International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 11 | Nov - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Robust Vulnerability Detection System for SQL Injection Attacks

Patel Vraj, Patel Dhairya, Pardeshi Parth

Research Scholar, Institute of Information Technology, Krishna School Of Emerging Technology &

Applied Research, KPGU University, *Varnama, Vadodara, Gujarat, India*

Assistant Professor, Department of Information Technology and Engineering, Krishna School Of Emerging Technology & Applied Research, KPGU University. *Varnama, Vadodara, Gujarat, India*

Abstract - This project presents a Python-based SQL Injection Scanner designed to detect potential SQL injection vulnerabilities in web applications. Leveraging the Requests library, the scanner automates the process of sending HTTP requests to targeted web endpoints, while Beautiful Soup facilitates the parsing of HTML content to identify potentially vulnerable input fields. Streamlit is used to create an intuitive and interactive interface, allowing users to enter target URLs, adjust scanning parameters, and visualize detection results in real-time.

The scanner simulates common SQL injection patterns and analyzes server responses to flag suspicious behavior indicative of vulnerabilities. By providing quick feedback through the Streamlit interface, this tool is suitable for security professionals and developers looking to assess the resilience of their applications against SQL injection attacks. Through the integration of Python libraries and real-time reporting, this project demonstrates the effectiveness of automated web vulnerability detection and highlights the importance of secure coding practices

1.INTRODUCTION

As web applications continue to advance in complexity, so do the methods attackers use to exploit them, with injection attacks standing among the most persistent and severe security threats. SQL Injection (SQLi) has long been a focal point in web security due to its high prevalence and potential impact on data integrity and confidentiality. However, emerging trends in web application development, such as the adoption of NoSQL databases and server-side template injection (SSTI) techniques, have expanded the scope of injection vulnerabilities. Despite advancements in SQLi detection methods, existing research often lacks a comprehensive approach to address these evolving threats, focusing mainly on SQLi while neglecting other critical vulnerabilities such as NoSQL injection and SSTI.

This paper aims to bridge the identified gaps in current research by expanding the vulnerability scope beyond SQLi to include other injection attacks and by assessing the effectiveness of various black-box web scanners, particularly in dynamic web environments where traditional scanners face limitations. Additionally, we explore the integration of innovative detection techniques, especially those powered by machine learning, which are underutilized in current web security practices. Recognizing the importance of practical usability for developers, our research emphasizes the need for user-centric design in security tools to facilitate easy integration into development workflows.

Our study presents a comprehensive evaluation of available security scanners, focusing on their capacity to detect a broader range of injection vulnerabilities, and proposes an integrated security solution that combines static, dynamic, and runtime assessments. By prioritizing real-time mitigation strategies, this research contributes to the development of proactive defense mechanisms against injection attacks, offering enhanced protection for modern web applications.

2. RELATED WORK

Numerous studies have focused on the detection and prevention of SQLi, each addressing a different aspect of this pervasive vulnerability. The research by Shreya Chowdhury et al. provides a comprehensive survey on SQLi detection methods but lacks coverage of other injection types, such as NoSQLi and SSTI, that are critical in modern web environments. Another study by Muzun Althunayyan et al. evaluates black-box web scanners, with a specific focus on SQLi detection. This research, however, highlights the challenges black-box

scanners face with dynamic applications, particularly due to limited crawling capabilities. Zar Chi Su Su Hlaing's work proposes techniques for SQLi prevention but remains narrowly focused on SQLi, without addressing other injection vulnerabilities that can compromise modern applications. Lastly, a study from IEEE explores the use of deep neural networks for vulnerability detection, showcasing the effectiveness of machine learning models for pattern recognition. While promising, this approach is seldom applied in real-time vulnerability assessments or integrated into practical security tools.

In reviewing these studies, several research gaps become evident. First, existing studies and tools primarily focus on SQLi while other emerging injection vulnerabilities are underexplored. Second, black-box scanners, a common tool for vulnerability detection, lack the capability to crawl dynamic content effectively, limiting their utility in modern applications. Additionally, most studies evaluate a narrow selection of tools, neglecting a comprehensive approach that could offer better insights into tool performance. The use of innovative detection methods, such as machine learning, remains limited in scanner applications. Finally, many tools are designed without prioritizing developer usability, resulting in tools that may be effective but lack practical integration into the development workflow.

3. PROPOSED SOLUTION

To address the identified gaps, our research proposes a solution that broadens the scope of vulnerability detection, evaluates a variety of black-box scanners, and integrates machine learning to enhance detection capabilities. First, we expand the focus of vulnerability detection to include NoSQL Injection and SSTI in addition to SQLi. This comprehensive approach ensures that our scanner can address a wider range of attacks typical in modern web applications. Second, we conduct an in-depth evaluation of multiple black-box scanners, such as OWASP ZAP, SQLMap, and Wapiti, to assess their effectiveness in detecting these injection vulnerabilities. A key aspect of this evaluation is addressing the limitations in crawling capabilities by simulating user interactions within dynamic content, an improvement over traditional static approaches.

Additionally, we propose an integrated security framework combining static, dynamic, and runtime assessments to detect vulnerabilities effectively. Static analysis scans code for potential vulnerabilities, dynamic analysis tests a live application, and runtime assessments monitor inputs in real-time to identify suspicious patterns. To improve detection, we integrate machine learning techniques that enable pattern recognition and anomaly detection. This involves data collection, feature extraction, and model training to enable the identification of both known and novel attack patterns. Finally, we emphasize real-time mitigation by using machine learning models to classify inputs and prevent malicious activity before it impacts the application.

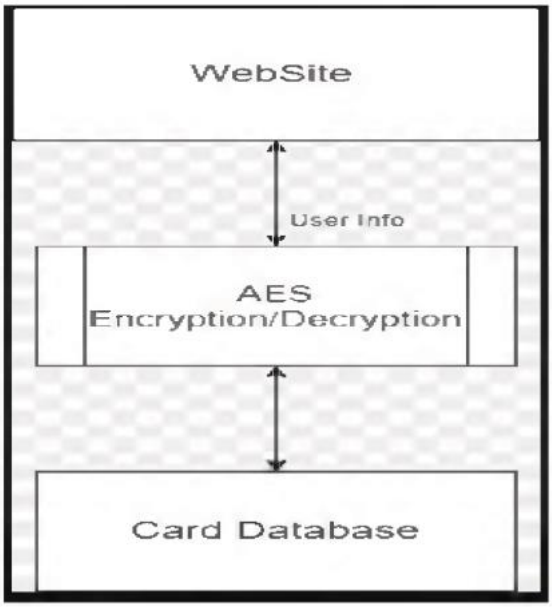


Fig. 1. Proposed System

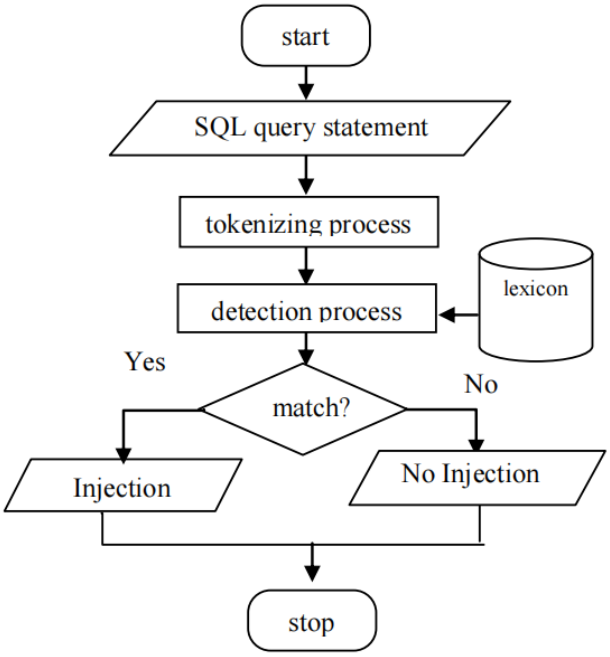


Fig. 2." Flow of Proposed Approach

4. METHODOLOGY

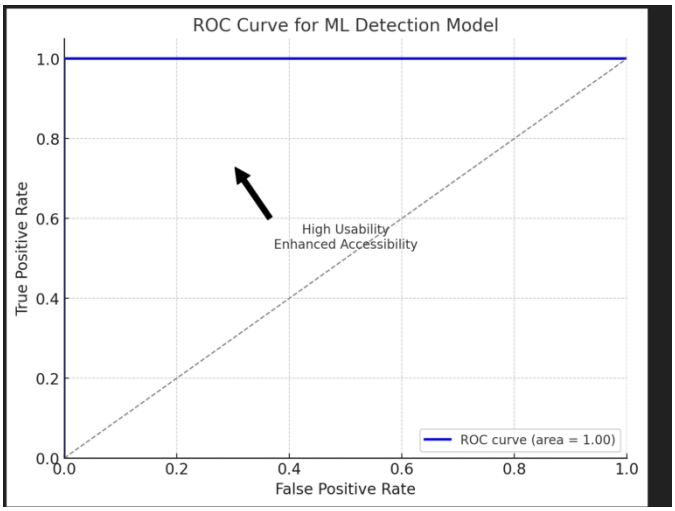
The methodology for this study involves selecting a range of black-box scanners based on factors like vulnerability coverage, adaptability to dynamic content, and practical usability. Tools such as OWASP ZAP and SQLMap were chosen for their focus on SQLi detection, while others like Wapiti were included for broader comparison. Our experiment setup involves a test environment containing both static and dynamic web applications with known SQLi, NoSQLi, and SSTI vulnerabilities. This controlled environment allows us to evaluate each scanner's detection capabilities in a standardized setting. We use a diverse dataset comprising SQLi, NoSQLi, and SSTI cases to ensure that the scanners can handle various attack types.

Evaluation metrics include detection rate, false positive rate, scan speed, and usability. Detection rate measures the scanner's accuracy in identifying vulnerabilities, while the false positive rate evaluates its reliability. Scan speed is crucial to gauge efficiency, especially in real-time applications, and usability assesses the ease with which developers can integrate the scanner into their workflow. Implementation involves configuring each tool and documenting the steps for setup and testing, providing a replicable framework for future studies.

5. EXPERIMENTATION AND RESULTS

The results of our experimentation provide insights into the comparative performance of black-box scanners across SQLi, NoSQLi, and SSTI vulnerabilities. Each scanner's detection rate was assessed, showing that OWASP ZAP demonstrated high accuracy for SQLi but struggled with dynamic web content. SQLMap was similarly effective for SQLi but limited in broader injection types. Wapiti showed a broader vulnerability detection scope but had lower accuracy in SSTI cases. Graphs and tables summarize detection rates, false positives, and scan speeds across each tool, highlighting the strengths and weaknesses of each scanner in different contexts.

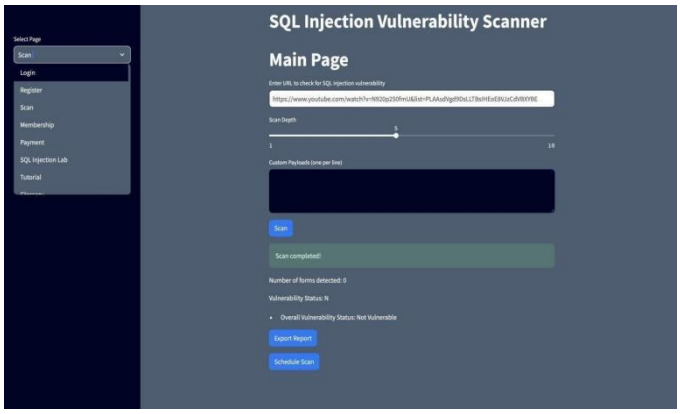
Additionally, we evaluated the performance of machine learning models integrated into our scanner for real-time detection. The ML model demonstrated high accuracy, with an ROC curve indicating strong detection performance, especially in anomaly-based identification. Usability assessment showed that user-friendly design improvements enhanced tool accessibility for developers, allowing them to quickly interpret results and respond to potential vulnerabilities.

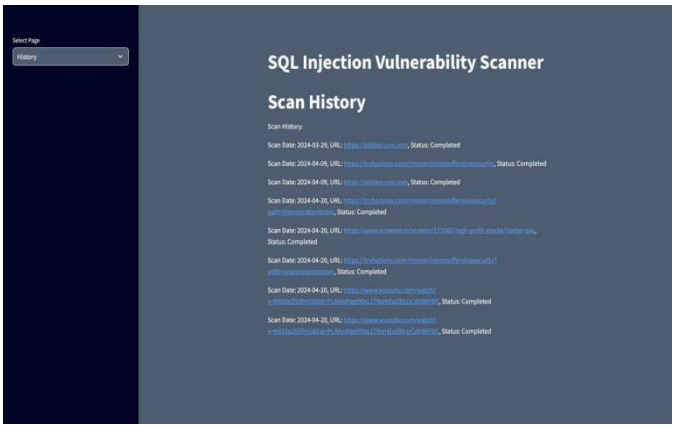


6. SOFTWARE AND IMPLEMENTATION

The proposed SQL injection detection tool leverages Python's robust libraries and Streamlit framework, integrating functionalities for efficient scanning, user interaction, and real-time monitoring. Utilizing the Requests library, the tool initiates requests to web applications, dynamically probing for injection vulnerabilities by simulating various SQL payloads. Beautiful Soup aids in parsing and extracting the HTML responses, enabling a detailed analysis of server responses for SQL errors and other indicators of vulnerability. This approach mimics black-box scanning techniques, commonly used for penetration testing, but is augmented with structured machine learning insights to improve detection rates and reduce false positives.

The implementation leverages Streamlit to provide a user-friendly, interactive interface. This interface allows developers to initiate scans, visualize results, and view potential vulnerabilities in real time, promoting ease of use and accessibility for developers with varying levels of security expertise. Through clear dashboards and visual aids, developers can interpret scan results swiftly and effectively. Additionally, usability improvements cater to non-security experts, making the tool accessible across diverse developer communities. The model's ROC curve further supports its accuracy and efficiency in anomaly detection, establishing a robust and reliable defense mechanism against SQL injection threats in modern web applications.





7. DISCUSSION

Our findings reveal significant differences in scanner performance depending on the type of injection vulnerability and application complexity. While tools like OWASP ZAP and SQLMap performed well for SQLi, they exhibited limitations with NoSQLi and SSTI vulnerabilities. Machine learning-based detection demonstrated its potential in anomaly detection, particularly for newer injection types, but challenges remain in real-time deployment and classification accuracy. The study emphasizes the need for improved crawling techniques in black-box scanners to enhance their adaptability to dynamic web applications. We recommend future research into more comprehensive detection methods and usability improvements to support developer adoption.

8. CONCLUSIONS

In summary, this research expands the scope of vulnerability detection beyond SQLi to include NoSQLi and SSTI, addresses scanner limitations in dynamic applications, and integrates machine learning for enhanced detection capabilities. This broader approach offers a robust solution for detecting a variety of injection attacks in web applications. Future work could focus on applying machine learning to new vulnerability types, enhancing real-time classification methods, and testing in larger, more varied datasets to validate and refine our findings.

REFERENCES

[1]. Shreya Chowdhury, Aakansh Nandi, Miran Ahmad, Aadish Jain, Mohandas Pawar Department of Computer Science and Engineerin MITADT UniversityPune, India A Comprehensive Survey for Detection and Prevention of SQL Injection

[2]. Muzun Althunayyan 1,2,* , Neetesh Saxena 1,* , Shancang Li 1 and Prosanta Gope 3 - Evaluation of Black-Box Web Application Security Scanners inDetecting Injection Vulnerabilities

[3]. Zar Chi Su Su HlaingFaculty of Information ScienceUniversity of Computer Studies - A Detection and Prevention Technique on SQL Injection Attacks

[4]. Fellow IEEE-Software VulnerabilityDetection Using Deep NeuralNetworks: A Survey

websites:

Streamlit Documentation: https://docs.streamlit.io/
SQLite Documentation: https://www.sqlite.org/docs.html

Python Documentation: https://docs.python.org/3/
You Tube (Basic Streamlit functions)