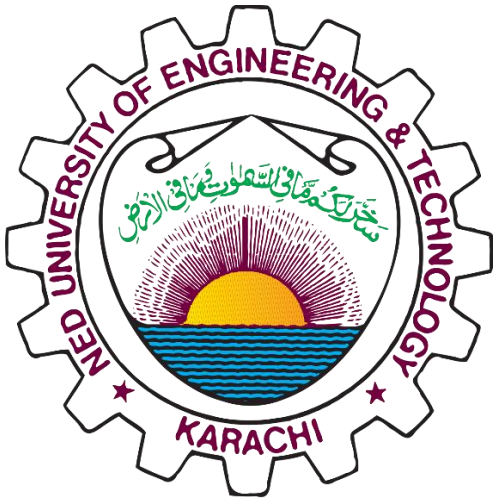
NED UNIVERSITY OF ENGINEERING & TECHNOLOGY



Project Title:
AI Enhanced Vulnerability Scanner

Group Members	**Syeda Wassama Ali(CR-22015)** **Uzma Haneef (CR-22020)** **Sarah Zafar (CR-22025)** **Syeda Alishba Liaquat (CR-22019)**
Course Instructor	**Muhammad Abdullah Siddiqui**
Course Title	**Artifical Intelligence & Expert System**
Course Code	**CT-361**

AI Enhanced Vulnerability Scanner Report:

1. Project Introduction

This project is a custom-built web vulnerability scanner that leverages the OWASP ZAP tool for scanning websites and identifies common security vulnerabilities. The scan results are cross-matched with a manually prepared Excel-based vulnerability database to enrich the output with structured information such as remediation steps, CVEs, CWEs, and more.
The tool provides a user-friendly GUI and generates well-structured reports in both PDF and HTML formats.

In addition, an **AI module** is integrated into the system to automatically simplify technical vulnerability descriptions and enrich the report with accessible explanations. This enhances the usability of the report for both technical and non-technical stakeholders.

2. Methodology / Tools Used

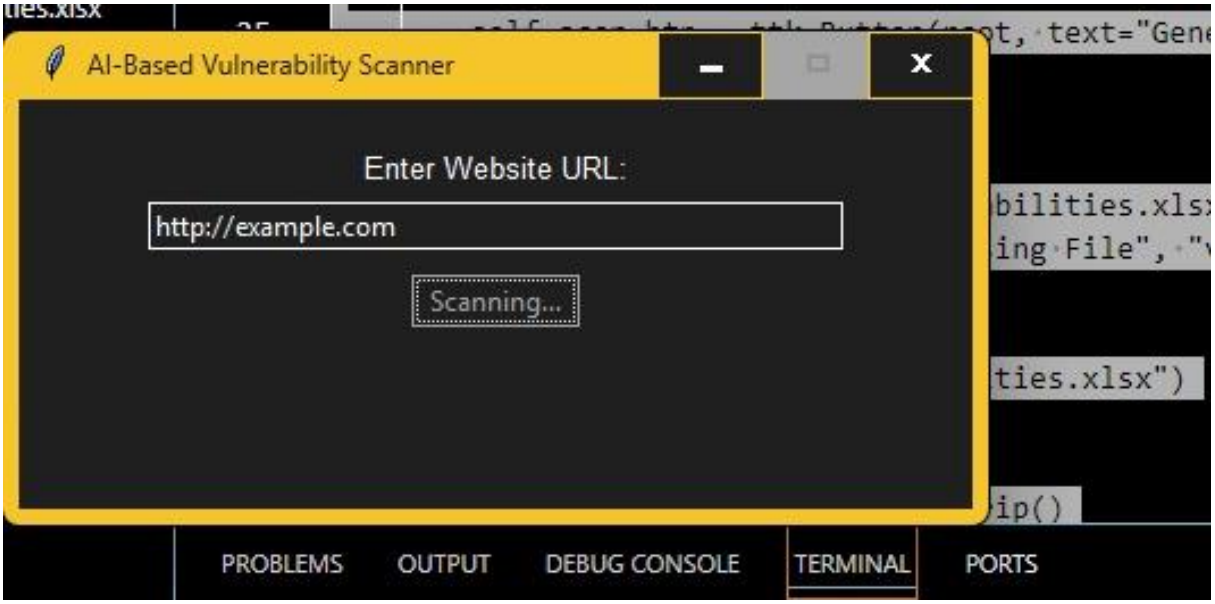
- **OWASP ZAP** – Web vulnerability scanner
- **Excel** – Manually created vulnerability mapping database
- **Pandas** – For reading and processing Excel files
- **HTML & CSS** – For GUI interface
- **PDF Generator Library** – For report export
- **Custom AI Module** – For natural language simplification and data enrichment (rule-based NLP techniques)

3. GUI Interface Description

The tool contains a clean and simple GUI that guides the user through the scanning and reporting process. The flow is as follows:

1. Target URL Entry – User enters the web address to scan
2. Scan Option Selection – ZAP scan is triggered
3. Scan Initiation – The system begins scanning and may take a few moments
4. Matching Phase – ZAP alerts are matched against the local Excel vulnerability database
5. Report Generation – The user can choose between PDF or HTML output format
6. Download/Save – Final report is downloadable or viewable

The AI module works silently during the Matching Phase and Report Generation to enhance explanations and data presentation.



4. Implementation

a. Scanning the Website

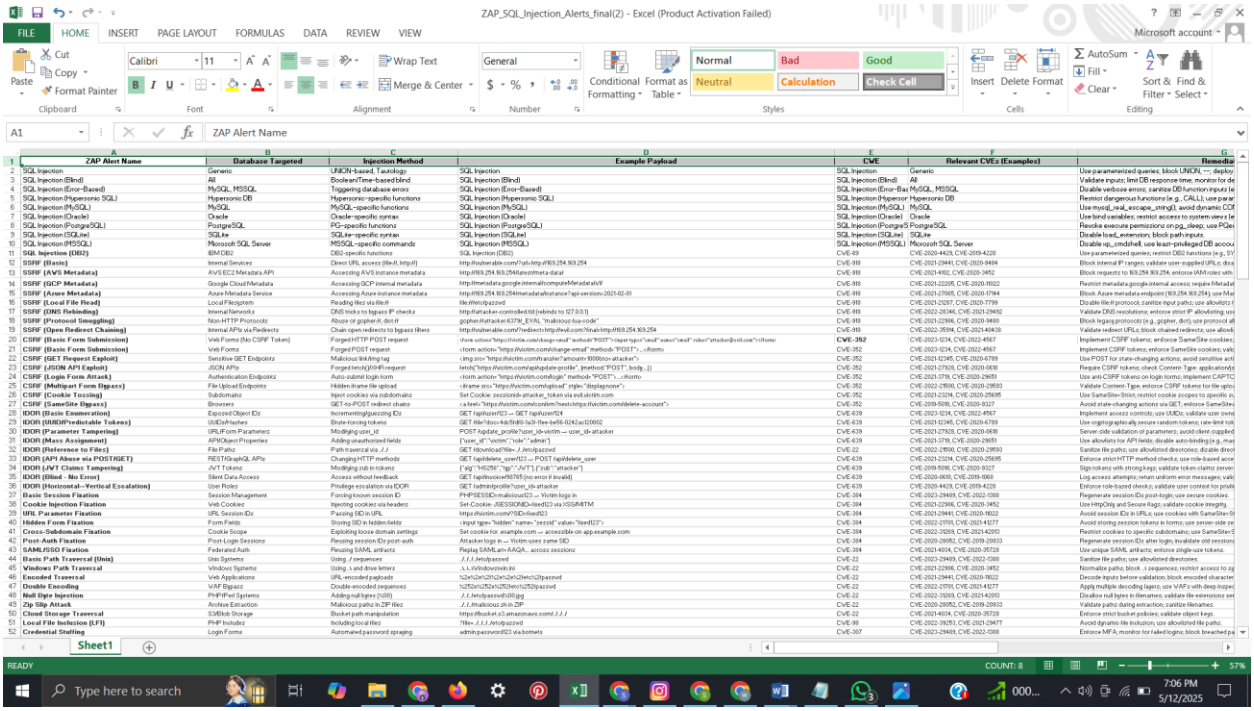
- The backend triggers OWASP ZAP to run an automated scan on the given target URL.
- The scan result contains multiple vulnerability alerts (e.g., XSS, SQL Injection, etc.).

b. Database Structure

The scan results are mapped with a manually built Excel vulnerability database containing the following columns:

Column Name	Purpose
ZAP Alert Name	The name of the vulnerability as reported by ZAP
Database Targeted	The component/area affected (e.g., Database, Input Field)
Injection Method	The type of injection (e.g., SQL, XSS, Command Injection)
Example Payload	A sample malicious input that can exploit the vulnerability
CWE	Common Weakness Enumeration ID

Relevant CVEs (Examples)	Known public CVEs related to the vulnerability
Remediation	Steps and strategies to fix the issue
Description	A brief explanation of the vulnerability and its impact



c. Matching Logic

- The ZAP alert name is used to search the Excel sheet for matching vulnerabilities.
- Once matched, corresponding CWE, CVE, remediation, and description are extracted.
- The **AI module performs text simplification** on descriptions to generate easier-to-understand explanations for users.

d. Report Generation

- Final matched data is formatted and converted into both PDF and HTML formats.
- Both formats include all matched vulnerabilities and relevant remediation data for easy review.
- The AI-generated simplified explanations are included to aid comprehension.

5. AI Modules and Their Roles

Overview

This project includes AI-driven enhancements primarily based on rule-based NLP and knowledge integration techniques. These modules improve report clarity, data enrichment, and classification.

5. AI Modules and Their Roles

This project integrates multiple AI-driven components that enhance vulnerability classification, explanation, and report clarity. These modules include both traditional machine learning models (SVM, Neural Networks) and modern NLP techniques (LLM-based summarization and simplification).

5.1 Vulnerability Classification using SVM and Neural Networks

- **Technique Used:** Supervised Machine Learning
- **Models:** Support Vector Machine (SVM) and Feedforward Neural Networks
- **Function:** These models are trained on labeled vulnerability data to classify the severity of newly discovered vulnerabilities (e.g., Low, Medium, High) based on features such as alert type, injection method, and targeted component.
- **Benefit:** This enhances accuracy in vulnerability prioritization, replacing fixed threshold classification with adaptive ML-based prediction.

5.2 Natural Language Processing using LLM (Large Language Model)

- **Technique Used:** Pre-trained LLM with fine-tuned prompt-based processing
- **Function:** Used to simplify technical vulnerability descriptions and generate human-readable summaries for each identified issue.
- **Example Usage:**
    - Translating technical CVE/CWE descriptions into plain English
    - Summarizing long descriptions or remediation steps
    - Recommending concise action points
- **Benefit:** Makes reports accessible to both technical and non-technical users by removing jargon and increasing clarity.

5.3 Data Enrichment & Matching Logic

- **Technique Used:** Rule-based AI and Knowledge Graph Matching
- **Function:** This module maps ZAP alerts to entries in the Excel vulnerability database using fuzzy matching, enriching the data with CVEs, CWEs, payloads, and remediation.
- **Benefit:** Reduces manual effort and improves the completeness of each vulnerability's metadata.

5.4 AI Module Table:

Module Name	AI Technique Used	Description
Severity Classification	SVM & Neural Networks	Classifies risk level of vulnerabilities based on features.
NLP Simplification & Summary	Large Language Model (LLM)	Converts technical text into simplified human-readable explanations.
Data Enrichment & Mapping	Rule-Based Matching	Matches ZAP alerts with a database of known issues to enhance scan output.

6. Report Generation Features

HTML Report Generation

The tool provides the capability to generate the vulnerability assessment report in HTML format. The HTML report includes all detected issues, matched vulnerabilities from the backend database, and detailed information such as CVEs, CWEs, payloads, and recommended remediations. This format is useful for interactive viewing within browsers and can be easily shared or reviewed by security teams.

The HTML output is styled for readability, with collapsible sections and color-coded severity levels (e.g., Medium, Low, Informational). This helps in prioritizing and filtering vulnerabilities based on risk levels. Additionally, the HTML format supports embedded links and tables for rich content presentation.

PDF Report Generation

The tool also supports exporting the full vulnerability assessment in PDF format, which is ideal for documentation and formal submission. The PDF report includes:

- Executive summary with metrics (total, medium, low-risk findings)
- Complete vulnerability details with CWE, CVE, payloads, and database mapping
- Screenshot or evidence fields if captured
- Recommendations for each identified vulnerability

The PDF is formatted with headers, proper sectioning, and visual highlights for severity levels, ensuring it can be printed or sent as an official audit report.

7. Conclusion and Recommendations

This tool helps automate web vulnerability identification and connects it to actionable data for security teams. The integrated AI modules enhance report clarity and prioritization by simplifying complex descriptions and enriching scan data with contextual knowledge.

Future improvements include:

- Adding automatic updates for the CVE/CWE database to keep knowledge current
- Integrating email alerts for immediate notification of critical vulnerabilities
- Supporting authenticated scans for deeper analysis
- Deploying as a web-based hosted tool for wider usage and ease of access
- Exploring machine learning techniques for vulnerability prediction and dynamic risk scoring

8. References

- OWASP ZAP Documentation
- CWE Database: https://cwe.mitre.org/
- CVE Details: https://cvedetails.com/