International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 267-272, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE43
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

Enhancement of Web Application Security using SQLMap and Machine Learning

[1]M. Fathima Begum, [2]Lekha Sree C, [3]Manasa P

[1,2,3]Computer Science & Engineering (Cyber Security), Madanapalle Institute of Technology & Science, Madanapalle, India
E-mail: [1]fathimabegum.it@gmail.com, [2]lekhasreelekha443@gmail.com, [3]manasaparampogu123@gmail.com

Abstract **- SQL Injection (SQLi) is a critical vulnerability that allows attackers to manipulate databases through malicious queries. To detect such vulnerabilities in web applications, we integrated SQLMAP, a penetration testing tool, with a Random Forest machine learning model. SQLMAP automates vulnerability detection, and its commands are further automated to enable users to perform tests using simple numerical inputs, improving usability and efficiency. Data collected through SQLMAP is analyzed by the Random Forest classifier, trained on labeled datasets of malicious and benign queries, to predict vulnerabilities with high accuracy. Automation streamlines the process, making penetration testing faster and accessible even to non-technical users. This scalable approach can be expanded to detect other vulnerabilities like Cross-Site Scripting or Remote Code Execution, providing an efficient and user-friendly solution that enhances web application security while contributing to broader cyber security advancements.**

Keywords: SQL Injection (SQLi), Machine Learning, SQLMAP, classifier, cyber security.

I. INTRODUCTION

The proliferation of web-based applications has revolutionized business operations by providing accessible and efficient tools that eliminate the need for traditional software installations. However, this growth has been accompanied by an increase in security threats. These attacks remain one of the most pervasive threats to web application security due to their simplicity and effectiveness, as evidenced by their continued prominence in open- and closed-source projects. For instance, recent reports indicate that 6.7% of vulnerabilities in open-source projects and 10% in closed-source projects are SQLi-related, with over 2,400 SQLi vulnerabilities expected to be identified in open-source projects by the end of 2024 [1].Traditional defenses against SQLi attacks, such as input validation, parameterized queries, and penetration testing, have proven insufficient against evolving attack strategies like blind and time-based injections. These conventional methods often struggle to adapt to new variations and require significant time and expertise. For example, Random Forests have shown robust performance in balancing precision and recall, making them particularly effective in detecting complex attack scenarios. This project builds on these advancements by employing an automated tool—SQLMap within Kali Linux—to streamline the detection of SQL injection vulnerabilities. SQLMap automates penetration testing processes by executing commands based on user inputs, reducing the complexity and time required for vulnerability assessments. A Random Forest model is then trained on this data to detect current threats and forecast potential future vulnerabilities [2]. In addition to leveraging ML for detection, advanced techniques like Natural Language Processing (NLP) are being integrated into cybersecurity frameworks to analyze query structures and differentiate between legitimate and malicious inputs. Such methods significantly reduce false positives and negatives while improving detection efficiency. Furthermore, predictive analytics combined with ML allows systems to anticipate vulnerabilities before they are exploited, enabling proactive security measures. Despite these advancements, challenges remain. ML models require extensive training on diverse datasets to achieve high accuracy across different attack scenarios. Issues such as computational overhead and reliance on specific datasets also pose limitations. Nonetheless, integrating ML into cybersecurity protocols offers significant advantages over traditional methods by providing adaptive and automated solutions capable of addressing the dynamic nature of cyber threats [14].Ethical considerations are paramount in implementing such technologies. This project ensures data confidentiality through anonymization and secure handling while adhering to licensing agreements for open-source tools like SQLMap [3]. Unauthorized probing or testing is strictly avoided, and safeguards are implemented to prevent misuse of the developed tools. By addressing these ethical concerns, the project not only enhances web application security but also sets a foundation for responsible research practices. In conclusion, machine learning represents a transformative approach to combating SQL injection attacks by automating detection processes and adapting to evolving attack strategies. While traditional methods still play a role in securing web applications, the integration of ML techniques offers a more robust defense against one of the most persistent cybersecurity

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 267-272, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE43
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

threats. This project underscores the importance of continuous innovation in cybersecurity research and highlights the potential of ML-based solutions in safeguarding digital infrastructures against SQL injection attacks.

II. LITERATURE REVIEW

2.1 Impact of SQL Injection on Web Security

SQL injection attacks can cause severe data breaches, leading to financial losses, reputational damage, and regulatory penalties. Mitigation strategies include the use of parameterized queries and web application firewalls. Additionally, fostering a culture of security awareness within organizations empowers employees to recognize and report suspicious activities, strengthening defenses against such threats. Machine learning monitors normal user activity to detect anomalies in real time[4].

2.2 Introduction to Automation in Web Application Penetration Testing

Automation in penetration testing streamlines the identification of system vulnerabilities by reducing human error and saving time. Automated tools perform repetitive tests efficiently, allowing testers to focus on other critical areas. This enhances the overall security posture by enabling faster and more accurate assessments.

2.3 Traditional vs. Automated Web Application Penetration Testing

Traditional penetration testing relies heavily on manual effort, making it time-consuming and prone to overlooking certain vulnerabilities. In contrast, automated testing delves deeper with minimal human intervention, facilitating continuous security assessments and quicker adaptation to emerging threats. Automated tools also generate detailed reports that help prioritize remediation efforts while providing advanced analytics for better threat understanding [5].

The project draws inspiration from the Social-Engineer Toolkit (SET), which simplifies complex attack vectors through a user-friendly interface. Similarly, this project aims to automate SQLMap commands with an intuitive design that allows users to perform SQL injection tests using simple numeric inputs. This approach enhances usability while reducing errors associated with manual testing, enabling faster and more efficient security assessments [15].

2.4 Introduction to Machine Learning in Cybersecurity

Machine learning revolutionizes cybersecurity by enabling real-time threat detection through pattern analysis and predictive analytics. By analyzing vast datasets for anomalies, machine learning systems continuously improve their accuracy over time. This fosters proactive threat management, allowing organizations to anticipate risks and mitigate them effectively while automating routine monitoring tasks[6].

2.5 Threat Detection

Machine learning models analyze historical data to identify patterns associated with cyberattacks before they occur. Predictive analytics strengthens defenses by providing insights into attackers' tactics, enabling organizations to implement effective preventive measures and foster continuous improvement in cybersecurity protocols.

2.6 Advanced Malware Detection

Traditional antivirus solutions rely on known signatures, leaving systems vulnerable to new malware variants. Machine learning-driven malware detection employs behavioral analysis to identify threats based on their actions rather than pre-existing patterns, ensuring robust protection against novel attacks[7].

2.7 Proposed Work

Existing SQL injection tools are often manual-intensive and inefficient at handling large datasets from penetration tests. While machine learning can enhance detection capabilities, its application in automating SQL injection detection remains limited. Few studies have explored using penetration test data for training machine learning models effectively. This project addresses these gaps by developing a Random Forest-based automation tool that utilizes test data for improved speed, accuracy, and efficiency in vulnerability detection.

III. DATA DESCRIPTION & METHODS

3.1 Tool Design and Architecture

The SQL Injection Automation Tool is designed with a modular architecture, making it highly maintainable and scalable. This modularity allows users to expand or modify the tool's components without affecting its core functionality. Python was chosen as the programming language due to its versatility and the availability of numerous libraries that support security operations. The tool comprises three main modules: the User Interface (UI), the Engine module, and the Reporting module. The UI is text-based, intuitive, and user-friendly, catering to both novice and experienced users. It is designed to work seamlessly in Linux environments, providing a simple interface for interacting with the tool. The

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 267-272, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE43
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

Engine module handles the execution of SQL injection operations and database interactions. It processes user inputs, formats them into SQL commands, presenting information on injection points, payloads used, and vulnerabilities discoveredinan organized format. Security is a top priority in the design of this tool. All incorporated codes are rigorously tested to ensure they are free from malicious elements or vulnerabilities [8]. This ensures that users can safely utilize the tool without compromising data integrity. The modular design also provides flexibility for users to customize or expand its functionality according to their specific needs. The proposed workflow architecture is explained in Fig.1.

3.2 Automation of SQL Injection Commands

The tool automates various SQL injection tasks by taking parameters such as target URLs and payloads. These inputs are processed by the Engine module, which generates SQL commands for injection. This automation simplifies penetration testing by eliminating manual effort while ensuring precision. The automation process begins with basic SQL injection tests to identify potential databases on a target server. Users can then enumerate all databases, list tables within a specific database, or retrieve column details from a specific table. The tool also allows users to extract data from tables, fingerprint and retrieve information about the current database and user. Additional functionalities include checking whether the current user has administrative privileges, enumerating all database users, retrieving password hashes of users, and testing for SQL injection vulnerabilities using various techniques such as Boolean-based blind injections, Error-based injections, Stacked queries, and Time-based blind injections. To enhance its effectiveness against security measures, the tool incorporates tamper scripts for bypassing such protections. When a specific task is selected by the user, the corresponding SQLMap command is executed. These reports provide a comprehensive analysis of vulnerabilities and actionable insights for remediation[9].

3.3 Integration of Machine Learning

To further enhance its capabilities in vulnerability detection, the tool integrates machine learning techniques. This integration allows it to analyze patterns in data from previous penetration tests and predict potential vulnerabilities in new inputs. The machine learning process begins with feature engineering, where data features are selected and transformed into a structured format suitable for model training. For this project, datasets derived from SQL injection penetration tests were used. Key features include SQL commands and their outputs, while vulnerabilities serve as labels. A Random Forest classifier is employed as the primary model due to its ability to construct decision trees that analyze patterns effectively. To make predictions practical for real-world applications, an interface based on ipy widgets allows users to input new data for analysis. The input is processed through TF-IDF vectorization before being fed into the trained model to predict whether it is vulnerable or not. This integration allows it to analyze patterns in data from previous penetration tests and predict potential vulnerabilities in new inputs. For this project, datasets derived from SQL injection penetration.

3.4 Practical Applications

The integration of machine learning makes this tool highly practical for cybersecurity professionals. Users can input new values into an interactive interface to check for vulnerabilities in real time. This predictive capability enhances efficiency in identifying risks while providing actionable insights through detailed reports. By combining automation with machine learning techniques, this tool offers a comprehensive solution for identifying and mitigating SQL injection vulnerabilities in web applications. Its ability to predict potential risks based on historical data patterns makes it an invaluable resource for securing systems against cyberattacks[11].

3.5 Importance of SQL Injection Automation in Cybersecurity

The SQL Injection Automation Tool represents a significant advancement in this domain by combining automation with machine learning to streamline penetration testing and vulnerability assessment processes. Automating the detection and exploitation of such vulnerabilities is essential for cybersecurity professionals to stay ahead of attackers. The tool's ability to automate repetitive tasks not only saves time but also reduces human error, ensuring consistent and accurate results. By integrating advanced features like database enumeration, data extraction, and vulnerability classification, it provides a comprehensive solution for identifying and mitigating risks. Moreover, its machine learning capabilities enable it to predict vulnerabilities based on historical data patterns, making it a proactive tool for securing web applications [12].

3.6 Enhancing User Experience through Design

The user experience is a critical factor in the success of any security tool. The SQL Injection Automation Tool is designed with a focus on simplicity and accessibility. Its text-based interface is intuitive, allowing users to navigate through various options effortlessly. This design choice ensures that even users with limited technical expertise can operate the tool effectively. At the same time, advanced users can leverage its

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 267-272, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE43
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

full range of functionalities to perform in-depth analyses. The modular architecture of the tool further enhances user experience by providing flexibility for customization and expansion. Users can integrate additional components or modify existing ones without disrupting the overall system. This adaptability makes the tool suitable for a wide range of use cases, from small-scale penetration tests to comprehensive security audits[13].

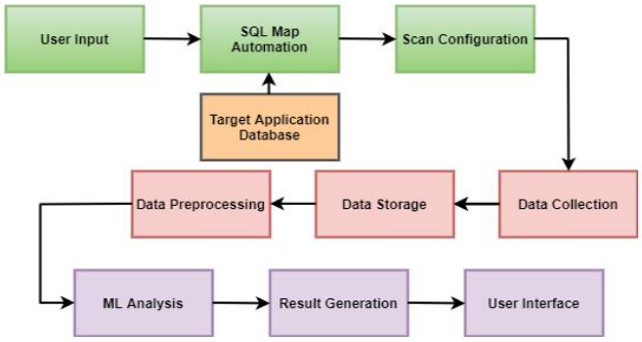

Fig. 1: Proposed Architecture

IV. RESULTS AND DISCUSSION

The tool also incorporated a machine learning model for vulnerability prediction, evaluated using metrics like accuracy, precision, recall, and F1-score. The Random Forest classifier achieved the highest accuracy and consistency in detecting vulnerabilities. Accuracy detection of the vulnerabilities that a given website is summarized in Table 1.

Table 1: Performance Analysis of SQL Injections

Category	Precision	Recall	F1-Score	Support
Boolean-based Blind SQL Injection (POST parameter)	1.00	1.00	1.00	5
Boolean-based Blind SQL Injection	1.00	1.00	1.00	3
Error-based SQL Injection	1.00	1.00	1.00	1
Error-based SQL Injection (POST parameter)	1.00	1.00	1.00	3
Stacked Queries SQL Injection	1.00	1.00	1.00	3
Time-based Blind SQL Injection	1.00	1.00	1.00	1
Union-based SQL Injection	1.00	1.00	1.00	1
Accuracy	-	-	1.00	17
Macro Avg	1.00	1.00	1.00	17
Weighted Avg	1.00	1.00	1.00	17

4.1 Performance and Analysis of the SQL Injection Automation Tool

The SQL Injection Automation Tool was evaluated for its efficiency using the open-source website testphp.vulnweb.com, which contains known vulnerabilities. The tool integrates SQLMap to automate SQL injection testing, significantly reducing the time and effort compared to manual methods.

Key Results:

- It detected vulnerabilities through techniques such as boolean-based blind, time-based blind, and UNION query injections.

Example payloads:

- Boolean-based blind: artist=1 AND 7789=7789
- UNION query: artist=-4544 UNION ALL SELECT NULL, NULL, CONCAT(0x7170717a71,0x684e695859474658565376 5)

The tool's accuracy was assessed by comparing its detection results with expected outcomes. It successfully identified various SQL injection types and provided detailed outputs, including affected parameters, injection techniques, titles of attacks, and payloads. This demonstrates its capability to comprehensively detect vulnerabilities.

4.2 Comparison of Manual vs. Machine Learning-Based Testing

Table 2 highlights the comparison of combining automation and machine learning over traditional manual testing and its advantages.

Table 2: Comparison of Manual and ML Based Testing

Metric	Manual Testing Results	Machine Learning Results	Remarks
Accuracy	Accuracy varied, depending on the tester's experience. It was usually less consistent.	89% accuracy using the Random Forest model.	Machine learning gave more accurate and consistent results.
Detection Speed	Testing took longer, around 3 minutes per test, because	Faster, around 30 seconds per test with automated	Automation and machine learning made testing much

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue INSPIRE'25, pp 267-272, April-2025
https://doi.org/10.47001/IRJIET/2025.INSPIRE43
International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25)

	it was done manually.	commands and machine learning.	quicker.
Resource Consumption	High effort and time required for manual testing, needed skilled people.	Low effort needed; automation and ML reduced the need for lots of human work.	The process was more efficient and took less time.
Consistency	Results could change depending on who was testing and when.	Consistent results, the machine learning model gave the same results every time.	Machine learning gave reliable and repeatable results.

4.3 Discussion and Future Enhancement

The integration of machine learning into the SQL Injection Automation Tool has demonstrated significant improvements in functionality and productivity. By automating vulnerability detection with SQLMap and leveraging machine learning algorithms like Random Forest, the tool ensures faster, more accurate, and scalable testing processes. Efforts could also focus on improving the model's effectiveness against advanced and evasive attack techniques and integrating it into broader cybersecurity frameworks. The project underscores the potential of automation and machine learning in real-time threat detection and prevention, opening avenues for more robust and adaptive cybersecurity solutions.

V. CONCLUSION

The research proposal outlines enhancing cybersecurity by automating the vulnerability assessment process and leveraging advanced machine learning techniques to identify SQL injection threats. This approach aims to reduce the time and effort required for penetration testing while improving the accuracy and efficiency of detecting vulnerabilities, a critical aspect of combating modern cyber threats. The project employs the Random Forest model, which has demonstrated strong performance in identifying SQL injection vulnerabilities. This success highlights the potential of machine learning models in cybersecurity, showcasing their ability to transform traditional methods into more efficient, automated processes. By enabling real-time analysis and results, the integration of machine learning enhances the effectiveness of tasks such as automating SQLMAP commands and simplifying complex security testing procedures for penetration testers. Additionally, the proposed method reduces reliance on technical personnel and shortens testing times, enabling more frequent and comprehensive security evaluations. This foundational work not only addresses current challenges in web application security but also serves as a starting point for future advancements.

REFERENCES

[1] K. R. Veerabudren and G. Bekaroo, "Security in web applications: A comparative analysis of key sql injection detection techniques," in 2022 4th International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM). IEEE, 2022, pp. 1–6.

[2] M. Lodeiro-Santiago, C. Caballero-Gil, and P. Caballero-Gil, "Collaborative sql-injections detection system with machine learning," in Proceedings of the 1st International Conference on Internet of Things and Machine Learning, 2017, pp. 1–5.

[3] J. A. Recio-Garc´ıa, M. G. Orozco-del Castillo, and J. A. Soladrero, "Case-based explanation of classification models for the detection of sql injection attacks." in ICCBR Workshops, 2023, pp. 200–215.

[4] D. T. Loughran, M. K. Salih, and V. H. Subburaj, "All about sql injection attacks," in Journal of The Colloquium for Information Systems Security Education, vol. 6, no. 1, 2018, pp. 24–24.

[5] A.Sadeghian, M. Zamani, and A. A. Manaf, "A taxonomy of sql injection detection and prevention techniques," in 2013 international conference on informatics and creative multimedia. IEEE, 2013, pp. 53–56.

[6] P. McDaniel and B. Nuseibeh, "Guest editors' introduction: Special section on software engineering for secure systems," IEEE Transactions on Software Engineering, vol. 34, no. 1, p. 3, 2008.

[7] H. Gupta, S. Mondal, S. Ray, B. Giri, R. Majumdar, and V. P. Mishra, "Impact of sql injection in database security," in 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). IEEE, 2019, pp. 296–299.

[8] I.Benikhlef, C. Wang, and S. Gulomjon, "Mutation based sql injection test cases generation for the web based application vulnerability testing," in 2nd International Conference on Electronics, Network and Computer Engineering (ICENCE 2016). Atlantis Press, 2016, pp. 546–551.

[9] A.A. Anaoval, A. T. Zy, and S. Suherman, "Analysis of manual and automated methods effectiveness in website penetration testing for identifying sql injection vulnerabilities," Journal of Computer Networks,

Architecture and High Performance Computing, vol. 6, no. 3, pp. 1204–1212, 2024.

[10] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, "Enhancing web application security through automated penetration testing with multiple vulnerability scanners," Computers, vol. 12, no. 11, p. 235, 2023.

[11] E. A. Altulaihan, A. Alismail, and M. Frikha, "A survey on web application penetration testing," Electronics, vol. 12, no. 5, p. 1229, 2023.

Citation of this Article:

M. Fathima Begum, Lekha Sree C, & Manasa P. (2025). Enhancement of Web Application Security using SQLMap and Machine Learning. In proceeding of International Conference on Sustainable Practices and Innovations in Research and Engineering (INSPIRE'25), published by *IRJIET*, Volume 9, Special Issue of INSPIRE'25, pp 267-272. Article DOI https://doi.org/10.47001/IRJIET/2025.INSPIRE43

****

