

Web Vulnerability Scanner: A Dockerized Web Vulnerability Scanner and Awareness Dashboard for Proactive Web Security and Education

Prince Sharma

Department of Computer Science
and Engineering, MIT School of Engineering,
MIT Arts Design and Technology University,
Pune, 412201, India
princesharma20003@gmail.com

Lokesh Patil

Department of Computer Science
and Engineering, MIT School of Engineering,
MIT Arts Design and Technology University,
Pune, 412201, India
lokeshpatil702@gmail.com

Aditya Jadhav

Department of Computer Science
and Engineering, MIT School of Engineering, MIT
Arts Design and Technology University,
Pune, 412201, India
jadhavaditya3010@gmail.com

Abstract

Web applications are now an inseparable part of our digital life, they nevertheless suffer from severe security vulnerabilities. Important reasons include unawareness and lack of developer tools. Cyber Enthusiastic tackles this issue by offering a full security environment in Docker. It is equipped with tools such as OWASP ZAP, SQLMap, Nikto, and Arachni, plus a learning dashboard. This system is a great vulnerability scanner that can be used by security enthusiasts, students, and professionals as a learning tool.

Rephrase

Undo

This project takes inspiration from previous research and is aimed at improving the ergonomics and modularity of the scanner. Cyber Enthusiastic provides these tools all in one place, allowing users to take a proactive approach towards web security while better understanding vulnerabilities and how to mitigate them.

I. Introduction

The rapid development of web applications has brought the need for security to the fore. As more organizations switch to digital technologies, the vulnerabilities that accompany these technologies have come to light. Alas, the existing security tools are fragmented and complex, making them difficult for users to utilize. Furthermore, the users struggle to grasp the importance of cybersecurity and the methods to stay secure as there are no guides.

Due to such issues, Cyber Enthusiastic aims to move ever-more closer to a complete cybersecurity ecosystem. The initiative aims to unify open-source scanning tools using Docker to make it easier for users. With the help of this new method, users will be able to interactively participate in cybersecurity activities with ease. In addition, it consolidates many

tools into one place. Having an awareness dashboard allows users to be empowered with real time and educational information.

In the end, the Cyber Enthusiastic aims to clear the air regarding security and make users understand it better. The project hopes to encourage individuals and organizations to focus more on cybersecurity by simplifying tools and resources available. By actively participating in the cyber security knowledge ecosystem, organizations can improve incident response and tackle fraud and threats. This is useful in the current context of cyber threats.

II. Literature survey

Automated Detection and Tool Capabilities.

As per the paper, Bazzoli et al. (2014) [1] carry out black-box vulnerability scanners especially to detect Cross-Site Scripting (XSS) vulnerabilities. The experts found that while Acunetix or Burp Suite tools work well on simple payloads, they don't perform well on advanced obfuscated XSSs. According to the research, ZAP or Nikto should have an adaptive payload strategy and contextual awareness. PreVulnScan intends to use the enhanced configuration of a scan to address this.

Benchmarking of OWASP ZAP and WebGoat.

Potti and his colleagues (2025) [2] benchmarked two versions of OWASP ZAP, 2.12.0 and 2.13.0, against the OWASP Benchmark and WebGoat. The measurements that were compared were true positives, negative false, and scan time. The new improved versions have shown to be much more precise. Hence, PreVulnScan includes such measures to recommend the appropriate scanning profile based on the use case sensitivity, e.g. whether the sensitivity is for penetration testing or compliance audit.

Detection Scope and Tool Coverage.

According to the researchers Jose, Abraham and others, the penetration coverage depth of three popular scanners (W3af, Vega, and ZAP) under a controlled environment with deliberately induced vulnerabilities like SQL injection and command injection was studied. Research shows that no single scanner can detect everything so hybrid or integration of scanners which is used by PreVulnScan detects better when done using combination analysis.

Comparative Usability and Limitations.

Can'o and Camacho et al. (2022) [4] have examined various low-code platforms with special reference to data analysis and their work has been extended to the security tool. The flexibility, usability, and automation of Rapid7, Nexpose, and Nikto were discussed in the work. The review highlighted the advantages of customizable interfaces and plug-in capabilities and this is embedded in PreVulnScan through extensible scanning modules and result visualization layers.

Systematic Analysis of Black-Box Testing Approaches.

Through an empirical evaluation, Doupe et al. (2010) [5] studied five web vulnerability scanners (including Skipfish, Arachni, and Grendel-Scan) using a standardised test suite to analyse the effectiveness of these scanners. They encountered difficulties in handling authentication, managing sessions, and enumerating JS-based attack surface. PreVulnScan has taken these insights on board to add support for a headless browser for scanning SPAs and improve session handling.

Component-based Front-End Integration Using React.

According to Rao and Smith et al. (2021), React utilizes a component-driven model to create user interfaces that respond well. They found that the maintainability and user interactivity improved when adopting tools, such as a vulnerability dashboard. PreVulnScan uses React to create a modular frontend UI to display logs for the scan along with CVSS score, vulnerability graphs and downloadable reports, in real-time.

III. Proposed System

1. The Cyber Enthusiastic framework is a sophisticated, modular web vulnerability scanning solution designed with a focus on clear separation of responsibilities. The Frontend is developed with the help of HTML, CSS, and JavaScript. With the help of it users are able to input target URL and run an overall scan. Moreover, it shows the severity and type of vulnerability discovered. This dashboard comes with

information panels that are interactive in nature to make it contextual with each identified weakness. This would make it a useful tool for professional projects as well as academically. The backend utilizes Python with Flask to manage API logic, orchestrate scans, and process the data. This results in a seamless experience on the application's interface for the end-user.

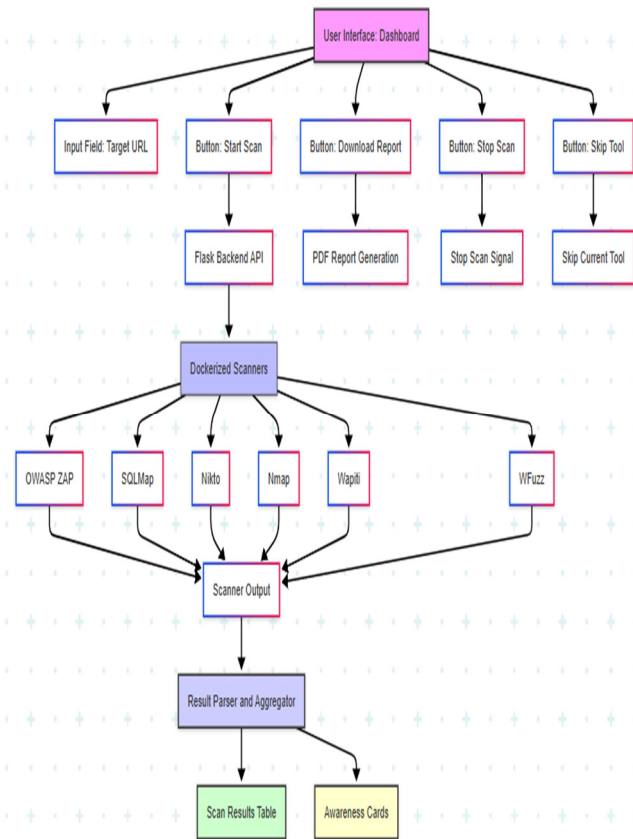


Fig 1: Flowchart of the proposed system

2. At the core of the backend architecture lies a suite of robust open-source security tools, each selected for its specific capabilities in vulnerability detection. The framework uses OWASP ZAP for passive and active scanning. SQLMap is used to identify SQL injection vulnerabilities. Also, Nikto checks for vulnerabilities in server configuration, while Nmap performs network-scanning tasks. Wapiti is included for total black box web application scanning and wfuzz (for brute force of different params and directories). Each tool is a CLI tool that runs in a separate Docker container to make sure it's portable and secure. The backend efficiently activates these scanners depending on the user inputs and gets their outputs. Subsequent to that, it normalizes the outputs into a common JSON schema before sending it back to the frontend. This design gives people insights they can use, but it also helps

them understand how and why the security gets hacked.

IV. Design Plan

1. User Interface (UI):.

The user interface will be created using HTML, CSS, and JavaScript, and it will be responsive and clear on all screens. The UI includes.

A dashboard where users input a target URL.

- Controls.
- Start Scan button to begin the process.
- Stop Scan to interrupt ongoing scans.
- Skip Tool to move past a selected scanning module.
- Download Report to export results.
- A results table that dynamically updates based on scan results by severity (Critical, High, Medium, Low).
- Awareness cards show examples from real life, risk descriptions, and steps you can take to avoid them.
- To show scanning status and error messages, we use simple modal dialogs and toasts.

2. Scan Management Module:.

- The front-end sends a request to the Flask backend API to initiate the scan.
- A URL submitted by the user is sent to the backend via a RESTful POST request.
- Real-time monitoring of scan status is provided, along with the capability to interrupt or bypass individual tools using job IDs or process flags.
- In order to modularise and build fault tolerance the each tool (ZAP, SQLMap, etc.) runs independently.

3. Using Back-end Engine and Scanner

Flask is a Python web framework for making web applications which is powerful, easy to use, and flexible.

- Receiving scan initiation requests.
- Triggering Docker-based scanners.

- OWASP ZAP for passive/active scanning.
- Nikto, Wapiti, Wfuzz, SQLMap, and Nmap as CLI subprocesses.
- We'll jot down the output of each tools as they flow through, parse the relevant data and store in the JSON schema.
- Handling concurrent tasks and execution timeouts for long-running tools.

4. Result Aggregation and Awareness Engine.

- The raw CLI output and ZAP output will be parsed into a JSON format.
- The parsed results include.
- Vulnerability Type.
- Affected Parameter or URL.
- Severity Score (mapped to CVSS where available).
- Tool Name and Description.
- The Awareness Engine checks every finding against a vulnerability database for display.
- Real-world attack examples.
- Prevention/Mitigation tips.
- OWASP Top 10 category alignment.

5. Visualization and Feedback:.

- the sortable column and filter results are shown in the data table.
- Using Chart.JS a pie chart or bar graph can show vulnerability distribution by severity.
- Scanning status of each tool is reflected through the progress bars.
- The success and error states have color codes (green/yellow/red) for a quick understanding.

6. Report Export Functionality:.

- After the scan completes, the user can export.
- A PDF report summarizing all findings, graphs and mitigation measures.

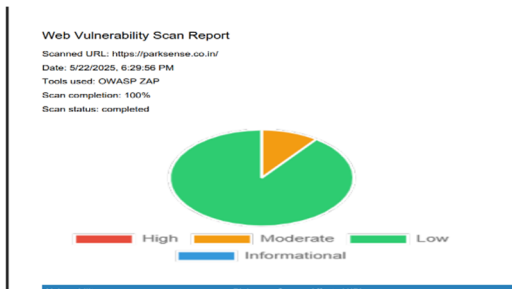


Fig. 7. Detailed Report

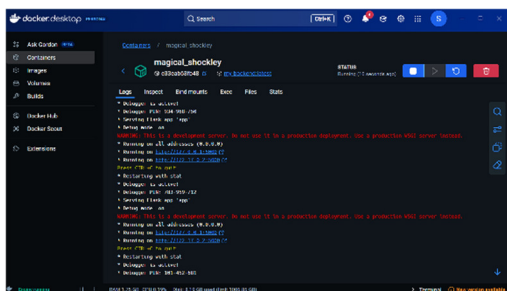


Fig. 8. Docker

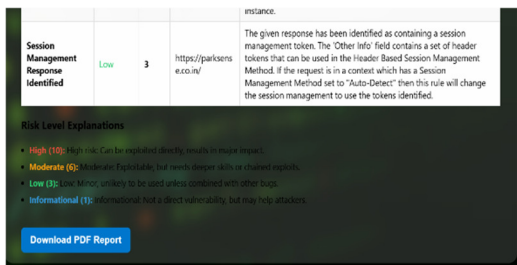


Fig. 9. Download

V. Conclusion

Cyber Enthusiast offers a complete, educational and user-friendly approach to web vulnerability scanning. By using some powerful Open Source tools, namely OWASP ZAP, SQLMap, Nikto, Wapiti, Nmap, and Wfuzz in containerized version, the platform ensures extensive vulnerability coverage and ease in deployment and operation. Flask is used in the backend, which helps in making the scans and communication with the frontend work (HTML, CSS, and JavaScript). This architecture is highly scalable and portable across different operating systems, allowing for compatibility in different user environments.

One major thing that this initiative does is creating security awareness and educating the user. Cyber Enthusiastic is different from scanners that produce

only a technical report. It enhances user experience by providing the result in a user-friendly format along with awareness cards. These cards have some examples along with explaining how a specific vulnerability was found and what can be done to diminish it. With this innovative idea, the platform becomes a detection tool as well as an educational tool. Thus, making it an implementation that is beneficial for cybersecurity professionals, developers, testers, and students who are new to the domain.

Also, it has dynamic scan management features that enable users to stop, skip, or download results etc. Thus, it enhances user experience and control. Often many scanner CLI doesn't have such features. The software can export reports as PDF and CSV files making it even more useful for compliance audits, documentations etc. To sum up, Cyber Enthusiastic connects technical vulnerability assessment with practical security awareness. It offers powerful yet easy-to-use security tools and knowledge enhancement for proactive security measures.

VI. Future Scope

Even though Cyber Enthusiastic is currently working as a local or containerized web vulnerability scanning platform. Its architecture and modular structure has a wide scope for enhancement in the future, especially in the cloud. The platform will become cloud Software as a Service (SaaS). With the scanners and dashboard hosted in the cloud, users will have a dashboard for performing vulnerability scans without any local installation of the tools. This removes the technical burden for end-users and creates a genuinely platform-independent environment, facilitating the inclusion of even more users.

In this proposed model, new users, specifically website owners or developers who are not cyber-savvy, can visit the Cyber Enthusiastic portal, give their desired URL and scan it. The backend cloud instance will execute all configured scanners like OWASP ZAP, Nikto, SQLMap, Wfuzz, etc., in isolated containers or serverless functions. The results from the scans are run live on a secure dashboard where users can see issues ranked by severity, with an easy-to-understand explanation and suggestion to fix.

Implementing authentication mechanisms and usage quotas while balancing loads can ensure responsible and reliable use. The cloud system can also be multi-tenancy. This means that universities, cybersecurity trainers and even organizations can create a sub account for students or employees for training. This would make Cyber Enthusiastic not only a vulnerability scanner but also a learning platform available on the cloud.

At the end, our vision is democratizing web security

by making powerful scanning tools accessible to anyone with a few clicks. No setup, no CLI, no knowledge. With automated scans and contextual guidance, Cyber Enthusiastic can help promote proactive security hygiene in the web development community around the world.

VII.. References

- [1] S. Bairwa, B. Mewara, and J. Gajrani, "Vulnerability Scanners: A Proactive Approach To Assess Web Application Security," IJCSA, vol. 4, no. 1, 2014.
- [2] R. Deeptha et al., "Website Vulnerability Scanner," J. Popul. Ther. Clin. Pharmacol., vol. 30, no. 15, pp. e43–e53, 2023.
- [3] OWASP Foundation, "OWASP Top 10," 2023. [Online]. Available: <https://owasp.org>
- [4] P. Li and B. Cui, "Comparative Study on Software Vulnerability Static Analysis Techniques," IEEE ICITIS, 2010.
- [5] SQLMap, [Online]. Available: <https://sqlmap.org/>
- [6] Nikto, [Online]. Available: <https://cirt.net/Nikto2>
- [7] OWASP ZAP, [Online]. Available: <https://www.zaproxy.org/>
- [8] Arachni Scanner, [Online]. Available: <https://www.arachni-scanner.com/>
- [9] S. Alazmi and D. C. de Leon, "Systematic Review on the Characteristics of WVSs," IEEE, 2017.