**SYNOPSIS**

**SEMINAR**

On

**XSSecure**

WebApp Vulnerability Scanner using XSS

MASTER OF COMPUTER APPLICATION

Of

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

By

**NAME: CHANDAN L**

**USN: 1MV23MC012**

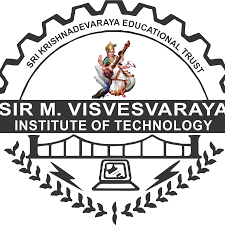
**Internal Guide**

Mr. VASANTHA S

Assistant Professor

Dept. of MCA, Sir MVIT

Bangalore-562157



**SIR M. VISVESVARAYA INSTUTE OF TECHNOLOGY**

HUNASAMARANAHALLI, VIA: YELAHANKA,

BANGALORE-562157

**SIR M. VISVESVARAYA INSTUTE OF TECHNOLOGY**

Department of Computer Application

**Seminar Synopsis**

|  |  |
| --- | --- |
| **Name of the Student** | CHANDAN L |
| **USN** | 1MV23MC012 |
| **Project Name** | XSSecure |
| **Estimated Duration** | July 2023 to October 2024 |
| **Name of the Organization and Address** | Sir MVIT, Dept of MCA |
| **Description of the Project** | XSSecure is a specialized web application vulnerability scanner designed to detect and analyze cross-site scripting (XSS) vulnerabilities.XSSecure aims to empower developers and security professionals with a powerful tool to identify potential XSS threats and safeguard their web applications from malicious attacks. |
| **Project Domain** | Cybersecurity |

**Problem Statement**

As web applications continue to evolve and become more complex, they are increasingly targeted by cyberattacks, particularly Cross-Site Scripting (XSS) vulnerabilities. XSS attacks enable malicious actors to inject harmful scripts into web pages viewed by other users, leading to unauthorized data access, session hijacking, and significant security breaches.

The challenge lies in the fact that many developers and security professionals lack efficient tools specifically designed to identify and mitigate XSS threats within their web applications. This gap leaves web applications vulnerable to potential attacks, compromising user data and threatening the overall integrity and security of the system. **XSSecure** aims to address this critical issue by providing a specialized, user-friendly, and comprehensive tool tailored for detecting and analysing XSS vulnerabilities in web applications.

**Objectives**

The primary objective of the **XSSecure** project is to develop a specialized web application vulnerability scanner that focuses on detecting and analysing Cross-Site Scripting (XSS) vulnerabilities.

1. **Automate XSS Detection:** Create an automated tool that performs comprehensive scans of web applications to identify reflected, stored, and DOM-based XSS vulnerabilities efficiently.
2. **Provide Detailed Reporting:** Deliver detailed, actionable reports that outline the detected vulnerabilities, including the number of executed and non-executed payloads and their impact, to help users understand and address security issues effectively.
3. **Enhance Web Application Security:** Equip developers and security professionals with an easy-to-use and improving the overall security of web applications.
4. **Monitoring:** Offer features like a user dashboard to track and review the history of scanned URLs and their vulnerabilities, supporting ongoing security assessments and improvements.

**Existing Systems**

**1. Burp Suite:** A comprehensive web vulnerability scanner with XSS detection capabilities. However, it is expensive and may be overwhelming for beginners.

**2. OWASP ZAP:** A free tool that offers XSS scanning among other features. It's powerful but can be complex to configure for specific use cases.

**3. Acunetix**: A paid, automated web application security scanner that includes XSS detection. It is highly effective but costly.

**Complexity:** Many tools require extensive configuration and a steep learning curve, which can be a barrier for less experienced developers’ dels with many numbers of vulnerability to is difficult for an average developer.

**Cost:** High-quality tools often come with significant costs, making them inaccessible to smaller teams or individual developers.

**Proposed Solution**

**XSSecure** provides an effective solution for detecting Cross-Site Scripting (XSS) vulnerabilities in web applications. The tool features an automated scanning engine that simulates browser interactions to identify XSS vulnerabilities across various input fields. It offers a user-friendly interface for initiating scans and viewing results, along with comprehensive reports that include detailed vulnerability analysis and visualizations. The system supports secure user authentication and maintains a dashboard for tracking scan history. **XSSecure** integrates advanced technologies to deliver a robust and scalable tool for enhancing web application security.

**Technology used**

 **HTML & CSS (Tailwind):** HTML structures the web pages, while CSS (with Tailwind) styles them to be visually appealing and responsive.

 **JavaScript:** The core language used for adding interactivity and functionality to the web application, working alongside React to enhance user experience.

 **Angular JS:** A library for building interactive and dynamic user interfaces, making the frontend of **XSSecure** user-friendly and efficient.

 **Node.js:** Allows us to run JavaScript on the server side, handling the backend processes and logic for **XSSecure**.

 **Express.js:** A framework for Node.js that simplifies the creation of server-side routes and APIs, making it easier to manage requests and responses.

 **My SQL:** A database that stores scan results and user data, providing a flexible and scalable solution for data management.

 **Chart.js:** A library used to create charts and graphs, helping to visually present the scan results and vulnerability data.

 **Cheerio.js:** A tool for parsing and manipulating HTML on the server side, used during the web scraping and scanning processes.

 **Puppeteer:** A library for controlling web browsers, used to automate web interactions and perform scans for vulnerabilities.

 **JWT (JSON Web Tokens):** Used for secure user authentication and authorization, ensuring that only authorized users can access the **XSSecure** platform.

**Hardware Requirements**

**Processor:** i3 or above

**RAM:** 4 GB or above

**Storage:** 20 GB or above (SSD preferred)

**Network:** Stable internet connection

**Software Requirements**

**Code Editor:** VS Code

**Operating System:** Windows 10 or above / macOS / Linux

**Node.js:** Latest LTS version

**Package Manager:** npm

**Database:** MongoDB

**Browser for Testing:** Chrome, Firefox or any

**Version Control:** Git

Signature of Student Signature of Guide

Date: Date:

(Seal and signature of Department Head)

Date: