About Web Application Development Security: A Review

Rolly Maulana Awangga*, Muhammad Rifqi Daffa Ulhaq

Department of Informatics Engineering, International University of Logistics and Business, Bandung, Indonesia

Abstract

In the digital age, web applications are increasingly exposed to sophisticated security threats that compromise sensitive user data and disrupt service operations. This systematic literature review (SLR) aims to explore the current methodologies, tools and practices used to ensure security in web application development. Based on established frameworks and guidelines for SLRs in software engineering, this study examines academic and industry sources to identify prevalent security vulnerabilities and the corresponding mitigation strategies employed at various stages of the web development lifecycle. Initial searches retrieved over 216 studies from databases such as IEEE Xplore and SpringerLink, from which 38 were rigorously selected based on their relevance to software development models that integrate security considerations. This review provides a comprehensive overview of development techniques that prioritise security, evaluates their effectiveness, and discusses the integration of security measures from the requirements phase through to maintenance. The findings are intended to guide developers, researchers and policy makers in improving the security robustness of web applications, thereby promoting a safer online environment.

Keywords: Web Applications, Security Vulnerabilities, Systematic Literature Review

1. Introduction

Web applications are integral to modern digital interactions, serving as platforms for everything from e-commerce and social networking to governmental and healthcare services. As these applications handle increasingly sensitive data, establishing robust security measures is crucial. The complexity and openness of web environments expose them to a multitude of security threats, necessitating comprehensive protection strategies [1].

The prevalence of web application vulnerabilities continues to rise, paralleling their increasing complexity and ubiquity. Common threats such as cross-site scripting (XSS), SQL injection, and session hijacking not only persist but evolve alongside technological advancements. This dynamic landscape presents continual challenges for developers and security professionals, highlighting the need for ongoing research and innovative defense mechanisms [2].

Systematic Literature Reviews (SLRs) in this field play a crucial role by compiling and synthesizing existing research, thus providing a comprehensive view of current security strategies and their effectiveness. SLRs help identify both the most prevalent security threats and the most effective countermeasures available to practitioners and researchers, aiding in the understanding and development of more effective security protocols [3].

This SLR aims to delve into the breadth of research conducted on web application security, emphasizing significant advancements and pinpointing gaps in current methodologies. The review will scrutinize the various security techniques proposed in the literature, assess their effectiveness, and offer insights into areas needing further investigation. The goal is to furnish a thorough resource that can guide future research efforts, enhance the development of secure web applications, and improve the overall understanding of web application security challenges and solutions [4, 5]

2. Related Work

As web applications continue to evolve, they attract not only greater user engagement but also an increasing number of security threats that exploit their widespread use and inherent complexities. This section reviews recent scholarly contributions that explore various aspects of web application security. From regional security trends and systematic vulnerability assessments to comparative studies on security parameters and methodological advancements in security testing, the following works provide a comprehensive overview of the challenges and solutions that shape the current landscape of web application security. These studies highlight the necessity for ongoing research and adaptation in security strategies to protect against both existing and emerging threats.

2.1. Web Security Trends

Web application security is an evolving field that continually adapts to counteract new and emerging threats. Recent studies have emphasized the need for comprehensive security assessments and the development of robust defenses against a variety of vulnerabilities. For instance, analyze web security incidents in China, focusing on vulnerabilities disclosed on platforms, highlighting the prevalent issues that impact web security regionally [6].

*Corresponding author

Email addresses: awangga@ulbi.ac.id (Rolly Maulana Awangga), 1204045@std.ulbi.ac.id (Muhammad Rifqi Daffa Ulhaq)

2.2. *Input Validation Vulnerabilities*

Offer a systematic review of input validation vulnerabilities, classifying and analyzing current state-of-the-art security measures in web applications. Their work underscores the critical role of input validation in securing web applications against common attacks such as SQL injection and cross-site scripting (XSS) [7].

2.3. *Analysis of Security Parameters*

Compare various web application security parameters, discussing current trends and future directions. Their comparative analysis provides a detailed examination of security frameworks and their effectiveness against web application threats [8].

2.4. *Vulnerability Assessment of Web Applications*

Conduct a vulnerability assessment of websites and web applications using tools like Nmap and Nessus. Their study identifies common security flaws across different institutions, proposing solutions to enhance security protocols [9].

2.5. *Methodological Advances in Security Assessments*

Propose a new methodology for assessing the security of web applications, particularly focusing on the use of Python-based tools for vulnerability assessment. This approach not only identifies but also helps mitigate vulnerabilities in a timely manner [10].

3. Literature Review

The exploration of security challenges in Software-as-a-Service (SaaS) platforms reveals a complex landscape of vulnerabilities that enterprises face today. Provide a multivocal literature review that synthesizes diverse perspectives on SaaS security issues. Their study emphasizes the importance of adopting best practices such as robust data encryption, regular security audits, and the development of incident response strategies to mitigate potential breaches effectively [11].

Further investigation into the security of e-commerce platforms by Baako et al. (2019) highlights significant concerns regarding privacy and data security on electronic commerce sites in Ghana. Their survey study evaluates how these platforms manage user data, emphasizing the critical need for comprehensive privacy policies and advanced encryption techniques to protect sensitive customer information from unauthorized access and potential data breaches [12].

Focus on the advancements in penetration testing techniques for web applications. Their survey provides an overview of the current tools and methods used to identify and mitigate vulnerabilities. The study serves as a crucial resource for developers and security professionals seeking to understand and implement effective penetration testing practices to enhance the security posture of web applications[13].

Karoui (2016) introduces a novel risk assessment framework for evaluating security threats, specifically focusing on distributed denial-of-service (DDoS) attacks against e-commerce servers. This framework utilizes reversible metrics for likelihood and impact, which facilitates better risk management and comparison of different risk assessment methods [14].

De Cremer et al. (2020) explore the enforcement of secure coding guidelines within integrated development environments through their tool, Sensei. This approach directly integrates security into the software development lifecycle, significantly reducing common security vulnerabilities by guiding developers in real-time [15].

Nagpal et al. (2016) present SECSIX, a security engine designed to combat prevalent web application attacks such as cross-site request forgery (CSRF), SQL injection, and cross-site scripting (XSS). Their work emphasizes the need for robust security engines that can dynamically adapt to the evolving nature of web threats [16].

Rath (2017) discusses the dual focus on quality of service and security in cloud computing environments. Her research emphasizes the importance of resource provisioning along with enhanced security measures to ensure reliable service delivery and safeguard client-side applications [17].

Malviya et al. (2021) develop a web browser prototype that includes embedded classification capabilities specifically designed to mitigate XSS attacks. This innovative approach enhances browser security by preemptively identifying and blocking malicious scripts before they execute [18].

Mesquida and Mas (2015) address the integration of security best practices in software lifecycle processes, particularly through the lens of the ISO/IEC 15504 Security Extension. Their study provides insights into how structured frameworks can enhance security measures from software development to deployment, underscoring the necessity of incorporating security at every phase of the development process [19].

Cartwright et al. (2023) explore cybersecurity risk management in UK's micro and small businesses, highlighting the crucial role of IT companies in cascading best practice information. This research sheds light on the significant impact of effective communication and the implementation of best practices in enhancing the overall cybersecurity posture of small enterprises [20].

Johns (2014) delves into the application of script-templates for enhancing Content Security Policy (CSP), offering a methodological approach to mitigate cross-site scripting (XSS) and other similar attacks. This study is pivotal in illustrating how tailored security policies can significantly fortify web applications against prevalent web threats [21].

Gao et al. (2020) examine security mechanisms in website account bindings, identifying potential vulnerabilities in OAuth implementations and other account binding methods. Their work proposes solutions to strengthen security in these processes, ensuring safer user interactions across various platforms [22].

Niemimaa and Niemimaa (2017) critique the gap between security policy best practices and their actual implementation, urging a shift towards more practical and context-sensitive security practices in information systems. This study is crucial for understanding the discrepancies between theoretical security measures and their practical applications [23].

Vakeel et al. (2016) investigate differences in security and privacy policies between B2B and B2C e-commerce. Their comparative content analysis reveals that B2B vendors are generally more focused on security aspects, while B2C vendors prioritize privacy concerns related to consumer data and intimacy [24].

Joo and Hovav (2015) examine the influence of information security on the adoption of web-based integrated information systems in e-government contexts in Peru, providing insights into how security perceptions can impact governmental information system implementations [25].

Liu et al. (2015) discuss the implications of client-side security restrictions on cloud computing services competition, highlighting how security measures at the client level can affect overall cloud service usage and market dynamics [26].

Vaithyasubramanian et al. (2019) propose enhancements in web security against bots and spam using a linguistic CAPTCHA, showcasing how advanced CAPTCHA systems can strengthen website defense mechanisms against automated attacks [27].

Alhogail (2020) explores the enhancement of information security best practices within virtual knowledge communities. This study emphasizes the importance of community-driven knowledge sharing to improve security practices across distributed networks, highlighting how collective intelligence can bolster overall cybersecurity measures [28].

Raponi and Di Pietro (2020) provide a longitudinal analysis of password management practices on websites, identifying widespread vulnerabilities in how passwords are managed and suggesting robust security measures to mitigate these risks. Their research contributes significantly to understanding the persisting challenges and necessary innovations in password security [29].

Calzavara et al. (2020) discuss the application of machine learning techniques to detect vulnerabilities in web applications, specifically focusing on cross-site request forgery (CSRF). Their research illustrates the potential of machine learning to enhance the detection of security vulnerabilities through automated tools[30].

Calzavara, Rabitti, and Bugliesi (2018) analyze the deployment of Content Security Policy (CSP) to protect against common web attacks, providing insights into how semantic understanding of web applications can improve CSP implementation and effectiveness [31].

Ayeni, Sahalu, and Adeyanju (2018) address the detection of Cross-Site Scripting (XSS) attacks using a fuzzy inference system, offering a novel approach that enhances detection accuracy and reduces false positives. Their method demonstrates how advanced analytical techniques can significantly improve the security of web applications[32].

Mateo Tudela et al. (2020) combine various analytical methods to improve the detection of security vulnerabilities according to the OWASP Top Ten, focusing on a holistic approach that integrates static, dynamic, and interactive analysis to provide thorough security testing for web applications [33].

Zhang et al. (2023) discuss a state-sensitive approach to black-box scanning specifically designed for detecting cross-site scripting (XSS) vulnerabilities in web applications. This method enhances the accuracy of vulnerability detection by adapting to the state changes within the application during the scanning process[34].

Bucko et al. (2023) propose an enhancement to JWT authentication and authorization mechanisms in web applications by integrating user behavior history. This approach aims to increase the security and reliability of authentication processes by considering patterns such as IP consistency and frequency of access[35].

Abdulghaffar et al. (2023) explore the effectiveness of enhancing web application security through automated penetration testing. Their study leverages multiple vulnerability scanners to provide a comprehensive assessment and mitigation of security risks[36].

Alsaffar et al. (2022) present a method for detecting web-based XSS attacks, focusing on improving the security measures against one of the most common and impactful security threats to modern web applications[37].

Nurul Atiqah Abu Talib and Kyung-Goo Doh (2021) assess the effectiveness of dynamic open-source filters designed to mitigate XSS attacks in web applications, offering a solution that adapts to evolving attack methods while maintaining user interaction integrity [38].

4. Question Formulations

The formulation of research questions is a critical step in conducting a systematic literature review [7]. The questions guide the review process, ensuring that it is focused and systematic. For this SLR, the following research questions have been formulated:

1. **RQ1: What are the prevalent security vulnerabilities in web applications?**
   - This question aims to identify and categorize the most common security vulnerabilities found in web applications based on existing literature.

2. **RQ2: What methods and tools are currently used to detect and mitigate these vulnerabilities?**
   - This question investigates the various methods and tools that are used for detecting and mitigating security vulnerabilities in web applications. It will include both commercial and open-source solutions.

3. **RQ3: How effective are the existing methods and tools in addressing web application security vulnerabilities?**
   - This question evaluates the effectiveness of the current methods and tools in mitigating web application security vulnerabilities. It will look

into their strengths and weaknesses, as well as their practical applicability.

4. **RQ4: What are the limitations and challenges associated with these methods and tools?**

   - This question aims to identify the limitations and challenges faced by the current methods and tools in addressing web application security vulnerabilities. It will provide insights into areas that require further research and improvement.

5. **RQ5: What are the best practices and recommendations for improving web application security?**

   - This question seeks to compile best practices and recommendations from the literature to enhance web application security. It will include guidelines and strategies for developers and security professionals.

5. Source Selection

The source selection process for this systematic literature review involved several stages of searching, screening, and including relevant articles. The process is visualized in the ROSES, SPAR-4-SLR, and PRISMA flow diagrams.

5.1. Keyword Identification

The identification of relevant keywords is essential for ensuring comprehensive search results. Table 1 lists the keywords used in the search process, along with the number of articles identified for each keyword.

Table 1
Keywords Used for Paper Search

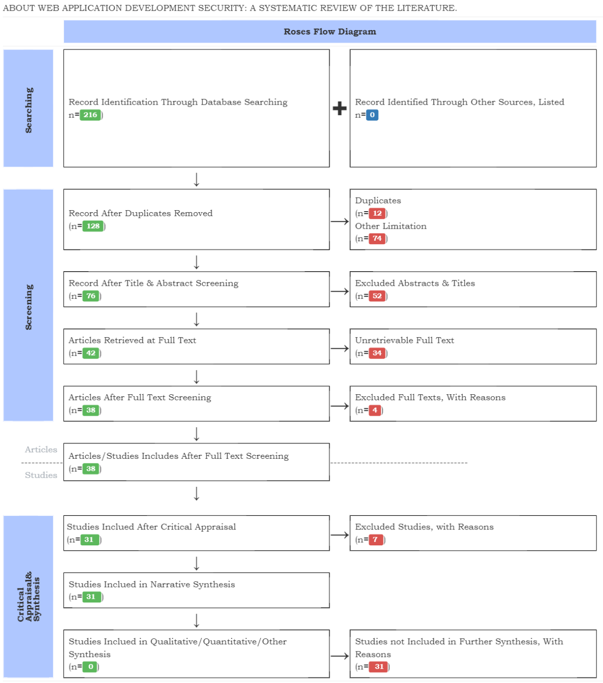
Keyword	**Raw**
Cross-Site Scripting, web, xss	19
Security Best Practices	38
Content Security Policy	14
Web application, Penetration Testing	9
Client-Side Security	8
Authentication, Authorization, web	3
Web application security, study	7
Web application security, tools	6
Web security study	22
CSRF	15
Website security	43
Login security	16
Input validation, web application	8
Vulnerability Assessment, web application	8

This table highlights the diversity of keywords and their effectiveness in identifying relevant articles for this systematic literature review.

5.2. ROSES Flow Diagram

The ROSES flow diagram illustrates the detailed process of source selection:

- **Record Identification:** A total of 216 records were identified through database searching. No records were identified through other sources.
- **Screening:** After removing duplicates (12) and other limitations (74), 128 records were screened based on title and abstract.
- **Eligibility:** Out of 76 records screened, 34 full-text articles were not retrievable. After assessing the remaining 42 full-text articles, 4 articles were excluded for various reasons.
- **Inclusion:** A total of 38 articles were included after full-text screening. After critical appraisal, 31 studies were included in the narrative synthesis.


ABOUT WEB APPLICATION DEVELOPMENT SECURITY: A SYSTEMATIC REVIEW OF THE LITERATURE.
Roses Flow Diagram
Searching
Record Identification Through Database Searching
n=216)
Record Identified Through Other Sources, Listed
n=0
Screening
Record After Duplicates Removed
(n=128)
Duplicates
(n=12)
Other Limitation
(n=74)
Record After Title & Abstract Screening
(n=76)
Excluded Abstracts & Titles
(n=52)
Articles Retrieved at Full Text
(n=42)
Unretrievable Full Text
(n=34)
Articles After Full Text Screening
(n=38)
Excluded Full Texts, With Reasons
(n=4)
Articles
Studies
Articles/Studies Includes After Full Text Screening
(n=38)
Critical Appraisal& Synthesis
Studies Inclued After Critical Appraisal
(n=31)
Excluded Studies, with Reasons
(n=7)
Studies Inclued in Narrative Synthesis
(n=31)
Studies Inclued in Qualitative/Quantitative/Other Synthesis
(n=0)
Studies not Included in Further Synthesis, With Reasons
(n=31)

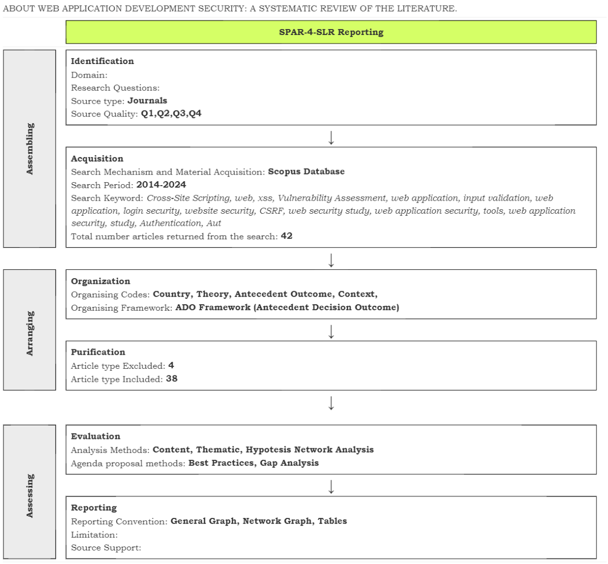
Generate From Watase Uake Tools, based on Roses Reporting

Figure 1. ROSES Flow Diagram for Source Selection

5.3. SPAR-4-SLR Reporting

The SPAR-4-SLR reporting provides a structured approach for source selection:

- **Identification:** 42 articles were identified based on the search mechanism from the Scopus database for the period 2014-2024 using relevant keywords.
- **Organization:** Articles were organized using the ADO framework and categorized by country, theory, antecedent outcome, and context.
- **Purification:** 4 articles were excluded, leaving 38 articles included for analysis.
- **Evaluation:** Content, thematic, and hypothesis network analysis were used for evaluating the included studies.


ABOUT WEB APPLICATION DEVELOPMENT SECURITY: A SYSTEMATIC REVIEW OF THE LITERATURE.
SPAR-4-SLR Reporting
Assembling
Arranging
Assessing

Identification
Domain:
Research Questions:
Source type: **Journals**
Source Quality: **Q1,Q2,Q3,Q4**

Acquisition
Search Mechanism and Material Acquisition: **Scopus Database**
Search Period: **2014-2024**
Search Keyword: *Cross-Site Scripting, web, xss, Vulnerability Assessment, web application, input validation, web application, login security, website security, CSRF, web security study, web application security, tools, web application security, study, Authentication, Aut*
Total number articles returned from the search: **42**

Organization
Organising Codes: **Country, Theory, Antecedent Outcome, Context,**
Organising Framework: **ADO Framework (Antecedent Decision Outcome)**

Purification
Article type Excluded: **4**
Article type Included: **38**

Evaluation
Analysis Methods: **Content, Thematic, Hypotesis Network Analysis**
Agenda proposal methods: **Best Practices, Gap Analysis**

Reporting
Reporting Convention: **General Graph, Network Graph, Tables**
Limitation:
Source Support:

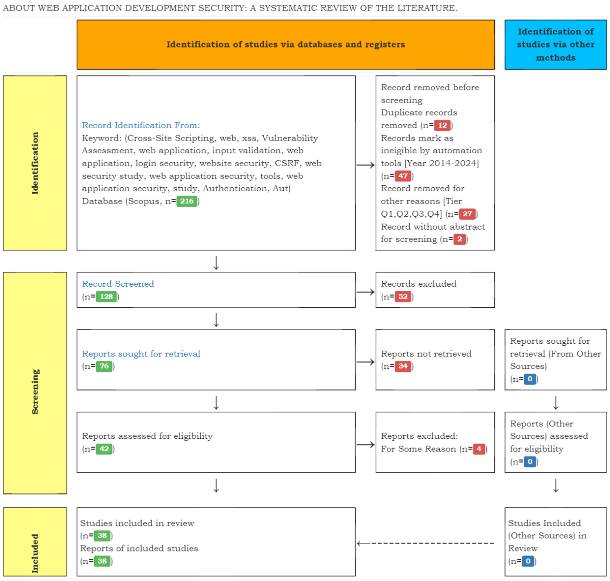
Generate From Watase Uake Tools, based on SPAR-4-SRL 2021 Reporting

Figure 2. SPAR-4-SLR Reporting for Source Selection

5.4. *PRISMA Flow Diagram*

The PRISMA flow diagram summarizes the source selection process:

- **Identification:** 216 records were identified through database searching.
- **Screening:** After removing duplicates and other limitations, 128 records were screened.
- **Eligibility:** 76 reports were sought for retrieval; 34 were not retrieved. Out of the remaining 42, 4 were excluded.
- **Inclusion:** 38 studies were included in the review.


ABOUT WEB APPLICATION DEVELOPMENT SECURITY: A SYSTEMATIC REVIEW OF THE LITERATURE.
Identification of studies via databases and registers
Identification of studies via other methods
Identification
Screening
Included

Record Identification From:
Keyword: (Cross-Site Scripting, web, xss, Vulnerability Assessment, web application, input validation, web application, login security, website security, CSRF, web security study, web application security, tools, web application security, study, Authentication, Aut)
Database (Scopus, n=216)

Record removed before screening
Duplicate records removed (n=12)
Records mark as ineigible by automation tools [Year 2014-2024] (n=47)
Record removed for other reasons [Tier Q1,Q2,Q3,Q4] (n=27)
Record without abstract for screening (n=2)

Record Screened
(n=128)

Records excluded
(n=52)

Reports sought for retrieval
(n=76)

Reports not retrieved
(n=34)

Reports sought for retrieval (From Other Sources)
(n=0)

Reports assessed for eligibility
(n=42)

Reports excluded:
For Some Reason (n=4)

Reports (Other Sources) assessed for eligibility
(n=0)

Studies included in review
(n=38)
Reports of included studies
(n=38)

Studies Included (Other Sources) in Review
(n=0)

Generate From Watase Uake Tools, based on Prisma 2020 Reporting

Figure 3. PRISMA Flow Diagram for Source Selection

6. Results

This section presents the results of the systematic literature review on web application development security. The data obtained from various studies have been processed and categorized based on the research methods used. The following subsections provide a detailed overview of the findings.

6.1. *Distribution of Articles Over Time*

The distribution of articles over time provides insights into the trends and focus areas in web application development security research. Graph 4 shows the number of articles published in different years.

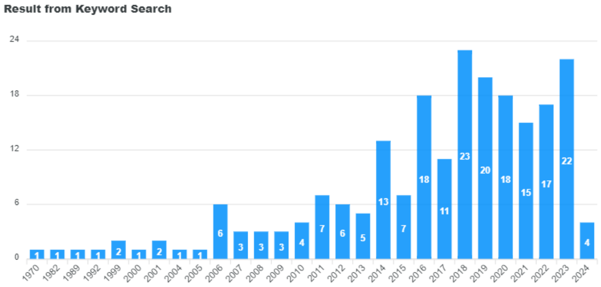


Figure 4. Distribution of Articles Over Time

6.2. *Research Methods Used*

The studies included in this review utilized a variety of research methods. Pie chart 5 summarizes the different methods and the number of studies employing each method.

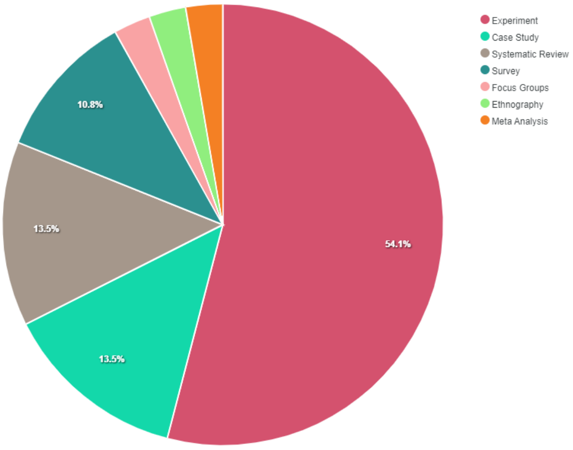


Figure 5. Research Methods Used in the Studies

6.3. *Vulnerabilities, Methods, and Solutions*

Table 2 provides an overview of the different vulnerabilities identified in the reviewed studies, the methods used to address these vulnerabilities, and the proposed solutions.

Table 2
Vulnerabilities, Methods, and Solutions

Vulnerability	Research Method	Solution	Reference
Cross-Site Scripting (XSS)	Experiment	Machine learning-based classification	Malviya et al. (2021) [18]
SQL Injection	Survey	Web application firewall	Altulaihan et al. (2023) [13]
CSRF	Case Study	SECSIX security engine	Nagpal et al. (2016) [16]
Input Validation	Systematic Review	Best practices for input validation	Fadlalla et al. (2023) [7]
DDoS Attacks	Case Study	Reversible metrics-based risk assessment framework	Karoui (2016) [14]
Secure Coding Guidelines	Experiment	IDE plugin for enforcing secure coding	De et al. (2020) [15]
Privacy and Security in E-Commerce	Survey	Security policy enhancements	Baako et al. (2019) [12]
SaaS Security Challenges	Systematic Review	Best practices and guidelines	Humayun et al. (2022) [11]
QoS in Cloud Computing	Experiment	Resource provision and QoS support system	Rath (2017) [17]
PreparedJS for XSS	Experiment	Safe script templating with script checksumming	Johns (2014) [21]

7. Discussion

The results of this systematic literature review provide a comprehensive overview of the current state of web application development security. The analysis of the reviewed articles reveals several key findings and trends that are critical for understanding the landscape of web application security vulnerabilities and the methods used to address them.

7.1. *Prevalent Vulnerabilities*

The most prevalent vulnerabilities identified in the reviewed studies include Cross-Site Scripting (XSS), SQL Injection, Cross-Site Request Forgery (CSRF), and input validation issues. These vulnerabilities are consistently highlighted across various studies, indicating their significance and the persistent challenges they pose to web application security. The consistent identification of these vulnerabilities underscores the need for continued focus and improvement in these areas.

7.2. *Methods and Tools for Vulnerability Detection and Mitigation*

The studies reviewed utilized a range of methods and tools for detecting and mitigating web application vulnerabilities. Commonly used methods include experiments, case studies, surveys, and systematic reviews. Tools such as web application firewalls, machine learning-based classifiers, and security engines like SECSIX were frequently mentioned. The diversity of methods and tools reflects the multifaceted nature of web application security and the need for a comprehensive approach to address different types of vulnerabilities.

7.3. *Effectiveness of Existing Solutions*

The effectiveness of existing solutions varies widely depending on the specific vulnerability and the context in which the solution is applied. Machine learning-based approaches have shown promise in detecting complex patterns of malicious activity, while traditional methods like web application firewalls continue to play a crucial role in providing a baseline level of protection. However, several studies highlighted the limitations of current solutions, such as high false positive rates and limited applicability to newer web technologies.

7.4. *Challenges and Limitations*

Several challenges and limitations were identified in the reviewed studies. One major challenge is the rapid evolution of web technologies, which often outpaces the development of corresponding security measures. Additionally, many studies pointed out the lack of empirical validation for proposed solutions, making it difficult to assess their real-world effectiveness. The high variability in the quality and scope of different studies also poses a challenge for drawing generalizable conclusions.

7.5. *Best Practices and Recommendations*

Based on the findings of this review, several best practices and recommendations can be made to improve web application security. These include:

- Implementing robust input validation mechanisms to prevent common vulnerabilities like XSS and SQL Injection.

- Utilizing a combination of traditional security tools and advanced methods such as machine learning for

Table 4
Comparison and analysis of literature review

Researcher	**Research Method**	**Evaluated Vulnerabilities**	**Main Contributions**	**Limitation**
Humayun et al. (2022) [11]	Sytematic Review	SaaS Security Challenges	Comprehensive review of SaaS security challenges and best practices.	Lack of empirical validation of proposed best practices.
Baako et al. (2019) [12]	Survey	Privacy and Security in E-Commerce	Survey of privacy and security issues in Ghanaian e-commerce websites.	Limited to a specific geographical region, Ghana.
Altulaihan et al. (2023) [13]	Survey	Web Application Vulnerabilities	Comprehensive review and comparison of web penetration testing tools.	Focuses mainly on tool comparison, less on practical implementation.
Karoui (2016)[14]	Case Study	DDoS Attacks on E-commerce Web Servers	Development of a novel risk assessment framework based on reversible metrics.	Case study limited to DDoS attacks; broader applications not covered.
De et al. (2020) [15]	Experiment	Secure Coding Guidelines	Development of an IDE plugin to enforce secure coding guidelines.	Tool applicability limited to specific development environments.
Faisal and Elshoush (2023) [7]	Systematic Review	Input Validation Vulnerabilities	Systematic review, classification, and analysis of input validation vulnerabilities in web applications.	Limited comparison in real environment of proposed tools/solutions.
Johns (2014) [21]	Experiment	Cross-Site Scripting (XSS)	Proposes PreparedJS, an extension to CSP, enhancing security against XSS attacks by combining safe script templating with script checksumming.	Limited to JavaScript, not extending to HTML or CSS.
Malviya et al. (2021) [18]	Experiment	Cross-Site Scripting (XSS)	Developed a prototype web browser with embedded classification capability to mitigate XSS attacks using machine learning.	Limited to Web 2.0; further work needed for Web 3.0 and HTML5 integration.
Rath (2017) [17]	Experiment	QoS and Security in Cloud Computing	Proposes a web-based application for resource provision and QoS support with added security for client-side applications in cloud computing.	Focuses on simulation results, requiring real-world validation.
Nagpal et al. (2016) [16]	Case Study	CSRF, SQL Injection, XSS	Presents SECSIX, a security engine to counter CSRF, SQL injection, and XSS attacks.	Implementation limited to PHP applications; broader applicability needs exploration.

comprehensive vulnerability detection and mitigation.

- Regularly updating and patching web applications to address newly discovered vulnerabilities.
- Conducting continuous security assessments and penetration testing to identify and address security gaps.
- Providing ongoing training and education for developers on secure coding practices and the latest security threats.

These best practices, if implemented effectively, can significantly enhance the security posture of web applications and reduce the risk of exploitation.

7.6. Future Research Directions

The review also highlights several areas for future research. There is a need for more empirical studies to validate the effectiveness of proposed security solutions in real-world scenarios. Additionally, research should focus on developing security measures that can keep pace with the rapid evolution of web technologies. The integration of emerging technologies such as artificial intelligence and blockchain into web application security frameworks also presents a promising area for future exploration.

Overall, this systematic literature review provides valuable insights into the current challenges and solutions in web application development security. By synthesizing the findings from a wide range of studies, this review offers a foundation for future research and practical recommendations for improving web application security. The table 4 will show some reviews of relevant research.

8. Acknowledgments

The authors would like to express their sincere gratitude to the Department of Informatics Engineering at the International University of Logistics and Business in Bandung, Indonesia, for providing the necessary resources and support throughout this research.

Special thanks go to my supervisor, Rolly Maulana Awangga, S.T., MT, CAIP, SFPC, for his invaluable guidance, constructive feedback, and continuous encouragement, which greatly contributed to the successful completion of this systematic literature review.

This research would not have been possible without the contributions and collaboration of various individuals and organizations. We appreciate the assistance and insights provided by all those who participated in this study.

References

[1] B. M. Shuaibu, N. M. Norwawi, M. H. Selamat, A. Al-Alwani, Systematic review of web application security development model, Artificial Intelligence Review.

[2] S. Liu, X. Yan, Q. Wang, Q. Xi, A systematic study of content security policy in web applications, Security and Communication Networks.

[3] M. Dadkhah, G. Borchardt, M. Lagzian, Do you ignore information security in your journal website?, Science and Engineering Ethics.

[4] P. Nunes, I. Medeiros, J. Fonseca, N. Neves, M. Correia, M. Vieira, An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios, Computing.

[5] B. Csontos, I. Heckl, Accessibility, usability, and security evaluation of hungarian government websites, Universal Access in the Information Society.

[6] C. Huang, J. Liu, Y. Fang, Z. Zuo, A study on web security incidents in china by analyzing vulnerability disclosure platforms, Computers Security.

[7] F. F. Fadlalla, H. T. Elshoush, Input validation vulnerabilities in web applications: Systematic review, classification, and analysis of the current state-of-the-art, IEEE Access.

[8] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, N. Crespi, A comparative study of web application security parameters: Current trends and future directions, Applied Sciences.

[9] V. Appiah, M. Asante, I. K. Nti, O. Nyarko-Boateng, Survey of websites and web application security threats using vulnerability assessment, Journal of Computer Science.

[10] M. A. A. Hammoudeh, A. Alobaid, A. Alwabli, F. Alabdulmunim, The study on assessment of security web applications, International Journal of Interactive Mobile Technologies (iJIM).

[11] M. Humayun, M. Niazi, M. F. Almufareh, N. Z. Jhanjhi, S. Mahmood, M. Alshayeb, Software-as-a-service security challenges and best practices: A multivocal literature review, Applied Sciences.

[12] I. Baako, S. Umar, P. Gidisu, Privacy and security concerns in electronic commerce websites in ghana: A survey study, International Journal of Computer Network and Information Security.

[13] E. A. Altulaihan, A. Alismail, M. Frikha, A survey on web application penetration testing, Electronics.

[14] K. Karoui, Security novel risk assessment framework based on reversible metrics: a case study of ddos attacks on an e-commerce web server, International Journal of Network Management.

[15] P. D. Cremer, N. Desmet, M. Madou, B. D. Sutter, Sensei: Enforcing secure coding guidelines in the integrated development environment, Software: Practice and Experience.

[16] B. Nagpal, N. Chauhan, N. Singh, Secsix: security engine for csrf, sql injection and xss attacks, International Journal of System Assurance Engineering and Management.

[17] M. Rath, Resource provision and qos support with added security for client side applications in cloud computing, International Journal of Information Technology.

[18] V. K. Malviya, S. Rai, A. Gupta, Development of web browser prototype with embedded classification capability for mitigating cross-site scripting attacks, Applied Soft Computing.

[19] A. L. Mesquida, A. Mas, Implementing information security best practices on software lifecycle processes: The iso iec 15504 security extension, Computers Security.

[20] A. Cartwright, E. Cartwright, E. S. Edun, Cascading information on best practice cyber security risk management in uk micro and small businesses and the role of it companies, Computers Security.

[21] M. Johns, Script-templates for the content security policy, Journal of Information Security and Applications.

[22] H. H. X. W. Gao Xi, Yu Lei, W. Yiwen, A research of security in website account binding, Journal of Information Security and Applications.

[23] N. Elina, N. Marko, Information systems security policy implementation in practice from best practices to situated practices, European Journal of Information Systems.

[24] K. A. Vakeel, S. Das, G. J. Udo, K. Bagchi, Do security and privacy policies in b2b and b2c e-commerce differ? a comparative study using content analysis, Behaviour Information Technologydoi:10.1080/0144929X.2016.1236837.

[25] J. Joo, A. Hovav, The influence of information security on the adoption of web-based integrated information systems: an e-government study in peru, Information Technology for Developmentdoi:10.1080/02681102.2014.899961.

[26] Y. Liu, X. Sheng, S. R. Marston, The impact of client-side security restrictions on the competition of cloud computing services, International Journal of Electronic Commercedoi:10.1080/10864415.2016.1057084.

[27] S. Vaithyasubramanian, D. Lalitha, C. K. Kirubhashankar, Enhancing website security against bots, spam and web attacks using lcaptcha, International Journal of Computers and Applicationsdoi:10.1080/1206212X.2019.1702285.

[28] A. Alhogail, Enhancing information security best practices sharing in virtual knowledge communities, VINE Journal of Information and Knowledge Management Systems.

[29] S. Raponi, R. D. Pietro, A longitudinal study on web-sites password management (in)security: Evidence and remedies, IEEE Access.

[30] S. Calzavara, M. Conti, R. Focardi, A. Rabitti, G. Tolomei, Machine learning for web vulnerability detection: The case of cross-site request forgery, IEEE Security Privacy.

[31] S. Calzavara, A. Rabitti, M. Bugliesi, Semantics-based analysis of content security policy deployment, ACM Transactions on the Web.

[32] B. K. Ayeni, J. B. Sahalu, K. R. Adeyanju, Detecting cross-site scripting in web applications using fuzzy inference system, Journal of Computer Networks and Communications.

[33] F. M. Tudela, J.-R. B. Higuera, J. B. Higuera, J.-A. S. Montalvo, M. I. Argyros, On combining static, dynamic and interactive analysis security testing tools to improve owasp top ten security vulnerability detection in web applications, Applied Sciences.

[34] T. Zhang, H. Huang, Y. Lu, K. Zhu, J. Zhao, State-sensitive black-box web application scanning for cross-site scripting vulnerability detection, Applied Sciences.

[35] A. Bucko, K. Vishi, B. Krasniqi, B. Rexha, Enhancing jwt authentication and authorization in web applications based on user behavior history, Computers.

[36] K. Abdulghaffar, N. Elmrabit, M. Yousefi, Enhancing web application security through automated penetration testing with multiple vulnerability scanners, Computers.

[37] M. Alsaffar, S. Aljaloud, B. A. Mohammed, Z. G. Al-Mekhlafi, T. S. Almurayziq, G. Alshammari, A. Alshammari, Detection of web cross-site scripting (xss) attacks, Electronics.

[38] N. A. A. Talib, K.-G. Doh, Assessment of dynamic open-source cross-site scripting filters for web application, KSII Transactions on Internet and Information Systems.