CYB333: Security Automation

Project PyPortScanner

Professor Brian Scavotto

National University

Taylor S. Barnes

04/26/2023

**Project Name**

PyPortScanner

**Project Plan**

1. Introduction
2. Project Scope
3. Project Goals
4. Problem Statement
5. Expected Outcomes
6. Project Timeline

**Introduction**

PyPortScanner is a Python-based tool designed to automate port scanning tasks. The tool aims to assist security professionals, IT administrators, and developers in discovering open ports, identifying services, and assessing the security of their networks and systems..

**Project Scope**

The scope of this project includes the development of a Python tool that automates port scanning functions. The tool will have the following features:

* Support for scanning single or multiple IP addresses
* Support for scanning specific or ranges of ports
* Display results in a user-friendly format
* Export results to various formats (e.g., CSV, JSON, XML)
* Provide options for customizing the scan intensity

**Project Goals**

The goals of this project are to:

* Simplify and streamline the port scanning process.
* Improve network and system security by identifying open ports and potential vulnerabilities.
* Provide a user-friendly tool that is accessible to both technical and non-technical users.
* Foster collaboration between security professionals, IT administrators, and developers.

**Problem Statement**

Organizations often face challenges in securing their networks and systems due to the complexity of the infrastructure and the multitude of potential vulnerabilities. Manual port scanning can be time-consuming and prone to human error. PyPortScanner aims to automate the port scanning process, allowing users to efficiently identify open ports and potential security risks.

**Expected Outcomes**

By the end of this project, we expect to have a fully functional Python tool that automates port scanning tasks. The tool will:

* Enable users to perform efficient and accurate port scans.
* Improve network and system security by identifying open ports and potential vulnerabilities.
* Provide a user-friendly interface that is accessible to both technical and non-technical users.
* Encourage collaboration between security professionals, IT administrators, and developers.

**Project Timeline**

Estimated timeline for full-scale project implementation (hypothetical):

* Week 1: Research and requirement gathering.
* Week 2: Design of the tool's architecture and user interface.
* Week 3: Development of the core port scanning functionality.
* Week 4: Implementation of additional features (customizing scan intensity, exporting results, etc.)
* Week 5: Testing and bug fixing.
* Week 6: User documentation and final deployment.

This project plan was developed to provide an outline for the development of PyPortScanner, a Python-based port scanning tool. By following this plan, the project can be used to develop a user-friendly and efficient tool that addresses the challenges of manual port scanning and improves network security.

GitHub Repository reference:

<https://github.com/BarnabyJones001/PyPortScanner>

The following is a screenshot of the code, and the execution of that code for the PyPortScanner:

A screenshot of a computer

Description automatically generated with medium confidence