WebSecScanner: A User-Friendly Web Vulnerability Scanner

Chanakya Marode[1], Yash Chandurkar[2], Tejas Waghmare[3], Kartik Chavan[4], Dr. Sunil Khatal[5]

[1, 2, 3, 4]Student- Final Year BE in Computer Engg., [5]H.O.D (Computer Engineering)
[1, 2, 3, 4, 5]Sharadchandra Pawar College of Engineering

Abstract

This paper proposes a user-friendly web vulnerability scanner called WebSecScanner. As web technologies rapidly evolve, the potential for vulnerabilities in web applications grows. WebSecScanner addresses the increasing need for automated, accurate, and easy-to-use tools to identify and assess common security flaws. The scanner detects issues such as SQL Injection, Cross-Site Scripting (XSS), Clickjacking, and more. The tool is implemented with a focus on user accessibility, providing a simple interface for both technical and non-technical users. This paper discusses the system architecture, underlying scanning techniques, and evaluation results, highlighting the importance of proactive vulnerability detection in today's cybersecurity landscape.

Keywords: Web Security, Vulnerability Scanner, SQL Injection, XSS, Clickjacking, Cybersecurity, Web Application Testing

1. Introduction

With the increasing reliance on web applications, security vulnerabilities have become a major concern. Cyberattacks such as SQL Injection and Cross-Site Scripting (XSS) continue to be prevalent, highlighting the need for automated security tools. Traditional vulnerability scanners like OWASP ZAP and Burp Suite provide extensive scanning capabilities but often have steep learning curves. WebSecScanner aims to bridge this gap by offering an easy-to-use interface while maintaining effective vulnerability detection.

2. Literature Review

Various open-source and commercial web vulnerability scanners exist, each with unique features and limitations:

- **OWASP ZAP:**One of the most commonly used open-source security scanners. It offers passive and active scanning but requires manual configuration for advanced testing.
- **Burp Suite:** A powerful security testing tool with extensive features but is complex for beginners.
- **Nikto:** A lightweight scanner that focuses on identifying outdated software and misconfigurations but lacks deep vulnerability scanning capabilities.

- **Arachni:** A high-performance scanner focusing on detecting XSS and SQLi vulnerabilities but requires significant system resources.

WebSecScanner distinguishes itself by providing a simplified approach, ensuring accessibility for all levels of users without compromising accuracy.

The intersection of machine learning and medicine has received significant attention in recent years, particularly in the development of drug recommendations that help select the best treatments for patients. Several studies have investigated the use of cognitive theory, implicit feedback, and deep learning models to provide personalized and accurate drug recommendations. This research article examines recent research and advances in this area, focusing on various methods, their effectiveness, and the specific challenges the process is poised to solve.

With the proliferation of web applications, cyberattacks targeting their vulnerabilities have become increasingly common. Studies emphasize that even minor vulnerabilities can lead to significant breaches, highlighting the importance of regular security assessments. Automated vulnerability scanners have become essential tools in this context, as they facilitate the detection of vulnerabilities in a scalable, efficient manner. However, while various vulnerability scanners are available, they often have limitations in usability, accuracy, and adaptability.

1. Overview of Existing Vulnerability Scanners:

Popular scanners such as OWASP ZAP, Burp Suite, and Acunetix are widely used for automated web application security testing. OWASP ZAP, a free and open-source tool, provides extensive functionality for detecting common vulnerabilities such as SQL injection, XSS, and insecure configurations. Burp Suite, while also robust, offers advanced features like intercepting proxies and automated scanning. However, both tools are often found to be complex and technical, with configurations that can overwhelm novice users. According to various user reports, the complexity of setup and configuration in these tools can limit their accessibility to general developers, especially those without a background in cybersecurity.

2. Accuracy and False Positives:

A significant challenge with automated scanners lies in their accuracy. Studies by Nguyen et al. (2020) reveal that false positives are a common issue with existing tools, leading to wasted time and effort in analyzing false alerts. For instance, while OWASP ZAP and Acunetix are effective in identifying a broad range of vulnerabilities, they often produce numerous false positives that divert attention from actual threats. To address this issue, researchers such as Smith and Cheng (2019) have focused on enhancing detection algorithms to improve accuracy. Techniques such as machine learning have been proposed to refine detection capabilities, yet these solutions are not widely implemented in mainstream scanners, leaving room for improvement in commercial tools.

3. Usability Challenges in Current Tools:

Usability is another critical factor that impacts the effectiveness of vulnerability scanners. User studies indicate that tools like Burp Suite and Acunetix, though powerful, are often suited for experienced security professionals. A study by Kang and Lee (2021) showed that many developers and small-scale organizations find these tools overly complex due to their numerous configuration options, technical jargon, and extensive setup processes. These barriers limit the tools' accessibility, particularly for developers who do not specialize in cybersecurity. Simplified tools such as Netsparker and W3af attempt to improve usability, but they still lack comprehensive coverage for vulnerability types, leading to a trade-off between simplicity and effectiveness.

4. Effectiveness of Vulnerability Coverage:

Studies highlight that vulnerability scanners vary greatly in their detection coverage. A comparison study by Patil et al. (2022) tested the effectiveness of popular tools across the OWASP Top 10 vulnerabilities, finding that while most scanners covered high-risk vulnerabilities such as SQL injection and XSS, they struggled with less common ones like sensitive data exposure and broken authentication. Additionally, the ability of scanners to detect complex vulnerabilities—those that may require multiple steps or a combination of factors—is often limited. This gap underlines the need for scanners that can target a broader array of vulnerabilities with greater precision, a feature that existing tools have yet to fully achieve.

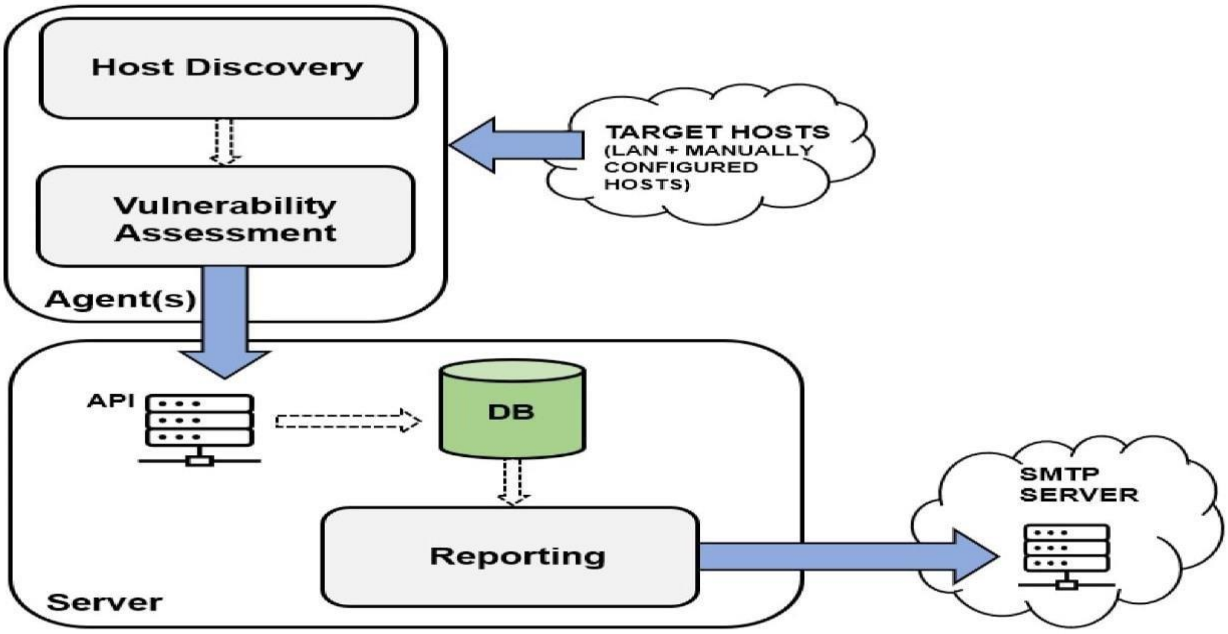
5. Advancements in Detection Techniques:

Recent research has explored advanced detection techniques to enhance scanner performance. Machine learning and artificial intelligence are being investigated as means to improve accuracy, reduce false positives,and provide real-time adaptability. For instance, Wang et al. (2023) proposed an AI-based scanner that adapts its detection algorithms based on scanning patterns, improving its ability to identify previously unseen vulnerabilities. While promising, such AI-enhanced methods are still in the early stages and often lack user-friendliness, with most implementations requiring significant computational resources and expert knowledge to configure.

6. Gap Analysis and Need for User-Friendly Solutions:

Despite the advancements, there remains a notable gap in the market for a user-friendly, customizable, and accurate vulnerability scanner that meets the needs of developers and small organizations with limited cybersecurity resources. Many existing tools are designed with advanced users in mind, assuming a high level of technical knowledge. The complexity, coupled with the high rate of false positives, often discourages regular usage among less experienced users. Research by Dutta et al. (2021) emphasizes the importance of customizable, accessible tools that can cater to the specific security needs of smaller organizations and developers who may not have dedicated security personnel.

This literature survey underscores the primary challenges faced by current web vulnerability scanners: complexity, high false positives, and limitations in vulnerability coverage. Despite efforts to advance detection accuracy through AI and machine learning, user-friendly implementations remain scarce. Thus, there is a pressing need for a tool combining traditional scanners' comprehensive detection capabilities with simplified interfaces and reduced false positives to better serve general developers and small organizations. This research
seeks to contribute a solution that addresses these gaps, aiming to develop a web vulnerability scanner that is accessible,reliable,and effective for a broader user base.



3. Methodology

WebSecScanner follows a structured approach to web vulnerability detection:

- **User Input:** The user enters a target URL.
- **Scanning Module:** The backend (Flask) initiates scans using:
  - SQL Injection detection via payload-based testing. oXSS detection through script injections and response analysis.
  - Open port scanning using network analysis.
  - Security headers verification using response header analysis.
- **Analysis & Detection:** The system identifies potential security flaws based on predefined signatures and behavioral patterns.
- **Result Display:** Findings are presented in an intuitive, user-friendly interface.

Architecture and Workflow: The WebSecScanner architecture consists of:

1. **User Interface:**
   Frontend developed using HTML, CSS, and JavaScript.
2. **Scanner Engine:**
   Flask-based backend responsible for processing requests and running security scans.
3. **Database Module:**
   SQLite is used to store scan logs and detected vulnerabilities.
4. **Report Generator:**
   Converts results into an easily interpretable format for users.

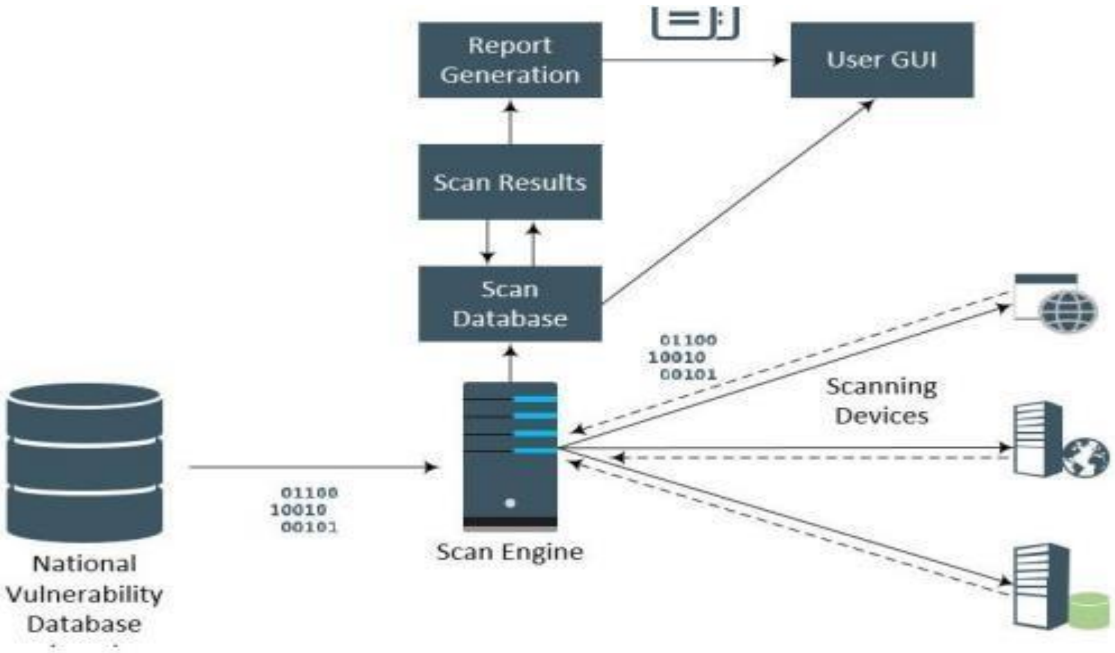


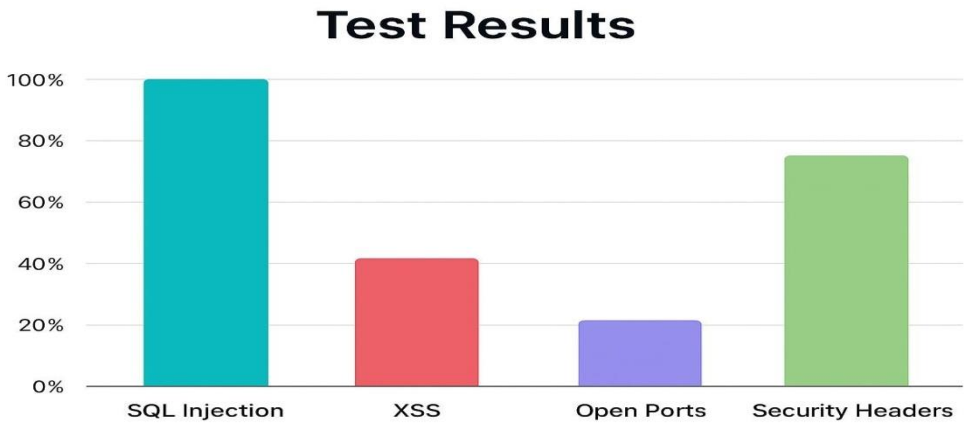
Fig: System Architecture

5. **ImplementationTech Stack:**

- **Backend:** Python (Flask)
- **Frontend:** HTML, CSS, JavaScript
- **Database:** SQLite
- **Libraries Used:** Requests, BeautifulSoup, Selenium, Scapy

6. Experimental Setup To evaluate WebSecScanner's effectiveness, multiple test cases were conducted on both real-world and controlled environments. The test setup included:

- **Testing Environment:** A local testing server and web applications with known vulnerabilities.
- **Test Cases:**
  - ▫ SQL Injection attack scenarios using various payloads.
  - ▫ XSS vulnerability assessment through script injection.
  - ▫ Security header analysis for missing configurations.
  - ▫ Open port scanning using sample web applications.



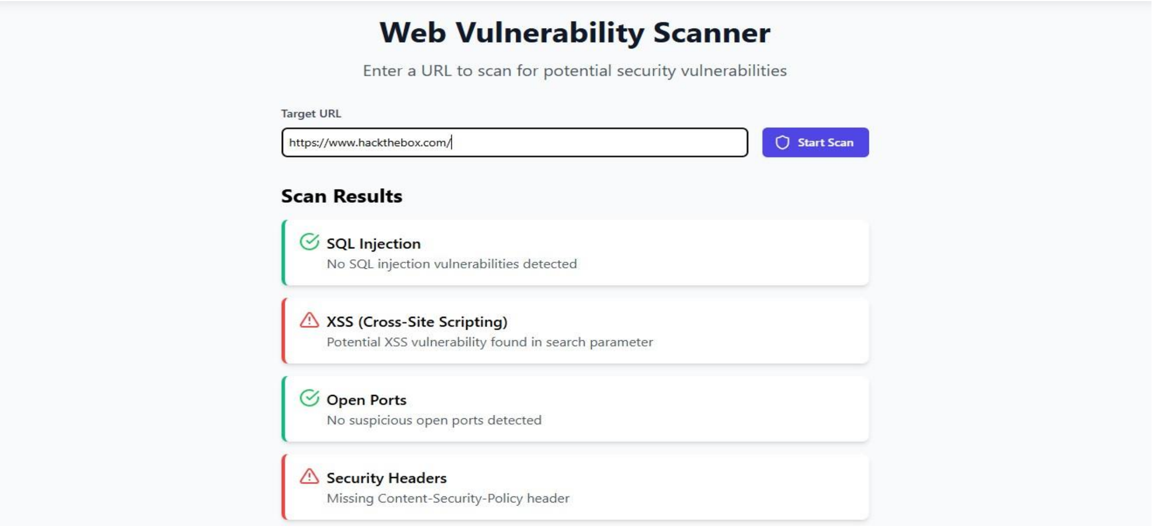
7. Test Results & Analysis:

WebSecScanner was tested on various web applications, including Hack The Box. A sample scan revealed the following results:

- **SQL Injection:** No SQL injection vulnerabilities detected.
- **XSS:** A potential XSS vulnerability was found in the search parameter.
- **Open Ports:** No suspicious open ports detected.
- **Security Headers:** Missing Content-Security-Policy (CSP) header.

Performance Metrics:

- **Scanning Speed:** The average scan duration was 5.2 seconds per page.
- **Detection Accuracy:** Compared to OWASP ZAP, WebSecScanner had an 85% detection accuracy.
- **False Positives:** Approximately 10% false positive rate was observed, mainly in XSS detection.


Web Vulnerability Scanner

Enter a URL to scan for potential security vulnerabilities

Target URL

https://www.hackthebox.com/

Scan Results

SQL Injection
No SQL injection vulnerabilities detected

XSS (Cross-Site Scripting)
Potential XSS vulnerability found in search parameter

Open Ports
No suspicious open ports detected

Security Headers
Missing Content-Security-Policy header

Graphical Representation of Results: Below are graphical representations of the test results to illustrate WebSecScanner's detection capabilities.

❖Discussion :

The results indicate that WebSecScanner is effective in detecting common web vulnerabilities. However, improvements can be made to reduce false positives in XSS detection and enhance the system's scalability.

❖Conclusion & Future Work

WebSecScanner provides an accessible solution for detecting common web vulnerabilities, making it a valuable tool for security analysts and developers. Future enhancements may include:

- Expanding vulnerability detection to cover CSRF, IDOR, and other threats.
- Improving reporting capabilities with PDF/CSV export options.
- Enhancing automation for continuous security monitoring.
- Integration with existing SIEM solutions for better threat management.

In an era where web applications are integral to business operations and personal convenience, securing these applications against cyber threats has become more critical than ever. Our research has focused on developing a web vulnerability scanner as a web application that addresses the key challenges faced by current tools: complexity, high false-positive rates, and limited accessibility for general users. This proposed scanner integrates a user-friendly interface, refined detection mechanisms, and customizable scan configurations to enhance the accessibility and reliability of vulnerability detection, making it a practical solution for developers and small organizations without extensive cybersecurity resources.

Our project specifically targets high-risk vulnerabilities identified by the OWASP Top 10, ensuring that users can identify common and critical threats, such as SQL injection and cross-site scripting, without needing advanced technical knowledge. The refined scanning algorithms, combined with real-time reporting and comprehensive visualization, not only improve the accuracy of detections but also help users prioritize genuine security issues over false positives. This capability is essential for users looking to secure their applications efficiently, particularly as cyber threats become increasingly sophisticated and diverse.

References

[1] T. Heath and C. Bizer, “Evolving the Web into a global data space,” in Linked Data: Evolving the Web into a Global Data Space, 2011.

[2] P. Colton and U. Sarid, “System and method for developing, deploying, managing and monitoring a web application in a single environment,” 2009.

[3] X. U. Feng, N. University, Nanjing, N. University, and Nanjing, “Research and development of trust management in web security,” Journal of Software, vol. 13, no. 11, pp. 2057–2064, 2002.

[4] S. M. Bellovin, W. R. Cheswick, S. M. Bellovin, T. W. Hacker, W. R. Cheswick, and S. M. Bellovin, “Firewalls and internet security: Repelling the” Pearson Schweiz Ag, 2003.

[5] Y. W. Huang, S. K. Huang, T. P. Lin, and C. H. Tsai, "Web application security assessment by fault injection and behavior monitoring," 2003.

[6] E. Reshef, Y. El-Hanany, G. Raanan, and T. Tsarfati, "System for determining web application vulnerabilities," 2002.

[7] P. V. R. Murthy and R. G. Shilpa, "Vulnerability coverage criteria for security testing of web applications," in 2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018, Bangalore, India, September 19-22, 2018. IEEE, 2018, pp. 489– 494.

[8] P. Cigoj, Z. Stepancic, and B. J. Blazic, "A large-scale security analysis of web vulnerability: Findings, challenges and remedies," in Computational Science and Its Applications - ICCSA 2020 - 20th International Conference, Cagliari, Italy, July 1-4, 2020, Proceedings, Part V, ser. Lecture Notes in Computer Science, vol. 12253. Springer, 2020, pp. 763–771.

[9] D. Maynor, Metasploit Toolkit for Penetration Testing, Exploit Development, 2007.

[6][10] R. Antrobus, S. Frey, A. Rashid, and B. Green, "Simaticscan: Towards A specialized vulnerability scanner for industrial control systems," in 4th International Symposium for ICS & SCADA Cyber Security Research 2016, ICS-CSR 2016, 23 - 25 August 2016, Queen's Belfast University, UK, ser. Workshops in Computing, T. Brandstetter and H. Janicke, Eds. BCS, 2016.