IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

Web Security Issues Analysis and Practical Ways to Prevent Them

Praveen kumar[1], Manjeet kaur[2]

School of Computational Sciences, GNA University, Phagwara , India

Abstract: Web applications have become an important part of our daily life and are widely used in communication, education, banking, shopping and business services. The use of web applications is increasing and also the security related problems are increasing rapidly. A lot of applications are vulnerable to SQL Injection, Cross-Site Scripting (XSS), weak authentication mechanism and insecure configuration settings. Such vulnerabilities can lead to unauthorised access, theft of sensitive information, or disruption of system operations, with serious financial and privacy-related consequences.

The main objective of this research is to study the security challenges present in modern web applications and to understand the impact of these vulnerabilities on data protection and system reliability. For this purpose, we shall utilise a real-world dataset collected from Kaggle for analysis. The aim of the study is to determine the frequency and distribution of different vulnerabilities and to look for the reasons of their appearance. It tries to find out which security problems are most frequent in development and maintenance of web applications [1].

The analysis of the dataset will allow the research to identify the areas most in need of security improvements and the factors that contribute to these weaknesses. The results will be used to recommend effective preventive measures and security practices for the reduction of risks and the improvement of the overall security of web applications [1].

Keywords: Web Application Security, SQL Injection, Cross-Site Scripting (XSS), Authentication, Cybersecurity, Vulnerability Analysis, Data Protection, Secure Web Development, Kaggle Dataset, Cyber Security

I. INTRODUCTION

Web applications are an inseparable part of the current-day digital world and support services such as online banking, e-commerce, healthcare, education as well as communication platforms. Organisations are increasingly relying on internet-based applications, leading to a vast amount of sensitive and personal information being stored and processed online. As a result, web applications are considered one of the main targets of cyber-attacks. These applications can have vulnerabilities that can lead to serious issues like data breaches, financial losses, and unauthorised access to systems. According to the OWASP Top 10 report [1], there are still many common security flaws in web applications such as broken access control, injection attacks, and insecure configuration settings.

SQL Injection and Cross Site Scripting (XSS) are known as some of the most dangerous and occurring vulnerabilities among the different types of threats. SQL Injection attacks happen when the attackers inject malicious commands in the database queries, granting them access, modification or deletion of sensitive data without permission [2]. XSS attacks involve injecting malicious scripts into web pages which can result in session hijacking, cookie theft, or unauthorised actions by the user [3]. These vulnerabilities are mostly due to poor input validation, insecure coding and weak authentication mechanisms. Several studies have shown that these vulnerabilities are still present in a large number of web applications, which means that the security measures are either insufficient or not properly implemented during the development [4].

To deal with these challenges, researchers and security professionals have come up with methods such as penetration testing, automated vulnerability scanning and secure coding frameworks. Vulnerabilities are often detected by tools such as OWASP ZAP and Burp Suite that simulate real-world attack scenarios [5]. Additionally, it has been proven that integrating security practices into the Software Development Life Cycle (SDLC) is an effective way to reduce vulnerabilities in the early stages of application development [6]. However, most of the existing solutions are theoretical or applicable to a specific environment, which are hard to be directly applied in the practical development projects.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155257

In this research, we study web application security issues by adopting a data-driven approach, by studying a real-world dataset collected from Kaggle. The dataset allows to identify patterns, frequency and severity of different vulnerabilities, leading to a better understanding on how security problems happen in practice. This data is analysed to identify major risk areas and to understand the impact of vulnerabilities on system performance and reliability. Previous studies have demonstrated that analysing real-world datasets can enhance the detection and prevention of web security threats [7].

The results of the study suggest practical security measures and preventive strategies that can be used in the course of web application development. The main goal of this study is to bridge the gap between theoretical concepts and their practical implementation by combining data analysis with effective security practices. This approach aims to help developers to build more secure, reliable and efficient web applications, reducing the risk of cyber threats in modern digital systems [8].

II. PROBLEM STATEMENT

Web applications play a crucial role in modern digital systems and are employed in various fields such as banking, healthcare, education, and online shopping. However, these applications are highly susceptible to security threats, such as SQL Injection, Cross-Site Scripting (XSS), and authentication-related attacks, leading to data breaches, unauthorised access, and system compromise. While there are many different security tools and techniques, many previous studies have primarily focused on theoretical analysis or limited attack scenarios and have not offered a comprehensive understanding of real-world security challenges. In addition, there are a few studies on the practical application of different security tools for detection and prevention of different types of attacks. Without data-driven insights, developers and security professionals are left guessing which vulnerabilities are critical, what risks to prioritise, and which security measures are most effective. Hence, a detailed data-driven study of web application security issues is required to better understand the vulnerability patterns, severity levels, and the practical use of security tools in real-world environments, to help improve the security and reliability of web applications.

III. LITERATURE REVIEW

The fast escalation of cyber threats to online systems has made web application security a crucial area of research. Today's web applications deal with large amount of user data and keep communicating with backend databases. So they are very vulnerable to different types of attacks. Many studies have shown that vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS) and broken authentication still remain the most common security threats affecting web applications of various platforms [1]. The researchers have found that these vulnerabilities mainly exist due to weak input validation, insecure coding method and poor security implementation that enable attackers to exploit the weakness of a system and gain unauthorised access to sensitive information [2].

Many research studies are centred on SQL Injection and XSS attacks as they are very prevalent and have a serious impact on web applications. Both vulnerabilities are mostly caused by improper handling and sanitisation of user input, studies show [3]. SQL Injection attacks allow the attacker to change database queries and read confidential information. XSS attacks inject malicious scripts into web pages to hijack user sessions and steal data [4]. These attacks are constantly identified by security organisations and researchers as serious threats in web application security [5] due to their high occurrence. Experimental studies also indicate that many websites are still vulnerable to such attacks, in spite of several prevention methods and security practices [6].

To mitigate these risks, researchers have proposed various techniques including penetration testing, automated vulnerability scanning and intelligent detection methods. Penetration testing is commonly used to identify vulnerabilities by simulating real-world attack scenarios, especially for finding SQL Injection problems in web applications [7]. Continuous security testing [8] also commonly uses automated tools like OWASP ZAP to find vulnerabilities like XSS, insecure configurations, and clickjacking. However, studies also show some limitations of these tools including false positive results and difficulties in detecting complex logic-based vulnerabilities [9].

More recent research in web application security has increasingly focused on data-driven and intelligent approaches to detecting and preventing vulnerabilities. Machine learning and deep learning based techniques have been introduced to improve the detection accuracy of attacks such as SQL Injection and XSS, while reducing false positives and supporting real-time threat detection [10]. The researchers also stress the need of integrating security practices into the Software Development Life Cycle (SDLC) to identify and stop vulnerabilities in the early phases of development [11].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

Moreover, the combination of different security techniques like static analysis, dynamic analysis and runtime protection has been considered as an effective approach to improve the overall security of web applications [12].

Comparison between Existing Research and Proposed Approach		
Feature / Criteria	**Existing Research Papers**	**Proposed Work (Your Research)**
Approach Type	Mostly theoretical or limited experimental	Data-driven + practical implementation
Dataset Usage	Limited or synthetic datasets	Real-world Kaggle dataset
Focus Area	Specific vulnerabilities (SQLi, XSS only)	Multiple vulnerabilities (SQLi, XSS, Auth, Config)
Analysis Method	Conceptual or tool-based	Statistical + pattern-based analysis
Real-Time Analysis	Limited or not included	Included (based on dataset patterns)
Security Tools Usage	Tools like OWASP ZAP, Burp Suite	Dataset analysis + practical security techniques
Development Perspective	Security-focused (less developer guidance)	Developer-friendly solutions (coding + validation)
Prevention Techniques	General recommendations	Practical and implementable strategies
SDLC Integration	Partially discussed	Clearly integrated into development lifecycle
Accuracy of Results	Depends on tools and assumptions	Based on real-world data insights
Limitation Handling	Limited discussion	Addresses real-world gaps using dataset analysis
Practical Implementation	Low	High

IV. RELATED WORK

The research on web application security has become more important as a result of the rapid increase of cyber-attacks on online systems. Previous work mainly focused on the discovery of common vulnerabilities like SQL injection, Cross-site Scripting (XSS) and authentication-related vulnerabilities, which still pose a major threat in today's web applications. Web applications, which are constantly processing user inputs and communicating with backend databases, are very vulnerable to unauthorised access and malicious attacks [1]. These vulnerabilities are caused by insecure coding practices, poor validation methods and security implementation during development of applications [2] the researchers have found.

To mitigate these security risks, researchers have proposed several methods to detect vulnerability such as penetration testing and automated security scanning tools. Penetration testing is considered a successful technique because it mimics real-world attack scenarios to identify vulnerabilities like SQL Injection and XSS vulnerabilities [3]. They are also used heavily as vulnerability scanners to speed up and broaden the scope of detection by finding security holes in web applications. However, these tools are known to sometimes generate false positives or miss complex logic-based vulnerabilities [4]. Furthermore, the integration of multiple approaches (e.g., static analysis and dynamic analysis) can enhance the overall accuracy and efficiency of vulnerability detection research [5].

In the recent studies, the security practices have been injected into the Software Development Life Cycle (SDLC) to identify and mitigate the vulnerabilities at the early design phase. According to the researchers, many security problems occur because traditional development methods often involve less focus on security measures [6]. Current methods ensure continuous monitoring, real-time threat detection and active security measures to defend web applications against the evolving cyber threats [7]. In addition, modern techniques such as machine learning and data-driven analysis are increasingly being used to analyse large datasets, to identify attack patterns and to enhance vulnerability detection [8]. However, there are still a number of challenges such as limited availability of real-world datasets, lack of practical implementation, and difficulties in effectively integrating security solutions into development workflows.

V. METHODOLOGY

The research methodology is designed to study web application security problems in a clear, organised and practical manner. The research is performed in a step-by-step procedure comprising: data collection, data preprocessing, vulnerability analysis, and identification of suitable prevention strategies. This structured approach helps to better

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

understand real-world security threats and to ensure that the research findings are based on real data and practical observations, and not only on theoretical concepts. The study tries to identify the common security problems, their reasons, and how they impact web applications by looking at real-world vulnerability information.

The research methodology is based on commonly used security assessment techniques such as vulnerability analysis, penetration testing and integration of security practices in to Software Development Life Cycle(SDLC). These techniques are widely used by researchers and security experts to identify, assess and mitigate security risks in web applications. Moreover, data-driven analysis application helps in identifying significant patterns, trends, and high-risk vulnerabilities from the dataset. This helps to understand the most important security issues and to develop more efficient and practical security solutions for modern web applications.

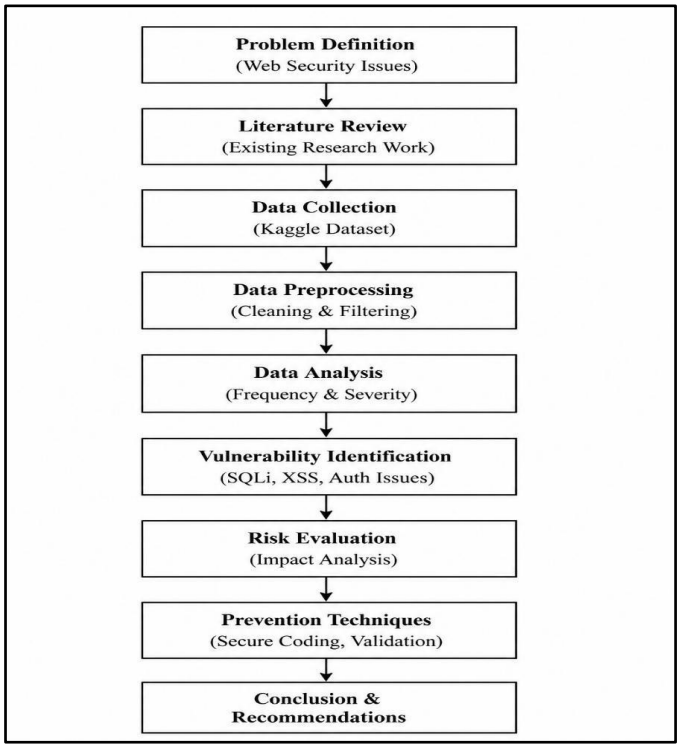

Fig. 1: Block Diagram of Web Application Security Analysis Process

VI. DATA COLLECTION

The data collection process is an important part of this research because it provides the basic information needed to analyze web application security issues. For this study, a real-world dataset was collected from Kaggle, a popular platform commonly used for data science and cybersecurity research. The dataset contains information about different types of web application vulnerabilities, including SQL Injection, Cross-Site Scripting (XSS), and authentication-related issues, along with details such as severity levels and possible impacts. Real world data is very important . Web applications are often riddled with vulnerabilities due to coding errors , insecure configurations , and weak security practices . Attackers use these vulnerabilities to gain unauthorised access or steal sensitive information .

The dataset was examined before the analysis to verify that the information was accurate, complete and relevant to the research objectives. The dataset was preprocessed using methods such as removing incomplete records, ordering relevant attributes, and cleaning irrelevant information before analysis. However, the analysis of real vulnerability data is more fruitful in understanding the patterns, frequency and impact of security threats, that results in more accurate results and better decision making in web security research [2]. This practical and data-driven approach increases the reliability of the research findings and their usefulness for the real-world development of web applications and their security improvement.

Dataset Link :
https://www.kaggle.com/datasets/tannubarot/cybersecurity-attack-and-defence-dataset?utm

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

VII. DATA PROCESSING

Data cleaning and data filtering are required steps to prepare the dataset for accurate and reliable analysis. The Kaggle dataset was carefully processed in this study to remove errors, inconsistencies and unnecessary information that could affect the quality of the results. Data cleaning is the process of correcting formatting errors, removing duplicate records, treating missing values, and removing noisy or invalid data in order to ensure that the dataset is more accurate, complete, and consistent for subsequent analysis. Filtering techniques are used to exclude irrelevant or unnecessary attributes, and only information relating to web application security vulnerabilities such as attack type, severity level, and impact are extracted after data cleansing.

The dataset has been cleaned and filtered, but it is also organised and structured into a suitable format to improve the efficiency of analysis. The proper preprocessing helps to spot important patterns and trends that are related to vulnerabilities such as SQL Injection and Cross-Site Scripting (XSS). These are important steps because high-quality and well-structured data leads to more accurate analysis, reliable results, and better decision-making in web application security. The use of a well-prepared dataset makes the research more practical and useful to understand and reduce the security risks in modern web applications.

VIII. DATA ANALYSIS

Data analysis is an important part of this research as it helps to transform the processed dataset to useful information on web application security vulnerabilities. This research uses various analytical methods to analyse the frequency, distribution, and severity of vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and authentication-related vulnerabilities. The analysis aims to identify patterns and trends in the dataset to understand which vulnerabilities are most prevalent and how they impact system performance, data security, and the overall reliability of the application. Statistical methods and visualisation techniques are also applied to analyse the relationship between factors like attack type, severity level, and affected system components. This approach helps identify high-risk vulnerabilities, understand their impact, and support early threat detection and better security decision-making. The analysis results can help the developers and security professionals to prioritise critical risks and implement better security measures to improve the security of modern web applications against cyber threats.

IX. VULNERABILITY IDENTIFICATIONS

In this research, we take a major step towards vulnerability identification by analysing the dataset to identify and classify major web application security threats such as SQL Injection (SQLi), Cross-Site Scripting (XSS) and authentication related vulnerabilities. SQL Injection attacks occur when a malicious input is supplied by an attacker into a database query, which permits the attacker to access, modify, or delete sensitive information. XSS vulnerabilities are created when user inputs are not properly validated allowing attackers to inject malicious scripts into web pages leading to session hijacking, data theft or unauthorised actions within a user's browser. Patterns like weak password management, insecure session handling, and poor login security mechanisms can identify authentication related issues where unauthorised users may access protected systems. The analysis utilises important dataset features such as attack type, payload patterns and severity levels to classify vulnerabilities and assess their possible impact in web applications. This process is useful to identify critical security threats and provide a basis for risk assessment and development of effective prevention strategies to improve web application security.

X. RISK EVALUATION

Risk evaluation is an important part of this research as it helps to know how the identified web application vulnerabilities can impact systems, data and users. In this phase, vulnerabilities like SQL Injection, Cross-Site Scripting (XSS), and authentication-related issues are analysed based on their severity and possible impact. SQL Injection is a high risk vulnerability that allows attackers to steal, manipulate or remove critical information from databases, leading to data breaches and complete system takeover. Likewise, session hijacking, identity theft and actions performed without the user's consent are some of the consequences of XSS attacks. Authentication vulnerabilities allow attackers to bypass the login system and gain unauthorised access to protected resources, increasing the security risks.

The impact analysis is carried out by analysing parameters like data confidentiality, integrity of the system and availability of service. Web application vulnerabilities can result in serious issues like loss of data, financial losses,

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

reputational damage and disruption of online services. Vulnerabilities are then classified into high, medium and low risk levels according to their impacts and probability of occurrence, so as to identify the most critical threats that need to be paid attention to immediately. This risk assessment process offers a more precise comprehension of the seriousness of web security problems and helps in the development of effective countermeasures to alleviate potential risks and improve the overall security of web applications.

XI. PREVENTION TECHNIQUES

Prevention techniques are essential to safeguard web applications from vulnerabilities like SQL Injection, Cross-Site Scripting (XSS) and attacks related to authentication. By adopting secure coding practices, developers can create more secure applications by implementing techniques such as parameterised queries, appropriate error handling, and encrypting sensitive data. Input validation is also very important because all the user inputs need to be validated and sanitised on both client side and server side so that malicious data does not enter into the system. Strong authentication methods such as Multi-Factor Authentication (MFA), secure password policies and proper session management also prevent unauthorised access. Moreover, embedding security in the Software Development Life Cycle (SDLC) allows for early detection and remediation of vulnerabilities, leading to enhanced reliability of the system and minimised security risks.

Common techniques to prevent vulnerabilities in web applications are :-

- Guard against SQL Injection attacks by using parameterised queries and prepared statements.
- Proper input validation and data sanitisation to filter out malicious inputs
- Use of strong authentication and authorisation procedures
- SSL/TLS protocols for encrypting sensitive data.
- Regular security testing with static and dynamic analysis tools
- Filtering out malicious traffic using Web Application Firewalls (WAF)
- Regular software updates and patching for known vulnerabilities
- Adhering to secure coding standards and security best practices such as OWASP

These prevention methods offer a layered security approach to help mitigate cyber threats and help to develop secure, reliable and efficient web applications.

XII. DATA ANALYSIS & RESULT

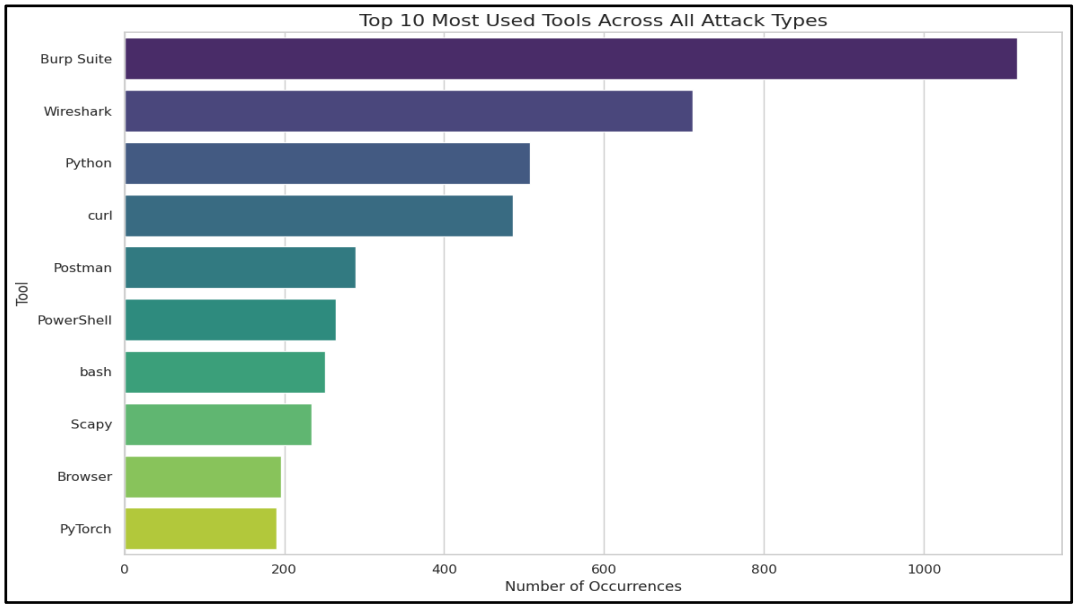

Figure 2: Distribution of top 10 most used tools for different types of web-based attacks, with Burp Suite being the most dominant tool, followed by Wireshark and Python.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

Today's digital systems are incomplete without web applications. However, web applications are still prone to serious security issues such as SQL Injection, Cross-Site Scripting (XSS), and authentication weaknesses. Such issues can lead to serious consequences including data breaches, unauthorised access, and system compromise.

There are many security tools and techniques to detect and prevent these vulnerabilities but many of the existing techniques are theoretical or limited to specific attack types. Also, the lack of thorough analysis with real-world datasets that can give more insights into the frequency, severity and trends of web security threats. Another significant limitation is the lack of understanding of the use of different tools among different attack types. This makes it hard for developers and security professionals to prioritise risks and adopt effective defence strategies.

Thus, a data-driven approach is necessary to analyse the real-world vulnerability data to identify the key security issues, understand attack patterns and to propose practical prevention techniques.

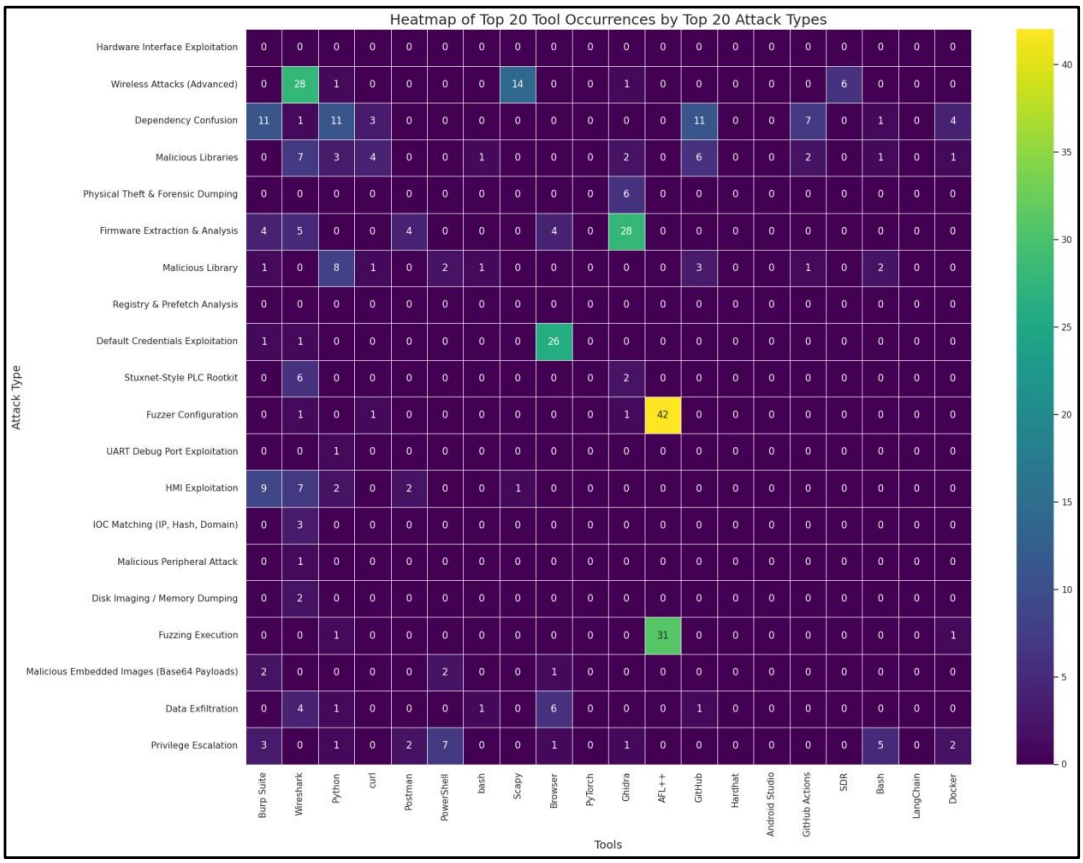

Figure 3: Heatmap representing the frequency of tool usage across different web attack types, highlighting the relationship between specific tools and targeted attack categories

The analysis shows that some security tools are mostly related to certain cyber-attacks. AFL++ is often seen in categories related to Fuzzer Configuration and Fuzzing Execution, which shows its substantial involvement in fuzz testing and vulnerability detection. Similarly, Wireshark is closely related to sophisticated wireless attacks, demonstrating its value in network traffic surveillance and packet examination. These observations suggest that some tools are designed to perform specific security testing tasks more efficiently.

The results also show that tools such as Burp Suite and Python are used across multiple attack categories, which reflects their flexibility and widespread use in web vulnerability testing, automation and scripting-based attacks. In particular, some attack categories (e.g., Hardware Interface Exploitation, Registry & Prefetch Analysis) exhibit very low or no tool usage in the dataset, which may be due to a lower occurrence rate or limited coverage in the collected

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

data. Moreover, the Browser tool is closely associated with Default Credentials Exploitation, highlighting the importance of web interfaces in these types of attacks.

The heatmap analysis also shows that the usage of tools is not evenly distributed across all attack types. Some tools are highly concentrated in certain categories, i.e. they are specialised in certain attack scenarios, not used universally. This trend shows how important it is to select the right tools according to the kind of vulnerability being examined. The overall findings indicate that the effectiveness of vulnerability assessment is highly reliant on the selection of appropriate security tools for different types of web application attacks and security testing requirements.

XIII. RESULT AND ANALYSIS

The Random Forest classification model showed good performance in predicting the experimental dataset with an overall accuracy of 95.83%. The weighted average precision, recall and F1-score obtained, respectively, of 0.96, 0.96 and 0.95 indicate a very good classification performance for the majority of the classes. Analysis of the confusion matrix reveals that the majority of the predictions are concentrated on the diagonal elements. This means that the classifier correctly classified a large number of instances and misclassified very few instances. The classes with higher sample distributions achieved excellent precision and recall values close to 1.00 indicating the robustness and stability of the Random Forest algorithm in multiclass classification problems. However, the macro average precision, recall and F1 score were relatively low (0.78, 0.70 and 0.72 respectively) due to the severe imbalance of the classes in the dataset. The model was not able to learn the minority classes efficiently as many of the minority classes had very few samples and showed low or zero precision and recall values. But the Random Forest model generalises well over the dataset and performs reliably for most of the classes. The experimental results demonstrate that Random Forest is an effective and accurate machine learning technique for multiclass classification tasks but there is potential to improve performance on rare classes using dataset balancing techniques, minority class augmentation and hyperparameter optimisation methods.

Accuracy	Macro Avg Precision	Macro Avg Recall	Macro Avg F1-Score	Weighted Avg Precision	Weighted Avg Recall	Weighted Avg F1-Score
95.83%	0.78	0.70	0.72	0.96	0.96	0.95

Table 2: Performance matrix

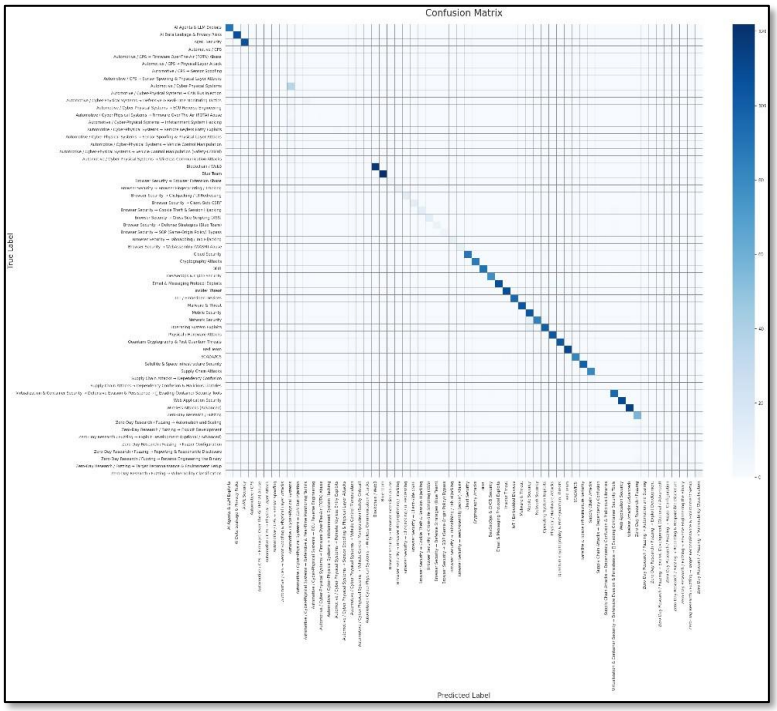

Figure 4: Confusion Matrix of the Random Forest Classification Model showing the relationship between actual and predicted cybersecurity attack categories.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155257

The confusion matrix shows the accuracy of the Random Forest model for classifying multiple classes of cyber security attacks. Most of the values are concentrated in the diagonal region which means most of the instances were classified correctly. There are only a few off-diagonal values, which means that there are some minor misclassifications between some classes. The darker blue colour in the diagonal indicates the high prediction accuracy and good classification capability of the proposed model.

XIV. CONCLUSION

In conclusion, the study has demonstrated that web applications are vulnerable to different security threats such as SQL injection, Cross-Site Scripting (XSS) and authentication problems, which can affect the confidentiality of information, the integrity of the system and the availability of services. The research finds common vulnerability patterns, their frequency and severity levels to provide a clear understanding of real-world security risks with the use of a real-world dataset from Kaggle. Results shows that most of the vulnerabilities are due to bad coding practices, lack of input validation and weak authentication mechanisms. This highlights that Web security is still a big concern in today's development environments and organisations need to take proactive steps to mitigate risks and protect user data.

The analysis yields a number of recommendations for the improvement of web application security. All user inputs should be validated by developers and secure coding practice should be applied to prevent injection attacks and illegal data entry. Implement strong authentication and authorisation controls (e.g. multi-factor authentication and secure session management) to prevent unauthorised access. Organisations should also embed security throughout the Software Development Lifecycle (SDLC) and conduct regular security testing, including static analysis, dynamic analysis and penetration testing to identify vulnerabilities early in the development process. Moreover, to ensure protection against cyber threats, encryption protocols, web application firewalls (WAF) and regular updating and patching are necessary. Using globally accepted standards like OWASP Top 10 can also help in improving security practices as a whole. Following the recommended design patterns and security best practices will help developers to create secure, reliable and resilient web application with minimum risk of cyber-attack.

XV. FUTURE WORKS

Future work in web application security should aim to develop intelligent, scalable and automated solutions to offset the changing nature of cyber threats. Traditional security techniques become less effective as web applications get more complex. There is a need for advanced approaches like Artificial Intelligence (AI) and machine learning to detect and prevent vulnerabilities in real time. Studies show AI-based systems can improve threat detection by sifting through vast amounts of data to identify unusual patterns that traditional defences may miss. Future work may also focus on the integration of security in the DevSecOps practices to ensure continuous monitoring and early detection of vulnerabilities throughout the development lifecycle. Moreover, it is important to work on larger and diverse real-world datasets for the improvement of the accuracy of the vulnerability analysis and prediction models. Emerging areas such as API Security, Cloud-based Security Models, and Automated Penetration Testing tools should also be considered to address modern attack vectors. Moreover, explainable and robust AI models can contribute to the transparency and trust of automated security systems. Future work should address the development of adaptive and real-time security solutions capable of responding to dynamic and increasingly sophisticated cyber threats.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my guide, **Ms. Manjeet Kaur**, for providing valuable guidance, encouragement, and continuous support throughout the preparation of this research paper. Their suggestions and technical knowledge greatly helped in improving the quality of this work.
I am also thankful to the faculty members of **School of Computational Sciences, GNA University, Phagwara**, for their support and motivation during the completion of this paper.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155257

REFERENCES

[1]. OWASP Foundation, "OWASP Top 10: Web Application Security Risks," OWASP Foundation, 2021. [Online]. Available: https://owasp.org/www-project-top-ten/.
[2]. S. Disawal and U. Suman, "An Approach to Detect and Prevent SQL Injection and XSS Vulnerability in Web Application," International Journal of Engineering Trends and Technology, vol. 71, no. 8, pp. 216–224, 2023. [Online]. Available: https://ijettjournal.org/.
[3]. Paul, R. Singh, and M. Kumar, "SQL Injection Attack: Detection, Prioritization and Prevention," Journal of Information Security and Applications, vol. 78, 2024. [Online]. Available: https://www.sciencedirect.com/journal/journal-of-information-security-and-applications.
[4]. S. P. Maniraj et al., "Securing Web Applications with OWASP ZAP for Comprehensive Security Testing," 2024. [Online]. Available: https://www.zaproxy.org/.
[5]. PortSwigger, "Web Security Academy," PortSwigger Ltd., 2025. [Online]. Available: https://portswigger.net/web-security.
[6]. Y. Chen et al., "Research on SQL Injection Detection Technology Based on Deep Learning," Computers, Materials & Continua, vol. 84, no. 1, pp. 1145–1167, 2025. [Online]. Available: https://www.techscience.com/cmc.
[7]. M. K. Anuvarshini, "Evaluation and Enhancement of Web Application Firewall Performance Using OWASP CRS," Computers & Security, vol. 145, 2025. [Online]. Available: https://www.sciencedirect.com/journal/computers-and-security.
[8]. C. He et al., "SecRASP: Next Generation Web Application Security Protection Framework," Computers & Security, vol. 146, 2025. [Online]. Available: https://www.sciencedirect.com/journal/computers-and-security.
[9]. R. Bakır et al., "A Novel Approach to Detect XSS and SQL Injection Attacks Using Machine Learning," Neural Computing and Applications, vol. 37, 2025. [Online]. Available: https://link.springer.com/journal/521.
[10]. T. R. Saputra et al., "Web Application Security Testing Against SQL Injection Attacks Using SQLMap," Jurnal E-Komtek, vol. 9, no. 2, pp. 434–439, 2025. [Online]. Available: https://sqlmap.org/.
[11]. Ayoubi, "Improved Detection of Cross-Site Scripting (XSS) Attacks," Security and Communication Networks, vol. 2026, 2026. [Online]. Available: https://onlinelibrary.wiley.com/journal/19390122.
[12]. J. Kettle, "Top 10 Web Hacking Techniques of 2025," PortSwigger Research, 2026. [Online]. Available: https://portswigger.net/research.
[13]. S. Neupane, "Detecting and Mitigating SQL Injection Vulnerabilities in Web Applications," arXiv preprint, 2025. [Online]. Available: https://arxiv.org/.
[14]. V. Babaey and A. Ravindran, "GenXSS: Framework for Automated Detection of XSS Attacks," arXiv preprint, 2025. [Online]. Available: https://arxiv.org/.
[15]. U. S. Potti et al., "Security Testing Framework for Web Applications Using OWASP Benchmark," arXiv preprint, 2025. [Online]. Available: https://owasp.org/www-project-benchmark/.
[16]. OWASP Foundation, "OWASP Web Security Testing Guide," OWASP Foundation, 2025. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide/.

