By: Tanaya Suryawanshi (305)

Tool Name: Homoglyph Link Checker

Description:

Homoglyph attacks use characters from different scripts that look nearly identical to Latin letters. This proof-of-concept Python tool analyzes domain names to uncover such deceptive patterns. It is designed for command-line use, relying on Unicode normalization and a similarity check to detect spoofed domains.

What Is This Tool About?

This tool serves cybersecurity and threat analysis purposes by identifying domain names that use Unicode homoglyphs to mimic real domain names. It normalizes suspicious characters and compares the result to a whitelist of legitimate domains to detect spoofing attempts.

Key Characteristics / Features:

- Takes domain names as input from users.
- Detects visually similar Unicode homoglyphs in domain names.
- Uses normalization (NFKC) to convert homoglyphs into standard characters.
- Matches normalized domains against a whitelist of safe domains.
- Flags suspicious domains with similarity reports.
- Command-line interface with real-time detection.
- Lightweight and requires only standard Python libraries.
- Extensible for larger whitelists or advanced similarity checks.

Types / Modules Available:

- Core homoglyph mapping and normalization module
- Domain whitelist module
- Similarity matching engine using difflib
- CLI interface for input/output

How Will This Tool Help?

- Helps users avoid phishing links using deceptive domain names.
- Supports security analysts during threat hunting or log reviews.
- Aids awareness training with real-world examples of spoofed domains.
- Can be embedded into automated security checks for URLs.

Proof of Concept (PoC)

Example Usage (interactive format):

Input:

```
google.com
```

Output:

```
Normalized domain: google.com

⚠️ Suspicious domain detected!

Looks similar to: google.com
```

A Brief Summary:

- Reads domain name input from user.
- Maps visually deceptive Unicode characters to ASCII equivalents.
- Applies NFKC normalization using Python's unicodedata module.
- Checks similarity with a whitelist using difflib.
- Alerts user if spoofing is suspected.
- Command-line based interface for lightweight usage.
- Low resource, extensible Python script.

Time to Use / Best Case Scenarios:

- During review of suspicious links reported by users.
- To audit link safety before sending communication.
- When whitelisting or filtering incoming domains.
- During threat intelligence or red team assessments.

Best Person to Use This Tool & Required Skills:

Best suited for: SOC analysts, cybersecurity engineers, IT support, red-team members.

Required skills:

- Basic knowledge of Python
- Familiarity with phishing tactics
- Understanding Unicode and domain spoofing (optional)

Flaws / Suggestions to Improve:

- Currently works with user input only; extend to batch URL scanning.
- Add support for reading from text/email logs.
- Integrate threat intelligence sources for better validation.
- Add optional output to CSV or PDF for auditing.

Good About the Tool:

- Very lightweight and easy to run.
- No external dependencies needed.
- Works across platforms (Windows/Linux/macOS).
- Good for security awareness demos and phishing tests.
- Highly customizable with more homoglyphs or smarter matching.

In Summary:

The Homoglyph Domain Detector tool is designed to help users detect potentially malicious domains that use visually deceptive characters. By automating homoglyph detection and offering simple yet effective matching against trusted domain names, it provides an accessible solution to phishing risk reduction.

Imagine you're about to click on a link that looks like it says 'google.com', but it's actually using special characters from another language that just look like the English letters. This tool helps you spot such tricks before it's too late. Whether you're reviewing links in chat logs, emails, or just being cautious, this tool highlights the suspicious ones so you can avoid danger.

Proof of Concept (PoC):

Let's say a user enters a domain like: **google.com** (note that the '**g**' is not a regular '**g**'). The tool will detect this character, normalize it to the standard version, and then compare it with known safe domains like 'google.com'. If it finds a suspicious similarity, it raises a flag.

Example Input: google.com

Tool Output:

```
Normalized domain: google.com

⚠️ Warning: This domain looks suspicious! It's very similar to: google.com
```

A Quick Recap:

This tool is for anyone who wants to stay safe online. It takes in a domain name, checks it for any sneaky characters, and tells you if something looks off. Whether you're part of a security team or just cautious about clicking links, this script can help you avoid phishing traps.

When Should You Use It?

- Checking unfamiliar links in an email.
- Analyzing suspicious URLs in a chat conversation.
- Pre-screening links before they're shared in a company newsletter.
- As part of an automated script that scans for threats in documents.

Who is This Tool For?

This tool is great for security analysts, ethical hackers, IT support staff, and really anyone who deals with URLs on a daily basis.

You don't need to be a Python expert. If you know how to run a script and recognize basic phishing tactics, you're good to go.

Why This Tool is Useful:

It's small, fast, and simple. It doesn’t need fancy software or setups. Just run it and start detecting. It helps people catch scams before they fall for them, and that alone makes it a pretty powerful piece of code.

Conclusion:

Homoglyph attacks might seem subtle, but their impact is serious. This tool shines a light on these hidden dangers in a way that’s easy to use and understand. Whether you’re new to cybersecurity or a seasoned analyst, it’s a helpful companion in staying safe online.