Grenze International Journal of Engineering and Technology, January Issue

VulneraXpert: Smart Web Vulnerability Detection Scanner

Adilakshmi Yannam[1], Jami Anjana Adi Sathvik[2], Persis Jandrajupalli[3], Harikrishna Metlapalli[4] and Puppala Sandeep[5]
[1-5]Dept. of CSE (AI & ML), S R Gudlavalleru Engineering College Gudlavalleru, India
Email: laxmi072003@gmail.com, sathviksacchu@gmail.com, persisjandrajupalli@gmail.com, metlapallihari@gmail.com, puppalasundeepkumar123@gmail.com

***Abstract*— Security plays a crucial role in modern web development, as web applications frequently become targets of cyber threats. To efficiently detect critical security vulnerabilities, this article introduces an advanced multi-threaded web vulnerability scanner. The scanner systematically analyzes web pages for threats such as Cross-Site Scripting (XSS), SQL Injection, Command Injection, CSRF, SSRF, Directory Traversal, Open Redirect, Insecure Server Configurations, and Insecure File Uploads. Utilizing multi- threading, the tool significantly enhances scanning efficiency while preserving accuracy. Furthermore, an automated reporting system generates standardized security assessments, streamlining evaluations for developers and cybersecurity professionals. The scanner's interface also incorporates a built-in feature to directly send reports via email to the concerned client, ensuring smooth communication and quicker responses to security risks. Experimental findings indicate a significant improvement in both scanning speed and detection precision compared to conventional sequential techniques. This study underscores the necessity of automated security testing and introduces a scalable, high-performance solution to strengthen web application security.**

***Keywords*— Web Security, XSS, SQL Injection, CSRF, SSRF, Command Injection, Insecure File Upload, Insecure Server Configuration, Directory Traversal, Open Redirect, Automated Reporting, Threat Intelligence.**

I. INTRODUCTION

Web applications are essential to industries including commerce, banking, healthcare, and governance in the current digital era, but because of their increasing dependence, they are also becoming popular targets for cyberattacks. Common flaws like Directory Traversal, CSRF, SSRF, SQL Injection, XSS, and others can cause serious data breaches, monetary losses, and damage to one's reputation. While firewalls and other traditional security measures are insufficient to prevent sophisticated application-layer assaults, many vulnerability scanners now in use are likewise constrained by their lack of automation, high false positive rate, and poor performance. In order to overcome these difficulties, this project suggests a sophisticated, multi-threaded web vulnerability scanner that can quickly and concurrently scan a large number of websites and identify a variety of serious security flaws.

Grenze ID: 01.GIJET.12.1.548
© Grenze Scientific Society, 2026

A user-friendly interface, email notifications for real-time stakeholder updates, and automatic reporting with severity-based insights are some of the key features. This tool, which is meant to be accurate and scalable, improves the security assessment process for developers, companies, and researchers.

II. Related Work

To help secure web applications, a variety of open-source and commercial web vulnerability scanners have been created. Popular open-source solutions that include a variety of vulnerability detection features include OWASP ZAP, Nikto, and Wapiti. Although OWASP ZAP has an easy-to-use interface and supports both passive and active scanning, it has a high false positive rate and sluggish speeds. Nikto lacks contemporary vulnerability detection and automation, instead concentrating on identifying server configuration errors and obsolete components. Wapiti offers black-box testing for SQL Injection and XSS risks, but its command-line interface and slower performance on dynamic sites are its drawbacks. Acunetix and Burp Suite are notable for their precision and in-depth analysis on the business side. Although Acunetix is expensive and resource-intensive, it provides robust automation and CI/CD integration. Although Burp Suite's setup complexity restricts its utility for quick, automated scans, it is preferred for manual testing and flexibility. Existing scanners have a number of drawbacks despite their advantages, including poor reporting systems, high false-positive rates, slow scanning speeds, and restricted multi-threading capabilities. These shortcomings highlight the urgent need for a contemporary scanner with multi-threading, automation, and real-time reporting capabilities that provides faster, more scalable, and accurate vulnerability evaluations. The goal of this project is to fill these gaps by developing a thorough, cutting-edge web vulnerability detector.

III. Our Methodology

Web applications are vulnerable to numerous security threats, making it essential to implement automated security testing tools. Our multi-threaded web vulnerability scanner is designed to detect critical security flaws, improving both accuracy and scanning efficiency.
This section details the core methodologies used in our scanner, covering URL discovery, scanning techniques, multi-threading, automated reporting, and vulnerability detection mechanisms.

A. System Architecture

Our scanner follows a modular approach, ensuring scalability and maintainability. The architecture consists of:

- URL Discovery Module – Extracts all accessible links from the target website.
- Vulnerability Scanner Module – Detects security flaws using signature-based and behavioral analysis.
- Multi-Threading Engine – Enhances scanning speed by running multiple checks simultaneously.
- Automated Reporting System – Generates detailed reports and emails them to the client.
- Graphical User Interface (GUI) – Provides an intuitive interface for users.

B. Functional Components

URL Discovery

Before scanning for vulnerabilities, the scanner must collect all URLs within the target website. Our system uses web crawling techniques to extract:

- Internal Links – Links within the same domain.
- External Links – Third-party links connected to the target website.
- Dynamic URLs – Links generated via JavaScript.

Methods Used:

- HTML Parsing – Extracts <a href> attributes from web pages.
- JavaScript Rendering – Uses headless browsers to detect dynamically generated links.
- Robots.txt Analysis – Identifies restricted paths.

Multi-Threaded Vulnerability Scanning

Traditional scanners operate sequentially, causing slow performance. Our scanner implements multi-threading to analyze multiple URLs simultaneously, significantly improving speed.
Advantages:

- Faster vulnerability detection
- Efficient resource utilization
- Real-time scanning results

Implementation:

- Thread Pooling – Prevents excessive resource consumption.
- Asynchronous Requests – Enables non-blocking operations for rapid execution.

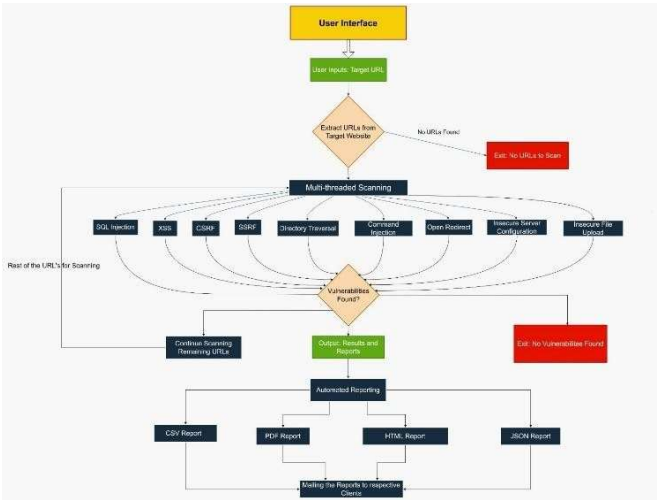

Fig: 1 Methodology

Automated Reporting & Email Notifications

Once scanning is completed, the system generates detailed security reports in multiple formats (PDF, HTML, and CSV) and automatically emails them to security teams.

Report Contents:

- Vulnerability type and severity
- Exploitation risks
- Recommended security fixes

C. Vulnerability Detection & Exploitation

Our scanner detects the following vulnerabilities, analyzing their impact, exploitation methods, and prevention techniques.

Cross-Site Scripting (XSS)

- Impact: Attackers inject malicious scripts into web pages, stealing user credentials.
- Exploitation: Inject <script> tags into input fields to execute unauthorized code.
- Prevention: Implement input validation and use Content Security Policy (CSP).

SQL Injection (SQLi)

- Impact: Attackers manipulate SQL queries to access, modify, or delete database records.
- Exploitation: Inject SQL payloads (' OR '1'='1) to bypass authentication.
- Prevention: Use prepared statements and parameterized queries.

Command Injection

- Impact: Attackers execute arbitrary system commands, gaining unauthorized access.
- Exploitation: Inject commands (; ls -la) into vulnerable input fields.
- Prevention: Restrict system command execution and validate inputs.

Cross-Site Request Forgery (CSRF)

- Impact: Forces users to perform unintended actions on authenticated websites.
- Exploitation: Sends hidden requests (<img src="http://victim.com/delete-account">).
- Prevention: Use CSRF tokens and implement SameSite cookie attributes.

Server-Side Request Forgery (SSRF)

- Impact: Attackers manipulate server-side requests to access internal networks.
- Exploitation: Modify URL parameters (http://localhost/admin).
- Prevention: Implement allow-lists and restrict internal access.

Directory Traversal

- Impact: Attackers access restricted directories and files.
- Exploitation: Modify URLs (../../etc/passwd) to access sensitive files.
- Prevention: Restrict direct file access and use secure path validation.

Open Redirect

- Impact: Redirects users to malicious sites, leading to phishing attacks.
- Exploitation: Manipulate URL parameters (?redirect=malicious.com).
- Prevention: Validate URL redirection and allow only trusted domains.

Insecure File Upload

- Impact: Attackers upload malicious files (backdoors, shell scripts) to exploit the system.
- Exploitation: Upload PHP scripts (shell.php) and execute commands.
- Prevention: Restrict file types and scan uploaded content.

IV. RESULTS

This section presents the results obtained from our multi-threaded web vulnerability scanner, demonstrating how it processes different types of websites. The results are categorized into three variations:

A. User Interface

Main Dashboard

- URL Input Field – Users enter the target website URL for scanning. It ensures correct formatting before initiating the scan.
- "Start Scan" Button – Begins scanning the provided URL for vulnerabilities. A progress bar appears to indicate the scan status.
- "Stop Scan" Button – Allows users to halt the scanning process immediately. It stops all active scanning threads.
- Scan Status Indicator – Displays real-time scanning progress with status updates like "Scanning in Progress," "Scan Completed," or "Scan Stopped."

Reporting Section

- CSV Report – Exports scan results in CSV format, useful for data analysis and custom processing.
- PDF Report – Generates a structured PDF report containing all detected vulnerabilities with descriptions.
- HTML Report – Provides a web-based report for easy viewing and sharing.
- JSON Report – Outputs scan results in JSON format, useful for integration with security tools.
- "Send Report via Email" Button – Sends the generated report directly to the respective client or security team via email.

UI Detailing

Waiting for the Input:

At the initial stage, the scanner waits for the user to input a target URL. The input field allows the user to enter a website link, and once the "Start Scan" button is clicked, the scanning process begins. The UI displays a progress indicator showing that the scan is in progress.

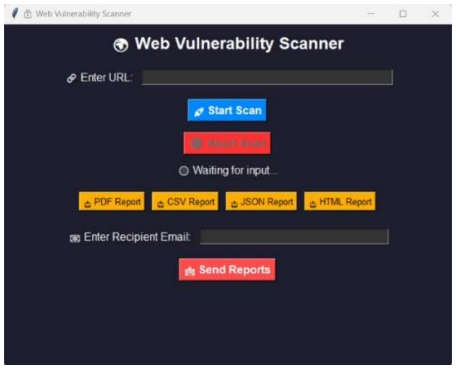

Fig: 2 User Interface

Scan Completed! Download the Reports

Once the scan is complete, the interface provides options to download reports in different formats, such as PDF, HTML, and CSV. A summary of detected vulnerabilities is displayed on the screen, along with options to view detailed reports or send them via email to the client.

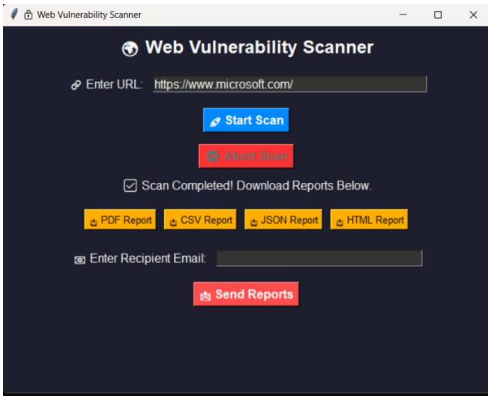

Fig: 3. Scan Completed for the Target URL

B. Results

Unsecured Websites

These are websites with high-risk vulnerabilities detected during the scan. The scanner extracts URLs and identifies security flaws, generating detailed reports.

Scanned Website: http://itsecgames.com/

Extracted URLs Image

- The Discovered URL's of http://itsecgames.com/

Fig: 4 URL's extracted for ITSEC Vulnerable Website

PDF Report

The website http://itsecgames.com/ has been identified as containing specific security vulnerabilities. These issues have been flagged for review.

Fig: 5 PDF Report for the ITSEC Website

The PDF report lists all detected vulnerabilities, their risk levels, and recommended security fixes.

HTML Report

- The website http://itsecgames.com/ provides an HTML report that offers a structured and interactive overview of all security vulnerabilities, simplifying the analysis process.

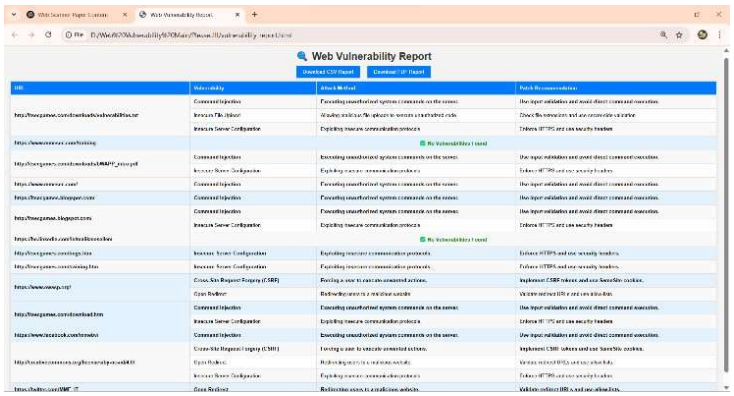

Fig: 6 HTML Report for the ITSEC Website

The HTML report presents a structured, interactive view of all security issues, making it easier to analyze.

C. Mailing the Reports

Sending the Reports to the Client via Email

Once the email is entered, the "Send Report" button allows users to send all generated reports (PDF, HTML, CSV) directly via email. A confirmation message appears upon successful email delivery, ensuring the client receives the scan results.

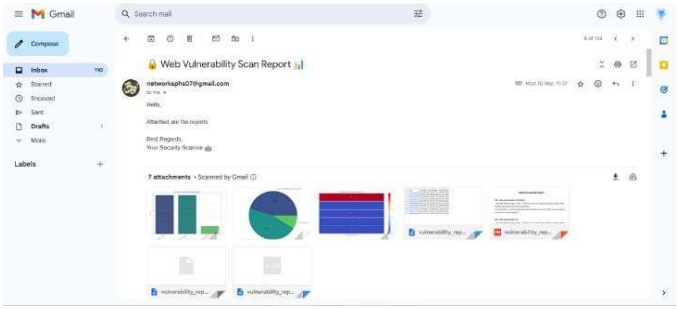

Fig: 7 Reports Sent to the Respective Client's Email ID

V. CONCLUSION

Critical security problems like XSS, SQL Injection, CSRF, and others can be quickly, accurately, and automatically detected with our multi-threaded online vulnerability scanner. It simplifies security evaluations in a variety of online environments by supporting a wide range of reporting formats and email notifications. Its functionality demonstrates how crucial automation is to contemporary cybersecurity. Its capabilities could be further enhanced by incorporating machine learning and real-time threat intelligence, establishing it as a solution for strong web application security that is ready for the future.

FUTURE SCOPE

Our scanner intends to evolve into an AI-powered, cloud-based security platform in order to keep up with changing cyberthreats. Deep learning for zero-day detection, automated patching, and real-time threat intelligence will be the main features of future updates. Its influence will be increased by extending support for blockchain, IoT, mobile, and API technologies. The scanner has the potential to develop into a complete, proactive cybersecurity solution for contemporary digital ecosystems with its intelligent reporting, continuous monitoring, and SIEM integration.

REFERENCES

[1] N. Antunes and M. Vieira, "Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach," IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 269–283, Mar.–Apr. 2015.

[2] M. Curphey and D. A. Scambray, "Web Application Security Assessment Using Automated Tools," IEEE Security & Privacy, vol. 4, no. 4, pp. 32–41, Jul.–Aug. 2006.

[3] S. Choudhary, R. Challa, and M. Gaur, "A Survey of Web Vulnerabilities and Automated Security Assessment Tools," in Proceedings of the 10th International Conference on Cyber Security and Privacy (ICSP), 2022, pp. 51–62.

[4] O. R. Chapple, Cybersecurity Essentials: Protecting Systems and Data, Hoboken, NJ, USA: Wiley, 2018.

[5] K. Patel and R. Gupta, "Enhancing Web Security through Automated Vulnerability Scanners," in Proceedings of the IEEE International Conference on Information Systems and Security (ICISS), 2020, pp. 212–219.

[6] OWASP Foundation, "OWASP Top 10: The Ten Most Critical Web Application Security Risks," Open Web Application Security Project (OWASP), 2021. [Online]. Available: https://owasp.org/www-project-top-ten/

[7] MITRE, "CVE Database: Common Vulnerabilities and Exposures," MITRE Corporation, 2023. [Online]. Available: https://cve.mitre.org

[8] N. S. Ashford and T. R. Wilson, "Machine Learning Approaches for Web Security and Threat Detection," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2198–2213, Aug. 2021.

[9] M. K. Jha and P. K. Verma, "Multi-Threaded Vulnerability Scanners: Performance Analysis and Optimization," in Proceedings of the IEEE International Conference on Software Engineering and Security (ICSES), 2021, pp. 98–105.

[10] S. Y. Wang, "Automated Reporting and Threat Intelligence for Web Application Security," in Proceedings of the IEEE Symposium on Cybersecurity Automation (ISCA), 2022, pp. 185–1