Received: 29 Aug 2023 | Revised: 16 Apr 2024 | Accepted: 16 Jun 2024 | Published: 30 Jun 2024
Volume 2(1) Pages 72-86 DOI: https://doi.org/ 10.56279/jicts.v2i1.51

JICTS
Journal of ICT Systems

Quantifying Vulnerabilities: A Systematic Review of the State-of-the-Art Web-Based Systems

Wilbard G. Masue[1], Daniel Ngondya, Tabu S. Kondo

Department of Computer Science and Engineering, The University of Dodoma

[1]Corresponding author
Email: wilbard.masue@udom.ac.tz

Funding information

This study was supported by the University of Dodoma

Keywords

Common web vulnerabilities
Cyber security
Systematic literature review
Web vulnerability ranking

Abstract

Web-based Systems Vulnerabilities (WSVs) have been existing over a long time in all Open System Interconnection (OSI) layers. WSV tends to affect online business operations by letting attackers to gain unauthorized access. Different researchers have been publishing common WSVs regularly. From the published vulnerabilities, it can be noted that the ranking of vulnerabilities is not static. Prevalence of common vulnerabilities tends to vary with time. Moreover, ranking of vulnerabilities from various practitioners, such as OWASP and CWE, at a particular point in time tends to be different because of different approaches and sources. This work sought to come up with an objective way of establishing the latest ranking of common WSV by conducting a Systematic Literature Review from scholarly sources. This study extracted 127 publications from Scholarly Databases: Association of Computing Machineries, ScienceDirect, Springer, IEEE, and Google scholar. After the review, only 62 articles were considered based on five inclusion and exclusion criteria. The review reveals that cross site script, structured query language injection, broken authentication and session management, operating system command injection and file inclusion are the most common WSV.

1. Introduction

Most businesses across the world are moving to online platforms to simplify their operations [1]. Online operations on web-based systems store critical business assets. With advancement of information and communication technologies, the number of hackers has been increasing. The kind of attacks performed by hackers includes stealing data, performing unauthorized deletion or modification of data [2-6].

Web-based system developers have been knowingly or unknowingly undermining security concerns during development of web-based systems from planning to testing and deployment phases [7-8]. As a result, attackers have continued to exploit vulnerabilities in web-based systems and compromise them while disrupting business operations.

Different researchers and practitioners have been publishing common Web-based Systems Vulnerabilities (WSVs) regularly [9-12]. From the published vulnerabilities, it can clearly be noted that the ranking of vulnerabilities is not static. There are over 30 different kinds of web vulnerabilities. The prevalence of the vulnerabilities varies. Prevalence of vulnerabilities tends to vary with time depending on various factors. Therefore, it becomes difficult for web developers to keep up and prioritize patching up of systems. Moreover, ranking of vulnerabilities from various practitioners at a particular point in time tends to be different because of different approaches and sources.

Similar systematic reviews have considered a few type of vulnerability, such as input validation, and have indicated prevalence change over time [13]. Previous works also lack discussion of related mitigation strategies of the vulnerabilities. This review sought to establish the state-of-art web vulnerabilities, taking into account different kinds of vulnerabilities and prevalence. This work sought to come up with an objective way of establishing the latest ranking of common WSV by conducting a Systematic literature review in Scholarly Databases (SDs) from 2012 to 2021.

The contributions of this work are as follows: firstly, it presents the quantitative systematic review of the state-of-the-art WSV from 2012 to 2021; secondly, it explains the vulnerability ranking similarities and differences between Open Web Application Security Project (OWASP), Common Weaknesses, and Enumeration (CWE) and common WSV obtained from SLR under this study.

The rest of this paper is organized as follows: it contains a methodology and Systematic Review Process Section, which explains procedures and approaches adopted to undertake this study; then, the Results and Discussion Section which presents results and their discussions obtained from the state-of-the-art WSV. Finally, the Conclusion and Recommendations Section.

2. Methodology and Systematic Review Process

The writing of this systematic literature review was based on the Kitchenham guiding principles, PRISMA (Preferred Reporting Items for Systematics Meta Analysis) and PICO (Population, Interventions, Context and Outcome) keyword search paradigm, as used in the works of Rafique [13] and Zarour [15]. Academic papers were searched from different publication sources, including IEEE, ACM, ScienceDirect, Springer, and Google scholar. A total of 127 published papers in the domain were extracted, and after a careful synthesis based on inclusion and exclusion criteria, 62 papers primarily related to WSV were considered for the study. Therefore, the paper presented a quantitative systematics literature review of the state-of-the-art WSV. The systematics literature review process based on the selected methodology was conducted under the following steps in a structured order, namely question formulation, source selection, study selection, selection execution, and information extraction.

2.1 Question Formulation

Based on the Kitchenham guidelines for SLR, research questions are one of the most crucial aspects of the review. They guide the process by ensuring that primary study selection and aggregation relate directly to the questions. This review has drawn up a research question "What are the common state-of-the-art WSVs". This research question has been subdivided into three research questions as follows:

I. Which specific types of web system vulnerabilities are most frequently encountered in the current state-of-the-art research across diverse web platforms?
II. How do the prevalence and distribution of common web system vulnerabilities vary across different categories of web applications?
III. What are the emerging trends and shifts in the landscape of web system vulnerabilities?

And, how have they evolved over time in response to advancements in web technologies and security practices?

2.2 Source Selection

To identify primary studies related to the mentioned research questions, the study carried a pilot search on some trustworthy publication sources. A pilot search on these sources has discovered that some similar publications are indexed in more than one source, and therefore the search selection is limited to IEEE, ACM, Springer, ScienceDirect and Google scholar.

At first, the search keywords were formed and extracted from the PICO paradigm. After the crucial assessment, the recommended keywords from the sources were combined with the list of keywords during the live search and the PICO mapping for keywords and keyword synonyms as were presented in Table 1 and Table 2, respectively.

The keywords were concatenated using Boolean "AND" and "OR" to bring the whole query strings as shown in Table 3. And this study has considered 1 page with 25 papers for IEEE, 1 page with 15 papers for ACM, 1 page with 20 papers for Springer, 1 page with 24 papers for ScienceDirect and lastly 5 pages with 46 papers for Google scholar. The reason for taking five pages in Google scholar is because it is not actually an SD but just an academic scholarly search engine that retrieves papers from SD, including, but not limited to, IEEE, Springer, ScienceDirect and ACM.

2.3 Study Selection

After searching papers from the publication sources, the extraction of paper that was primarily considered and used for this study was done based on inclusion and exclusion criteria.

The criteria that were considered to include or exclude a research article in this systematic review paper was: the initial stage excluded all articles that have not focused on web security vulnerabilities. The second stage included all papers that have been published from 2012 to 2022. The aim is to identify current common web vulnerabilities around the globe for the past decade. The third selection for inclusion was based on whether the paper's title, abstract, and introduction have a clear connection with the study research questions.

The fourth selection criteria excluded articles published as journal proof, symposium or workshop. The fifth and last criteria removed the duplicate articles that have appeared in more than one source.

Table 1. PICO paradigm of the study.

S/N	PICO Paradigm	Keyword
1	Population	Web-based systems
2	Intervention	Security vulnerabilities
3	Context	Domain of web-based systems
4	Outcome	Quantity and type of web-based system vulnerabilities

Table 2. Keywords synonyms.

S/N	Keywords	Keywords Synonyms
1	Web	Web, Internet, online, website
2	Web-based system	Web-based system, web application, web service, internet application, web-based application, web software, web system
3	Security	Security, secure, insecurity
4	vulnerabilities	Vulnerability

Table 3. Keywords combinations used for search in each source.

S/N	Keywords combination using "AND" and "OR"	Sources	Page(s)	Papers
1	((Web OR online OR internet OR website OR web system OR web service OR internet application) AND (security OR secure OR insecurity) AND (vulnerabilities OR web vulnerabilities OR web-based vulnerabilities OR web application vulnerabilities))	IEEE	1	25
2.	((Web OR online OR internet OR website OR web system OR web service OR internet application) AND (security OR secure OR insecurity) AND (vulnerabilities OR web vulnerabilities OR web-based vulnerabilities OR web application vulnerabilities))	ACM	1	15
3.	((Web) AND (security) AND (web vulnerabilities OR web-based vulnerabilities OR web application vulnerabilities))	Springer	1	20
4.	((Web) AND (security) AND (web vulnerabilities OR web-based vulnerabilities OR web application vulnerabilities))	ScienceDirect	1	24
5.	((Web OR online OR internet OR website OR web system OR web service OR internet application) AND (security OR secure OR insecurity) AND (vulnerabilities OR web vulnerabilities OR web-based vulnerabilities OR web application vulnerabilities))	Google Scholar	1-5	46
	Total Number of Extracted Papers			**127**

2.4 Selection Execution

From Figure 1, The first criteria excluded 15 articles that were not related to WSV. The second criteria excluded 26 articles that either were published before 2012 or those not having information of year of publication. The third criteria excluded 15 articles based on relevance of articles in title, abstract and introduction in relation with the research questions. The fourth criteria excluded 3 articles that were either published as workshop, symposium or journal proof. And lastly, the fifth criteria excluded 6 research articles that appeared duplicate in more than one source.

Figure 2 presents the percentages of papers that were selected from the scholarly sources where the most relevant publications were selected from ScienceDirect (24.2%), Google Scholar (24.2%) and IEEE (24.2%). Other publications were selected from ACM (8.1%) and Springer (19.3%).

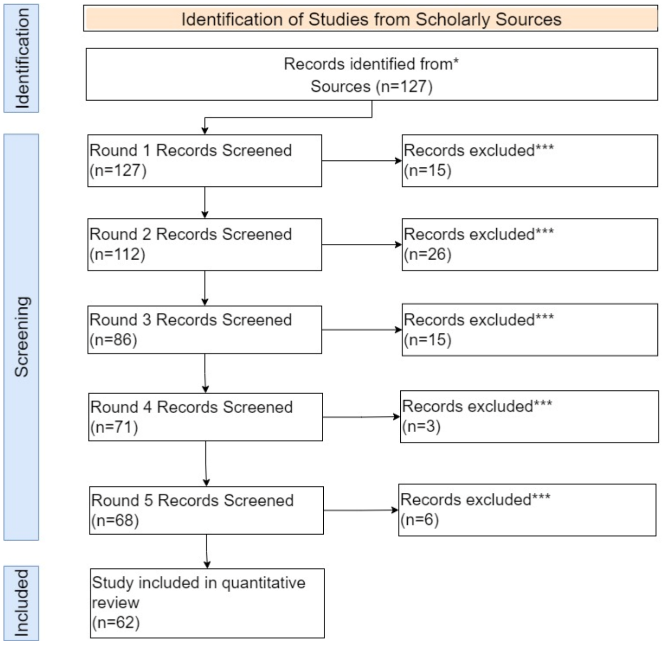
Identification
Screening
Included
Identification of Studies from Scholarly Sources
Records identified from* Sources (n=127)
Round 1 Records Screened (n=127)
Records excluded*** (n=15)
Round 2 Records Screened (n=112)
Records excluded*** (n=26)
Round 3 Records Screened (n=86)
Records excluded*** (n=15)
Round 4 Records Screened (n=71)
Records excluded*** (n=3)
Round 5 Records Screened (n=68)
Records excluded*** (n=6)
Study included in quantitative review (n=62)

Figure 1. Screening for the papers reviewed extended from [15].

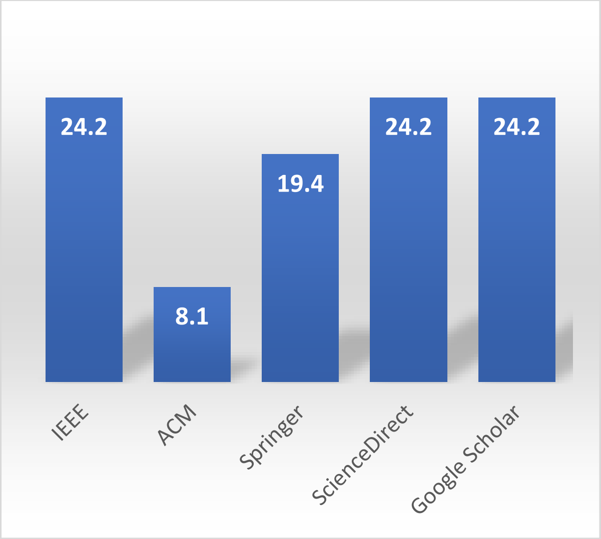

Figure 2. Percentages of articles selected from each source.

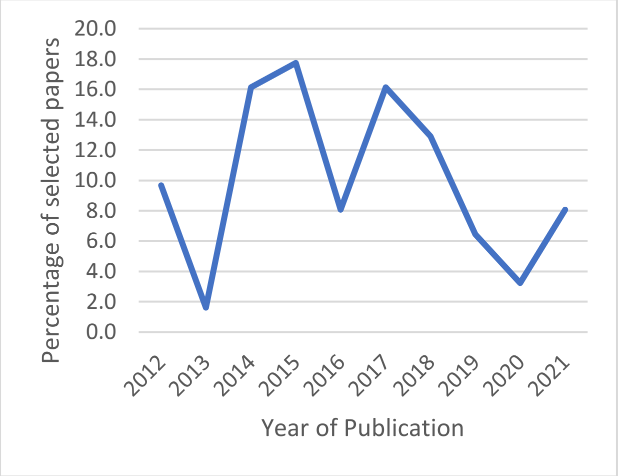

Figure3. Percentage of selected articles in each year.

Figure 3 presents the line graph with percentage of selected publications from each year in which the percentage of selected papers from 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020 and 2021 were 9.7%, 1.6%, 16.1%, 17.7%, 8.1%, 16.1%, 12.9%, 6.5%, 3.2%, and 8.1%, respectively.

Figure 4 presents a pie chart showing classification of selected papers based on location where the research was conducted. The study found that 4% of papers published in the African continent were selected. Australia did not contribute any paper for the review. Percentage of selected papers from Asia and Europe were 57% and 25%, respectively. The American continent had 14% of papers selected for the review.

Figure 5 is the histogram which presents the classification of selected papers based on methodology used, in which percentages of papers selected under survey, experimental, comprehensive literature review, systematic literature mapping and systematic literature review methodologies were 8.1%, 69.4%, 4.8%, 4.8% and 12.9%, respectively.

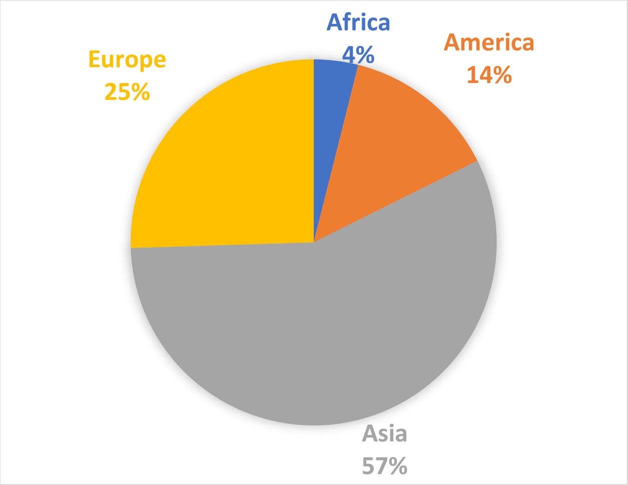

Figure 4. Percentage of selected articles based on locations.

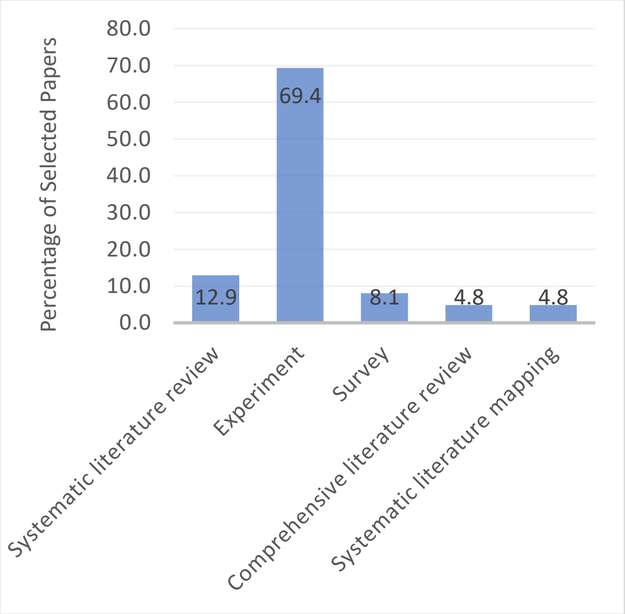

Figure 5. Classification of selected articles based on methodolog

3. Result

Table 4 shows the ranked WSV based on the 62 selected articles which present the quantitative state-of-the-art of the common WSV from 2012 to 2021. The total of 41 vulnerabilities have been obtained in the reviewed articles, in which XSS, SQLi, Broken Authentication and Session Management, CSRF, OS Command injection and File inclusion have appeared the most. It can be seen that out of 62 research articles selected for this study, 43 articles have been researching on XSS and make it mostly common with 17.8 %, SQLi has appeared in 40 articles with 16.5% of mostly common WSV. Other vulnerabilities that have appeared common are, respectively, Broken Authentication and Session Management with 25 appearance and 10.4%, CSRF with 14 appearance and 5.8%, OS Command injection and File inclusion both with 13 appearance and 5.3%, directory traversal and path traversal both with 10 appearances and 4.1%. The rest of vulnerabilities have less than 10 appearances and thus are less common compared with others that have more than 10 appearances.

Figure 6 shows a sunburst chart which presents the common WSV from the state-of-the-art of the reviewed articles by years from 2012 to 2021 where by 2012 there were two common vulnerabilities, namely XSS and CSRF, which appeared in 5 articles each. In 2013, there were no common WSV extracted. In 2014 and 2015, XSS and SQLi were the common vulnerabilities which appeared in 8 articles each. In the year 2016, SQLi was the common vulnerability with 4 appearances in reviewed articles. In 2017, XSS appeared as the common vulnerability in 8 reviewed articles. In 2018 and 2019, both XSS and SQLi were the common vulnerabilities in which they both appeared in 4 reviewed articles each. Not only that, but also in 2020, XSS, SQLi, and CSRF were the common vulnerabilities in which they each appeared in 2 of the reviewed articles. Finally, XSS in 2021 was the common vulnerability that appeared in 3 of the reviewed articles.

Table 4. Ranked common WSV from the selected articles.

S/N	Vulnerabilities Name	Frequency	Percentage	Rank	References
1	XSS	43	17.8	1	[12-13], [16–37]
2	SQL Injection	40	16.5	2	[3], [12-13], , [31–46], [16 – 28]
3	Broken Authentication and Session Management	25	10.4	3	[12-13] , [31–37], [16-28]
4	CSRF	14	5.8	4	[12-13], [36-37], [42], [47], [16–19], [21], [28], [31], [33]
5	OS Command injection	13	5.3	5	[13], [16], [19], [30-31], [38], [48–52]
6	File inclusion	13	5.3	5	[12], [16], [35], [38], [50-51], [53– 55]
7	Directory traversal and path traversal	10	4.1	7	[12], [16-17], [27], [30], [48-49], [52-53]
8	Xpath injection	8	3.3	9	[28], [40-41], [47], [51-52], [56-57]
9	XML injection	6	2.4	10	[31-33], [47], [56-57]
10	Security Misconfiguration	6	2.4	10	[12-13], [33-34], [37], [58]
11	Remote code execution	6	2.4	10	[17], [41], [49-50], [53], [59]
12	LDAP injection	4	1.6	13	[31], [36], [41], [52]
13	Insecure Direct Object References	4	1.6	13	[13], [33], [36-37]

S/N	Vulnerabilities Name	Frequency	Percentage	Rank	References
14	Sensitive Data Exposure	4	1.6	13	[12], [33-34], [60]
15	Failure to Restrict URL Access	4	1.6	13	[13], [26], [36-37]
16	Insecure Cryptographic Storage	4	1.6	13	[13], [26], [36-37]
17	Parameter Tampering	3	1.2	18	[31-32], [48]
18	Source code disclosure	3	1.2	18	[49],[50], [53]
19	Logic errors	3	1.2	18	[16], [32], [47]
20	Unvalidated redirects	3	1.2	18	[12], [33],[37]
21	Shellshock	3	1.2	18	[16], [61-62]
22	Buffer overflow	3	1.2	18	[24], [28], [31]
23	XQuery injection	2	0.8	24	[47], [57]
24	PHP code injection	2	0.8	24	[49], [53]
25	Unvalidated Redirects and Forwards	2	0.8	24	[12], [33]
26	Heartbleed	2	0.8	24	[16], [62]
27	XXE	2	0.8	24	[36], [42]
28	Insufficient Transport Layer Protection	2	0.8	24	[13], [37]
29	SSL flaw	2	0.8	24	[62-63]
30	HTTP Response Splitting	1	0.4	31	[17]
31	Missing Function Level Access Control	1	0.4	31	[12]
32	Using Components with Known Vulnerabilities	1	0.4	31	[12]
33	Weak passwords	1	0.4	31	[16]
34	Taint-style vulnerability	1	0.4	31	[48]
35	Workflow bypass	1	0.4	31	[47]
36	HTTP Protocol Violation	1	0.4	31	[38]
37	Second Order Denial-of-Service	1	0.4	31	[64]
38	Trust Boundary Violation	1	0.4	31	[52]
39	Weak Encryption Algorithm	1	0.4	31	[52]
40	Weak Hash Algorithm	1	0.4	31	[52]

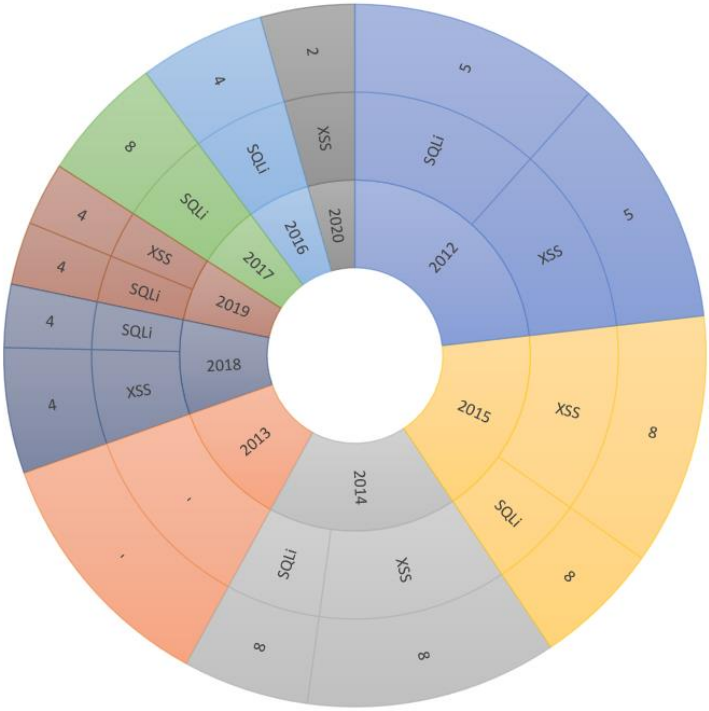

Figure 6. Common WSV from the state of the art of the reviewed articles by year.

Table 5. OWASP top 10 vulnerabilities in 2013, 2017 and 2021.

OWASP Top 10 - 2013	OWASP Top 10 - 2017	OWASP Top 10 - 2021
Injection	Injection	Broken access control
Broken Authentication and Session Management	Broken Authentication	Cryptographic Failures
Cross-Site Scripting (XSS)	Sensitive Data Exposure	Injection
Insecure Direct Object References	XML External Entities (XXE)	Insecure Design
Security Misconfigurations	Broken Access Control	Security Misconfiguration
Sensitive Data Exposure	Security Misconfigurations	Vulnerable and Outdated Components
Missing Function Level Access Control	Cross-Site-Scripting (XSS)	Identification and Authentication Failures
Cross-Site Request Forgery	Insecure Deserialization	Software and Data Integrity Failures
Using Components with Known Vulnerabilities	Using Components with Known Vulnerabilities	Security logging and Monitoring Failures
Unvalidated Redirects and Forwards	Insufficient Logging and Monitoring	Server-Side Request Forgery (CSRF)

OWASP publishes the list of top ten common vulnerabilities after every 4 years [10, 65]. It publishes from penetration testing companies' data around the world in which OWASP aggregates the data and then rank the vulnerabilities to produce the most critical top ten. This is done to help web developers to understand and consider them from their design phase because they are the most common and thus most attackers exploit them. Table 5 shows the OWASP top 10 vulnerabilities for 2013, 2017 and 2021, respectively.

4. Discussion

From Table 4 and Table 5, it can be shown that, while in the year 2013 and 2017 OWASP common vulnerability was injection vulnerability, in scholarly works the common WSV in the year 2013 and 2017 was SQLi. This implies the similarities between the common WSV from the scholarly works and those from penetration testing organizations. In the year 2021, OWASP common vulnerability was Broken access control. XSS was not even in the top 10 list, while from the scholarly works the XSS was the leading common vulnerability. This implies dissimilarity between the common vulnerability from the penetration testing industries and those from the scholarly works due to different approaches and sources used in ranking.

Not only that but also MITRE (Massachusetts Institute of Technology Research and Engineering) CWE (Common Weakness and Enumeration) publishes a list of 25 software weaknesses [66]. The list of software weaknesses in the year 2021 and 2022 are shown in Table 6 [67-68]. CWE generates the list by analyzing public vulnerability data from NVD (National Vulnerability Database), CVSS (Common Vulnerability Scoring System), and CVE records from Cybersecurity and Infrastructure Agency (CISA). It can be seen that the data and records are vulnerability data obtained from penetration tests performed in Known Exploited Vulnerability (KVE) Catalog. Comparing CWE list and that of common vulnerability from scholarly work, it can be seen that XSS and SQLi are still the top common software weaknesses by CWE.

Despite the existence of other ranked groups and organizations, the review of common vulnerability from scholarly work from 2012 to 2021 is an objective method of ranking and validating the common vulnerabilities from ranking organizations, such as OWASP and CWE, that use different records and databases to rank the vulnerabilities.

5. Conclusion

Web-based systems are an interesting field with a long history of research. The field is faced by challenges of attacks caused by the presence of unending known and unknown vulnerabilities that are announced every day. From the published vulnerabilities, it can clearly be noted that the ranking of vulnerabilities is not static. Prevalence of common vulnerabilities tends to vary with time. Moreover, ranking of vulnerabilities from various practitioners, such as OWASP and CWE, at a particular point in time tends to be different because of different approaches and sources. This work presents an objective way of establishing the latest ranking of common WSV by conducting a SLR from SD. This paper has bridged the gap by presenting quantitative systematic literature review to present the current state-of-the-art WSV. It also presents similarities and differences between the current state-of-the-art vulnerabilities from scholarly works and those from different ranking organizations. It will help practitioners and researchers to be informed with common WSV in the world while comprehending the security problems that require future research investigations.

Other Researchers may review the state-of-the-art WSV using scholarly work in specific web technologies, either based on the programming language or programming framework.

Table 6. CWE 25 software weakness for the year 2021 and 2022.

S/N	2021	2022
1	Out-of-bounds Write	Out-of-bounds Write
2	Cross-site Scripting	Cross-site Scripting
3	Out-of-bounds Read	SQL Injection
4	Improper Input Validation	Improper Input Validation
5	OS Command Injection	Out-of-bounds Read
6	SQL Injection	'OS Command Injection’
7	Use After Free	Use After Free
8	'Path Traversal'	'Path Traversal'
9	CSRF	CSRF
10	Unrestricted Upload of File with Dangerous Type	Unrestricted Upload of File with Dangerous Type
11	Missing Authentication for Critical Function	NULL Pointer Dereference
12	Integer Overflow or Wraparound	Deserialization of Untrusted Data
13	Deserialization of Untrusted Data	Integer Overflow or Wraparound
14	Improper Authentication	Improper Authentication
15	NULL Pointer Dereference	Use of Hard-coded Credentials
16	Use of Hard-coded Credentials	Missing Authorization
17	Improper Restriction of Operations within the Bounds of a Memory Buffer	Command Injection
18	Missing Authorization	Missing Authentication for Critical Function
19	Incorrect Default Permissions	Improper Restriction of Operations within the Bounds of a Memory Buffer
20	Exposure of Sensitive Information to an Unauthorized Actor	Incorrect Default Permissions
21	Insufficiently Protected Credentials	Server-Side Request Forgery (SSRF)
22	Incorrect Permission Assignment for Critical Resource	Race Condition
23	Improper Restriction of XML External Entity Reference	Uncontrolled Resource Consumption
24	SSRF	Improper Restriction of XML External Entity Reference
25	Command Injection	Code Injection

ACKNOWLEDGEMENT

Authors would like to thank the University of Dodoma for supporting this research work.

CONTRIBUTIONS OF CO-AUTHORS

Wilbard G. Masue	[ORCID: 0000-0001-6333-9456]	Conceived the idea and wrote the paper
Daniel Ngondya	[ORCID: 0000-0003-4267-6351]	Conducted review and language editing
Tabu S. Kondo	[ORCID: 0000-0002-0222-4951]	Conducted review and language editing

REFERENCES

[1] Farah, T., Shojol, M., Hassan, M., Alam, D., *Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF*, In 2016 6th International Conference on Digital Information and Communication Technology and Its Applications, (DICTAP), p. 74–78, 2016.

[2] Jang, Y, S., Choi, J. Y., *Detecting SQL injection attacks using query result size*, Computers & Security, **44**, p. 1–15, 2014.

[3] Mantra, I. G. N., Hartawan, M. S., Saragih, H., Abd Rahman, A., *Web vulnerability assessment and maturity model analysis on Indonesia higher education*, Procedia Computer Science,**161**, p. 1165–1172, 2019.

[4] Xu, Z., Hu, Q., Zhang, C., *Why computer talents become computer hackers: Start with talent and skills driven by curiosity and hormones, constrained only by moral values and judgment*, Communications of the ACM, **56**(4):p. 64–74, 2013.

[5] Chen, B., Zavarsky, P., Ruhl, R., Lindskog, D., *A study of the effectiveness of CSRF guard*, In IEEE international conference on privacy, security, risk and trust and IEEE international conference on social computing, PASSAT, p. 1269–1272, 2011.

[6] Bhatia, M., Maitra, J. K., *E-learning platforms security issues and vulnerability analysis*, In international conference on computational and characterization techniques in engineering & sciences, p. 276–285, 2018.

[7] Adebiaye, R., *Mitigating vulnerability risks in cybersecurity using predictive measures*, International Journal of Advanced Scientific Research & Development, **4**(10): p. 12, 2017.

[8] Rexha, B., Halili, A., Rrmoku, K., Imeraj, D., *Impact of secure programming on web application vulnerabilities*, In 2015 IEEE International Conference on Computer Graphics, Vision and Information Security, p. 61–66, 2015.

[9] Touseef, P., Alam, K. A., Jamil, A., Tauseef, H., Ajmal, S., Asif, R., Mustafa, S., *Analysis of automated web application security vulnerabilities testing*, In the 3rd international conference on future networks and distributed systems, p. 1-8, 2019.

[10] Pacey, PD., Purnell, JH., OWASP_Top_10 vulnerabilities, 2017.

[11] Singh, U. K., Joshi, C., *Quantifying security risk by critical network vulnerabilities assessment*, International journal of computer applications, **156**(13): p. 26–33, 2016.

[12] Idrissi, S. E., Berbiche, N., Guerouate, F., Shibi, M., *Performance evaluation of web application security scanners for prevention and protection against vulnerabilities*, International journal of applied engineering research, **12**(21): p. 11068–11076, 2017.

[13] Rafique, S., Humayun, M., Hamid, B., Abbas, A., Akhtar, M., Iqbal, K., *Web application security vulnerabilities detection approaches: A systematic mapping study*, In IEEE/ACIS 16th international conference on software engineering, artificial intelligence, networking and parallel/distributed computing, p. 1–6, 2015.

[14] Musa Shuaibu, B., Md Norwawi, N., Selamat, M. H., Al-Alwani, A., *Systematic review of web application security development model,* Artificial intelligence review, **43**(2): p. 259–276, 2015.

[15] Zarour, M., Alenezi, M., Ansari, M., Pandey, A., Ahmad, M., Agrawal, A., Khan, R. A., E*nsuring data integrity of healthcare information in the era of digital health*, Healthcare technology letters, **8**(3): p. 66–77, 2021.

[16] Huang, C., Liu, J., Fang, Y., Zuo, Z., *A study on web security incidents in China by analyzing vulnerability disclosure platforms*, Computers, and security, **58**, p. 1–30, 2016.

[17] Sultana, K. Z., Williams, B. J., Bhowmik, T., *A study examining relationships between micro patterns and security vulnerabilities*, Software quality journal **27**(1): p. 5–41, 2019.

[18] Amankwah, R., Chen, J., Kudjo, P. K., Agyemang, B. K., Amponsah, A. A., A*n automated framework for evaluating open-source web scanner vulnerability severity*, Service-oriented computing, and applications, **14**(4): p. 297–307, 2020.

[19] Priyanka, A. K., Smruthi, S. S., *Web ApplicationVulnerabilities : Exploitation and Prevention*, In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) IEEE, p. 729-734, 2020.

[20] Nunes, P., *An empirical study on combining diverse static analysis tools for web security vulnerabilities based on development scenarios*, Journal Computing, **101**, p. 161–185, 2018.

[21] Holík, F., Neradova, S., *Vulnerabilities of modern web applications*, In 40th international convention on information and communication technology, electronics and microelectronics, p. 1256–1261, 2017.

[22] Gupta, M. K., Govil, M. C., Singh, G., *Static analysis approaches to detect SQL injection and cross-site scripting vulnerabilities in web applications: A survey*, In international conference on recent advances and innovations in engineering, p. 9–13, 2014.

[23] Yadav, D., Gupta, D., Singh, D., Kumar, D., Sharma, U., *Vulnerabilities and security of web applications, In 2018 4th international conference on computing communication and automation*, p. 1–5, 2018.

[24] Saleh, A., Rozali, N., Buja, A., Jalil, K., Ali, F., Rahman, T., *A method for web application vulnerabilities detection by using boyer-moore string matching algorithm*, Procedia computer science, **72**, p. 112–121, 2015.

[25] Marashdih, A, Zaaba, Z, Suwais, K., Mohd, N., *Web application security: An investigation on static analysis with other algorithms to detect cross-site scripting*, Procedia computer science, **161**, p. 1173–1181, 2019.

[26] Zhang, K., *A machine learning based approach to identify SQL injection vulnerabilities*, In 34th IEEE/ACM international conference on automated software engineering, p. 1286–1288, 2019.

[27] Muñoz, F., Cortes, I., Villalba, L., *Enlargement of vulnerable web applications for testing*, The Journal of Supercomputing, **74**(12): p. 6598–6617, 2018.

[28] Muñoz, F., Villalba, L., *An algorithm to find relationships between web vulnerabilities*, The Journal of Supercomputing, **74**(3): p. 1061–1089, 2018.

[29] Leithner, M., Garn, B., Simos, D., Hydra: *Feedback-driven black-box exploitation of injection vulnerabilities.*, Information and software technology, **140**, p. 106703, 2021.

[30] Caseirito, J., Medeiros, I., *Finding web application vulnerabilities with an ensemble fuzzing*, In 51st Annual IEEE/IFIP international conference on dependable systems and networks-supplemental volume, p. 19–20, 2021.

[31] Deepa, G., Thilagam, P. S., *Securing web applications from injection and logic vulnerabilities: Approaches and challenges*, Information and software technology, **74**, p. 160–180, 2016.

[32] Deepa, G., Thilagam, P. S., Praseed, A., Pais, A. R., DetLogic: *A black-box approach for detecting logic vulnerabilities in web applications*, Journal of network and computer applications, **109**, p. 89–109, 2018.

[33] Priya, R. L., Lifna, C. S., Jagli, D., Joy, A., *Rational unified treatment for web application vulnerability assessment*, In International conference on circuits, systems, communication and information technology applications, p. 336–340, 2014.

[34] Helmiawan, M., Firmansyah, E., Fadil, I., Sofivan, Y., Mahardika, F., Guntara, A., *Analysis of web security using open web application security project 10*, In 8th International conference on cyber and IT service management, p. 1-5, 2020.

[35] Makino, Y., & Klyuev, V., *Evaluation of web vulnerability scanners*, In IEEE 8th international conference on intelligent data acquisition and advanced computing systems: technology and applications p. 399–40, 2015.

[36] Masood, A., *Cyber security for service-oriented architectures in a Web 2.0 world: An overview of SOA vulnerabilities in financial services*, In IEEE International Conference on technologies for homeland security, p. 1–6, 2013.

[37] Thankachan, A., Ramakrishnan, R., Kalaiarasi, M., *A survey and vital analysis of various state of the art solutions for web application security*, In International conference on information communication and embedded systems, p. 978, 2014.

[38] Rahman, M., Amjad, M., Ahmed, B., Siddik, M., *Analyzing web application vulnerabilities: An empirical study on e-commerce sector in Bangladesh*, In international conference on computing advancements, p. 5–10, 2020.

[39] Yu, F., Tung, Y. Y., Patcher: *An online service for detecting, viewing and patching web application vulnerabilities*, In 47th Hawaii international conference on system sciences, p. 4878–4886 2014.

[40] Antunes, N., Vieira., *Designing vulnerability testing tools for web services: approach, components, and tools*, International Journal of Information Security, **16**(4): p. 435–457, 2017.

[41] Stasinopoulos, A., Ntantogian, C., Xenakis, C., *Commix automating evaluation and exploitation of command injection vulnerabilities in Web applications*, International Journal of Information Security, **18**, p. 49–72, 2018.

[42] Nirmal, K., Janet, B., Kumar, R., *It's more than stealing cookies - exploitability of XSS*, In 2nd International conference on intelligent computing and control systems, p. 490–493, 2018.

[43] Scholte, T., Balzarotti, D., Kirda, E., *Quo vadis? A study of the evolution of input validation vulnerabilities in web applications*, In International Conference on Financial Cryptography and Data Security, p. 284–298, 2011.

[44] Fonseca, J., Vieira, M. Madeira, H., *Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection*, IEEE Transactions on Dependable and Secure Computing, **11**(5): p. 440-453, 2013.

[45] Fonseca, J., Seixas, N., Vieira, M., *Madeira, H., Analysis of field data on web security vulnerabilities*, IEEE Transactions on Dependable and secure computing, **11**(2): p. 89-100, 2014.

[46] Shar, L. K., Tan, H. B. K., *Predicting common web application vulnerabilities from input validation and sanitization code patterns*, In Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, p. 310–313, 2012.

[47] Deepa, G., Thilagam, P. S., Khan, F. A., *Praseed, Pais, A. R. Palsetia, N., Black-box detection of XQuery injection and parameter tampering vulnerabilities in web applications*, International Journal of Information Security, **17**(1): p. 105–120, 2018.

[48] Sahu, D. R., Tomar, D. S., *Analysis of Web Application Code Vulnerabilities using Secure Coding Standards*, Arabian Journal for Science and Engineering, **42**(2): p. 885–895, 2017.

[49] Medeiros, I., Neves, N. F., Correia, M., *Automatic detection and correction of Web application vulnerabilities using data mining to predict false positives*, In Proceedings of the 23rd International Conference on World Wide Web, p. 63–74, 2014.

[50] Medeiros, I., Neves, N., & Correia, M., DEKANT: *A static analysis tool that learns to detect web application vulnerabilities*, In Proceedings of the 25th International Symposium on software testing and analysis, p. 1–11, 2016.

[51] Zhao, J., & Gong, R., *A New Framework of Security Vulnerabilities Detection in PHP Web Application*, In 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, p. 271–276, 2015.

[52] Kritikos, K., Magoutis, K., Papoutsakis, M., Ioannidis, S., *A survey on vulnerability assessment tools and databases for cloud-based web applications*, Array, **3**(4): p. 1–60, 2019.

[53] Medeiros, I., Neves, N., Correia, *Detecting and Removing Web Application Vulnerabilities with Static Analysis and Data Mining*, IEEE Transactions on Reliability, **65**(1): p. 54–69, 2016.

[54] Amankwah, R., Kudjo, P. K., Antwi, S. Y., *Evaluation of Software Vulnerability Detection Methods and Tools: A Review*, International Journal of Computer Applications, **169**(8): p. 22-27, 2017.

[55] Sur, C., *DeepSeq: learning browsing log data based personalized security vulnerabilities and counter intelligent measures*, Journal of Ambient Intelligence and Humanized Computing, **10**(9): p. 3573-3602, 2019.

[56] Salas, M. I. P., Martins, E., S*ecurity testing methodology for vulnerabilities detection of XSS in web services and WS-security*, Electronic Notes in Theoretical Computer Science, **302**: p. 133-154, 2014.

[57] Palsetia, N., Deepa, G., Khan, F. A., Thilagam, P. S., Pais, A. R., *Securing native XML database-driven web applications from XQuery injection vulnerabilities*, Journal of Systems and Software, **122**: p. 93-109, 2016.

[58] Zhao, B., Ji, S., Xu, J., Tian, Y., Wei, Q., Wang, Q., Beyah, R., *A large-scale empirical analysis of the vulnerabilities introduced by third-party components in IoT firmware*, In Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, p. 442-454, 2022.

[59] Zech, P., Felderer, M., Breu, R., *Knowledge-based security testing of web applications by logic programming, International Journal on Software Tools for Technology Transfer*, **21**: p. 221-246, 2017.

[60] Sharma, C., Jain, S. C., *Analysis and classification of SQL injection vulnerabilities and attacks on web applications*, In 2014 International Conference on Advances in Engineering & Technology Research, p. 1-4, 2014.

[61] Mary, D. S. N., Begum, A. T., *An algorithm for moderating DoS attack in a web-based application*, In 2017 International Conference on Technical Advancements in Computers and Communications, p. 26–31, 2017.

[62] Delamore, B., Ko, R. K., *A global, empirical analysis of the shellshock vulnerability in web applications*, In 2015 IEEE Trustcom/BigDataSE/ISPA, p. 1129–1135, 2015.

[63] Qianqian, W., Xiangjun, L. *Research and design on Web application vulnerability scanning service*, In 2014 IEEE 5th International conference on software engineering and service science, p. 671–674, 2014.

[64] Olivo, O., Dillig, I., Lin, C., *Detecting and exploiting Second Order denial-of-service vulnerabilities in web applications*, In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, p. 616–628, 2015.

[65] OWASP, *OWASP Top 10 Vulnerabilities*, https://owasp.org/Top10/ (accessed Aug. 24, 2021), 2021.

[66] CWE, *Common Weakness and Enumeration*, Mitre, https://cwe.mitre.org/index.html (accessed Apr. 05, 2022), 2022.

[67] CWE, *2021 CWE Top 25 Most Dangerous Software Weaknesses*, Mitre, https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html (accessed Apr. 05, 2022), 2021.

[68] CWE, *CWE Top 25 Most Dangerous Software Weaknesses*, Mitre. https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html (accessed Apr. 05, 2022), 2022.