

Project Progress Report on

WebSec Scan : A Web Security Scanning Tool

**Submitted in partial fulfilment of the requirement for the award of
the degree of**

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING (CORE)

Submitted by:

Sakshi Lingwal

2019064

Rabia Bakshi

2019257

Under the Guidance of

Mr. Siddhant Thapliyal

Assistant Professor

Project Team ID: ID No. MP24CSE094

Project Progress Report No: 2



**Department of Computer Science and Engineering
Graphic Era (Deemed to be University)**

Dehradun, Uttarakhand

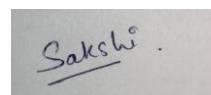
2024-25

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the report entitled “**WebSec Scan : A Web Security Scanning Tool**” in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering(**CSE**) in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **Mr. Siddhant Thapliyal, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

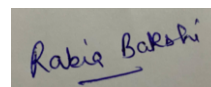
Sakshi Lingwal

2019064



Rabia Bakshi

2019257



The above mentioned students shall be working under the supervision of the undersigned on the “**WebSec Scan : A Web Security Scanning Tool**”.

Signature

Supervisor

Signature

Head of the Department

Examination

Name of the Examiners:

Signature with Date

1.

Table of Contents

Chapter No.	Description	Page No.
Chapter 1	Introduction and Problem Statement	1
Chapter 2	Objectives	2
Chapter 3	Project Work Carried out	3-5
Chapter 4	Future Work Plan	6-7
Chapter 5	Weekly Task	8
	References	

Chapter 1

Introduction and Problem Statement

1.1 Introduction

WebSec Scan is an advanced web security scanning tool designed to help users analyze and identify vulnerabilities in web applications. With cyber threats evolving rapidly, organizations and developers must ensure their websites are secure against potential attacks. This project aims to simplify the security assessment process by offering an automated and user-friendly platform where users can input a website URL and choose from various security scans, such as SQL Injection, Cross-Site Scripting (XSS), DNS Records Analysis, Website Stress Testing, and more. The system processes the selected scan and generates an outline highlighting the vulnerable sections and detected security flaws.

Traditional security testing requires manual effort, in-depth cybersecurity knowledge, and expensive tools, which often limits its accessibility. To bridge this gap, WebSec Scan is designed as an automated web security scanning tool that enables users to analyze a website's security posture effortlessly. The system allows users to input a website URL and select from various security scans to identify vulnerabilities. These scans include:

1. **Basic Scan** – General website security check.
2. **SQL Injection (SQLi) Testing** – Detects database injection vulnerabilities.
3. **Cross-Site Scripting (XSS) Testing** – Checks for script injection vulnerabilities.
4. **DNS Records Analysis** – Extracts domain-related security information.
5. **Full Security Scan** – Performs a comprehensive assessment.
6. **Website Stress Testing** – Evaluates resilience against high traffic loads.
7. **Weak Password Detection** – Identifies weak authentication mechanisms.
8. **Defacement Attack Detection** – Checks for unauthorized website alterations.

By automating security analysis, WebSec Scan provides a streamlined and efficient approach for businesses, security professionals, and developers to proactively safeguard their applications. The WebSec Scan platform integrates multiple security testing methodologies into a single application, making it a powerful tool for proactive cybersecurity. As cyber-attacks become more sophisticated, having an automated security scanner that can detect potential exploits and vulnerabilities in real time is a crucial step toward maintaining a secure web environment.

1.2 Problem Statement

The increasing reliance on web applications for business, communication, and commerce has made them prime targets for cybercriminals. Organizations face constant threats, including SQL Injection, Cross-Site Scripting (XSS), DNS hijacking, defacement attacks, weak authentication mechanisms, and denial-of-service (DoS) attacks. Many businesses, especially small and medium enterprises, lack the technical expertise and resources to conduct in-depth security assessments on their websites. Traditional penetration testing and manual vulnerability assessments are time-consuming, costly, and require skilled security professionals.

Existing challenges include:

- Difficulty in identifying user input parameters and potential attack vectors in web applications.
- Manual testing processes that are time-consuming and prone to human error.
- Lack of affordable tools that can detect and address critical vulnerabilities like SQL Injection and XSS.
- Limited awareness and resources for smaller teams to implement security best practices.
- Inefficiency in scaling security assessments for larger, enterprise-level applications.

WebSec Scan addresses this issue by providing an automated, efficient, and accessible web security scanning tool that allows users to identify and assess vulnerabilities without requiring extensive cybersecurity knowledge. This tool is designed to be an easy-to-use platform where users simply input a website URL, select the type of scan they want to perform, and receive a comprehensive security report.

Chapter 2

Objectives

The proposed work objectives are as follows:

1. Security Awareness :

- To raise awareness about the importance of website security by detecting common vulnerabilities and providing users with insights into potential risks.

2. Website Security Assessment :

- To offer a simple yet comprehensive tool for website administrators, developers, and security professionals to test the security health of websites.

3. Vulnerability Identification :

- To detect specific vulnerabilities such as SQL injection (SQLi), Cross-Site Scripting (XSS), weak passwords, and more, and provide a detailed report on each issue.

4. Stress Testing :

- To test the robustness of websites under heavy traffic, helping organizations identify weaknesses in their infrastructure and improve resilience.

5. Automated Scanning :

- To automate security scanning and generate actionable reports with minimal manual intervention, making it easy to monitor websites' security regularly.

6. Monitoring Weak Passwords :

- The tool helps monitor the security of login mechanisms, alerting the user about weak passwords or common attack vulnerabilities, encouraging better authentication practices.

7. Comprehensive Security Audit:

- The full security scan option provides a thorough evaluation of all key website security aspects, from data injection vulnerabilities to external security threats. This helps ensure a complete security posture check.

8. Defacement Detection:

- Organizations that are concerned about unauthorized changes to their websites can use the tool to regularly check if any content has been altered, ensuring that their site remains secure from defacement.

Chapter 3

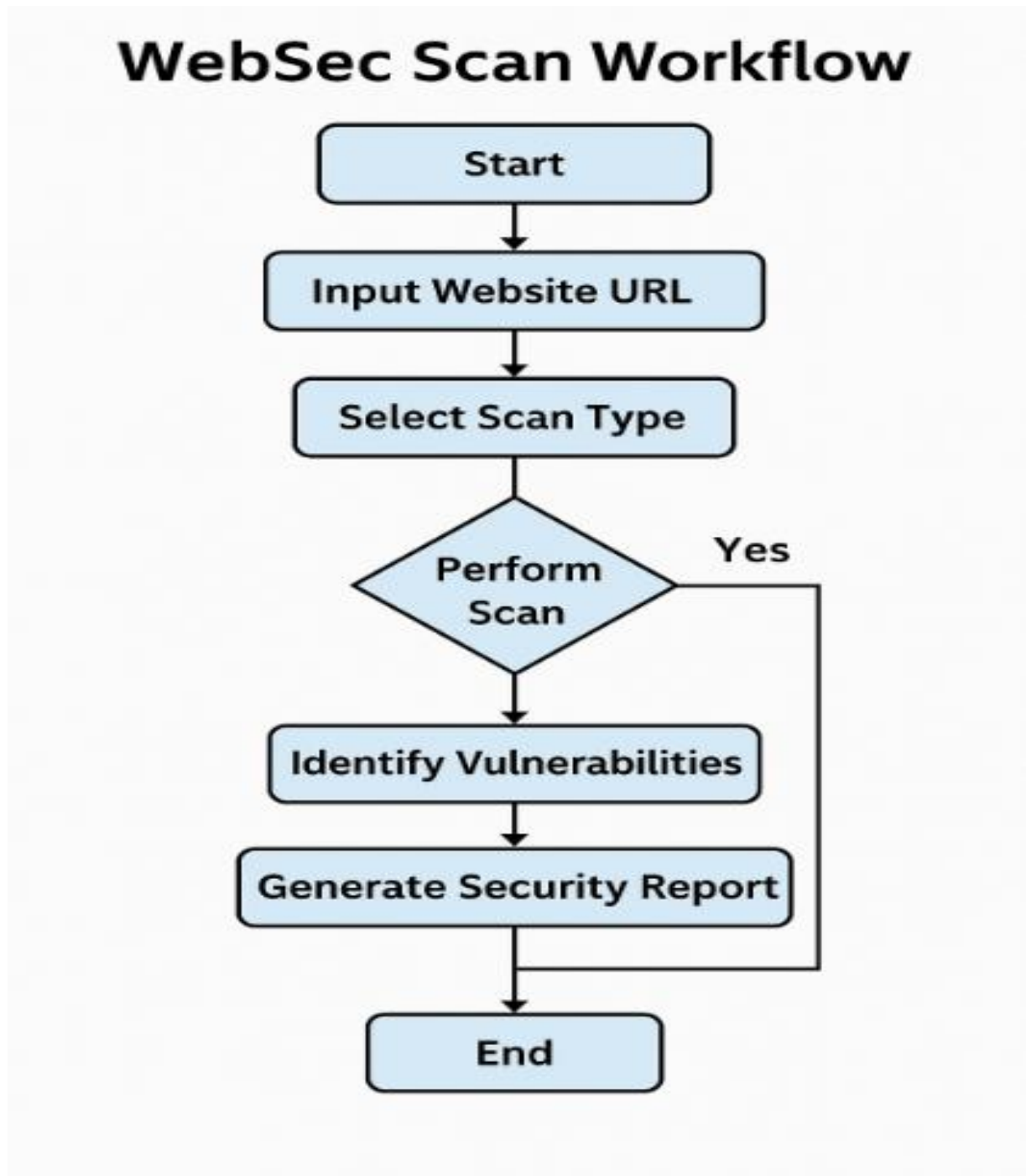
Project Work Carried Out

Steps Involved in the Project Development :

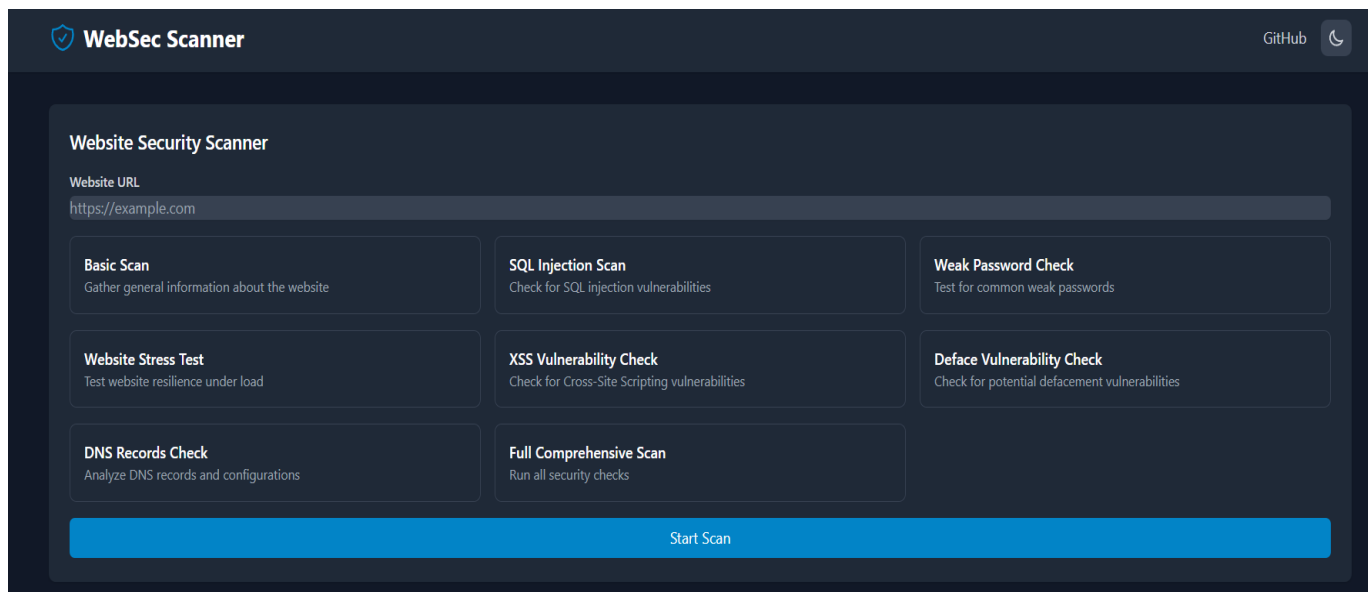
1. **Requirement Analysis and Research :** The first step involved conducting thorough research on common web vulnerabilities, particularly focusing on the OWASP Top 10 security risks. We identified key areas such as SQL Injection, Cross-Site Scripting (XSS), weak password detection, DNS records analysis, and website defacement attacks. This helped in defining the project scope and determining the features to be included.
2. **Designing the Architecture:** After gathering the requirements, we designed the overall architecture of the tool, including the backend, frontend, and communication between components. The backend would handle the vulnerability scanning and analysis, while the frontend would allow users to interact with the tool via a user-friendly interface.
3. **Backend Development:** The backend was built using programming languages like Python, integrating libraries such as `dns.resolver`, `whois` and `Requests` for web scraping and HTTP requests. We developed various scanning modules for SQLi, XSS, DNS records analysis, stress testing, and weak password detection, making sure each scan was customizable based on the user's needs.
4. **Vulnerability Testing and Logic Implementation:** For each vulnerability scan, we implemented specific logic. For SQLi, we used known attack vectors and payloads to test if the site was vulnerable to database injections. For XSS, we injected script tags into input fields to see if the website sanitized them properly. Similarly, DNS analysis was carried out using DNS query tools, and stress tests were performed by simulating high traffic loads on the website.
5. **Frontend Development:** A web-based interface was created to allow users to input website URLs, select different types of scans, and view results. The frontend was designed to be intuitive, providing a smooth user experience with clear and actionable results, including detailed information about the identified vulnerabilities.
6. **Integration and Communication Between Frontend and Backend:** The frontend was integrated with the backend through API calls. This allowed users to trigger scans directly from the user interface, which would then send the website URL to the backend for processing. Once the scan was completed, the backend returned the results, and the frontend displayed them in an easy-to-understand format.
7. **Deployment and Final Adjustments:** After final testing, the tool was deployed on a local server, and any necessary adjustments were made based on user feedback.
8. **Ongoing Maintenance and Updates:** Once the tool was deployed, we continued to maintain and update it by adding new features, improving scanning accuracy, and addressing emerging vulnerabilities to ensure the tool remains effective against evolving web security threats.

This project can be used for identifying vulnerabilities that attackers could exploit to gain unauthorized access, inject malicious scripts, or compromise the system. By addressing these vulnerabilities, we ensure that the system is secure and resilient against common threats.

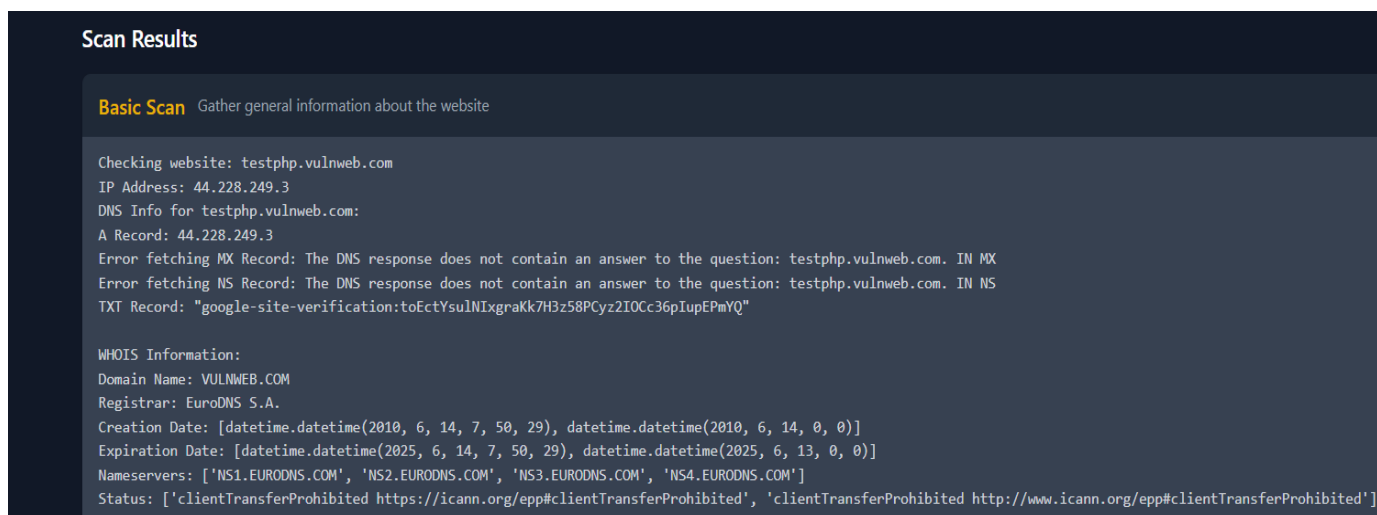
- **Workflow of the project :**



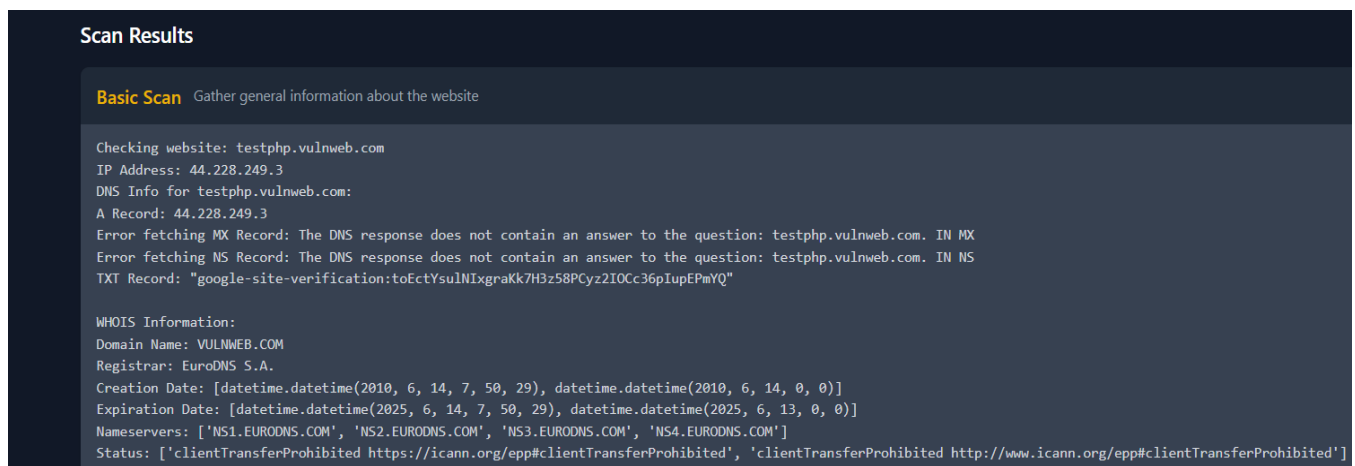
- **Frontend of the WebSec Scan :**



- **Output for a Vulnerable Website:**



- **Output For a Non-Vulnerable Website:**



Chapter 4

Future Work Plan

The future work plan of our project are as follows:

Sl. No.	Work Description	Duration in Days
1.	Real-time Web Application Testing: integrate the tool with real-time threat intelligence feeds to enhance its vulnerability detection capabilities.	20
2.	Machine Learning: implement machine learning models to identify and classify vulnerabilities based on patterns in data or known exploits.	12
3.	Automation and Scheduling: Enable scheduled scans or automation via a task scheduler to scan regularly.	15
4.	Reporting and Alerts: Generate detailed, user-friendly reports in PDF or HTML format summarising the findings.	7
5.	User Interface (UI): Develop a web-based or graphical user interface (GUI) to make the tool more user-friendly and accessible for non-technical users.	15

Chapter 5

Weekly Task

The report of project work allocated by the supervisor is as follows:

Week No.	Date: From-To	Work Allocated	Work Completed (Yes/No)	Remarks	Guide Signature
1	24-10-24 to 31-10-24	Project Planning & Setup	Yes		
2	01-11-24 to 20 -11-24	Backend Setup & Basic Scanning Logic	Yes		
3	21-11-24 to 01-12-24	SQL Injection (SQLi) Testing & Cross-Site Scripting (XSS)	Yes		
4	02-12-24 to 05-01-25	DNS Records Analysis & Full Security Scan	Yes		
5	06-01-25 to 25-01-25	Weak Password Detection & Stress Testing	No		
6	26-01-25 to 10-02-25	Website Defacement Detection & Frontend Development (Initial Phase)			
7	11-02-25 to 02-03-25	Frontend Development (Advanced Features)	Yes		
8	03-03-25 to 30 -03-25	Testing & Validation	Ongoing		

References

1. Owasp : https://owasp.org/www-community/Vulnerability_Scanning_Tools
2. SQL Injection : OWASP SQL Injection Guide: https://owasp.org/www-community/attacks/SQL_Injection
3. Cross-Site Scripting (XSS) : OWASP XSS Guide: <https://owasp.org/www-community/attacks/xss/>
4. Weak Passwords & Authentication Vulnerabilities : NIST Guidelines on Password Security: <https://pages.nist.gov/800-63-3/sp800-63b.html>
5. OWASP Defacement Attack Reference : https://owasp.org/www-community/attacks/Web_Defacement
6. Website Stress Testing (DDoS & Load Testing) : <https://www.cloudflare.com/learning/ddos/glossary/stress-test/>
7. https://www.researchgate.net/publication/378491802_Web_Vulnerability_Scanning_Tools
8. <https://www.mdpi.com/2079-9292/12/20/4299>
9. <https://www.arcjournals.org/pdfs/ijrscse/v10-i1/2.pdf>