[1] **Carlos P. Flores Jr.**

[2] **Richard N. Monreal**

Evaluation of Common Security Vulnerabilities of State Universities and Colleges Websites Based on OWASP

Journal of Electrical Systems

Abstract: - The security of state universities' and colleges' websites in the Philippines is vital because they play a critical role in delivering education and information to a wide variety of users. However, these institutions are also exposed to several security flaws due to their growing reliance on digital platforms. The objective of this study is to analyze security vulnerabilities in state universities and colleges websites, utilizing the OWASP Zed Attack Proxy (ZAP), an open-source tool. By adhering to the Open Web Application Security Project (OWASP) Top 10, we can identify potential hazards and suggest appropriate measures to mitigate risks. The steps of the test include gathering data about the test target, using OWASP ZAP to do automatic scanning, exploitation of the scan results, reporting, and offering recommendations. Seventeen (17) SUCs were examined, and the results show that 23.53% are vulnerable to injection, 40.06% had insecure design, 70.59% had outdated components, 88.24% have security misconfiguration, and 94.12% are vulnerable to Broken Access Control. Malicious actors use these vulnerabilities to obtain unauthorized access to software, networks, and systems. By raising the privileges and granting the user ID additional access inside the ecosystem, it can harm the availability, confidentiality, or integrity of data. SUCs should embrace the OWASP Top 10 and begin the process of ensuring that the risks associated with their websites are minimized.

Keywords: Cybersecurity, OWASP, ZAP, SUC.

I. INTRODUCTION

Technological innovation and social e-trends toward digitalization have recently had a significant impact on schools all around the world[1]. State Universities and Colleges (SUCs) are essential to supporting research and development and providing high-quality education in the Philippines. As online platforms and digital services have expanded quickly, SUCs have gradually embraced web technology to improve their educational options, operational procedures, and communication channels. SUCs are exposed to security flaws during digitalization, which can compromise sensitive data's availability, confidentiality, and integrity[2]. Academic institutions are a prime target for cybercriminals, espionage, and hacktivists because they handle vast volumes of sensitive personal data and significant research[3].

According to the Department of Information and Communications Technology (DICT), the Philippines experienced around 3,000 high-level cyber-attacks from 2020 to 2022.[4], and almost 30 schools were hacked in just June 2020 [5]. Attackers frequently use mistakes made when developing program code when creating web-based applications; in this case, vulnerabilities like SQL Injection, Authentication, and XSS are frequently exploited by attackers[6]. XSS occurs when a hacker injects potentially dangerous injection scripts into a webpage for browser users to use. These scripts include JavaScript and other similar content destroyers[7].

This study evaluates common security vulnerabilities in SUCs websites in the Philippines using Penetration Testing and security parameters from Open Web Application Security Project (OWASP) TOP 10. The OWASP version used was OWASP Top 10 2021 and the website vulnerability scanning tool used is the OWASP Zed Attack Proxy (ZAP). ZAP is an integrated web application penetration testing tool that is open source and simple to use Developers and functional testers with little to no security testing experience can benefit from ZAP [8] and are the best among scanners[9]..

II. LITERATURE REVIEW

The biggest issue with any network communication is data security[10]. In just one year, around 1 billion emails were exposed to the public, which affected 1 in 5 internet users. The average cost of a data breach was $4.35 million in 2022. In the first half of 2022, there were over 236.1 million ransomware attacks worldwide.[11].

Over the years, several standards and certifications have been suggested to mitigate software vulnerabilities. The most famous and well-known initiative in this regard is arguably the (OWASP)[12]. OWASP Top 10 2021 aims to enhance software security worldwide[13]. A study [14] benchmarked 30 web vulnerability scanners towards OWASP Web Security 2010,2013, 2017, and 2021.

[1] Carlos P. Flores Jr., University of the Cordilleras, cpf4204@students.uc-bcf.edu.ph

[2] Richard N. Monreal, University of the Cordilleras, rnmonreal@uc-bcf.edu.ph

The OWASP Top Ten 2007 vulnerability categories must be tested for by each scanner in the study of [15] because they are required to be PCI compliant. In the study [16] the security metrics are based on the OWASP top 10-2010. In the paper [17] they provide guidance on evaluating Open Source Web Application Security Scanners based on the OWASP Top 10-2013 application security risks. OWASP Top 10 2017 was first published in the spring of 2017 but was later withheld due to community disagreements on the validity of it [18] the complete version was finally released on October 20. According to the OWASP Top 10 2021 standard, security flaws are categorized. This classification is based on facts and information provided by companies that specialize in application security or gathered through industry surveys[19].
Any vulnerabilities on the SUC's website will be found using OWASP ZAP, and a report will be written summarizing the issue[20]. OWASP ZAP detected a significantly large number of vulnerabilities compared to other scanners[21].

III. RESEARCH METHODS

A. Website Vulnerability Evaluation

Automated vulnerability scanners for web applications carry out the process of checking. They typically consist of three parts: a web crawler for gathering website data, an attacker component for sending erroneous and random input to the web application, and an analyzer for examining the returned data, finding vulnerabilities, and producing a report[22]. In this study, the researchers evaluated the SUCs' website using the web application vulnerability scanner OWASP Zed Attack Proxy (OWSAP ZAP). This study used a qualitative methodology and had two sources of data: 1) document analysis of seventeen (17) SUCs websites and 2) OWASP Top 10 21 lists. This qualitative study was conducted as a needs assessment to identify common security vulnerabilities of SUCs websites.

B. Security Level Assessment with OWASP

The OWASP TOP 10 is a widely-used method of assessing the risk of application vulnerabilities on websites [23]. OWASP was established to offer a comprehensive platform for web developers to learn and improve website security. Developers can assess whether a website is secure by creating a checklist based on the OWASP 10 list of ten website security standards[24][25] including.

A01:2021-Broken Access Control: when unauthorized access occurs when an attacker gains access to information or systems that are restricted and not intended for their use.

A02:2021-Cryptographic Failures: a breach in security happens when a third party (apps, web pages, or other websites) exposes sensitive information.

A03:2021-Injection: threats like SQL/NoSQL Injection occur when user data mixes with the interpreter (such as a database engine, shell, template engine).

A04:2021-Insecure Design: a new category which refers to the lack of security controls being incorporated throughout the development cycle of an application.

A05:2021-Security Misconfiguration: configuration settings for the system or an application are either absent or implemented incorrectly, allowing unwanted access occurs.

A06:2021-Vulnerable and Outdated Components: refers to the use of unofficial, out-of-date, or known-vulnerable third-party libraries or frameworks in web applications.

A07:2021-Identification and Authentication Failures can occur when an application fails to implement or provide sufficient protection for user identification, authentication, or session management functions.

A08:2021-Software and Data Integrity Failures refers to situations where software and infrastructure lack sufficient protection against data tampering and other forms of unauthorized modifications, which can lead to integrity failures and data breaches.

A09:2021-Security Logging and Monitoring Failures: result in inadequate information for network defenses to detect hostile system breaches.

A10:2021-Server-Side Request Forgery: when a web application accesses a remote resource without validating the user-provided URL, it can lead to a security vulnerability.

C. Research Flow

The OWASP risk assessment method simplifies calculating, evaluating, and addressing application risks. Knowing potential dangers will help the save time as a developer and lessen the likelihood of more serious dangers [26]. This research consists of four (4) stages, namely, planning, scanning, exploitation, reporting, and recommendations as shown in Fig. 1.

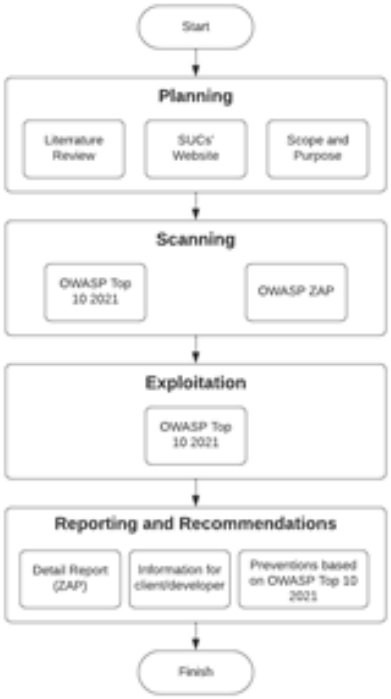

Fig. 1. Research Flow

To ensure a successful test, proper planning is crucial. The initial step in planning entails clearly outlining the test's scope and objectives and compiling a comprehensive list of SUCs that require evaluation. This phase culminates in collecting the requisite data for testing, including network and domain information, the system's functionality, and the flow of the computer system to be utilized.

During the scanning phase of the second step, the target is moved to identify security gaps that might allow unauthorized access to the system. One recommended tool for this task is Zed Attack Proxy (ZAP) from OWASP, which is simple, free of charge, and can automatically scan the target system to detect any vulnerabilities or open ports that can be exploited. These vulnerabilities and open ports can then be used to gain access to the system during the exploitation phase.

The third stage, exploitation, is to check the scanned security vulnerabilities in the form of injection, XSS, security misconfiguration, vulnerable and outdated components for possible loopholes.

The fourth and final phase of penetration testing is the reporting and recommendations stage, which entails the automatic creation of a comprehensive report. This report encompasses a summary of alert counts categorized by risk and confidence, as well as site and risk. Additionally, it includes a breakdown of alert counts

by alert type, references to discovered vulnerabilities, and recommendations based on the OWASP Top 10 to safeguard the website against potential attacks.

D. Actual Test Process

To start the scanning process after opening the ZAP tool the URL of the SUC website needs to be entered on the URL text box and select the options to be used and click the attack button as shown in Fig. 2.

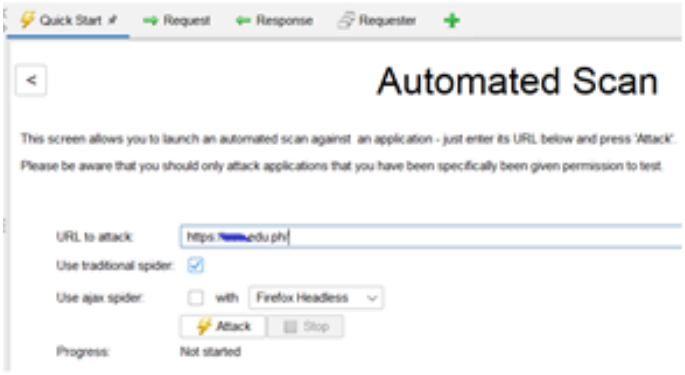

Fig. 2. Initializing ZAP Automated Scan

While ZAP undergoes the scanning process it provides details such as the current progress of the scan, URL found, Nodes Added, the method applied, URI, and alerts (high, medium, low, informational) as shown in Fig. 3 and wait for the scan to finish.

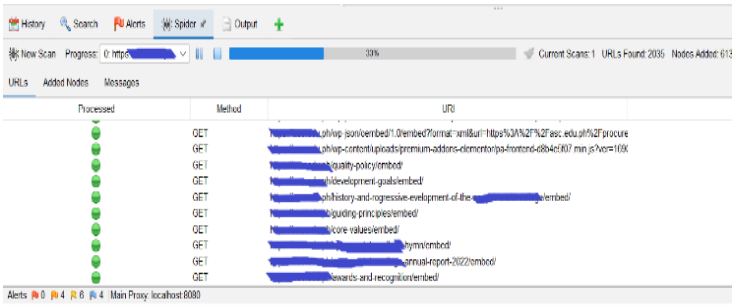

Fig. 3. ZAP starts the scanning process.

ZAP allows us to view the detailed information of the selected alert on the alert list that includes the alert, description, other information, and possible solutions as shown in Fig. 4.

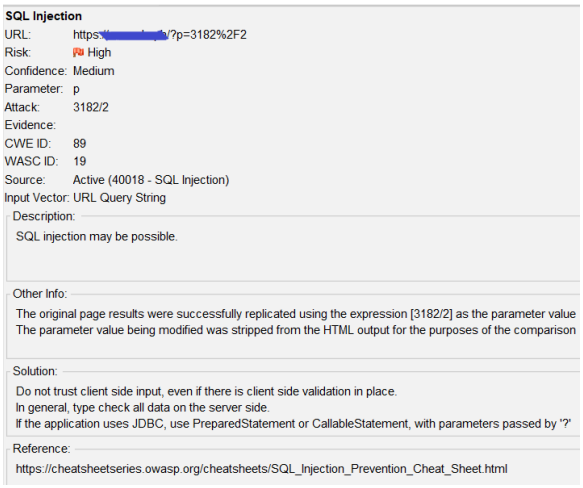

Fig. 4. Alert's Detailed Information

Once the scan is finished, the researchers can generate a report and save if for future reference as shown in Fig. 5.

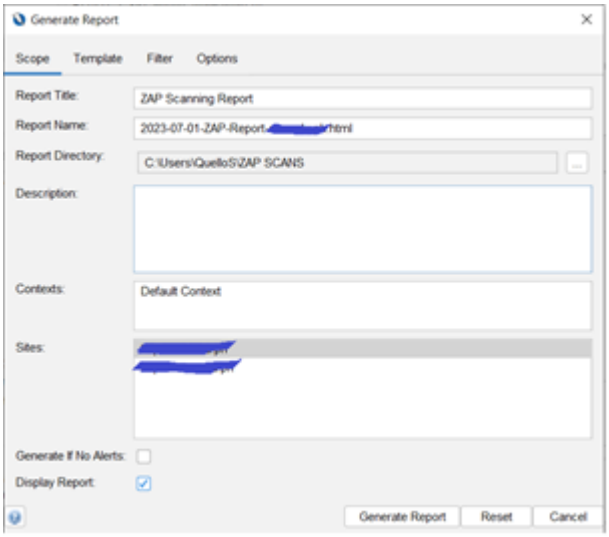

Fig. 5. ZAP Report Generation

III. RESULTS AND DISCUSSIONS

These are the observations the researchers made utilizing the OWASP ZAP after completing the four (4) stages of the study research flow. The findings of the experiment using the OWASP ZAP tool are shown as alert flags which indicate gaps, the level of danger caused by these alerts, and other information such as descriptions of security holes, along with preventive actions to overcome the solution of these loopholes. Fig. 6 show the results of seventeen (17) SUCs websites evaluated and scanned using ZAP.

SUC	Scan Results of 17 SUCs based on OWASP Top 10									
	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
SUC1	✓			✓	✓					
SUC2	✓	✓			✓	✓				
SUC3	✓			✓	✓					
SUC4	✓		✓		✓	✓				
SUC5	✓				✓	✓				
SUC6	✓			✓		✓				
SUC7	✓		✓	✓	✓	✓				
SUC8	✓				✓	✓				
SUC9	✓		✓		✓	✓				
SUC10	✓			✓	✓	✓				
SUC11	✓			✓	✓	✓				
SUC12	✓				✓					
SUC13	✓				✓	✓				
SUC14					✓	✓				
SUC15	✓				✓					
SUC16	✓		✓	✓	✓	✓				
SUC17	✓			✓						

Fig. 6. ZAP Result by SUC

Seventeen (17) SUCs were examined, and the results show that 23.53% are vulnerable to injection, 40.06% had an insecure design, 70.59% had outdated components, 88.24% had security misconfiguration, and 94.12% are vulnerable to Broken Access Control as shown. Fig. 7 shows that six out of 10 (6/10) of the OWASP Top 10:2021 where found on the evaluated SUCs websites.

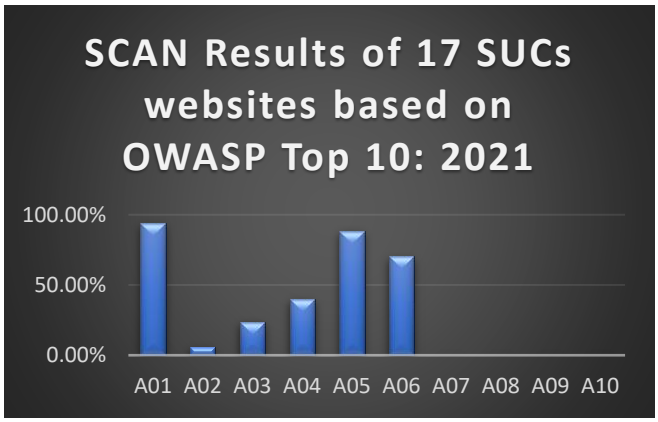

Fig. 7. Bar Chart of the 17 SUCs Scan Results

After analyzing the data collected on the SUC5 website, ZAP scanning generated three (3) high alerts, five(5) medium alerts, eight(8) low alerts, and nine(9) informational alerts. Scan results can be seen in Fig. 8.

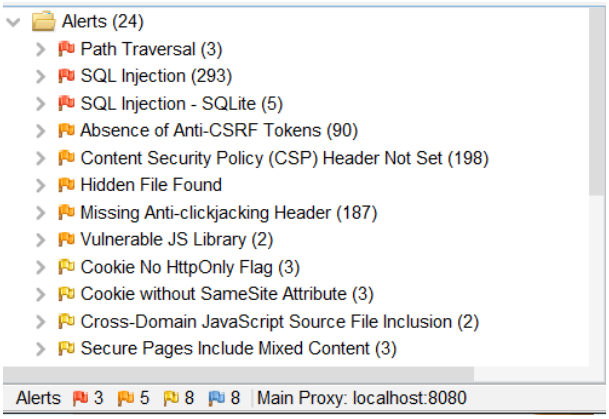

Fig. 8. Scan Result from a SUC4 Website

ZAP also produces the number of alerts for each level of risk and confidence, like on the case of SUC8 9.1% of the scan result marked as high risk, 27.3% as medium risk, 36.4% as low risk, and for informational risk the 27.3% were found is shown in Fig. 9.

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	1 (4.5%)	1 (4.5%)	0 (0.0%)	2 (9.1%)
	Medium	0 (0.0%)	1 (4.5%)	4 (18.2%)	1 (4.5%)	6 (27.3%)
	Low	0 (0.0%)	1 (4.5%)	6 (27.3%)	1 (4.5%)	8 (36.4%)
	Informational	0 (0.0%)	0 (0.0%)	1 (4.5%)	5 (22.7%)	6 (27.3%)
	Total	0 (0.0%)	3 (13.6%)	12 (54.5%)	7 (31.8%)	22 (100%)

Fig. 9. Alert counts by risk and confidence of one SUC8 website.

The detailed list of alert types from high to informational and their respective counts found by ZAP with link to its source, reference and possible prevention is shown in Fig. 10.

Alert type	Risk	Count
Hash Disclosure - Mac OSX salted SHA-1	High	1 (10.0%)
Path Traversal	High	4 (40.0%)
Absence of Anti-CSRF Tokens	Medium	2 (20.0%)
Content Security Policy (CSP) Header Not Set	Medium	615 (6,150.0%)
Vulnerable JS Library	Medium	16 (160.0%)
Big Redirect Detected (Potential Sensitive Information Leak)	Low	3 (30.0%)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	595 (5,950.0%)
Information Disclosure - Suspicious Comments	Informational	610 (6,100.0%)
Modern Web Application	Informational	614 (6,140.0%)
Re-examine Cache-control Directives	Informational	442 (4,420.0%)
Total		10

*Fig. 10. Alert type of SUC17 website*VII. CONCLUSION

Using IoT-based electronic health records (EHRs) and pooled learning to predict heart illnesses opens up a lot of benefits and possibilities in healthcare. This method solves important problems in healthcare data management by using the spread structure of IoT devices and the joint learning technique of federated learning. These problems include data protection, scalability, and data variety. One of the best things about shared learning in this situation is that it protects data privacy by keeping private EHR data on the IoT devices. Concerns about data leaks and illegal access are eased by this open method, which keeps patient data safe and private. Federated learning also makes it possible to combine data from many different Internet of Things (IoT) devices. This lets us look at patient health data in a more complete way and makes heart disease forecast models more accurate. Federated learning is also a good way to handle big amounts of EHR data created by IoT devices because it can be scaled up or down as needed. The pooled method lets model changes from multiple devices be combined, which lets strong and accurate prediction models be made without having to store or process data in one place. The shared learning method for predicting heart diseases in IoT-based EHRs is a great step forward in healthcare technology. It gives us a way to use IoT data to improve patient care and results that is private, flexible, and effective. In the future, researchers could work on making federated learning algorithms better at certain jobs like predicting heart disease, finding new ways to combine data and keep models up to date, and checking how federated learning works in clinical settings.

IV. CONCLUSION

The State Universities and Colleges' website in the Philippines has been successfully evaluated using the OWASP ZAP tool and checks the results based on the OWASP Top 10 2021. Based on the results above, it shows that most of the SUCs' websites are vulnerable to A01: Broken Access Control specifically in the (Absence of Anti-CSRF Tokens) with 94.12%, where 88.24% are reported to be at risk of A05: Security Misconfiguration that consists of Application Error Disclosure, Missing Anti-clickjacking Header, and Multiple X-Frame-Options Header entries. A06: Vulnerable and outdated components specifically on JS Library were found on 70.59% of the evaluated websites, 40.06% had A04: insecure design, and 23.53% are vulnerable to A03: injection. IT officials can use the test results as a guide to enhance the website application's security, particularly about issues with A01-Broken Access Control, A05-Security Misconfiguration, A06-Vulnerable and Outdated Components, A04-Insecure Design, and A03-Injection.

REFERENCES

[1] M. A. Mohamed Hashim, I. Tlemsani, and R. Matthews, "Higher education strategy in digital transformation," Educ. Inf. Technol., vol. 27, no. 3, pp. 3171–3195, 2022, doi: 10.1007/s10639-021-10739-1.

[2] D. Ghelani, "Diptiben Ghelani. Cyber Security, Cyber Threats, Implications and Future Perspectives: A Review," Am. J. Sci. Eng. Technol., vol. 3, no. 6, pp. 12–19, 2022, doi: 10.11648/j.XXXX.2022XXXX.XX.

[3] J. B. Ulven and G. Wangen, “A systematic review of cybersecurity risks in higher education,” Futur. Internet, vol. 13, no. 2, pp. 1–40, 2021, doi: 10.3390/fi13020039.
[4] J. C. Paunan, “PH records 3K high level cyberattacks,” 2023. https://pia.gov.ph/news/2023/04/13/ph-records-3k-high-level-cyberattacks (accessed Jul. 24, 2023).
[5] P. DEL PUERTO, “Almost 30 Philippine Schools Hacked Just This June,” 2020. https://blog.secuna.io/more-than-20-philippine-schools-hacked-just-this-june-are-we-ready-to-do-online-education/ (accessed Jul. 23, 2023).
[6] I. M. Edy Listartha, I. M. A. Premana Mitha, M. W. Aditya Arta, and I. K. W. Yuda Arimika, “Analisis Kerentanan Website SMA Negeri 2 Amlapura Menggunakan Metode OWASP (Open Web Application Security Project),” Simkom, vol. 7, no. 1, pp. 23–27, 2022, doi: 10.51717/simkom.v7i1.63.
[7] R. M. Wibowo and A. Sulaksono, “Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd,” Indones. J. Inf. Syst., vol. 3, no. 2, pp. 149–159, 2021, doi: 10.24002/ijis.v3i2.4192.
[8] S. Bennetts, “An Introduction to ZAP Zed Attack Proxy What is ZAP ?,” 2012.
[9] S. Tyagi, D. Sagar, S. Kukreja, J. Brahma, and P. Jain, “Studying Open Source Vulnerability Scanners for Vulnerabilities in Web Applications,” Iioab, vol. 9, pp. 43-49Tyagi, Shobha et al. 2018. “Studying Open Sou, 2018, [Online]. Available: www.iioab.org
[10] J. Kaur and K. R. Ramkumar, “The recent trends in cyber security: A review,” J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 8, pp. 5766–5781, 2022, doi: 10.1016/j.jksuci.2021.01.018.
[11] C. Griffiths, “The Latest 2023 Cyber Crime Statistics (updated July 2023),” 2023. https://aag-it.com/the-latest-cyber-crime-statistics (accessed Jul. 30, 2023).
[12] P. Ferrara, A. K. Mandal, A. Cortesi, and F. Spoto, “Static analysis for discovering IoT vulnerabilities,” Int. J. Softw. Tools Technol. Transf., vol. 23, no. 1, pp. 71–88, 2021, doi: 10.1007/s10009-020-00592-x.
[13] O. T. 10, “OWASP Top 10:2021.” https://owasp.org/Top10/ (accessed Jul. 29, 2023).
[14] 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, doi: 10.1109/ACCESS.2022.3161522.
[15] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, “State of the art: Automated black-box web application vulnerability testing,” Proc. - IEEE Symp. Secur. Priv., pp. 332–345, 2010, doi: 10.1109/SP.2010.27.
[16] N. M. N. and N. B. Ala A. Abdulrazeg, “Security measurement based on GQM to improve application security during requirements stage,” Int. J. Cyber-Security Digit. Forensics, vol. 1, no. 3, p. 221+, 2012.
[17] F. Abbas Saeed and E. E. Abed Elgabar, “Assessment of open source web application security scanners,” J. Theor. Appl. Inf. Technol., vol. 61, no. 2, pp. 281–287, 2014.
[18] H. Sohoel, M. G. Jaatun, and C. Boyd, “OWASP Top 10 - Do Startups Care?,” 2018 Int. Conf. Cyber Secur. Prot. Digit. Serv. Cyber Secur. 2018, no. 0102, 2018, doi: 10.1109/CyberSecPODS.2018.8560666.
[19] V. Barletta, G. Desolda, D. Gigante, R. Lanzilotti, and M. Saltarella, “From GDPR to Privacy Design Patterns: The MATERIALIST Framework,” no. Secrypt, pp. 642–648, 2022, doi: 10.5220/0011305900003283.
[20] G. Angga Septiawan, K. W. S. Irawan, I. Mayasari, and I. M. E. Listartha, “Analisis Kerentanan XSS dan Rate Limiting Pada Website SMAN 8 Denpasar Menggunakan Framework OWASP ZAP,” J. Inform. Upgris, vol. 8, no. 1, pp. 6–8, 2022, doi: 10.26877/jiu.v8i1.10271.
[21] 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.
[22] H. S. Abdullah, “Evaluation of Open Source Web Application Vulnerability Scanners,” Acad. J. Nawroz Univ., vol. 9, no. 1, p. 47, 2020, doi: 10.25007/ajnu.v9n1a532.
[23] E. B. Setiawan and A. Setiyadi, “Web vulnerability analysis and implementation,” IOP Conf. Ser. Mater. Sci. Eng., vol. 407, no. 1, 2018, doi: 10.1088/1757-899X/407/1/012081.
[24] R. Revo, G. Made, A. Sasmita, I. P. Agus, and E. Pratama, “Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application),” J. Ilm. Teknol. dan Komput., vol. 1, no. 1, 2020.
[25] Y. W, I. Riadi, and A. Yudhana, “Analisis Deteksi Vulnerability Pada Web Server Open Journal System Menggunakan OWASP Scanner,” Jurnal Rekayasa Teknologi Informasi (JURTI), vol. 2, no. 1. p. 1, 2018. doi: 10.30872/jurti.v2i1.1319.
[26] D. Priyawati, S. Rokhmah, and I. C. Utomo, “Website Vulnerability Testing and Analysis of Internet Management Information System Using OWASP,” Int. J. Comput. Inf. Syst. Peer Rev. J., vol. 03, no. 03, pp. 2745–9659, 2022, [Online]. Available: https://ijcis.net/index.php/ijcis/index