NetSentry

NetSentry is a professional red teaming tool designed for detecting hidden Wi-Fi networks, IP cameras, and CCTV systems. Built with open-source libraries, it offers advanced network scanning, traffic analysis, and visualization capabilities, wrapped in a sleek, dark-mode GUI with a red teamer aesthetic. Developed by SunnyThakur25, NetSentry is ideal for ethical penetration testing and security assessments.
Features

Wi-Fi Network Scanning: Detects hidden SSIDs using Scapy with deauthentication and probe injection (ethical use only).
Device Detection: Leverages advanced Nmap scripts (http-enum, rtsp-url-brute, http-auth) to identify IP cameras and vulnerabilities.
Traffic Analysis: Identifies camera streams (RTSP, MJPEG) with pyshark for real-time monitoring.
Visualization: Generates graphical network trees using Graphviz, displayed in a modern GUI.
Stealth Operations: Supports MAC spoofing, Tor/proxy rotation, and randomized scan timing to evade detection.
Secure Storage: Encrypts scan results in an AES-protected SQLite database.
Pentest Reporting: Produces detailed Markdown reports with findings and recommendations.
GUI: ttkbootstrap-powered interface with dark mode, real-time progress, and a cyberpunk-inspired design.

Project Structure
NetSentry/
├── src/
│ ├── **init**.py
│ ├── wifi_scanner.py
│ ├── device_scanner.py
│ ├── vendor_lookup.py
│ ├── traffic_analyzer.py
│ ├── visualizer.py
│ ├── storage.py
│ ├── anonymizer.py
│ ├── gui.py
│ └── main.py
├── config/
│ ├── config.json
│ └── keys.py
├── data/
│ ├── oui.txt
│ ├── cache.db
│ └── logs/
├── output/
│ ├── netsentry_results.json
│ ├── netsentry_tree.png
│ └── report.md
├── tests/

```
│ ├── test_wifi.py
│ └── test_devices.py
├── requirements.txt
└── README.md
```

Prerequisites

Operating System: Linux (Ubuntu/Kali recommended)
Python: 3.8 or higher
Privileges: Root access for Scapy and pyshark
Dependencies:
Python libraries: scapy, python-nmap, graphviz, pycryptodome, requests, pyshark, ttkbootstrap, pillow
System tools: nmap, graphviz, tshark, tor

OUI Database: Download from IEEE OUI

Installation

Clone the Repository:
git clone https://github.com/SunnyThakur25/NetSentry.git
cd NetSentry

Install Dependencies:
pip install -r requirements.txt

Install System Tools:

Graphviz:sudo apt-get install graphviz

Nmap:sudo apt-get install nmap

Tshark:sudo apt-get install tshark

Tor:sudo apt-get install tor
sudo systemctl start tor

Download OUI Database:
wget https://standards-oui.ieee.org/oui/oui.txt -O data/oui.txt

Generate Encryption Key:Edit config/keys.py and replace ENCRYPTION_KEY with a secure 16-byte key:
python -c "import os; print(os.urandom(16))"

Configure Network Interface:Update config/config.json with your Wi-Fi interface (e.g., wlan0):

```
{
"interface": "wlan0",
"network_range": "192.168.1.0/24",
"output_json": "output/netsentry_results.json",
"output_tree": "output/netsentry_tree.png",
"oui_file": "data/oui.txt",
"tor_proxy": {
"http": "socks5://127.0.0.1:9050",
```

"https": "socks5://127.0.0.1:9050"
}
}

Usage
Command Line Interface (CLI)
Run the tool via the main script:
sudo python src/main.py

Graphical User Interface (GUI)
Launch the GUI for an interactive experience:
sudo python src/gui.py

GUI Features:

Scan Tab: Input interface and network range, start/stop scans, view real-time progress.
Results Tab: Browse networks, devices, and streams in a treeview; view network tree image.
Report Tab: Generate and view detailed pentest reports in Markdown.

Outputs:

output/netsentry_results.json: Scan results in JSON format.
output/netsentry_tree.png: Graphical network tree.
output/report.md: Pentest report with findings and recommendations.
data/cache.db: Encrypted scan results.
data/logs/netsentry.log: Audit logs.

Ethical Considerations

Legal Compliance: Obtain explicit permission before scanning networks or devices to comply with laws (e.g., CFAA, GDPR).
Ethical Use: Deauthentication and active scanning are restricted to authorized environments.
Responsible Disclosure: Report vulnerabilities to system owners promptly.

Testing
Run unit tests to verify functionality:
python -m unittest discover tests

Contributing
Contributions are welcome! Please follow these steps:

Fork the repository: https://github.com/SunnyThakur25/NetSentry
Create a feature branch: git checkout -b feature/your-feature
Commit changes: git commit -m "Add your feature"
Push to the branch: git push origin feature/your-feature
Open a pull request.

Report issues or suggest features via the GitHub Issues page.
License

This project is licensed under the MIT License.
Acknowledgments

Developer: SunnyThakur25
Tools: Scapy, Nmap, Pyshark, Graphviz, TTKBootstrap
Community: Open-source contributors to cybersecurity tools

Contact
For questions or collaboration, reach out via GitHub or open an issue.

NetSentry: Empowering red teams with precision and stealth.