JAIIT (Journal of Advances in Information and Industrial Technology)
Vol. 6, No. 2 Nov 2024, ISSN 2716-1935, E-ISSN 2716-1927

Evaluation of Information Security at the XYZ Foundation Using OWASP Top 10 2021 Framework

Mustafa Kamal[*1)], Muhammad Nasrullah[2)], Rully Rosadi[3)], Yuvens Anggito[4)], Sujan Chandra Roy[5)]

[1, 3, 4)] Department of Information Technology, Telkom University, Surabaya, Indonesia
[2)] Department of Information Systems, Telkom University, Surabaya, Indonesia
[5)] Department of Computer Science Engineering, Bangabandhu Sheikh Mujibur Rahman University, Kishoreganj, Bangladesh
[1*]mustafakamale@gmail.com, [2]emnasrul@telkomuniversity.ac.id, [3]rosadirully5@gmail.com, [4]yuvens112@gmail.com, [5]sujan.cse@iubat.edu

ARTICLE INFO

Article history:
Received 30 October 2023
Revised 07 October 2024
Accepted 11 October 2024
Available online 21 November 2024

Keywords:
Cybercrime
Information security
Non-governmental organization
Top 10 OWASP 2021
Vulnerability test

ABSTRACT

More than three billion users use the Internet in various economic, commercial, cultural, social, and governmental fields. The XYZ Foundation is a non-governmental organization with more than one hundred thousand donors, and its partners also use the Internet for their operations, including online zakat and alms transactions. Increasing the use of online transactions also increases the opportunities for cybercrime. Vulnerability testing is required to observe information security in the XYZ foundation's online zakat and alms transactions. The OWASP vulnerability testing method is one of the most widely used. This study uses the top 10 OWASP 2021 vulnerability tests on the XYZ foundation's online zakat and alms transaction websites. This research shows that one aspect has a medium risk, one is low, and eight are shallow. Based on these results, the weak aspects of online zakat and alms transactions in the XYZ foundation must be immediately improved.

This is an open access article under the CC BY-SA license.

1. Introduction

The XYZ Foundation is a non-governmental organization that manages the collection and distribution of zakat and alms in Indonesia [1]. The XYZ Foundation has more than one hundred thousand donors and partners spread across Indonesia. It extensively uses information technology in its activities with donors and partners and uses an e-zakat service to facilitate transactions with its donors [2]. An organization specializing in cybersecurity can attain a high level of standing and numerous accomplishments, as these successes are directly linked to the company's ability to safeguard personal and customer data from competitors and malicious actors such as hackers and cybercriminals. To ensure the safety of its assets and clients, a company or organization must prioritize robust security measures and continuously improve its defenses [3].

Information security has become a top priority for organizations due to the rising threats in the digital landscape. The OWASP Top 10 framework, particularly the 2021 version, provides updated insights into the most critical security risks that web applications face. By researching information security evaluations based on the OWASP Top 10, organizations can align with Global Security Standards. The OWASP Top 10 is widely recognized as an industry standard for web security, offering organizations a reliable benchmark for improving their defenses. Organizations can also address Modern Security Challenges: The 2021 update reflects emerging vulnerabilities like Server-Side Request Forgery (SSRF), which has been increasingly exploited. Assessing organizations using this updated framework protects them against old and new threats. Also, organizations can tailor security improvement by conducting evaluations using the OWASP Top 10. This allows organizations to

pinpoint and address specific vulnerabilities through targeted interventions, ultimately safeguarding sensitive data and maintaining trust [4].

Figure 1. CIA Triad of information Security

The CIA trinity of secrecy, integrity, and availability is fundamental to information security. Information security centers on three fundamental security aspects. The CIA triad constitutes the fundamental information security framework, with alternative models exhibiting aspects inherent to the CIA triad [5]. The traditional CIA triad poses a significant inquiry: it regards all three security traits as equivalent when, in actuality, interdependence exists among them. Dependency can exist even in its absence [6].

Web application vulnerability assessment solutions are available as open-source and private software. They can automate the vulnerability testing process and function across many operating systems [7]. Vulnerability assessments may be conducted either manually or automatically. Most vulnerability assessment systems can autonomously scan and evaluate vulnerabilities, generating comprehensive reports to mitigate exploitable weaknesses against cyber threats. Moreover, these technologies gather comprehensive vulnerability data from online databases, enhancing subsequent vulnerability assessment processes. Most web application vulnerability assessment solutions gather data from within the web application to find exploitable flaws. Upon the tool's completion of all activities, the identical procedures are reiterated to enhance the precision of randomizing attacks on web applications [8].

Previous research conducted with Security Analysis on Websites Belonging to the Health Service Districts in Indonesia Based on the Open Web Application Security Project (OWASP) Top 10 2021. Of the eleven vulnerabilities discovered, nine vulnerabilities were successfully exploited by following the WSTG 4.2 guidelines. By using the OWASP Risk Assessment Calculator, four vulnerabilities have a medium severity level, and five others have a low severity level. Based on these findings, recommendations for improving each vulnerability are given [4].

Other research was conducted using the Security Evaluation of the Insurance Portal Agency Information System based on ISO/IEC 25010 Quality Standard Utilizing OWASP ZAP. The objective is to suggest security enhancements and draw comparisons between the two. The testing of ISO 25010 is segmented into several phases: identifying security characteristics, establishing measurements, assessing security on two application portals, conducting evaluations and comparisons, and providing recommendations. Testing revealed that the older portal outperforms the newer version in confidentiality and integrity despite the latter's advanced technology. However, the new portal excels in authentication, and both applications demonstrate high scores in accountability. Given the absence of digital signatures, both portals must enhance the non-repudiation characteristic. Based on the analysis, additional recommendations are made to improve the security of both applications [9].

 DOI: https://doi.org/10.52435/jaiit.v6i2.397

The proxy is a crucial component of web vulnerability assessment tools, enabling access to web applications. The Burp Suite, created by PortSwigger Co., Ltd., is among the most prevalent proprietary tools for web vulnerability assessment. Acunetix is a proprietary vulnerability assessment tool that uses powerful crawling techniques to identify vulnerabilities in online applications. Conversely, OWASP ZAP is an open-source web vulnerability assessment tool created by the Open Web Application Security Foundation (OWASP), a non-profit organization engaged in various web security enhancement initiatives. OWASP is renowned for its top 10 vulnerability rankings. SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) are prominent web vulnerabilities listed in the OWASP Top 10 of 2021 [10].

2. Methods

Analyzing the web application under test is important for determining the testing approach. The user environment, development platform, version used for deployment, possible weaknesses or access points in the target web application, appropriate test settings, and tool selection are important factors that must be identified and derived from.

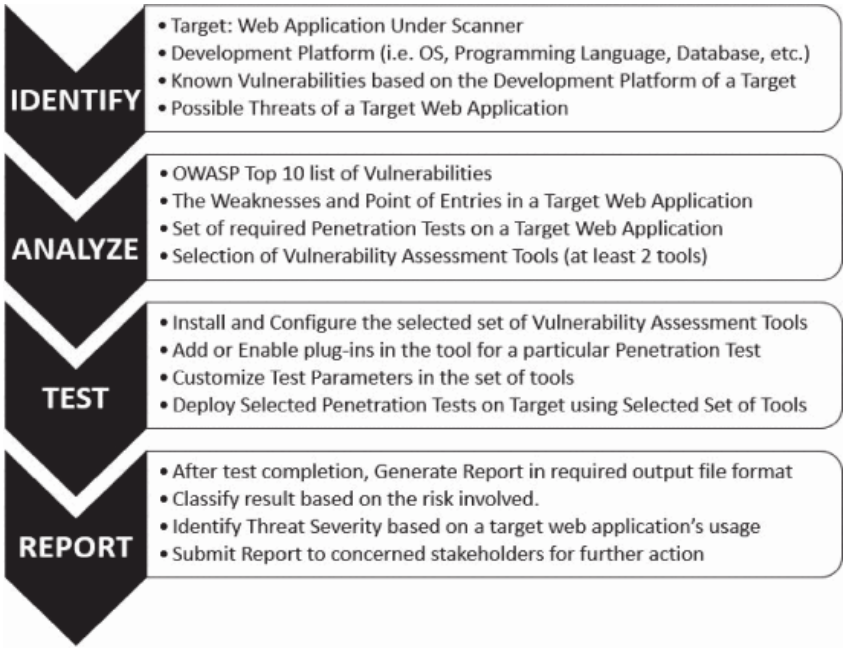

Figure 2. Research methodology

Figure 2 shows the recommended testing methodology to perform vulnerability assessment testing on the target web application in four phases: identifying, analyzing, testing, and reporting. Identifying: The first stage involves identifying all components of the system or software to be tested. They include applications, servers, networks, and system configurations [11]. Identification also includes collecting information about the technology used in the system, such as the software version, operating system, and hardware. Analysis: After identification, the next step is to carry out an analysis of the information that has been found. This includes the evaluation of potential vulnerabilities based on the information gathered. The analysis also involves determining the level of risk associated with each discovered vulnerability, such as the impact and probability of exploitation. Testing: The testing phase involves concrete actions to evaluate whether vulnerabilities identified in the analysis phase can be exploited. During testing, researchers try to find ways to exploit the vulnerability, try penetration techniques, and perform a series of test scenarios to test whether the system or software can be hacked. It is important to conduct testing carefully and ethically to ensure that discovered vulnerabilities are not used for malicious purposes. Reporting: After the identification, analysis, and testing stages were completed, the final step was to create a detailed vulnerability report. Vulnerability reports typically contain information on discovered vulnerabilities, including a description of the vulnerability, risk level, recommended

remediation steps, and evidence supporting the findings. This report will be submitted to the owner of the system or software under test so that they can take necessary actions to fix the vulnerability and improve the security of their system [12]. During this process, it is essential to maintain the integrity and confidentiality of the data being tested and to operate by the ethical and legal standards applicable to information security.

The Open Web Application Security Project (OWASP) is an organization that focuses on software security. It released the "OWASP Top 10" list, which includes the 10 most critical and common software security vulnerabilities. This list has been periodically revised to reflect changes in software security threats [13]. There have been several years of OWASP Top 10 releases. (1) OWASP Top 10 2004. This is the first release of the OWASP Top 10, which provides insight into 10 common vulnerabilities encountered in web applications. (2) OWASP Top 10, 2007. This release is an improvement on the previous list and reflects changes in software security threats. (3) OWASP Top 10 2010. 2010 saw another revision of the OWASP Top 10, with further changes in emphasis on the most frequently exploited vulnerability. (4) OWASP Top 10, 2013. This release includes further changes in the vulnerability rankings and descriptions. It also covers new and relevant security vulnerabilities. (5) The OWASP Top 10, 2017 [14][15]. This release reflects the changes in web security threats and provides updates on security vulnerabilities. It is worth noting that OWASP continually updates its Top 10 list to reflect changes in software security threats. The latest version of the OWASP Top 10 is currently the OWASP Top 10 2021 [16].

The following are the most important parameters that can be used as an OWASP Top 10 2021 guide to identify and then select appropriate vulnerability assessment tools. These settings can be configured in the tools using a plugin to add this specific testing requirement. Based on these selected parameters, we provide a comparative analysis of vulnerability assessment tools. The following are some randomly selected web application vulnerability assessment tools that provide the comparative analysis needed for tool selection.

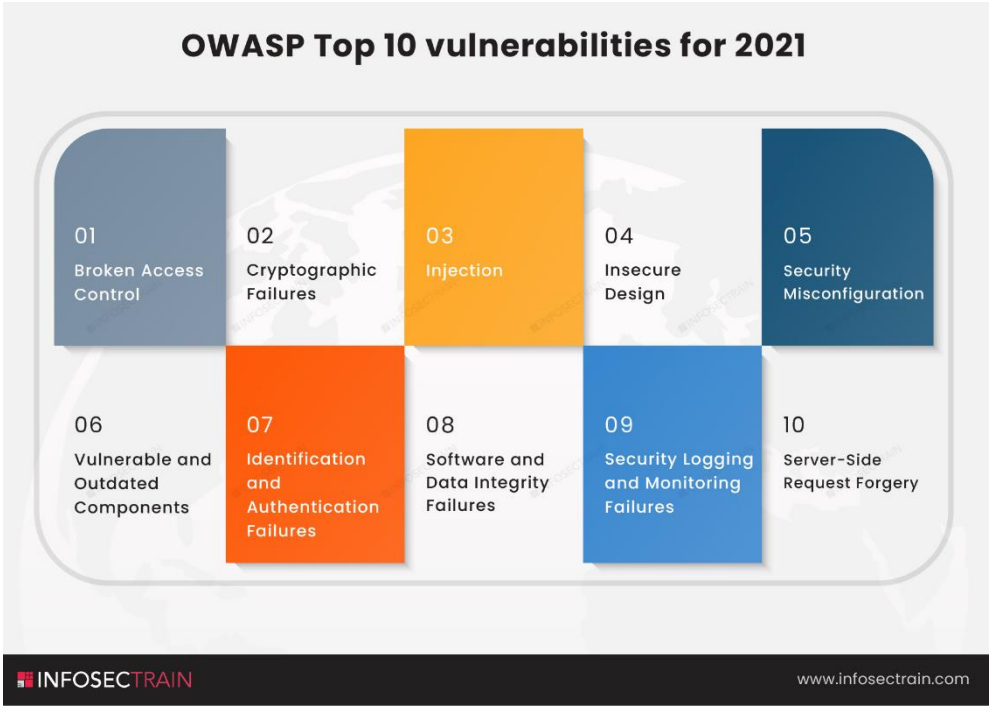

Figure 3. *Top 10 OWASP 2021 Vulnerability Items*

Vulnerability testing was conducted on https://yayasanxyz.org utilizing Beagle Security. The objective of the penetration test performed by Beagle Security is to detect any existing vulnerabilities in the application and assess the degree to which it can be compromised by an attacker. Beagle Security

 DOI: https://doi.org/10.52435/jaiit.v6i2.397

automates the penetration testing procedure under the oversight of human specialists. The platform utilizes a systematic methodology, segmented into four steps, to detect all existing vulnerabilities in the application under examination. The processes include vulnerability scanning, vulnerability exploitation and penetration, report creation and manual verification, and final approval. Beagle Security upholds a current repository of the latest vulnerabilities and test cases, guaranteeing the identification and resolution of all potential security flaws in the application.

3. Results and Discussions

Penetration testing of the website revealed multiple security vulnerabilities that may be exploited to compromise the site and its data. These vulnerabilities encompass the absence of the X-Frame-Options header, the lack of a Content Security Policy (CSP) header, cookies established without the 'Secure' flag, the non-implementation of the HTTP Strict Transport Security (HSTS) header, the discovery of a Google Captcha Key, information leakage within Exif data of images, Cross-Origin Resource Sharing (CORS) configured with public access, the omission of the XContent-Type-Options header, the failure to implement Subresource Integrity (SRI), web server fingerprinting, and the disclosure of email addresses. These vulnerabilities can be exploited by hostile parties to access sensitive data or disrupt website operations. It is very advisable that identified vulnerabilities be rectified promptly. Ongoing penetration testing is crucial for ensuring website security. Due to the continual evolution of websites, it is imperative to routinely assess for emerging vulnerabilities and confirm that current ones are sufficiently mitigated. Consistent penetration testing aids in identifying emerging threats and guarantees the website's security and reliability.

Table 1. Vulnerability Risk Results

TOP10 ID	Name of Vulnerability Items	Risk Results
A01	Broken Access Control	Very Low
A02	Cryptographic Failures	Low
A03	Injection	Very Low
A04	Insecure Design	Very Low
A05	Security Misconfiguration	Medium
A06	Vulnerable and Outdated Components	Very Low
A07	Identification and Authentication Failures	Very Low
A08	Software and Data Integrity Failures	Very Low
A09	Security Logging and Monitoring Failures	Very Low
A10	Server-Side Request Forgery (SSRF)	Very Low

The research, which was carried out on October 7th, 2023, with the target https://yayasanxyz.org, produced one item with medium-risk status and one item with low-risk status. Meanwhile, eight other items had a very low-risk status. Medium-risk results occur on the security misconfiguration item, and low-risk results occur on the cryptographic failure item. More complete results are presented in Table 1.

The findings indicate that the system possesses a minimal overall risk of being compromised by most OWASP Top 10 vulnerabilities. Two vulnerabilities were cryptographic Failures (Low) and Security Misconfiguration (Medium) pose any noticeable risk. Cryptographic Failures mean that the low-risk rating suggests that cryptographic protocols need a closer review to ensure the system uses up-to-date, robust data transmission and storage encryption methods. Security Misconfiguration means The medium-risk rating here indicates that further action is necessary to review and tighten security settings, patch any misconfigurations, and ensure security policies are correctly implemented and maintained.

These findings indicate that the system is generally secure, but the areas of cryptographic controls and security configuration need additional attention to reduce potential risks further.

Table 2. Results Problem Details and Recommendations

Problem ID	TOP10 ID	Problem Details	Recommendations
4.1.1	A05	X-Frame-Options header not implemented	1. Determine the application or website that requires security measures. 2. Incorporate the subsequent code into the web page header to inhibit the page from loading within an iFrame: *<meta http-equiv="X-Frame-Options" content="deny">* 3. If the program or website utilizes Apache, incorporate the subsequent code into the htaccess file: *Header always append X-Frame-Options DENY* 4. If the application or website use Nginx, incorporate the subsequent code into the server configuration file: *add_header X-Frame-Options DENY;* 5. Verify the modifications to confirm that the page is not rendered within an iFrame.
4.1.2	A05	Content Security Policy (CSP) header not implemented	1. Define the policy. The Content Security Policy (CSP) is a security measure that helps protect your website from malicious code injections. It is a set of rules that dictate which sources are allowed to load on a web page. 2. Implement the policy. The CSP policy should be implemented in the HTTP response header of the web page. This can be done using the Content-Security-Policy header. 3. Test the policy. Once the policy has been implemented, it is important to test it to ensure that it is working correctly. There are various tools available that can be used to test CSP policies. 4. Monitor and review the policy.
4.1.3	A05	A Composer File Found	1. Check if the composer {le contains any sensitive information: Run a grep command to search for sensitive information such as passwords, API keys, etc. *grep -rni --include=composer.json'password|api_key|secret|token'/path/to/project* 2. Check if the composer file contains any vulnerable packages: Run a composer show command to list all the installed packages and their versions. 3. Update the vulnerable packages to the latest version: Run a composer update command to update the packages to the latest version. 4. Remove unnecessary packages: Run a composer remove command to remove any unnecessary packages. 5. Add a *.gitignore* file to the project root: Create a *.gitignore* file to prevent any sensitive information from being committed to the repository.
4.1.4	A05	Cookie set without 'Secure' flag	1. Verify that the Secure flag is enabled when configuring cookies. The Secure setting directs the browser to transmit the cookie exclusively over HTTPS connections, hence complicating an attacker's ability to intercept the cookie. 2. Consider using the *HttpOnly* flag when setting cookies. The *HttpOnly* flag instructs the browser to not allow JavaScript

 DOI: https://doi.org/10.52435/jaiit.v6i2.397

Problem ID	TOP10 ID	Problem Details	Recommendations
			to access the cookie. This prevents attackers from stealing the cookie using Cross-Site Scripting (XSS) attacks. 3. Consider using the *SameSite* flag when setting cookies. The SameSite flag instructs the browser to not send the cookie with cross-site requests. This prevents attackers from stealing the cookie using Cross-Site Request Forgery (CSRF) attacks. 4. Consider using the Expires flag when setting cookies. The Expires flag instructs the browser to only send the cookie if it is within a certain time frame. This prevents attackers from stealing the cookie if it has expired. 5. Consider using a secure cookie store. A secure cookie store encrypts the cookie data, making it more difficult for an attacker to access.
4.1.5	A05	HTTP Strict Transport Security (HSTS) header not implemented	1. Implement HSTS (HTTP Strict Transport Security) on your web server: *Header set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"*. 2. Add the HSTS header to the server con{guration file. Depending on the web server, this may be done in the Apache con{guration file (*httpd.conf*) or in the site configuration file (e.g. *.htaccess*). 3. If the site is using HTTPS, add the HSTS header to the HTTPS virtual host configuration. 4. If the site is using a content delivery network (CDN) such as Cloudfare, add the HSTS header to the CDN configuration. 5. If the site is using a web application firewall (WAF) such as ModSecurity, add the HSTS header to the WAF configuration. 6. Test the implementation of HSTS by using tools such as Qualys SSL Labs or Security Headers.
4.1.6	A02	Google Captcha Key found	1. Ensure that the Google Captcha Key is not stored in the code. 2. Validate the Captcha Key. 3. Ensure that the Captcha Key is not exposed to the public. 4. Implement a secure Captcha Key generation. 5. Implement a secure storage for the Captcha Key. 6. Implement a secure Captcha verification. 7. Implement an access control mechanism.
4.1.7	A02	Information leakage in Exif data of images	1. Disable Exif data on server side. 2. Disable Exif data on client side. 3. Validate and sanitize user-uploaded images.
4.1.8	A05	Cross origin Resource Sharing Implemented with Public Access	1. Ensure that all CORS requests are explicitly whitelisted in the server configuration. 2. Ensure that the Access-Control-Allow-Credentials header is not set to true in the server configuration. 3. Ensure that the Access-Control-Allow-Methods header is set to an appropriate value in the server configuration. 4. Ensure that the Access-Control-Allow-Headers header is set to an appropriate value in the server configuration. 5. Ensure that the Access-Control-Max-Age header is set to an appropriate value in the server configuration.

Problem ID	TOP10 ID	Problem Details	Recommendations
			6. Ensure that all CORS requests are sent with the Origin header set to an appropriate value. 7. Ensure that all CORS requests are sent with the X-Requested-With header set to an appropriate value.
4.1.9	A05	X-Content-Type-Options header not implemented	1. Incorporate the X-Content-Type-Options header into your web application. 2. Configure your web server to include the X-Content-Type-Options header. 3. Test the configuration.
4.1.10	A05	Sub-resource Integrity (SRI) is not implemented and external scripts are not loaded securely	1. Configure Sub-resource Integrity (SRI). SRI is a security mechanism that allows browsers to confirm that resources retrieved (such as from a CDN) are delivered without unauthorized alterations. 2. Utilize HTTPS for Loading External Scripts. Alongside employing SRI to authenticate the integrity of external scripts, it is imperative to guarantee that the scripts are loaded safely via HTTPS. This guarantees that the scripts remain unaltered throughout transmission.
4.1.11	A05	Fingerprinting Web Server	1. Disable Server Header from sending the version number in the response header. 2. Disable Directory Listings in the response. 3. Disable File Extensions in the response header. 4. Disable Error Messages in the response.
4.1.12	A02	Email Address Disclosure	1. Validate user input. To ensure that only valid email addresses are accepted, all user input should be validated using a regular expression to ensure that only valid email addresses are accepted. 2. Sanitize user input. User input should be sanitized to prevent malicious scripts from being injected into the application. This can be done by using a library such as *HTMLPurifier* to remove any malicious HTML or JavaScript code. 3. Encrypt user data. To protect user data from being disclosed, it should be encrypted using a secure algorithm such as AES-256. 4. Mask user data. To prevent user data from being disclosed, it should be masked to hide sensitive information. This can be done by using a library such as PHP Mask to mask the user data.

In Table 2, it can be observed that the security misconfiguration item with code A05 has worrying risk results. This security misconfiguration consists of several detailed problems, namely the X-Frame-Options header not implemented, Content Security Policy (CSP) header not implemented, Composer File Found, Cookie set without 'Secure' flag, HTTP Strict Transport Security (HSTS) header not implemented, cross-origin Resource Sharing Implemented with Public Access, and others. Cryptographic failure items with code A02 also have risks that need to be corrected. In more detail, cryptographic failure consists of Google Captcha Key, information leakage in the Exif data of images, and Email Address Disclosure. More details regarding the problem can be seen in Table 2.

The primary A05 risk ID with 4.1.1 indicates that the X-Frame-Options header is absent from the webpage. In the absence of the X Frame-Options header, the browser is unable to determine whether to render the page within a <frame> or <iframe>, hence preventing the site from guaranteeing that its

 DOI: https://doi.org/10.52435/jaiit.v6i2.397

contents are not embedded in external sites. This vulnerability results in numerous attacks, such as Clickjacking. Recommendations are shown in Table 2.

4. Conclusion

This research shows that with a vulnerability assessment, several weaknesses in social institution websites can be identified, namely, security misconfiguration items with code A05 and cryptographic failure items with code A02. The latest version of the OWASP Top 10 2021 also has the advantage of being up-to-date with the gaps that are most often found in most information technology applications. The latest version of the OWASP Top 10 2021 can also provide recommendations that are more updated than most information technology applications.

This research produced several recommendations, namely, the need to repair security misconfiguration items with code A05 on nine detailed problems and cryptographic failure items with code A02 on three detailed problems. By addressing these 12 problems, it is hoped that information security risks can be better controlled. Online zakat and alms transactions in the XYZ foundation are also more secure.

This study has limitations that can be developed in the future. For example, monitoring the repair of further information security gaps in the XYZ foundation. Thus, after the repairs, the vulnerability test can be repeated until the test results are better. All items in the OWASP Top 10 2021 had very low-risk status results. It is also necessary to develop information security testing using a vulnerability test framework other than OWASP TOP 10 2021.

Acknowledgement

In this research activity, we received significant help from various parties. We thank the Lembaga Penelitian Pengabdian Masyarakat (LPPM) for their support in this research activity, the executive director of the XYZ Foundation, who has given permission to the management and employees of the XYZ Foundation so that this activity can run smoothly, and the good parties who directly or indirectly contribute to research activities.

Reference

[1] N. Ardiani, "THE EFFICIENCY OF ZAKAT COLLECTION AND DISTRIBUTION: EVIDENCE FROM DATA ENVELOPMENT ANALYSIS," *al-Uqud J. Islam. Econ.*, vol. 3, no. 1 SE-Articles, pp. 54–69, Jan. 2019, doi: 10.26740/al-uqud.v3n1.p54-69.

[2] A. R. Hakim, A. S. Mulazid, and E. Meiria, "E-Zakat: Redesign the collection and distribution of Zakat," *KnE Soc. Sci.*, pp. 433–452, 2018.

[3] T. Taya *et al.*, "An Automated Vulnerability Assessment Approach for WebAPI that Considers Requests and Responses," in *2022 24th International Conference on Advanced Communication Technology (ICACT)*, 2022, pp. 423–430. doi: 10.23919/ICACT53585.2022.9728941.

[4] A. Choiriyah and N. Qomariasih, "Security Analysis on Websites Belonging to the Health Service Districts in Indonesia Based on the Open Web Application Security Project (OWASP) Top 10 2021," in *2023 International Conference on Information Technology and Computing (ICITCOM)*, 2023, pp. 267–272.

[5] B. Lundgren and N. Möller, "Defining information security," *Sci. Eng. Ethics*, vol. 25, pp. 419–441, 2019.

[6] M. Snehi and A. Bhandari, "Security management in SDN using fog computing: A survey," in *Strategies for e-Service, e-Governance, and Cybersecurity*, Apple Academic Press, 2021, pp. 117–126.

[7] Y. Diogenes and E. Ozkaya, *Cybersecurity-attack and defense strategies: Infrastructure security with red team and blue team tactics*. Packt Publishing Ltd, 2018.

[8] M. Idris, I. Syarif, and I. Winarno, "Development of vulnerable web application based on OWASP API security risks," in *2021 International Electronics Symposium (IES)*, 2021, pp. 190–

194.

[9] M. D. Fadilah, "Evaluasi Keamanan Sistem Informasi Portal Agen Asuransi Berdasarkan Kualitas Standar ISO/IEC 25010 Menggunakan OWASP ZAP." Institut Teknologi Sepuluh Nopember, 2023.

[10] M. Aljabri *et al.*, "Testing and exploiting tools to improve owasp top ten security vulnerabilities detection," in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022, pp. 797–803.

[11] F. Mateo Tudela, J.-R. Bermejo Higuera, J. Bermejo Higuera, J.-A. Sicilia Montalvo, and M. I. Argyros, "On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications," *Appl. Sci.*, vol. 10, no. 24, p. 9119, 2020.

[12] S. K. Lala, A. Kumar, and T. Subbulakshmi, "Secure web development using owasp guidelines," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 323–332.

[13] S. Alazmi and D. C. De Leon, "A systematic literature review on the characteristics and effectiveness of web application vulnerability scanners," *IEEE Access*, vol. 10, pp. 33200–33219, 2022.

[14] S. Rafique, M. Humayun, Z. Gul, A. Abbas, and H. Javed, "Systematic review of web application security vulnerabilities detection methods," *J. Comput. Commun.*, vol. 3, no. 09, p. 28, 2015.

[15] M. Aydos, Ç. Aldan, E. Coşkun, and A. Soydan, "Security testing of web applications: A systematic mapping of the literature," *J. King Saud Univ. Inf. Sci.*, vol. 34, no. 9, pp. 6775–6792, 2022.

[16] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, "A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions," *Appl. Sci.*, vol. 12, no. 8, p. 4077, 2022.

 DOI: https://doi.org/10.52435/jaiit.v6i2.397