───────────────────────────────────────────────

**Summer Internship Project Report**

**PayloadGeneration + ZAP Integration Toolkit**

───────────────────────────────────────────────

Submitted by:Unaiza Fatima

Program: BS Cyber Security

Internship Organization: ITSOLERA Cyber Department

Date: July 04, 2025

#### Supervised By:

**Muhammad Ahsan Aya**z  
Cyber Security Department

ITSOLERA Summer Internship Program 2025

|  |
| --- |
| ****Table of Contents**** |
| 1. **Introduction** |
| 1. **Objectives** |
| 1. **Tools & Technologies Used** |
| 1. **Project Architecture** |
| 1. **Implementation Steps** |
| 1. **Screenshots** |
| 1. **Reflection & Learning** |
| 1. **Conclusion** |

# 1. Introduction

This project, titled **PayloadGen**, aims to simplify and automate the offensive security process by generating custom payloads and integrating them with **OWASP ZAP (Zed Attack Proxy)** to perform vulnerability scanning on target web applications. The tool provides a user-friendly graphical interface that allows security analysts and ethical hackers to generate various types of payloads such as **XSS**, **SQL Injection**, and **Command Injection** and then launch an active scan using ZAP with selected scanner rules.

By combining payload generation with automated scanning, PayloadGen helps speed up the penetration testing process and supports better vulnerability management for web applications. This project was developed as part of the ITSOLERA Summer Internship Program under the Cyber Security Department.

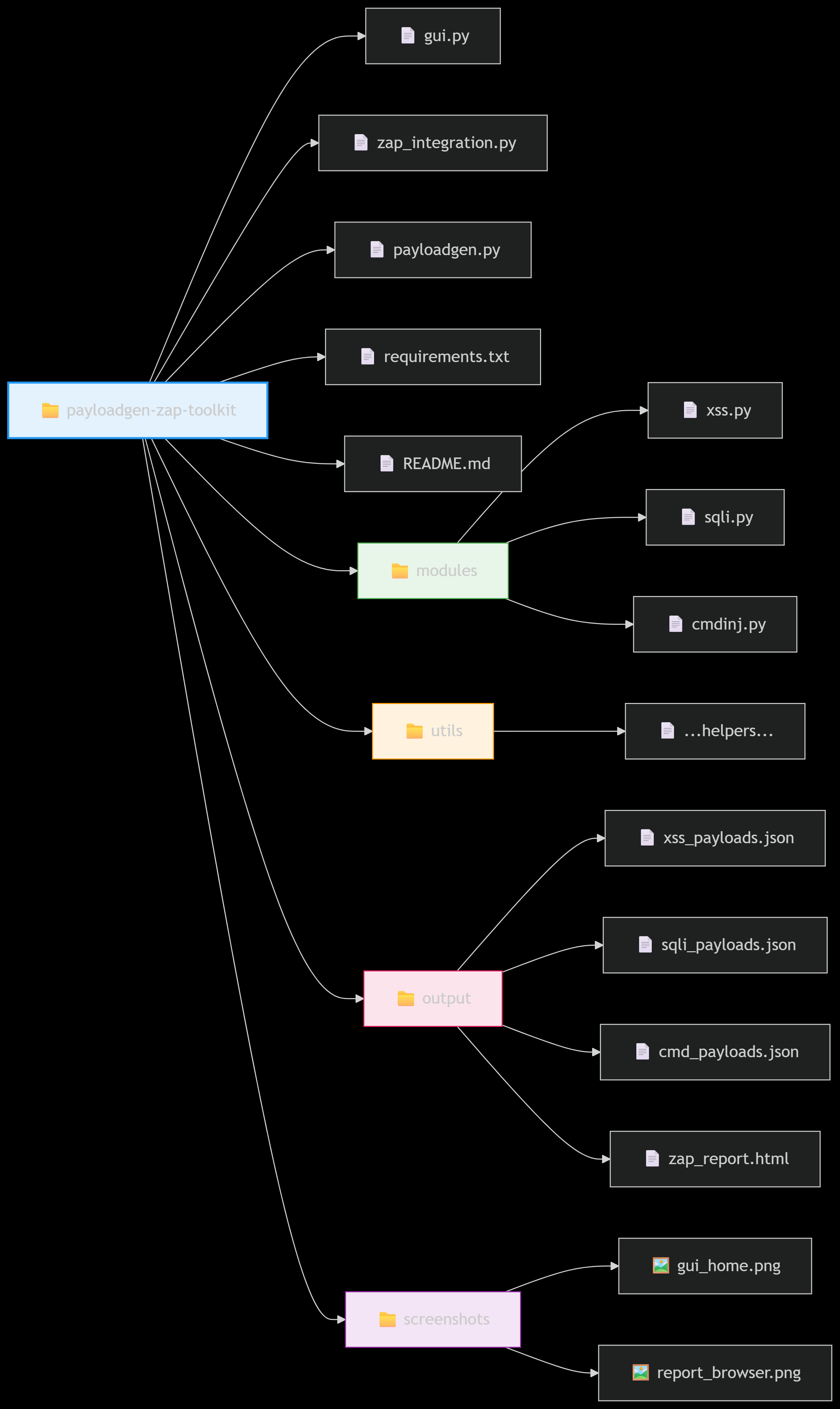
# 2. Objectives

* **Payload Automation**: Generate test payloads for XSS, SQL Injection, and Command Injection attacks.
* **ZAP Integration**: Use OWASP ZAP’s API to scan target web applications with selected payloads.
* **GUI-Based Control**: Provide a simple interface for launching scans without using the terminal.
* **Performance Optimization**: Implement timeout logic and enable selective scanners to avoid delays.
* **Report Generation**: Export clean, readable HTML scan reports.
* **Practical Offensive Use**: Assist in red teaming, penetration testing labs, and CTF automation tasks.

# 3. Tools & Technologies:

|  |  |
| --- | --- |
| Language | Python 3 |
| Libraries | tkinter, zapv2, os, webbrowser, time |
| Platform | Kali Linux |
| External Tool integrated | OWASP ZAP |

# 4. Project Architecture:

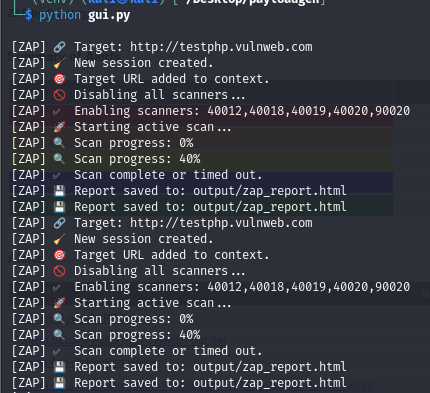


# 5. Implementation Steps:

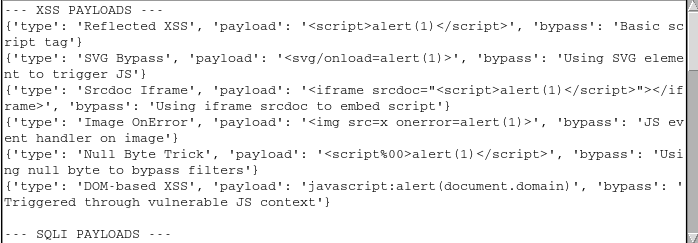
1. Launch gui.py  
2. View or generate payloads (XSS, SQLi, CMD)  
3. Enter a target URL  
4. Run ZAP scan with selected scanner rules  
5. View generated HTML report  
6. Use 'Show Report' button in GUI to open results

# 6. Screenshots:

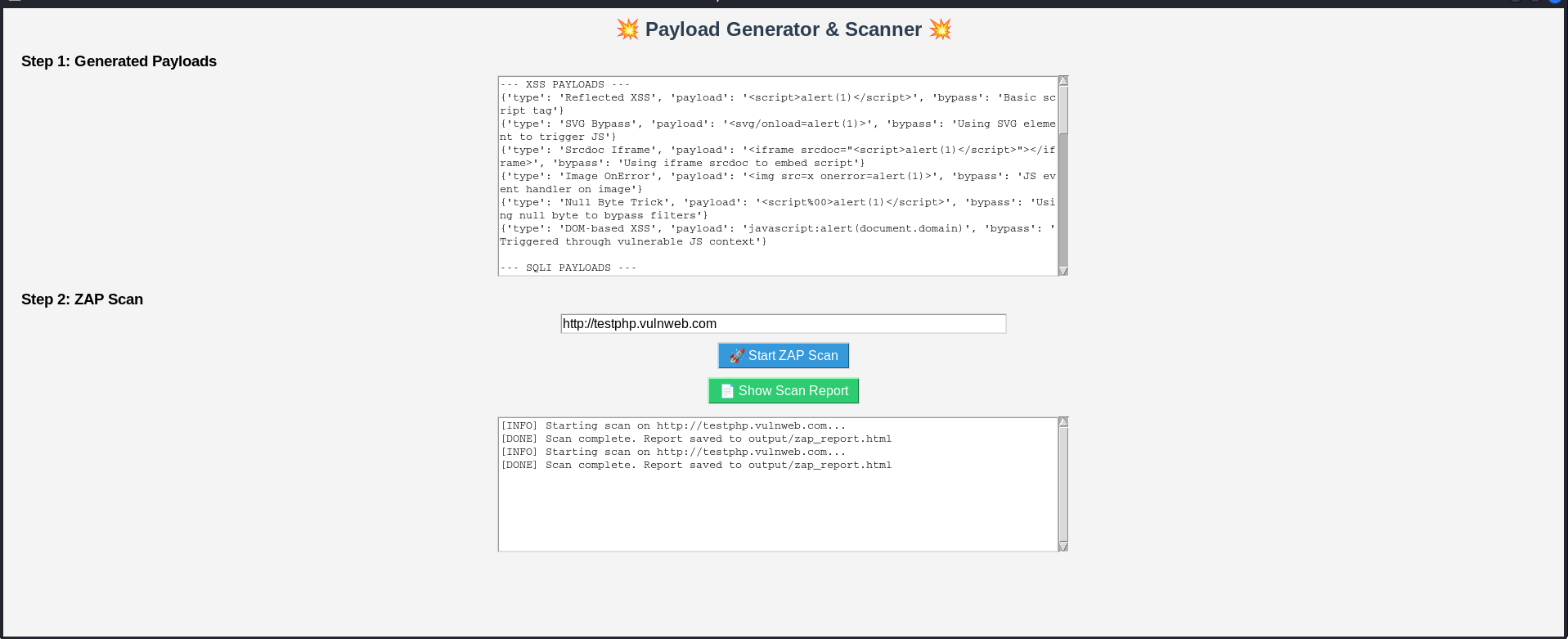
# - GUI:



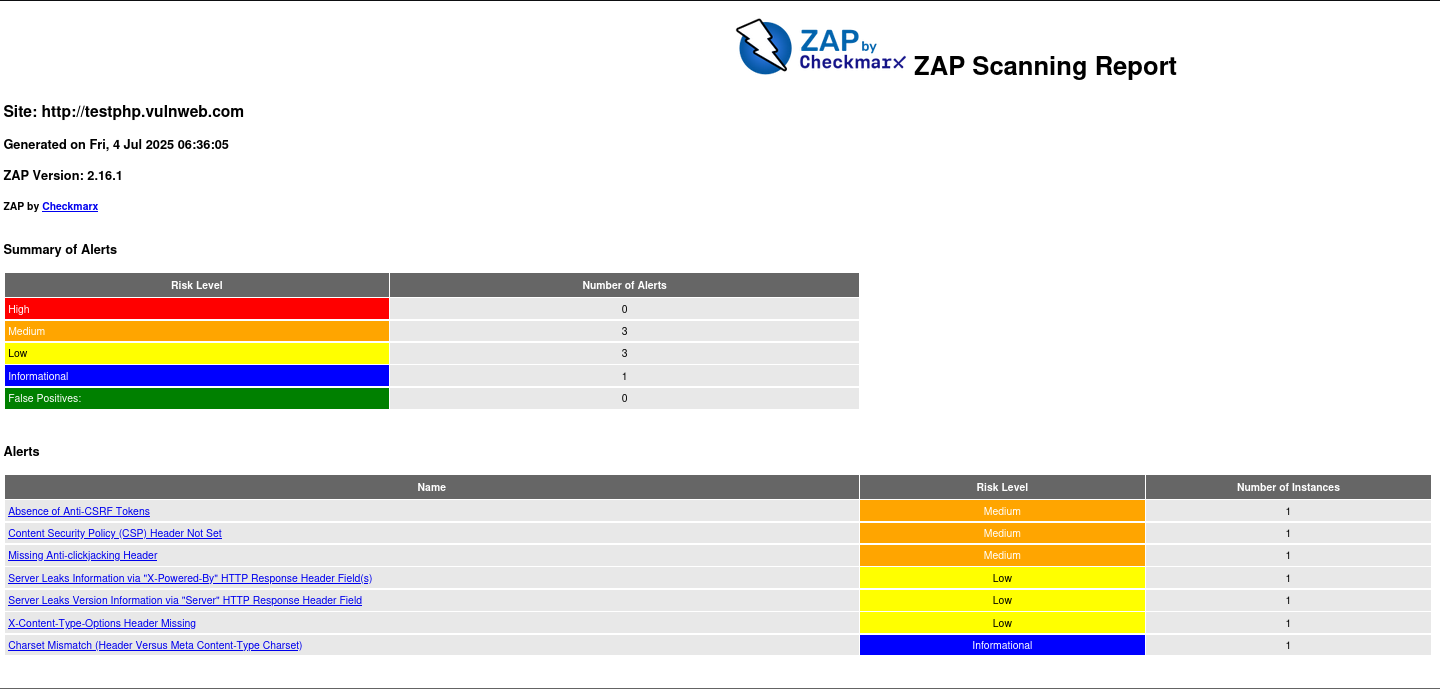
Generated payloads:



Enter targeted URL:



HTML REPORT:



**7. Reflection & Learning**

This project significantly enhanced my understanding of automation in cybersecurity, especially integrating OWASP ZAP with Python for web application scanning. I strengthened my skills in scripting, GUI development using Tkinter, and vulnerability detection workflows.

It also taught me the importance of clean code, user-friendly interfaces, and handling real-world issues like scan delays and integration bugs. Overall, this project helped me grow both technically and professionally.

# 8. Conclusion:

The PayloadGen + ZAP Toolkit provided valuable hands-on experience in building a functional security tool from scratch. It reinforced my learning of offensive security concepts and laid the groundwork for future work in red teaming, penetration testing, and automation. This project demonstrated how scripting and open-source tools can be combined to streamline essential security tasks.