Security Evaluation of Insurance Portal Agency Information System Based on ISO/IEC 25010 Quality Standard Utilizing OWASP ZAP

Muhammad Darmawan Fadilah   
*Faculty of Intelligent Electrical and Informatics Technology   
Teknologi Sepuluh Nopember*Surabaya, Indonesia  
[6025221046@mhs.its.ac.id](mailto:6025221046@mhs.its.ac.id)

Siti Rochimah  
*Faculty of Intelligent Electrical and Informatics Technology*  
*Institut Teknologi Sepuluh Nopember*Surabaya, Indonesia  
[siti@its-sby.edu](mailto:siti@its-sby.edu)

*Abstract*—Information System, like insurance agency portals, bring substantial benefits for user and the company but also pose security risks due to the sensitive data they handle. Addressing these risks through regular security evaluations and enhancements is crucial to prevent potential financial, reputational, and legal consequences. This research focuses on security measurement in two versions of the agency portal: the old and the new. Both serving the same purpose but differing in technology and infrastructure. Employing the ISO 25010 security standard, tools like OWASP ZAP are used alongside manual testing to assess each ISO 25010 characteristic and subsequently evaluate both portals. 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. Both portals need to enhance the non-repudiation characteristic, given the absence of digital signatures. Based on the analysis, additional recommendations are made to improve the security of both applications.

Keywords—Security Measurement, ISO 25010, Information System, OWASP ZAP, Comparison

# Introduction

Insurance portal agency website facilitate communication and engagement between users and corporations, offering services ranging from agent registration and product information to presenting sales and achievement reports. These information tools help businesses work better, compete effectively, and quickly adapt to changes [1]. Although they provide numerous benefits, web application usage also brings an escalating risk of security breaches. Given the surge in website utilization and the fact that the insurance agency portal stores personal client data and financial account information, It's essential for organizations to protect their data and keep sensitive information private. This means stopping unauthorized access and changes to data. If there's a breach, it could result in money loss, harm to the organization's reputation, and legal problems [2].

This research focuses on a company that uses two distinct agent portal websites; an older website operational since 2010 and a new one introduced in 2019. The older application, crafted with outdated Java 1.6 technology and constrained by the JBoss 5.1 framework, became limited post-2018 due to end-of-support challenges, preventing further upgrades. To address this and potential security threats, a new application was developed using Java 1.8 and the Spring Boot framework, promising enhanced security. Notably, both applications yield identical results despite their distinct developmental backgrounds. While the new application has been launched, not all modules from the old application have been transferred. As a result, the company still runs both websites at the same time. It's very important to check how secure both sites are because of possible security threats.

With the rise of internet security threats, including malware attacks, exploitation, database injections, and more. Reports indicate that approximately 90% of internet crimes in 2016 were executed via web application attacks, with database injections leading at 47.06% [3]. For a comprehensive assessment, security levels should be evaluated against recognized international measurement standards, such as the McCall model, Boehm, FURPS, Dromey, ISO/IEC 9126, and ISO/IEC 25010 [1]. Among these, ISO/IEC 25010 stands out [1], [4] for its comprehensive measurement aspects and is a refinement of ISO/IEC 9126. Additionally, its adaptability and generality make it suitable for measuring the quality of specific application domains. Adopting ISO/IEC 25010 as a software quality evaluation standard with a focus on security aspects, this research will test the security levels of both agent portal web applications and proceed with the comparison phase.

Effective evaluation is contingent upon accurate metrics, making the ISO 25010 standard a crucial framework for web security assessment. However, benchmarks alone cannot achieve their potential without the support of an adept measurement tool. In this research have opted for the Open Web Application Security Project Zed Attack Proxy (OWASP ZAP) as our primary tool. The selection of OWASP ZAP is based on its unique advantages, from its ability to conduct comprehensive security testing using the WAST technique to its global recognition supported by a devoted team. Furthermore, the extensive corporate support for its free security testing features amplifies its reliability [5]. By utilizing OWASP ZAP, this study seeks to assess alignment with the ISO 25010 standard, offering insights into the security strength of digital platforms.

This research harness the capabilities of the renowned security testing tool, OWASP ZAP, mapping its findings to the characteristics defined by the ISO/IEC 25010 standard to evaluate the security differences between the previous and the updated agent portal applications. Through this comprehensive evaluation, we aim to determine the degree of security enhancements in the updated version. The Evaluation can provide guidance for developers and application managers, helping them pinpoint strengths and areas of potential concern, ensuring that the revamped portal excels in terms of security compared to its predecessor.

# Related Work

## ISO/IEC 25010

A framework to measure software security based on the ISO 25010 standard [4]. Their strength lies in providing comprehensive weighting criteria for five security sub-characteristics. However, the framework's limitation is its restricted testing to only the prototype of ITS AIS and not on other applications. Their method involved the development of the framework based on ISO 25010, evaluated by six experts through questionnaires.

In a more recent study, evaluated the security of the PPDB Surabaya 2019 information system [6]. They conducted tests on five security sub-characteristics according to ISO 25010 and provided recommendations based on their assessments. Nonetheless, their research did not include evidence of testing against SQL Injection attacks. This oversight might be due to their reliance on the CodeIgniter framework, which is specifically designed to prevent such attacks and is deemed highly secure. Their methodology incorporated the measurement and evaluation of various sub-characteristics of ISO 25010.

Focused on the security assessment of an ERP system [7]. Their unique contribution was the inclusion of an Archivability test that they suggested should be implemented in ERP systems. However, their research lacked a detailed description of the testing procedure for achieving the solution to specific questions. Their approach centered around posing questions on every ISO 25010 sub-characteristic and Archivability to gauge the security level of the ERP system.

Embarked on measuring the security of the Halodoc system using ISO 25010 [8]. They assigned percentage weights to security priorities based on ISO 25010 sub-characteristics. Their research identified a shortcoming in their confidentiality testing method, which employed blackbox testing for log checks. They noted that log checks should be historical references and shouldn't serve as benchmarks for system quality assessments. Their methodology spanned testing for confidentiality, integrity, and authentication using blackbox testing.

Compared the three applications - KeePass, KeePassXC, and Password Safe - using the ISO/IEC 25010 standard [9]. They found certain limitations in their study, particularly the inability to test some characteristics due to constraints like the absence of an application log.

Investigated the Silampari Smart City Of Lubuklinggau using ISO/EIC 25010 [10]. They allocated percentage priorities to each ISO 25010 sub-characteristic utilizing the AHP method. Their study identified certain security issues, including six unresolved access rights issues, yet deemed the system secure.

Assessed the Academic Management Android application using ISO/EIC 25010 [11]. They assigned scores ranging from 1 to 5 for the tested sub-characteristics, including security, non-repudiation, and integrity. Their research faced criticism for not providing detailed explanations behind the scores or supporting evidence for certain claims.

The use of the ISO 25010 standard has been widely explored in recent literature. However, none of these studies have compared two applications as our research intends to do. In this study, we aim to improve upon existing methods for implementing all ISO 25010 characteristics comprehensively and enhancing the deficiencies found in previous research.

## OWASP ZAP

The Zed Attack Proxy (ZAP) has emerged as a notable vulnerability scanner in recent literature due to its open-source development and automated security testing features. Operating similarly to the Brup Suite, ZAP serves as a proxy, allowing users to manage all web traffic, including HTTPS [12]. When utilized as a proxy, users can modify the transiting traffic. Additionally, ZAP offers daemon mode operation via its REST API and incorporates features Active/Passive Scanners, Spider Scan, and Report Generation [13].

A comparative study by Parvez, Zavarsky, and Khoury [14] evaluated three web vulnerability scanners, namely, Acunetix WVS, IBM AppScan, and OWASP ZAP. Their assessments were made on a specifically designed vulnerable web application. The focus of the evaluation was primarily on the identification of two types of web application vulnerabilities: cross-site scripting (XSS) and SQL injection. The findings from this study indicated that these scanners have shown improvements in detecting second-level XSS and SQL injection vulnerabilities compared to prior research.

Another comparative research by Makino and V. Klyuev [15] assessed two open-source web vulnerability scanners: OWASP ZAP and Skipfish. The evaluations were conducted on two vulnerable web applications, the Web Application Vulnerability Scanner Evaluation Project (WAVSEV) and Damn Vulnerable Web Application (DVWA). The studied vulnerabilities encompassed SQL injection, both stored and reflected cross-site scripting (XSS), local and remote file inclusion, command injection, and cross-site request forgery (CSRF). Their findings elucidated that OWASP ZAP exhibited superior performance compared to Skipfish.

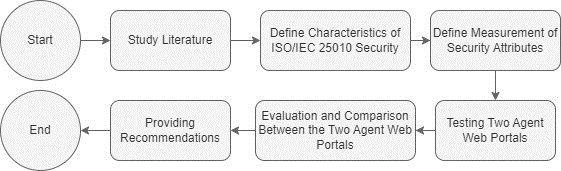
A comparative evaluation by B. Mburano [16] found that OWASP ZAP performed better than the Arachni scanner in categories such as SQL injection (SQLI), cross-site scripting (XSS), and command injection (CMDI). There was a marked performance variation between OWASP ZAP and Arachni across both the OWASP and WAVSEP benchmarks. The evaluation results depicted OWASP ZAP as having higher scores in all four vulnerability categories compared to Arachni. Consequently, the researchers advocate for the adoption of OWASP ZAP as the primary choice in scanner evaluations to enhance application security.

# Propose Method

The measurement standard proposed in this study is ISO/IEC 25010 with measurement tools using OWASP ZAP in analyzing the level of security on an insurance website. This research entails various steps in examining the security features of ISO/IEC 25010 and using OWASP ZAP on both the new and old web portal agents.

## Research Flow

The research flow is used as a comprehensive roadmap in carrying out this research process so that it is in accordance with the initial objectives. The following is a diagram of this research, which can be seen in Figure 1.



In the flowchart presented in Figure 1, the study unfolds in several phases. It begins with a literature review on the ISO/IEC 25010 and OWASP ZAP standards enabling the research to defines the characteristics of ISO/IEC 25010, detailing each feature and then establishing the measurement of security attributes. This involves assessing each characteristic of ISO 25010 using OWASP ZAP features and manual testing. The next stage involves testing two web portal agents, followed by evaluation, comparison, and providing recommendations based on the test outcomes, all of which are elaborated in the experiment section.

## Define Characteristic of ISO/IEC 25010

This research focuses on security using the ISO/IEC 25010 standard. Defined by ISO 25010 [17], security ensures a product or system protects information, granting access based on specific authorization levels. The sub-characteristics can in Table 1 are crucial in assessing system security and safeguarding sensitive information

Table 1. Security Characteristics and Description as Defined by ISO/IEC 25010.

|  |  |
| --- | --- |
| **Characteristics of ISO/IEC 25010** | **Purpose** |
| Confidentiality | Ensuring only authorized individuals access data. |
| Integrity | Preventing unauthorized modifications to programs or data. |
| Non-repudiation | Confirming actions or events occurred, preventing future denial. |
| Accountability | Tracing actions to a specific entity. |
| Authentication | Verifying a subject or resource's claimed identity. |

## Define Measurement of Security Attributes

This research will assess each ISO/IEC 25010 characteristic using all the features provided by OWASP ZAP, as presented in Table 2. Each feature in OWASP ZAP is associated with a risk factor that can be evaluated on a scale ranging from 1 (low) to 3 (high) (Li, 2020), and informational issues are rated as 0. For characteristics that cannot be assessed using OWASP ZAP, manual testing will be conducted.

Quality Metrics Score = (3.1)

The evaluation of security quality is determined by the ratio described in section 3.1, which involves calculating the number of positive conditions applied divided by the total number of conditions.

Table 2. Mapping of OWASP ZAP Plugins to ISO/IEC 25010 Characteristics

|  |  |  |
| --- | --- | --- |
| **Characteristics of ISO/IEC 25010** | **Measurement Plugin OWASP ZAP** | **Risk** |
| Confidentiality | Source Code Disclosure - /WEB-INF folder | High |
| Heartbleed OpenSSL Vulnerability | High |
| Source Code Disclosure - CVE-2012-1823 | High |
| Cross Site Scripting (Reflected) | High |
| Cross Site Scripting (Persistent) - Prime | High |
| Cross Site Scripting (Persistent) - Spider | High |
| Cross Site Scripting (Persistent) | High |
| Cross Site Scripting (DOM Based) | High |
| XML External Entity Attack | High |
| Generic Padding Oracle | High |
| Cloud Metadata Potentially Exposed | High |
| Directory Browsing | High |
| ELMAH Information Leak | Medium |
| Trace.axd Information Leak | Medium |
| .htaccess Information Leak | Medium |
| .env Information Leak | Medium |
| Hidden File Finder | Medium |
| Integrity | Path Traversal | High |
| Remote File Inclusion | High |
| Remote Code Execution - CVE-2012-1823 | High |
| Server Side Include | High |
| SQL Injection | High |
| SQL Injection - MySQL | High |
| SQL Injection - Hypersonic SQL | High |
| SQL Injection - Oracle | High |
| SQL Injection - PostgreSQL | High |
| SQL Injection - SQLite | High |
| SQL Injection - MsSQL | High |
| Server Side Code Injection | High |
| Remote OS Command Injection | High |
| Buffer Overflow | Medium |
| Format String Error | Medium |
| CRLF Injection | Medium |
| Parameter Tampering | Medium |
| XSLT Injection | Medium |
| User Agent Fuzzer | Informational |
| SOAP Action Spoofing | High |
| SOAP XML Injection | High |
| Authenticity | External Redirect | High |
| GET for POST | Medium |

Digital signatures play a fundamental role in ensuring non-repudiation and authenticating actions within digital systems [6], [18]. Their significance in digital transactions has been well-recognized [19]. Systems maintain accountability by implementing comprehensive audit trails and retaining system logs. The necessity to track all user actions is underscored by [19], while [18] highlights the importance of proficient system logging to trace actions back to their sources. Regular access audits and consistent log storage are deemed essential by [4]. Notably, to validate non-repudiation through digital signatures and ensure accountability via audit trails and log retention, manual evaluations are indispensable since tools like OWASP ZAP lack automated verification capabilities in these domains.

## System and Tool Specifications for Penetration Testing

To ensure that users experience no disruptions in the application's performance, this software will undergo testing on a personal computer. This personal computer will be used to test both the old portal and the new portal on the user acceptance test server. In Table 3 are the detailed specifications for the testing environment.

Table 3. System and tool spesification

|  |  |  |
| --- | --- | --- |
| **No** | **Device** | **Specification** |
| 1 | Processor | Intel® Core™ i7-8665U CPU @ 1.90GHz (8 CPUs) |
| 2 | Operating system | Windows 10 |
| 3 | Oracle Virtual Box | Windows |
| 4 | Tool Penetration | OWASP ZAP (2.12.0) |
| 5 | Modem | Biznet |

# Experiment

## Testing on the Old Portal and New Portal

Testing was done using the OWASP ZAP tool, set up to work with new portal agen and old portal agen links. This research used automatic scan in OWASP ZAP which had three main steps: spider scanning, active scanning, and reporting results [20]. First, spider scanning went through the entire web application to find all parts users can access. Then, active scanning sent test requests to these parts to check for security problems. The results were summarized in a report showing any found security issues.

Subsequently, manual testing methods were employed, focusing on the condition of the two test servers and the application's source code. During the evaluation, special attention was given to non-repudiation and authentication.

## Evaluation and Comparation Old Portal and New Portal

1. Confidential

After conducting a confidentiality test using OWASP ZAP as mapped in Table 2, the new portal application was found to have a high vulnerability to Generic Padding Oracle, while no security issues were discovered for other features. As a result, the security score for the new portal is in terms of confidentiality. In contrast, the old portal agent did not have any security findings mapped, resulting in a confidentiality score of 1.

1. Integrity

The integrity was also assessed using OWASP ZAP based on Table 2. It can be observed in table 4 that the old portal agent had a total of 3 high security findings and 1 informational finding, resulting in an integrity score of . Meanwhile, the new portal agent had 5 high-security findings and 1 informational finding, the score for new portal agent integrity score is .

Table 4. Findings of integrity security issues

|  |  |  |
| --- | --- | --- |
| **Measurement Plugin OWASP ZAP** | **New Portal Agent** | **Old Portal Agent** |
| Path Traversal | 0 | 2 |
| SQL Injection - MySQL | 1 | 0 |
| SQL Injection - PostgreSQL | 1 | 0 |
| SQL Injection - SQLite | 10 | 1 |
| SQL Injection - MsSQL | 7 | 0 |
| Remote OS Command Injection | 0 | 1 |
| User Agent Fuzzer | 1097 | 8711 |

1. Non-repudiation

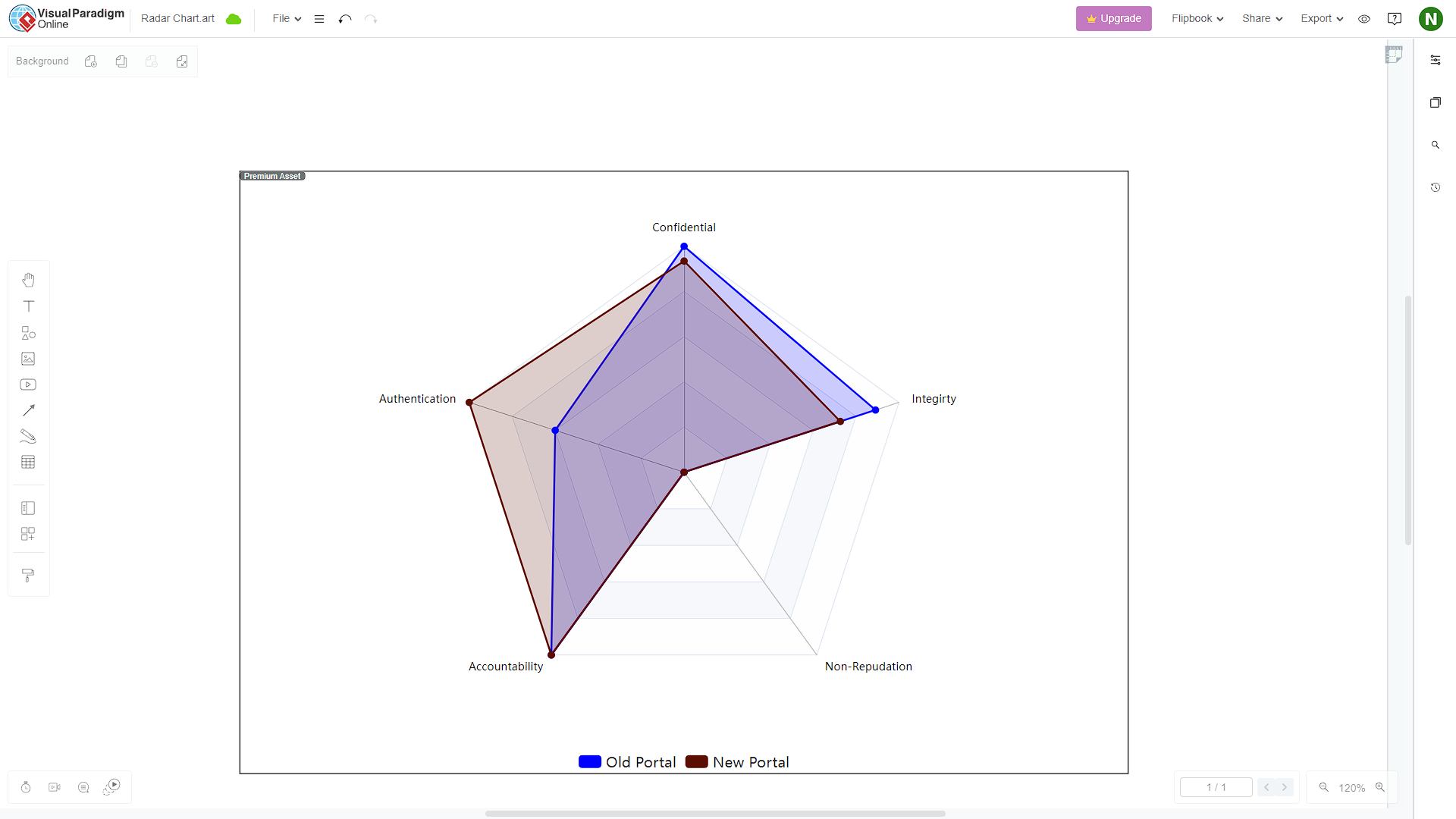
For the non-repudiation characteristic, it was tested by conducting manual testing of the registration module on both the old portal agent and the new portal. In the registration process of both applications, digital signatures have not been implemented yet, resulting in a score of 0 for both. These applications have yet to implement digital signatures to verify the authenticity of agents registering, ensuring that the agent in the ID photo is indeed the one registering. Currently, this process is still manual, involving the submission of agent contract signatures to the central office.

1. Accountability

The new portal and the old portal both have a structured log storage system with daily backups, including data from previous years, which is retained regularly. Additionally, both the new portal and the old portal agent maintain detailed user activity logs and record transaction log data, including who initiated the transaction, when it occurred, and the input values. As a result of these comprehensive logging capabilities, both the old and new portal agents have achieved an accountability rating of 1.

1. Autentication

The final testing was also conducted using OWASP ZAP as mapped in Table 2. Based on this mapping, it can be observed that the new portal agent did not have any security findings related to the mapped issues, resulting in an authentication score of 1. On the other hand, the old portal agent had a moderate security issue with GET for POST, resulting in an authentication score .



After evaluating both the new and old portal agents using the ISO 25010 standard and the OWASP ZAP tool and manual evaluation. Figure 2 show that The old portal scored 1 for confidentiality, better than the new portal's 0.9348. For integrity, the old portal led with 0.8909 compare to new portal's 0.7273. On accountability, the new portal topped with a score of 1, while the old one got 0.6. Both scored 1 for authentication, but both received 0 for non-repudiation.

Based on the analysis and report from OWASP ZAP, several security issues were identified in both the new portal agent and the old portal agent.

In the new portal agent, a confidentiality issue was detected due to a generic padding oracle problem. This issue arises from one of the encrypted parameters, where the padding could be manipulated. Additionally, there was an integrity concern in the new portal where the values of parameters sent to the server could be altered using various SQL injection methods. This manipulation led to successful responses from the server, indicating a security issue.

As for the old portal agent, security vulnerabilities were identified in the integrity aspect. These included a path traversal problem caused by parameters used to access file paths in links. Modifying these parameters to point to other available paths on the Linux server resulted in successful responses from the server, indicating a security flaw. Furthermore, an SQL injection vulnerability was also found in the old portal agent, where parameter values could be modified using SQLite formatting, resulting in successful responses. Lastly, a security concern was discovered in the old portal agent's authentication process related to GET for Post. This occurred when logging into the old portal using GET requests, which exposed usernames and passwords in the URL, contrary to best practices that recommend using POST to secure credentials.

Regarding the Agent Fuzzer for the integrity characteristic, both the old and new portal agent experienced an issue where certain links could be accessed, and despite different users accessing them, the output remained consistent. In this study, the Agent Fuzzer issue was not considered a security problem, as both applications require users to log in initially before using the application.

## Recommendation

Based on the security assessment of two agent portals using OWASP ZAP, suggested enhancements are outlined in Table 5. For manual evaluations, we advise implementing a digital signature to enhance Non-Repudiation.

Table 5. Recommended Mitigations

|  |  |
| --- | --- |
| **Security Issue** | **Recommendation** |
| Generic Padding Oracle | Use current cryptographic libraries.  Avoid revealing error messages that disclose encryption or decryption failures. |
| Path Traversal | Validate input for file or path parameters. Do not use user input directly for file access. |
| SQL Injection | Implement input validation and sanitize user inputs. Use Prepared Statements or secure query technologies like JPA or Hibernate. |
| Remote OS Command Injection | Ensure the application code does not allow direct OS command execution. Employ strict whitelisting or input validation. |
| Get for Post | Change sensitive input methods to POST. |

# Conclusion

From the results, it's clear that while the new portal agent uses newer technology, it has more evident security weaknesses in confidentiality and integrity compared to the old portal. This gap arises from a lack of thorough security testing on the new portal. Implementing the suggested security strategies is vital. Given its Java 1.8 and Spring Boot foundation, the new portal has a better chance for security improvements than the old portal, which relies on the now-unsupported Java 1.6.

Implementing security recommendations from the research requires a thorough evaluation of the application's features to ensure seamless integration. This process, of course, raises considerations of costs and potential risks for the company. Therefore, concrete steps have not yet been taken. For future research, it's recommended to utilize tools such as OWASP ZAP in evaluations to gain a clearer understanding of the effectiveness of the provided recommendations. Even though the new portal agent and the old portal agent were built with different generational technologies, both still prioritize security to support business sustainability. From the test results, it was found that the security levels between the two portals do not have significant differences. For future research, it's suggested to compare other aspects beyond technology. For instance, contrasting an application designed with deep security principles against one more user-need oriented, to gain a clearer picture of the security levels each application possesses.

##### References

[1] S. Budi, W. Gata, M. Noor, S. Pangabean, and C. S. Rahayu, “News Portal Website Measurement Analysis Using Iso/Iec 25010 and Mccall Methods,” *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 1, pp. 273–285, 2022, doi: 10.37385/jaets.v4i1.1094.

[2] S. Duggineni, “Impact of Controls on Data Integrity and Information Systems,” *Sci. Technol.*, no. July, 2023, doi: 10.5923/j.scit.20231302.04.

[3] E. A. Altulaihan, A. Alismail, and M. Frikha, “A Survey on Web Application Penetration Testing,” *Electron.*, vol. 12, no. 5, 2023, doi: 10.3390/electronics12051229.

[4] I. Saptarini, S. Rochimah, and U. L. Yuhana, “Security Quality Measurement Framework for Academic Information System (AIS) Based on ISO/IEC 25010 Quality Model,” *IPTEK J. Proc. Ser.*, vol. 0, no. 2, p. 128, 2017, doi: 10.12962/j23546026.y2017i2.2310.

[5] “Owasp Zap.” https://www.zaproxy.org/docs/ (accessed May 15, 2023).

[6] I. Conference, I. Technology, and E. Engineering, “ICITEE 2022 - Proceedings of the 14th International Conference on Information Technology and Electrical Engineering,” *ICITEE 2022 - Proc. 14th Int. Conf. Inf. Technol. Electr. Eng.*, pp. 120–124, 2022.

[7] E. Peters and G. K. Aggrey, “Evaluating the Effectiveness of ERP Systems in HEIs: A Proposed Analytic Framework,” *Proc. - 2019 Int. Conf. Comput. Comput. Model. Appl. ICCMA 2019*, pp. 40–45, 2019, doi: 10.1109/ICCMA.2019.00014.

[8] A. A. Pratama and A. B. Mutiara, “Software Quality Analysis for Halodoc Application using ISO 25010:2011,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 383–392, 2021, doi: 10.14569/IJACSA.2021.0120844.

[9] L. Nurdayenti and A. Amiruddin, “Comparative Analysis of Usability, Performance, and Security of Open-Source, Windows-Based Password Manager Applications Based on ISO/IEC 25010,” *Proc. - 4th Int. Conf. Informatics, Multimedia, Cyber Inf. Syst. ICIMCIS 2022*, pp. 178–183, 2022, doi: 10.1109/ICIMCIS56303.2022.10017543.

[10] Y. I. Irawan and E. S. Negara, “Evaluation of Software Quality Assurance Silampari Smart City Of Lubuklinggau Based On ISO/IEC 25010:2011 Analysis Model,” *Proc. - 4th Int. Conf. Informatics, Multimedia, Cyber Inf. Syst. ICIMCIS 2022*, pp. 154–160, 2022, doi: 10.1109/ICIMCIS56303.2022.10017834.

[11] S. J. R. Manglapuz and L. L. Lacatan, “Academic management android application for student performance analytics: A comprehensive evaluation using ISO 25010:2011,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 5085–5089, 2019, doi: 10.35940/ijitee.L2735.1081219.

[12] N. Kade, M. Handayani, G. Made, A. Sasmita, A. Agung, and K. Agung, “Evaluation Security Web-Based Information System Application Using ISSAF Framework ( Case Study : SIMAK-NG Udayana University ),” *JITTER - J. Ilm. Teknol. dan Komput.*, vol. 1, no. 2, 2020.

[13] I. Firman Ashari, M. Affandi, H. T. Putra, and M. T. Nur, “Security Audit for Vulnerability Detection and Mitigation of UPT Integrated Laboratory (ILab) ITERA Website Based on OWASP Zed Attack Proxy (ZAP),” *J. Teknol. Inf. dan Komunikasi)*, vol. 7, no. 1, p. 2023, 2023, [Online]. Available: https://doi.org/10.35870/jti

[14] K. Anagandula and P. Zavarsky, “An Analysis of Effectiveness of Black-Box Web Application Scanners in Detection of Stored SQL Injection and Stored XSS Vulnerabilities,” *Proc. - 2020 3rd Int. Conf. Data Intell. Secur. ICDIS 2020*, pp. 40–48, 2020, doi: 10.1109/ICDIS50059.2020.00012.

[15] Y. Makino and V. Klyuev, “Evaluation of web vulnerability scanners,” *Proc. 2015 IEEE 8th Int. Conf. Intell. Data Acquis. Adv. Comput. Syst. Technol. Appl. IDAACS 2015*, vol. 1, no. September, pp. 399–402, 2015, doi: 10.1109/IDAACS.2015.7340766.

[16] B. Mburano and W. Si, “Evaluation of web vulnerability scanners based on OWASP benchmark,” *26th Int. Conf. Syst. Eng. ICSEng 2018 - Proc.*, no. February, 2019, doi: 10.1109/ICSENG.2018.8638176.

[17] ISO 25000, “Iso 25000 Standards,” *The ISO/IEC 25000 series of standards*, 2022. https://iso25000.com/index.php/en/iso-25000-standards (accessed May 20, 2023).

[18] S. Karnouskos, R. Sinha, P. Leitão, L. Ribeiro, and T. I. Strasser, “The applicability of ISO/IEC 25023 measures to the integration of agents and automation systems,” *Proc. IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 2927–2934, 2018, doi: 10.1109/IECON.2018.8592777.

[19] M. N. Aziz, I. M. Sapta, and S. Rochimah, “Security Characteristic Evaluation Based on ISO/IEC 25023 Quality Model, Case Study: Laboratory Management Information System,” *2018 Electr. Power, Electron. Commun. Control. Informatics Semin. EECCIS 2018*, pp. 332–336, 2018, doi: 10.1109/EECCIS.2018.8692982.

[20] T. Rangnau, R. V. Buijtenen, F. Fransen, and F. Turkmen, “Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines,” *Proc. - 2020 IEEE 24th Int. Enterp. Distrib. Object Comput. Conf. EDOC 2020*, pp. 145–154, 2020, doi: 10.1109/EDOC49727.2020.00026.