 ISSN 2579-7298

Security Mitigation of the Open Journal System (OJS) Against Online Gambling Content Hijacking Using the ISSAF Framework

Sarjimin [a,1,*], Anggit Gusti Nugraheni[a,2]

[a] Universitas Putra Bangsa, Jl Rongggowarsito No 18 Pejagoan, Kebumen 54312, Indonesia
[1] jimin@fst.universitasputrabangsa.ac.id*; [2] anggitgusti@fst.universitasputrabangsa.ac.id;
* corresponding author

ARTICLE INFO

Article history
Received 27 July 2025
Revised 17 October 2025
Accepted 28 November 2025
Available Online 30 December 2025

Keywords

Vulnerability
OJS
ISSAF
Webmin

ABSTRACT

The urgency of this research is to identify the causes, develop mitigation methods, and enhance the security of OJS websites, as many are infiltrated or hijacked for online gambling or other harmful content. Securing OJS websites is never easy because attacks are increasingly diverse and innovative every day. OJS system security is essential to protect the information contained therein and protect the services provided by scientific journal publishers. The ISSAF framework, which uses a simulation approach similar to a real server, can serve as a basis for identifying OJS Website vulnerabilities in Webmin for a system administrator. The results of the identification in this study indicate that the leading cause of OJS web server attacks originates from outside the simulation environment, specifically the internet network via ports 80/443. Vulnerability Session Hijacking with Cookies receives a CVSS vulnerability score of 9.1. A vulnerability in the web server configuration folder structure, traceable by crawler tools, receives a CVSS vulnerability score of 5.3. Repeated login attempts to the OJS system are not banned, and blocking the Attacker's IP receives a CVSS vulnerability score of 6.5. A file with the .php extension was successfully uploaded; it may be a backdoor file with a CVSS vulnerability score of 5.3. Although the OJS PKP changed/forced the file to .txt, the malicious file could be exploited in the future by unauthorized users. The novelty of this research lies in a server simulation that mimics a real server and the ISSAF framework for assessing the security of the Webmin web-based system administration tool on OJS websites.

1. Introduction

Many organizations develop web servers without adequately considering whether the deployed servers comply with established security standards, whether the implemented systems are secure, or whether they remain susceptible to potential disruptions. [1]. For instance, in the case of the Servio website, several vulnerabilities were identified, including HTML pages lacking Cross-Site Request Forgery (CSRF) protection, susceptibility to clickjacking attacks, and the presence of multiple informational web alerts [2]. Developing a secure web server is essential to ensure data confidentiality, integrity, and service availability within the implemented system [3], [4]. An insecure

website poses the risk that managed data may be easily accessed or misused by unauthorized parties. [3], [5]or even compromised through website hijacking by undesirable content, such as online gambling advertisements. On the other hand, ensuring cybersecurity is never an easy task, as attacks are becoming increasingly innovative every day. Therefore, it is crucial to clearly define cybersecurity and continuously enhance the level of protection [6].

The Open Journal System (OJS) is an open-source platform for scholarly article publication [7]. Open-source licenses permit all or part of a software's source code to be viewed by Internet users, enabling anyone to study and modify the code to suit their needs. This openness likewise applies to malicious actors, who can examine OJS source code and its weaknesses and subsequently exploit them for harmful or criminal purposes.

Website hijacking is a form of cyber threat that is increasingly prevalent, whereby attackers take over or modify a website without the consent of its owner. This phenomenon not only threatens commercial websites but also targets university websites, which are often perceived as having lower levels of security. University websites are prime targets because they are considered to have strong reputations in search engine rankings. This enables attackers to exploit high traffic for their own purposes, such as displaying online gambling advertisements. Fig. 1 illustrates an instance of website hijacking involving online gambling content on an OJS website. In this case, the hacker successfully gained administrator access to the OJS by exploiting several zero-day vulnerabilities [8], enabling them to obtain journal manager privileges merely by registering as an author on the OJS platform.

Fig. 1. Website hijacking online gambling on the OJS platform

Webmin is a web-based interface for system administration on Unix/Linux servers. Virtualmin is an additional module (plugin) for Webmin specifically designed to manage virtual hosts, such as in shared hosting environments. Virtualmin provides facilities and ease in creating and managing virtual domains (similar to cPanel), email accounts, databases, DNS, SSL, FTP, and file hosting. According to Enlyft data, 1,179 companies use Webmin, primarily in the Information Technology and Services industry, with employee numbers ranging from 50 to 200 and revenues between $1 million and $10 million (as shown in Fig. 2). This Webmin user data is relevant to the needs of universities, which typically have administrators and resources that are relatively limited.

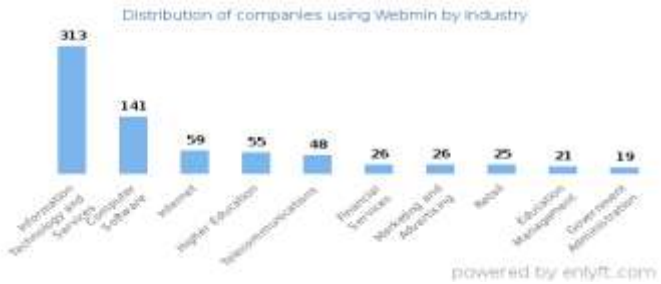

Fig. 2. Top Industries that use Webmin

Source: https://enlyft.com/tech/products/webmin

Literature Review

Ensuring the security of servers managed through the Webmin control panel, which host Open Journal Systems (OJS) websites, is crucial for maintaining data confidentiality, integrity, and service availability. [9, p. 304], [10, p. 9].

Several prior studies have examined the security of Open Journal Systems (OJS) websites; however, these investigations primarily utilized older versions such as OJS 2.4.7 [3]. In contrast, the current version, OJS 3.4, exhibits distinct vulnerabilities, including susceptibility to Denial-of-Service (DoS) attacks due to the system's inability to restrict or block repeated failed login attempts [1]. The use of Webmin as a web-based system administration interface for Open Journal Systems (OJS) has received limited research attention. Prior studies have predominantly examined cPanel-based administration environments, where default configuration settings have been shown to expose security weaknesses that may be exploited for Distributed Denial-of-Service (DDoS) attacks. [11]. Therefore, there is a need to strengthen and optimize the security configurations within cPanel to mitigate existing vulnerabilities and enhance system resilience against potential exploitation. Previous studies on information system websites or platforms similar to OJS have primarily focused on the web application layer itself, without considering the impact of the underlying web-based administration tools [12], [13], [14], [15], [16], [17], [18]. However, the security posture of a website, particularly OJS, strongly depends on the configuration and robustness of the web server it operates on [18].

Webmin acts as a solution offering a web-based graphical interface, easing the configuration and management of web servers. Experimental results show that Webmin provides satisfactory throughput and resource efficiency while maintaining secure and manageable administrative access [19]. However, research specifically focused on the security aspects of OJS combined with Webmin remains limited.

The ISSAF framework is a highly structured methodology that categorizes information system security into specific domains and evaluation stages [1], [10]. Applying this framework provides a robust methodological foundation and strong justification for the research findings. ISSAF serves as an appropriate approach for conducting a comprehensive analysis of system vulnerabilities and developing effective countermeasures. Consequently, the conclusions drawn from the ISSAF-based assessment and the formulated mitigation strategies can serve as a strategic reference to enhance the overall security of the OJS website.

Based on the aforementioned background, the novelty of this research lies in the security evaluation of the latest version of Open Journal Systems (OJS) 3.4.0, managed through the Webmin web-based system administration tool. This study employs the ISSAF framework as a methodological approach to identify, analyze, and mitigate potential security vulnerabilities arising from the integration between OJS and Webmin.

2. Method

2.1. Research Object

This study focuses on OJS version 3.4.0.8, installed on the subdomain jurnal.universitas**.ac.id. It employs a quantitative experimental approach, using a case study of Open Journal Systems (OJS) version 3.4.0.8 to assess security vulnerabilities.

2.2. Research Stages

The research process stages in this study follow the ISSAF framework. The ISSAF framework helps in thoroughly identifying potential vulnerabilities within the system. The stages of the ISSAF method are shown in Fig. 3.

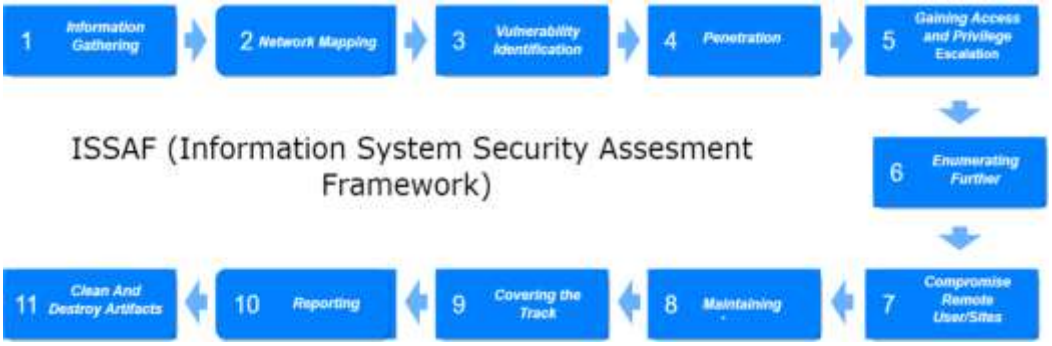

Fig. 3. ISSAF Framework Stages

The procedures corresponding to each stages of the ISSAF framework are detailed as follows:

1. Information Gathering
   The information-gathering stage collects data about the system and prepares for penetration assessment. Tests in this stage include, but are not limited to, verifying SSL configuration, DNS enumeration, gathering domain information, and identifying CMS.
2. Network Mapping
   The network-mapping stage uses a technical method by injecting a system footprint to identify network topology and exposed services.
3. Vulnerability Identification
   During vulnerability identification, various activities are performed to uncover weaknesses in the system. This stage employs automated and manual assessment to detect security flaws.
4. Penetration
   The penetration stage involves active attacks aimed at gaining unauthorized access by exploiting known vulnerabilities and bypassing security controls to achieve different privilege levels.
5. Gaining Access and Privilege Escalation
   This follow-up stage tests the system using access obtained during penetration to escalate privileges and broaden control within the environment.
6. Enumerating Further
   After gaining penetration and escalating privileges, further enumeration gathers more detailed data and information. This deeper reconnaissance assists in identifying additional security weaknesses in the targeted system.
7. Compromise Remote Users/Sites
   In this stage, identified vulnerabilities are leveraged remotely to obtain broader system access, facilitating deeper exploration and assessment of the target.
8. Maintaining Access
   The maintaining-access stage involves techniques (e.g., installing backdoors or rootkits) to preserve access to the system even if initial entry vectors are closed.
9. Covering the Track
   The pentester hides files and removes or modifies log entries to eliminate traces of assessment activities. This is done to mimic real attacker behavior and evaluate detection and response capabilities.
10. Reporting
    The reporting stage documents the results of the tests and provides recommendations. Its purpose is to inform stakeholders of findings, impact, and remediation measures.
11. Clean and Destroy Artifacts
    The final stage wipes all data and artifacts from the assessment. The goal is to make sure no traces stay on the system after it's finished.

2.3. Assessment Tools and Data Analysis

In the assessment phase of the OJS and Webmin websites, the researcher using several relevant tools. The tools utilized in this study are presented in Table 1.

Table 1. Tools Used in the Research

Stage	Source	Tools
Information Gathering	Domain Info	Nikto, Whois.
	SSL	SSL Scan, DNS Lookup
Network Mapping	Network Info	NMap.
Vulnerability Identification	Web Scanner Vulnerability	Acunetix, OWASP ZAP, Nikto, Nessus
Exploitation	DoS Attack	Low Orbit Ion Canon
	SQL Inject	Wireshark, SQL Map
	Metasploit	Metasploit
Gaining Access & Privileges	Backdoor	PHP rootkit
Enumerating Further	Backdoor	PHP rootkit
Compromise Remote User/Site	Backdoor	PHP rootkit
Maintaining Access	Backdoor	PHP rootkit
Covering Tracks	Backdoor	PHP rootkit
Reporting		Manual
Clean and destroy artifacts		Manual

The final stage involves conducting analysis and preparing a report based on the penetration assessment results obtained during the ISSAF framework stages.

2.4. Assessment Framework

The assessment in this study was carried out within a virtualization topology created using VirtualBox. The research object (jurnal.universitas**.ac.id) was hosted on a CT Web Server container with the IP address 192.168.169.4/29. The control panel for the CT Web Server was Webmin, accessible on port 10000, and it ran on Debian Linux 12. The CT Web Server was located on a physical virtualization server running Proxmox 12 with the IP address 192.168.169.3/29. This server was connected to a firewall, which served as a filtering and defense mechanism against potential external threats. The overall assessment framework is shown in Fig. 4.

Access to the Webmin Control Panel on port 10000 was restricted solely to the Administration Admin host with IP address 192.168.169.2/29. Likewise, access to the Proxmox Manager on the Virtualization Server was limited to port 8443 and was only available through the same Administration Admin host. All other unspecified ports toward the Administration Admin and Virtualization Server were blocked.

The firewall configuration for the CT Web Server only permitted public connections through port 80 (HTTP) and port 443 (HTTPS). Although port 80 is naturally less secure, it was enabled to allow web access. To reduce risks, all HTTP traffic on port 80 was automatically redirected to port 443, ensuring communication with the research object was encrypted with SSL. SSL encryption involves a handshake to start the connection, encrypts data with cryptographic keys (public and private), and verifies data integrity to prevent tampering during transmission. Additionally, SSL offers authentication to establish trust between parties. The main reason for permitting only port 443 was to protect sensitive information during internet transmission, making sure that only the intended recipient could interpret the data.

Security assessment in this study was conducted over the internet, where, as shown in Fig. 4, only ports 80 and 443 were exposed.

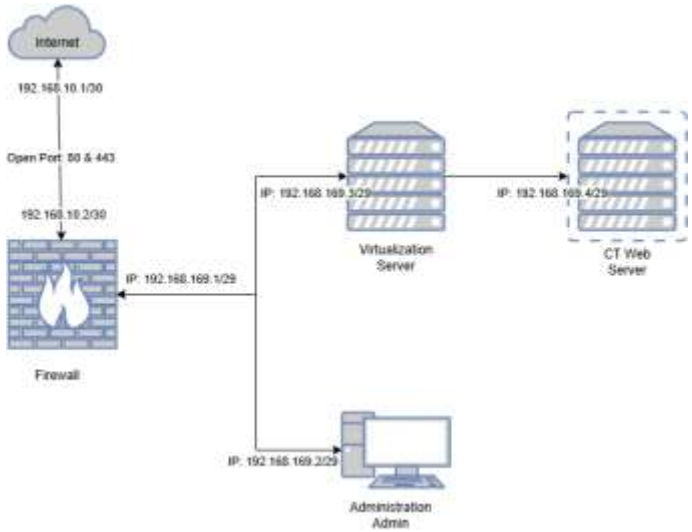

Fig. 4. Assessment framework

To enhance security for login and registration, the researcher implemented Google CAPTCHA, as shown in Fig. 5. This feature ensured that only human users could access these pages, reducing the risk of brute force attacks [20], [21].

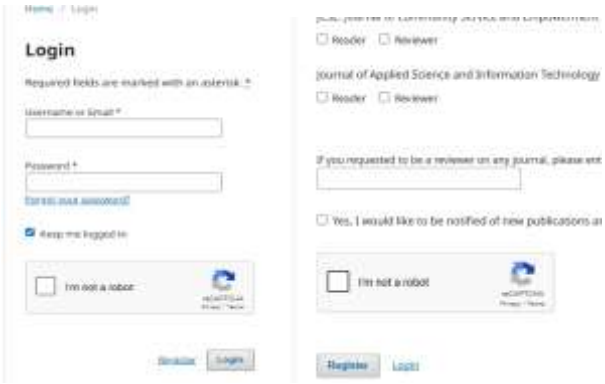

Fig. 5. Implement CAPTCHA on login and registration forms

Furthermore, the researcher set file permissions (chmod) on the OJS directory within Webmin. The OJS folder was assigned chmod 755, with the owner having write (4) + read (2) + execute (1) = 7, and both group users and the public having read (2) + execute (1) = 5. All files inside the folder were given chmod 644, enabling the owner to read (4) + write (2) = 6, while the public was only allowed to read (4) = 4.

The OJS PKP application also included development configuration files, such as composer.json and composer.lock, which should not be publicly accessible because they contain information about libraries and their versions used in the OJS PKP system. To protect these files, the researcher set permissions to chmod 640, allowing the owner read (4) plus write (2) = 6, while denying public users any permission to view or execute these files.

2.5. Vulnerability Assessment

The vulnerabilities identified and their severity levels were assessed using the Common Vulnerability Scoring System (CVSS) [22], [23], [24] through the official platform at https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator. The vulnerability ratings are detailed in Table 2.

Table 2. CVSS v3.x Ratings

Severity	Severity Score Range
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

3. Results and Discussion

The security gaps and vulnerabilities identified in this study, along with their severity levels, were assessed using the Common Vulnerability Scoring System (CVSS) via the official CVSS v3 calculator available at https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator. Table 3 summarizes the assessment results and vulnerability evaluations. These findings offer important insights into the system's security posture and highlight the areas that need targeted mitigation to enhance the overall resilience and robustness of the OJS environment.

Table 3. Reporting

Vulnerability	Description	CVSS	Risk assessment categories
Session Hijacking dengan Cookie	A successful login was achieved by exploiting a session cookie without requiring user credentials	9.1	Critical — the highest priority for security improvement
Brute Force attack	The system is equipped with a CAPTCHA mechanism; however, no restrictions are enforced on the number of login attempts. As a result, the system is unable to block brute-force attackers	6.5	Medium — requires urgent security measures

Vulnerability	Description	CVSS	Risk assessment categories
HTML form without CSRF protection	The URI index.php/jastech/login? redirects to the page index.php/jastech/login because, in this study, the trailing question mark (?) does not alter the functionality of the endpoint. In cases where a question mark is present without accompanying parameters, the server disregards the empty query string since no additional information is provided for processing. Consequently, the server treats the URI as equivalent to its parameter less form and returns the same response page to the user	4.3	Medium–Low — a necessary remediation, especially when the form is used for sensitive transactions
Possible sensitive directories	The directory /lib/pkp/classes/config was found to be exposed to the public; however, when accessed, it returns a "Forbidden Access" message, indicating restricted access. Nevertheless, directory permissions should be further hardened to prevent directory listing or probing attempts by automated security scanning tools	5.3	Medium — reducing the attack surface for reconnaissance is recommended
Upload File pada halaman submission	File upload assessment demonstrated that files with a .php extension, which potentially act as backdoors, were successfully uploaded. However, the system automatically renamed such files to .txt, thereby preventing execution of the uploaded .php scripts	5.3	Medium — mitigation is required to prevent escalation

Session Hijacking via Cookie — A CVSS score of 9.1 (Critical) signifies a highly exploitable vulnerability with severe consequences. If a session cookie is stolen, an attacker can hijack the user session without valid credentials, impersonate the victim, access personal data, and carry out malicious actions, especially if the compromised account has elevated privileges. Because this vulnerability affects session authentication directly, its business impact is considered critical, threatening privacy, transaction integrity, and regulatory compliance.

Brute Force Attack on Login Form (CAPTCHA Enabled, No Login Attempt Limit) — A CVSS score of 6.5 (Medium) indicates a moderate level of risk. While CAPTCHA helps reduce automated attacks, the lack of login attempt restrictions—such as rate limiting, account lockout, or progressive delays—enables attackers to make repeated login attempts, including distributed brute force attacks using IP rotation or proxies. This leaves accounts with weak passwords vulnerable to compromise. These findings support previous reports [18], confirming that Open Journal Systems (OJS) does not have an effective mechanism to limit repeated login attempts.

HTML Form Without CSRF Protection (index.php/jastech/login?) — CVSS 4.3 (Medium–Low) score of approximately 4.3 indicates a medium-to-low risk depending on context. Cross-Site Request Forgery (CSRF) could allow attackers to force authenticated users to perform unintended actions (e.g., changing settings). For a login endpoint, the impact is usually lower since login itself is not typically a state-changing action for other users. However, if the form handles sensitive operations, the risk should be considered.

Exposure of Potential Sensitive Directories (/lib/pkp/classes/config) — CVSS 5.3 (Medium) The exposure of directory paths (such as folder names) without content access (HTTP 403 Forbidden) is classified as a medium risk with a score of 5.3. Although direct exploitation is limited, revealing structural information aids reconnaissance and application mapping, which could speed up other exploits if configurations are modified. If sensitive configuration files (like credentials) are exposed, the risk could increase significantly.

File Upload on Submission Page (.php to .txt, Non-Executable) — CVSS 5.3 (Medium): The upload endpoint accepts potentially malicious files, but the server renames them (e.g., from .php to .txt) or forces them to download, preventing immediate execution. A score of 5.3 indicates limited direct impact; however, it still poses a persistent risk. Malicious files (e.g., potential webshells or payloads) stored in the system could be exploited later if server settings change or files are moved into an executable directory.

3.1. Mitigation

1. Brute Force Attack Mitigation

   Mitigation against brute force attacks was implemented by enforcing login attempt limits. Each IP address was limited to three consecutive login attempts. After three failed tries, the IP was automatically blocked for one hour, as shown in Fig. 6. This rate-limiting (lockout) system

effectively stops automated login attempts, since brute force attacks depend on numerous quick tries. Limiting attempts forces attackers to spend more time or resources (such as IP rotation) to succeed.

The probability of a breach is significantly reduced because setting a low threshold of three attempts minimizes the chances of discovering weak passwords. Security was further improved by using Fail2ban, which detects repeated failed login attempts, triggers logging and alerts, and allows security teams to respond more promptly. Fail2ban was successfully implemented as a login attempt limiter, effectively preventing unauthorized login attempts (Fig. 6) [25].

Fig 6. Fail2ban in Webmin banning IP addresses after failed login attempts

Additionally, login restrictions were strengthened by requiring email verification. Only users who had verified their email addresses were allowed to log in. This was enforced by enabling the configuration setting require_validation = On in the OJS PKP config.inc.php file.

2. Host Header Attack Mitigation

To mitigate host header attacks, the website domain and subdomains were explicitly defined in the configuration. In OJS PKP, this was achieved by setting allowed hosts in the config.inc.php file according to the official journal domain:

allowed_hosts = '["jurnal.universitasputrabangsa.ac.id", "www.jurnal.universitasputrabangsa.ac.id"]'

This configuration ensures that only requests directed to the specified domain and subdomains are accepted, thereby reducing the risk of host header manipulation.

3. BREACH Attack Mitigation

Mitigation against BREACH attacks was performed by enforcing the use of the latest TLS encryption standards [26] and disabling port 80 for the web server. To strengthen security claims and improve system resilience, OJS administrators are advised to conduct regular configuration audits, test filtering mechanisms using controlled payloads, and apply the principle of least privilege to both file system permissions and process execution rights.

4. Conclusion

The potential security vulnerabilities of Open Journal Systems (OJS) managed through Webmin as a system administration tool mainly result from open ports and web server misconfigurations. This study's proposed OJS web server setup using Webmin shows its ability to improve system security against attacks targeting ports 80 and 443.

This study's novelty lies in evaluating the security of the latest version of Open Journal Systems (OJS) 3.4.0, managed through the Webmin web-based system administration tool. It uses the ISSAF framework as a methodological approach to identify, analyze, and address potential security vulnerabilities caused by integrating OJS and Webmin. The research shows that the main attack vectors targeting the OJS web server come from the external network via ports 80 and 443. Key vulnerabilities identified include Session Hijacking through cookies (CVSS 9.1), exposed configuration folder structures (CVSS 5.3), repeated login attempts without IP blocking or account lockout mechanisms (CVSS 6.5), and PHP file upload vulnerabilities (CVSS 5.3). Although OJS PKP tries to prevent uploaded PHP files by renaming them with a .txt extension, such files can still pose future risks if exploited by unauthorized users.

To improve the security of OJS web servers managed with Webmin, several strategies are suggested:

1. Implement Fail2ban to prevent repeated login attempts by enforcing account lockouts and blocking attacker IP addresses.
2. Configure explicit domain and subdomain restrictions in the config.inc.php file using the allowed_hosts parameter to prevent unauthorized access.
3. Enforce the latest TLS encryption standards while disabling port 80 for web services, thereby ensuring that all communications occur over secure channels.

These measures collectively strengthen the resilience of OJS against common web-based attacks and provide a more robust security baseline for academic journal management systems.

Attacks targeting web servers have become increasingly large and sophisticated, making manual post-incident analysis more difficult. Future research on the security of OJS within the Webmin server management environment should focus on implementing preventive measures through real-time attack detection and prevention techniques, using Artificial Intelligence (AI) [27], [28] and Machine Learning (ML) [29] approaches.

Such intelligent systems are expected to improve the ability of Webmin-based OJS environments to automatically detect abnormal traffic patterns, identify zero-day exploits, and adaptively respond to changing cyber threats. Combining AI-driven anomaly detection with the ISSAF security framework could offer a more proactive, adaptive, and data-driven defense approach, thereby enhancing the overall resilience and reliability of the OJS infrastructure against new attack methods.

Acknowledgment

This research was supported by the Ministry of Higher Education, Science, and Technology (Kemdiktisaintek), Republic of Indonesia, through the Beginner Lecturer Research (PDP) Grant scheme under Contract No. 056/LL6/PL/AL.04/2025. The authors would also like to acknowledge Universitas Putra Bangsa for its institutional support. In addition, the authors gratefully recognize the contributions, assistance, and constructive feedback from colleagues and all parties involved in completing this study and preparing this manuscript.

References

[1] G. Guntoro, L. Costaner, and M. Musfawati, “Analisis Keamanan Web Server Open Journal System (OJS) Menggunakan Metode Issaf Dan Owasp (Studi Kasus Ojs Universitas Lancang Kuning),” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 5, no. 1, p. 45, 2020, doi: 10.29100/jipi.v5i1.1565.

[2] F. Kristianto, S. Rahman, and S. Bahri, “Analisis Kerentanan Pada Website Servio Menggunakan Acunetix Web Vulnerability,” *JTRISTE*, vol. 9, no. 1, pp. 46–55, 2022, doi: https://doi.org/10.55645/jtriste.v9i1.363.

[3] I. Riadi, A. Yudhana, and Y. W, “Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 853–860, 2020, doi: 10.25126/jtiik.2020701928.

[4] T. Saleh, M. Malkawi, Z. Elgammal, A. K. Calayır, and R. Alhajj, “Scenario-Based Cross-Site Request Forgery (CSRF) Attack Simulation,” in *2024 6th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Alkhobar, Saudi Arabia: IEEE, 2024, pp. 1–5. doi: https://doi.org/10.1109/ISAECT64333.2024.10799863.

[5] N. Albalawi, N. Alamrani, R. Aloufi, M. Albalawi, A. Aljaedi, and A. R. Alharbi, “The Reality of Internet Infrastructure and Services Defacement: A Second Look at Characterizing Web-Based Vulnerabilities,” *Electron.*, vol. 12, no. 12, 2023, doi: 10.3390/electronics12122664.

[6] D. Y. Perwej, S. Qamar Abbas, J. Pratap Dixit, D. N. Akhtar, and A. Kumar Jaiswal, “A Systematic Literature Review on the Cyber Security,” *Int. J. Sci. Res. Manag.*, vol. 9, no. 12, pp. 669–710, 2021, doi: 10.18535/ijsrm/v9i12.ec04.

[7] Willy, W. S. Priatna, S. R. Manalu, A. M. Sundjaja, and Noerlina, “Development of Review Rating and Reporting in Open Journal System,” *Procedia Comput. Sci.*, vol. 116, pp. 645–651, 2017, doi: https://doi.org/10.1016/j.procs.2017.10.035.

[8] R. P3I, “Kerentanan Keamanan Open Journal System,” UM Surabaya. Accessed: Aug. 09, 2025. [Online]. Available: https://lp2ihki.um-surabaya.ac.id/homepage/news_article?slug=kerentanan-keamanan-open-journal-system

[9] R. Weaver, D. Weaver, and D. Farwood, *Guide to Network Defense and Countermeasures*. Boston,

2014.

[10] M. Ozkan-okay, A. A. Yilmaz, E. Akin, A. Aslan, and S. S. Aktug, "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions," *Electronics*, vol. 12, no. 1333, 2023.

[11] R. Umar, I. Riadi, and M. I. A. Elfatiha, "Analisis Keamanan Sistem Informasi Akademik Berbasis Web Menggunakan Framework ISSAF," *J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 12, no. 1, pp. 280–292, 2023, doi: http://dx.doi.org/10.35889/jutisi.v12i1.1191.

[12] M. Fronita, S. Informasi, S. Teknologi, and U. I. N. S. Riau, "Analisis Celah Keamanan Website Sitasi Menggunakan Vulnerability Assessment," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 9, no. 1, pp. 1–7, 2023, doi: http://dx.doi.org/10.24014/rmsi.v9i1.21823.

[13] Z. Tamin, "Optimalisasi Analisis Keamanan Menggunakan Acunetix Vulnerability Pada Rekam Medis Elektronik," *KESATRIA J. Penerapan Sist. Inf. (Komputer Manajemen)*, vol. 5, no. 4, pp. 1732–1740, 2024, doi: https://doi.org/10.30645/kesatria.v5i4.494.

[14] K. Huda and D. A. Saputri, "Evaluasi Kinerja Open Journal Systems (OJS) dengan Black Box Testing : Studi Kasus pada JITE Universitas Karya Husada," *JITE*, vol. 01, no. 01, 2025.

[15] A. Zirwan, "Pengujian dan Analisis Kemanan Website Menggunakan Acunetix Vulnerability Scanner," *J. Inf. dan Teknol.*, vol. 4, no. 1, pp. 70–75, 2022, doi: 10.37034/jidt.v4i1.190.

[16] E. I. Alwi and L. B. Ilmawan, "Analisis Keamanan Sistem Informasi Akademik (SIAKAD) Universitas XYZ Menggunakan Metode Vulnerability Assessment," *INFORMAL Informatics J.*, vol. 6, no. 3, p. 131, 2021, doi: 10.19184/isj.v6i3.27053.

[17] Rusydi Umar, Imam Riadi, and M. I. A. Elfatiha, "Security Analysis of Web-based Academic Information System using OWASP Framework," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 9, no. 4, Nov. 2024, doi: 10.22219/kinetik.v9i4.2015.

[18] M. N. A. Nur and H. Hijriani, "cPanel Server Hosting Security Against Malware and DDoS Attacks on the Open Journal System Platform," *Sci. J. Informatics*, vol. 11, no. 3, pp. 761–772, 2024, doi: 10.15294/sji.v11i3.11605.

[19] D. Apriyanto and A. Prihanto, "Implementasi Dan Analisis Kinerja Webmin Sebagai Alat Managemen Bind DNS Server Studi Kasus Pada Virtual Private Server," *J. Informatics Comput. Sci.*, vol. 6, no. 4, pp. 1109–1119, 2025.

[20] C. N. Siahaan, M. Rufisanto, R. Nolasco, S. Achmad, and C. R. P. Siahaan, "Study of Cross-Site Request Forgery on Web-Based Application: Exploitations and Preventions," *Procedia Comput. Sci.*, vol. 227, pp. 92–100, 2023, doi: 10.1016/j.procs.2023.10.506.

[21] S. Sivakorn, J. Polakis, and A. D. Keromytis, "I'm not a human: Breaking the Google reCAPTCHA," *Black Hat*, no. i, pp. 1–12, 2016.

[22] M. Nowak, M. Walkowski, and S. Sujecki, "Conversion of CVSS Base Score from 2.0 to 3.1," in *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2021, pp. 1–3. doi: 10.23919/SoftCOM52868.2021.9559092.

[23] A. Younis, Y. K. Malaiya, and I. Ray, "Evaluating CVSS base score using vulnerability rewards programs," *IFIP Adv. Inf. Commun. Technol.*, vol. 471, pp. 62–75, 2016, doi: 10.1007/978-3-319-33630-5_5.

[24] D. Zou, J. Yang, Z. Li, H. Jin, and X. Ma, "AutoCVSS: An Approach for Automatic Assessment of Vulnerability Severity Based on Attack Process," in *International Conference on Green, Pervasive, and Cloud Computing*, R. Miani, L. Camargos, B. Zarpelão, E. Rosas, and R. Pasquini, Eds., Cham: Springer International Publishing, 2019, pp. 238–253. doi: https://doi.org/10.1007/978-3-030-19223-5_17.

[25] R. Ramadhan, J. Latuny, and S. J. Litiloly, "Perancangan Pengamanan Server Apache Menggunakan Firewall Iptables Dan Fail2Ban," *J. ISOMETRI*, vol. 1, no. 1, pp. 9–15, 2022, doi: 10.30598/isometri.2022.1.1.9-15.

[26] H. Krawczyk and H. Wee, "The OPTLS Protocol and TLS 1.3," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 81–96. doi: 10.1109/EuroSP.2016.18.

[27] O. Chakir *et al.*, "An empirical assessment of ensemble methods and traditional machine learning techniques for web-based attack detection in industry 5.0," *J. King Saud Univ. - Comput. Inf. Sci.*, vol.

35, no. 3, pp. 103–119, 2023, doi: 10.1016/j.jksuci.2023.02.009.

[28] A. Razaque, S. Hariri, A. M. Alajlan, and J. Yoo, "A comprehensive review of cybersecurity vulnerabilities, threats, and solutions for the Internet of Things at the network-cum-application layer," *Comput. Sci. Rev.*, vol. 58, p. 100789, 2025, doi: https://doi.org/10.1016/j.cosrev.2025.100789.

[29] M. Ramadan, B. Osama, M. Zaher, H. Mansour, and W. El Sersi, "Enhancing Web Security: A Comparative Analysis of Machine Learning Models for CSRF Detection," in *2024 Intelligent Methods, Systems, and Applications (IMSA)*, Giza, Egypt: IEEE, 2024, pp. 18–25. doi: 10.1109/IMSA61967.2024.10652629.