Security Project Submission

Vulnerability Scanner Dashboard

Team Members: Tushar Gulyani

Samarth Khandelwal

Prabhjot Singh

Yusuf Ejaz

Urvi Patil

Srujan

Table of Contents

INTRODUCTION

The Vulnerability Scanner Dashboard is a web-based application designed to provide a comprehensive and user-friendly interface for scanning websites for vulnerabilities. It leverages the OWASP Zed Attack Proxy (ZAP) to perform real-time scanning and presents the results in an intuitive and visually appealing dashboard. This report details the features, functionalities, and technical aspects of the application.

OBJECTIVE

The primary objective of the Vulnerability Scanner Dashboard is to enhance website security by identifying potential vulnerabilities in real-time. It aims to:

- Provide a seamless interface for users to initiate and monitor vulnerability scans.
- Present scan results in a clear and visually appealing manner.
- Enable users to take timely actions based on the identified vulnerabilities.

Features

1. Real-Time Scanning

- Utilizes OWASP ZAP to perform comprehensive security scans on websites.
- Capable of detecting various types of vulnerabilities, including SQL injection, Cross-Site Scripting (XSS), and insecure server configurations.

2. Intuitive Dashboard

- Presents scan results in an easy-to-understand format.
- Features graphs, and charts for detailed analysis.
- Provides real-time updates and notifications.

3. List of top vulnerable pages

4. Severity Distribution

5. Recent Scan History

Technical Specifications

- Frontend: HTML, JavaScript, Tailwind CSS
- Backend: Python with Flask
- Charting: Chart.js
- Scanning Tool: OWASP ZAP (Zed Attack Proxy)

Prerequisites

1. Python 3.7+
2. OWASP ZAP
3. pip (Python package manager)

Installation

1. Clone the repository: git clone https://github.com/Tushar4059x/cybersecurity-dashboard
2. Install the required Python packages: pip install flask python-owasp-zap-v2.4
3. Install and set up OWASP ZAP: Download ZAP from https://www.zaproxy.org/download/

   Follow the installation instructions for your operating system

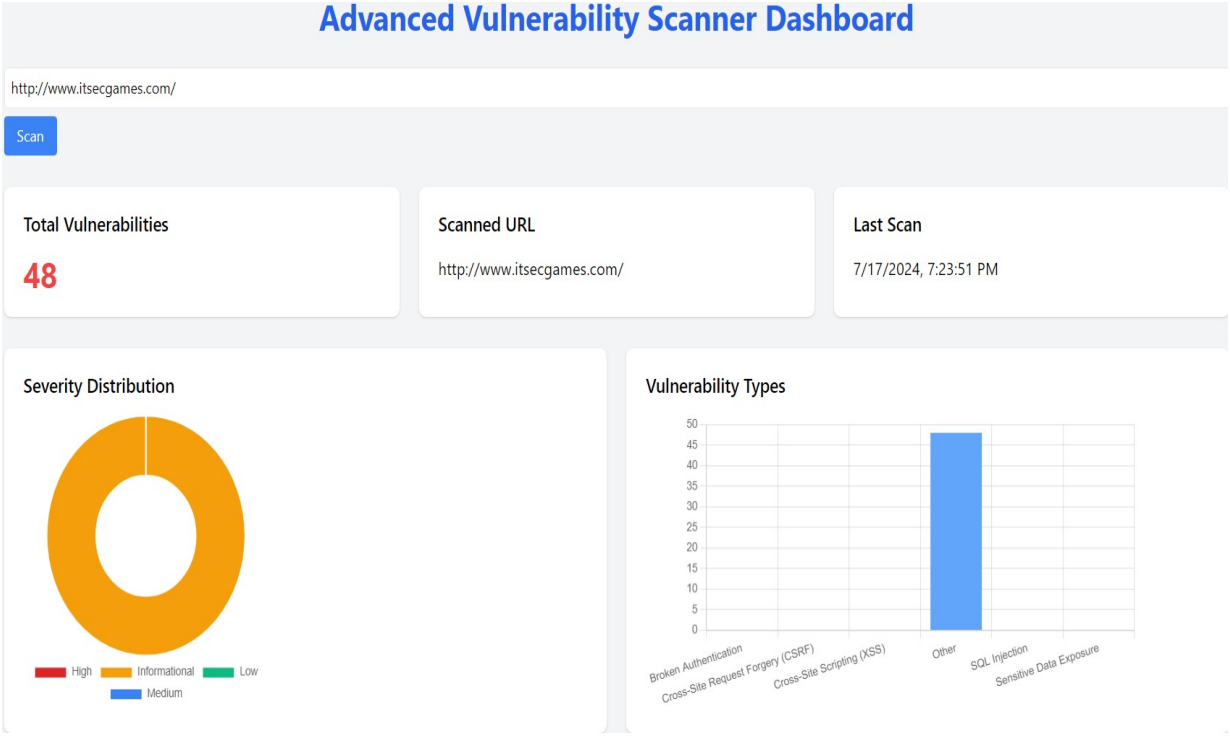
Configuration

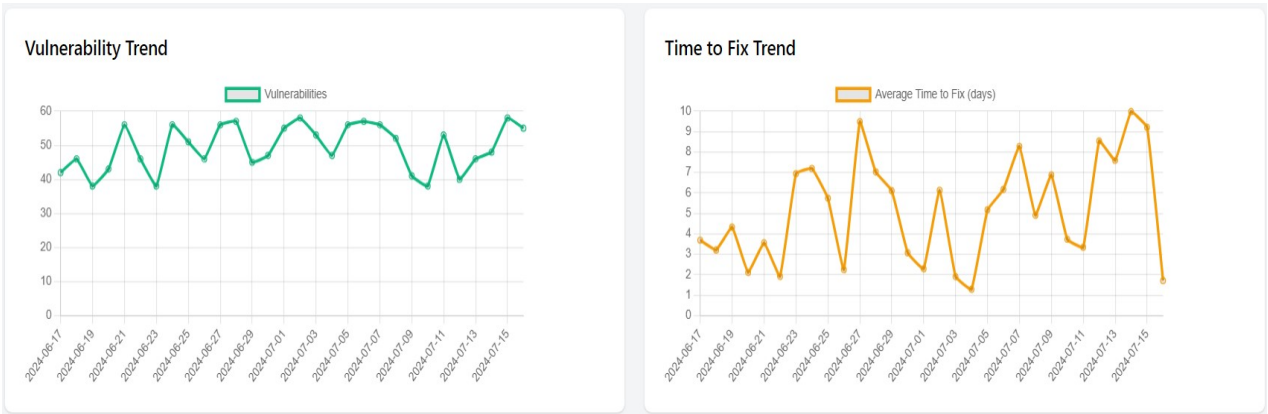
1. Open app.py and replace 'YOUR_ZAP_API_KEY' with your actual ZAP API key.
2. Ensure ZAP is running and accessible at http://localhost:8080 (default setting).

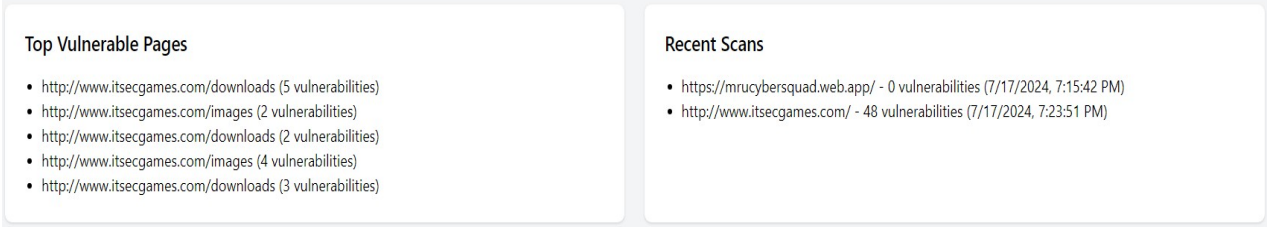
Usage

1. Start the Flask application: python app.py
2. Open a web browser and navigate to http://localhost:5000
3. Enter the URL of the website you want to scan in the input field.
4. Click the "Scan" button to initiate the vulnerability scan.
5. View the results in the interactive dashboard.

Dashboard snapshots







Conclusion

The Vulnerability Scanner Dashboard successfully provides a robust solution for scanning websites for vulnerabilities. With its real-time scanning capabilities and user-friendly interface, it empowers users to proactively identify and address potential security issues, thereby enhancing the overall security posture of their web applications.