STAP
Smart Technologies Academic Press

Journal of Cyber Security and Risk Auditing

https://www.jcsra.thestap.com/

Check for updates

Analytical Analysis of Cyber Threats and Defense Mechanisms for Web Application Security

Bashaer Almelehy[1], Mohammad Ahmad [2], Ghalia Nassreddine [3], Mohammed Maayah[4], Aparna Achanta[5]

[1]*Department of Computer Networks and Communications, King Faisal University, Al-Ahsa 31982, Saudi Arabia*
[2] *Institute for Research in Applicable Computing, University of Bedfordshire, UK*
[3] *Dept. of Computer and Information Systems, Rafik Hariri University, Meshref, Lebanon*
[4]*Fellowship Researcher, INTI International University, Nilai 71800, Malaysia*
[5] *Principal Security Architect, IBM, USA*

ARTICLE INFO

Article History

Received: 01-05-2025
Revised: 30-06-2025
Accepted: 01-05-2025
Published: 02-07-2025

Academic Editor:

Prof. Youakim Badr

Vol.2025, No.3

DOI:
https://doi.org/10.63180/jcsra.thestap.2025.3.4

ABSTRACT

The use of internet technologies offers numerous advantages and has significantly transformed our daily lives, becoming a primary means of communication. Additionally, many businesses have shifted their services to digital platforms by leveraging web application technologies. As a result, vast amounts of data are exchanged between users and web applications—much of which contains sensitive and critical information. This makes them prime targets for cyber-attacks, including data theft and the unauthorized disclosure of confidential information. According to the Open Web Application Security Project (OWASP), there are ten major risks that pose significant threats to web applications. In response, this paper aims to provide a thorough understanding of web applications, the potential cyber threats they face, and a detailed review of existing literature related to cybersecurity risks in web applications. To achieve this, a comprehensive literature review will be conducted to identify the primary vulnerabilities in web applications and explore current methods for mitigating and preventing these security threats.

Keywords: Web Applications Cyber-Attacks, Open Web Application Security Project (OWASP), Security threats.

How to cite the article
Almelehy, B., Ahmad, M., Nassreddine, G., Maayah, M., & Achanta, A. (2025). Analytical Analysis of Cyber Threats and Defense Mechanisms for Web Application Security. Journal of Cyber Security and Risk Auditing, 2025(3), 57–76. https://doi.org/10.63180/jcsra.thestap.2025.3.4

1. Introduction

The use of internet technology offers numerous advantages and has significantly transformed our lives. The technological revolution has made the internet one of the most widely used means of communication worldwide. In today's digital era, many of our daily transactions rely heavily on internet-based applications [1]. Among these, web applications stand out as a powerful and essential tool. Web applications are defined as interactive websites powered by server-side programs that

*Corresponding author. Email: 221445338.student@kfu.edu.sa

enable user interaction and offer a variety of services [2]. These services include e-banking, e-governance, e-commerce, and many others [3].

As a result, organizations are becoming increasingly dependent on web applications, and the trend of online interaction through websites is expected to continue growing [4]. Developing a secure web platform is therefore a critical task and a vital aspect of any business operating an internet-connected portal. It ensures that clients can have a safe and trustworthy online experience [5]. Given the sensitivity and value of the data stored within these applications, businesses must prioritize the protection of their digital assets. Any vulnerabilities could potentially be exploited, thereby compromising core security principles such as confidentiality, integrity, and availability [6].

To enhance the security of web applications, various technical solutions have been introduced and implemented. One of the key security challenges in web applications is handling user input, which is often a primary source of vulnerabilities [7]. To mitigate this issue, multiple input-handling techniques have been developed, including sanitization, whitelisting, and blacklisting. Detailed explanations of these techniques will be provided in the following sections.

Understanding and identifying potential risks, along with their corresponding countermeasures, is essential to strengthen web application security. This study aims to review the major threats affecting web applications and analyze the primary countermeasures employed to mitigate these risks. Additionally, a real-world case study involving Gama Hospital will be examined in detail to explore the types of cyberattacks the institution encountered and how it effectively responded to them.

Web applications have become increasingly vital in the era of technological advancement and digital transformation [8]. The volume of data exchanged between users and these applications has grown substantially, making the protection of such digital assets a complex and pressing challenge. In recent years, various types of cyberattacks have emerged that exploit existing vulnerabilities, many of which remain inadequately addressed.

To effectively mitigate these threats, it is essential to gain a comprehensive understanding of potential attacks targeting web applications, along with adopting best practices from both academic research and industry standards. Rather than relying on temporary or reactive prevention strategies, identifying the root causes of vulnerabilities enables organizations to resolve critical issues more thoroughly and reduce the likelihood of recurring attacks.

As web portals often store and process sensitive information, businesses must prioritize safeguarding these assets by upholding the three fundamental principles of information security. Confidentiality ensures that data is accessible only to authorized individuals; availability guarantees that systems and data are accessible whenever needed; and integrity ensures that data remains accurate and unaltered during storage or transmission. Maintaining these principles is essential for preserving trust, functionality, and the resilience of web applications in today's interconnected environment.

Web applications have gained significant relevance in recent years due to the rapid pace of digital transformation and the wide range of services they facilitate. Today, most businesses worldwide rely on web technologies to digitize their services, aiming to reduce both operational costs and processing time [9]. These applications often store and manage critical information, making them prime targets for cybersecurity attacks that exploit vulnerabilities and flaws—many of which are either overlooked or insufficiently addressed by developers.

Given the sensitivity of the data stored and the essential nature of services provided through web applications, it is crucial to identify and assess the potential risks that may threaten business operations. This risk assessment must be accompanied by the implementation of robust controls and countermeasures to mitigate vulnerabilities and protect key security objectives—namely, confidentiality, availability, and integrity.

According to the Open Web Application Security Project (OWASP), there are ten major risks that continue to pose serious threats to web applications. Among these, injection attacks and broken access control have been ranked as top threats in the 2021 OWASP Top Ten list. Therefore, the aim of this project is to offer a comprehensive understanding of web applications, explore the nature of these cybersecurity threats, and conduct a detailed review of current research focused on web application vulnerabilities—particularly those related to injection and access control flaws.

Building a robust and secure website capable of protecting sensitive data and minimizing the risk of vulnerability exploitation is a fundamental goal for any organization. In recent years, web applications have been increasingly targeted due to various inherent flaws that attackers have successfully exploited. This study aims to identify the potential risks

threatening web applications by drawing insights from both academic literature and industry practices. It will provide an overview of common attack types, offering detailed explanations of each. To support this, a comprehensive literature review will be conducted to analyze existing studies related to cybersecurity threats in web applications. Additionally, the study will examine a real-world case study from the industry, highlighting actual threats encountered and the countermeasures implemented to mitigate them. This practical example will illustrate how organizations can effectively secure their web applications and prevent future exploitation of vulnerabilities.
This paper is organized in four main sections. Section 2 illustrates the related works. The analysis and results are presented in Section 3. The paper is concluded in Section 4.

2. Related Works

Nowadays, the huge development of web applications for business operations and customer engagement has made guaranteeing security a critical challenge. In this section, the existing studies that examine web application security, Injection attacks and Cybersecurity Threats and vulnerabilities are reviewed.

2.1 Security of Web Applications

There is no doubt that, security of web applications became a concern for any business that utilized this technology in order to return profit and gain more benefit. These applications considered as a functional-oriented where different services provided to its user in term of log-in, register, search for specific content and most of sites provide transactions which is considered a critical process that might be steeled from criminals [10]. Furthermore, it is customized for specific users with dynamic content, unlike the traditional sites which are known as static websites. The way of transforming and changing the data in these applications is between the server-side and client-side as shown in Figure 1, which is known as a three-tier architecture [11]. Therefore, one of the big challenges that any organization or business might be faced securing their assets and information from any attempt to hack or disclose information or exploit existing vulnerabilities.

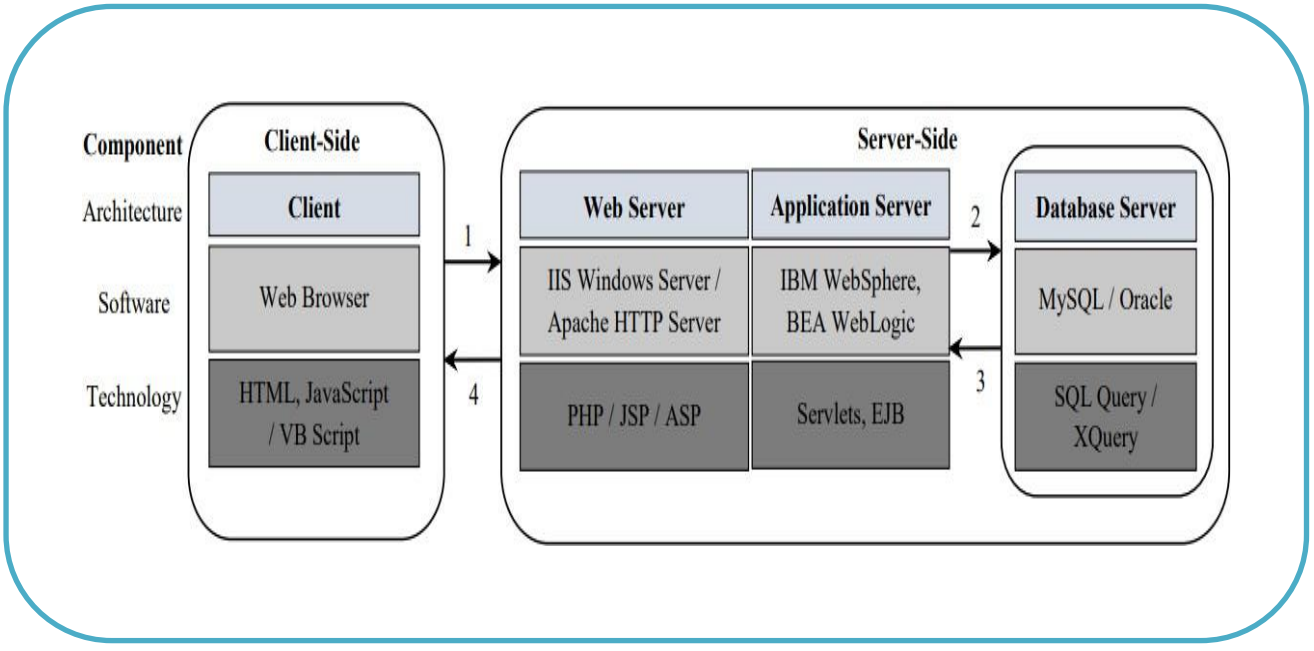


Figure 1. Architecture of web applications [11].

2.2 Cybersecurity Threats and vulnerabilities on Web Applications

Various kinds of threats arises on a web application that might be exploited by the attackers and causes serious consequences. Identifying the main cause of problems and any potential threats before happened can be reduce losses and protect their assets, prevention is better than cure [12]. Figure 2 shows the top 10 attacks that threat web applications and should be considered by web application developers, according to the Open Online Application Security Project (OWASP).

Whereas the OWASP is a nonprofit community dedicated to raising awareness about web application security in terms of the most dangerous risks that may threaten web applications as well as countermeasures, perhaps the purpose of this study is to shed light on the top two risks that threaten web application security, which are as follows: Injection and broken access control. As shown in Figure 2 the broken access control shifted from fifth place to the top of the list according to their classification in 2021. On the other hand, the injection attack was moved down from the top of the list to the third place in their new classification. More details about each attack will be presented in the following section.

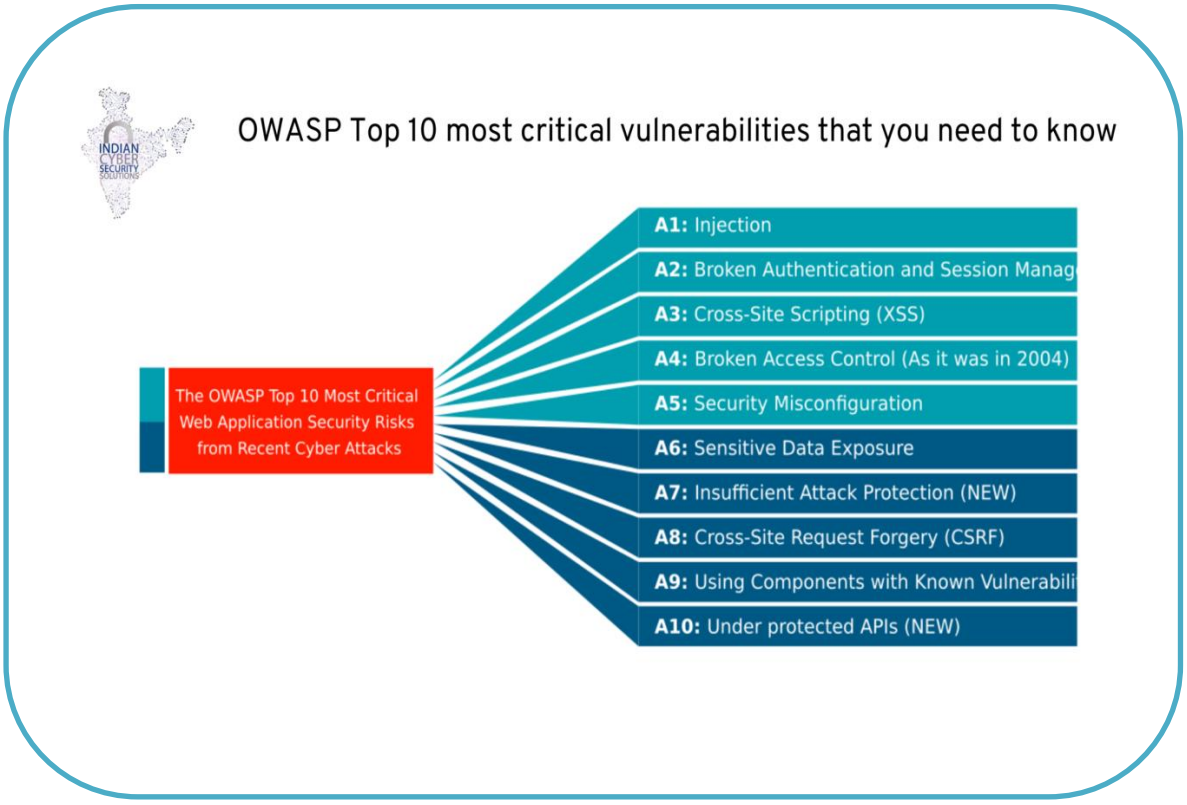


Figure 2. Top ten risks of web applications [12].

In the study by [13], a tool was developed to assess web vulnerabilities and analyze potential threats that could be exploited by cybercriminals and malicious actors. The tool provides a list of recommended solutions and countermeasures to web application administrators. The research highlighted that the lack of proper input handling mechanisms, such as sanitization and whitelisting, often leads to vulnerabilities, which attackers exploit to manipulate source code or gain unauthorized access as legitimate users. Similarly, the paper by [14] emphasizes the importance of conducting vulnerability assessments and penetration testing (VAPT), which are widely adopted by businesses to identify and remediate security flaws. These techniques help organizations evaluate whether their security systems function correctly. The study outlines the lifecycle of VAPT and introduces tools used to detect system weaknesses. It also stresses the need for continuous updates to security procedures at all organizational levels to defend against evolving cyber threats. Given the ubiquity, accessibility, and high availability of web applications, they have become prime targets for cybercriminals. Even minor implementation flaws can provide attackers with access to sensitive information or allow for malicious actions. As a result, both academia and industry have invested significant effort into developing defensive strategies for securing web applications. In [15], the authors review the current state of web application security, focusing on critical vulnerabilities such as SQL Injection and Cross-Site Scripting (XSS). The study concludes that while various countermeasures exist, regular source code reviews and fixing identified flaws are essential, as no single solution can eliminate all vulnerabilities. In [16], a comprehensive review of web security and common vulnerabilities is presented, covering various models applied in both academic environments and industrial sectors such as e-commerce. The researchers offer a set of strategic recommendations to aid government entities and affiliated sectors in adapting to digital transformation by enacting regulations and punitive

measures to deter cybercriminal activity. The study also observes a growing trend among developers to use platforms like Bootstrap and AJAX for web development—tools that prioritize usability but often overlook essential security considerations.

The study in [17] highlights the growing number of threats targeting the core dimensions of security—integrity, availability, and confidentiality—across organizational resources and assets. It notes the widespread use of VAPT to assess the effectiveness of current defenses and analyze system vulnerabilities. However, as attackers continue to develop new techniques, emerging vulnerabilities must be addressed proactively. The study recommends enhancing existing tools to support the detection of new attack signatures and ensure adaptability to evolving threats. In a related study, [18] draws attention to the increased focus on the OWASP Top Ten risks, urging businesses to ensure their web applications are protected from these critical threats. The authors explain how hackers can exploit vulnerabilities and recommend proactive mitigation strategies that should be integrated into the early stages of the development lifecycle to prevent such attacks before they occur.

Intrusion detection systems, forensic procedures, and Firewalls, are some of the tools being examined in this research [19]. Several of ML methods and data mining methods, which seek to overcome the flaws of existing technologies and generate useful solutions are also being studied. This study aims at concentrate on analytical data mining in efficient ways to identify WA attacks. The result of this study revealed that existing detection approaches like WAF and application intrusion detection systems have good accuracy and efficiency accuracy for common threats.
In the work of [20], an ML model has been constructed in order to detect and identify anomalies in the HTTP traffic of Web applications. The proposed architecture utilizes deep learning techniques, particularly the convolution neural network, which is known as CNN. There is no doubt that the main component of any ML model is the dataset. The dataset employed for this study is called "CSIC-2010v2", which contains the set of HTTP requests. The proposed model trained the CNN to automatically detect and identify the hidden pattern in HTTP. The results of this model achieved a promising result with an accuracy of 97, 07%. In the work of [21], a state of art framework has been proposed for analyzing and detecting any vulnerability in the source code at the same time as the web applications development life cycle. Furthermore, the proposed model is able to detect vulnerabilities that were previously impossible to identify using traditional techniques. The results of the proposed framework show that, by utilizing dynamic queries, it is feasible to identify the common attacks against authentication breaches and SQL injection threats, which were previously deemed impossible to detect.

[22] Reported that, the number of online applications that are hacked every day is believed to be approximately 30,000, and in the vast majority of instances, web programmers or website administrators have no idea what is going on with their sites in the first place. Due to the importance of the security of web applications, specifically, the scope of this study includes the establishment of mitigations against the most common web application attacks, as well as the provision of methods for web administrators to detect phishing links, which is a type of social engineering attack. Additionally, the study demonstrates the generation of web application logs, which simplifies the process of analyzing the actions of abnormal users to show when behavior is out of bounds, out of scope, or against the rules, and the demonstration of the generation of web application logs. The mitigation techniques that have been proposed as the following, secure coding techniques, while the detecting phishing link attacks the authors of this study employed different machine learning model along with deep learning. The developed application has been tested and evaluated against a variety of attack scenarios; the results of the testing process revealed that the website had successfully mitigated these dangerous web application attacks; and for the detection of phishing links part, a comparison between different algorithms is made in order to find the best one; the results of the best model revealed 98 % in term of accuracy.

2.3 Injection Attack

Recently injection attacks have been reached the top of OWASP top ten risks, where putting it one of the most severe types of attacks against web applications [23]. Different types of injection such as LDAP, Object Graph Navigation Library (OGNL), Expression Language (EL), OS command, NoSQL and SQL injection [24]. These attacks take place when an interpreter receives arbitrary data as part of a statement and query from the user input [25]. In our research, the SQL injection attack has been studied due to its dangers. Furthermore, most previous studies and the security associations and organizations have been rated it as the one of top ten attacks that might be threatening web applications. SQL stands for Structured Query Language, which is a well-known programing language used for managing and controlling the data in the database such as, INSERT a new record, DELETE and RETRIEVE and much more operations can be executed by utilizing these commands [26].

SQL injection attacks can also be used to retrieve sensitive data or might be gain unauthorized access to websites without usernames or passwords. As shown in Figure 3, the attacker bypassed the authentication mechanism successfully by inserting an expression of Boolean algebra. When the following vulnerabilities exist, an application is exposed to attack. According to OWSAP, there are different vulnerabilities that must be addressed to ensure the application is saved from injection attacks as the following. Lack of input handling techniques, there is no proper way to filter the user input such as sanitization ad validation methods.

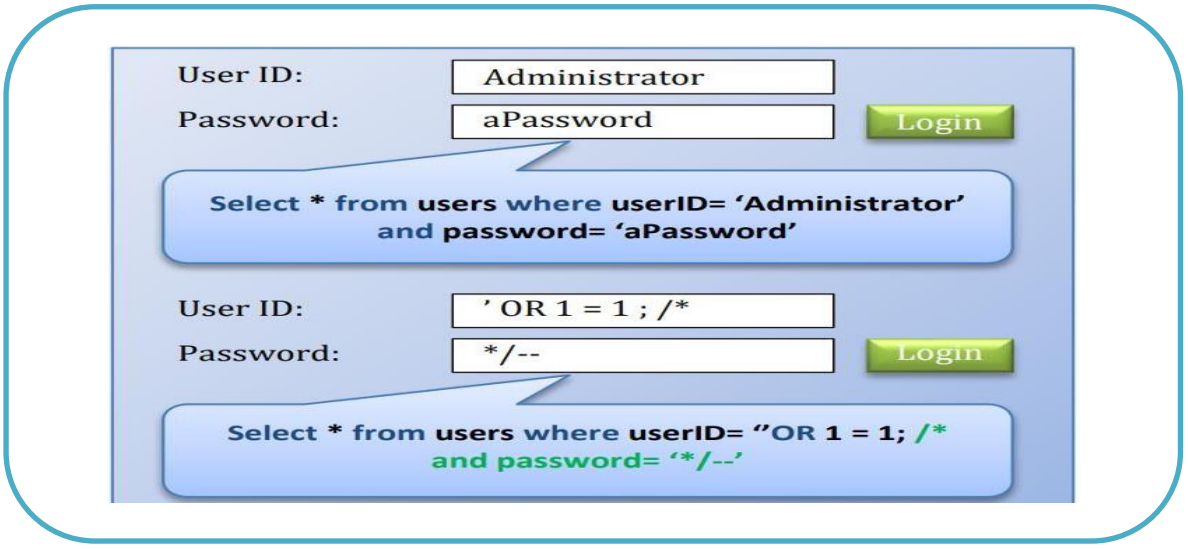


Figure 3. SQL Injection Attack [25].

2.3.1 String SQL Injection

SQL Injection is regarded as one of the most dangerous attacks against databases. In this attack, the attacker can abuse and exploit SQL code by injecting it straight into the input field, allowing them to change data stored in the database and obtain unauthorized access to system resources. The attacker can take advantage of these flaws because of the weakness of the input handling mechanism in web applications. An SQL Injection attack can have a variety of effects on a database, including unauthorized access to the database, manipulation, and extraction of sensitive data. This problem is dangerous because it could lead to data loss or unauthorized access, which would compromise functionality and confidentiality [26]. This type of attack relies on injecting SQL statements inside different conditional statements, such as "1=1" and --, to evaluate the condition as true. OR/AND Attack [26] is another name for this type of attack. This type of attack is particularly dangerous because, rather than collecting a single row from the database, it retrieves full target rows [26]. The hackers use this type of attack to extract data from a database or break the authentication mechanisms which is known as bypassing authentication [26]. Figure 4 shows an example of a string SQL injection query.

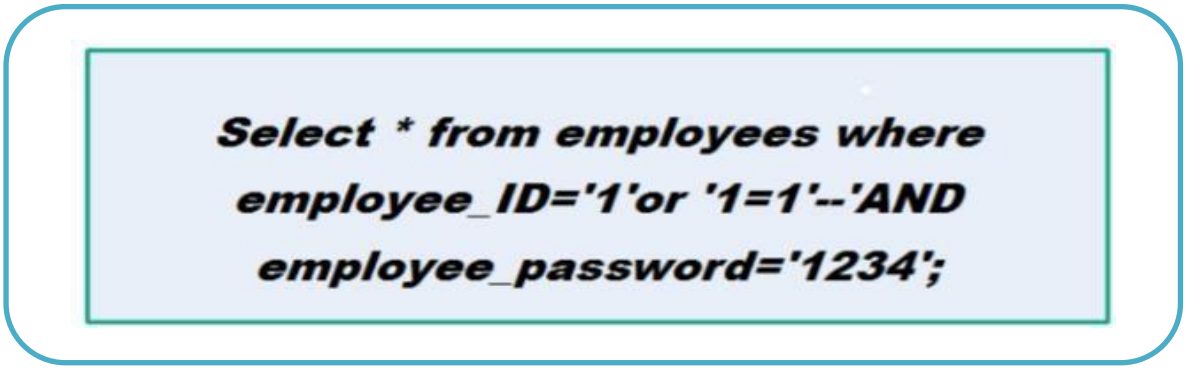


Figure 4. SQL Injection Attack [26].

2.3.2 Data Extraction

This type of attack employs strategies for retrieving data values from databases. This information may be sensitive and highly wanted by the attacker, depending on the online application type. The most often exploited SQLIA attacks are those with this purpose [27]. The attacker is attempting to inject the SQL WHERE clause into the injectable field. As a result of the conditional statement's translation to a tautology, all of the intended entries in the database were obtained. In general, hackers must analyze the vulnerable parameters and the code structures that evaluate the query. When the code either displays all of the returned data or performs some action if just one record is retrieved, the attack is usually successful [27].

2.3.3 Bypassing authentication

This attack can be defined as bypassing the authentication mechanism of web applications. If the attacker successfully launches such an attack, he will be able to take the privilege of another user and log in to the system as the legitimate user [28]. As we mentioned example of this attack was previously shown in Figure 1.

2.4 Detection and prevention of SQLI Attacks

Many studies have been conducted to identify and prevent SQL injection attacks at an early phase by proposing countermeasures and preventative approaches. The previous recent studies are reviewed and presented in this section, along with the intended aims and methodology. In the work of [28], a novel ML-based model has been proposed to detect an SQLI attack by employing a Support Vector Machine (SVM) classifier by representing SQL queries as a network and sequences of tokens to assess and calculate the centrality of nodes. They investigate several techniques for constructing token graphs and propose alternate schemes that include SVMs. The proposed solution is developed to address the firewall layer of database targeting a shared server provider which can provide the protection to web applications. The proposed solution is designed for web applications that are developed in PHP and MySQL. Meanwhile, it is easily adaptable to other platforms. The result of this study revealed that this technique may effectively identify fraudulent SQL queries with a few performance impacts.

The study conducted by [29] has intended to propose a novel framework known as DIAVA which is capable of detecting and analyzing SQLI traffic. DIAVA tool was able to deliver users proactive alerts. DIVA can correctly determine attempted SQLIAs across all potential threats by evaluating the two-directional network traffic of SQL Statements and implementing the suggested multilayer regular expression framework. Despite the fact that most cloud service providers deliver web application firewalls, users are unwilling to subscribe because only a few methods can detect and report reliable SQLIA information and present it to the installed application. The findings of this study show that DIAVA excels on WAFs in respect of identifying SQLI, but it also enables a real-time vulnerability detection of data leaked via SQL injection.

Meanwhile, [30] discussed and reviewed the different types of SQLI, such as tautologies, union queries, piggy-backed queries, and prevention and detection techniques corresponding to above-mentioned attacks. Considering the prevention techniques, the authors of this study reviewed the following techniques: using stored procedures, preparing a statement, validating user input, and encrypting data. Consequently, the study proposed a model for detecting SQLIA by detecting a query token using a lexicon. To mitigate SQLIA, the approach includes two steps: starting with developing a lexicon, followed by tokenizing the input query phrase, with each string token being recognized against a predetermined terms lexicon. The proposed model was evaluated and the revealed results were promising.

In as study [31] have developed a model that is capable of detecting SQLI depending on a dynamic-based scanner analyzer. The suggested model has different features, including broad scalability, reliable performance, support of a large variety of SQL Injections techniques, and resource efficiency. Compared to another tool, the proposed solution is based on a dynamic scanner and performs well in detecting and blocking the SQLIA. Furthermore, no updating needs for the web application's source code, and it uses the system's resources to the bare minimum. One of the key benefits of the proposed tool is that it can detect the advanced level of SQLIA methods because the knowledge base has been updated to deal with the latest threats. The study of [32] was carried out in order to detect and prevent SQL injection vulnerabilities using three approaches [3]. Prepared statements, stored processes, and white list input validation are the three preventative approaches used. To avoid SQL injection threats, basic queries are used in the methods. The assessment of this technique for identifying and preventing web server vulnerabilities.

The study in [33] Introduced a structured approach in order to eliminate and protect systems against different types of SQLIA with respect to multi-languages syntax. Codify tautology and alternate solution encoding attacks using SQL statements and limited automata, and provide SQL statements and code for ASP.net server that developers utilized to detect any chance of SQLA that threats the web applications implemented with Microsoft SQL Server. The result of this study revealed that the proposed model could detect and prevent common types of SQLA. Furthermore, the model can detect attacks written in a language other than English. The paper introduced by [34] has proposed a model for detecting and eliminating SQLA by validating the code in runtime and can be applied to any legacy system without requiring any client-server modifications or knowledge about web application source code. Additionally, modification tolerance is achieved by introducing an additional layer of middleware placed between the server-client architecture. As a result, any validation method is performed on this middleware like a proxy which is capable of sanitizing users' inputs in order to identify and eliminate SQLIA. Consequently, the results of the proposed medal achieve high accuracy with a percentage of 86.6%. A comparative study of ML-based of SQLA detection and prevention techniques was conducted by [35]. Here, the majority of the reviewed papers used ML algorithms and achieved high accuracy and the Artificial Neural Network outperformed with an accuracy of 99.23%. According to the authors of this paper, there is a shortage of a dataset that can be used to test and validate the ML models for identifying and avoiding SQLA.

While [36] utilized qualitative and quantitative methodology to assess different standard SQLIA tools and SQLIA prevention techniques, indeed, the virtual website was designed as simulated websites for modeling in their exploratory test platform, then conducted SQLIA penetration testing on the virtual websites to assess SQLIA tools and prevention methods. The findings of the studies reveal that SQLIA tools can effectively breach the DBMS of websites along with operating systems. This study contributes by evaluating many commonly used SQLIA tools and SQLIA prevention strategies and recommending and testing numerous novel filters for preventing SQL injection. In the work of [37], an ML model based on SQLIA was developed. The authors of this study employed a dataset consisting of 616 SQL statements to train and evaluate the model. Furthermore, twenty-three common machine learning classifiers were employed. They select the top five classifiers from among those employed based on their detection accuracy. Additionally, they designed a graphical user interface (GUI) that uses these five classifiers as its foundation to make the results clear for users. According to the findings of this experiment, the proposed model achieved a promising result with an accuracy of 99.5% and successfully detected the SQLIA The paper introduced by [38] aims to develop a Machine learning Model based that allows the building of expert systems in cyber data attacks by employing fuzzy rules, emphasizing the SQL Injection attack. The tests were carried out on genuine SQLIA databases. According to the findings, obtained, the viability of building a system using fuzzy rules achieved 99% accuracy, which is considered a promising result.

2.5 Broken Access Control

This kind of attack ranked as the top one of the web applications vulnerabilities according to OWSAP in 2021 ("OWASP Top Ten Web Application Security Risks | OWASP", 2022). Access Control referred to a maintaining privilege by preventing users from working beyond the scope of their specified permissions. Flaws generally take place and break the three dimensions of security confidence, for Instance, illegal information disclosure, integrity such as data alteration or loss, and availability. It can be presented in different forms such as the absence of an access control method i.e. POST and DELETE access restrictions are not present when using the API. Meanwhile, attackers may exploit the vulnerability of access control mechanisms by altering the URL parameter of the HTL page or by abusing hacking tools to intercept and manipulate API responses and requests.

2.6 Detection and preventions of Broken Access Control Attacks:

Many projects have been carried out for detecting and preventing broken access control attacks: either by preventing this kind of attack in the early stage or proposes and recommend a set of countermeasures and preventions techniques. [39] Conducted a quantitative assessment study to identify and discuss broken access control vulnerabilities in web applications in order to raise awareness among web application designers, developers, administrators, and web owners about the possibility of the existence of BAC vulnerabilities in web applications, taking into account the document's facts and findings before releasing the application to the public. According to the findings of this study, failure to resolve session

misconfiguration and input validation concerns is a crucial contributor to the application's BAC vulnerability. According to [40], there are many other kinds of access control policies, but the three most common are the DAC, MAC, and RBAC policies, which are all based on rules of privilege. ABAC is a viable alternative to classic access control policies (DAC, MAC, and RBAC) in the current era, and it has received considerable attention in recent academic research as well as industrial. These numerous policies may be used in conjunction with one another to make systems more secure. Furthermore, OWSAP provides a list of good practices that might be utilized by the developers as the following ("OWASP Top Ten Web Application Security Risks | OWASP", 2022): Disable and deny the default access of the critical resources by employing just in time access. Reduce the utilization of cross-origin resource sharing by implementing access control methods and then reusing them across the web API. Assessing and evaluating access control repeatedly, which might help the developer to detect and prevent any vulnerabilities and considered an effective way.

3. Analysis and Results

Following a comprehensive review of existing literature on cybersecurity threats targeting web applications, this section aims to address the following key research questions:
- What are the cybersecurity attacks that pose a threat to web applications?
- What are the mitigation strategies or solutions available to protect web applications from these attacks?

To answer these questions, Table 1 provides a summary of the reviewed studies, highlighting various types of cyberattacks and their corresponding mitigation techniques aimed at enhancing web application security. Before presenting the direct answers to the research questions, additional contextual analysis is provided. For example, Figure 5 and 6 illustrate the distribution of the reviewed publications by year, reflecting the growing scholarly interest in this domain. Additionally, the research encompasses both qualitative and quantitative methodologies, as shown in Figure 6.

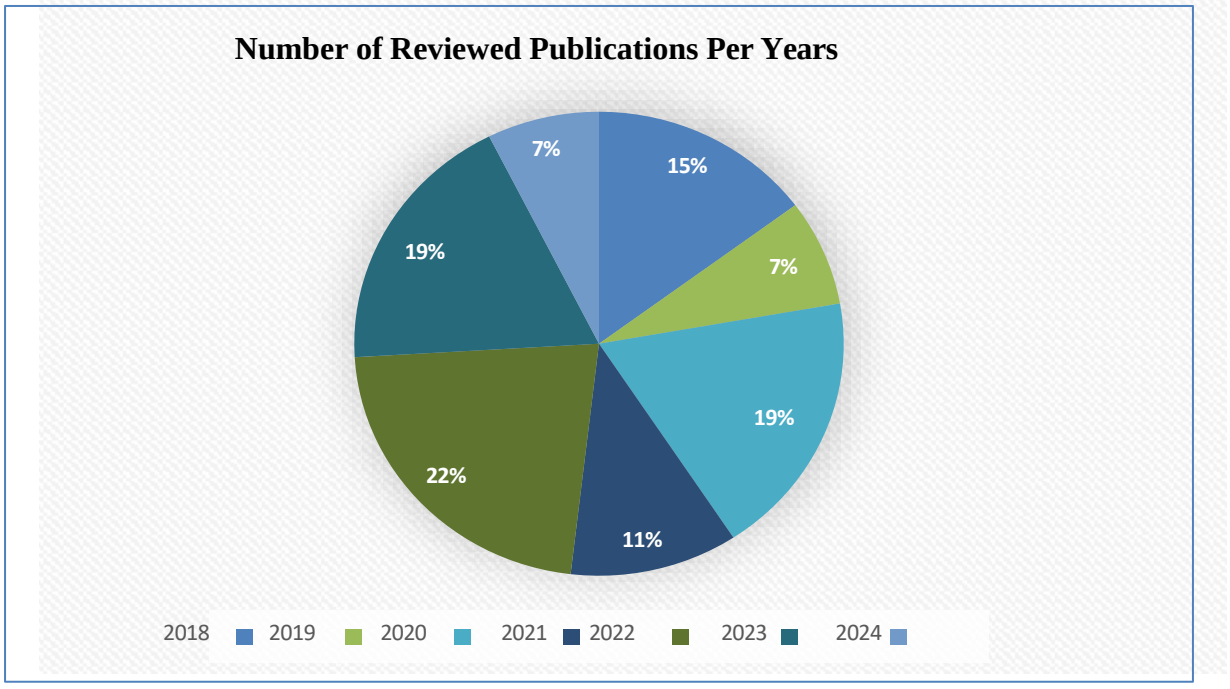


Figure 5. Number of Reviewed Publications Per years

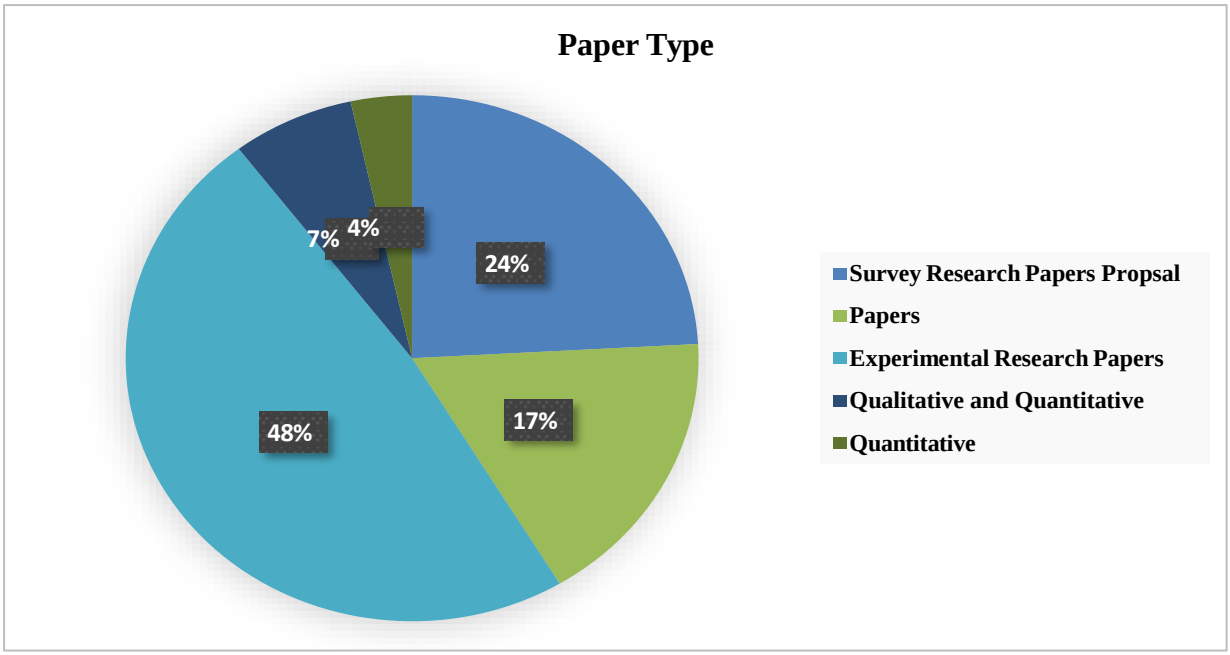


Figure 6. Paper Type

Table 1. Comprehensive analysis of reviewed literature

Reference	Title	Type of attacks in Web applications	Proposed mitigation techniques
(Darus, 2020)	Web Vulnerability Assessment Tool for Content Management System	This paper discussed different kind of vulnerabilities that might be attack web application such as Cross-Site Scripting (XSS), SQL Injection, CRLF Injection Local File Inclusion (LFI) and Remote File Inclusion (RFI).	This study proposed a novel evaluation tool called "SNEAKERZ" that automatically detected and assessed web application vulnerabilities, as well as provided mitigations techniques for such vulnerabilities.
(Khera et al., 2019)	Analysis and Impact of Vulnerability Assessment and Penetration Testing	Injection, Broken Authentication, Sensitive Data Exposure, XML External Entities (XXE), Broken Access Control, Cross-Site Scripting (XSS), Security Mis-configuration	This study examines the life cycle of vulnerability assessment and penetration testing, as well as the tools used to investigate system flaws, such as Wire Shark, Nmap, and Metasploit.
(Deepa&Thilagam,2016)	Securing web applications from injection and logic vulnerabilities: Approaches and challenges	This paper shed light on reviewing the status of web applications security and the significant flows such as SQL injection and Cross-site scripting,	Reviewing the source code and repairing any flaws is highly recommended since no one solution can address all of the system flaws

(Divyaniyadav et al.,2018)	Vulnerabilities and Security of Web Applications	A comprehensive study of web security and its vulnerabilities was presented. Different models have been described that are extremally employed within different sectors such as academics and applications of industry like e-commerce	A set of recommendations that can help the government and its sectors adopt the digital transformation to legislate regulations and restrictive rules to prevent the attackers from any cybersecurity attack by imposing punishments. Furthermore, according to the researchers' observation, recently, most developers tend to utilize the following platforms to implement the websites Bootstrap and AJAX.
(Shinde & Ardhapurkar, 2016)	Cyber Security Analysis using Vulnerability Assessment and Penetration Testing	Cross-Site Scripting (XSS), SQL Injection (SQLi)	Vulnerability Assessment and Penetration Testing (VAPT) have been used by organizations to test their systems' defensive capabilities and examine the security of their systems.
(Kumar et al., 2021)	Risks and Threats to Web Applications and Their Preventions:	Injection, Broken Authentication, Sensitive Data Exposure, XML External Entities, Broken Access Control, Security	One of the good practices that must be followed don't trust user input. In order to secure the data, don't develop the database within the same server of web applications since admin accounts are conveniently targeted.
(Babiker Karaarslan & Hoscan, 2018)	Web Application Attack Detection and Forensics: A Survey	The authors of this paper don't mention specific attacks	Intrusion detection systems, forensic procedures, and Firewalls, are some of the tools being examined in this research
(Tekerek , 2021)	A Novel Architecture for Web-Based Attack Detection Using Convolution Neural Network	An ML model has been constructed in order to detect and identify anomalies in the HTTP traffic of Web applications.	The proposed architecture utilizes deep learning techniques, particularly the convolution neural network, which is known as CNN. There is no doubt that the main component of any ML model is the dataset. The dataset employed for this study is called "CSIC-2010v2", which contains the set of HTTP requests. The proposed model trained the CNN to automatically detect and identify the hidden pattern in HTTP. The results of this model achieved a promising result with
(Kubota et al., 2020)	A New Feature to Secure Web Applications	SQL injection	This study proposed a novel framework for analyzing and detecting any flaws in the source code of developed web applications. Furthermore, the proposed model is able to detect vulnerabilities that were previously impossible to identify using traditional techniques. The results of the proposed framework show that, by utilizing dynamic queries, it is feasible to identify the common attacks against authentication breaches and SQL

(Hashim et al., 2021)	Defenses Against web Application Attacks and Detect Phishing Links Using Machi Learning	Detect phishing links	The methods of mitigation are done through the use of secure coding techniques, while the methods of phishing link detection are completed through the use of different machine learning algorithms and deep learning techniques. The developed application has been tested and evaluated against a variety of attack scenarios; the results of the testing process revealed that the website had successfully mitigated these dangerous web application attacks; and for the detection of phishing links part, a comparison between different algorithms is made in order to find the best one; the results of the best model revealed 98 % in term of accuracy.
(Maruf Hassan et al. 2018)	Broken Authentication and Session Management Vulnerability: A Case Study Of Web Application	Broken access control vulnerabilities within WA.	In order to raise awareness among web application designers, developers, administrators, and web owners about the possibility of the Existence of BAC vulnerabilities in web applications, taking into consideration the facts and findings of the document before putting the application live on the internet. The result of this study shows that the failure to address session misconfiguration and input validation issues is considered a significant contributor to the application's BAC vulnerability.
(Aliga, John-Otumu, Imhanhahimi & Akpe, 2018)	"Cross-Site Scripting Attacks in Web-Based Applications: A Critical Review on Detection and Prevention Techniques"	Cross-Site Scripting	In this paper critical review has been done to study detection and prevention techniques of cross-site scripting, the authors of this paper reported that most of the solutions proposed were addressed on the client-side.
(Singh, Sharma, Sharma, Kaushik & Bhushan, 2019)	"Taxonomy of Attacks on Web Based Applications "	This paper addresses the following attacks SQL Injection, and XPath Injection	Some mitigation measures include utilizing a web-based firewall to prevent unwanted user access, adopting mitigation and detection mechanisms to safeguard the network, to preventing SQLIA from employing different input handling techniques such as sanitizing and parameterizing SQL statements.

According to the studied literature, the SQL injection attack is one of the most dangerous attacks that threaten Web applications, as illustrated in Figure 7. According to our findings, the total number of reviewed papers that reported an SQLI attack is nine papers, whereas six papers reported the Cross-Site Scripting attacks. Furthermore, the Open Online Application Security Project (OWASP) reported that SQL injection attacks have risen to OWASP's top 10 threats, making them one of the most severe web application attacks [41]. Furthermore, one of this study objectives was to expose the cybersecurity attacks that are reported by academia along with the reviewing the security institutions reports from the industry specifically

in this study OWASP was considered, Figure 8, illustrates the common cybersecurity attack that threaten web applications regarding to the academia, which all the reviews papers and the OWSAP report.

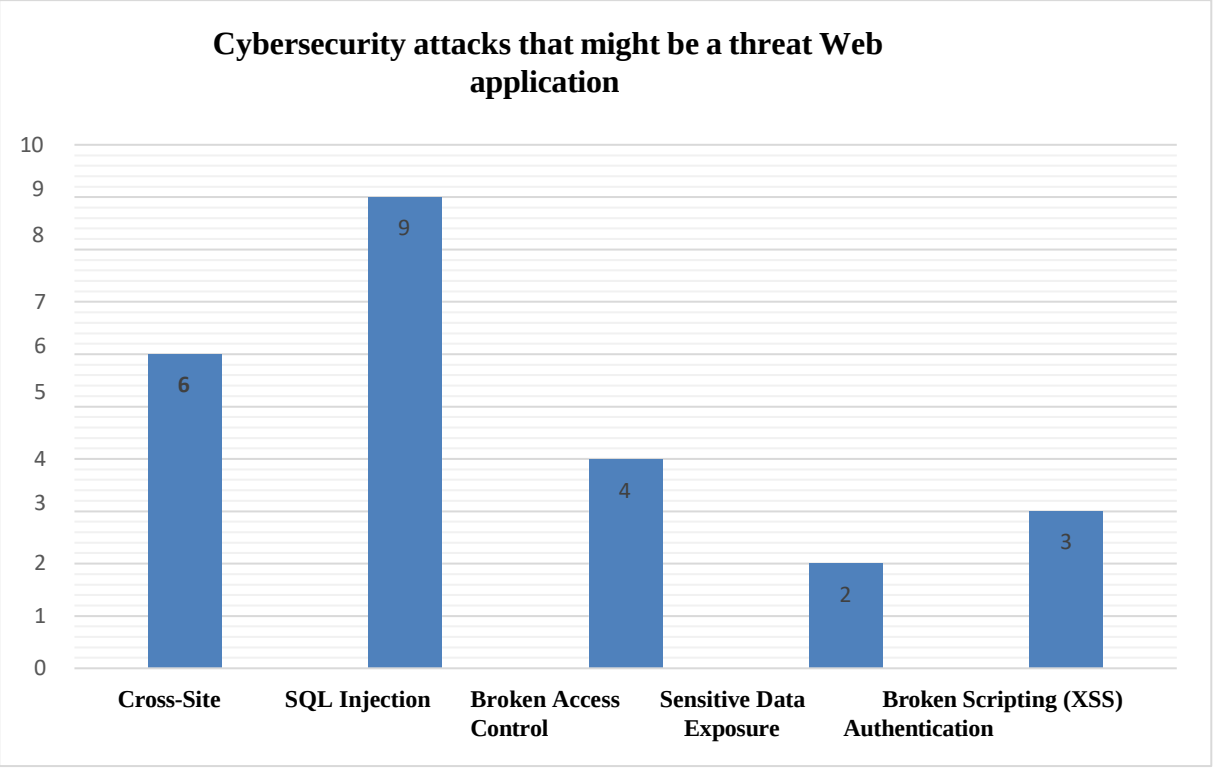


Figure 7. Cybersecurity attacks that might be a threat Web application

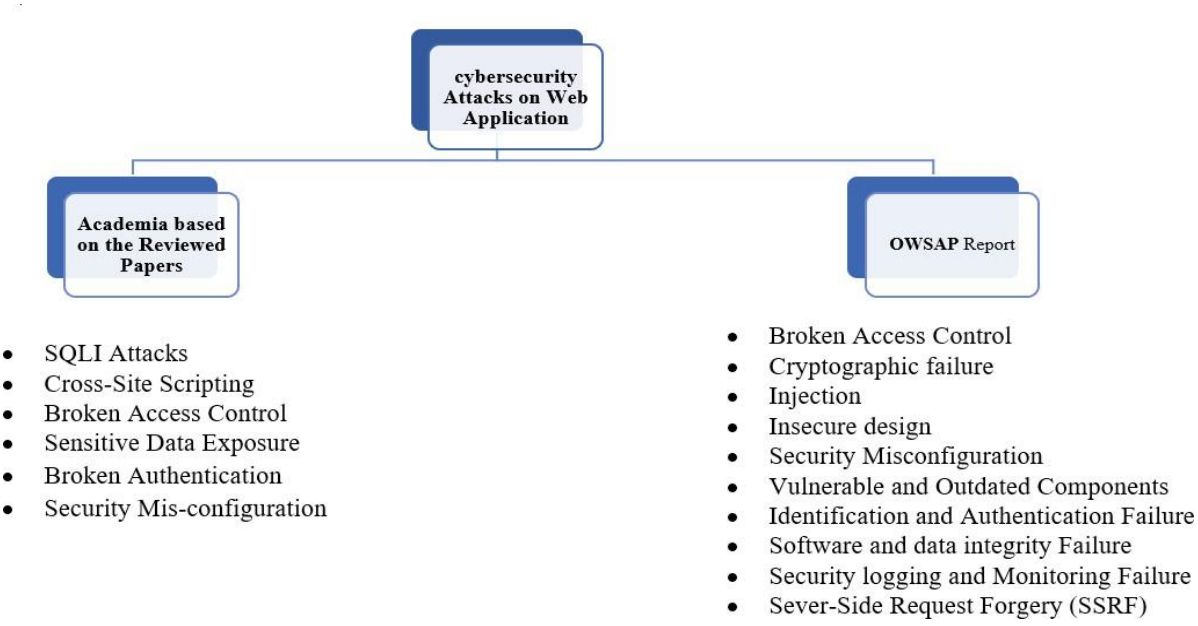


Figure 8. Common Cybersecurity Attacks that threaten web applications

Due to that, we want to shed light on this type of attack and answering the following question: How can Web applications be protected against SQL injection attacks? Before answering this question, this study shed light on the common SQLIAs that reported by the reviewed literature Figure 9 shows the common type of SQLIA from the reviewed literature [42].

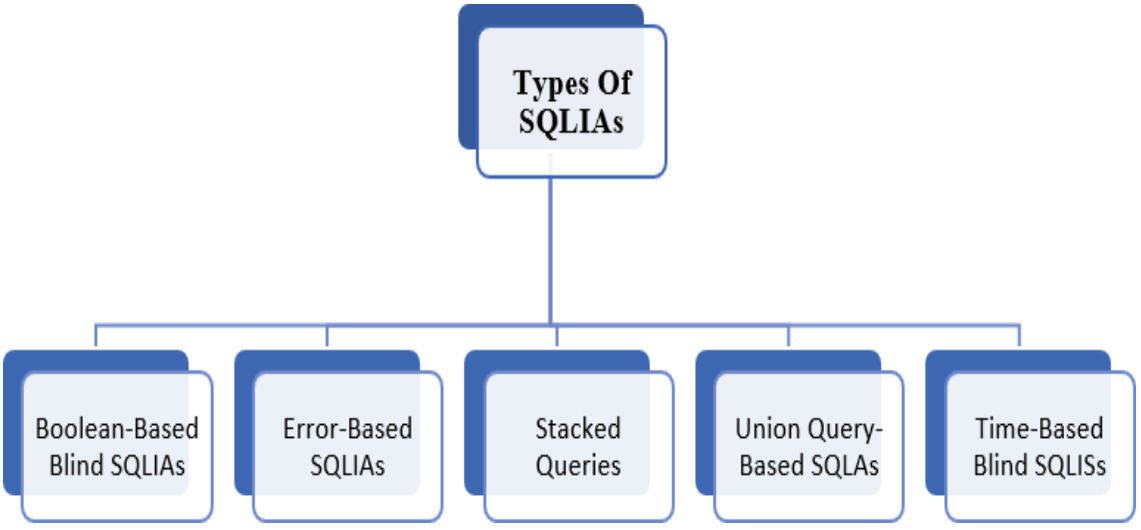


Figure 9. Types of SQLIAs

Table 2. Comprehensive analysis of reviewed literature

Literature	Title	Goal of the study	Mitigation techniques- findings
(Kar et al.,2016)	Detecting SQL injection attacks using graph of tokens and SVM. Computers and Security	Detect an SQLI attack	A novel ML-based model has been proposed by employing a Support Vector Machine (SVM) classifier by representing SQL queries as a network and sequences of tokens to assess and calculate the centrality of nodes. They investigate several techniques for constructing token graphs and propose alternate schemes that include SVMs.
(Gu et al.,2020)	DIAVA: a traffic-based framework for detection of SQL injection attacks and vulnerability analysis of leaked data	Detecting and analyzing SQLI traffic	This paper intended to propose a novel framework known as DIAVA which is capable of detecting and analyzing SQLI traffic. DIAVA tool was able to deliver users proactive alerts. DIVA can correctly determine attempted SQLIAs across all potential threats by evaluating
(Su et al. 2020)	Detection and Prevention Technique on SQL Injection Attacks		The two-directional network traffic of SQL Statements and implementing the suggested multilayer regular expression framework.

		Discussed and reviewed the different types of SQLI, such as tautologies, union queries, piggy-backed queries, and prevention and detection techniques corresponding to above- mentioned attack	Considering the prevention techniques, the authors of this study reviewed the following techniques: using stored procedures, preparing a statement, validating user input, and encrypting data. Consequently, the study proposed a model for detecting SQLIA by detecting a query token using a lexicon. To mitigate SQLIA, the approach includes two steps: starting with developing a lexicon, followed by tokenizing the input query phrase, with each string token being recognized against a predetermined terms lexicon. The proposed model was evaluated and the revealed results were promising.
Nadeem et al. (2017)	Detection and Prevention of SQL Injection Attack by Dynamic Analyzer and Testing Model	This paper aimed at to develop a model that is capable of detecting SQLI depending on a dynamic-based scanner analyzer	The suggested model has different features, including broad scalability, reliable performance, support of a large variety of SQL Injections techniques, and resource efficiency. Compared to another tool, the proposed solution is based on a dynamic scanner and performs well in detecting and blocking the SQLIA. Furthermore, no updating needs for the web application's source code, and it uses the system's resources to the bare minimum.
(Kumar et al., 2018)	Vulnerability detection and prevention of SQL injection	This study was conducted to employ three different techniques for detecting and preventing SQL injection vulnerabilities	Prepared statements, stored processes, and white list input validation are the three preventative approaches used. To avoid SQL injection threats, basic queries are used in the methods. The assessment of this technique for identifying and preventing web server vulnerabilities
(Qbea'H et al., 2016)	Detecting and preventing SQL injection attacks: a formal approach.	Introduced a structured approach in order to eliminate and protect systems against different types of SQLIA with respect to multi- languages syntax	Based on codify tautology and alternate solution encoding attacks using SQL statements and limited automata, and provide SQL statements and code for ASP.net server that developers utilized to detect any chance of SQLA that threats the web applications implemented with Microsoft SQL Server. The result of this study revealed that the proposed model could detect and prevent common types of SQLA. Furthermore,
(Hadabi et al., 2022)	An Efficient Model to Detect and Prevent SQL Injection Attack.	This paper has proposed a model for detecting and eliminating SQLIA	By validating the code in runtime and can be applied to any legacy system without requiring any client-server modifications or knowledge about web application source code. Additionally, modification tolerance is achieved by introducing an additional layer of middleware placed between the server-clint architecture. As a result, any validation method is performed on this middleware like a proxy which is capable of sanitizing users' inputs in order to identify and eliminate SQLIA. Consequently, the results of the proposed medal achieve high accuracy with a percentage of 86.6%.

(Jemal et al., 2020)	SQL Injection Attack Detection and Prevention Techniques Using Machine Learning	A comparative study of ML- based of SQLA detection and prevention techniques was conducted in this study	The result of this study revealed that the majority of the reviewed papers used ML algorithms and achieved high accuracy and the Artificial Neural Network outperformed with an accuracy of 99.23%. The authors reported that there is a lack of a dataset that can be utilized to test and evaluate the ML models for detecting and preventing SQLA.
(Zhu & Yan 2017)	Exploring defense of SQL injection attack in penetration testing.	This study utilized qualitative and quantitative methodology to assess different standard SQLIA tools and SQLIA prevention techniques.	The virtual website was designed as simulated websites for modeling in their exploratory test platform, then conducted SQLIA penetration testing on the virtual websites to assess SQLIA tools and prevention methods. The findings of the studies reveal that SQLIA tools can effectively breach the DBMS of websites along with operating systems. This study contributes by evaluating many commonly used SQLIA tools and SQLIA prevention strategies and recommending and testing numerous novel filters for preventing SQL injection.
(Hasan et al., 2019)	Detection of SQL injection attacks: A machine learning approach	Detect an SQLI attacks by ML model	An ML model based on SQLIA was developed. The authors of this study employed a dataset consisting of 616 SQL statements to train and evaluate the model. Furthermore, twenty-three common machine
(Batista et al., 2018)	Fuzzy neural networks to create an expert system for detecting attacks by SQL Injection	Detecting SQLIA	Machine learning Model has been constructed based on expert systems in cyber data attacks by employing fuzzy rules, emphasizing the SQL Injection attack. The tests were carried out on genuine SQLIA databases. According to the findings, obtained, the viability of building a system using fuzzy rules achieved 99% accuracy, which is considered a promising result.
(Zuech, Hancock & Khoshgoftaar, 2021)	Detecting SQL Injection Web Attacks Using Ensemble Learners and Data Sampling	Detecting SQL Injection	The ensemble learner model (Light GBM,) achieved the highest AUC score of 94%
(Raut, Nikhare, Punde, Manerao & Choudhary, 2019)	A Review on Methods for Prevention of SQL Injection Attack	Conducting a review paper about the prevention techniques of SQLIA	This paper developed a scheme for detecting and preventing SQL Injection Attacks. The system is fully automated, and it recognizes SQLIAs using a model-based technique that strengthens static and segment examination. This model can work with a variety of databases.

(Thombare & Soni, 2022)	Prevention of SQL Injection Attack by Using Black Box Testing	Prevent SQL Injection Attack	They develop a black-box testing tool in the fully automated method for eliminating SQL injection attacks of SQL Injection Vulnerability. When such attacks are conducted, the SQLIV is immediately analyzed.
(Harefa et al., 2021)	The Prevention of SQL Injection Attacks on Web Applications	Detect and prevent different types of SQLIA	The proposed model is designed to detect and prevent the following attacks Tautologies, Logically Incorrect Queries, Union Queries, Piggy Backed Queries, and Stored Procedures.

Based on the literature review, there are different schema and model has been proposed and developed to detect SQL injection attacks. Various results obtained with the different performance levels. Since the AI considered an emerging filed, the most reviewed papers were employed an ML based model to mitigate and detect SQLI attacks with percent of (33%), Figure 10 shows the mitigations techniques that have been developed from the literature.

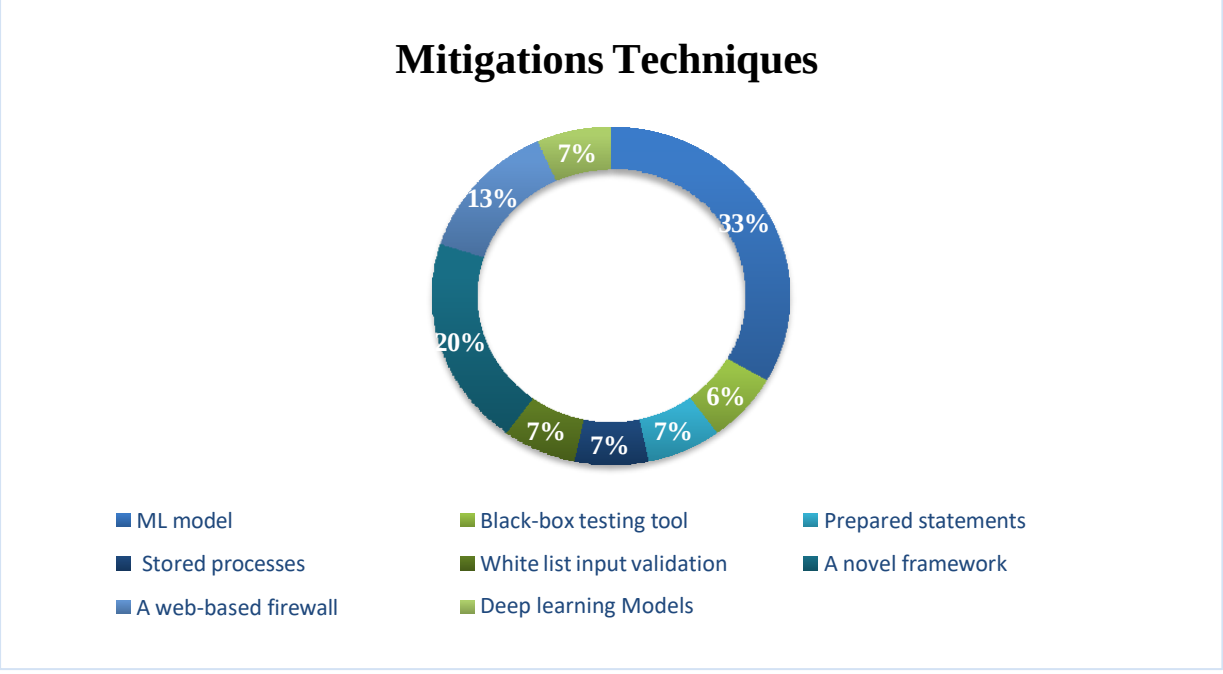


Figure 10. Mitigations Techniques

4. Conclusion and Future Works

Web applications have gained much relevance due to the digital transformation and abundance of services provided throughout these applications. Most businesses globally tend to utilize web technology to digitize their services to save time and money. Web applications contain critical information that must be protected against cybersecurity attacks, which expose the vulnerabilities and flaws ignored or not adequately addressed by the application developer. Therefore, this study aimed to thoroughly understand the web application and potential threats that might be affected and attack these applications. A comprehensive review of the current studies regarding cyber-security threats in Web applications and their mitigations: Based on our findings, SQL injection attacks are considered a dangerous attack that threatens web applications;

considering the mitigation techniques, the most proposed techniques were employed in AI models. This paper sheds light on the SQLI attacks and other attacks and are highly recommended to other researchers. There is a lack of studies that shed lights on the broken access control mitigation techniques. Furthers studied in this field are highly recommended.

Corresponding author
Bashaer Almelehy
221445338.student@kfu.edu.sa

Acknowledgements
All authors would like to thank King Faisal University, Saudi Arabia for all supports in terms of labs, funding etc.

Funding
This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. KFU 210587).

Contributions
B.A; M.A; G.N;M.M; A.A; Conceptualization, B.A; M.A; G.N;M.M; A.A;; Investigation, B.A; M.A; G.N;M.M; A.A; Writing (Original Draft), B.A; M.A; G.N;M.M; A.A;; and B.A; M.A; G.N;M.M; A.A; Writing (Review and Editing) Supervision, B.A; M.A; G.N;M.M; A.A; Project Administration.

Ethics declarations
This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication
Not applicable.

Competing interests
All authors declare no competing interests.

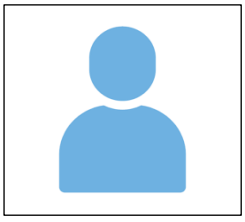
References

[1] Abdoulaye Kindy, D., & Khan Pathan, A.-S. A Detailed Survey on Various Aspects of SQL Injection in Web Applications: Vulnerabilities, Innovative Attacks, and Remedies. International Journal DRAFT.
[2] Aliga, A. P., John-Otumu, A. M., Imhanhahimi, R. E., & Akpe, A. C. (2018). Cross Site Scripting Attacks in Web-Based Applications. Journal of Advances in Science and Engineering, 1(2), 25–35.
[3] Alenezi, M., Nadeem, M., & Asif, R. (2020). SQL Injection Attacks Countermeasures Assessments. Indonesian Journal of Electrical Engineering and Computer Science, 21(2), 1121–1131. https://doi.org/10.11591/ijeecs.v21.i2.pp1121-1131
[4] Babiker, M., Karaarslan, E., & Hoscan, Y. (2018, March). Web Application Attack Detection and Forensics: A Survey. In 2018 6th International Symposium on Digital Forensic and Security (ISDFS) (pp. 1–6). IEEE.
[5] Batista, L., Silva, G., Araújo, V., Rezende, T., Guimarães, A., & Souza, P. (2018). Fuzzy Neural Networks to Create an Expert System for Detecting Attacks by SQL Injection. The International Journal of Forensic Computer Science, 13(1), 8–21. https://doi.org/10.5769/j201801001
[6] Gu, H., Zhang, J., Liu, T., Hu, M., Zhou, J., Wei, T., & Chen, M. (2019). DIAVA: A Traffic-Based Framework for Detection of SQL Injection Attacks and Vulnerability Analysis of Leaked Data. IEEE Transactions on Reliability, 69(1), 188–202.
[7] Darus, M. Y. (2020). Web Vulnerability Assessment Tool for Content Management System. International Journal of Advanced Trends in Computer Science and Engineering, 9(1.3), 440–444. https://doi.org/10.30534/ijatcse/2020/6991.32020
[8] Deepa, G., & Thilagam, P. S. (2016). Securing Web Applications from Injection and Logic Vulnerabilities: Approaches and Challenges. Information and Software Technology, 74, 160–180. https://doi.org/10.1016/j.infsof.2016.02.005
[9] Divyaniyadav, Gupta, D., Singh, D., Kumar, D., & Sharma, U. (2018, December). Vulnerabilities and Security of Web Applications. 2018 4th International Conference on Computing Communication and Automation (ICCCA). https://doi.org/10.1109/CCAA.2018.8777558
[10] Hadabi, A., Elsamani, E., Abdallah, A., & Elhabob, R. An Efficient Model to Detect and Prevent SQL Injection Attack.
[11] Halfond, W. G. J., Viegas, J., & Orso, A. (2006). A Classification of SQL Injection Attacks and Countermeasures.
[12] Hashim, A., Medani, R., & Attia, T. A. (2021, February 26). Defences against Web Application Attacks and Detecting Phishing Links Using Machine Learning. Proceedings of the 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE 2020). https://doi.org/10.1109/ICCCEEE49695.2021.9429609
[13] Hasan, M., Balbahaith, Z., & Tarique, M. (2019, November). Detection of SQL Injection Attacks: A Machine Learning Approach. In 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA) (pp. 1–6). IEEE.

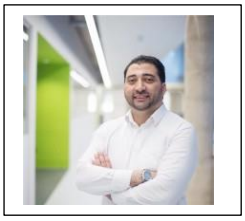
[14] Hassan, M. M., Nipa, S. S., Akter, M., Haque, R., Deepa, F. N., Rahman, M., ... & Sharif, M. H. (2018). Broken Authentication and Session Management Vulnerability: A Case Study of Web Application. International Journal of Simulation Systems, Science & Technology, 19(2), 6–1.
[15] Harefa, J., Prajena, G., Alexander, A., Muhamad, A., Dewa, E., & Yuliandry, S. (2021). SEA WAF: The Prevention of SQL Injection Attacks on Web Applications. Advances in Science, Technology and Engineering Systems Journal, 6(2), 405–411. https://doi.org/10.25046/aj060247
[16] Hernes, M., Rot, A., & Jelonek, D. (n.d.). Studies in Computational Intelligence 887: Towards Industry 4.0 – Current Challenges in Information Systems. http://www.springer.com/series/7092
[17] Jemal, I., Cheikhrouhou, O., Hamam, H., & Mahfoudhi, A. (2020). SQL Injection Attack Detection and Prevention Techniques Using Machine Learning. International Journal of Applied Engineering Research, 15(6). http://www.ripublication.com
[18] Kar, D., Panigrahi, S., & Sundararajan, S. (2016). SQLiGoT: Detecting SQL Injection Attacks Using Graph of Tokens and SVM. Computers and Security, 60, 206–225. https://doi.org/10.1016/j.cose.2016.04.005
[19] Kareem, F. Q., Ameen, S. Y., Salih, A. A., Ahmed, D. M., Kak, S. F., Yasin, H. M., Ibrahim, I. M., Ahmed, A. M., Rashid, Z. N., & Omar, N. (2021). SQL Injection Attacks Prevention System Technology: Review. Asian Journal of Research in Computer Science, 13–32. https://doi.org/10.9734/ajrcos/2021/v10i330242
[20] Khera, Y., Kumar, D., Sujay, S., & Garg, N. (2019). Analysis and Impact of Vulnerability Assessment and Penetration Testing. Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects (COMITCon 2019), 525–530. https://doi.org/10.1109/COMITCon.2019.8862224
[21] Kubota, K., Oo, W. K. K., & Koide, H. (2020). A New Feature to Secure Web Applications. Proceedings of the 2020 8th International Symposium on Computing and Networking Workshops (CANDARW 2020), 334–340. https://doi.org/10.1109/CANDARW51189.2020.00071
[22] Kumar, Y., Satyanarayana, A. S., Kumar, A., & Sharma, V. (2021). Risks and Threats to Web Applications and Their Preventions: A Theoretical Study on Vital Risks and Threats. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 432–438. https://doi.org/10.32628/cseit217281
[23] Kumar, S., Amrita, B. J., Vidyapeetham, V., Santhosh Kumar, B. J., & Anaswara, P. P. (2018). Vulnerability detection and prevention of SQL injection. International Journal of Engineering & Technology, 7(2). https://www.researchgate.net/publication/346624353
[24] Nadeem, R. M., Saleem, R. M., Bashir, R., & Habib, S. (2017). Detection and prevention of SQL injection attack by dynamic analyzer and testing model. International Journal of Advanced Computer Science and Applications, 8(8), 209–214.
[25] Qbea'h, M., Alshraideh, M., & Sabri, K. E. (2016, August). Detecting and preventing SQL injection attacks: A formal approach. In 2016 Cybersecurity and Cyberforensics Conference (CCC) (pp. 123–129). IEEE.
[26] Raut, S., Nikhare, A., Punde, Y., Manerao, S., & Choudhary, S. (2019). A review on methods for prevention of SQL injection attack. International Journal of Scientific Research in Science and Technology, 463–470. https://doi.org/10.32628/ijsrst196258
[27] Sadqi, Y., & Maleh, Y. (2022). A systematic review and taxonomy of web applications threats. Information Security Journal, 31(1), 1–27. https://doi.org/10.1080/19393555.2020.1853855
[28] Singh, A., Sharma, A., Sharma, N., Kaushik, I., & Bhushan, B. (2019). Taxonomy of attacks on web-based applications. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT). https://doi.org/10.1109/icicict46008.2019.8993264
[29] Shobana, R., & Suriakala, M. (2021). Bypassing two-factor authentication based on classification using Aho-Corasick matching algorithm for NoSQL databases. Turkish Journal of Computer and Mathematics Education, 12(10).
[30] Shinde, P. S., & Ardhapurkar, S. B. (2016, February). Cyber security analysis using vulnerability assessment and penetration testing. In 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave) (pp. 1–5). IEEE.
[31] Stuttard, D., & Pinto, M. (n.d.-a). The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws.
[32] Su, Z. C., Hlaing, S., & Khaing, M. (n.d.). A detection and prevention technique on SQL injection attacks.
[33] Torkaman, A., Bahrololum, M., Tadayon, M. H., Atashzar, H., & Tadayon, M. H. (2011). A survey on web application vulnerabilities and countermeasures. http://www.soumu.go.jp
[34] Thombare, B., & Soni, D. (2022). Prevention of SQL injection attack by using black box testing. 23rd International Conference on Distributed Computing and Networking. https://doi.org/10.1145/3491003.3493233
[35] Varol, A., Karabatak, M., Varol, C., Fırat Üniversitesi, Institute of Electrical and Electronics Engineers. Turkey Section, & Institute of Electrical and Electronics Engineers. (n.d.). 6th International Symposium on Digital Forensic and Security: Proceeding Book, 22–25 March 2018, Antalya, Turkey.
[36] Zhu, A., & Yan, W. Q. (2017). Exploring defense of SQL injection attack in penetration testing. International Journal of Digital Crime and Forensics, 9(4), 62–71. https://doi.org/10.4018/IJDCF.2017100106
[37] Zuech, R., Hancock, J., & Khoshgoftaar, T. M. (2021, July). Detecting SQL injection web attacks using ensemble learners and data sampling. In 2021 IEEE International Conference on Cyber Security and Resilience (CSR) (pp. 27–34). IEEE.
[38] Ali, A., Almaiah, M. A., Hajjej, F., Pasha, M. F., Fang, O. H., Khan, R., ... & Zakarya, M. (2022). An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network. Sensors, 22(2), 572.
[39] Al Nafea, R., & Almaiah, M. A. (2021, July). Cyber security threats in cloud: Literature review. In 2021 international conference on information technology (ICIT) (pp. 779-786). IEEE.
[40] Almaiah, M. A., Hajjej, F., Ali, A., Pasha, M. F., & Almomani, O. (2022). A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS. Sensors, 22(4), 1448.

[41] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2022). Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions. Electronics, 11(20), 3330.
[42] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms. Sensors, 24(2), 713.

Biographies



Bashaer Almelehy received a master degree in Cybersecurity from King Faisal University. He has an excellent experience in the cybersecurity field in both theoretical and practical. He has several certificates in cybersecurity like CEH and others. He several publications in cyber risk assessment. His research interests including cyber security, risk assessment and cyber-attacks. 221445338.student@kfu.edu.sa



Mohammad Ahmad currently studies PhD of Computer Science at the University of Bedfordshire in the Institute for Research in Applicable Computing. Mohammad is working on a research project in the field of Blockchain, Cybersecurity, and Healthcare. Mohammad does research in Software Engineering, Information Systems, Blockchain Technology, and Explainable AI. mohammad.ahmad@study.beds.ac.uk

Dr. Ghalia Nassreddine is an Associate Professor at Rafik Hariri University, Lebanon. She earned her Ph.D. in Information Technology and Systems from the University of Technology of Compiègne, France, in 2009. Dr. Nassreddine has taught a wide range of Computer Science courses at several universities across Lebanon. She has also conducted numerous training workshops for national and international organizations, particularly in the fields of Machine Learning and Data Science. Her research interests focus on machine learning and its interdisciplinary applications in renewable energy systems, business analytics, and healthcare. Dr. Nassreddine has an extensive publication record that includes peer-reviewed journal articles, conference proceedings, and book chapters. Email: nassreddinega@rhu.edu.lb

Dr. Mohammed Maayah Maayah is among the top 2% scientists in the world from 2020 up to now. He is working as Editor in Chief for the International Journal of Cybersecurity and Risk Assessment. He has published over 115 research papers in highly reputed journals such as the Engineering and Science Technology, an International Journal, Education and Information Technologies, IEEE Access and others. Most of his publications were indexed under the ISI Web of Science and Scopus. His current research interests include Cybersecurity, Cybersecurity-Risk Assessment and Blockchain. m.almaiah@ju.edu.jo

Aparna Achanta is a Security Architect and Leader at IBM Consulting with extensive experience driving mission-critical cybersecurity initiatives, particularly in federal agencies. She successfully implemented cybersecurity frameworks like Zero Trust and Security by Design for Federal Clients, strengthening the security posture and enhancing data protection and security standards across cloud applications. Her leadership resulted in the establishment of security review boards, secure development practices, vendor evaluations, threat modeling, vulnerability management, code scanning, observability, and performance monitoring to ensure that enterprise comply with stringent federal guidelines. aparnaachanta@ieee.org