Ethical Security Tools Suite

This project provides a suite of ethical security tools developed in Python. It includes tools for file integrity monitoring, malware detection, network analysis, vulnerability scanning, and password cracking.

Project Structure

```
/EthicalSecurityTools/
│
├── /tools/
│   ├── file_monitor.py
│   ├── malware_detector.py
│   ├── network_analyzer.py
│   ├── vulnerability_scanner.py
│   └── password_cracker.py
│
├── /docs/
│   ├── file_monitor_README.md
│   ├── malware_detector_README.md
│   ├── network_analyzer_README.md
│   ├── vulnerability_scanner_README.md
│   └── password_cracker_README.md
│
├── /tests/
│   ├── test_file_monitor.py
│   ├── test_malware_detector.py
│   ├── test_network_analyzer.py
│   ├── test_vulnerability_scanner.py
│   └── test_password_cracker.py
│
├── main.py
├── requirements.txt
├── setup.py
└── README.md
```

Installation

1. **Clone the repository:** `bash git clone https://github.com/yourusername/EthicalSecurityTools.git cd EthicalSecurityTools`
2. **Install dependencies:** `bash pip install -r requirements.txt`

Note: Some tools like `scapy` (used in Network Analyzer) might require root/administrator privileges for full functionality.

Usage

All tools can be run via the `main.py` script. Use `python main.py <tool_name> --help` for specific tool options.

1. File Integrity Monitor (`filemon`)

Monitors file changes by tracking content and attributes.

Example:

```
python main.py filemon /path/to/monitor -i 60
```

For more details, refer to `docs/file_monitor_README.md`.

2. Malware Detector (`malware`)

Detects malware using YARA rules and predefined signatures.

Example:

```
python main.py malware /path/to/scan --rules /path/to/malware_rules.yar --
output json
```

For more details, refer to `docs/malware_detector_README.md`.

3. Network Analyzer (`network`)

Scans networks for connected devices and sniffs/analyzes network packets.

Examples:

```
python main.py network --scan 192.168.1.1/24
python main.py network --sniff eth0 --count 100
```

For more details, refer to `docs/network_analyzer_README.md`.

4. Vulnerability Scanner (`vuln`)

Identifies common web vulnerabilities like XSS and SQL Injection, and scans for open ports.

Examples:

```
python main.py vuln --target example.com --scan-ports
python main.py vuln --target http://testphp.vulnweb.com/ --check-xss --check-sqli
```

For more details, refer to `docs/vulnerability_scanner_README.md`.

5. Password Cracker (`crack`)

Cracks hashed passwords using brute-force and dictionary attacks.

Examples:

```
python main.py crack --hash 5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8 --type sha256 --bruteforce --charset lower --max-length 4
python main.py crack --hash 0cc175b9c0f1b6a831c399e269772661 --type md5 --dictionary common_passwords.txt
```

For more details, refer to `docs/password_cracker_README.md`.

Running Tests

To run the unit tests for each tool:

```
python -m unittest discover tests
```

License

Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

Contact

For any questions or suggestions, please contact your.email@example.com.