Introduction

The Web Application Vulnerability Scanner is a Python project I built as part of the Elevate Labs Cybersecurity Internship in June 2025. The tool is designed to detect basic but common web vulnerabilities like **Cross-Site Scripting (XSS)** and **SQL Injection (SQLi)** by automatically scanning and testing forms on web pages. This project helped me get hands-on experience with ethical hacking concepts and understand how vulnerabilities can be discovered in a real-world setup.

Abstract

This scanner automates the process of testing web input forms for potential vulnerabilities. It works by scanning a webpage, extracting all forms, and injecting test payloads into any text fields. The form is then submitted using either `GET` or `POST`, depending on what the form uses. Once submitted, the response is analyzed to see if the payload appears — which would indicate a possible vulnerability.

I tested this tool on two safe, purposely vulnerable websites:

- `http://testphp.vulnweb.com/`
- `https://xss-game.appspot.com/level1/frame`

These sites allowed me to simulate how attacks like XSS or SQLi could occur without violating any ethical or legal boundaries.

Tools Used

- **Python 3** – Main programming language
- **requests** – To send web requests
- **BeautifulSoup** – To find and analyze HTML forms
- **urllib.parse** – To properly build the form submission URLs
- **Linux Terminal** – To run and test the script

Steps Involved in Building the Project

1. **Set up the Python environment** and installed the required libraries.
2. **Fetched HTML content** from a target URL using `requests`.
3. **Parsed the HTML** using `BeautifulSoup` to extract all form tags.
4. **Extracted form data** including the method (GET/POST), action URL, and input fields.
5. **Inserted payloads** into text inputs:
    - XSS: `<script>alert(1)</script>`
    - SQLi: `' OR '1'='1`

6. **Submitted forms** and captured the responses.
7. **Analyzed the results** by checking whether the payload was reflected in the response or triggered any error messages.
8. **Displayed results** in the terminal showing whether a potential vulnerability was detected.

Conclusion

This project helped me understand how attackers can exploit input fields and how developers should protect against that. Building this scanner strengthened my Python skills, especially in handling web requests and HTML parsing. It also gave me a clear picture of how ethical hacking tools work behind the scenes. This scanner is a great starting point for learning about web security and makes a solid project to showcase in interviews or on GitHub.

Project By:

Aravind Kumar M
Elevate Labs Cybersecurity Internship — June 2025