Project Proposal

A Tool for Automating Pre-Penetration Testing Reconnaissance

Team Members:
Muhammad Asim (23PWBCS0984)
Muhammad Sattar (23PWBCS0963)
Talal Rashid (23PWBCS1002)

Supervisor: Ms. Kanwal Aneeq

Department: Computer Science & IT
University of Engineering & Technology, Peshawar, Pakistan

Abstract

This project is all about building a web-based tool to automate the very first part of a security test, which we call reconnaissance. Right now, gathering all the information needed for these tests is a completely manual job, and it eats up a ton of time and resources [1]. Our tool will fix this by automatically collecting data on web-based systems. We'll do this using a smart mix of web scraping and specialized online APIs, which will make the process way more efficient. We believe this tool will be a huge help for security analysts and developers, giving them faster, simpler reports so they can focus on fixing security issues instead of just spending time finding them.

1. Introduction

In today's digital world, it feels like cyber-attacks are always in the news, so keeping web systems secure is more important than ever. That's why we have penetration testing—it's like playing the role of an ethical hacker to find and fix security weaknesses before anyone with bad intentions can use them [2]. The very first step of this process, called reconnaissance, is all about gathering as much information as possible on the target system to figure out where the weak spots might be.

The problem is that this information gathering is almost always done by hand. It's a slow, repetitive process that uses up a lot of time and resources, which means security tests are often only done at big project milestones [1]. Our goal is to create a web-based solution that makes things faster and more efficient. A web-based tool is great because it can be

accessed from anywhere and updates are handled centrally, which will free up security testers to focus on the truly difficult and interesting parts of their job.

2. Problem Statement

Right now, there isn't a good tool that automates the classification of web-based systems based on their security needs. The manual process is inefficient, takes a lot of time, and requires a lot of manual work to gather information and classify assets.

3. Objectives

The main goals for our project are:

- To build a web-based tool that classifies publicly available web systems.
- To use automation to collect all the necessary information for this classification.
- To produce clear and easy-to-understand reports on the classified systems.
- To make reports for both individual systems and a whole group of them.

4. Scope of the Project

In Scope:

- Developing a web-based application for security asset classification.
- Automating information gathering using web scraping and APIs from public reconnaissance tools.
- Assessing security requirements based on the CIA Triad and Exposure Factor.
- Generating reports in PDF and CSV formats.

Out of Scope:

- The project will not include any hardware deployment or integration with external devices beyond a standard computer setup.
- The tool will be a web application, not a native mobile one.

5. Proposed Solution / Methodology

Our proposed solution is a web-based application that will automate the security classification process. The tool's method is built on a smart, two-part approach to data collection and analysis.

The main way we'll get our information is through something called Open-Source Intelligence (OSINT), which just means gathering data from public sources [3]. Our tool will do this without directly interacting with the target system in a way that would get logged. We'll combine general web scraping with special security-focused APIs, which gives us a more complete picture of the target's security.

- **Data Collection:** We will use a library like Jsoup to pull data directly from a webpage's HTML code. We'll also use information from search engines like Shodan, which constantly scan the internet for exposed devices and services, giving us a specific kind of security data that we can't get from just scraping.
- **Classification:** After we get all the info, the tool will classify the system. It will use the CIA Triad (Confidentiality, Integrity, and Availability) to give a qualitative rating of the system's security. To make it even more useful, it will also calculate an "Exposure Factor," which gives a simple, measurable score of the potential risk.

Tools & Technologies:

- **Programming Language:** We're going to use Python with a web framework like Flask or Django for the backend, since they're great for building web tools. We'll also use HTML, CSS, and JavaScript for the front end.
- **Libraries:** We'll use libraries for things like web scraping, handling data, and creating reports.
- **Operating System:** Since it's a web tool, it will be OS-independent and work on any modern browser.

6. Stakeholders

- **Primary Stakeholders:** Security Testers, Security Analysts, and Application Developers.
- **Secondary Stakeholders:** Organizations and university administration.

7. Expected Outcomes

- A working prototype of our automated security classification tool.
- A significant reduction in the time and effort needed for the first phase of a security test.
- Clear and detailed reports that simplify security assessments.
- A demonstration of how automation can make security operations much more efficient.

8. References

Citation	Reference
[1]	N. Samant, "AUTOMATED PENETRATION TESTING," Master's Project, San Jose State University, 2011. DOI: 10.31979/etd.fxpj-pt6k. https://www.ijnrd.org/papers/IJNRD2405399.pdf

[2]	V. Singh and K. Jaiswal, "An AI-Based Approach for Automating Penetration Testing," *International Journal of Research Publication and Reviews*, vol. 4, no. 7, pp. 2231-2234, 2023. [Online]. Available: https://www.ijrpr.com/uploads/V5ISSUE6/IJRPR30027.pdf
[3]	V. Kumar and S. Singh, "Osint Automation Application," *Journal of Physics: Conference Series*, vol. 2315, no. 1, 2022. [Online]. Available: https://www.researchgate.net/publication/370484915_Osint_Automation_Application