Real-Time Phishing URL Detector

A lightweight, open-source tool to detect phishing and malicious URLs using free blocklists, typosquatting, homoglyph detection, and sandbox analysis. Designed for both expert users (CLI with Cuckoo Sandbox) and normal users (browser extension with Browserling).

Features

Blocklist Checks: Uses OpenPhish, PhishTank, and user-defined blocklists.

Typosquatting Detection: Compares URLs against 400 legitimate domains using Levenshtein distance.

Homoglyph Detection: Identifies Unicode spoofs (e.g., google.com) via normalization and visual rendering.

Suspicious TLD Detection: Flags 50 TLDs commonly used in phishing (e.g., .xyz, .top).

Sandbox Analysis:

Normal Users: Browserling free-tier sandbox for quick URL inspection.

Expert Users: Cuckoo Sandbox for detailed analysis (HTTP, JS, files).

Customizable: Expandable config files for domains, TLDs, and user blocklists.

Project Structure

```
phishing-detector/
├── config/
│ ├── legit_domains.txt # 400 legitimate domains for typosquatting
│ ├── suspicious_tlds.txt # 50 suspicious TLDs
│ └── user_blocklist/ # User-defined phishing URLs
│ ├── custom1.txt # 100 example phishing URLs
├── src/
│ ├── cli/
│ │ ├── phishing_detector.py # Main CLI script
│ │ └── sandbox.py # Sandbox integration
│ └── extension/
│ ├── manifest.json # Browser extension manifest
│ ├── background.js # Blocklist and analysis logic
│ ├── content.js # URL extraction from webpages
│ ├── popup.html # Popup UI
│ └── popup.js # Popup logic
├── data/
│ └── blocklist_cache.db # SQLite cache for blocklists
├── requirements.txt # Python dependencies for CLI
├── setup.sh # Setup script
└── README.md # This file
```

Requirements

CLI (Expert Users)

OS: Linux, macOS, or Windows

Python: 3.8+

Dependencies (requirements.txt):

requests==2.32.3

tldextract==5.1.2

python-Levenshtein==0.26.0
Pillow==10.4.0

Optional (Expert Mode):
Cuckoo Sandbox (requires VirtualBox, Python 2.7+, guest VM)

Internet: For blocklist fetching

Browser Extension (Normal Users)

Browser: Chrome (v100+) or Firefox (v91+)
Dependencies: None (JavaScript-based)
Config Files: Host legit_domains.txt and suspicious_tlds.txt on a server (e.g., http://localhost:8080/config/) or bundle locally
Internet: For blocklist fetching and Browserling

Setup Instructions
CLI Setup

Clone or Create Directory:mkdir phishing-detector
cd phishing-detector

Install Dependencies:pip install -r requirements.txt

Run Setup Script:chmod +x setup.sh
./setup.sh

Creates config/, data/, and sample config files.
Optional Cuckoo Setup (Expert Mode):
Install: git clone https://github.com/cuckoosandbox/cuckoo.git
Follow Cuckoo docs: https://cuckoo.sh/docs/installation/
Start Cuckoo: python cuckoo.py

Browser Extension Setup

Place Files:
Create src/extension/ and add manifest.json, background.js, content.js, popup.html, popup.js.

Host Config Files:
Host legit_domains.txt and suspicious_tlds.txt on a server (e.g., http://localhost:8080/config/).
Alternatively, bundle locally and update background.js:legitDomains = (await (await fetch(chrome.runtime.getURL("legit_domains.txt"))).text()).split("\n").filter(d => d);
suspiciousTlds = (await (await fetch(chrome.runtime.getURL("suspicious_tlds.txt"))).text()).split("\n").filter(t => t);

Load Extension:
Chrome: chrome://extensions/, enable Developer Mode, click “Load unpacked,” select src/extension/.
Firefox: about:debugging#/runtime/this-firefox, click “Load Temporary Add-on,” select manifest.json.

File Descriptions and Customization
Config Files

legit_domains.txt:
Purpose: 400 legitimate domains for typosquatting detection.
Customize: Add domains (e.g., example.com) on new lines.

suspicious_tlds.txt:
Purpose: 50 TLDs for suspicious TLD detection.
Customize: Add TLDs (e.g., icu) on new lines.

user_blocklist/custom1.txt:
Purpose: 100 example phishing URLs for user-defined blocklist.
Customize: Add URLs to custom1.txt or create new files (e.g., custom2.txt) in user_blocklist/.

CLI Files

src/cli/phishing_detector.py:
Main script for URL analysis and sandbox integration.
Customize: Modify CONFIG_DIR or add new heuristic checks.

src/cli/sandbox.py:
Handles Browserling (normal) and Cuckoo (expert) sandboxing.
Customize: Adjust Cuckoo API endpoint or add report parsing.

src/utils/blocklist.py:
Fetches and caches blocklists (OpenPhish, PhishTank, user-defined).
Customize: Add new blocklist sources (e.g., other free feeds).

src/utils/homoglyph.py:
Detects homoglyphs via Unicode normalization and visual rendering.
Customize: Add more homoglyph patterns (e.g., Armenian, Hebrew).

src/utils/typosquatting.py:
Detects typosquatting using Levenshtein distance.
Customize: Adjust distance threshold or add new algorithms.

Browser Extension Files

src/extension/manifest.json:
Defines extension metadata and permissions.
Customize: Add permissions for new features (e.g., notifications).

src/extension/background.js:
Fetches blocklists and performs analysis.
Customize: Add local blocklist caching with chrome.storage.

src/extension/content.js:
Extracts URLs from webpages and highlights phishing links.
Customize: Modify highlighting style or add DOM element scanning.

src/extension/popup.html:
Popup UI for results and sandbox button.

Customize: Update styles or add settings UI.

src/extension/popup.js:
Handles popup logic and sandbox interaction.
Customize: Add result logging or user feedback options.

Usage Instructions
CLI Usage (Expert Users)

Run:python src/cli/phishing_detector.py

Input:
Choose expert mode (y/n) for Cuckoo or Browserling.
Enter a URL to analyze.

Output:
Results: “Phishing detected” (blocklist, typosquatting, homoglyph, TLD) or “Uncertain.”
For uncertain URLs, choose to open in sandbox (Cuckoo/Browserling).

Customization:
Add domains to config/legit_domains.txt.
Add TLDs to config/suspicious_tlds.txt.
Add phishing URLs to config/user_blocklist/custom1.txt or new files.

Browser Extension Usage (Normal Users)

Access:
Click the extension icon in the browser toolbar to open the popup.

Output:
Popup displays analysis for the current page’s URL.
Phishing links on the page are highlighted in red.
For uncertain URLs, click “Open in Browserling” to inspect in a sandbox.

Customization:
Update legit_domains.txt or suspicious_tlds.txt on the server or locally.
Modify background.js to add new blocklist sources or detection heuristics.

Limitations

Free blocklists may lag behind paid services.
Browserling’s free tier is limited to 3-minute sessions.
Cuckoo requires setup and VM resources.
Extension homoglyph detection is simplified (no visual rendering; CLI offers advanced checks).

Ethical Use

Scan only URLs with explicit permission.
Comply with legal and privacy regulations.
Do not target or disrupt legitimate websites.

Future Enhancements

Cache blocklists in chrome.storage for offline extension use.
Expand homoglyph patterns (e.g., Armenian, Hebrew).
Parse Cuckoo reports for automated verdicts.
Schedule blocklist updates with cron (CLI) or setInterval (extension).
Add a settings UI for the extension.

License
MIT License. See LICENSE for details.
Contact
For issues or contributions, open a pull request or contact the maintainer at [your-email@example.com].