CENTIVE
VOL. 4, NO. 1, PP.1154-1162, DEC 2023
ISBN : 978-602-53004-0-0

Conference on Electrical Engineering, Informatics, Industrial Technology, and Creative Media 2023

Integrating OWASP and MOORA for Comprehensive Vulnerability Assessment and Prioritization in Educational Web Applications

Bisma Nanda Satria[1], Soetam Rizky Wicaksono*[2], Rudy Setiawan[3]

Information System Study Program, Universitas Ma Chung
Malang - Indonesia

[1] 321910016@student.machung.ac.id
[2] soetam.rizky@machung.ac.id
[3] rudy.setiawan@machung.ac.id

Dikirim pada 22-11-2024, Direvisi pada 27-11-2024, Diterima pada 04-12-2024

Abstract

Web applications in higher education institutions operating under the "ac.id" domain play a critical role in managing academic data and services but are highly susceptible to cybersecurity threats. This study aims to assess vulnerabilities in five selected university websites using the OWASP Top 10 framework and prioritize them with the MOORA decision-support system. The research employs a systematic methodology comprising reconnaissance, active scanning with tools like OWASP ZAP and Nessus, exploitation, and validation to identify vulnerabilities. MOORA is applied to rank vulnerabilities based on severity, frequency, impact on confidentiality, integrity, and availability (CIA), and ease of remediation. The findings reveal critical vulnerabilities, such as Remote Code Execution (RCE) and Sensitive Data Exposure, alongside medium risks like missing anti-CSRF tokens and insecure configurations. Website 1 ranked highest in vulnerability severity, demanding immediate remediation, while other websites exhibited medium to low vulnerabilities that still require attention. By integrating OWASP with MOORA, the study provides an objective, data-driven approach to cybersecurity prioritization. The integration of OWASP and MOORA provided a structured, objective approach to vulnerability management, ensuring that resources are allocated efficiently to address the most pressing security concerns. This methodology underscores the importance of adopting comprehensive security frameworks and decision-support systems to enhance the cybersecurity posture of educational institutions.

Keywords: Vulnerability Assessment,Web Application Security,OWASP,MOORA,Decision-Support Systems

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

Corresponding Author:
Soetam Rizky Wicaksono
Program Studi Sistem Informasi, Universitas Ma Chung, Villa Puncak Tidar Blok N no. 1, Doro, Karangwidoro, Kec. Dau, Kabupaten Malang, Jawa Timur 65151 Indonesia Email: soetam.rizky@machung.ac.id

I. INTRODUCTION

In the modern era of the internet, web applications have become a critical necessity for educational institutions, particularly those utilizing “ac.id” domains or subdomains. These web applications streamline data management and services, fostering efficiency within academic environments. However, the vulnerabilities inherent in web applications pose significant risks to cybersecurity, threatening the integrity, confidentiality, and availability of information [1]. These risks not only compromise the functionality of higher education institutions but also jeopardize the quality of academic operations. According to BSSN data (2022), Indonesia experienced 370.03 million cyberattacks, with a notable increase in attacks targeting university websites compared to previous years [2]. This pressing challenge highlights the urgent need for research focused on identifying and addressing vulnerabilities in web applications operating under the “ac.id” domain [3], [4].

The Open Worldwide Application Security Project (OWASP) serves as a globally recognized framework for addressing common security vulnerabilities in web applications, particularly through its OWASP Top 10 guidelines [5], [6]. Employing OWASP as a reference for this study enables a structured and comprehensive approach to vulnerability analysis. Understanding and mitigating these vulnerabilities is crucial for enhancing the security of web applications [7]. However, addressing these vulnerabilities demands a multi-faceted approach, considering the complexities and varying severities of the potential threats [8].

This research aims to analyze vulnerabilities in “ac.id” web applications by integrating OWASP Top 10 standards with a Multi-Objective Optimization-based Ratio Analysis (MOORA) decision-support system. This integration is designed to optimize decision-making processes, enabling effective prioritization of remediation strategies. By leveraging MOORA, this study provides a scalable solution for ranking vulnerabilities, ensuring objective assessments that can inform actionable security measures. The combination of analytical rigor and decision support promises to bridge gaps in vulnerability management for higher education institutions [9].

Furthermore, the MOORA method's adaptability in multi-criteria decision-making scenarios underscores its relevance to this research. Unlike TOPSIS, which emphasizes proximity to an ideal solution, or AHP, which involves hierarchical complexity [10], MOORA offers superior sensitivity and flexibility in weight determination without relying on complex hierarchical frameworks. This makes it an ideal tool for analyzing the multi-dimensional challenges posed by web application vulnerabilities [11]. The findings are expected to contribute significantly to the development of more robust cybersecurity practices for educational institutions, mitigating risks associated with sensitive and critical data breaches [12].

One of the key goals of this research is to carry out an exhaustive vulnerability assessment of online applications. The MOORA decision-support system will serve as the evaluation instrument, and the OWASP Top 10 standards will serve as the analytical framework. In order to improve the overall cybersecurity posture of educational institutions that fall within the "ac.id" domain, the purpose of this study is to identify important vulnerabilities, prioritise remedial steps, and improve overall cybersecurity posture. The purpose of this research is to equip decision-makers with actionable insights necessary to effectively increase web application security. This will be accomplished by offering a structured and objective evaluation process [13].

To maintain focus and feasibility, this study imposes specific boundaries. The scope is limited to analyzing vulnerabilities in five selected web applications operating within the “ac.id” domain in East Java, Indonesia. The evaluation is based on the OWASP Top 10 framework [5], focusing solely on identifying and categorizing vulnerabilities without delving into specific remediation implementations. Additionally, the MOORA system be employed exclusively for prioritizing and ranking vulnerabilities, ensuring an objective assessment of security risks across the selected applications [11]. These limitations are intended to streamline the research process and ensure the depth and reliability of the findings within a well-defined context.

II. RESEARCH METHOD

The Open Web Application Security Project (OWASP) is an internationally recognized organization dedicated to improving the security of software, particularly web applications [5]. It provides resources, tools, and guidelines aimed at identifying and mitigating common vulnerabilities that can compromise the safety of web applications. Among its most prominent contributions is the OWASP Top 10, a standard reference document that outlines the most critical security risks facing web applications. This framework is widely adopted by developers, security professionals, and organizations as a benchmark for designing and maintaining secure applications [14].

One of the core principles of OWASP is its focus on raising awareness about the importance of secure software development. The OWASP Top 10 is updated periodically to reflect emerging threats in the cybersecurity landscape [5]. It serves as a practical guide for assessing and addressing risks, covering issues such as injection attacks, broken authentication, security misconfigurations, and insufficient logging. By providing actionable insights, OWASP bridges the gap between theoretical security concepts and real-world applications, making it a cornerstone for vulnerability assessments.

A significant advantage of the OWASP methodology is that it provides a comprehensive explanation of every risk that has been detected. For instance, the most common problem, known as Injection (A01), defines the manner in which malicious programs might take advantage of vulnerabilities in order to control the behaviour of applications. In a similar manner, Broken Authentication (A02) draws attention to vulnerabilities that hackers could exploit in order to get access to user accounts. There are explanations of potential attack vectors, examples of exploitations, and recommended remediation procedures included in

each of the categories that make up the OWASP Top 10 security vulnerabilities. In addition to assisting in the identification of vulnerabilities, this organised method also provides organisations with direction for the implementation of preventative measures [5].

The OWASP Top 10 framework's adaptability makes it particularly relevant for this study. It can be applied across diverse web applications, providing a consistent methodology for evaluating security risks. In this research, the OWASP Top 10 serve as the foundation for assessing vulnerabilities in educational institutions' web applications. Its comprehensive nature ensures that critical threats are systematically identified and categorized, enabling a focused and impactful analysis of cybersecurity risks. By integrating OWASP's insights with decision-support tools such as MOORA, this study aims to deliver a robust and actionable evaluation framework.

The **OWASP Top 10** is a widely recognized list that highlights the most critical security risks to web applications, serving as a fundamental resource for developers and security professionals. The 2021 edition identifies the following key vulnerabilities:

1. **Broken Access Control**

   Access control mechanisms are essential for ensuring that users can only perform actions and access resources for which they have explicit permissions. When these mechanisms are improperly implemented or misconfigured, it leads to broken access control vulnerabilities. Such flaws can allow unauthorized users to gain access to restricted functionalities or data, leading to potential data breaches, unauthorized operations, and privilege escalation. Common manifestations include:

   - **Insecure Direct Object References (IDOR):** Occurs when an application exposes references to internal implementation objects, such as files or database keys, allowing attackers to manipulate these references to access unauthorized data.
   - **Forced Browsing:** Involves accessing parts of a website that are not linked or are restricted, by directly entering URLs.
   - **Privilege Escalation:** Happens when a user gains higher-level permissions than intended, often due to flaws in role-based access controls.

   The implementation of robust access control rules that enforce the principle of least privilege is essential for mitigating vulnerabilities caused by failed access control. These policies should ensure that users have just the permissions that are required for their roles of responsibility. Regular audits and tests ought to be carried out in order to ensure that access restrictions are operating in accordance with their intended purpose. In addition, the utilisation of security mechanisms such as session management, the appropriate validation of user inputs, and the guarantee that sensitive resources are not immediately accessible without the right authorisation can be of assistance in preventing unauthorised access. Additionally, developers should avoid hardcoding roles and permissions within the application code. Instead, they should choose for access control techniques that are centralised and customisable. Through the implementation of these measures, organisations have the ability to dramatically lessen the likelihood of unauthorised access and safeguard sensitive information against the possibility of breaches.

2. **Cryptographic Failures**

   Cryptographic failures, previously referred to as sensitive data exposure, occur when critical data is inadequately protected through encryption or hashing mechanisms. Such failures can lead to unauthorized access, data breaches, and non-compliance with data protection regulations. Common issues include:

   - **Weak or Deprecated Algorithms:** Utilizing outdated cryptographic algorithms that are susceptible to attacks, such as MD5 or SHA-1.
   - **Improper Key Management:** Storing encryption keys in insecure locations or failing to rotate them regularly, increasing the risk of key compromise.
   - **Lack of Encryption:** Transmitting or storing sensitive data in plaintext, making it easily accessible to unauthorized parties.

   To prevent cryptographic failures, organizations should adopt strong, industry-accepted encryption algorithms and ensure proper implementation. Effective key management practices are essential, including secure storage, regular rotation, and access controls to limit key exposure. Additionally, all sensitive data should be encrypted both in transit and at rest to protect against interception and unauthorized access. Regular security assessments and staying updated with the latest cryptographic standards can help identify and remediate potential weaknesses in data protection strategies.

3. **Injection**

Injection vulnerabilities occur when untrusted data is sent to an interpreter as part of a command or query, allowing attackers to execute unintended commands or access unauthorized data. Common types include SQL injection, OS command injection, and LDAP injection. These vulnerabilities can lead to data breaches, loss of data integrity, and unauthorized system access. For example, in SQL injection, an attacker can manipulate a SQL query by injecting malicious input, potentially retrieving or altering database information without authorization. To mitigate injection risks, it's essential to use parameterized queries, prepared statements, and input validation to ensure that user-supplied data cannot alter intended commands or queries. Regular code reviews and security testing can help identify and address injection flaws before they become exploitable.

4. **Insecure Design**
   Insecure design refers to flaws in the architecture and design of an application that create security vulnerabilities. Unlike implementation bugs, these are systemic issues that arise from inadequate security considerations during the design phase. Examples include lack of threat modeling, insufficient security controls, and failure to anticipate potential attack vectors. To address insecure design, it's crucial to integrate security into the software development lifecycle from the outset. This includes conducting threat modeling, applying secure design principles, and performing security-focused code reviews. By proactively considering security during the design phase, organizations can build more resilient applications and reduce the likelihood of vulnerabilities.
5. **Security Misconfiguration**
   Security misconfiguration occurs when security settings are improperly implemented, leaving applications vulnerable to attacks. This can include default configurations, incomplete configurations, or ad-hoc changes that are not thoroughly tested. Common issues involve unnecessary features enabled, default accounts with unchanged passwords, and verbose error messages that reveal sensitive information. To prevent security misconfigurations, organizations should establish secure installation processes, disable unnecessary features, enforce strong authentication mechanisms, and regularly review and update configurations. Automated tools can assist in detecting misconfigurations and ensuring compliance with security best practices.
6. **Vulnerable and Outdated Components**
   Using components with known vulnerabilities can compromise the security of applications. This includes libraries, frameworks, and other software modules that are no longer supported or have unpatched security flaws. Attackers can exploit these vulnerabilities to execute arbitrary code, steal data, or disrupt services. To mitigate this risk, it's essential to maintain an inventory of all components, monitor them for known vulnerabilities, and apply updates or patches promptly. Additionally, organizations should consider the security implications of third-party components and prefer those with a strong track record of security maintenance.
7. **Identification and Authentication Failures**
   Identification and authentication failures occur when applications incorrectly implement mechanisms to identify and authenticate users, leading to unauthorized access. Issues include weak password policies, flawed multi-factor authentication, and improper session management. These failures can result in account takeovers, data breaches, and privilege escalation. To address these vulnerabilities, applications should enforce strong password policies, implement robust multi-factor authentication, securely manage sessions, and ensure proper validation of credentials. Regular security assessments can help identify and remediate authentication-related weaknesses.
8. **Software and Data Integrity Failures**
   Software and data integrity failures occur when applications rely on untrusted sources for software updates, critical data, or continuous integration/continuous deployment (CI/CD) pipelines without verifying their integrity. This can lead to unauthorized access, data breaches, or system compromise. For example, if an application fetches updates over an unsecured channel without verifying their authenticity, attackers can inject malicious code into the software. To mitigate these risks, it's essential to implement digital signatures for code and data, enforce strict access controls, and regularly audit CI/CD pipelines to ensure only authorized changes are deployed. Additionally, monitoring systems for unexpected changes can help detect potential integrity failures.
9. **Security Logging and Monitoring Failures**
   Security logging and monitoring failures refer to the absence or inadequacy of logging mechanisms that capture essential security-related events. Without proper logging, detecting and responding to security incidents becomes challenging, allowing attackers to exploit vulnerabilities without notice. Common issues include insufficient log storage, lack of monitoring for suspicious activities, and failure to analyze logs regularly. To address these failures, organizations should

implement comprehensive logging strategies that capture critical events, ensure logs are securely stored and protected from tampering, and establish real-time monitoring and alerting systems. Regular analysis of logs can help in early detection of potential security incidents and facilitate prompt response.

10. **Server-Side Request Forgery (SSRF)**
    Server-Side Request Forgery (SSRF) occurs when an application fetches a remote resource without validating the user-supplied URL, allowing attackers to make unauthorized requests to internal or external services. This can lead to information disclosure, unauthorized actions, or even complete system compromise. For instance, an attacker might exploit SSRF to access internal metadata services within cloud environments, potentially exposing sensitive information. To prevent SSRF vulnerabilities, applications should validate and sanitize all user inputs that generate requests, implement allowlists for acceptable URLs, and restrict network access where possible. Additionally, disabling unused protocols and services can reduce the attack surface for SSRF exploits.

Understanding and addressing these vulnerabilities is crucial for enhancing the security posture of web applications. Implementing best practices, conducting regular security assessments, and fostering a security-aware development culture can significantly mitigate the risks associated with these common vulnerabilities. By addressing these vulnerabilities, organizations can significantly enhance their web application security posture.

In selecting the five university websites for this study, we focused on higher education institutions within East Java, Indonesia. Initially, a broad scanning process was conducted across various "ac.id" domains to assess general security postures. This preliminary assessment identified 25 websites exhibiting significant vulnerabilities. Subsequently, a detailed evaluation using the OWASP Top 10 framework was performed on these 25 sites to analyze specific security weaknesses. Based on the severity and frequency of identified vulnerabilities, five websites were selected for an in-depth analysis and prioritization. This systematic selection process ensured that the study focused on websites with the most critical security concerns, providing a targeted approach to vulnerability assessment and remediation.

The methodology employed for vulnerability testing began with passive scanning during the reconnaissance phase, where technologies and frameworks used by the websites were identified [1], [15]. Subsequently, active scanning was performed to detect potential vulnerabilities, using tools configured to align with OWASP standards. The exploitation phase followed, where detected vulnerabilities were manually verified to ensure the absence of false positives. This multi-layered approach ensured a comprehensive assessment of each website's security posture [16], [17].

Application of MOORA in Analyzing OWASP Results

The **Multi-Objective Optimization on the Basis of Ratio Analysis (MOORA)** method is a prominent tool within Decision Support Systems (DSS), particularly effective for tackling complex decision-making scenarios involving multiple criteria. MOORA's integration into DSS enhances the system's capability to evaluate various alternatives against a set of criteria, facilitating objective and data-driven decisions.

In the context of DSS, MOORA operates by normalizing decision matrices, which allows for the comparison of different units and scales across criteria. This normalization is followed by the application of weights to each criterion, reflecting their relative importance in the decision-making process. The method then computes a composite score for each alternative, enabling the ranking of options from most to least favorable.

The application of MOORA within DSS has been demonstrated across various domains. For instance, in educational settings, MOORA has been utilized to develop decision support systems for selecting the best teachers, aiding in objective evaluations based on predefined criteria. Similarly, in the corporate sector, MOORA has been applied to assist in employee evaluations, ensuring that selections are made based on comprehensive and quantifiable data. By incorporating MOORA, DSS can handle both qualitative and quantitative data, providing a structured approach to decision-making that reduces subjectivity and enhances transparency. This integration is particularly beneficial in environments where decisions must be justified and documented, as MOORA offers a clear methodology for evaluating and ranking alternatives.

The **Multi-Objective Optimization on the Basis of Ratio Analysis (MOORA)** method holds a significant position within the realm of **Multi-Criteria Decision Making (MCDM)**. MCDM encompasses a set of methodologies designed to evaluate and prioritize multiple conflicting criteria, facilitating informed and balanced decision-making in complex scenarios.

MOORA contributes to MCDM by offering a systematic approach that simplifies the evaluation process through its ratio analysis technique. It enables decision-makers to assess various alternatives against multiple criteria by normalizing performance ratings and applying a ratio system to determine the most favorable options. This method enhances the objectivity and transparency of the decision-making process, making it a valuable tool in fields such as supplier selection, investment analysis, and resource allocation.

The Multi-Objective Optimization based on Ratio Analysis (MOORA) method provides a structured approach to decision-making, especially in scenarios involving multiple criteria [18]. In this study, MOORA is applied to the results generated from the OWASP Top 10 vulnerability analysis to prioritize and rank the security risks identified in web applications. By evaluating each vulnerability based on predefined criteria, such as severity, frequency, and impact, MOORA offers a systematic way to determine which vulnerabilities pose the highest risk and require immediate attention.

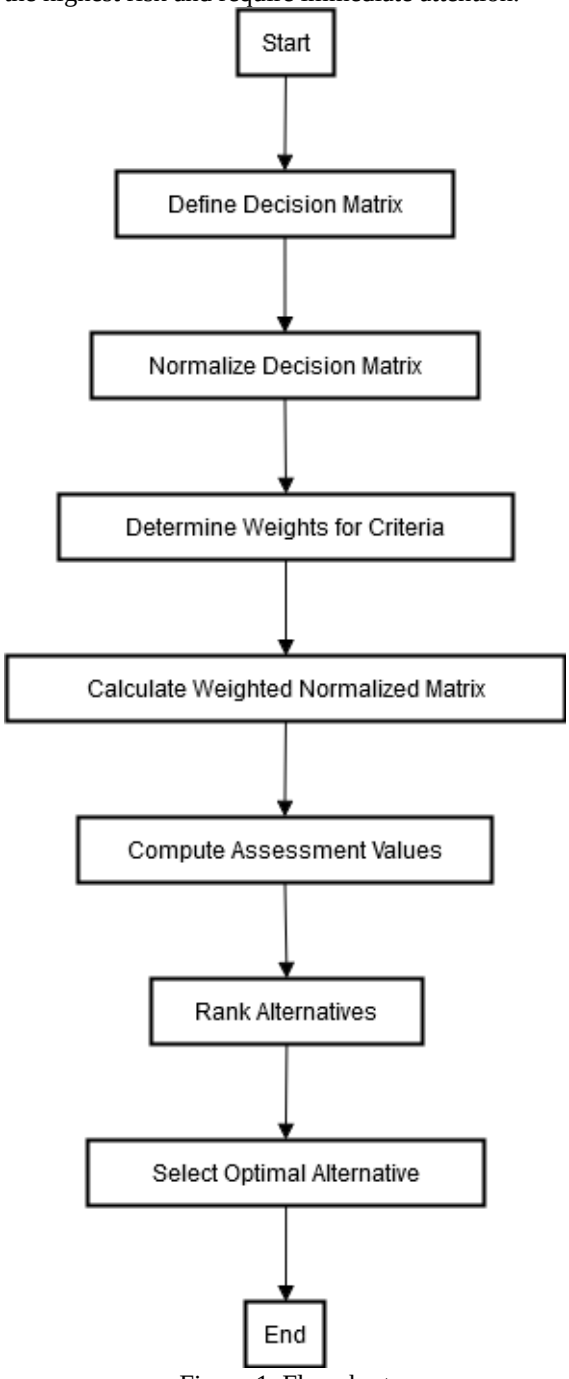

Figure 1. Flowchart

The MOORA process begins by compiling the identified vulnerabilities into a decision matrix. Each row represents a specific vulnerability (e.g., Injection, Broken Authentication), while each column corresponds to a criterion, such as impact on confidentiality, likelihood of exploitation, or remediation complexity. The data for this matrix is sourced from the OWASP analysis results, where vulnerabilities are quantified based on their characteristics and risk levels. Once the matrix is constructed, normalization is performed to ensure that all criteria are measured on a comparable scale, allowing for fair evaluation across different metrics.

Next, a weighted normalization is applied by assigning relative importance to each criterion based on its significance in the context of web application security. For example, vulnerabilities that critically impact data integrity may receive higher weights than those affecting availability. The MOORA method then calculates an optimization score for each vulnerability by combining the weighted criteria. These scores are used to rank the vulnerabilities, highlighting the most critical issues. This systematic prioritization enables decision-makers to allocate resources efficiently, focusing first on addressing the highest-priority vulnerabilities identified through the OWASP analysis. The integration of MOORA ensures that the evaluation process remains objective, data-driven, and aligned with organizational security goals. Each vulnerability was assessed based on multiple criteria, including severity, frequency, impact on confidentiality, integrity, and availability (CIA), and ease of remediation [19].

III. RESULTS AND DISCUSSION

The vulnerability assessment of five selected university websites, all operating under the "ac.id" domain, was conducted using the OWASP framework. The testing aimed to identify vulnerabilities based on the OWASP Top 10 standards, emphasizing critical risks such as Injection, Broken Authentication, and Security Misconfigurations [13]. Each website was evaluated through a systematic process involving reconnaissance, scanning, exploitation, and validation. Tools like OWASP ZAP and Nessus were utilized to uncover vulnerabilities and determine their potential impact [20].

The results revealed a range of vulnerabilities across the five websites. Common issues included the absence of HTTP Strict Transport Security (HSTS), use of outdated frameworks or plugins, and lack of anti-CSRF tokens in HTML forms. Additionally, certain vulnerabilities, such as Cross-Site Scripting (XSS) and Sensitive Data Exposure, were observed in specific websites [21]. These findings underscore the critical need for robust security measures tailored to the unique requirements of educational institutions.

Table 1. Vulnerability Testing Results

Website	Critical Vulnerabilities	Medium Vulnerabilities	Low Vulnerabilities
Website 1	Remote Code Execution (RCE)	Vulnerable JS Library, Content Security Policy (CSP)	Missing HSTS, Cookie Without Secure Flag
Website 2	Sensitive Data Exposure (PII Disclosure)	Absence of Anti-CSRF Tokens, Mixed Content	Leaked Server Information, Missing Anti-clickjacking
Website 3	None	Absence of Anti-CSRF Tokens, XSS Vulnerabilities	Mixed Content, Missing HSTS
Website 4	None	Vulnerable JS Library, Absence of Anti-CSRF Tokens	Leaked Server Information, Insecure Cookies
Website 5	None	Absence of Anti-CSRF Tokens, CSP Header Not Set	Missing Anti-clickjacking, Cookie Without Secure Flag

The findings highlight varying security weaknesses across the five websites. Website 1 was found to be the most vulnerable, with critical issues like Remote Code Execution (RCE), which poses severe risks to system integrity and user data. Medium-severity issues, such as outdated JavaScript libraries and missing content security policies, were prevalent in most websites, indicating a broader need for regular updates and compliance with security standards.

Despite the absence of critical vulnerabilities in Websites 3, 4, and 5, the medium and low vulnerabilities identified could still expose them to significant risks if left unaddressed. Common issues, such as missing HSTS and insecure cookies, suggest a lack of focus on fundamental security configurations. These results demonstrate the critical importance of integrating OWASP principles and decision-support systems like MOORA to prioritize and resolve vulnerabilities effectively [11].

The MOORA method was applied to the results from the OWASP vulnerability testing to prioritize the identified vulnerabilities and recommend mitigation actions effectively. A decision matrix was constructed with rows representing vulnerabilities and columns representing these criteria, quantified using the OWASP findings. The decision matrix was then normalized to ensure consistency in the scales of measurement across criteria. Weighted normalization followed, where higher weights were assigned to criteria deemed critical to cybersecurity, such as impact on confidentiality and severity of the vulnerability. The normalized

values were used to calculate the MOORA optimization scores for each website, combining the weighted criteria to generate a single ranking score.

This process provided an objective ranking of the vulnerabilities, with higher scores indicating a greater priority for remediation. Websites with severe vulnerabilities like Remote Code Execution (RCE) ranked highest in terms of priority, while websites with only medium and low vulnerabilities ranked lower. The MOORA framework thus enabled a structured approach to vulnerability management, ensuring that critical risks are addressed first.

Table 2. MOORA Calculation Results

Website	Severity (Weight: 0.4)	Frequency (Weight: 0.3)	Impact (CIA) (Weight: 0.2)	Ease of Remediation (Weight: 0.1)	MOORA Score	Rank
Website 1	0.8	0.7	0.9	0.6	0.78	1
Website 2	0.7	0.9	0.8	0.5	0.76	2
Website 3	0.4	0.6	0.7	0.8	0.59	3
Website 4	0.3	0.5	0.6	0.7	0.51	4
Website 5	0.3	0.4	0.5	0.6	0.46	5

The MOORA results indicate that Website 1 has the highest vulnerability score, prioritizing it for immediate remediation. The presence of Remote Code Execution (RCE) and other severe vulnerabilities significantly contributed to its high score. Website 2, while lacking critical vulnerabilities, scored high due to the frequency and widespread nature of medium vulnerabilities such as PII disclosure and anti-CSRF token absence, which require timely intervention.

Websites 3, 4, and 5 ranked lower, reflecting fewer severe vulnerabilities. However, their medium and low vulnerabilities still pose risks if left unresolved. The prioritization provided by MOORA allows decision-makers to focus resources on the most critical issues, ensuring an efficient and systematic approach to enhancing web application security. This structured ranking also highlights areas where proactive measures can mitigate vulnerabilities before they escalate.

IV. CONCLUSION

The findings from this research underscore the pressing need for a structured and comprehensive approach to managing vulnerabilities in web applications, particularly in educational institutions. The integration of OWASP Top 10 standards with the MOORA decision-support system has demonstrated its effectiveness in identifying, analyzing, and prioritizing vulnerabilities. This approach not only ensures the identification of critical risks like Remote Code Execution (RCE) but also provides a clear framework for ranking vulnerabilities based on severity, frequency, and impact. The results highlight the importance of adopting systematic vulnerability management practices to safeguard sensitive educational data.

The application of MOORA in this study reveals a stark contrast in the security posture of the analyzed websites. While some websites, such as Website 1, exhibited critical vulnerabilities that demand immediate remediation, others showed patterns of medium and low risks that still warrant attention. This prioritization offers a targeted strategy for institutions to allocate resources effectively, addressing high-risk vulnerabilities first while developing longer-term strategies for mitigating less severe issues. The objective ranking system provided by MOORA eliminates bias and ensures that decisions are based on quantifiable criteria, fostering confidence in the remediation process.

This research highlights a broader implication: the dynamic and evolving nature of cybersecurity threats requires institutions to continuously monitor, assess, and improve their systems. Educational institutions must adopt not only robust tools like OWASP but also decision-making frameworks such as MOORA to stay ahead of potential threats. By embedding these practices into their security protocols, institutions can achieve not just compliance with security standards but also a resilient defense against future risks. This study serves as a stepping stone, advocating for the integration of analytical and decision-making tools to strengthen cybersecurity measures in critical sectors.

Future research should focus on expanding the scope of vulnerability assessments to include a larger number of educational institutions and incorporate dynamic testing methodologies, such as penetration testing with real-time monitoring, to evaluate the resilience of web applications against active cyberattacks. Additionally, integrating advanced decision-making models, such as hybrid approaches combining

MOORA with Artificial Intelligence (AI), could enhance the prioritization process by predicting future vulnerabilities based on emerging cyber threats. Exploring the integration of automated patch management systems and assessing their effectiveness in mitigating vulnerabilities identified by frameworks like OWASP would also provide valuable insights for improving institutional cybersecurity practices.

REFERENCES

[1] P. B. Tarigan, "Web Security Testing Guide v4.1," *Journal of Chemical Information and Modeling*, vol. 53, no. 9. pp. 1689–1699, 2013.

[2] F. S. Pratiwi, "BSSN Catat 370,02 Juta Serangan Siber ke Indonesia pada 2022," *Https://Dataindonesia.Id/*. p. 1, 2023.

[3] I. Sulistyowati and R. V. H. Ginardi, "Information Security Risk Management with Octave Method and ISO/EIC 27001: 2013 (Case Study: Airlangga University)," *IPTEK J. Proc. Ser.*, vol. 0, no. 1, pp. 32–38, 2019.

[4] N. Nelmiawati, F. R. Destrianto, and M. A. R. Sitorus, "Manajemen Risiko Ancaman pada Aplikasi Website Sistem Informasi Akademik Politeknik Negeri Batam Menggunakan Metode OCTAVE," *J. Integr.*, vol. 9, no. 1, p. 35, 2018.

[5] OWASP Foundation, "OWASP Risk Rating Methodology." pp. 1–5, 2013.

[6] INFOSECTRAIN, "OWASP Top 10 Vulnerabilities 2021 Revealed," *Infosectrain*. 2021.

[7] OWASP Foundation, "OWASP Vulnerability Management Guide (OVMG)." 2020.

[8] D. Stuttard and M. Pinto, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*, 2nd ed. Indianapolis, IN: Wiley, 2011.

[9] W. K. M. Brauers and E. K. Zavadskas, "The MOORA method and its application to privatization in a transition economy," *Control Cybern.*, vol. 35, no. 2, pp. 445–469, 2006.

[10] V. M. M. Siregar, M. R. Tampubolon, E. P. S. Parapat, E. I. Malau, and D. S. Hutagalung, "Decision support system for selection technique using MOORA method," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1088, no. 1, p. 12022.

[11] S. R. Wicaksono, "Implementation of Decision Support in Mutual Fund Investment Selection using MOORA," *TIERS Inf. Technol. J.*, vol. 4, no. 1, pp. 66–72, Jun. 2023.

[12] A. T. Elliott, "Information technology security." 2003.

[13] OWASP Foundation, "Vulnerability Scanning Tools @ owasp.org." 2020.

[14] O. Foundation, "Infrastructure as Code Security." [Online]. Available: https://snyk.io/product/infrastructure-as-code-security/.

[15] N. Kavyashree, M. C. Supriya, and M. R. Lokesh, "Critical Success Factor Estimation for Software Security in Small and Medium Scale Industry Using AHP and TOPSIS Approach," *Proc. 3rd Int. Conf. Integr. Intell. Comput. Commun. Secur. (ICIIC 2021)*, vol. 4, no. Iciic, pp. 137–147, 2021.

[16] L. Crews, "Guide to OWASP Top 10 Vulnerabilities and Mitigation Methods." 2023.

[17] D. Kennedy, J. O'Gorman, D. Kearns, and M. Aharoni, *Metasploit: The Penetration Tester's Guide*, 1st ed. San Francisco, CA: No Starch Press, 2011.

[18] S. R. Wicaksono and R. Setiawan, "Optimasi Sistem Penilaian Essay Secara Obyektif dengan Menggunakan MOORA," 2024.

[19] D. M. A. Syed, H. Hasan, and M. S. Trigui, "Information Systems Threats and Vulnerabilities," *Int. J. Comput. Appl.*, vol. 89, no. 3, pp. 975–8887, 2014, Accessed: Dec. 11, 2017.

[20] OWASP Foundation, "Vulnerable Dependency Management - OWASP Cheat Sheet Series." [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Vulnerable_Dependency_Management_Cheat_Sheet.html.

[21] G. Weidman, *Penetration Testing: A Hands-On Introduction to Hacking*, 1st ed. San Francisco, CA: No Starch Press, 2014.