**National University of Computer & Emerging Sciences Karachi Campus**

****

**(CYL4001) Secure Software Design**

## **Secure Software Design Lab (SSD-Lab) Project**

**Shah Muhammad Qureshi 21k-3557**

**Ali Mehdi Badami 21k-4771**

**Date**: November 26, 2024

### **1. Introduction**

The **XSS Testing Tool** is an advanced utility designed to identify Cross-Site Scripting (XSS) vulnerabilities in web applications. XSS attacks allow attackers to inject malicious scripts into websites, which can compromise user data, steal session tokens, or manipulate site content. This tool automates vulnerability detection for three primary types of XSS: **Reflected**, **Stored**, and **DOM-based XSS**, providing security professionals with actionable insights for strengthening web defenses.

### **2. Objectives**

* Automate detection of **Reflected**, **Stored**, and **DOM-based XSS** vulnerabilities.
* Identify exploitable input fields in web applications.
* Streamline payload injection and response analysis.
* Generate detailed documentation for vulnerability remediation.

### **3. Working**

The XSS Testing Tool operates through the following steps:

1. **Web Crawling**:  
   * The tool scans the target site for forms and links to identify potential input points.
   * It extracts form data, including actions, methods, and input fields.
2. **Payload Injection**:  
   * Predefined malicious payloads are injected into form inputs or URL parameters to simulate XSS attacks.
   * Both GET and POST methods are tested for form submissions.
3. **Response Analysis**:  
   * The tool checks server responses for payload execution, indicating an XSS vulnerability.
   * It categorizes findings into **Reflected**, **Stored**, and **Safe** states.
4. **Manual Testing for DOM-based XSS**:  
   * The tool constructs URLs containing payloads and provides them for manual verification in browser consoles.

**Working Diagram**: *(Insert a flow diagram illustrating the steps: Crawling → Payload Injection → Response Analysis → Documentation.)*

### **4. Tool Testing**

**Environment**:

* Tested on **Windows 11**.
* Dependencies: Python 3.10, requests, bs4, and art libraries.

**Evidence of Success**:

* Successfully detected Reflected XSS in sample test websites.
* Detected unsafe form inputs and vulnerable endpoints on testing platforms such as DVWA (Damn Vulnerable Web Application).
* Generated detailed reports with timestamps in a structured format.

### **5. Limitations**

* **Manual Verification for DOM-based XSS**: The tool requires manual inspection of browser console logs to confirm DOM-based XSS.
* **False Positives**: Automated detection may sometimes flag non-vulnerable inputs as unsafe.
* **Restricted Crawling**: Websites with CAPTCHA or authentication requirements may limit the tool's effectiveness.
* **No Real-Time Monitoring**: The tool only tests static scenarios, not live user interactions.

### **6. Conclusion**

The XSS Testing Tool is a robust solution for automating the detection of XSS vulnerabilities. It simplifies security assessments by integrating payload injection and response analysis into a single workflow. Through this project, we gained insights into web security testing, vulnerability exploitation, and the importance of comprehensive reporting for vulnerability management. The tool is a step forward in ensuring web applications are resilient to XSS attacks, protecting user data and system integrity.

### **7. References**

* OWASP Foundation. (n.d.). *Cross-Site Scripting (XSS)*. Retrieved from [https://owasp.org](https://owasp.org/)
* PortSwigger. (n.d.). *XSS vulnerabilities*. Retrieved from [https://portswigger.net](https://portswigger.net/)
* Sahay, P. (2021). *Python Web Penetration Testing Cookbook*. Packt Publishing.