Web Vulnerability Scanner Report

Developed for OWASP Top 10 Vulnerability Testing

Date: July 8, 2025, 08:49 PM IST

Submitted by: Laba Kumar Kalita

Contents

1 Introduction

This report details a web vulnerability scanner developed to identify OWASP Top 10 vulnerabilities, focusing on Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF). Tested on the Damn Vulnerable Web Application (DVWA) hosted via XAMPP at `http://localhost/DVWA-master/DVWA-master/`, the scanner addresses challenges like slow scan times, incorrect vulnerability reporting (e.g., HSTS linked to `login.php`), and template errors, delivering a user-friendly tool with a downloadable report.

2 Abstract

The scanner, built in Python, crawls websites, injects payloads, analyzes responses and scripts, and displays results in a browser with a downloadable `report.txt`. Key features include:

- Detection of XSS (reflected and DOM-based), CSRF, SQL Injection, and Cryptographic Failures.
- Performance optimized by limiting payloads (3), forms (1), inputs (1), and using a 3-second timeout.
- Fixes for Jinja2 `strftime` error and incorrect `form_action` reporting.
- Centered UI in `index.html` and downloadable report in `results.html`.

Tested on DVWAs `/vulnerabilities/csrf/` and `/xss_d/`, it successfully identified vulnerabilities like missing HSTS headers and DOM-based XSS.

3 Tools and Technologies

The scanner uses:

- **Python**: Core logic.
- **Flask**: Web interface and scan endpoint.
- **Requests**: HTTP requests for crawling.
- **BeautifulSoup**: HTML parsing for forms, inputs, scripts.
- **HTML/JavaScript**: Centered UI and report download.
- **XAMPP**: Hosts DVWA locally.

4 Implementation Steps

Development involved:

1. **Crawling**: Extracts forms, `<select>` inputs, links, headers, and scripts.
2. **Payload Injection**: Tests payloads like `<script>alert('XSS')</script>` for XSS and SQLi.

3. **Vulnerability Detection**: Checks server responses for reflected payloads and scripts for DOM-based XSS (e.g., `eval`).
4. **Optimization**: Limits requests to reduce scan time to under 3 seconds.
5. **UI Enhancements**: Centers `index.html` and adds report download in `results.html`.
6. **Error Fixes**: Resolved `strftime` error with JavaScript timestamp and fixed HSTS `form_action` misreporting.

5 Challenges and Solutions

Key challenges addressed:

- **Slow Scans**: Limited payloads and forms reduced scan time.
- **DOM-based XSS**: Detected via script pattern analysis (`eval`, `innerHTML`).
- **Incorrect URL Reporting**: Fixed HSTS linking to `login.php` by adding `url` field.
- **Template Error**: Replaced `strftime` with JavaScript `Date.toISOString()`.

6 Conclusion

The scanner effectively identifies OWASP Top 10 vulnerabilities on DVWA, with optimized performance and accurate reporting. Future enhancements could include advanced CSRF detection. The project, tested on `http://localhost/DVWA-master/DVWA-master/vulnerabilit` and `/xss_d/`, demonstrates robust web security testing.