# Comprehensive Port Scanner with Vulnerability Insights

## Table of Contents



---

## 1. Introduction

The **Comprehensive Port Scanner with Vulnerability Insights** is a Python-based tool that scans open ports on a given target (IP address or domain name) and provides associated vulnerabilities for those ports. The project focuses on essential ports commonly used by network services and filters out closed ports for a clean, efficient output. This scanner is built with a graphical user interface (GUI) using `Tkinter` and integrates `nmap` to perform the scanning. Vulnerability information is statically linked to popular services for demonstration purposes.

## 2. Key Features

- **Scan Open Ports**: The tool scans only essential, well-known ports and filters out closed ports for better clarity.
- **Display Vulnerabilities**: For each open port, the tool displays static vulnerability information related to the service running on that port.
- **IP and Domain Support**: The scanner accepts both IP addresses and domain names for ease of use.
- **User-Friendly GUI**: The graphical interface built with Tkinter provides a simple, clean user experience.
- **Real-Time Feedback**: Scanning progress is displayed in real-time, with messages indicating when the scan is in progress and when it is complete.

## 3. Tools and Libraries

- **Python 3.x**: The project is implemented in Python, providing portability and ease of use.
- **nmap**: This is used for performing network scanning, detecting open ports, and identifying services.
- **socket**: This is used to resolve domain names to IP addresses.
- **Tkinter**: The built-in Python library for creating a graphical user interface.

- **threading**: To run the scan in a separate thread, ensuring that the GUI remains responsive during scanning.

### Python Libraries:

- `nmap`: Install using `pip install python-nmap`
- `Tkinter`: Pre-installed with Python on most systems
- `socket`: Part of Python's standard library

# 4. Project Structure

- **main.py**: The primary script containing the GUI and scanning logic.
- **Tkinter GUI Elements**: The interface includes:
    - Title label
    - Entry box for inputting an IP address or domain
    - Button to start the scan
    - Text box to display the results

# 5. How It Works

The project operates in the following steps:

1. **Input**: The user enters a target IP address or domain name in the input field.
2. **Scan Initialization**: The domain (if entered) is converted into an IP address using the `socket.gethostbyname()` method.
3. **Port Scanning**: Using the `nmap` library, the tool scans the target for open ports, specifically focusing on essential ports (SSH, HTTP, HTTPS, FTP, etc.).
4. **Open Port Filtering**: Only open ports are displayed in the results, while closed ports are omitted for clarity.
5. **Vulnerability Information**: For each open port, the tool displays relevant vulnerability information linked to the service running on that port.
6. **Results Display**: The scan results, including the open ports and vulnerabilities, are displayed in the GUI's text box for the user to view.

### Supported Essential Ports:

- **22 (SSH)**: Weak password, outdated version vulnerabilities.
- **80 (HTTP)**: Insecure HTTP, open directory vulnerabilities.
- **443 (HTTPS)**: SSL misconfiguration vulnerabilities.
- **21 (FTP)**: Anonymous access, cleartext credentials vulnerabilities.
- **53 (DNS)**: DNS cache poisoning vulnerabilities.
- **25 (SMTP)**: Open relay vulnerabilities.
- **110 (POP3)**: Cleartext credentials vulnerabilities.
- **143 (IMAP)**: Cleartext credentials vulnerabilities.
- **3306 (MySQL)**: Default credentials, SQL injection vulnerabilities.
- **8080 (HTTP-Proxy)**: Misconfigured proxy vulnerabilities.

# 6. Installation and Setup

### Prerequisites:

- Python 3.x installed
- `nmap` installed on the system (can be installed using package managers like `apt` on Linux or by downloading from the official Nmap website).
- Install the required Python libraries.

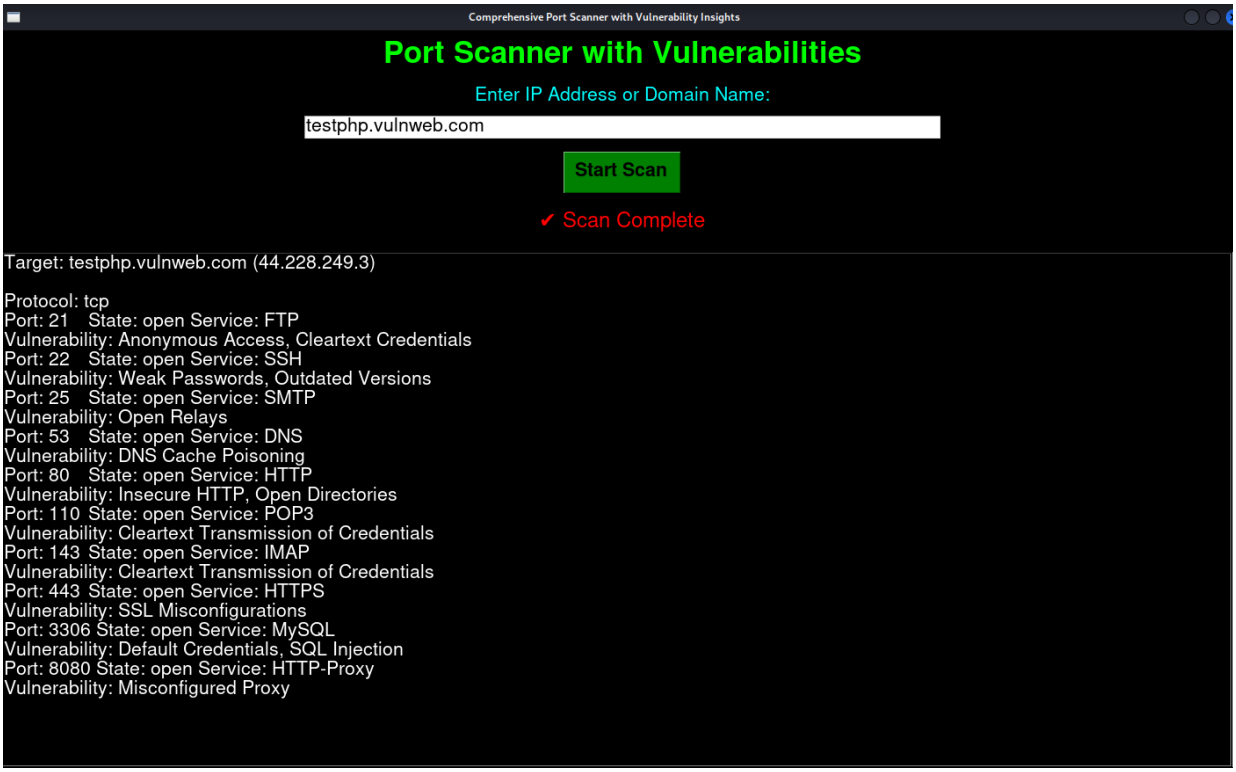
# 7. Usage

- **Enter IP/Domain**: Start by entering an IP address or a domain name (e.g., `example.com`).
- **Click "Start Scan"**: After entering the target, click the "Start Scan" button. The scan may take some time depending on the target and network conditions.
- **View Results**: The results will show open ports, the associated services, and potential vulnerabilities in the text box below.

### Example Scan:

- **Input**: `testphp.vulnweb.com`
- **Output**:

8.




# Future Improvements

- **Dynamic Vulnerability Information**: Integrating real-time vulnerability scanning APIs like the CVE database or services like Shodan to provide up-to-date vulnerability details for each service.
- **Report Export**: Add a feature to export the scan results and vulnerability information to a file (e.g., `.txt` or `.csv`) for further analysis.

- **Progress Bar**: Implement a progress bar that visually indicates the progress of the scan.
- **Advanced Port Selection**: Allow the user to specify custom ports for scanning instead of just essential ones.

# 9. Conclusion

The **Comprehensive Port Scanner with Vulnerability Insights** is a practical tool for beginners and cybersecurity enthusiasts who want to quickly identify open ports and associated vulnerabilities on a target machine. The project's clean, user-friendly GUI and streamlined output make it an accessible solution for basic network reconnaissance tasks.

This tool can serve as a foundation for future enhancements, making it a valuable resource for learning about port scanning and vulnerability analysis in real-world scenarios.