Available online to ejournal.pnc.ac.id

Journal of Innovation Information Technology and Application

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

Website Security Analysis Using Vulnerability Assessment Method (Case Study: Universitas Internasional Batam)

Haeruddin[1], Gautama Wijaya[2], Hendra Winata[3], Sukma Aji[4], Muhamamd Nur Faiz[5]

[1,2,3] Faculty of Computer Science, Universitas Internasional Batam, Indonesia
[4] Faculty of Science and Technology, Universitas Muhammadiyah Sidoarjo, Indonesia
[5] Department of Computer and Business, Politeknik Negeri Cilacap, Indonesia

email:[1] haeruddin@uib.ac.id, [2] gautama.wijaya@uib.ac.id, [3] hndrawnt135@gmail.com,[4] sukmaaji@umsida.ac.id, [5] faiz@pnc.ac.id

ARTICLE INFO

Article history:
Received 2 October 2024
Revised 13 December 2024
Accepted 21 December 2024
Available online 30 December 2024

Keywords:
Cybersecurity in Education
Vulnerability Assessment
OWASP ZAP
Nessus
Web Application Security

IEEE style in citing this article:
H. Winata, G. Wijaya, H. Haeruddin, S. Aji, and M. N. Faiz, "Website Security Analysis Using Vulnerability Assessment Method (Case Study: Universitas Internasional Batam)," Journal of Innovation Information Technology and Application (JINITA), vol. 6, no. 2, pp. 173–180, Dec. 2024.

ABSTRACT

In today's digital era, ensuring website security is crucial, especially in the education sector, frequently targeted by cyber attacks. This research aims to test the security of the Universitas Internasional Batam (UIB) website using OWASP ZAP and Nessus. The method used in this research was vulnerability assessment. It will involve gathering information using Nmap, whois, and nslookup tools. OWASP ZAP detected 11 vulnerabilities, categorized into 6 medium-level and 5 low-level, including Content Security Policies (CSP) and anti-clickjacking headers. Otherwise, Nessus only detected one medium-level vulnerability, the absence of HTTP Strict Transport Security (HSTS). The difference in detection results from the tools is that OWASP ZAP is better at finding web application weaknesses consistent with the OWASP Top Ten 2021, while Nessus specifically targets server and network configuration. For educational institutions, these results emphasize the importance of conducting regular vulnerability assessments to protect sensitive data. Recommended action includes implementing CSP to prevent Cross-site scripting (XSS) and other injection attacks, enforcing HSTS to secure communication, and updating software to mitigate unknown vulnerabilities. By adopting these measures, institutions can reduce their exposure to cyber-attacks, maintain user trust, and strengthen overall security. This research provides a practical framework for strengthening the security of educational websites against evolving threats. These findings highlight that the importance of using multiple tools can provide a more comprehensive view of security gaps.

1. INTRODUCTION

The number of development websites in Indonesia is currently very high, as the number of internet service users continues to over time. The data from APJII announced that the number of Indonesian internet users in 2024 will reach 221,563,479 out of a total population of 278,696,200 Indonesians in 2023 [1]. Such a large number of internet users can encourage the opening of cybercrime gaps as a medium of attack. Therefore, security is essential in developing web applications to minimize risks such as theft, manipulation, and data loss [2]. The threat of cyber-attacks in 2021, according to the National Cyber and Crypto Agency for the education sector, is the largest threat to others [3]. The main goal of the attacker is to steal the login credentials of the victim so they can perform spam and phishing attacks [4].

The development of web technology can make it easier for the educational sector to introduce or promote the institution [5]. Universitas Internasional Batam (UIB) is one of the universities located in the city of Batam, Riau Island. This university uses the website as a medium for media promotion, academic portal, E-Learning, etc [6]. The university website also contains student identity, lecturer identity, and

*) *Corresponding Author : hndrawnt135@gmail.com*

various important confidential information [7]. Violating application security for academics is fatal because it will decrease public trust in academics, in this case, the educational sector [8]. Therefore, the information available on the university website should be comprehensively guarded so that it cannot result in integrity violations or data theft. Knowing the security vulnerabilities can be done by utilizing the Vulnerability Assessment method in the hope of knowing the available security gaps.

Understanding security vulnerabilities is key to protecting web applications. This research uses the Vulnerability Assessment (VA) method, which is a systematic process for identifying, analyzing, and evaluating security weaknesses in web applications. It will also be used in one part of the penetration testing method, namely information gathering. This stage is to collect as much information as possible about the device or system, including details such as its IP address, subdomains, and type of CMS in use [9]. The VA phase aims to identify potential threats that malicious actors could exploit. The tools chosen for this study are OWASP ZAP and Nessus, which were widely recognized for their effectiveness in web vulnerability scanning. OWASP ZAP is an open-source penetration testing tool developed by the Open Web Application Security Project (OWASP) [10]. OWASP ZAP also offers many various features that allow users to perform automated scanning of vulnerabilities in web applications and provide detailed reports that could help in understanding and fixing such vulnerabilities [11], [12]. Nessus is a vulnerability scanner that can be used to detect security vulnerabilities in host OS, patches, and targeted services; it also shows the ability to propose solutions that can be used to mitigate vulnerabilities [13]. This combination of vulnerability scanner can be used to provide a comprehensive analysis of potential security gaps.

In carrying out this research, several previous studies were collected, which will be used as references in carrying out this research. Previous research done Muh. Adha, Zitnaa Dhiaaul KWA, and Alva Hendi Muhammad [2], in their research, conducted security testing on the University of Mataram website. In his research using the VA method, which also includes Vulnerability Assessment and Penetration Testing Life Cycle (VAPT-LC), the main focus of the research is on VA without conducting penetration testing. The VA process on the university mataram website went well and resulted in the findings of weaknesses or vulnerabilities. Using the OWASP ZAP tool, 14 data points were found for weakness, and OpenVAS found 2 data points for weakness.

Riyan Farismana and Dian Pramadhana did the next research [14] I. Riadi, A. Yudhana, and Y.W [15] and S. A. Putra, A. Budiono, and U. Y. K. S. Hediyanto [16], Focused on VA various systems using different tools. Riyan Farismana and Dian Pramadhana[14], in their research, conducted a comparison VA using OWASP ZAP and Acunetix tools, which was carried out on POLINDRA'S repository information system. The results of the study found that OWASP ZAP detected 22 alerts, and Acunetix detected 10 Alerts. I. Riadi, A. Yudhana, and Y.W [15] in their research testing the security of the OJS version 2.4.7 website. The tool used in this research was OWASP ZAP. The test carried out in this study found 70 high vulnerabilities, 1929 medium vulnerabilities, and 4050 low vulnerabilities. This shows that OJS version 2.4.7 still has a lot of vulnerabilities, and it suggests that the latest version of OJS be used to avoid existing vulnerabilities. S. A. Putra, A. Budiono, and U. Y. K. S. Hediyanto [16] in their research on student final project proposal web using acunetix and Nmap. The results of the study indicate that 12 vulnerabilities were detected on the dashboard website of the final project proposal of students at the Faculty of Industrial Engineering, Telkom University, with 5 medium-risk vulnerabilities and 7 low-risk vulnerabilities. The research was done by S. Eko. Prasetyo and N. Hassanah [17] in their research analyzing websites using the ISSAF method. The results of this study concluded that the security website is quite safe, but the website still can be attacked with DDOS penetration and cause the server to go down temporarily but with an active firewall security backup it is enough to help the server to avoid hacker attacks against subsequent attacks.

Several previous studies summarized above have similar characteristics and have become the foundation for carrying out this research. In this research, the author will test the security of the Universitas Internasional Batam website. The website needs to be tested for vulnerabilities to avoid unwanted or occurring things, such as data manipulation, data theft, or others by people who are not responsible for the website. Therefore, the author conducts research on the website with the aim of knowing the weaknesses that might occur so that these weaknesses can be found and improved.

2. METHOD

The methodology used in this research is Vulnerability Assessment, which involves a systematic process for identifying, analyzing, and evaluating security weaknesses in web applications [18]. This research aims to evaluate the security of Universitas Internasional Batam's main website by using OWASP

ZAP and Nessus as vulnerability scanner tools. Figure 1 shows the methodological steps in the university web application vulnerability assessment.

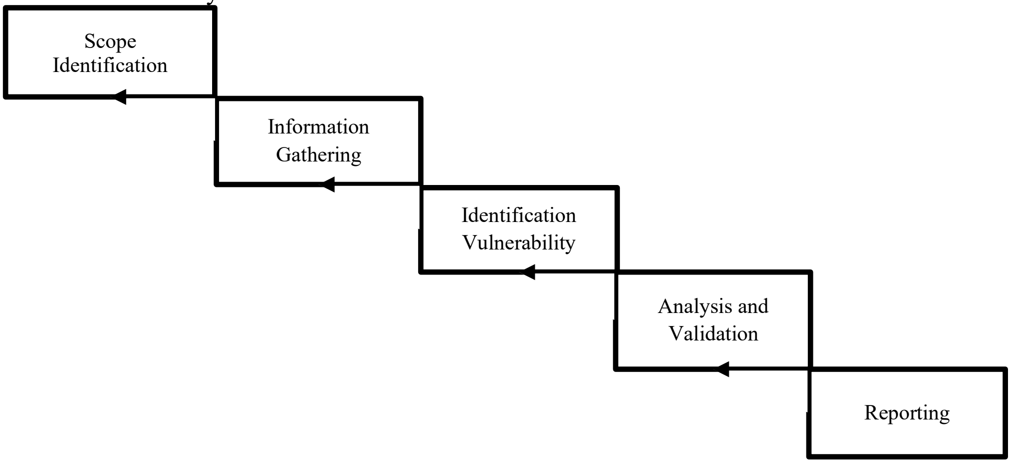


Figure 1. Web Application Vulnerability Assessment Methodology

2.1. Scope Identification

The first step was to define the limit of the assessment, where it was focusing on the university's main website. The university used this website to promote media and increase knowledge of the university.

2.2. Information Gathering

At this stage, the researcher will collect the relevant information about the target to prepare for the vulnerability assessment. It will use some tools such as Whois and Nmap to gather essential information, including IP addresses, subdomains, and open ports and services.

2.3. Identification Vulnerability

At this stage, the researcher will conduct automated vulnerability scans using two tools, OWASP ZAP and Nessus. In OWASP ZAP, an automated scan will be utilized to find vulnerabilities on a website, and there are also several advantages of the tools, namely traditional spider, AJAX spider, and active scan [19]. In Nessus, a web application test will be utilized to find vulnerabilities. This tool also provides the report with a Common Vulnerability Scoring System (CVSS) v3 score, which is a scoring system obtained from the National Vulnerability Database (NVD) [20].

2.4. Analysis and Validation

At this stage, the identified vulnerabilities will be analyzed to determine their severity and potential for the website. The vulnerabilities found by OWASP ZAP will be organized based on the 2021 OWASP Top Ten categories and will be categorized into risk and confidence, while Nessus will be categorized based on severity and CVSS V3.0 score.

2.5. Reporting

In the final stage, the vulnerabilities found by both vulnerability scanners will be compiled into a comprehensive report. The report will include the vulnerabilities detected during the VA with the vulnerability impact, and it will make recommendations for solutions for each vulnerability detected to improve website security.

3. RESULTS AND DISCUSSION

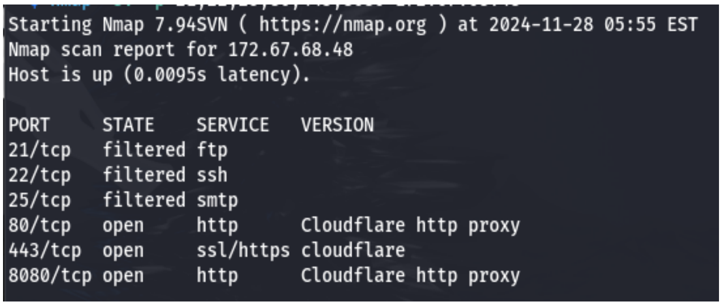
The VA process was focused on evaluating the security of the Universitas Internasional Batam (UIB) website. The scope was limited to the primary website used for promotion, ensuring the research remained target and actionable. The results from the tools used OWASP ZAP and Nessus.

3.1. Information Gathering

In conducting the vulnerability assessment, it will be using a Linux-based system with the following specifications:

Type	: Linux
Version	: Debian (64-bit)
Memory	: 6144 MB
Storage	: 100 GB

The next step is the information gathering stage, by using nslookup, whois and Nmap. nslookup will be used to retrieve the IP address of the target website, and using whois; the results will be obtained in the form of registrant information (domain owner company name, physical address, telephone number, and affiliated organization), registration information (domain registration date, next domain renewal date, and registrar name), and technical information (server name used). After determining the ip of the target, the next step is using the Nmap tool, with the command used is nmap -sV -p 21,22,25,80,443,8080 <target ip>; this command is used to identify services running on ports 21,22,25,80,443 and 8080 on the website and is used to see the version of the service contained on the ports mentioned above. The results of scanning the port with Nmap will be shown at Figure 2.



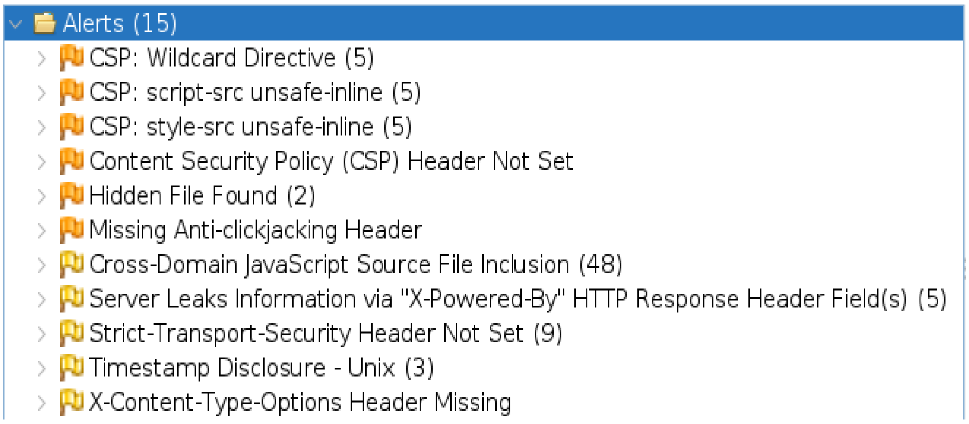
```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-28 05:55 EST
Nmap scan report for 172.67.68.48
Host is up (0.0095s latency).

PORT     STATE    SERVICE   VERSION
21/tcp   filtered ftp
22/tcp   filtered ssh
25/tcp   filtered smtp
80/tcp   open     http      Cloudflare http proxy
443/tcp  open     ssl/https cloudflare
8080/tcp open     http      Cloudflare http proxy
```

Figure 2. NMAP Scanning Results

3.2. Identification Vulnerability

The next step after information gathering is the identification of vulnerability by using vulnerability scanners such as OWASP ZAP and Nessus to identify security issues on the website. By using the automated vulnerability scanner tool, OWASP ZAP identified 11 types of alerts. With details, there are 6 data at the medium level and 5 data at the low level, while no weaknesses were found at the high level, as shown in Figure 3.



Alerts (15)
- CSP: Wildcard Directive (5)
- CSP: script-src unsafe-inline (5)
- CSP: style-src unsafe-inline (5)
- Content Security Policy (CSP) Header Not Set
- Hidden File Found (2)
- Missing Anti-clickjacking Header
- Cross-Domain JavaScript Source File Inclusion (48)
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (5)
- Strict-Transport-Security Header Not Set (9)
- Timestamp Disclosure - Unix (3)
- X-Content-Type-Options Header Missing

Figure 3. OWASP ZAP Scanning results

The next vulnerability scanner tool to be used was Nessus. While scanning using Nessus only 1 vulnerability was identified and no vulnerability with critical. High, and low were found. The results of scanning website using Nessus are presented in Figure 4.

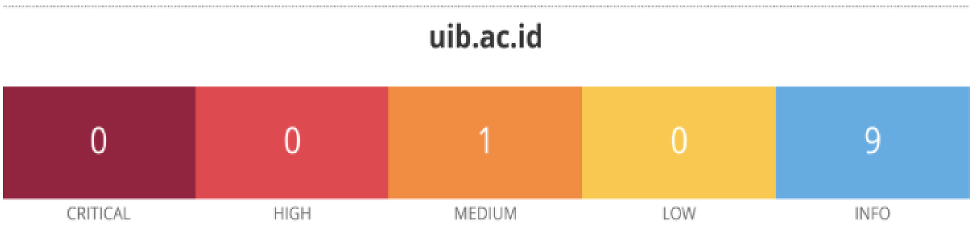


Figure 4. Nessus Scanning results

3.3. Analysis and Validation

At this stage every vulnerabilities was found using OWASP ZAP and Nessus will be categorized based on the severity and the potential to the website. The vulnerabilites found by OWASP ZAP will be categorized into alert name, risk, and OWASP Top Ten 2021, the results will be shown at Table 1.

Table 1. Alert Category based on OWASP Top Ten 2021

No	Alert	Risk	Category
1	CSP : Wildcard Directive	Medium	A05
2	CSP : script-src unsafe-inline	Medium	A05
3	CSP : style-src unsafe-inline	Medium	A05
4	Content Security Policy (CSP) Header Not Set	Medium	A05
5	Hidden File Found	Medium	A05
6	Missing Anti-clickjacking Header	Medium	A05
7	Cross-Domain JavaScript Source File Inclusion	Low	A08
8	Server Leaks Information via “X-Powered-By” HTTP Response Header Field(s)	Low	A01
9	Strict-Transport-Security Header Not Set	Low	A05
10	Timestamp Disclosure – Unix	Low	A01
11	X-Content-Type-Options Header Missing	Low	A05

While Nessus scanning results will be categorized as the vulnerability name, severity, and CVSS V3.0 score, the results will be shown at Table 2

Table 2. Vulnerability found by Nessus with CVSS score

No	Vulnerability	Severity	CVSS V3.0
1	HSTS Missing From HTTPS Server (RFC 6797)	Medium	6.5

3.4. Report

At the final stage the results of the scanning website using OWASP ZAP and Nessus was revealed a significant difference, which OWASP ZAP was detected 11 vulnerabilities with 6 data at the medium-level and 5 data at the low-level, while Nessus only detected 1 vulnerability with the medium-level. The comparison of scanning results will be shown at Table 3.

Table 3. Comparison of Scanning Results

Software	Results		
	High	Medium	Low
OWASP ZAP	0	6	5
Nessus	0	1	0

OWASP ZAP tools were strong in identifying web application vulnerabilities, specifically issues such as CSP, cross-domain misconfiguration, and server misconfiguration. It was focused on detecting vulnerabilities related to the OWASP Top Ten categories, which is highly effective for application-level security. While the Nessus vulnerability scanner doesn’t find many vulnerabilities, its primary strength lies in its network-level assessments. The mid-level vulnerabilities that were identified emphasized its narrower focus when used for web application security. Based on the scanning results it shows that relying on a single tool can lead to an incomplete vulnerability assessment. Using a combination of tools such as OWASP ZAP and Nessus can provide a more comprehensive view of security gaps, addressing both application and infrastructure-level vulnerabilities.

These findings emphasize the importance of regular vulnerability assessments for educational websites. Identified vulnerabilities, such as missing HSTS and poor CSP, can lead to sensitive user data being targeted by attackers such as MiTM and XSS attacks. Mitigating these issues is critical to protecting user trust and ensuring strict compliance with security best practices. Each vulnerability discovery detected by the tools will be explained, along with the impact of the vulnerability and a recommendation for a solution. The vulnerabilities found by OWASP ZAP will be shown in Table 4, while Nessus vulnerabilities will be shown in Table 5.

Table 4. Vulnerability Impact and Recommendation found by OWASP ZAP

No	Alert	Vulnerability Impact	Recommendation
1	CSP : Wildcard Directive	An overly permissive Content Security Policy (CSP) can allow script execution from any source, including malicious websites. This can lead to many different types of attacks, such as XSS, data theft, and redirection.	Configure your web server, application server, load balancer, etc. to set the Content-Security-Policy header correctly.
2	CSP : script-src unsafe-inline	Allows execution of scripts embedded directly in HTML, which can be exploited by attackers to inject malicious code.	
3	CSP : style-src unsafe-inline	Allows execution for styles (CSS). Attackers can inject malicious styles to change the appearance and behavior of the page.	
4	Content Security Policy (CSP) Header Not Set	Without CSP, browsers have no clue about the resources they are allowed to load, making it easier for attackers to inject malicious content.	Make sure the web server, application server, load balancer, etc. are set up to include the CSP header.
5	Hidden File Found	Hidden files that should not be accessible to general users may contain sensitive information or be utilized as a starting point for further attacks.	Evaluate whether the component is important in production. If not, deactivate it. If yes, ensure that the component requires proper authentication and authorization or restrict access to certain internal systems or source IP.
6	Missing Anti-clickjacking Header	Clickjacking attacks can trap users into clicking on an invisible object, which allows the perpetrator to execute actions on their behalf without their knowledge.	Modern web browsers support the CSP and X-Frame-Options HTTP headers. Always ensure that one of these headers is included on every webpage served by your website or application. If you anticipate the page will be framed exclusively by a page on your server, use SAMEORIGIN.
7	Cross-Domain JavaScript Source File Inclusion	The inclusion of JavaScript files from other domains can lead to XSS attacks if the files are not properly sanitized.	Make sure that the JavaScript source files are downloaded from a reliable source and cannot be manipulated by the application's end user.
8	Server Leaks Information via “X-Powered-By” HTTP Response Header Field(s)	These headers can provide information about the server software being used, which can help attackers identify specific vulnerabilities.	Ensure that the web server, application server, load balancer, etc., are set up to trigger the "X-Powered-By" headers.
9	Strict-Transport-Security Header Not Set	Without HSTS, browsers will not automatically redirect all requests to HTTPS, allowing man-in-the-middle attacks.	Configure your server infrastructure (including web servers, application servers, and load balancers) to enforce strict transport security (HSTS).
10	Timestamp Disclosure – Unix	Accurate timestamp disclosure can help attackers perform playback or brute-force attacks.	Manually verify that the timestamp information is non-sensitive and cannot be combined or analyzed in a way that reveals exploitable patterns.
11	X-Content-Type-Options Header Missing	If the X-Content-Type-Options header is missing, then the browser will try to guess the content type of the received file by itself. This opens the door to a variety of attacks, including XSS (Cross-Site Scripting), Malicious file uploads, and Bypassing security mechanisms.	Ensure that the web or application server correctly sets the Content-Type header for all responses and applies the X-Content-Type-Options header with a value of 'nosniff' to prevent MIME type sniffing on all web pages.

Table 5. Vulnerability Impact and Recommendation found by Nessus

No	Name	Vulnerability Impact	Recommendation
1	HSTS Missing From HTTPS Server (RFC 6797)	This vulnerability allows attackers to downgrade the connection from HTTPS to HTTP, opening up opportunities for interception and manipulation of sensitive user data during transmission. Without HSTS, communication between users and websites becomes vulnerable to man-in-the-middle attacks, where attackers can insert malicious code or redirect users to fake sites. Improper HSTS configuration makes websites vulnerable to various types of cyberattacks, including downgrades and man-in-the-middle attacks	Configure the remote web server to use HSTS.

4. CONCLUSION

The results of the vulnerability assessment process on the Universitas Internasional Batam (UIB) website using two vulnerability scanner tools, OWASP ZAP and Nessus. The results revealed significant differences in each tool. Where OWASP ZAP detected 11 vulnerabilities, including 6 at the medium level and 5 at the low level, whereas Nessus only detected 1 medium-level vulnerability, it can be said that OWASP ZAP was strengthened in identifying web application vulnerability, particularly issues related to OWASP Top Ten 2021, while Nessus primary strength lies on network-level assessments.

This research has emphasized the importance of regular and comprehensive vulnerability assessments to protect sensitive data in educational institutions. Vulnerabilities such as HSTS missing configuration and poor CSP can lead to sensitive user data being targeted by attackers, such as MiTM and XSS attacks. Implementing recommended security measures, such as configuring HSTS and strengthening CSP, can significantly reduce these risks. In future research, penetration testing could be conducted to explore the exploitation and impact of vulnerabilities in the real world. Additionally, additional vulnerability scanning tools could be evaluated to look for vulnerabilities in other security systems, such as dynamic testing and network security, to provide greater insight into the effectiveness of different approaches to web security.

REFERENCES

[1] A. T. Haryanto, “APJII: jumlah pengguna internet Indonesia tembus 221 juta orang,” inet.detik.com. Accessed: Aug. 12, 2024. [Online]. Available: https://inet.detik.com/cyberlife/d-7169749/apjii-jumlah-pengguna-internet-indonesia-tembus-221-juta-orang

[2] M. Adha, Z. D. KWA, and A. H. Muhammad, “Website security test at the university of mataram using vulnerability assessment,” *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 2, pp. 647–655, May 2023, doi: 10.29100/jipi.v8i2.3830.

[3] I. F. A. 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),” *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 7, no. 1, pp. 24–34, Jan. 2023, doi: 10.35870/jtik.v7i1.657.

[4] S. A. Jawaid, “Cyber Security Threats to Educational Institutes: A Growing Concern for the New Era of Cybersecurity,” *International Journal of Data Science and Big Data Analytics*, vol. 2, no. 2, May 2023, doi: 10.51483/ijdsbda.2.2.2022.11-17.

[5] Ş. S. Macakoğlu, S. Peker, and İ. T. Medeni, “Accessibility, usability, and security evaluation of universities’ prospective student web pages: a comparative study of Europe, North America, and Oceania,” *Univers Access Inf Soc*, vol. 22, no. 2, pp. 671–683, Jun. 2023, doi: 10.1007/s10209-022-00869-9.

[6] M. Huda, “Analisis kualitas website universitas sebagai media informasi dengan metode webqual 4.0,” *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 4, no. 1, pp. 241–254, Jan. 2023, doi: 10.35870/jimik.v4i1.166.

[7] N. A. Syarifudin and L. Setiyani, “Analysis of higher education SIAKAD website security gaps using the vulnerability assessment method,” *International Journal of Multidisciplinary Approach Research and Science*, vol. 1, no. 03, pp. 332–344, Aug. 2023, doi: 10.59653/ijmars.v1i03.177.

[8] N. Sulisnawati, “Implementation of Open Web Application Security Project for Penetration Testing on Educational Institution Websites,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 2, pp. 250–267, 2023, doi: 10.26555/jiteki.v9i2.25987.

[9] H. Hermanto and H. Haeruddin, “Peningkatan Sistem Keamanan Website Menggunakan Metode OWASP,” *Jurnal Ilmu Komputer dan Bisnis*, vol. 13, no. 1, pp. 94–104, May 2022, doi: 10.47927/jikb.v13i1.277.

[10] A. Alhogail and M. Alkahtani, “Automated extension-based penetration testing for web vulnerabilities,” *Procedia Comput Sci*, vol. 238, pp. 15–23, 2024, doi: 10.1016/j.procs.2024.05.191.

[11] N. Herawati, V. Budiyanto, and Uminingsih, “Analisis keamanan sebuah domain menggunakan open web application security project (OWASP) Zap,” *JURNAL TEKNOLOGI TECHNOSCIENTIA*, vol. 15, no. 2, pp. 27–37, Mar. 2023, doi: 10.34151/technoscientia.v15i2.4013.

[12] F. P. E. Putra, U. Ubaidi, A. Hamzah, W. A. Pramadi, and A. Nuraini, “Systematic Literature Review: Security Gap Detection On Websites Using Owasp Zap,” *Brilliance: Research of Artificial Intelligence*, vol. 4, no. 1, pp. 348–355, Jul. 2024, doi: 10.47709/brilliance.v4i1.4227.

[13] D. Priyawati, S. Rokhmah, and I. C. Utomo, “Website vulnerability testing and analysis of website application using OWASP,” *International Journal of Computer and Information System (IJCIS)*, vol. 3, no. 3, pp. 143–147, Sep. 2022, doi: 10.29040/ijcis.v3i3.90.

[14] 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,” *Applied Sciences*, vol. 12, no. 8, p. 4077, Apr. 2022, doi: 10.3390/app12084077.

[15] Riyan Farismana and Dian Pramadhana, “Perbandingan vulnerability assesment menggunakan owasp zap dan acunetix pada sistem informasi repositori politeknik negeri indramayu,” *Jurnal Teknik Informatika dan Teknologi Informasi*, vol. 3, no. 2, pp. 26–32, Aug. 2023, doi: 10.55606/jutiti.v3i2.2853.

[16] I. Riadi, A. Yudhana, and Y. W, “Analisis keamanan website open journal system menggunakan metode vulnerability assessment,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 4, pp. 853–860, Aug. 2020, doi: 10.25126/jtiik.2020701928.

[17] S. A. Putra, A. Budiono, and U. Y. K. S. Hediyanto, “Vulnerability assesment web proposal tugas akhir mahasiswa menggunakan acunetix dan NMAP,” *e-Proceeding of Engineering*, vol. 10, no. 2, pp. 1615–1622, 2023.

[18] S. Eko Prasetyo and N. Hassanah, “Analisis keamanan website universitas internasional batam menggunakan metode ISSAF,” *JURNAL ILMIAH INFORMATIKA*, vol. 9, no. 02, pp. 82–86, Sep. 2021, doi: 10.33884/jif.v9i02.3758.

[19] T. Adeniran *et al.*, “Vulnerability assessment studies of existing knowledge-based authentication systems: a systematic review,” *Sule Lamido University Journal of Science & Technology*, vol. 8, no. 1, pp. 34–61, 2024, doi: 10.56471/slujst.v7i.485.

[20] M. Rizkillah and F. Astutik, “Analisis Kerentanan Web Server pada Aplikasi Elearning (Studi Kasus Universitas Muhammadiyah Mataram),” 2023. [Online]. Available: https://journal.ummat.ac.id/index.php/jintens/index

[21] A. D. Tudosi, A. Graur, D. G. Balan, and A. D. Potorac, “Research on Security Weakness Using Penetration Testing in a Distributed Firewall,” *Sensors*, vol. 23, no. 5, Mar. 2023, doi: 10.3390/s23052683.