DNS Spoofing Detector

This project detects potential DNS spoofing attacks by monitoring DNS traffic in real time. It highlights mismatched or suspicious responses and provides a web interface to view and filter logs.

Features

- Real-time DNS traffic monitoring
- Spoofed response detection using expected DNS records
- Web interface with filtering by domain and date
- Alert banners and siren for spoofed entries
- Option to mute alerts and export filtered logs
- Works with real interfaces or loopback simulation

Dependencies

Install the required packages using:

```
pip install flask scapy tldextract
```

File Structure

```
.
├── detector.py            # Flask web server
├── simulate.py         # Simulates spoofed DNS packets
├── interface.py        # Lists all available network interfaces
├── templates/
│   └── index.html       # Main frontend UI
├── static/
│   └── siren.mp3        # Alert audio
├── spoof_log.txt       # Log file (auto-generated)
```

Usage

1. Run the Interface Detector

To list available interfaces:

```
python interface.py
```

Use this to determine which network interface to monitor.

2. Start the Detector

Update `interface = "<your_interface>"` in `app.py` and run:

```
python app.py
```

Then open your browser at `http://localhost:5000`.

3. Simulate DNS Spoofing (Optional)

For testing in a safe environment:

- Use the loopback interface (`npf_loopback` on Windows, `lo` on Linux/macOS):

```
python simulate.py
```

This sends fake DNS responses to test spoof detection.

Selecting the Correct Network Interface

For Real Monitoring

- Run `ipconfig` (Windows) or `ifconfig/ip a` (Linux/macOS) to find your active IP and match that IP with the interface name from `interface.py`.

For Simulation

Use the loopback interface:

- `npf_loopback` (Windows)
- `lo` (Linux/macOS)

Set this in `detector.py` to simulate spoofed packets locally.