🐍 Vulnerability Port Scanner (VPS)

MADE BY ARON :)

A fast, multi-threaded TCP Port Scanner written in Python 3. Designed for ethical security professionals and penetration testers, this tool quickly identifies open ports, performs banner grabbing to determine running services, and flags known vulnerable software versions.

This project uses native Python libraries, making it highly portable across Linux (Kali), macOS, and Windows environments.

✨ Features

- ⚡ **High-Speed Scanning:** Utilizes multi-threading (concurrent.futures) for rapid scanning of large port ranges.
- 📡 **Banner Grabbing:** Attempts to retrieve service banners (including specific HTTP Server headers) to identify software and version.
- 🚨 **Basic Vulnerability Check:** Compares captured banners against a small internal database of common critical vulnerabilities (e.g., outdated Apache versions, known backdoored FTP servers).
- 💻 **Professional CLI:** Uses argparse for clean, non-interactive command-line usage.
- 🎨 **Cybersecurity Theming:** Outputs results using ANSI color codes (Red for critical, Green for success) for immediate visual analysis.

🛠️ Prerequisites

You only need Python 3 installed on your system.

```
python3 --version
# Should output 3.x.x
```

🚀 Installation and Setup

1. **Clone the Repository** (or download the file directly):
   ```
   git clone [YOUR_REPOSITORY_URL_HERE]
   cd [YOUR_REPOSITORY_NAME]
   ```

2. **Grant Execute Permissions** (Recommended for Linux/Kali users):
   ```
   chmod +x simple_port_scanner.py
   ```

💡 Usage

The scanner requires a target IP or hostname and the ports (-p) to be specified.

Syntax

```
./simple_port_scanner.py <target> -p <ports> [-w <workers>]
```

Argument	Shorthand	Description	Example Value
target	(positional)	IP address or hostname to scan.	127.0.0.1 or mywebsite.com
--ports	-p	**(REQUIRED)** Port range (e.g., 1-1024) or specific ports (e.g., 21,22,80,443).	
--workers	-w	Number of concurrent threads for speed. Defaults to 100.	200

Examples

1. Scan the 1000 most common ports on a local machine:

```
./simple_port_scanner.py 127.0.0.1 -p 1-1024
```

2. Scan a specific list of ports on a router, using maximum speed:

```
./simple_port_scanner.py 192.168.1.1 -p 21,22,80,443,3306 -w 150
```

🛑 Important Disclaimer

This tool is for educational purposes and ethical testing only. You must have explicit, written permission from the owner of the system or network before running any scans. Unauthorized scanning is illegal and unethical. The author (Aron) and contributors are not responsible for any misuse or damage caused by this program.