Malware Detection Tool with Nmap Integration

Introduction

Malware detection is crucial for maintaining system security. This project aims to develop a **Python-based malware detection tool** that scans directories to detect malicious files using **hash signatures**. Additionally, it uses **Nmap** to scan the local network for open ports and running services, enhancing the overall security assessment.

How It Works

1. **Malware Detection**:
    - The tool compares the **SHA-256 hashes** of files in a specified directory against known malware hashes stored in a database file.
    - If a match is found, the file is flagged as potentially malicious.
2. **Network Vulnerability Scan**:
    - The tool uses **Nmap** to scan the local network for:
        - **Open ports**
        - **Running services**
        - **Service versions**
    - This helps identify potentially vulnerable devices connected to the network.

Enhanced Code Explanation

1. **Importing Required Modules**:
    - The tool uses:
        - `hashlib` for computing SHA-256 hashes.
        - `os` for directory traversal.
        - `tqdm` for showing progress bars during scanning.
        - `nmap` for performing network scans.
2. **Loading Malware Hashes**:
    - Hashes are loaded from a file using the `load_malware_hashes()` function.
    - Handles errors such as missing files and empty lines efficiently.
3. **Computing SHA-256 Hash**:
    - The `get_sha256()` function reads files in chunks to efficiently calculate the hash, minimizing memory usage.
    - Enhanced error handling is implemented for permission issues and other file read errors.

4. **Scanning Directory**:

   - The `scan_directory()` function recursively scans all files in the specified directory, computing their SHA-256 hashes and checking them against the malware database.
   - A progress bar shows the scanning status in real-time.

5. **Network Scanning using Nmap**:

   - The `nmap_scan()` function scans the local network to:
     - Detect open ports
     - Identify running services and versions
   - Provides a comprehensive overview of network vulnerabilities.

6. **Alert Mechanism**:

   - If a hash matches any entry in the malware database, the tool alerts the user, indicating potential malware.
   - The network scanner also flags devices with open ports and potentially vulnerable services.

Enhanced Features

- **Nmap Integration**:
  - Scans the local network for open ports, running services, and service versions.
  - Displays detailed information about each device found in the network.
- **Progress Indicator**:
  - Implemented using `tqdm` for real-time feedback on the scanning progress.
- **Error Handling**:
  - Enhanced to manage file read errors, permission issues, and network scanning errors gracefully.
- **Comprehensive Security Check**:
  - Combines **network vulnerability scanning** with **file integrity checks**, providing a more robust security solution.

How to Run the Tool

1. Install the required libraries using:

   ```
   pip install tqdm, python-nmap
   ```

2. Ensure **Nmap** is installed on your system:

   ○ On macOS:

     ```
     brew install nmap
     ```

4. Prepare a file named **malware_hashes.txt** containing known malware hashes, one per line.

5. Update the directory path in the `scan_directory()` function and the **target IP range** in the `nmap_scan()` function.

Conclusion

This tool provides a **comprehensive security solution** by combining **malware detection** using SHA-256 hash signatures with **network vulnerability scanning** using Nmap. It enhances system security by:

- Detecting **malicious files**.
- Identifying **vulnerable devices** on the network.
- Providing **detailed reports** of open ports and running services.