

PENT-PEEK

An integrated Dynamic Application Security Testing Tool

PREPARED BY:

YARA ALMANEE	JUMANAH ALHARBI
WASAN THABIT	ATHEER ALEYADHY
NORAH ALMACKENZY	

Table of Contents

3 ABSTRACT

4 INTRODUCTION

5 PROBLEM STATEMENT

6 SOLUTION

7 RELATED WORK

8 SYSTEM DESIGN

9 METHODOLOGY

10 FUTURE WORK

11 REFERENCES



Abstract

Our project, PentPeek, aims to create an advanced tool that simplifies web application penetration testing while ensuring high accuracy. This tool is built on the Dynamic Application Security Testing (DAST) approach, which is a method that identifies security vulnerabilities in web applications by simulating real-world attacks.

With PentPeek, users can select specific vulnerabilities they want to test for, and the tool will automatically run tests using a selection of standard penetration testing tools. The results are then compiled into a clear, comprehensive report. This user-friendly tool combines in-depth analysis with an intuitive interface, making it easier to secure web applications.

Introduction

Web applications have become an integral part of modern business operations. However, as these applications evolve in complexity, the challenge of securing them against sophisticated cyber threats intensifies. While a variety of security tools are available, many are limited in scope, focusing on specific vulnerabilities and leaving other risks unaddressed. This approach can make comprehensive security testing both time-consuming and challenging to manage.

Recognizing these challenges, we developed **PentPeek**. PentPeek utilizes Dynamic Application Security Testing (DAST) to simulate real-world attack scenarios, identify a wide range of vulnerabilities, and recommend effective mitigation measures. By integrating several security testing tools into a unified, user-friendly platform, PentPeek streamlines the security assessment process, improving both efficiency and accessibility, even for users without extensive cybersecurity expertise.

In this report, we will explore how PentPeek works, discuss its design and features, and examine how it can effectively address the security needs of today's web applications.

Problem Statement

As web applications become increasingly complex, the need for robust security measures has grown significantly. A report by Positive Technologies found that 93% of web applications tested had vulnerabilities that could be exploited, and 70% of these had vulnerabilities classified as “high severity.” This underscores a critical issue: the current state of web application security is not sufficient to keep pace with the ever-evolving threat landscape.

Existing tools for web security testing, while important, often fall short of providing comprehensive protection. Many focus on identifying specific types of vulnerabilities, such as SQL Injection, or Cross-Site Scripting (XSS). While detecting these vulnerabilities is crucial, the narrow focus of these tools means they may miss other significant security flaws. Furthermore, relying on a single tool or using multiple specialized tools can leave gaps in protection, potentially leading to successful cyberattacks.

Managing multiple security testing tools can also be challenging. Security professionals often have to juggle different platforms, configure various tools, and manually compile results to get a complete picture of an application’s security. This fragmented approach not only increases the workload but also raises the risk of human error, which can lead to critical vulnerabilities being overlooked, potentially resulting in data breaches or financial losses.

1.5 million

Number of vulnerabilities discovered in the past five years



23%

Percentage of vulnerabilities due to human error

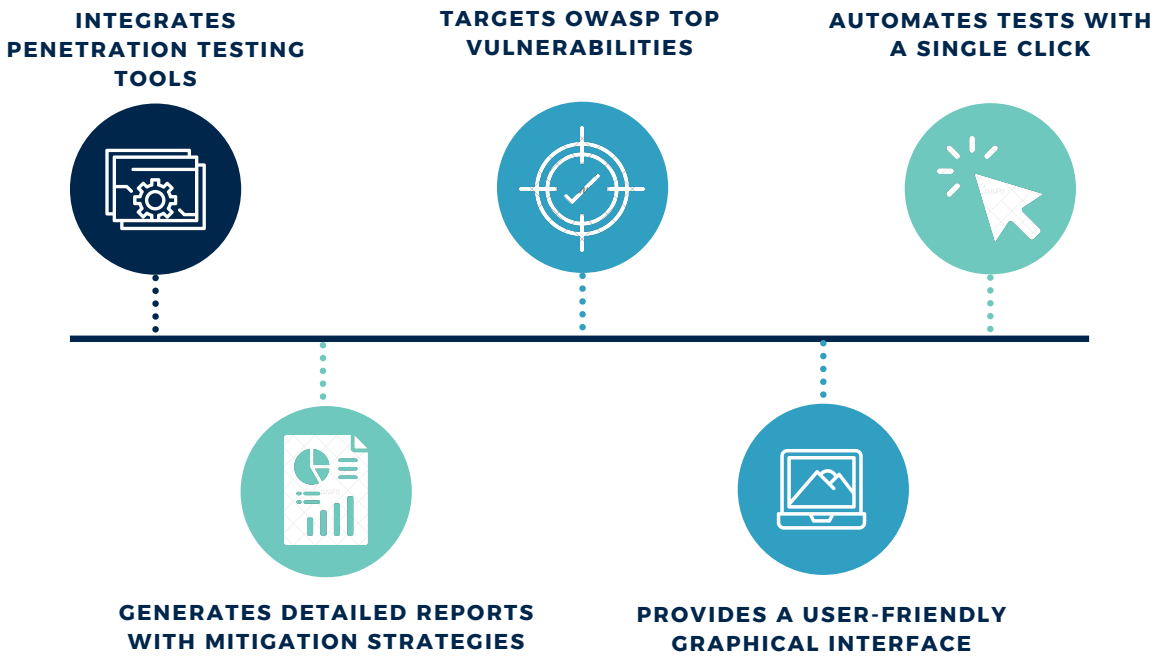
30%

Percentage of vulnerabilities that could have been prevented with early detection

Solution

Given the rapid advancement of technology and the increasing sophistication of cyber threats, there is an urgent need for an automated, user-friendly solution that streamlines the web security testing process. The proposed solution is to develop an integrated dashboard that combines multiple specialized tools for detecting a wide range of vulnerabilities into a single, easy-to-use interface.

This platform will specifically target the top OWASP vulnerabilities, such as SQL injection, Cross-Site Scripting (XSS), Directory Listing, and Directory Traversal. It will automate the testing process using these tools and compile the results into a comprehensive report. This report will not only list each identified vulnerability but also provide detailed mitigation strategies to help users address them effectively. By doing so, the solution will enhance the efficiency of security teams and ensure that web applications are more resilient against the evolving threats they face in today's digital environment.



Related Work

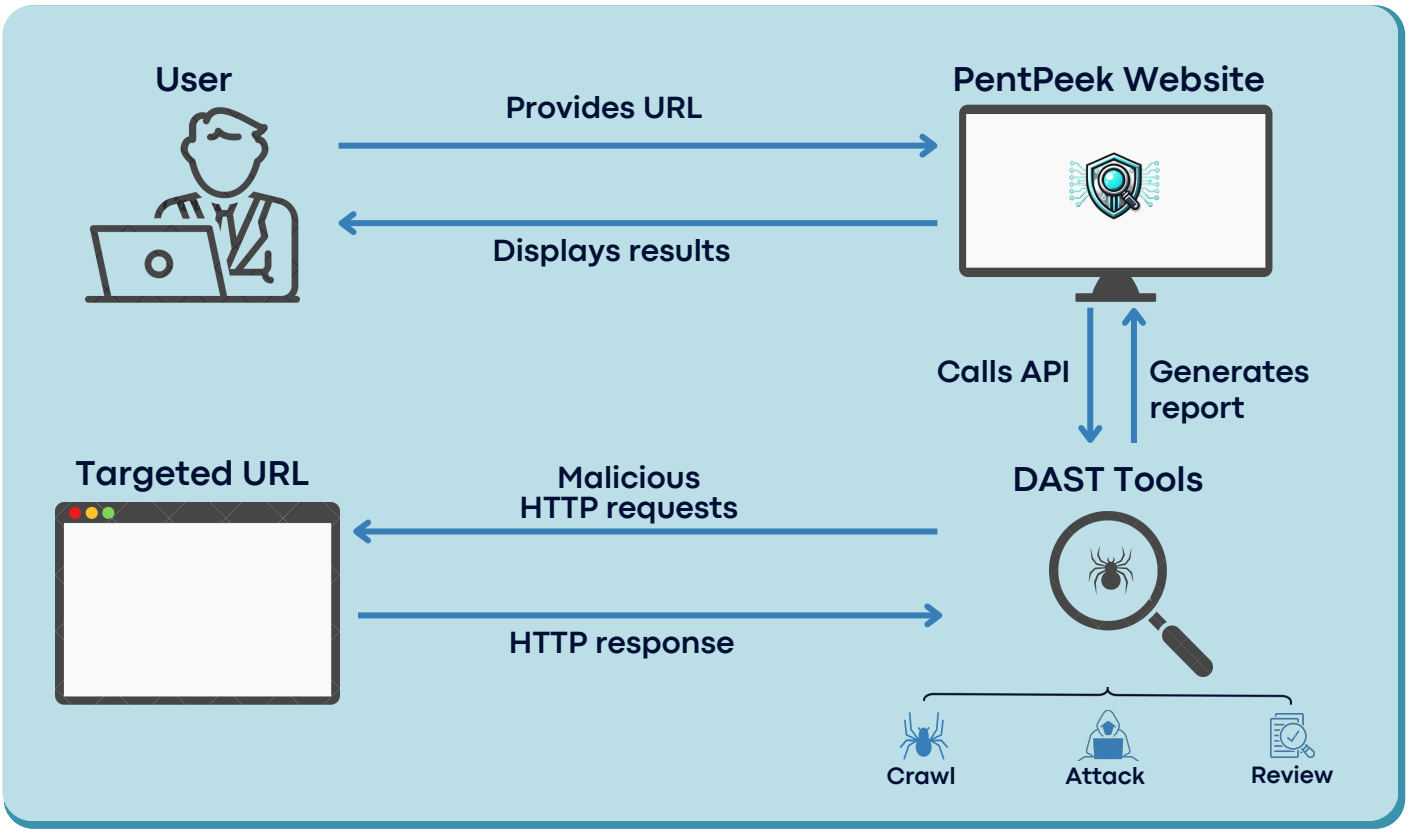
1. Dastco

A command-line tool that is designed to manage and interact with a wide range of Dynamic Application Security Testing (DAST) tools, including industry-leading options like **WebInspect** and **Burp Suite** Enterprise. It simplifies the complex process of security testing by providing seamless scan management and deployment, all facilitated through Docker. This integration ensures that users can easily set up, execute, and monitor security scans in isolated, consistent environments, reducing the potential for configuration errors. Dastco's streamlined interface and automation capabilities make it an invaluable resource for security professionals, enabling efficient and effective testing of web applications with minimal manual intervention.

2. Analyzing the Accuracy and Time Costs of Web Application Security Scanners

A study that evaluates the performance of various web application security scanners by meticulously comparing their ability to accurately detect vulnerabilities across different types of web applications. The analysis delves into each scanner's effectiveness, examining not only the precision with which they identify potential security threats but also the time efficiency of their scanning processes. By providing a detailed comparison of how quickly and accurately these tools can identify critical vulnerabilities, the study offers valuable insights for organizations looking to optimize their security testing strategies. The goal is to help businesses make informed decisions about which security scanners best balance speed and accuracy, ultimately enhancing the protection of their web applications against potential threats.

System Design



The system design diagram for the PentPeek web application shows how users interact with the application to perform security tests on targeted URLs. Users provide a URL, which is submitted through the PentPeek website. The website then calls an API to initiate Dynamic Application Security Testing (DAST) tools, which perform tasks such as crawling, attacking, and reviewing the web application.

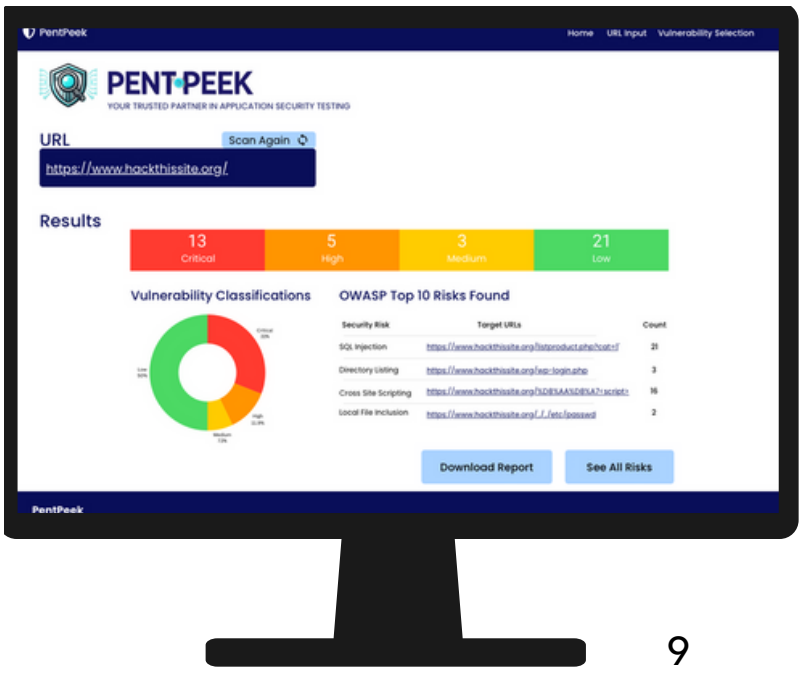
These tools send HTTP requests that simulate the behavior of an attacker to identify potential vulnerabilities. After completing the tests, the DAST tools generate a report summarizing any identified security issues. This report is then sent back to the PentPeek website, where the results are displayed to the user, providing insights into the security status of the web application.

Methodology

The methodology began with defining project requirements and objectives, ensuring that the tool would meet the needs of its users. The design phase utilized Figma to create intuitive and user-friendly interface prototypes, providing a clear layout for inputting URLs, selecting security tests, and displaying results.

Python was used as the primary scripting language to develop the core functionality of the security tests, leveraging relevant libraries to detect vulnerabilities effectively. These vulnerabilities were chosen based on the OWASP top 10 vulnerabilities from the year 2021. Each of the individual scripts were integrated into a web interface created using Streamlit, enabling users to interact with the security testing features in a straightforward and accessible manner. Visual Studio Code (VSCode) was employed as the IDE for writing and debugging these scripts. The project leveraged GitHub for version control and collaboration, which allowed for efficient code management, issue tracking, and team coordination.

After thorough testing and validation, the PentPeek application was successfully deployed. By following a structured approach that integrated design, scripting, and established security practices, the project delivered a reliable and user-friendly solution for web application security testing.



Tools Used

- Python
- Visual Studio Code
- GitHub
- Streamlit
- Figma
- OWASP



Future Work

INTEGRATING MACHINE LEARNING (ML)

By incorporating Machine Learning (ML) algorithms, we aim to significantly reduce false positives and enhance the overall accuracy of vulnerability detection. This approach involves training ML models on extensive datasets that encompass both web vulnerabilities and benign patterns, enabling the system to more effectively distinguish between genuine threats and harmless anomalies.

UTILIZING NATURAL LANGUAGE PROCESSING (NLP)

By leveraging Natural Language Processing (NLP) capabilities, we aim to enable users to interact with the tool through intuitive natural language queries. This feature will enhance accessibility, making the tool easier to use, while also generating clear, contextualized reports that translate complex technical findings into actionable insights.

IMPLEMENTING THE TOOL AS BROWSER EXTENSION

By deploying the tool as a browser extension, we aim to enable real-time detection and analysis of web vulnerabilities directly within users' browsers. The extension will seamlessly integrate with popular browsers, providing instant feedback on potential security issues as users interact with websites. By leveraging real-time scanning and alerting mechanisms, the extension will proactively identify and mitigate security risks on-the-fly, significantly enhancing web safety and user protection.

References

- Guimaraes, B. D. A., & Stampar, M. (2024). *sqlmap: Automatic SQL injection and database takeover tool.* Retrieved from https://sqlmap.org

- Zed Attack Proxy. (2024). *ZAP: The world's most widely used web app scanner.* Retrieved from https://www.zaproxy.org/

- v0re. (2024). *dirb: Web fuzzer* [GitHub repository]. Retrieved from https://github.com/v0re/dirb

- s0md3v. (2024). *XSStrike: Most advanced XSS scanner* [GitHub repository]. Retrieved from https://github.com/s0md3v/XSStrike

- jcesarstef. (2024). *dotdotslash: Search for directory traversal vulnerabilities* [GitHub repository]. Retrieved from https://github.com/jcesarstef/dotdotslash

- mkorejo. (2024). *dastco: Versatile CLI for programmatic interaction with Dynamic Application Security Testing (DAST) tools* [GitHub repository]. Retrieved from https://github.com/mkorejo/dastco

- Suto, L. (2010). *Analyzing the accuracy and time costs of web application security scanners.* [White paper], San Francisco, CA.



PENT·PEEK

View More