Journal of Informatics and Web Engineering
Vol. 3 No. 3 (January 2026) eISSN: 2821-370X

A Review of Cyber-Attacks on Web Applications and Their Countermeasures

Salaheddin Beskri[1*], Kok Why Ng[2]
[1]Department of Computer Science, Tripoli College for Sciences and Technology, Hay al-andalus District, Tripoli-Libya
[1,3]Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia
**corresponding author: (eng.salaheddin@gmail.com; ORCiD: 0009-0002-5131-5697)*

Abstract - Leaders have always wondered where and how precisely Information Technology (IT) can be of value in their organizations. However, the pacing rate of technology development and cutting-edge features that technologies can provide can cause the decision-maker of an organization to rush to invest in technology upgrades without taking risks into account. The risks that may cost the existence of organizations can be caused by an adversary who commits a cyberattack against the IT assets of an organization, seizing leaders' underestimations of the risks of IT upgrades. Inevitably, there will always be new vulnerabilities in IT assets, especially in information systems. This study aims to analyse the most recent cyberattacks and their criticalities. In addition, a review of countermeasures was conducted to withstand the attacks. Furthermore, this study addresses the key factors that contribute to the neglect of vulnerabilities in web applications. This study provides researchers and organizations with a review of recent real-world cyber-attack incidents using industrial reports and case studies as sources. The results of this study will provide practitioners and researchers with blueprint knowledge about cybercrimes and how critical cybersecurity is, especially in the field of software engineering, regardless of the size of the development team.

Keywords—Cybersecurity, Web Application, Vulnerabilities, Injection Attacks, Mitigations, Software Design

Received: 04 December 2025; Accepted: 20 March 2026; Published: 16 June 2026

1. INTRODUCTION

Cybersecurity management is considered the responsibility of Information Technology (IT) teams worldwide [1]. In fact, organizations' core functions have been relying heavily on IT, and any malfunctioning activity found in IT assets will have a critical impact on the organization's workflow and reputation. However, researchers have been conducting IT-based risks, determining the proper methods and approaches to avoid and mitigate the potential future risks that might be encountered. Currently, most of the world's social, cultural, financial, medical, and governmental activities are conducted in cyberspace [2]. The space has been filled with all information, either public or sensitive, by the said activities, and any insecurity or challenges in cyberspace may threaten many citizens' lives.

The cost of neglecting vulnerabilities in the cyberspace infrastructure, either software or hardware assets, is significant [3]. The United Kingdom's National Cyber Security Centre (NCSC) reported that in 2024, there were 430 cyberattacks recorded in 2024 and 89 of which were committed only against the UK [4]. These attacks cost more than a million dollars. Cyberspace, which all parties worldwide have been using has no borders and requests

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

no visa to enter any of cyberspace's assets. Hence, the urge to pay more attention to protecting cyberspace must be the responsibility of organizations, users, customers, governments, and communities around the world. In other words, vulnerabilities in our system should be given the highest priority, as well as the involvement of top-down management.

The main goal of the cybersecurity team is to achieve zero vulnerability in the IT infrastructure. The moment an IT asset becomes vulnerable, cybersecurity teams start to carry out the vulnerability and add every effort to mitigate it. As the team is trying to patch the vulnerable system, the system is still in danger and is exposed to an anonymous attacker. New attacks against leading web applications have been reported every day because of their vulnerabilities that have not been mitigated [5]. Fortunately, Web applications are technically accessible at the application layer. Cybersecurity teams tend to tighten access policies in the lower layers to prevent hackers from reaching the production environment. However, cybercrimes are still being reported and questioned.

This study reviews recent attacks discussed in empirical case studies and industrial reports, providing a review of how those attacks were successfully committed and how cybersecurity teams found the attack. Furthermore, this study reviews the countermeasures that have been taken to remedy attacks and the key factors that contribute to the negligence of web application vulnerabilities.

This research adds the following contributions to the domain of web application security:

- Proposes a Case Enhanced-Systematic Literature Review (CS-SLR) approach that addresses a key limitation of systematic literature review's detachments from operational reality by systematically integrating academic findings with real world breach incidents.
- Provides web application cybersecurity experts with evidence based exploited vulnerabilities from real incidents, discovering discrepancies between academic findings and cyberattacks prevalence.
- Identifies critical factors that contribute to vulnerabilities mitigation's delay across sectors.
- Identifies countermeasures that impacted organizations and companies were conducted, aligning academic research recommendations with incidents' real responses.

2. LITERATURE REVIEW

Web applications are considered the weakest point in cyberspace to attack due to their common vulnerabilities and online availability [3]. Factors that have created vulnerabilities vary in a way that different aspects play roles in vulnerabilities existence in web applications, either technical or non-technical. Since cyberattacks incidents have been soaring due to the rise dependence on information systems to deliver services around the world, cybersecurity researchers and experts from different organizations have been building security tools and attacks patterns awareness for all users around the world. The subsequent sections provide a brief background of non-profit organizations that run cyberattack awareness campaigns, as follows:

2.1 The Open Worldwide Application Security (OWASP) Top Ten Vulnerabilities

OWASP project is a nonprofit foundation that has been in chase for the most common flaws in web applications that attackers can exploit, although it frequently publishes an up-to-date list of the most common vulnerabilities for practitioners' and researchers' references, cyberattacks are still reported [6]. Lists are updated frequently by certified contributors who export newly discovered vulnerabilities in the form of a dataset to the OWASP project's teams. However, the lists of vulnerabilities are built by collaborative contributions of trusted organizations and experts around the world, reflecting a retrospective consensus and that, often, lags the emergence of new vulnerabilities exploitation [7].

2.2 The Web Hacking Incident Database Program (WHID)

Apart from the OWASP project's top ten list of web application vulnerabilities efforts to make organizations and researchers aware of the vulnerabilities, another non-profit program has been established recently by the OWASP foundation, namely the WHID program [8]. The program focuses primarily on cyberattack incidents that have been reported by organizations and businesses and stores all incidents in a large database for further research and awareness considerations. The WHID database provides incident details in relation to every reported vulnerability

by the OWASP top ten lists. The WHID methodology relies on third-party media to report incidents which may not comprehensively capture all incidents as well as technical details [8].

2.3 The Common Vulnerabilities Exposure Program (CVE)

The MITRE foundation's CVE program has added more advantages to practitioners and researchers by covering the gap of common vulnerabilities that an organization or business might find in one IT asset [9]. There are 433 companies and organizations that have partnered with the program, and vulnerabilities in the IT infrastructure will eventually end up in the CVE program repository.

These non-profit programs should be part of the everyday tasks of an IT management as their databases are provided in the GitHub repositories, and they can be easily imported into any information system [6], [8], [9]. However, the non-profit organizations publications are limited by the scope of datasets imported from different sources, the exclusion of emerging threats and the lack of detailed context regarding the application of security measures in real scenarios as IT managements differ in IT infrastructure's size, complexity and geography[7].

2.4 Web Application Cybercrimes' Reviews

Cybercrime is defined as an illegal activity conducted on an IT asset or an IT asset that plays an unintentional role in creating an illegal activity on a victim's IT asset [2]. The number of cybercrimes against web applications has soared recently, [10] and frequently reviewing recent cyberattacks on web applications has contributed significantly to this field of study [3].

Many studies analysed major web application vulnerabilities. In 2025, Riskhan et al. [11] investigated seven major web application vulnerabilities, including SQL injection, Cross Site Scripting (XSS), insecure configuration, Cross Site Request Forgery (CSRF), insecure file upload, Zero Day flaws and insecure direct object reference by systematically conducting security assessment using automated penetration-testing tools and controlled lab environments. This study demonstrated in detail how these vulnerabilities can be identified and recommended strategies that prevent exploitation. While the study provides valuable technical insight, it does not incorporate real world attack incidents and its findings relied on controlled environmental scenarios in which tool-based assessments were involved. In addition, vulnerabilities persistence root cause that contribute to delayed mitigation and further exploited vulnerabilities are not examined. The countermeasures were prescribed in best practices that have been recommended by cybersecurity industry rather than as reactive responses derived from real-world breached organizations. These gaps underscore the limitations in web applications' vulnerabilities and countermeasures research, motivating the adoption of the proposed CS-SLR approach that integrate academic findings with real world web application attack incidents to align exploited vulnerabilities, root causes of unmitigated vulnerabilities and factual impacted organizations countermeasures.

A study by Phanireddy et al. [12] explored the integration of AI-driven techniques in web application vulnerabilities detection and mitigation, depicting how Machine Learning (ML) can enhance web application security beyond traditional vulnerability scanners and penetration testing tools. The study proposed a hybrid framework that integrates ML based source code inspection with runtime behaviour monitoring. A deliberate testbed and a controlled e-commerce environment were involved in the evaluation of the framework, resulting in reduced false positives, detection accuracy and rea-time responses. While this study addressed vulnerabilities detection capability by leveraging AI, it does not articulate how adversarial AI technique and Large Language Model (LLM) generated attacks evade vulnerabilities detection techniques. Including these AI based threats is critical and real-world attacks incidents gradually embracing AI-assisted exploitations and adaptive evasion techniques. The CS-SLR approach comprehensively synthesizes the vulnerabilities exploitations vector, mitigation techniques and countermeasures, including AI driven attacks that reinforce bridging a gap between academic research and real-world incident evidence.

A review study was conducted in 2024 in the financial sector to assess the impact of cyberattacks on financial markets rather than focusing on exploited vulnerabilities or technical details [13]. The study uses the European Financial market as a targeted sample in its methodology. A mixed method approach was adopted in which a quantitative analysis of European market data and qualitative case studies of cyber incidents in the financial sector were combined. As a result, this study found that social engineering and ransomware attacks have the highest impact on the financial market sector, and the losses will reach 10.5 trillion USA Dollars yearly by 2025. In addition, the study highlighted that the IT infrastructure of a financial sector is huge and complex, therefore the average vulnerability is high, as well as their broad impact. While these findings provide valuable insight into the broader consequences of social engineering and ransomware cyber-attacks on financial sectors, the study does not review vulnerabilities exploitation at web applications' level. This separation between high level impact analysis

only while analysing the case studies and exploited vulnerabilities technical analysis depicts broader fragmentation in cybersecurity research in which every dimension of organizational, economic and technical are analysed in isolation. The adoption of the CS-SLR approach addresses the fragmentations by synthesizing evidence across various attack vectors, preserving a united analytical framework.

A comprehensive review of the past 20-year studies towards XSS attacks was conducted in 2024 [14]. This study aims to reveal the gaps that have been missing during the development of attack-detection techniques, as well as in the proposed approaches since the XSS attack was discovered in 1999. The research identified a significant gap between the academic attention which was intensified in 2016 and the discovery of the attack in 1999. This lag reveals a broad disconnect between academic research focus priorities and real-world incidents' exploitation. Moreover, although the study introduces a taxonomy for newer variants of XSS attacks, the study lacks industrial incident data as it relies on academic sources. Therefore, it does not introduce how frequently these attacks were exploited in practice or why these vulnerabilities were not mitigated despite extensive research. Thus, the CS-SLR approach covers these gaps by integrating academic research findings and industrial reports with real world cyberattacks incidents.

Almaiah et al. [15] conducted research in 2024 in which a comprehensive analysis of cyberthreats against Database Management System (DBMS) with emphasis on SQL injection attacks. Besides, it examined the organizational responses to the DBMS-centric cyber-attacks that had been taken by impacted organizations' IT infrastructures. Thus, this study proposes a risk assessment framework to provide an approach for addressing all types of threats, vulnerabilities, and countermeasures. The framework comprises the following four stages.

1) Key components identification.
2) Threat identification.
3) Vulnerability identification.
4) Countermeasure identification.

The findings indicate that twenty-eight collected threats were analysed and classified as technical and non-technical threats. The study found that 9% of the threats against database systems were caused by SQL injections and Denial Of Service (DOS) attacks. In addition, this study revealed that weak authentication, unpatched databases, weak audit trails, and multiple usages of a single account were the most prevalent technical vulnerabilities in database systems. While this study provided valuable insight into real world countermeasures against DBMS systems, it remains partially relevant to the scope of web application security and does not relate to evidence from web application real world incidents. These limitations reinforce the need for the CS-SLR that integrate academic and incident levels evidence across diverse web application cyber-attacks within a unified evaluative model.

A comprehensive research review of security threats against Global Software Development (GSD) SDLC model phases was conducted in November 2022 [16]. The study was motivated by findings from a previous study by the same authors, which revealed that the lack of prioritizing security is one of the common causes of vulnerabilities exploitation [17]. Therefore, this research was mainly aimed at prioritizing the critical software security risks in the SDLC phases. A previous review paper sampling of 120 papers, conducted by the same authors in November 2021, was used in this study [17]. A total of 145 software security risks were identified based on the inclusion, exclusion, and quality rating criteria, although the data source did not include industrial insights and reports. Khan et al. [16] adopted a non-methodological method to assess the practical significance of software engineering owing to the direct involvement of the targeted population.

Besides, an online questionnaire survey was constructed to validate the findings of previous studies and to discover other security risks and associated practices. Fifty responses were retained from experts working in a GSD environment (Software Engineering Research Group, Pakistan; The King Fahd University of Petroleum and Minerals, Saudi Arabia; and Qatar University, Doha, Qatar). Finally, the Fuss Analytical Hierarchy Protocol (FAHD) was implemented to rank the discovered risks based on their importance in securing the SDLC phases in the context of GSD. Consequently, forty-five software risks were identified and mapped for each phase of the SDLC using the FAHP technique. After the Critical Software Risks (CSR) are converted into triangular fuzzy numbers (TFN) using a geometer mean method, all CSRs in each level of the hierarchy are compared together with a pairwise comparison of each other and then compared with pair-wise comparison of all categories of requirements engineering, designing, coding, deployment, testing, and maintenance. This study revealed that the design phase is the most important phase among other SDLC phases and the lack of threat model updating, output validation, certification in final release and archive, and spoofing were the most Critical Software Risks in the SDLC phases in the GSE context.

However, the study relied solely on experts' reviews who work at a research centre and two universities to classify and prioritize CSR risk, overlooked industrial reports, non-profitable organizations' databases and real-world

incidents that may have a significant impact on the results. This reflects the need for the CS-SLR methodology that integrates academic research findings with real world incidents in a unified analytical framework.

In 2020, Sohan et al. [18] conducted a Systematic Literature Review (SLR) review and qualitative analysis of JavaScript Malware Detection (JSMD) techniques from 2009 to 2019. The SLR process investigated thirty-two articles that mostly used ML malware detection models. This study aimed to assess JSMD techniques for malware detection using their results. The findings of the results concluded that most of the ML models were performed on average, but the study reported uncertainty in the results of all those results due to the publication bias of reviewers of scientific journals in which positive results were published over negative ones. In addition, the study revealed the difficulties that had been encountered in performing ML models to verify the results due to different dataset usage, and private datasets were used to train their models that the study could not access. Although the performance assessment process suffered certain limitations, the study revealed significant facts that were reported by the reviewed studies.

1) The performance of the ML models was affected by browser discrepancies. In other words, different browsers have distinctive characteristics; however, it is challenging to detect malware when a webpage is opened with various browsers.
2) The processing time of the detection models is challenging, and the studies proposed that further research is required to reduce it.

Sohan et al. [18] solely relied on the academic literature (conferences and journals) to review vulnerabilities and JSM detection techniques while incapable of validating performance due to lack of datasets availability. This reflects the traditional SLR's major limitations in which its design is isolated from operational real world incidents technical details and that is not a flaw in its execution. The proposed CS-SLR methodology in this work addresses these limitations through methodological integration.

To raise awareness of Social Engineering Attacks (SEA) among organizations and individuals, a comprehensive survey was conducted in 2025 by Rathod et al. [19]. Motivated by loopholes, which have not been sufficiently investigated, this study analysed industrial reports and ten distinct social engineering attack types and built a detailed solution taxonomy to derive solid knowledge about the attributes of social engineering attacks, namely definition, implementation, and real-world cases. The study relied on Tier 1 academic journals to synthesize recent related publications regarding the SEA and found that the SEA's taxonomies have not been discussed. In addition, the solutions provided were impractical and were not built on real-world cases. The study proposed an AI and blockchain-based malicious URL detection framework to prevent SEA attacks. Hence, an experiment was conducted in which two popular Meta-platforms, Facebook and Instagram, were targeted. Several URLs on the platforms were collected and pre-processed for storage in a database. The framework, using AI and blockchain, was successfully capable of detecting malicious URLs in a database. The naïve Bayes algorithm was found to be the best performer in terms of speed, time, and training cost. While the study advances the traditional SLR practice by including real-world cases, incidents data were primarily served for illustrative purposes rather than driving systematic integration for risk-based reprioritization and vulnerabilities exploitability validation. This highlights more structured frameworks needed like the proposed CS-SLR that systematically produces actionable insights.

In 2024, research reviewed recent ransomware cyberattack incidents to identify the nature of the attack and organizational responses across divers sectors using a proposed counteract framework [20]. The researcher categorized ransomware attacks into three types: screen locking, file encryption, and double extortion ransomware. The research method employed searching tools to collect and review relevant papers, such as Google Scholar, IEEE Xplore, MDPI, and Grover Hermann Library at UC. In addition, reputable, independent new blog posts were used in the research. Consequently, six cyberattack incidents relevant to ransomware attacks were identified and reviewed. The study found that there are many factors that contributed to successfully executing ransomware attacks against computer systems, and that financial gain was the main motivation. This study determined nine factors that lead to successful ransomware attacks. Most of the identified factors were human error contributions to the incident and web application exploitation was the only technical factor. The framework comprises three strategic alignments of preventive, protective, and deterrent measures.

1) Endpoint Detection and Response (EDR)
2) Threat intelligence
3) Secure Management of Privileged Credentials
4) Multi-Factor Authentication (MFA)
5) Backup strategies
6) Network and Host-based detection
7) Incident Response Plan
8) Patching Cadence

9) Training awareness
10) No More Ransome (NMR) Initiatives

While the findings of this study provide valuable insight, its methodology primarily relied on illustrative aggregation of academic research, industrial reports and media, excluding a methodological synthesis of real-world cyber-attack incidents and mapping between vulnerabilities and exploited patterns and countermeasures. This gap highlights the significance of CS-SLR methodology that integrates literature reviews with empirical cyber-attacks evidence to generalize vulnerabilities exploited, defensive strategies and rather to address ransomware attacks only.

Khan et al. [21] conducted a systematic literature review of the 2019 data breach against the Capital One bank. Capital A hierarchical control structure was reconstructed to identify technical failures and why they failed. The Cybersafe methodology analysed cybercrimes from technical issues up to top management, the Board of Directors, and government regulators. Even though the study focused on one real-world cybercrime, which took two years, the findings provided many key factors for the failure to prevent cyberattacks. They recommended measures that could be critical to other organizations to secure their systems. The Cybersafe 4-step processes cost the researchers approximately 11 months of research to deliver accurate results. These steps include collecting the required information about the incident, building a hierarchical functional control structure where all controllers and interactions were modelled, examining the control structure functionality against the attack, and identifying the flaws and conditions of every control structure that lead the system to behave ineffectively and create recommendations for the safety control structure. The study demonstrated that the incident was not caused due to technical flaws only but a cumulative of failures encompassing different levels. The study found that one of the major risks that played a main role in the Capital One Bank incident was the misunderstanding of the shared security responsibility model with respect to the cloud service provider. The study provided rich qualitative insight into the governance failure and root causes within a single large-scale incident though it does not review and align systematically exploited vulnerabilities across many real-world incidents with academic taxonomies nor does it analyse the post incident countermeasures adopted by the impacted organization. These motivate the adoption of the CS-SLR approach in which generalizes such case evident insights by mapping real world cyber-attacks vulnerabilities to academic research findings and industrial reports, determining root cause of unmitigated vulnerabilities and reviewing countermeasures across multiple incidents.

To better comprehend the literature synthesized in this study, Findings and limitations of every literature are summarized in Table 1.

3. RESEARCH METHODOLOGY

A CS-SLR was conducted in this study. CS-SLR is a research method that reinforces the literature with real-world case studies that can be obtained from published or unpublished sources. The CS-SLR method was inspired by the Multivocal Literature Review (MLR) research method developed by Zhao et al. [22]. While the MLR method is specially designed to use formal and commercial, either published or unpublished, the CS-SLR is a more case-prone method that aims to provide solid convergent views in answering research questions.

3.1 Objectives and Research Questions

This research provides the pattern of recent cybersecurity aspects for researchers and practitioners by reviewing recent relevant studies, supported by real-world case studies. This study aims to analyse the most recent cyberattacks and their criticality. In addition, countermeasures deployed to encounter attacks are also reviewed. Furthermore, the reasoning behind the negligence of vulnerability mitigation and remediation within organizations and companies was investigated.

The following are the research questions and associated sub-questions to be addressed.

- RQ1: What are the most recent web application vulnerabilities and countermeasures?
    - Sub-question 1: What have been the most discussed vulnerabilities and countermeasures in recent studies?
    - Sub-question 2: How are vulnerabilities and countermeasures discussed and aligned with the OWASP top ten list of vulnerabilities?
- RQ2: What vulnerabilities are seized and what countermeasures are applied against cyberattacks by the impacted organizations and companies?
- RQ3: What factors play a major role in delaying web application vulnerability mitigation?

Table 1. A Summary of the Literature Review

Ref	Year	Findings	Gap(s)
[11]	2025	- Seven major web application vulnerabilities were investigated - Identified vulnerabilities exploitation strategies and recommended how to preventive countermeasures	- Findings derived from a controlled lab environment. - Root cause of unmitigated vulnerabilities delays was not analysed. - Countermeasures were recommended based on industrial organization's best practices
[12]	2025	- Proposed a hybrid AI based source code vulnerability detection framework. - The hybrid framework achieved high detection accuracy, lower false positives and real-time responses.	- It does not articulate the AI assisted attacks prevention capability as they have been reported in real world incidents.
[19]	2025	- Social Engineering Attacks were mapped. - Proposed Malicious URL detection technique	- Exploited Vulnerabilities in real-world incidents were included as descriptive analysis.
[13]	2024	- Social Engineering and Ransomware are predominant - Average amount of vulnerabilities in financial sector is often high	- Exploited Vulnerabilities and countermeasures were Not technically Detailed. - Review from a financial analysis perspective.
[14]	2024	- A comprehensive taxonomy of the types of XSS attacks (Application-based, third party-based and collaboration-based)	- lacks industrial incident data as it relies solely on academic sources. - Root cause of unmitigated vulnerabilities delays was not analysed.
[15]	2024	- SQLi and DOS attacks against DBs were predominant	- Partially relevant to web application security - Real-world attack incidents were not included.
[20]	2024	- 9 factors related to human error contributed to the ransomware attacks whereas 1 factor was unpatched vulnerabilities.	- Technical vulnerabilities and countermeasures were not sufficiently discussed. - Real world cyber-attack incidents analysis was not considered in the finding's formulation.
[16], [17]	2022	- 45 software risks were identified, prioritized and mapped in each phase of the SDLC phases in the context of the Global Software Development	- No Real-World cyberattack incidents validation
[18]	2020	- The publication bias of reviewers of scientific journals in which positive results were published over the negative ones. - ML performances' verification was infeasible due to lack of datasets availability.	- Traditional SLR design in which real-world attack incidents were not reviewed.
[21]	2019	- The attack key factor was a misunderstanding of the shared security responsibility model with respect to the cloud service provider. - The study revealed multiple factors that were urged as to the cause and the environmental conditions	- The study investigated one study, the Capital One bank incident that will make the findings ungeneralizable

Figure 1 and Figure 2 show the CS-SLR framework implemented in this study. The research method in the CS-SLR framework begins with the research questions addressed in this research. Then, two search tasks began in parallel to retrieve white literature (research papers) and Gray literature (academic and industrial reports, works, blogs, and case studies). The Grey Literature (GL) is "Grey literature stands for manifold document types produced on all levels of government, academics, business and industry in print and electronic formats that are protected by intellectual property rights, of sufficient quality to be collected and preserved by library holdings or

institutional repositories, but not controlled by commercial publishers, that is, where publishing is not the primary activity of the producing body." [23].

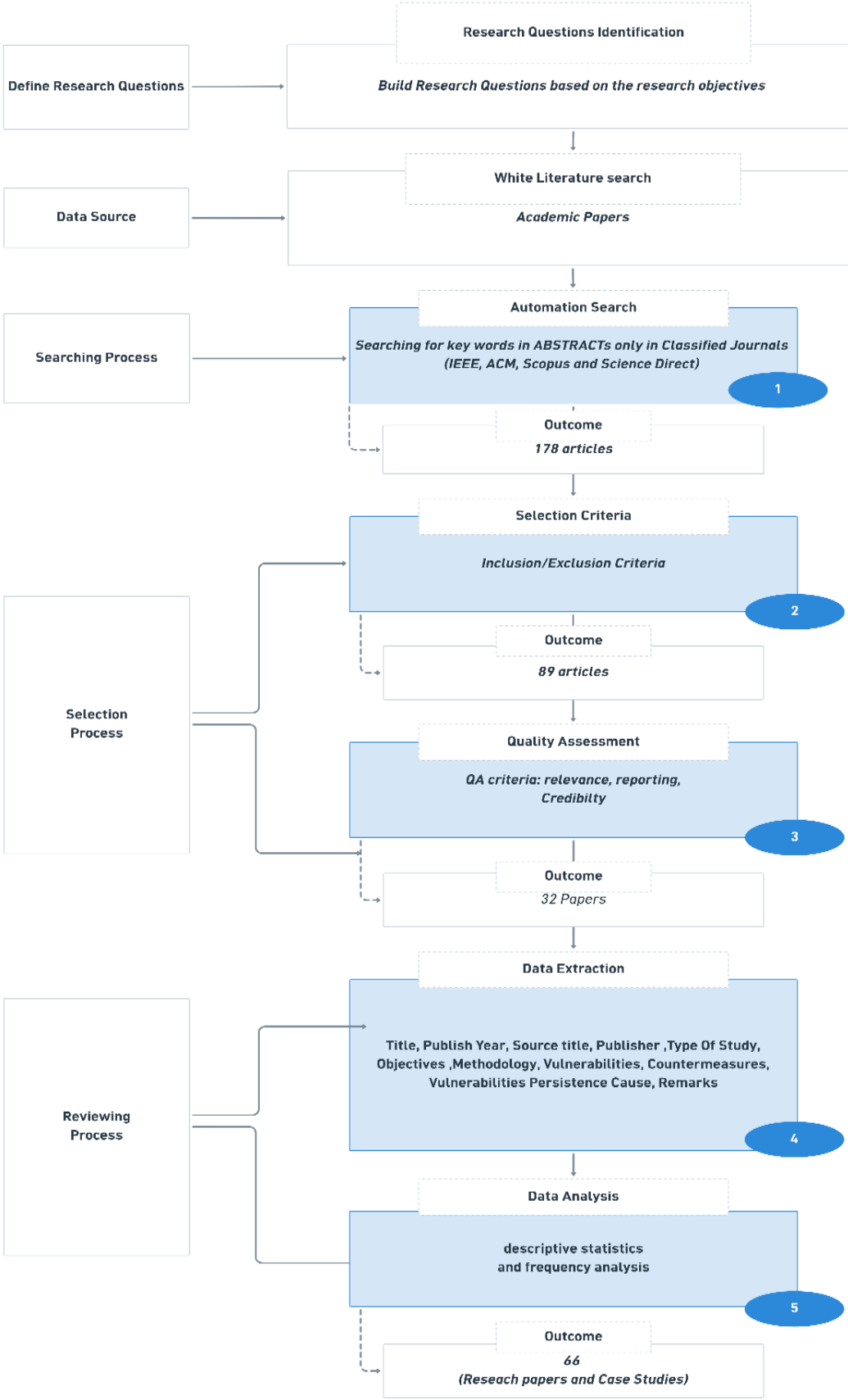

Figure 1. An Overview of the CS-SLR framework (White Literatures Process)

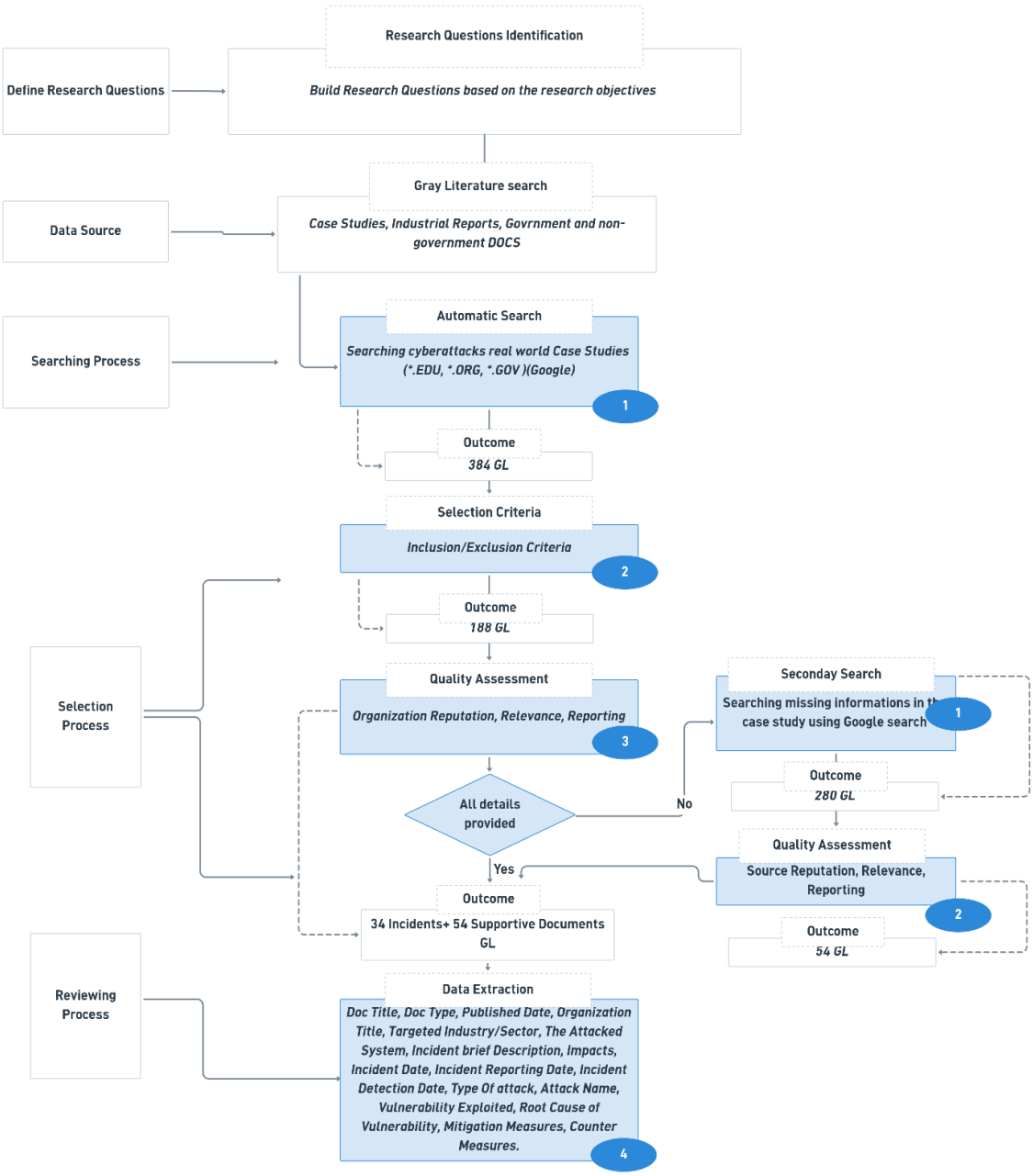

Figure 2. The CS-SLR Framework (Gray Literature)

3.2 Data Source

Various data sources were considered before the search process started. As this research method relies on different types of documents and data, academic sources such as Scopus, IEEE, ScienceDirect, and ACM libraries were included in this study's search tasks to retrieve, whereas Google engine was mainly used to collect real-world cyberattack incidents and reports. The selection of the Google search engine was based on the nature of the GL literature type, which can be an industrial, governmental, or academic document.

3.3 Searching Process

In this section, the process of collecting academic and industrial documents is described using formulated search strings against various identified data sources. Prior to applying the search processes, search strings were formulated to retrieve as many relevant results as possible. The search string formulas were built for the White Literature (WL) search and GL search. The subsequent sections discuss the search string and methods that were designed and applied.

1) WL Search String

The key string to retrieve relevant studies was (web application AND security AND vulnerability AND (survey OR review)). Since this study's focus is on web applications framework, the phrase "web application" was fixed in the search string. To address the possibility of missing any studies, the search string was conducted during the search process against the abstracts of recent papers.

2) GL Search Strings

Searching grey literature is challenging because real-world incidents are one of the main objectives of this research. GLs exist in various data formats, including industrial reports, web blogs, research papers, government reports, and jurisdictional verdicts. The search string (allintext: cyberattack incident "web application" site:*.edu; OR site:. org OR site:.gov file type:pdf) were searched using Google search engine.

To investigate each case study, secondary search processes were conducted in which the keyword strings of the cyberattacks were fed into the search processes, although supportive documents were retrieved.

3.4 Selection Process

In this section, the initial results of the three search processes were filtered based on a certain inclusion and exclusion list. The results were selected and critically assessed based on the search objectives.

3.4.1 Inclusion and Exclusion Processes

At this stage, the results obtained from the parallel search processes were categorized into included and excluded documents, based on the objective of each search process.

1) White Literature Inclusion/ Exclusion Criteria

As a result of the WL search, a total of 178 research papers were selected and filtered. Many criteria were applied to the search results to ensure the relevance of the research to the objectives of this study.

- The studies must be published in a period ranged from 2020 to 2024.
- The studies must be a journal or conference proceeding written in English.
- Inaccessible papers were eliminated.
- Irrelevant papers on Web application cybersecurity were excluded by skimming the title of each resulting paper.
- Duplicate papers were excluded from the results.

As a result of the applied criteria, 89 research papers remained for quality assessment.

2) Grey Literature Inclusion/ Exclusion Criteria

Unlike academic papers, 384 GL documents were retrieved from various sources in PDF extension file format. Hence, the applied criteria were aimed at retrieving the data from the most reliable sources. The following exclusion and inclusion criteria were used:

- The incidents were reported within the period that ranged from 2014 to 2024.
- The source of the case study must be educational, governmental, or organizational institutions.
- The documents must be written in English.
- Incidents must be relevant to web application attacks, as they are real-world incidents.
- Duplicate incidents were eliminated though the most reporting and credible documents were kept.

The application of this list yielded 188 GL documents. The remaining documents were different types of documents, such as reports, research papers, master 's or PHD theses, and law force investigations.

3.4.2 Quality Assessment

The second stage of the selection process involves revising every document to assess its quality based on the following three criteria: credibility, relevance, and reporting.

Unlike the approach of Zhao et al. [22], in which the quality of using one QA checklist for WL and GL was evaluated, the WL and GL documents were revised, and two QA checklists were created for each search process, resulting in documents to verify the criteria's accurate application.

The QA checklist for the WL documents consisted of several questions to be answered for each search process's results as follows:

- Is a paper published by a reputable publisher?
- Is a paper directly relevant to the web application attacks?
- Has a paper had a comprehensive research design?
- Has a study mentioned the vulnerabilities, mitigations and threat countermeasure?

The assessment of GL documents using the QA checklist was critically challenging because of unreliable and misleading victims' storytelling. Therefore, a set of subprocesses was conducted to retrieve missing and reliable information regarding certain incidents. The QA checklist was deployed in the incident assessment, as well as in the assessment of supportive documents.

The following QA checklist was used to retrieve the most reliable and reported investigative incidents:

- Has a document been published by a reputable institution?
- Has a document had a real-world incident?
- Were incidents relevant to web application attacks?
- Has a vulnerability been exploited by the attack mentioned?
- Has the incident been detailed?

A secondary search was conducted throughout the quality assessment process to complete the missing information regarding the incidents.

1) Secondary Search

Most incidents were not comprehensively documented in one document, although further search and quality assessment were involved in building fully covered case investigations.

In the secondary search, a common title for an incident was used in the Google search engine for every case study, and the first page of every search result was considered, as it includes the most relevant results to the searched incidents. As a result, 280 results were selected for further quality assessment.

2) Quality Assessment

In the secondary search, the results were assessed using the GL documents' QA checklist, as stated in Section *3.4.2*, because the nature of the documents and the objective of this search was to investigate a real-world web attack incident.

As an overall result of the quality assessment, 32 research papers, 34 real-world web application incidents and 54 supportive documents for the incidents passed the quality assessment, Figure 3 and Figure 4 show the numbers of included GL and WL documents that they discussed or reported incidents. In section *F*, WL and GL data were extracted and analysed to answer the research questions.

3.5 Reviewing Process

At this stage, the selected data were extracted and analysed to answer the research questions.

3.5.1 Data Extraction

After the selection stage was completed, important and relevant data and information on the main objectives of this study were extracted. Research papers and case studies were carefully and manually synthesized, and data were extracted based on the nature of each document and its purpose. Data were extracted from the selected academic papers, as shown in Table 2.

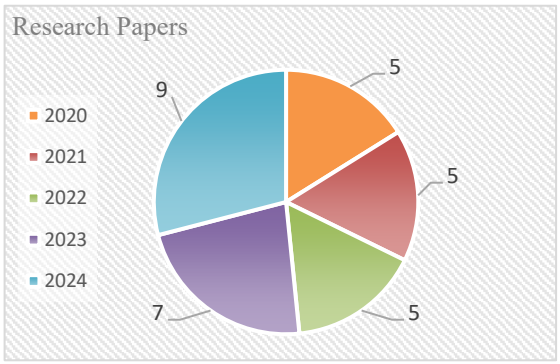

Figure 3. Numbers of Collected Research Papers

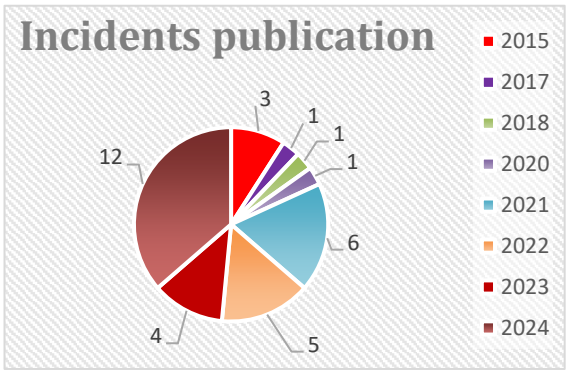

Figure 4. Numbers of Web Application Attacks Incidents

Table 2. White Literature Extraction Table

Key
Title
Publish Year
Source title
Publisher
Type Of Study
Objectives
Methodology
Vulnerabilities
Cyberattacks Countermeasures
Vulnerabilities Persistence Cause
Remarks

Unlike the WL studies, the grey literature documents were collected to represent most of the web application real-world cyberattack incidents, although the extracted data were purposefully selected, as depicted in Table 3. The extracted data were documented and analysed by the authors to answer the research questions. In the following section, we provide descriptive analysis and research answers for future studies and practitioners.

Table 3. Grey Literature Extraction Table

Key
Case. Study Doc Title
Doc. Type
Published Date
Organization Title
Targeted Industry/Sector
The Attacked System
Incident Description
Impacts
Incident Date
Incident Reporting Date
Incident Detection Date
Type Of attack
Attack Name
Vulnerability Exploited
Root Cause of Vulnerability
Mitigation Measures
Threats Countermeasures

3.5.2 Data Analysis

The extracted features were analysed and statistically processed using data analysis tools. 66 research papers and cyberattack incidents were processed to discover the correlation between incidents and academic research in web application security.

4. RESULTS AND DISCUSSIONS

In this section, the research questions are answered based on an in-depth analysis of the extracted data. However, vulnerabilities and countermeasures have been studied from different perspectives in review papers in which different research designs were adopted by reviewers. Table 4 classifies the papers based on their research designs.

Table 4. Classifies Papers Based on Their Research Designs

Research	Number of Papers	Papers
SLR	17	[14],[24],[25], [26], [27], [28], [29], [30].[31], [32], [33], [34], [35], [36],[37], [38]
Survey	4	[39], [40], [41]
Experimental Study	8	[42], [43], [44], [45], [46], [47], [48], [49]
Comparative Analysis	2	[50], [51]
Literature Review	1	[52]
Systematic Mapping	4	[36], [37],
Exploratory Study		[44]
Case Study	2	[45], [48]
Experimental Research	1	[49]
Systematic Literature Mapping (SLM)	1	[37]

4.1 RQ1: What are the most recent web application vulnerabilities and countermeasures?

To answer the question, a search string was applied in verities of reputable digital libraries seeking relevant studies which were published for the period from 2014 to 2024. On the other hand, real-world cyberattack incidents that have been published between 2020 and 2024 were collected to achieve credibility in answering the research questions.

The study discovered that web applications' vulnerabilities have drawn worldwide attention owing to their severe impact. Although researchers have been conducting intensive research in such a significant field of study, there are still vulnerabilities in information systems, as reported by industries and researchers. Thus, it is critical to firstly explore what vulnerabilities and countermeasures are examined in academic literature. Accordingly, the answer to Sub-question 1 reveals the discussed web application vulnerabilities and countermeasures recommended in peer reviewed studies.

4.1.1 Sub-question 1: What are the most discussed vulnerabilities and countermeasures throughout the recent studies?

Several vulnerabilities have been discussed by researchers, whereas other vulnerabilities have not been sufficiently studied. Based on the frequencies of the studies that discuss vulnerabilities, Given the total number of thirty-two reviewed papers (T=32), A Vulnerability Density (VD) was computed to normalize the academic attention paid for each vulnerability, providing proportional comparison among them. Equation (1) computes vulnerability density across the reviewed papers.

$$\mathrm{VD} = \frac{Number\ of\ Mentions\ in\ Reviewed\ Studies\ (N)}{Total\ Number\ of\ Studies\ (T)} \qquad (1)$$

To obtain an analytical result from the vulnerabilities discussed in research, vulnerabilities were ranked based on their research attention VD from high to low, as shown in Table 5. Significant gaps can be observed in research attention across web application vulnerabilities discussed in the collected studies between 2014 and 2024. Table

5 revealed a clear dominance of XSS attacks in research, which was scored 0.56 VD, reflecting 56% of research attention paid throughout ten years, exceeding SQL Injection and CSRF with 0.19 and 0.16 VD respectively. Moreover, moderate research attention was revealed towards other severe vulnerabilities namely, Improper Input Validation, Authentication and Authorization weaknesses, Web Server and Security Misconfiguration and Source Code and Information Disclosure with 0.13, 0.13, 0.13 and 0.09 VDs accordingly. Whereas File Upload Issues, Path traverse, Unvalidated redirected and Forwards and Domain Spoofing were understudied as they scored 0.03 VD each even though their impact is severe. These findings show that literature strongly focuses on injection based and front-end vulnerabilities due to their ease of prevalence and alignment with industrial reports such as the OWASP project. However, this study discovered that AI-driven web application vulnerabilities throughout the period from 2014 to 2024 were often perceived merely as supporting mechanism to reconnaissance, phishing and automation rather than exploitable vulnerabilities [12] while AI driven countermeasures were viewed as overly complex due to their limitations such as reliability, interpretability and generalizability for all applications and they are still maturing [53].

Table 5. Vulnerabilities Density in Research

ID	Research Attention Priority	Vulnerability	Studies Freq(N)	VD
V1	1	Cross-Site Scripting (XSS)	18	0.56
V2	2	SQL Injection	6	0.19
V3	3	CSRF (Cross-Site Request Forgery)	5	0.16
V4	4	Input Validation Issues	4	0.13
V5	5	Authentication and Authorization weaknesses	4	0.13
V6	6	Web Server and Security Misconfiguration	4	0.13
V7	7	Source Code and Information Disclosure	3	0.09
V8	8	File Upload Vulnerability	1	0.03
V9	9	Path Traverse	1	0.03
V10	10	Unvalidated Redirects and Forwards	1	0.03
V11	11	Domain Spoofing	1	0.03

Web Application vulnerabilities are still prevalent, and IT managements are challenged to achieve consensus perspective on specific countermeasures assignment to every day reported new vulnerabilities. In academic research, vulnerabilities have been encountered by various recommended security measures that can be adopted by IT managements with critical factors being primary considerations such as their associated complexities. Having identified vulnerabilities in literature, this study subsequently synthesizes the recommended countermeasures.

To synthesize countermeasures in research, countermeasures were counted when a paper clearly recommended, suggested, discussed or evaluated distinctively regardless of how many times a countermeasure was mentioned in the same paper.

Also, a countermeasure that was mentioned while describing a vulnerability was excluded. Moreover, countermeasures were described in different labels, and they are mapped into standardized countermeasure categories. This avoids overlapping countermeasures that may inflate counts. As a result, the number of studies recommending a countermeasure is derived for every countermeasure across all reviewed studies. Given the total number of reviewed thirty-two papers (T=32), the Countermeasure Recommendation Density (CRD) is computed for every recommended countermeasure category using a formula, as shown in Equation (2).

$$\text{CRD} = \frac{Number\ of\ given\ countermeasures\ in\ Reviewed\ Studies\ (N)}{Total\ Number\ of\ Studies\ (T)} \qquad (2)$$

The discussed countermeasures and their recommendation density in research are given in Table 6. The frequency of recommendations CRD enables proportional comparison regardless of real-world incidents countermeasure effectiveness. As a result, the synthesis of Countermeasures in academic research shows an explicit dominance of preventive countermeasures at the technological level. The source code testing using SAST and DAST techniques is observed as the most emphasized countermeasure, with 0.88 CRD representing 88% of research recommendations. Other technological level countermeasures received high recommendations scores such as frond-end security controls, including CSP and secure headers which was scored 0.59 CRD corresponding to 59%

of the total recommendations. In contrast, AI-driven detective countermeasures including supervised and unsupervised ML models were discussed less frequently and often reported insufficiently validated in real world deployment which scored 0.53 CRD accounting 53% of research recommendations. The CRD based synthesis revealed that research recommended fundamentally preventive countermeasures over AI driven automated countermeasures, emphasizing the significant need for future research work to bridge this gap. Given the synthesis of vulnerabilities and countermeasures, a structured and evidence-based foundation is ready for aligning with the recently published OWASP top ten list of common vulnerabilities.

Table 6. The Countermeasures Recommendation Density in Research

ID	Normalized Countermeasure	Top Category	Industry Category	Studies (N)	CRD
C1	Static, Dynamic and Hybrid Application Security Testing (SAST, DAST)	Automated Vulnerability Detection	Technological	28	0.88
C2	Content Security Policy (CSP), Secure headers (X-Content-Type-Options, Cache-Control)	Browser-side Security Controls	Technological	19	0.59
C3	Input validation & sanitization (allow-list, escaping, regex)	Secure Input Handling	Technological	18	0.56
C4	ML–based detection (DL, CNN, GNN, AE)	ML-Based Detection	Technological	17	0.53
C5	Parameterized queries / prepared statements	Secure Database Interaction	Technological	14	0.44
C6	Secure coding practices & developer awareness	Human Centric Controls	People	13	0.41
C7	Secure SDLC / DevSecOps integration	Secure SDLC Integration	Organizational	10	0.31
C8	CSRF protection (tokens, synchronizer pattern, CSRFGuard)	Session & Request Integrity	Technological	9	0.28
C9	Web Application Firewall (WAF)	Perimeter Defence	Technological	8	0.25
C10	Logging, monitoring & real-time analysis	Security Monitoring & Detection	Technological	7	0.22
C11	Cryptography best practices	Secure Encryption	Technological	7	0.22
C12	Automated code refactoring tools	Automated Code Remediation	Technological	6	0.19
C13	Authentication & Authorization (RBAC)	Access Control Handling	Technological	6	0.19
C14	Secure API usage & templating	Safe APIs	Technological	6	0.19
C15	Organizational policies & management support	Policy Controls	Organizational	6	0.19
C16	Secure configuration of parsers (XML)	XML parser hardening, framework configs	Technological	5	0.16
C17	Database security reviews & testing	Database Audits	Technological	5	0.16
C18	Vulnerability disclosure & reporting practices	Vulnerability Disclosure Practices	Organizational	4	0.13
C19	CAPTCHA / bot mitigation	Bots Prevention	Technological	4	0.13
C20	Email / SMTP security controls	Email Security	Technological	3	0.09
C21	Honeypots	Deception-Based Protection	Technological	2	0.06

4.1.2 Sub-question 2: How are vulnerabilities and countermeasures discussed and aligned with the OWASP top ten list of vulnerabilities?

To align web application vulnerabilities and countermeasures discussed in Sub-question 2, the study maps each vulnerability and countermeasure to their corresponding categories in the OWASP top ten 2025, as shown in Table 7.

Table 7. Research-recommended Vulnerabilities Mapping

VD	Research-Discussed Vulnerability	Mapped to CRD-based Countermeasure	Mapped OWASP Top 10 List
V1	XSS	Secure Input Handling (C3) Browser-side Security Controls(C2) Safe API (C14) Automated Vulnerability Detection(C1)	A05: Injection
V2	SQL Injection	Secure Input Handling (C3) Secure Database interactions (C5) Web Application Firewall(C9) Automated Vulnerability Detection(C1)	A05: Injection
V3	CSRF	Session & Request Integrity (C8) Bots Prevention(C19)	A01: Broken Access Control
V4	Input Validation Issues	Secure Input Handling (C3) Secure Database Interaction (C5)	A05: Injection A10: Mishandling of Exceptional Conditions
V5	Auth & Auth weaknesses	Access Control Handling (C13)	A07: Authentication Failures
V6	Web Server & Security Misconfiguration	Browser-side Security Controls (C2) XML parser hardening, framework configs(C16)	A02: Security Misconfiguration
V7	Source Code & Info Disclosure	Vulnerability Disclosure Practices (C18)	A01: Broken Access Control A02: Security Misconfiguration A06: Insecure Design A04: Cryptographic Failures A10: Mishandling of Exceptional Conditions
V8	File Upload	Secure Input Handling (C3) Web Application Firewall (C9) Access Control Handling (C13)	A02: Security Misconfiguration A05: Injection A01: Broken Access Control
V9	Path Traverse	Secure Input Handling (C3) Access Control Handling (C13) Web Application Firewall(C9)	A05: Injection A01: Broken Access Control
V10	Unvalidated Redirects and Forwards		
V11	Domain Spoofing	Email Security (C11) Secure Encryption(C20) Access Control Handling (C13)	A07: Authentication Failures A04: Cryptographic Failures

VD-based Vulnerabilities were mapped to countermeasures and the OWASP top ten vulnerabilities, The mapping process was conducted based on research papers reporting and the OWASP top ten vulnerabilities descriptions. As a result, the new OWASP top ten vulnerabilities categories were not mapped to the studies-reviewed vulnerabilities namely A03:Software Supply Chain Failures, A08:Software or Data Integrity Failures and A09: Security Logging and Alerting Failures due to limited representation of discussed vulnerabilities in the reviewed studies, revealing a systematic research gap between industry and empirically studied vulnerabilities. On the other hand, many cross-cutting countermeasures were recommended in research that can be mapped to all vulnerabilities rather than specific vulnerabilities. Therefore, several recommended countermeasures were intentionally excluded namely organizational, people level and cross-cutting countermeasures to avoid artificial

inflation of vulnerabilities and countermeasures alignment.

Based on the mapped vulnerabilities to the OWASP top ten vulnerabilities, vulnerabilities alignments were identified using the Vulnerability Density of each vulnerability in research. Table 8 shows the vulnerabilities alignment with the OWASP top ten list of vulnerabilities.

Table 8. The OWASP-Aligned Vulnerabilities Based on VD

Seq	OWASP List / VD Category	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
V1	XSS	0	0	0	0	0.56	0	0	0	0	0
V2	SQL Injection	0	0	0	0	0.19	0	0	0	0	0
V3	CSRF	0.16	0	0	0	0	0	0.16	0	0	0
V4	Input Validation Issues.	0	0	0	0	0.13	0	0	0	0	0.13
V5	Auth & Auth weaknesses	0.13	0	0	0	0	0	0	0	0	0
V6	Web Server & Security Misconfiguration	0	0.13	0	0	0	0	0	0	0	0
V7	Source Code & Info Disclosure	0.09	0.09	0	0.09	0	0.09	0	0	0	0.09
V8	File Upload	0.03	0.03	0	0	0.03	0	0	0	0	0
V9	Path Traverse	0.03	0	0	0	0.03	0	0	0	0	0
V10	Unvalidated Redirects and Forwards	0.03	0	0	0	0.03	0	0	0	0	0
V11	Domain Spoofing	0	0	0	0.03	0	0	0.03	0	0	0

Figure 5 illustrates the correspondence between the literature's vulnerabilities and the OWASP top ten 2025 categories. Darker cells represent the frequency of vulnerabilities in research while lighter cells indicate no correspondence. This visualization highlights both alignment and gaps between research and OWASP project. While A05: Injection vulnerability received a high research density in literature, A01: broken Access Control and A02: Security Misconfiguration were underrepresented in the literature, reinforcing the need for research attention shifts towards these gaps.

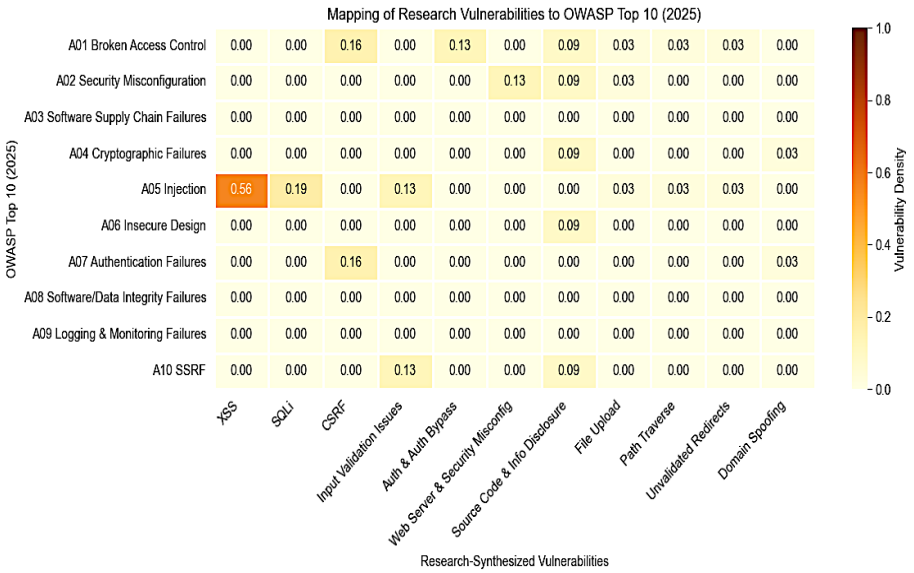

Figure 5. OWASP-Aligned Vulnerabilities Heat Map

To align countermeasures recommended in research, the Countermeasure Recommendations Density were assigned in the corresponding table cell based on the mapped countermeasure to the primary corresponding OWASP category only in Table 8.

Table 9 illustrates the literature-recommended countermeasure alignment with the OWASP top ten 2025 categories.

Table 9. The OWASP-Aligned Countermeasures Based on CRD

Seq	OWASP List / CRD Category	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
C2	Browser-side Security Controls	0	0	0	0	0.59	0	0	0	0	0
C3	Secure Input Handling	0	0	0	0	0.56	0	0	0	0	0.56
C5	Secure Database Interaction	0	0	0	0	0.44	0	0	0	0	0.44
C8	Session & Request Integrity	0.28	0	0	0	0	0	0.28	0	0	0
C11	Secure Encryption	0	0	0	0.22	0	0	0.22	0	0	0
C13	Access Control Handling	0.19	0.19	0	0.19	0.19	0	0.19	0	0	0
C14	Safe APIs	0	0	0	0	0.19	0	0	0	0	0
C16	XML parser hardening, framework configs	0	0.16	0	0	0	0	0	0	0	0
C19	Bots Prevention	0.13	0	0	0	0	0	0	0	0	0
C20	Email Security	0	0	0	0.09	0	0	0.09	0	0	0

Academic research prioritizes injection vulnerability prevention countermeasures while broken access control vulnerabilities were less encountered in research. Figure 6 shows the heat map of OWASP-aligned countermeasures, revealing the gap between research and industry.

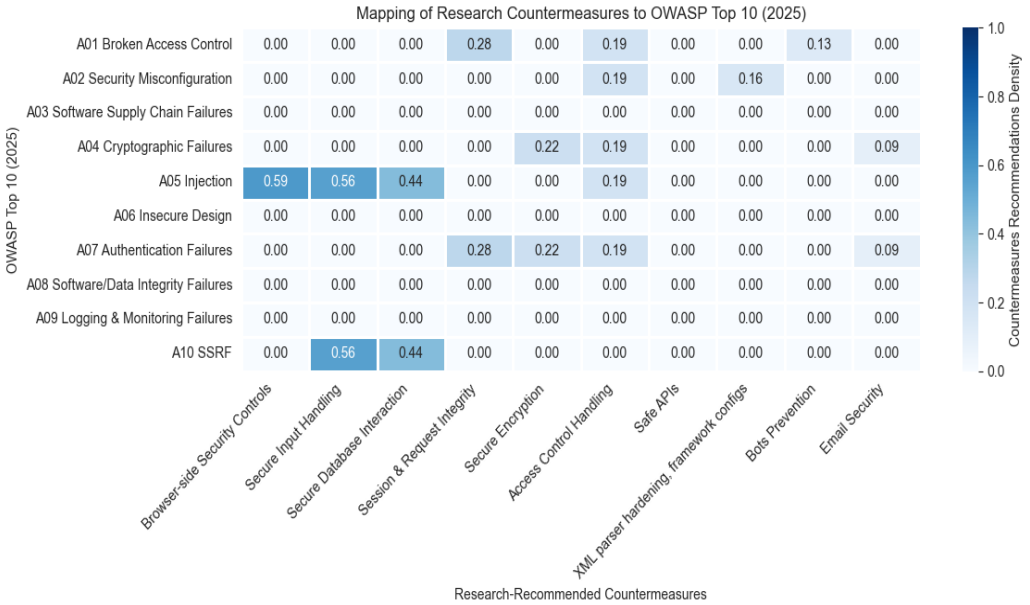

Figure 6. The OWASP-Aligned Countermeasures Heat Map

The OWASP top ten 2025 categories alignment with research-reviewed vulnerabilities and countermeasures highlights a significant attention of research paid on injection attacks (A05) vulnerabilities and countermeasures, while A01, A02, A03 and A04 remain understudied, indicating misalignment between industry-centric reports and academic research.

This study next synthesizes web application vulnerabilities and countermeasures alignment with the rea world cyberattack incidents.

4.2 RQ2: What are the vulnerabilities seized and the countermeasures applied against cyberattacks by the impacted organizations and companies?

A total of 34 real-world severe cyberattack incidents were involved in this study; fourteen out of thirty-four incidents occurred between 2020 and 2024. Attackers exploited various vulnerabilities to breach a system. Excluding four cross-cutting vulnerabilities such as human, organizational factors and two unidentified exploited Zero-Day vulnerabilities [54], [55], thirty-four vulnerabilities were exploited, successfully bypassing all security countermeasures and breaching victims' information systems. To avoid vulnerabilities overlapping as well as to avoid exploding frequencies of mentions, the exploited vulnerabilities were categorized based on the OWASP 2025 list of top ten vulnerabilities categorization. Also, the Real-World Vulnerability Density (RWD) is accordingly computed for every normalized vulnerability via an equation, as shown in (3).

$$\text{RWD} = \frac{Number\ of\ Case\ Evident\ Vulnerabities\ (N)}{Total\ Number\ of\ Incidents\ (T)} \qquad (3)$$

With the above formula applied to every categorized exploited vulnerability, OWASP-categorized vulnerabilities, corresponding with their RWD metrics, are introduced in Table 10.

Table 10. The Vulnerabilities Exploited in Real-World Cyberattack Incidents

Seq	Vulnerability	OWASP-Based Category	Incident(s) N	RWD	Ref
RWD2	lack of comprehensive effective patching	Software Supply Chain Failures	9	0.26	[56],[57],[58] [59], [60],[61],[54] ,[62]
RWD1	Use of Stolen Credential	Authentication Failures	7	0.41	[63],[64],[65], [66], [67]
	Password Reuse across all systems		3		[55]
	Lack MFA (Brute force)		2		[67]
	Insufficient monitoring of privileged accounts		2		[56], [61]
RWD3	Misconfigured Database (No Password required)	Security Misconfiguration	2	0.15	[63], [68]
	Weak Password		2		[69], [70]
	Misconfigured Web Application Firewall (WAF)		1		[21]
RWD4	SQL Injection Vulnerability	Injection	2	0.09	[54]
	URL injection		1		[72]
RWD5	Encryption Failure	Cryptographic Failures	2	0.06	[61]
RWD6	File Upload broken access control	Insecure Design	1	0.03	[73]

Although most attacks were initially bypassed by network security measures, web application security should have defended the system from the breach. The vulnerabilities of third-party components were the most exploited vulnerabilities by attackers, which has been reported in eight incidents. The incidence of stolen credentials has been increasingly reported recently. Stolen credentials are provided and sold in some cases to the public for free and effortlessly anyone can access a victim's account. Unlike technical web application vulnerabilities, there have been non-technical vulnerabilities where human errors have unintentionally played a major role in insecure systems' designs or misconfigured security tools. Human errors have been reported in many industrial reports and incidents, and 93% of all reported cyberattack incidents were associated with human errors [59], [67], [68], [69], [74].

Given the synthesized vulnerabilities densities in real world cyberattack incidents, categorized based on OWASP 2025 top ten list of vulnerabilities categorization, vulnerabilities alignment between research attention towards vulnerability and real world cyberattack incidents was conducted to identify gaps and key milestones. Using Real World Vulnerability Density (RWD) metric and its corresponding research VD metric that its value can be

retrieved from Table5, a gap in research pattern towards web application vulnerabilities for the period from 2014 to 2024 is computed, depicted in Equation (4).

$$\text{Gap}_\text{i} = \text{RWD}_\text{i} - VD_j \quad (4)$$

Based on a result of the equation Gap_i, a web application vulnerability is either overstudied or understudied. Subtracting RWD from VD where i=1 and j=5 for the authentication and authorization vulnerability produces as follows: Gap_1=0.41-0.13, Gap=0.28

Given a positive result for the parameter $\text{Gap}_1 = 0.28$ depicts that the research understudied the authentication and authorization vulnerability, reinforcing the need to align the research focus to the overlooked vulnerability. Table 11 demonstrates the research gaps towards web application vulnerabilities. Research gaps have been identified in the pattern-driven research toward web application vulnerabilities. After the synthesized vulnerabilities were categorized based on OWASP reported vulnerabilities categorization, significant gaps have been identified in research pattern towards vulnerabilities. Research pattern alignment was categorized into three alignment categories, as follows:

1. Aligned: This category represents in vulnerabilities reviewed that are aligned with vulnerabilities exploited in the reviewed real world cyberattack incidents.
2. Understudied: This category represents vulnerabilities research priority that are not aligned with real world exploited vulnerabilities priority and that shifts in a research concentration should be applied.
3. Overstudied: his category portrays research overstudied a vulnerability while attackers are deploying other vulnerabilities to breach information systems.

Table 11. Gaps in Research Pattern Trend Towards Web Application Vulnerabilities

Research Vulnerability	**VD**	**OWASP Category**	**RWD**	**GAP**	**Research Pattern Alignment category**
XSS	0.56	Injection	0.09	-0.46	Overstudied
SQL Injection	0.19			-0.1	Overstudied
Path Traverse	0.03			0.06	Aligned
Unvalidated Redirects and Forwards	0.03			0.06	Aligned
Input Validation Issues	0.13			-0.04	Aligned
CSRF	0.16	Authentication Failures	0.41	0.25	Understudied
Authentication and Authorization weaknesses	0.13			0.28	Understudied
Domain Spoofing	0.03			0.38	Understudied
Web Server and Security Misconfiguration	0.13	Security Misconfiguration	0.15	0.02	Aligned
Source Code and Information Disclosure	0.09			0.06	Aligned
(New Category)	-	Software Supply Chain Failures	0.26	-	Understudied
File Upload Vulnerability	0.03	Insecure Design	0.03	0	Aligned
(Not Reported)	-	Cryptographic Failures	0.06	0.06	Understudied

As a result of the vulnerabilities categorization process, injection cyberattacks have received sufficient research attention, with overstudied classification for XSS and SQL injection attacks, followed by two common vulnerabilities namely Security Misconfiguration and Insecure Design vulnerabilities. However, significant gaps in web application vulnerability research were observed in which Authentication and Authorization Failures, Software Supply Chain Failures and Cryptographic Failures vulnerabilities were understudied in the reviewed studies in comparison to the real world exploited vulnerabilities pattern, reinforcing a need to align research focus toward the understudied vulnerabilities. The Software Supply Chain Failures vulnerability is relatively new OWASP top ten category and despite it potentially high impact, fewer exploited cases are publicly reported, revealing its limited representation in real world cyberattack incidents.

Countermeasures, on the other hand, were taken against the reported incidents varied in such a way that different measures were taken at different times throughout the incident. A list of categorized countermeasures was created

to represent the detailed countermeasures taken by the impacted organizations and companies, as depicted in Table 12.

Table 12. Deployed Security Countermeasures Against Cyberattack Incidents

	Countermeasure	Mapped to OWASP Top Ten Vulnerabilities	Freq
1	Immediate third-party components patching	A02 – Security Misconfiguration	18
2	Multifactor authentication enforcement	A07 – Authentication Failures	16
3	Security logs real-time monitoring	A09 – Logging & Monitoring Failures	13
4	Users' accounts resets	A07 – Authentication Failures	12
5	Web server ports restriction	A02 – Security Misconfiguration	11
6	Vulnerabilities Management	A06 – Insecure Design	10
7	Data encryption enhancement	A04 – Cryptographic Failures	9
8	WAF deployment	A05 – Injection	7
9	Privileged access management	A01 – Broken Access Control	7
10	Third-party contractors and admins remote access restriction	A01 – Broken Access Control	7
11	Secure web services configuration	A02 – Security Misconfiguration	6
12	Public Untrusted libraries removal	A03 – Software Supply Chain Failures	6
13	Best practices secure SDLC implementation	A03 – Software Supply Chain Failures	5
14	Threat Intelligence deployment	A09 – Logging & Monitoring Failures	5

As a result of reviewing the security countermeasure impacted organizations deployed, third party components and services were the most exploited vulnerability by attackers, emphasizing the results of the understudied vulnerability gap found in research for a period from 2014 to 2024. Therefore, eighteen impacted organizations out of thirty-four incidents have patched their third-party components and services. While injection countermeasures (C2, C3 and C5) were prioritized by researchers to prevent injection attacks, seven organizations out of thirty-four organizations reported injection attacks countermeasures, relying on WAFs. On the other hand, authentication and authorization enforcement, including MFA authentication and Users' accounts Resets countermeasures were widely deployed across the impacted organizations and companies.

With the given research gaps derived from synthesizing vulnerabilities and countermeasures in research as well as in real world cyberattack incidents, research attention alignment will enhance web application security vastly by developing security protection mechanisms and frameworks. This study next summarizes the root cause of unmitigated vulnerability delay

4.3 RQ3: What are the factors that play a main role in delaying web application vulnerability mitigations?

There were many unmitigated vulnerabilities when the cyberattacks occurred. One of the objectives of this study is to determine the root cause of unmitigated vulnerabilities reported in either WL or GL documents. As a result, many factors, either technical or non-technical, contributed unintentionally to the incident. The list of factors is summarized as follows:

- Development of improper practices
- Patch management delay due to resources constraints.
- Reliance on third-party software.
- Human awareness and social engineering
- Outdated system deployment
- Vulnerabilities' mitigation prioritization

In vulnerability management tasks, prioritizing the mitigation of vulnerability may reduce the risk of vulnerability

exploitation, as recommended by the Crowdstrike company [75]. Crowdstrike reported that attackers have been heavily targeting identities to access IT systems using social engineering factors, and it is recommended that multifactor authentication solutions must be deployed. Social Engineering is a non-technical vulnerability, but to build a detailed profile for each user, adversaries have utilized AI, specifically an LLM, to gather information from social media [76]. The LLM model synthesizes data and builds the victims' interests and personal activities, which the attackers would seize to be trusted by the targeted system. Factors have recently gained the attention of industries, owing to the rise in their number.

However, the list of factors consists of the responsibilities that every organization or company must consider securing its information systems. The list of causes of unmitigated vulnerability persistence can be categorized into three levels: organizational, management, and development. At the organizational level, security policies, organizational culture, and resource allocation should be addressed for bottom-level management to maintain the best security practices in place. At the management level, technical decisions must be made to ensure that the development teams are properly aligned with security policies. At the development level, developers are responsible for applying the most secure development practices during the development or maintenance of applications.

5. CONCLUSION

Web-application vulnerability is an easy entry point for adversaries to access long-maintained data for assorted reasons. The dependence on supply chains to build IT systems has considerably raised the challenge of securing information systems owing to the complexity of cyberspace [77]. Unfortunately, less-resourced organizations or companies are vulnerable to sophisticated cyberattacks, owing to the prevalence of enhanced AI attacks.

Web application vulnerabilities and countermeasures alignment with industry-centric reviews and with technical details reported from real-world cyberattack incidents revealed that research gaps in information system security enhancement. This study found that injection vulnerabilities and countermeasures were overstudied while Authentication Failures, Software Supply Chain Failures and Cryptographic Failures vulnerabilities and their countermeasures were understudied. In addition, lack of proper security implementation knowledge, human awareness, social engineering, vulnerabilities mitigation prioritization and resource constraints are the factors behind vulnerabilities mitigation delay.

This research study reviewed real-world cyberattack incidents as well as recent research on the web application security aspect to provide future research and industries with the most recently deployed cyberattack tactics and exploited vulnerabilities. In addition, this research tackled the challenges of mitigating vulnerabilities detected in information systems and listed the root causes of delayed vulnerabilities mitigation in information systems. This study contributes to the information system security with the CS-SLR approach that integrates the research reviews towards web application security with real-world cyberattack incidents, highlighting vulnerability and countermeasures research pattern misalignment. Moreover, the study reveals the root cause of unmitigated vulnerabilities with web applications, providing experts with critical insights into vulnerability exploitation. This paper emphasizes the need for further research in the field of web application security and recommends the adoption of the CS-SLR methodology to achieve better results.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for the suggestions to improve the paper.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Salaheddin Beskri: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;
Kok Why Ng: Supervision, Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics followed The Committee of Publication Ethics (COPE) guidelines. https://publicationethics.org/

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] J. Mackeen and H. Smith, *It Strategy: Issues and Practices*, Third. New Jersey: Pearson Education, 2015.

[2] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Reports*, vol. 7, pp. 8176–8186, Nov. 2021, doi: 10.1016/j.egyr.2021.08.126.

[3] A. I. Mallick and R. Nath, "Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments," *International Scientific Journal*, pp. 1–69, 2024, [Online]. Available: www.worldscientificnews.com

[4] Center for Strategic and International Studies (CSIS), "By USA -This timeline records significant cyber incidents since 2006 with losses more than a million dollars," Washington, D.C., 2024. Accessed: Dec. 21, 2024. [Online]. Available: https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents

[5] D. Kaur and P. Kaur, "Empirical analysis of web attacks," in *Physics Procedia*, Elsevier B.V., pp. 298–306, 2016, doi: 10.1016/j.procs.2016.02.057.

[6] OWASP Foundation, "Open web application security project (Top ten web application vulnerabilities)." Accessed: Dec. 28, 2025. [Online]. Available: https://owasp.org/Top10/2025/

[7] J. Li and H. Li, "Evolution of application security based on OWASP top 10 and CWE/SANS Top 25 with predictions for the 2025 OWASP Top 10," in *Proceedings of 8th International Conference on Inventive Computation Technologies, ICICT 2025*, Institute of Electrical and Electronics Engineers Inc., pp. 1178–1183, 2025, doi: 10.1109/ICICT64420.2025.11004742.

[8] OWASP Foundation, "Web hacking incidents database program," https://owasp.org/www-project-web-hacking-incident-database/.

[9] Common Vulnerabilities Exposure program (CVE), "Common Vulnerabilities Exposure Program (CVE) database," https://www.cve.org/Downloads.

[10] A. A. Alobaidi and N. B. Al Dabbagh, "Web attacks and defenses," *Journal of Education and Science*, vol. 32, no. 2, pp. 91–100, Jun. 2023, doi: 10.33899/edusj.2023.137855.1319.

[11] B. Riskhan, M. A. Ullah Sheikh, M. S. Hossain, K. Hussain, Z. Zainol, and N. Z. Jhanjh, "Major vulnerabilities of web application in real world scenarios and their prevention," in *ICoICC 2025 - 3rd International Conference on Intelligent and Cloud Computing*, Institute of Electrical and Electronics Engineers Inc., 2025, doi: 10.1109/ICoICC64033.2025.11052016.

[12] S. Phanireddy, "Securing modern web applications using AI-driven static and dynamic analysis techniques," *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 6, pp. 73–82, 2025, doi: 10.63282/3050-9262.ijaidsml-v6i2p108.

[13] F. Jimmy, "Assessing the effects of cyber attacks on financial markets," *Journal of Artificial Intelligence General Science*, 2024, doi: 10.60087.

[14] A. Hannousse, S. Yahiouche, and M. C. Nait-Hamoud, "Twenty-two years since revealing cross-site scripting attacks: A systematic mapping and a comprehensive survey," *Elsevier Ireland Ltd*, May 01, 2024, doi: 10.1016/j.cosrev.2024.100634.

[15] M. A. Almaiah, L. M. Saqr, L. A. Al-Rawwash, L. A. Altellawi, R. Al-Ali, and O. Almomani, "Classification of cybersecurity threats, vulnerabilities and countermeasures in database systems," *Computers, Materials and Continua*, vol. 81, no. 2, pp. 3189–3220, 2024, doi: 10.32604/cmc.2024.057673.

[16] R. A. Khan, S. U. Khan, M. A. Akbar, and M. Alzahrani, "Security risks of global software development life cycle: Industry practitioner's perspective," *Journal of Software: Evolution and Process WILEY*, vol. 36, no. 3, Mar. 2024, doi: 10.1002/smr.2521.

[17] R. A. Khan, S. U. Khan, H. U. Khan, and M. Ilyas, "Systematic literature review on security risks and its practices in secure software development," *Institute of Electrical and Electronics Engineers Inc.*, 2022, doi: 10.1109/ACCESS.2022.3140181.

[18] M. F. Sohan and A. Basalamah, "A systematic literature review and quality analysis of javascript malware detection," *Institute of Electrical and Electronics Engineers Inc.*, 2020, doi: 10.1109/ACCESS.2020.3031690.

[19] T. Rathod, N. K. Jadav, S. Tanwar, A. Alabdulatif, D. Garg, and A. Singh, "A comprehensive survey on social engineering attacks, countermeasures, case study, and research challenges," *Information Processing & Management*, vol. 62, no. 1, Jan. 2025, doi: 10.1016/j.ipm.2024.103928.

[20] S. Temara, "The ransomware epidemic: Recent cybersecurity incidents demystified," *Asian Journal of Advanced Research and Reports*, vol. 18, no. 3, pp. 1–16, Feb. 2024, doi: 10.9734/ajarr/2024/v18i3610.

[21] S. Khan, I. Kabanov, Y. Hua, and S. Madnick, "A systematic analysis of the capital one data breach: Critical lessons learned," *ACM Transactions on Privacy and Security*, vol. 26, no. 1, Nov. 2022, doi: 10.1145/3546068.

[22] X. Zhao, T. Clear, and R. Lal, "Identifying the primary dimensions of DevSecOps: A multi-vocal literature review," *Journal of Systems and Software*, vol. 214, p. 112063, 2024, doi: 10.5281/zenodo.7.

[23] J. Schopfel, "Towards a Prague definition of grey literature current definition."

[24] N. S. Harzevili *et al.*, "A systematic literature review on automated software vulnerability detection using machine learning," *ACM Computing Surveys*, vol. 57, no. 3, Nov. 2024, doi: 10.1145/3699711.

[25] J. Svacina *et al.*, "On Vulnerability and Security Log analysis: A systematic literature review on recent trends," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 175–180, Oct. 2020, doi: 10.1145/3400286.3418261.

[26] R. Lin *et al.*, "Vulnerabilities and security patches detection in OSS: A survey," *ACM Comput Surv*, Jan. 2024, doi: 10.1145/3694782.

[27] K. Rahman and C. Izurieta, "A mapping study of security vulnerability detection approaches for web applications," in *Proceedings - 48th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2022*, Institute of Electrical and Electronics Engineers Inc., pp. 491–494, 2022, doi: 10.1109/SEAA56994.2022.00081.

[28] N. Tewari and G. Datt, "A study on the systematic review of security vulnerabilities of popular web browsers," in *Proceedings of International Conference on Technological Advancements and Innovations, ICTAI 2021*, Institute of Electrical and Electronics Engineers Inc., pp. 314–318, 2021, doi: 10.1109/ICTAI53825.2021.9673463.

[29] S. Alazmi and D. C. De Leon, "A systematic literature review on the characteristics and effectiveness of web application vulnerability scanners," *Institute of Electrical and Electronics Engineers Inc.*, 2022, doi: 10.1109/ACCESS.2022.3161522.

[30] M. Kaniaki, J. Dobaa, and D. Kermek, "Deep learning within the web application security scope-literature review."

[31] A. De Jesus Dominguez-Garcia, X. Limon, J. O. Ocharan-Hernandez, and J. C. Perez-Arriaga, "Security testing for web applications: A systematic literature review," in *Proceedings - 2023 11th International Conference in Software Engineering Research and Innovation, CONISOFT 2023*, Institute of Electrical and Electronics Engineers Inc., pp. 82–91, 2023, doi: 10.1109/CONISOFT58849.2023.00020.

[32] S. Carlos, M. Hugo, and A. Myriam, *The evolution from traditional to intelligent web security: Systematic literature review*. IEEE, 2020.

[33] T. Y. Khaw, A. Amran, and A. P. Teoh, "Building a thematic framework of cybersecurity: a systematic literature review approach,", *Emerald Publishing*, May 07, 2024, doi: 10.1108/JSIT-07-2023-0132.

[34] M. M. Hassan, B. R. Ahmad, A. Esha, R. Risha, and M. S. Hasan, "Important factors to remember when constructing a cross-site scripting prevention mechanism," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 2, pp. 965–973, Apr. 2022, doi: 10.11591/eei.v11i2.3557.

[35] G. Rodriguez-Galan and J. Torres, "Personal data filtering: A systematic literature review comparing the effectiveness of XSS attacks in web applications vs cookie stealing," *Annales des Telecommunications/Annals of Telecommunications*, Dec. 2024, doi: 10.1007/s12243-024-01022-8.

[36] F. Heiding, S. Katsikeas, and R. Lagerstrom, "Research communities in cyber security vulnerability assessments: A comprehensive literature review," *Elsevier Ireland Ltd*, May 01, 2023, doi: 10.1016/j.cosrev.2023.100551.

[37] M. Aydos, C. Aldan, E. Coskun, and A. Soydan, "Security testing of web applications: A systematic mapping of the literature," *King Saud bin Abdulaziz University*, Oct. 01, 2022, doi: 10.1016/j.jksuci.2021.09.018.

[38] C. N. Siahaan, M. Rufisanto, R. Nolasco, S. Achmad, and C. R. P. Siahaan, "Study of cross-site request forgery on web-based application: Exploitations and preventions," in *Procedia Computer Science*, Elsevier B.V., pp. 92–100, 2023, doi: 10.1016/j.procs.2023.10.506.

[39] A. Marchand-Melsom and D. B. Nguyen Mai, "Automatic repair of OWASP Top 10 security vulnerabilities: A survey," in *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*, Association for Computing Machinery, Inc, pp. 23–30, Jun. 2020, doi: 10.1145/3387940.3392200.

[40] E. A. Altulaihan, A. Alismail, and M. Frikha, "A survey on web application penetration testing,", *MDPI*, Mar. 01, 2023, doi: 10.3390/electronics12051229.

[41] G. E. Rodriguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-site scripting (XSS) attacks and mitigation: A survey," *Computer Networks*, vol. 166, Jan. 2020, doi: 10.1016/j.comnet.2019.106960.

[42] D. Das, N. S. Mathews, and S. Chimalakonda, "Exploring security vulnerabilities in competitive programming: an empirical study," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 110–119, Jun. 2022, doi: 10.1145/3530019.3530031.

[43] J. R. Henriques, J. D'Abruzzo Pereira, and M. Vieira, "Mining vulnerability and code repositories to study software security," in *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, New York, NY, USA: ACM, pp. 11–16, Nov. 2024, doi: 10.1145/3697090.3697103.

[44] A. Ikegami *et al.*, "On the use of refactoring in security vulnerability fixes: An exploratory study on maven libraries," 2022, doi: 10.1145/353001.

[45] M. F. Safitra, M. Lubis, and A. Widjajarto, "Security vulnerability analysis using Penetration Testing Execution Standard (PTES): Case study of government's website," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, pp. 139–145, Mar. 2023, doi: 10.1145/3592307.3592329.

[46] M. Z. Zakaria and R. Kadir, "Risk assessment of web application penetration testing on Cross-Site Request Forgery (CSRF) attacks and Server-Side Includes (SSI) injections," in *2021 International Conference on Data Science and Its Applications, ICoDSA 2021*, Institute of Electrical and Electronics Engineers Inc., pp. 85–90, 2021, doi: 10.1109/ICoDSA53588.2021.9617554.

[47] R. A. Correa, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, M. S. Rubio, and Alberto Magrenan, "Hybrid security assessment methodology for web applications," *CMES - Computer Modeling in Engineering and Sciences*, vol. 126, no. 1, pp. 89–124, 2021, doi: 10.32604/CMES.2021.010700.

[48] M. A. K. Rifat, Y. Sultana, and B. M. Mainul Hossain, "Vulnerabilities assessment of financial and government websites: A developing country perspective," *International Journal of Information Engineering and Electronic Business*, vol. 15, no. 5, pp. 42–53, Oct. 2023, doi: 10.5815/ijieeb.2023.05.05.

[49] L. Gallo, D. Gentile, S. Ruggiero, A. Botta, and G. Ventre, "The human factor in phishing: Collecting and analyzing user behavior when reading emails," *Computers Security*, vol. 139, Apr. 2024, doi: 10.1016/j.cose.2023.103671.

[50] B. Hulloowan and G. Bekaroo, "Defending against XML External Entity (XXE) attacks: A review and comparative analysis of prevention mechanisms," in *4th International Conference on Next Generation Computing Applications, NextComp 2024 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2024, doi: 10.1109/NextComp63004.2024.10779957.

[51] A. Aborujilah, J. Adamu, S. M. Shariff, and Z. A. Long, "Descriptive analysis of built-in security features in web development frameworks," in *Proceedings of the 2022 16th International Conference on Ubiquitous Information Management and Communication, IMCOM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, doi: 10.1109/IMCOM53663.2022.9721750.

[52] C. A. S. Murty, H. Rana, R. Verma, R. Pathak, and P. H. Rughani, "A review of web application security risks: Auditing and assessment of the dark web," in *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2021*, Institute of Electrical and Electronics Engineers Inc., Oct. 2021, doi: 10.1109/ICECCME52200.2021.9591031.

[53] K. Kiashemshaki, M. J. Torkamani, and N. Mahmoudi, "Secure coding for web applications: Frameworks, challenges, and the role of LLMs," Sep. 2025, [Online]. Available: http://arxiv.org/abs/2507.22223

[54] G. Ali, M. M. Mijwil, B. A. Buruga, and M. Abotaleb, "A comprehensive review on cybersecurity issues and their mitigation measures in FinTech," *College of Education, Al-Iraqia University*, 2024, doi: 10.52866/ijcsm.2024.05.03.004.

[55] The Academy of ICT Essentials for Government Leaders, "Information security and privacy," 2021.

[56] "Conti cyber attack on the HSE Independent Post Incident Review Commissioned by the HSE Board in conjunction with the CEO and Executive Management Team," Dec. 2021.

[57] M. Aljaidi, "A comprehensive technical analysis of URL redirect attacks: A case study of British Airways data breach," in *2023 24th International Arab Conference on Information Technology, ACIT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, doi: 10.1109/ACIT58888.2023.10453784.

[58] P. Rob and C. Tom, "How Equifax neglected cybersecurity and suffered a devastating data breach staff report permanent subcommittee on investigations United States senate," 2018.

[59] N. Manworren, J. Letwat, and O. Daily, "Why you should care about the Target data breach," *Elsevier Ltd*, May 01, 2016, doi: 10.1016/j.bushor.2016.01.002.

[60] S. R. Mugu, B. Zhang, H. Kolla, S. R. A. Balaji, and P. Ranganathan, "Lessons from the CrowdStrike Incident: Assessing endpoint security vulnerabilities and implications," in *2024 Cyber Awareness and Research Symposium (CARS)*, IEEE, pp. 1–10, Oct. 2024, doi: 10.1109/CARS61786.2024.10778784.

[61] R. Denuwan, "Marriott international data breach," *Research Gate*, 2023, [Online]. Available: https://www.researchgate.net/publication/372524901

[62] "Leadership for it security & privacy across HHS. HHS. Cybersecurity program office of information security," 2022.

[63] J. Sorn, P. Carroll, of Pang, S. Bhunia, M. Salman, and P. A. Regis, "Exploring the CAM4 Data Breach: Security Vulnerabilities and Response Strategies," in *Proceedings - 2024 IEEE/ACM 24th International Symposium on Cluster, Cloud and Internet Computing Workshops, CCGridW 2024*, Institute of Electrical and Electronics Engineers Inc., pp. 174–179, 2024, doi: 10.1109/CCGridW63211.2024.00028.

[64] "SWIFT systems and the SWIFT customer security program," Baku, Azerbaijan, 2021.

[65] PowerSchool company, "Notice of power school data breach for individuals in the united states," Incident Notice.

[66] Van der Meulen, "Directorate general for internal policies policy department c: citizens' rights and constitutional affairs cybersecurity in the European Union and beyond: Exploring the threats and policy responses STUDY," 2015.

[67] Global Privacy Assembly (GPA), "International enforcement cooperation working group," 2021.

[68] M. Shcherbakov, "Code-reuse attacks in managed programming languages and runtimes," KTH Royal Institute of Technology, Stockholm, 2024.

[69] S. Mansfield-Devine, "The Ashley Madison affair," *Elsevier*, no. 9, pp. 8–16, Sep. 2015, [Online]. Available: http://tools.ietf.org/html/

[70] "Line between cyberthreats and physical impact continues to blur," 2023.

[71] M. Jay and L. Coli, "Cybersecurity and deposit insurance: An introduction," Nov. 2024. [Online]. Available: https://www.fsb.org/wp-content/uploads/P130423-3.pdf

[72] A. L. Katherine, S. H. Justin, and A. F. Elizabeth, "Second amended statement of charges and notice of hearing in the matter of first American title insurance company," New York, Jun. 2021.

[73] "Cybersecurity trends & predictions 2024," Mar. 2024.

[74] S. Waelchli and Y. Walter, "Reducing the risk of social engineering attacks using SOAR measures in a real world environment: A case study," *Computers & Security*, vol. 148, Jan. 2025, doi: 10.1016/j.cose.2024.104137.

[75] CrowdStrike Inc, "Crowdstrike 2025 global threat report 2," 2025.

[76] Fortinet, "Cyberthreat predictions for 2025 an annual perspective from FortiGuard Labs REPORT," 2025.

[77] CrowdStrike Inc, "Global cybersecurity outlook 2025," Jan. 2025.

BIOGRAPHIES OF AUTHORS

Salaheddin Beskri is a PhD student in Web Application Cybersecurity at the Faculty of Computing and Informatics (FCI) of Multimedia University (MMU), Malaysia and a senior lecturer at the Tripoli Collage for Sciences and Technology (TCST), Libya. His research focuses on securing information systems. He has experience in securing software development, Deep Learning–based threat detection and applied research on improving the security of modern web infrastructures. He is also involved in developing practical security frameworks and tools for real-world deployment. He can be contacted at eng.salaheddin@gmail.com

Kok Why Ng is an Associate Professor in the Faculty of Computing and Informatics (FCI) in Multimedia University (MMU), Malaysia. He did his B.Sc. (Math) in USM, Penang, and his M.Sc (IT) and Ph.D (IT) in MMU, Malaysia. His research interests are in Recommender System, 3D Geometric Modeling and Animation. He is also active in some research projects related to artificial intelligence, deep learning and human blood cells. He can be contacted at kwng@mmu.edu.my. Salaheddin Beskri,