools)	10.75	5.514	67

Table 3: The data shows the mean and standard deviation of unique vulnerabilities, total vulnerabilities (including duplicates), and the difference representing duplicates found across different security tools.

Based on the table provided, we can calculate the percentage of duplicated vulnerabilities using the following formula:

Percentage of Duplicates = (Difference (Duplicates Found Across Tools) / Total Vulnerabilities (Including Duplicates)) × 100

The percentage difference in vulnerabilities found (due to duplicates) is 36.84%. This means that 36.84% of the total vulnerabilities found were duplicates. On average, 18.43 unique vulnerabilities are detected per web application, while the tools identified 29.18 total vulnerabilities, with duplicates accounting for 36.84% of the total findings, indicating significant overlap between tools. The standard deviation for total vulnerabilities (10.748) suggests considerable variability in duplicate findings, while the standard deviation for unique vulnerabilities (6.514) indicates consistent detection. The variation in duplicates (5.514) reflects differing levels of overlap across tools. Although multiple tools lead to duplicate findings, selecting and configuring them strategically can improve scanning efficiency and reduce redundancy. This supports our hypothesis that a significant fraction of detected vulnerabilities would be duplicates across tools – indeed 36.8% of all findings overlapped. Employing correlation or result-merging techniques could thus meaningfully streamline multi-tool assessments.

CONCLUSION

In summary, OWASP ZAP emerged as the top all-around performer, excelling in both detection and consistency. Each of the other tools performed well in specific areas: Nessus excelled in identifying critical and high-severity vulnerabilities, BeSECURE was effective in detecting medium-level vulnerabilities, and Acunetix provided stable results with fewer findings. These findings suggest that no single tool is sufficient for all scenarios, and therefore, a multi-faceted strategy is recommended for comprehensive vulnerability detection. So, Organizations must adopt a strategic, multi-tool framework tailored to their specific operational needs and risk profiles.

FUTURE WORK

The Future work could be organized into Improving the Comparative Framework. Building on the findings of this study, future research should focus on enhancing the comparative framework for vulnerability scanning tools. This could involve expanding the sample size to include a wider variety of web applications, integrating more tools for a more comprehensive evaluation, and leveraging machine learning techniques to deepen the analysis of vulnerability detection. Longitudinal studies, tracking tool performance across multiple software versions and evolving threat intelligence, would provide valuable insights into how these tools adapt over time. Furthermore, integrating manual and dynamic penetration testing with automated detection can improve validation and increase the overall accuracy of the results.

Another key area for future research is extending the role of vulnerability scanning tools beyond detection. Integrating these tools with remediation processes and DevSecOps pipelines is essential to bridge the gap between identifying vulnerabilities and mitigating them effectively. Incorporating automated remediation suggestions and security orchestration tools would enable more streamlined and efficient vulnerability management. Additionally, using AI and machine learning for risk prioritization and automation could strengthen cybersecurity defenses and foster a proactive approach to security management.

REFERENCES

1. Ali, N. S., Shibghatullah, A. S. B., Alhilali, A. H., Al-Khammasi, S., Kadhim, M. F., & Fatlawi, H. K. (2020). A comparative analysis and performance evaluation of web application protection techniques against injection attacks. International Journal of Mobile Communications, 18(2), 196–228. https://doi.org/10.1504/IJMC.2020.105855
2. Kejiou, A., & Bekaroo, G. (2022). A review and comparative analysis of vulnerability scanning tools for wireless LANs. In 2022 3rd International Conference on Next Generation Computing Applications (NextComp) (pp. 1-8). IEEE. https://doi.org/10.1109/NextComp55567.2022.9932245
3. Lamrani Alaoui, R., & Nfaoui, E. H. (2022). Deep learning for vulnerability and attack detection on web applications: A systematic literature review. Future Internet, 14(4), 118. https://doi.org/10.3390/fi14040118
4. N. S. Ali, "Investigation framework of web applications vulnerabilities, attacks and protection techniques in structured query language injection attacks," Int. J. Wireless Mobile Comput., vol. 15, no. 2, pp. 103-122, 2018, DOI:10.1504/IJWMC.2018.091137
5. F. Alaca and P. C. Van Oorschot, "Comparative analysis and framework evaluating web single sign-on systems," ACM Comput. Surv. (CSUR), vol. 53, no. 5, Article 112, 2020, doi: 10.1145/3409452.
6. P. Yeng, S. Wolthusen, and B. Yang, "Comparative analysis of software development methodologies for security requirement analysis: Towards healthcare security practice," 13th Int. Conf. Inf. Syst., Sofia, Bulgaria, Mar. 2020, DOI:10.33965/is2020_202006L009
7. Hamza, Z. A., & Hammad, M. (2020). Testing approaches for web and mobile applications: An overview. International Journal of Computer and Digital Systems, 9(4), 13. https://doi.org/10.12785/IJCDS/090413
8. Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cybersecurity vulnerabilities, threats, attacks, and solutions. Electronics, 12(6). https://doi.org/10.3390/electronics12061333
9. Yohanandhan, R. V., Elavarasan, R. M., Manoharan, P., & Mihet-Popa, L. (2020). Cyber-physical power system (CPPS): A review on modeling, simulation, and analysis with cybersecurity applications. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.3016826
10. Ahmad, W., Rasool, A., Javed, A. R., Baker, T., & Jalil, Z. (2021). Cybersecurity in IoT-based cloud computing: A comprehensive survey. Electronics, 11(1). https://doi.org/10.3390/electronics11010016
11. Besimi, A., & Shehu, V. (2020). Technology: COVID-19 and the 'new-normal' lifestyle vs. security challenges. SEEU Review, 15(1), 71. https://doi.org/10.2478/seeur-2020-0005
12. M. Alhamed and M. M. Hafizur Rahman, "A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions," Appl. Sci., vol. 13, no. 12, p. 6986, Jun. 2023, doi: 10.3390/app13126986.
13. A. Tundis, W. Mazurczyk, and M. Mühlhäuser, "A review of network vulnerabilities scanning tools: Types, capabilities, and functioning," ARES '18: Proceedings of the 13th International Conference on Availability, Reliability and Security, vol. 1, pp. 1-10, Aug. 2018, doi https://doi.org/10.1145/3230833.3233287
14. Khalid, M. N., Iqbal, M., Rasheed, K., & Abid, M. M. (2020). Web Vulnerability Finder (WVF): Automated black-box web vulnerability scanner. Journal of Information Technology and Computer Science, 2020(4), 38–46. https://doi.org/10.5815/ijitcs.2020.04.05

15. Systematic Literature Review: Security Gap Detection on Websites Using OWASP ZAP." Brilliance: Research of Artificial Intelligence, vol. 4, no. 1, May 2024. h https://doi.org/10.47709/brilliance.v4i1.4227
16. Y. Chen, A. E. Santosa, A. Sharma, and D. Lo, “Automated identification of libraries from vulnerability data,” ICSE-SEIP '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice, pp. 90–99, Sep. https://dl.acm.org/doi/10.1145/3377813.3381360
17. R. W. Scholz, R. Czichos, P. Parycek, and T. J. Lampoltshammer, “Organizational vulnerability of digital threats: A first validation of an assessment method,” European Journal of Operational Research, 2019. https://doi.org/10.1016/j.ejor.2019.09.020
18. Mi, F., Wang, Z., Zhao, C., Guo, J., Ahmed, F., & Khan, L. (2021). VSCL: Automating vulnerability detection in smart contracts with deep learning. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). IEEE. https://doi.org/10.1109/ICBC51069.2021.9461050
19. Chancusi, A., Diestra, P., & Nicolalde, D. (2021). Vulnerability analysis of the exposed public IPs in a higher education institution. In ICCNS '20: Proceedings of the 2020 10th International Conference on Communication and Network Security (pp. 83–90). https://doi.org/10.1145/3442520.3442523
20. Rathi, S. C., Misra, S., Colomo-Palacios, R., Adarsh, R., Neti, L. B. M., & Kumar, L. (2023). Empirical evaluation of the performance of data sampling and feature selection techniques for software fault prediction. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2023.119806
21. Li, X., Wang, L., Xin, Y., Yang, Y., & Chen, Y. (2020). Automated vulnerability detection in source code using minimum intermediate representation learning. Applied Sciences, 10(5), 1692. https://doi.org/10.3390/app10051692
22. Jorepalli, S. (2022). Trends in threat vulnerability management: Advanced techniques for proactive network security. International Journal on Recent and Innovation Trends in Computing and Communication, 10(10), 218. http://www.ijritcc.org
23. Alqarni, M., & Azim, A. (2022). Low level source code vulnerability detection using advanced BERT language model. 35th Canadian Conference on Artificial Intelligence. https://www.researchgate.net/publication/363018292_Low_Level_Source_Code_Vulnerability_Detection_Using_Advanced_BERT_Language_Model
24. Arifi, D., & Arifi, B. (2020). Cybercrime: A challenge to law enforcement. SEEU Review, 15(2), 42. https://doi.org/10.2478/seeur-2020-0016
25. Fetahi, E., Hamiti, M., Susuri, A., Zenuni, X., & Ajdari, J. (2024). Integrating handcrafted features with machine learning for hate speech detection in Albanian social media. SEEU Review, 19(2), 80. https://doi.org/10.2478/seeur-2024-0025
26. A. W. Ayeni, "Empirics of standard deviation," Research Presentation, Covenant Univ., May 2014, DOI:10.13140/2.1.1444.6729.
27. Przystupa, K., Kolodiy, Z., Yatsyshyn, S., Majewski, J., Khoma, Y., Petrovska, I., Lasarenko, S., & Hut, T. (2023). Standard deviation in the simulation of statistical measurements. Metrology and Measurement Systems. https://doi.org/10.24425/mms.2023.144403
28. Markevych, M., & Dawson, M. (2023). A review of enhancing intrusion detection systems for cybersecurity using artificial intelligence (AI). Knowledge-Based Organization, 29(3). https://doi.org/10.2478/kbo-2023-0072
29. Luo, F., Jiang, Y., Zhang, Z., Ren, Y., & Hou, S. (2021). Threat analysis and risk assessment for connected vehicles: A survey. Security and Communication Networks, 2021, Article 1263820. https://doi.org/10.1155/2021/1263820

30. Moore, E. L., Fulton, S. P., Mancuso, R. A., Amador, T. K., & Likarish, D. M. (2021). A layered model for building cyber defense training capacity. In Information Security Education for Cyber Resilience (pp. 64–80) https://link.springer.com/chapter/10.1007/978-3-030-80865-5_5
31. Priyawati, D., Rokhmah, S., & Utomo, I. C. (2022). Website vulnerability testing and analysis of website application using OWASP. International Journal of Computer and Information System (IJCIS), 3(3). http://www.ijcis.net/index.php/ijcis/article/view/90