Proof of Concept (PoC)

(made under Digisuraksha Parhari Foundation)

This PoC focuses on a CLI-based Python tool named **Homolyph**, developed to detect **homograph (homoglyph) domain attacks**, which are commonly used in phishing, spoofing, and domain impersonation attacks in cyber security and digital forensics scenarios.

Tool: Homolyph Detector

---

1. History:

The term "homograph attack" or "IDN spoofing" gained prominence in early 2000s, when researchers demonstrated how visually similar Unicode characters could be abused to impersonate legitimate websites (e.g., g00gle.com, ɢoogle.com, or xn--oogle-qmc.com).
Homolyph was developed in 2025 by a student contributor under the Digisuraksha Parhari Foundation as a CLI-based detection tool focused on catching such spoofed domains during forensic analysis or threat hunting.

---

2. Description:

Homolyph is a command-line utility built using Python that detects **visually deceptive domains** designed to impersonate trusted ones. It utilizes Unicode normalization and Levenshtein similarity comparison to identify malicious domains mimicking popular sites such as google.com, paypal.com, or facebook.com.

It operates in interactive mode or batch mode via file input and displays alerts for domains that show a high similarity to known trusted domains.

---

3. What Is This Tool About?

Homolyph is a **forensic-grade detection tool** that identifies suspicious domains based on **homoglyph similarity**, helping analysts spot potential phishing URLs, DNS impersonation, or URL typosquatting attacks. It also provides warning

alerts with similarity scores and identifies fake Unicode characters that are common in social engineering campaigns.

---

4. Key Characteristics / Features:

- Accepts domain input via CLI or file
- Performs Unicode normalization
- Calculates similarity using Levenshtein ratio
- Color-coded output (suspicious domains in red)
- Interactive "live" input mode
- Lightweight & dependency-minimal
- Ready for packaging as .exe (via PyInstaller)

---

5. Types / Modules Available:

- CLI interactive scanner (default)
- Single domain mode (--domain)
- Batch mode (--file)
- Optionally expandable to:
    - JSON/CSV reporting
    - GUI version using Tkinter
    - Web version using Flask

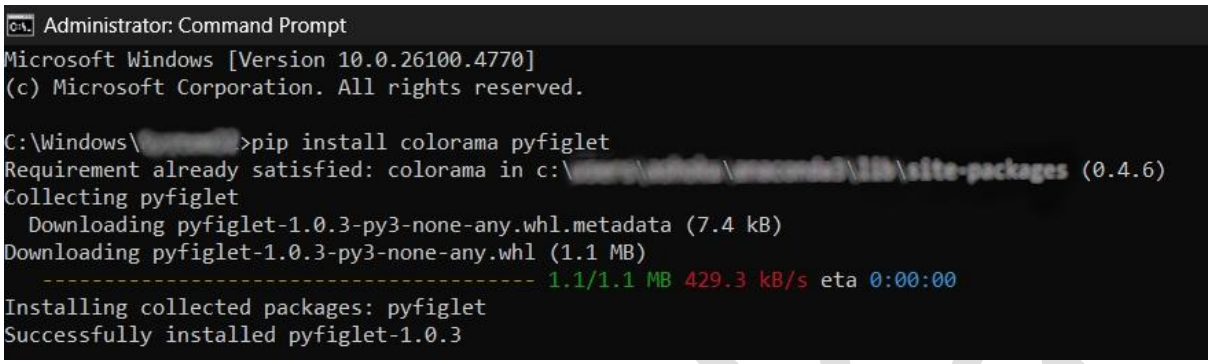
---

6. How Will This Tool Help?

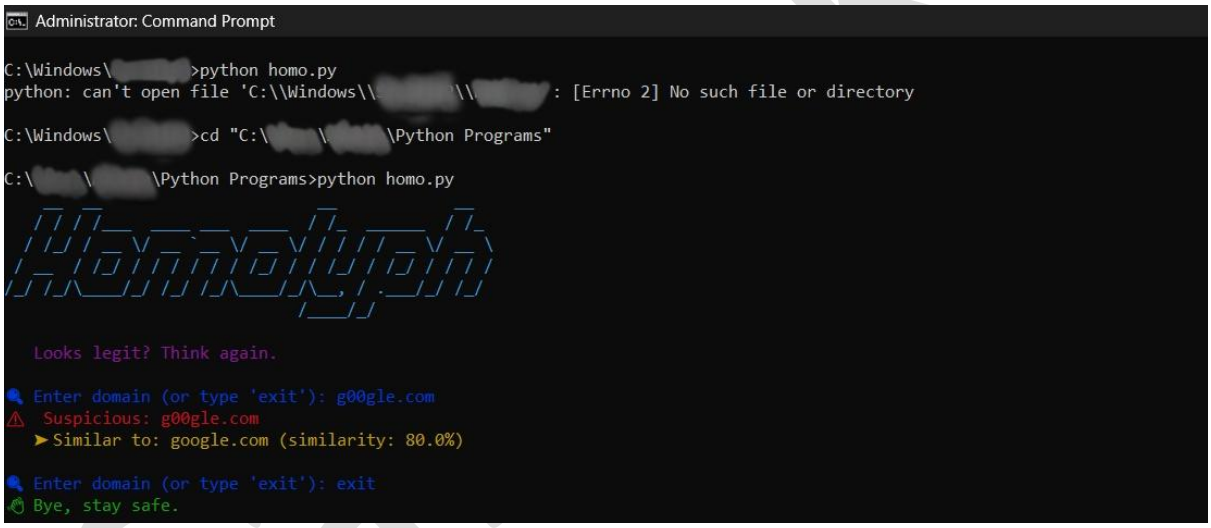
- Detect phishing domains impersonating known brands
- Assist during URL analysis in forensic cases
- Prevent clicking on lookalike malicious domains
- Verify domain legitimacy during malware triage
- Aid SOC analysts in detecting spoofing in DNS logs or emails

7. Proof of Concept (PoC) Images:

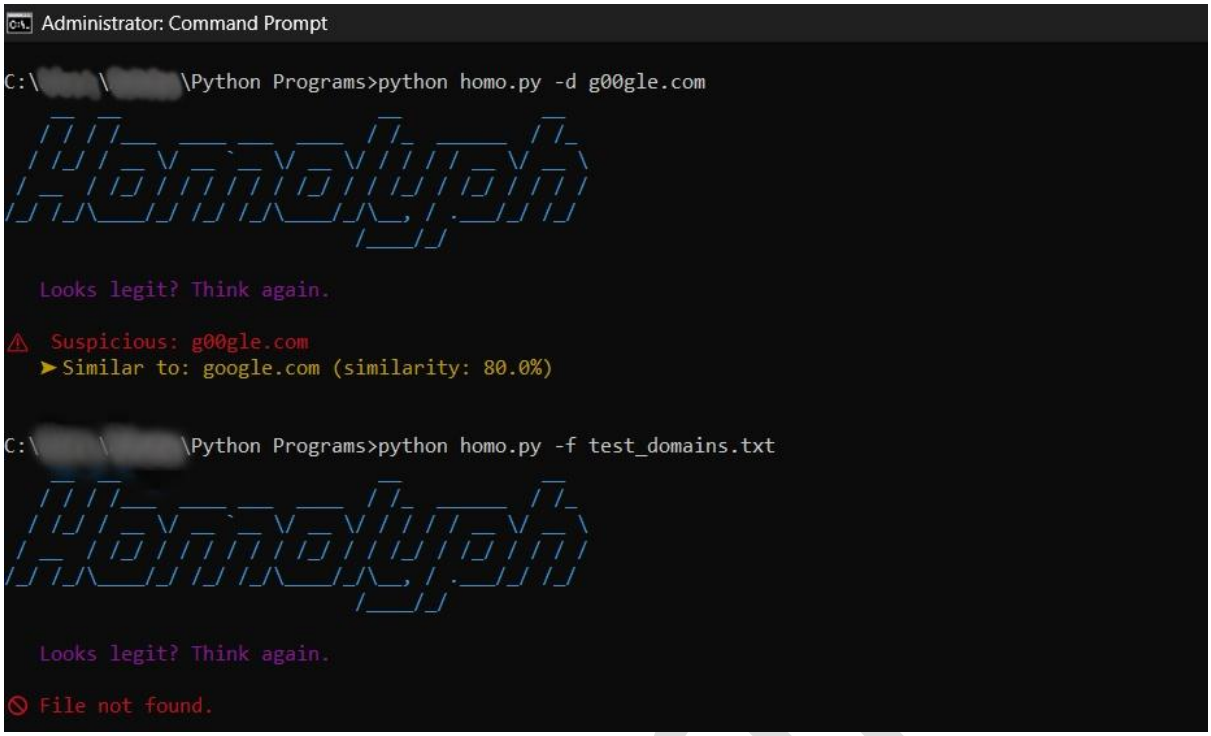
Requirements:



- Screenshot 1: **Interactive Mode**



- Screenshot 2: **Batch Mode**



🧪 Sample Suspicious Entry

Domain: google.com

Unicode-normalized: google.com

Similarity: 100%

Result: Flagged as spoofed (Greek omicron used)

---

8. 🛠 How to Set Up & Run Homolyph on a System

Requirements:

- Python 3.6+
- Modules: colorama, pyfiglet

a. Step 2: Install dependencies

pip install colorama pyfiglet

b. Step 3: Run the tool

python homo.py

c. Optional: Run in batch mode

python homolyph.py -f domains.txt

d. Optional: Create .exe (for Windows)

pip install pyinstaller

pyinstaller --onefile homolyph.py

---

9. Usage Scenarios & Examples

a. Detecting Phishing in Email Headers

Use Homolyph to scan URLs found in suspicious emails:

python homolyph.py -d g0ogle.com

b. Triaging Web Traffic Logs

Use batch mode to scan domains in proxy or firewall logs.

c. DNS Log Analysis in Incident Response

Check for spoofed domains trying to bypass allow-lists.

d. Preventing Domain Hijack

Run a scheduled check on new domain registrations in your org's environment.

---

10. 15-Liner Summary:

1. Detects homograph (homoglyph) domains
2. Built in Python
3. CLI based — minimal and fast
4. Levenshtein similarity detection
5. Supports Unicode normalization
6. Interactive input mode
7. Batch file input mode
8. Color-coded alerts
9. Flag domains impersonating trusted brands
10. Lightweight — only 2 dependencies

11. Easily expandable to GUI/Web
12. Perfect for phishing detection
13. Great for forensic triage
14. Open-source and modifiable
15. No admin rights required

---

11. Time to Use / Best Case Scenarios:

- During email phishing investigations
- When analyzing shady URLs
- During domain reputation checks
- DNS log or URL filter analysis
- Post-incident forensic review

---

12. When to Use During Investigation:

- In the initial triage of an email phishing case
- During domain intel enrichment in SOC
- While analyzing IOC lists from malware samples
- In red-team engagements to test domain lookalikes

---

13. Best Person to Use & Required Skills:

Best User:

- SOC Analyst
- Threat Hunter
- DFIR Practitioner

Skills Needed:

- Basic terminal/command-line usage

- Understanding of phishing tactics
- Familiarity with domain spoofing attacks

---

14. Flaws / Suggestions to Improve:

- Doesn’t check real-time DNS resolution or WHOIS
- No integrated threat intel feeds (e.g., VirusTotal)
- No confusable character highlight yet
- File-only input; lacks API mode
- Doesn’t auto-learn new trusted domains

---

15. Good About the Tool:

- Fast and minimal
- Easy to use and extend
- Great for triage & PoC
- Clean and informative output
- Runs cross-platform
- Ideal for low-resource setups

- Made by
  Sarthaka Subhankara
  Singh
  Intern ID – 438
  Digisuraksha Parhari
  Foundation