Review on The Most Common Web Applications Vulnerabilities

Tahani Alshammari[1†] and Saloua Hendaoui [2††]

Department of computer Science, College of Computer and Information Sciences, Jouf University,
Jouf, Skaka, Saudi Arabia

Summary
In recent years, technical development evolves exponentially to meet human needs and to facilitate our lives. In addition, in current circumstances, due to the Covid 19 pandemic, the information technology community is trying to obtain secure access to most government and private services. The activities of individuals have become dependent on the availability of information and services over the net, through web applications, such as applications for government services, e-learning, e-commerce, electronic banking services, and others. This increasing demand for web applications and the great dependence on them in our daily activities, and due to the privilege of these applications being easily accessible from anywhere, makes them vulnerable to the exploitation of attackers.This paper systematically reviewed various common vulnerabilities in web applications. The systematic literature review (SLR) results show that SQL Injection and Cross-Site Scripting (XSS), then Cross-Site Request Forgery (CSRF) are that tops the most common vulnerabilities.

Keywords:
WebApplications; Vulnerabilities; SQL Injection; XSS; CSRF

1. Introduction

Nowadays, web applications are the backbone of the net due to their role in our lives. Today's education, health, commerce, and banking services rely heavily on web applications. Due to the increasing use of web applications and the volume of information available on the Internet through these applications, individuals' information and accounts have become vulnerable to hacking and fraud.
It is estimated that 70% of web applications are vulnerable to penetration as these applications can be accessed from anywhere [10]. This percentage is increasing with the technical development and the widespread of web applications. Hence, we must recognize the security vulnerabilities in web applications to reveal their sources and plan for solutions to prevent our systems and limit the risks.
The remaining of this paper is organized as follows: Section 2 presents a background and overview of web applications, and vulnerabilities according to 15 posted papers, Section 3 discusses the methodology followed in this paper and describes the application of systematic literature review protocol, and section 4 describes data extraction and section 5 presents the finding of SLR and answer a research question and finally, section 6 concludes this review.

Manuscript received September 5, 2025
Manuscript revised September 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.9.3

2. RELATED WORK

Many studies and previous systematic literature reviews have been conducted on web application vulnerabilities, including:

Sandip Sarkar [1] described different types of Web application vulnerabilities like
(SQL Injection, Cross-Site Scripting, Cross-Site Request Forgery, Broken Authentication and Session Management, Security Misconfiguration).

Aliero et al. [2] determined SQL Injection Attack as the most severe attack that can be used against web database-driven applications by proposing automatic black-box testing for SQL Injection Vulnerability (SQLIV) to automate an SQLIV assessment in SQLIA. To validate the accuracy of their work, they developed three vulnerable web applications.

Moniruzzaman et al. [3] evaluated the vulnerabilities of selected web- sites of Bangladesh against a set of most common and prevalent attack vectors namely (SQL Injection, Cross-Site Scripting (XSS), Broken Authentication and Session (BAS), Cross-Site Request Forgery (CSRF), Port Scan Attack, Transport Layer Security (TLS)) Then, they presented the vulnerability status of the websites in different graphical formats. The result showed that 64% of the selected web applications in Bangladesh are running with vulnerabilities and specifically, government websites are in a critical state.

Su et al. [4] consider SQL Injection the most common type of web vulnerability, they studied the law of SQL injection attacks according to the different outflow channels, then they proposed a model of SQL injection to guide the generation of the use cases used in the penetration testing.
Mitropoulos et al. [5] claim that Cross-Site Scripting and SQL injection are the most dangerous web attacks. They provided an analysis of various defense mechanisms against web code injection attacks. Then they proposed a model that highlights the key weaknesses enabling these attacks, and that provided a common perspective for studying the available defenses.
Touseef et al. [6] analyzed an extensive literature review on web application vulnerabilities security testing. The results reveal that SQL injection followed by XSS and Sensitive data exposure is the most recurring risk of web applications.
Nirmal et al. [7] determined the most common attack as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).

Aljamea et al. [8] explained the most common security issues of web applications and how those can be dealt with through minimal efforts in a structured manner. They determined the three most serious vulnerabilities (SQL injection, Cross-site Scripting (XSS), Cross-Site Request Forgery (CSRF)).

Román et al. [9] proposed a method to map web vulnerability classification and selected the Vulnerabilities to test the algorithm implementation as SQL injection, Session fixation, Cross-Site Scripting (XSS), path traversal, and Cross-Site Request Forgery.

Vijayalakshmi et al. [10] defined the most as SQL injection and Cross-site Scripting (XSS), they proposed a defending mechanism based on Extenuating Web Vulnerability with a detection and protection mechanism for secure web access.

Hernández-Saucedo et al. [11] presented a hybrid process that enables organizations to detect and manage vulnerabilities in their web applications, they considered the most vulnerabilities that are found in web applications as SQL Injection, Cross-site Scripting (XSS), Cross-Site Request Forgery CSRF, and Insecure Configuration Management

Gupta et al. [12] reported Cross-site Scripting (XSS) as the most serious vulnerability in web applications, they proposed a context-sensitive approach based on static taint analysis and pattern matching techniques to detect and mitigate the XSS vulnerabilities in the source code of web applications.

Saleh et al. [13] considered SQL Injection, Buffer Overflow, Cross-Site Scripting, and Cross-Site Request Forgery the most web application vulnerabilities. They proposed a technique that aims to solve these issues by developing a detection method for detecting the web application vulnerabilities by using Boyer-Moore String Matching Algorithm.

Rafique et al. [14] performed a Systematic Review of Web Application Security Vulnerabilities Detection Methods. They consider that the top 10 web application vulnerabilities are: code injection, Cross-site scripting (XSS), Broken Authentication and Session Management, Insecure direct object references, Cross-Site Request Forgery (CSRF), Security misconfiguration, Failure to restrict URL access, Invalidated redirects, and forwards, insecure cryptographic storage, Insufficient transport layer protection.

Antunes et al. [15] considered The two most common risks in the Web as SQL injection, and cross-site scripting (XSS). They explained Defending mechanism against web application Vulnerabilities. All 15 papers studied talked about a limited number of web application vulnerabilities. No paper has examined all common web application vulnerabilities. Therefore, in this recent paper, we will present a comprehensive study.

3. Methodology

In this research paper, a systematic literature review methodology has been followed. A systematic literature review (SLR) is a systematic, explicit and comprehensive process to identify, evaluate and synthesize from results of works produced by researchers, scholars, and practitioners [16]. It identifies, selects, and critically evaluates research to answer a formulated question [17]. Firstly, SLR was used in healthcare researches. Currently, it is used by several other fields of researches such as computer science and information technology [18]. According to [17] SLR methodology reduces research bias and [16] argued that results obtained from SLR research have a greater level of validity.

A Systematic Literature Review (SLR) protocol is a multitude of tasks that must be performed on a group of researchers to answer a specific research question by gathering and summarizing all empirical evidence from research that is consistent with the topic.

3.1. Research question

This research paper focuses on understanding and identify web applications vulnerabilities. In order to identify to what extent has been studied.

This paper investigates to answer the following research question: What are the main vulnerabilities in web applications?

3.2. Search strategy and term

The following search strategy is used for the construction of search terms. Derivation of major terms from research questions. alternative synonyms for these major terms. Search in databases and Use of Boolean Operators for conjunction if the database allows. ‘OR’ operator for the concatenation of alternative spellings and synonyms whereas ‘AND’ for the concatenation of major terms.

(intitle: “Vulnerabilities” OR) AND (“Web Application”) And (“Most” OR “common”).

4. Data Extraction

4.1. Research sources

The search selection is limited to the following database sources:

- IEEE Explorer.
- ACM Digital Library.
- Springer Link.
- Science Direct

Steps taken for data extraction :

a. The first step taken was to eliminate papers that didn’t include web applications and vulnerabilities as a keyword.

b. The second step was to eliminate papers that didn't research web application vulnerabilities in abstract then full text.
c. The last step was a manual procedure that carefully selected papers according to publication qualities.
d.

4.2. Selected Studies

The studies selected are 15, present in table 1.

Study ID	Title
1	Detecting Vulnerabilities of Web Application Using Penetration Testing and Prevent Using Threat Modeling
2	An algorithm for detecting SQL injection vulnerability using black-box testing
3	Measuring Vulnerabilities of Bangladeshi Websites
4	Research on SQL Injection Vulnerability Attack model
5	Defending Against Web Application Attacks: Approaches, Challenges and Implications
6	Analysis of Automated Web Application Security Vulnerabilities Testing
7	Web Application Vulnerabilities - The Hacker's Treasure
8	Effective Solutions for Most Common Vulnerabilities in Web Applications
9	An algorithm to find relationships between web vulnerabilities
10	Extenuating web vulnerability with a detection and protection mechanism for a secure web access
11	Proposal of a Hybrid Process to Manage Vulnerabilities in Web Applications
12	XSSDM: Towards detection and mitigation of cross-site scripting vulnerabilities in web applications
13	A Method for Web Application Vulnerabilities Detection by Using Boyer-Moore String Matching Algorithm
14	Systematic Review of Web Application Security Vulnerabilities Detection Methods
15	Defending against Web Application Vulnerabilities

4.3. Publication Years

Selected papers have been restricted from the search time frame between 2012 and 2021 (table 2). 15 studies have been selected to identify vulnerabilities in web applications. The list is not exhaustive, but it represents high-quality publications that of web applications vulnerabilities research. Below is a breakdown of studies published between 2012 and January 2021.

TABLE.2: PUBLICATION YEAR OF PAPERS SELECTED

Year	Study reference number	Total
2012	15	1
2013	0	0
2014	0	0
2015	12,13,14	3
2016	11	1
2017	10	1
2018	7,8,9	3
2019	3,4,5,6	4
2020	2	1
Jan-2021	1	1
		15

We didn't find publication in 2013 and 2014 that meet selection criteria.

5. Results

We performed a systematic literature review of the selected 15 papers to provide answers to the research question: **What are the main vulnerabilities in web applications?**
The results are summarized in Table.3:

TABLE.3: THE COMMON VULNERABILITIES IN WEB APPLICATIONS

Year	Study ID	Most common Web Application Vulnerabilities
2021 (january)	1	• SQL Injection • Cross-Site Scripting (XSS) • Cross-Site Request Forgery (CSRF) • Broken Authentication and Session (BAS) • Security Misconfiguration
2020	2	• SQL Injection
2019	3	• SQL Injection • Cross Site Scripting (XSS) • Broken Authentication and Session (BAS) • Cross-Site Request Forgery (CSRF) • Port Scan Attack, Transport Layer Security (TLS)
2019	4	• SQL Injection
2019	5	• SQL Injection • Cross-Site Scripting
2019	6	• SQL Injection • Cross Site Scripting (XSS) • Sensitive data exposure
2018	7	• SQL Injection • Cross Site Scripting (XSS) • Cross-Site Request Forgery (CSRF)
2018	8	• SQL Injection • Cross Site Scripting (XSS) • Cross-Site Request Forgery (CSRF)
2018	9	• SQL injection • Session fixation • Cross Site Scripting (XSS) • Path traversal • Cross-Site Request Forgery (CSRF)
2017	10	• SQL Injection • Cross Site Scripting (XSS)
2016	11	• SQL Injection • Cross-site Scripting (XSS)

		• Cross Site Request Forgery (CSRF) • Insecure Configuration Management
2015	12	• Cross-site Scripting (XSS)
2015	13	• SQL Injection • Buffer Overflow • Cross Site Scripting (XSS) • Cross Site request Forgery (CSRF)
2015	14	• Injection vulnerability • Cross site scripting (XSS) • Broken Authentication and Session Management • Insecure direct object references • Cross-Site Request Forgery (CSRF) • Security misconfiguration • Failure to restrict URL access • Invalidated redirects and forwards • insecure cryptographic storage • Insufficient transport layer protection
2012	15	• SQL Injection • Cross Site Scripting (XSS)

The most common vulnerabilities in seemed web applications according to iteration times are described in Table 4.

TABLE.4: THE MOST COMMON VULNERABILITIES IN WEB APPLICATIONS

Number	Vulnerabilities	Study ID	Iteration times
1	SQL Injection	1,2,3,4,5,6,7,8,9,10,11,13,15	13
2	Cross-Site Scripting (XSS)	1,3,5,6,7,8,9,10,11,12,13,14,15	13
3	Cross-Site Request Forgery (CSRF)	1,3,7,8,9,11,13,14	8
4	Broken Authentication and Session (BAS)	1,3,14	3
5	Security Misconfiguration	1,14	2
6	Port Scan Attack, Transport Layer Security	3	1
7	Sensitive data exposure	6	1
8	Session fixation	9	1
9	Path traversal	9	1
10	Insecure Configuration Management	11	1
11	Buffer Overflow	13	1
12	Injection vulnerability	14	1
13	Insecure direct object references	14	1
14	Failure to restrict URL access	14	1
15	Invalidated redirects and forwards	14	1
16	insecure cryptographic storage	14	1
17	Insufficient transport layer protection	14	1

The findings show that SQL Injection and Cross-Site Scripting (XSS) have been mentioned and detailed in 13 out of 15 studies. Cross-Site Request Forgery (CSRF) has been mentioned and discussed in 8 out of 15 studies. Broken Authentication and Session (BAS) vulnerability has been mentioned and discussed in 3 out of 15 studies. Security Misconfiguration has been mentioned in 2 out of 15 studies. These are the most common vulnerabilities, depending on their frequencies, as shown in fig..

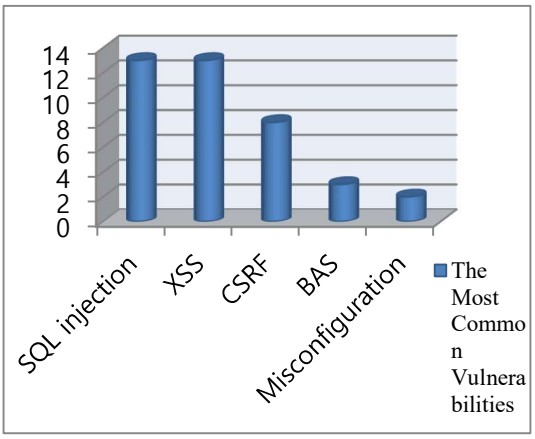

Figure.1 Most common vulnerabilities in web applications

Based on previous findings, for the 15 papers reviewed. No research paper has mentioned all common vulnerabilities in web applications. So, in this paper, we have provided a comprehensive review that pointed out all the most common vulnerabilities in web applications:

1. SQL Injection
2. Cross-Site Scripting (XSS)
3. Cross-Site Request Forgery (CSRF)
4. Broken Authentication and Session (BAS)
5. Security Misconfiguration
6. Port Scan Attack, Transport Layer Security
7. Sensitive data exposure
8. Session fixation
9. Path traversal
10. Insecure Configuration Management
11. Buffer Overflow
12. Injection vulnerability

13. Insecure direct object references
14. Failure to restrict URL access
15. Invalidated redirects and forwards
16. insecure cryptographic storage
17. Insufficient transport layer protection

6. Conclusion

In this paper, a systematic review of the most common vulnerabilities in web applications is conducted. It turns out the most common vulnerabilities are SQL Injection and Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), then respectively was Broken Authentication and Session (BAS), and Security Misconfiguration.

Given the continuing proliferation of vulnerabilities in web applications and according to the findings reached through this paper, our future goals and aspirations are: Increase the number of studies and collect new research on vulnerabilities in web applications. · Finding the best protection tools, methods, and methodologies for web applications.

Finally, regardless of the limitations of this paper, I hope this systematic review will help readers understand the current vulnerabilities in web applications and pave the way for a systematic review on a larger scale.

Acknowledgments

The authors would like to thank the Deanship of Graduate Studies at Jouf University for funding and supporting this research through the initiative of DGS, Graduate Students Research Support (GSR) at Jouf University, Saudi Arabia.

References

[1] S. Sarkar, "Detecting Vulnerabilities of Web Application Using Penetration Testing and Prevent Using Threat Modeling," 2021, pp. 21–32.

[2] M. S. Aliero, I. Ghani, K. N. Qureshi, and M. F. Rohani, "An algorithm for detecting SQL injection vulnerability using black-box testing," J. Ambient Intell. Humaniz. Comput., vol. 11, no. 1, pp. 249–266, 2020.

[3] M. Moniruzzaman, F. Chowdhury, and M. S. Ferdous, "Measuring Vulnerabilities of Bangladeshi Websites," in 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Feb. 2019, pp. 1–7, doi: 10.1109/ECACE.2019.8679426.

[4] G. Su, F. Wang, and Q. Li, "Research on SQL Injection Vulnerability Attack model," Proceedings of 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2018. pp. 217–221, 2019, doi: 10.1109/CCIS.2018.8691148.

[5] D. Mitropoulos, P. Louridas, M. Polychronakis, and A. D. Keromytis, "Defending Against Web Application Attacks: Approaches, Challenges and Implications," IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 2. pp. 188–203, 2019, doi: 10.1109/TDSC.2017.2665620.

[6] P. Touseef et al., "Analysis of Automated Web Application Security Vulnerabilities Testing," in Proceedings of the 3rd International Conference on Future Networks and Distributed Systems, Jul. 2019, pp. 1–8, doi: 10.1145/3341325.3342032.

[7] K. Nirmal, B. Janet, and R. Kumar, "Web Application Vulnerabilities - The Hacker's Treasure," Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018. pp. 58–62, 2018, doi: 10.1109/ICIRCA.2018.8597221.

[8] M. Aljamea, C. S. Iliopoulos, and M. Samiruzzaman, "Effective Solutions for Most Common Vulnerabilities in Web Applications," Lecture Notes in Networks and Systems, vol. 16. pp. 738–754, 2018, doi: 10.1007/978-3-319-56991-8_53.

[9] F. Román Muñoz and L. J. García Villalba, "An algorithm to find relationships between web vulnerabilities," Journal of Supercomputing, vol. 74, no. 3. pp. 1061–1089, 2018, doi: 10.1007/s11227-016-1770-3.

[10] K. Vijayalakshmi and A. A. Leema, "Extenuating web vulnerability with a detection and protection mechanism for a secure web access," 2017 4th International Conference on Signal Processing, Communication and Networking, ICSCN 2017. 2017, doi: 10.1109/ICSCN.2017.8085652.

[11] A. L. Hernández-Saucedo and J. Mejía, "Proposal of a Hybrid Process to Manage Vulnerabilities in Web Applications," 2016, pp. 59–69.

[12] M. K. Gupta, M. C. Govil, G. Singh, and P. Sharma, "XSSDM: Towards detection and mitigation of cross-site scripting vulnerabilities in web applications," 2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015. pp. 2010–2015, 2015, doi: 10.1109/ICACCI.2015.7275912.

[13] A. Z. M. Saleh, N. A. Rozali, A. G. Buja, K. A. Jalil, F. H. M. Ali, and T. F. A. Rahman, "A Method for Web Application Vulnerabilities Detection by Using Boyer-Moore String Matching Algorithm," Procedia Comput. Sci., vol. 72, pp. 112–121, 2015, doi: 10.1016/j.procs.2015.12.111.

[14] S. Rafique, M. Humayun, Z. Gul, A. Abbas, and H. Javed, "Systematic Review of Web Application Security Vulnerabilities Detection Methods," J. Comput. Commun., vol. 03, no. 09, pp. 28–40, Feb. 2015, doi: 10.4236/jcc.2015.39004.

[15] N. Antunes and M. Vieira, "Defending against Web Application Vulnerabilities," Computer (Long. Beach. Calif)., vol. 45, no. 2, pp. 66–72, Feb. 2012, doi: 10.1109/MC.2011.259.

[16] Okoli, Chitu and Schabram, Kira, A Guide to Conducting a Systematic Literature Review of Information Systems Research (May 5, 2010). Available at SSRN: https://ssrn.com/abstract=1954824 or http://dx.doi.org/10.2139/ssrn.1954824

[17] Dewey, A. & Drahota, A. (2016) Introduction to systematic reviews: online learning module *Cochrane Training* https://training.cochrane.org/interactivelearning/module-1-introduction-conducting-systematic-reviews

[18] V. M. Silva, J. I. M. S. Junior, D. N. Prata, and P. Letouze, ―The importance of systematic review as a scientific research method for computer science: a quantitative systematic review,‖ International Proceedings of Economic Development and Research, vol.78, p.44, 2014.

Tahani Alshammari: Master student in jouf university received the B.E.. degrees, from jouf Univ

Saloua Hendaoui received the B.E. and M.E. degrees, from tunis Univ. in 2011 and 2009, respectively. She received the Dr.. degree from Cartage Univ. in 2017. Working as a assistant professor (from 2018) in the Dept. of computer Science Jouf University