p-ISSN: 2252-3006
e-ISSN: 2685-2411

Penetration Testing of XYZ Application using OWASP WSTG and NIST SP 800-115

Komang Kurnia Suestiana[a1], Gusti Made Arya Sasmita[a2], I Made Sunia Raharja[a3]

[a] Department of Information Technology, Faculty of Engineering, Udayana University
Jimbaran, Badung, Bali, Indonesia
e-mail: [1]suestiana@student.unud.ac.id, [2] aryasasmita@unud.ac.id, [3]sunia.raharja@unud.ac.id

Abstrak

Keamanan merupakan aspek terpenting dalam sistem informasi, sebuah sistem yang terhubung ke internet membuka potensi adanya lubang keamanan. Oleh karena itu diperlukan adanya pengujian keamanan untuk mengurangi risiko penyerangan terhadap sistem, yaitu dengan melakukan penetration testing. Penelitian ini bertujuan untuk mengimplementasikan pengujian keamanan pada aplikasi XYZ menggunakan framework OWASP WSTG (Web Security Testing Guide) dan NIST SP 800-115. Pengujian berhasil mengidentifikasi sejumlah celah keamanan dengan tingkat keparahan yang berbeda. Kedua framework dapat digunakan untuk melakukan pengujian penetration testing. Namun, terdapat beberapa perbedaan di antara keduanya, seperti pada fokus utama, pendekatan pengujian, tingkat detail teknis, kedalaman pengujian, serta keahlian yang dibutuhkan oleh penguji. OWASP WSTG menyediakan panduan yang lebih detail untuk pengujian kerentanan pada aplikasi web, sedangkan NIST SP 800-115 lebih berfokus pada metodologi dan tahapan dalam proses penetration testing. Hasil dari penelitian ini memberikan rekomendasi yang dapat digunakan untuk meningkatkan keamanan aplikasi web XYZ serta mengurangi potensi risiko serangan siber.

Kata kunci: *Penetration Testing, Framework OWASP WSTG, Framework NIST SP 800-115*

Abstract

Security is one of the most important aspect in information system, a system that connected to the internet opens up the potential of security hole. Therefore security testing is required to minimize the risk of attack on the system, namely by conducting penetration testing. The objective of this research is to implement security testing on XYZ application using OWASP WSTG (Web Security Testing Guide) and NIST SP 800-115 framework. The conducted testing was able to identify several security vulnerabilities of varying severity. Both frameworks can be used to conduct penetration testing. However, there are several differences between the two, such as the primary focus, testing approach, level of technical detail, testing depth, and required expertise of the tester. The OWASP WSTG provides more detailed guidance for vulnerability testing in web applications, while the NIST SP 800-115 focuses more on the methodology and stages of the penetration testing process. The results of this research provide recommendations that can be used to improve the security of XYZ's web application and reduce the potential risk of cyberattacks.

Keywords : *Penetration Testing, Framework OWASP WSTG, Framework NIST SP 800-115*

1. Introduction

The rapid development of information technology offers various conveniences, one of which is data management through information systems. Security is a crucial aspect of information systems. A system connected to the internet opens up the potential for security holes and makes it vulnerable to cyberattacks [1]. The consequences of these attacks include data breach, which can be detrimental to all parties involved [2].

Based on the data that published by BSSN (Badan Siber dan Sandi Negara), a total of 593 suspected cyberattack incidents were recorded in 2024, including 241 suspected data breach cases. The government administration sector experienced the highest number of incidents, with 183 cases [3].

Penetration testing is a method of evaluating the security of computer system or network by simulating attacks from malicious source [4]. Penetration testing is a crucial aspect of ensuring the overall quality and integrity of a system because it helps identify functional errors and security vulnerabilities before the system is widely deployed [5]. Furthermore, web applications are vulnerable to various types of vulnerabilities, particularly those related to input processing, such as input validation issues in forms, which are often the starting point for attackers to exploit [6].

Security testing on the XYZ web application was conducted to determine its level of security and to protect it from potential attacks. The results of this research are expected to provide recommendations for application security.

2. Research Method / Proposed Method

This research uses penetration testing to evaluate web application security by conducting two different testing frameworks, OWASP WSTG (Web Security Testing Guide) and NIST SP 800-115. Stages of research is shown in figure 1.

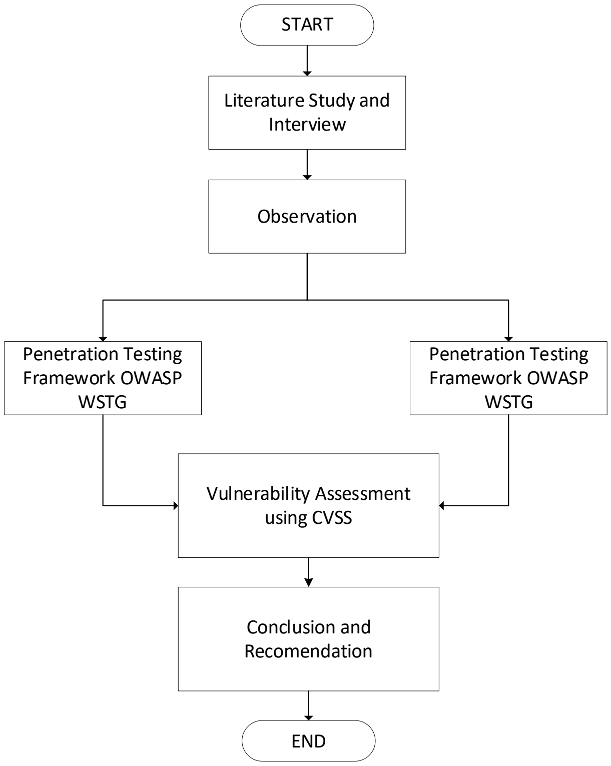

START
Literature Study and Interview
Observation
Penetration Testing Framework OWASP WSTG
Penetration Testing Framework OWASP WSTG
Vulnerability Assessment using CVSS
Conclusion and Recomendation
END

Figure 1. Research flow

This research consists of six stages, first stage began with literature study and interview to gain knowledge about concept of application security and penetration testing technique. Next stage is observation to understand about functionality and scope of the application that being tested. Security testing then conducted separately using OWASP WSTG and NIST SP 800-115 framework. OWASP WSTG was used to identify vulnerabilities based on the web application testing module, while NIST SP 800-115 was implemented following the penetration testing stages. The test results from both frameworks were analyzed through a vulnerability assessment using the CVSS (Common Vulnerability Scoring System) to determine vulnerability severity. The final stage of the research is making conclusions and recomendations based on the test result and a comparative analysis of the two frameworks.

3. Literature Study

The concepts and ideas used in this research were obtained through a literature review, including scientific journals, research reports, and various books relevant to the research topic. These theoretical foundations support the implementation of the research and provide a basis for comparison with related previous studies.

3.1 Penetration Testing

Penetration testing is an approach to assessing the security of a computer system or network by conducting controlled attack simulations as part of a security audit process. These simulations are designed to mimic potential attack scenarios carried out by unauthorized parties, such as hackers or malicious attackers, to identify and exploit security vulnerabilities in the system [7].

3.2 Black Box Testing

Black box testing is a software testing approach that evaluates system functionality by analyzing input-output behavior without knowledge of internal code structures. This method is widely applied to validate system requirements and detect functional defects from an end-user perspective [8]. This approach aligns with the goal of penetration testing to assess the system from the perspective of an end user or external attacker, so the tester does not need prior access or internal information of the system [9].

3.3 Framework OWASP WSTG

OWASP WSTG (Web Security Testing Guide) is a guide for conducting application security testing focused on web application, compiled by many experts in the field of cybersecurity, so it can be used as a guide in testing web applications [10]. OWASP WSTG framework uses a module-based approach to testing. These testing activities are grouped into several categories that cover various aspects of web application security, such as information gathering, authentication testing, authorization testing, and input validation. [11].

3.4 Framework NIST SP 800-115

NIST SP 800-115 provides a structured methodology for conducting penetration testing by defining four key stages: planning, discovery, attack, and reporting [12]. During the planning stage, the scope and goals of the test are defined. The discovery stage focuses on collecting information and scanning for vulnerabilities to understand possible attack points. In the attack stage, the identified weaknesses are tested to see how they could affect system security. Finally, the reporting stage summarizes the results and provides recommendations to fix the identified issues. This step-by-step process helps ensure that penetration testing is carried out in a consistent and practical way, similar to real attack situations [13].

3.5 CVSS

CVSS (Common Vulnerabiltiy Scoring System) is a framework used to assess vulnerabilities in a system. In this research, the assessment focused on Base Metrics, which reflect the inherent nature of a vulnerability. Base Metrics include exploitability aspects such as Attack Vector (AV), Attack Complexity (AC), Attack Requirements (AT), Privileges Required (PR), and User Interaction (UI), as well as the impact on Confidentiality (VC), Integrity (VI), and Availability (VA). The final result of this assessment is a number indicating the severity of the vulnerability, ranging from 0.0 to 10.0 and divided into four levels: low (0.1-3.9), medium (4.0-6.9), high (7.0-8.9), and critical (9.0-10.0) [14].

3.6 Common Web Application Vulnerabilties

Web applications are often exposed to security risks caused by common vulnerabilities found in many systems. Examples include injection attacks such as SQL injection, where user input is not properly validated and can change database queries. Other issues include cross-site scripting (XSS), which allows attackers to run malicious scripts in users browsers, and cross-site request forgery (CSRF), which takes advantage of active user sessions. Weaknesses in authentication and session management can also lead to unauthorized access. These problems usually occur due to poor coding practices, insufficient input validation, or incorrect system configuration, and they are commonly identified during vulnerability assessments and penetration testing [15].

4. Result and Discussion

This section contain result of penetration testing on XYZ Application using OWASP WSTG and NIST SP 800-115 also recommendation for mitigating the vulnerability found during penetration testing.

4.1. OWASP WSTG Penetration Testing Results

Penetration Testing in OWASP WSTG is carried out using a module-based testing approach. Test Result are shown in table 1

Table 1 OWASP WSTG Result

Sub-Module	Objective	Result	CVSS Score
Conduct Search Engine Discovery Reconnaissance for Information Leakage (WSTG-INFO-01)	Exploiting search engines to collect sensitive information that indexed by search engines.	No sensitive information found	-
Fingerprint Web server (WSTG-INFO-02)	To Identify all service running on the web server	found all service running on the web server and no unusual services were detected	-
Review Webserver Metafiles for Information Leakage (WSTG-INFO-03)	To find information leaks on website paths or directories based on analysis from robots, spiders, and crawlers	No information leakage found	-
Enumerate Application on Webserver (WSTG-INFO-04)	Find how many web application run on a same web server as well as open port	Found 10 other website run on the same web server, however this is not a vulnerability	-
Testing for Credentials Transported over an Encrypted Channel (WSTG-ATHN-01)	Ensure that the data exchange that occurs is encrypted over HTTPS	No vulnerability found, all data exchange is done via HTTPS protocol	-
Testing for Default Credentials (WSTG-ATHN-02)	Ensure that the configuration of tools or services installed on the server does not use the default password of the said tool.	No vulnerability found	-
Testing for Weak Lock Out Mechanism (WSTG-ATHN-03)	Testing the Lockout mechanism, this mechanism functions to prevent brute force	There is no lockout mechanism, however there is captcha feature that will	-

Sub-Module	Objective	Result	CVSS Score
	attacks, where the account will be locked if repeated failed login attempts are made.	prevent brute force attack.	
Testing Directory Traversal File Include (WSTG-ATHZ-01)	Testing parameters in urls so that users cannot exploit the system to read or write files that are not intended to be accessible	Vulnerability found, some parameter susceptible to this attack	8.7 (High)
Testing for Reflected Cross Site Scripting (WSTG-INPV-01)	Testing the system's resilience against Reflected Cross-Site Scripting (XSS) attacks	A vulnerability was found, but it has minimal impact because it only occurs when the user is not logged in and does not enable session theft.	0 (None)
Testing for Stored Cross Site Scripting (WSTG-INPV-02)	Testing the system's resilience against Stored Cross-Site Scripting (XSS) attacks	vulnerability found in form due to lack of input sanitization	8.6 (High)
Testing for SQL Injection (WSTG-INPV-05)	Testing the system's resilience against SQL Injection attacks	No vulnerability found	-
Testing for Cross Site Request Forgery (WSTG-SESS-05)	Testing the system's resilience against CSRF (Cross Site Request Forgery) attacks	Vulnerability found due to absence of Anti-CSRF header	6.9 (Medium)

Based on the OWASP WSTG testing results, four vulnerabilities were identified in the XYZ application. One vulnerability was rated 0 (None) under CVSS due to its minimal impact, while two were classified as medium severity and one as high severity.

4.2. NIST SP 800-115 Penetration Testing Result

NIST SP 800-115 Penetration testing is carried out in four stages, namely planning, discovery, attack, and reporting.

4.2.1 Planning

At the planning stage, the scope of the application being tested is determined based on an agreement with the relevant team at PT XYZ. The scope is limited to the domain of the XYZ application, and testing stages such as discovery and attack are conducted on a prepared cloned application to ensure that the production environment is not affected.

4.2.2 Discovery

The discovery stage involves collecting information and mapping the target to understand its structure, configuration, and potential attack surface. During this stage, information gathering and vulnerability scanning activities are performed.

4.2.2.1 Information Gathering

Table 2 Information Gathering NIST SP 800-115

Process	Objective	Result
Identifying Web server and OS	Identifying the web server and operating system versions used by the server in order to determine whether the currently deployed versions contain known weaknesses.	The Nmap scan shows that the XYZ application uses an nginx web server, but the version is unknown. The operating system could not be detected because Nmap did not meet the requirements for OS detection.
Search for domain registration and IP Block information	Identifying domain-related information used by the target application, including registration data and IP address ranges, in order to gather details about the target's infrastructure.	The results of testing using the whois tool indicate that no domain registration information or IP block details were found for the XYZ application.
Reverse DNS Lookup	Identify network resources based on target IP addresses.	No information leaks were found.
Reverse IP Lookup	Identify other domains that are on the same IP address as the target domain.	The IP lookup results show that there are ten other domains sharing the same IP address as the XYZ application.
Port Scanning	Identify services running on the target system based on detected open ports	The Nmap TCP port scan identified several open ports, including port 53 for DNS, and ports 80 and 443 for HTTP and HTTPS services using nginx. However, these open ports are standard for web applications, and no unusual services were detected.

Based on the table, the information gathering identified basic infrastructure and service information of the XYZ application. The results show a standard web application configuration with no unusual services or information leaks detected. Overall, these findings provide baseline information for further testing stages.

4.2.2.2 Vulnerability Scanning

Vulnerability scanning was conducted to identify potential security weaknesses that could be exploited in the XYZ application using the ZAP tool. The scan results were classified into four risk categories, namely high, medium, low, and informational.

Table 3 Vulnerability Scanning NIST SP 800-115

Vulnerability	Risk Level
Cross Site Scripting (Reflected)	High
Path Traversal	High
SQL Injection	High
Absence of Anti-CSRF Token	Medium
Backup File Disclosure	Medium
Authentication Request Identified	Informational

4.2.3 Attack

At this stage, testing is performed on the vulnerabilities identified during the discovery stage to confirm that they exist in the system.

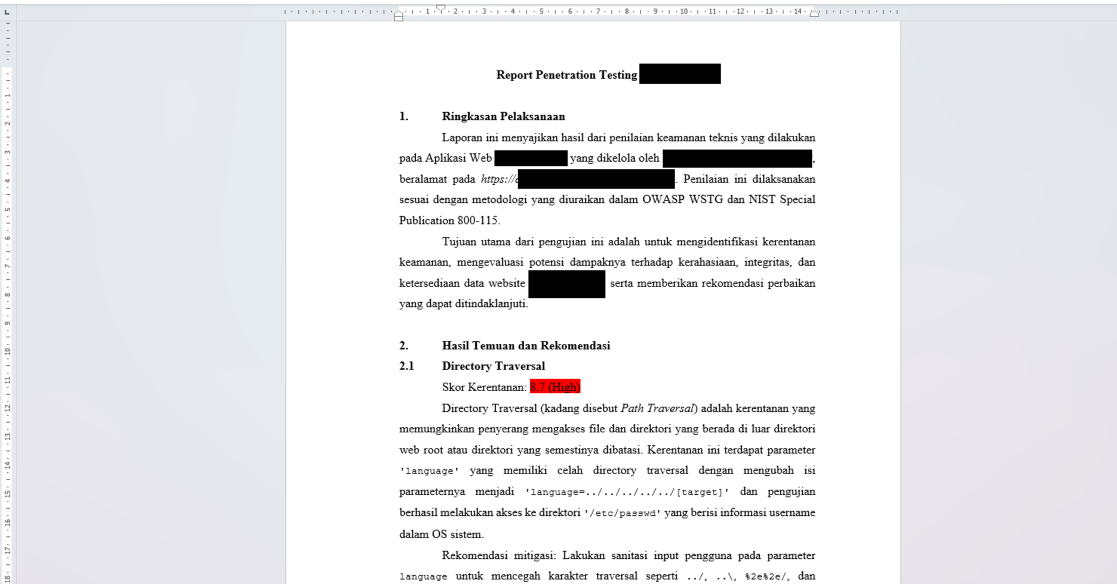
Table 4 Attack Stage NIST SP 800-115

Vulnerability	Risk Level	Result	CVSS Score
Cross Site Scripting (Reflected)	High	A vulnerability was found, but it has minimal impact because it only occurs when the user is not logged in and does not enable session theft.	0 (None)
Path Traversal	High	Vulnerability was discovered that could allow users to access files outside of the specified directory.	8.7 (High)
SQL Injection	High	The testing showed that the vulnerability was not found and was classified as a false positive.	-
Absence of Anti-CSRF Token	Medium	vulnerability discovered that could lead to CSRF attacks	6.9 (Medium)
Backup File Disclosure	Medium	ZAP detected a backup file located at “/leaflet.zip” on the website. Further investigation showed that the file is a third-party Leaflet library used to display maps, not an actual backup file of the website. Therefore, this finding was classified as a false positive.	-
Authentication Request Identified	Informational	This finding does not directly represent a security vulnerability, but only provides technical information useful for the information gathering stage.	-

The table presents the validation results of the identified vulnerabilities. Path traversal and the absence of anti-CSRF tokens were confirmed as actual security issues, while reflected XSS showed minimal impact. Other findings, such as SQL injection and backup file disclosure, were identified as false positives. Informational findings do not pose direct security risks but provide supporting technical information.

4.2.4 Reporting

At this stage, a report is prepared documenting the test findings, including a summary of the vulnerabilities found, the methods used, evidence of exploitation, the potential impact of the weaknesses, and mitigation recommendations.



Report Penetration Testing [redacted]

1. Ringkasan Pelaksanaan

Laporan ini menyajikan hasil dari penilaian keamanan teknis yang dilakukan pada Aplikasi Web [redacted] yang dikelola oleh [redacted], beralamat pada *https://*[redacted]. Penilaian ini dilaksanakan sesuai dengan metodologi yang diuraikan dalam OWASP WSTG dan NIST Special Publication 800-115.

Tujuan utama dari pengujian ini adalah untuk mengidentifikasi kerentanan keamanan, mengevaluasi potensi dampaknya terhadap kerahasiaan, integritas, dan ketersediaan data website [redacted] serta memberikan rekomendasi perbaikan yang dapat ditindaklanjuti.

2. Hasil Temuan dan Rekomendasi

2.1 Directory Traversal

Skor Kerentanan: 8.7 (High)

Directory Traversal (kadang disebut *Path Traversal*) adalah kerentanan yang memungkinkan penyerang mengakses file dan direktori yang berada di luar direktori web root atau direktori yang semestinya dibatasi. Kerentanan ini terdapat parameter `'language'` yang memiliki celah directory traversal dengan mengubah isi parameternya menjadi `'language=../../../../../[target]'` dan pengujian berhasil melakukan akses ke direktori `'/etc/passwd'` yang berisi informasi username dalam OS sistem.

Rekomendasi mitigasi: Lakukan sanitasi input pengguna pada parameter `language` untuk mencegah karakter traversal seperti `../`, `..\`, `%2e%2e/`, dan

Figure 2 Report of Penetration Testing Result on XYZ Application

4.3 Recommendation

The following are mitigation recommendations based on penetration testing results on both frameworks.

Table 5 Recommendation Mitigation

Vulnerability	CVSS Score	Mitigation
Path Traversal	8.7 (High)	Implement strict input validation and sanitization for file paths, restrict access to allowed directories only, and apply proper server-side access controls.
Stored Cross-Site Scripting	8.6 (High)	Apply output encoding, validate and sanitize all user inputs, and implement Content Security Policy (CSP) to prevent malicious script execution.
CSRF Attack	6.9 (Medium)	Implement anti-CSRF tokens for state-changing requests and ensure proper validation of user sessions.
Reflected Cross-Site Scripting	0 (None)	No mitigation required due to minimal impact. however, input validation and output encoding are recommended as preventive measures.

4.4 Comparison between OWASP WSTG and NIST SP 800-115

This research compares OWASP WSTG and NIST SP 800-115 as two independent penetration testing frameworks used to assess the security of a web-based application. Although both frameworks aim to identify security weaknesses, they differ in primary focus, testing approach, level of technical detail, depth and tester skill dependancy.

Table 6 Comparison between OWASP WSTG and NIST SP 800-115

Aspect	OWASP WSTG	NIST SP 800-115
Primary Focus	Web application security testing	General penetration testing methodology
Testing Approach	Module based testing	Phase or stage based testing
Level of Technical Detail	High, includes specific test cases	Only provides general guidance without specific test case
Depth	Provides more in depth and specific testing guidance	Depends on the results of vulnerability scanning
Tester Skill Dependency	Higher, due to complexity of testing guide	Lower, easier to understand and implement

Several differences can be seen between OWASP WSTG and NIST SP 800-115. OWASP WSTG mainly focuses on web application security testing and uses a module-based testing approach. It provides detailed technical guidance and specific test cases, allowing testers to perform deeper and more detailed testing on web application components. On the other hand, NIST SP 800-115 focuses on a general penetration testing methodology that follows several stages of testing. The framework mostly provides general guidelines without detailed technical test cases, and the depth of testing depends on the results obtained from vulnerability scanning during the discovery stage. Because of its detailed guidance, OWASP WSTG usually requires higher technical understanding from the tester, while NIST SP 800-115 is generally easier to understand and implement.

5. Conclusion

This research evaluated the security of the XYZ web application using two penetration testing frameworks, OWASP WSTG and NIST SP 800-115. The results show that penetration testing can help identify security weaknesses in web applications. Based on the testing using OWASP WSTG, four vulnerabilities were found, namely Path Traversal, Stored Cross-Site Scripting, Cross-Site Request Forgery (CSRF), and Reflected Cross-Site Scripting. Path Traversal and Stored Cross-Site Scripting were categorized as high risk, CSRF was categorized as medium risk, while Reflected Cross-Site Scripting had minimal impact. Testing using NIST SP 800-115 was conducted through the stages of planning, discovery, attack, and reporting. The results confirmed that Path Traversal and the absence of anti-CSRF tokens were real vulnerabilities, while some findings from the scanning process, such as SQL Injection and backup file disclosure, were identified as false positives. The comparison between the two frameworks shows that OWASP WSTG provides more detailed guidance for testing web application vulnerabilities, while NIST SP 800-115 focuses more on the penetration testing process and testing stages. Overall, the findings indicate that the XYZ application still has several security weaknesses, and improvements such as better input validation, output encoding, and the implementation of anti-CSRF tokens are needed to improve the security of the system.

References

[1] Aslan, Ö.; Aktu ˘g, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. Electronics 2023, 12, 1333.

[2] Telkom. (2023). *TANTANGAN DAN ANCAMAN KEAMANAN SIBER DI ERA DIGITAL*. Telkom Digital Soution. https://www.telkomdigitalsolution.com/storage/file/magazine/bAx0ClFHbatV179uDK3GQhJeUkJDfjslxaqHEc3Z.pdf

[3] BSSN. (2025). *LANSKAP KEAMANAN SIBER INDONESIA 2024*. https://www.bssn.go.id/wp-content/uploads/2025/02/LANSKAP-KEAMANAN-SIBER-2024-1.pdf

[4] EC-Council. (2018). *CEH V10 EC-Council Certified Ethical Hacker*. IPSpecialist LTD.

[5] Al Zulfi, F. A., & Suyatno, D. F. (2023). Pengujian Fungsionalitas dan Celah Keamanan Website Kampoeng Sinaoe Menggunakan Equivalence Partition, Boundary Value Analysis, Fuzzing, dan Penetration Testing. *Journal of Emerging Information Systems and Business Intelligence (JEISBI)*, *4*(3), 139–146.

[6] Fefbi S, Mochamad A, Pengujian Celah Keamanan Input Validation Pada Aplikasi Website Menggunakan Framework OWASP, Jurnal Penelitian Ilmu Komputer. 2023; 1(4):50-55.

[7] Naikson F, Reinhard T, Indra M. Analisis dan Implementasi Secure Code Pada Pengembangan Sistem Keamanan Website fikom-methodist.com Menggunakan Penetration Testing Dan Owasp ZAP. Jurnal TIMES. 2023; 12(1): 28-39.

[8] Miftakhul M, Fithrotuz Z, Rizqiyah H, Sonhaji A, Pengujian Black Box Testing Pada Sistem Website Pemesanan Online Toko Ayam Krispy, Jurnal Media Akademik (JMA), 2025; 3(5): XX.

[9] I. K. Putrawan et al., “Pengukuran Efisiensi Qr Code pada Sistem Presensi Karyawan The Seminyak Beach Resort and Spa,” Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI), vol. 8, no. 1, Feb. 2025, doi: 10.32672/jnkti.v8i1.8272.

[10] The OWASP Foundation. *OWASP Web Security Testing Guide (WSTG)*. Available online: https://owasp.org/www-project-web-security-testing-guide/, diakses tanggal 20 Juli 2025

[11] Rafeli, A. I., Seta, H. B., & Widi, I. W. (2022). Pengujian Celah Keamanan Menggunakan Metode OWASP Web Security Testing Guide (WSTG) pada Website XYZ. Informatik : Jurnal Ilmu Komputer, 18(2), 97–103. https://doi.org/10.52958/IFTK.V18I2.4632

[12] Scarfone, K., Souppaya, M., Cody, A., & Orebaugh, A. (2008). Special Publication 800-115 Technical Guide to Information Security Testing and Assessment Recommendations of the National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-115

[13] Fitriana, D. N., Mas’udia, P. E., & Kusumawardani, M. (2023). NIST SP 800-115 Framework Implementation using Black Box Method on Security Gaps Testing on JTD Polinema’s Official Website. Journal of Telecommunication Network (Jurnal Jaringan Telekomunikasi), 13(4).

[14] FIRST Organization. (2023). Common Vulnerability Scoring System. https://www.first.org/cvss/v4-0/, diakses tanggal 20 Juli 2025

[15] Sayed, E. S., Mohammed, F. N., Khosraw, S. Identifying and Mitigating Web Application Vulnerabilities: A Comparative Study of Countermeasures and Tools. International Journal Software Engineering and Computer Science (IJSECS). 2024; 4(3): 1109-1127.