

# Proof of Concept Report

## Homograph-based Short Link Redirection PoC

### 1. Objective

This PoC demonstrates how homograph attacks can be simulated using a local short-link redirection tool. It visually replaces letters in a legitimate domain with Unicode look-alikes to make phishing links appear genuine.

### 2. Technical Overview

Language: Python 3

#### Libraries Used:

- Tkinter – GUI for user interaction
- Flask – Local web server for redirecting links
- pyperclip – Clipboard management
- random, string – Token generation

#### Homograph Mapping:

'a' → ['a', 'ᴀ']

'o' → ['o', 'ᵒ']

'l' → ['l', 'ℓ', 'ℓ']

#### Workflow:

1. Input a long URL.
2. Select or enter a target domain.
3. Replace characters with Unicode look-alikes.
4. Generate a fake preview link & a local redirect link.
5. Store mapping in Flask backend.
6. Clicking link redirects locally to original URL.

### 3. PoC Execution Steps

Step 1 — Setup:

```
pip install flask pyperclip  
python homograph_poc.py
```

Step 2 — Input Long URL:

Example: <https://www.youtube.com>

Step 3 — Choose Domain:

Select from dropdown (e.g., google.com) or enter a custom one.

Step 4 — Generate Short Link:

Click 'Generate Short Link' → The tool creates:

- Preview Link: Homograph altered (e.g., <http://google.com/x9A3dQ>)

- Local Link: Redirects via 127.0.0.1:5000

Step 5 — Test Redirect:

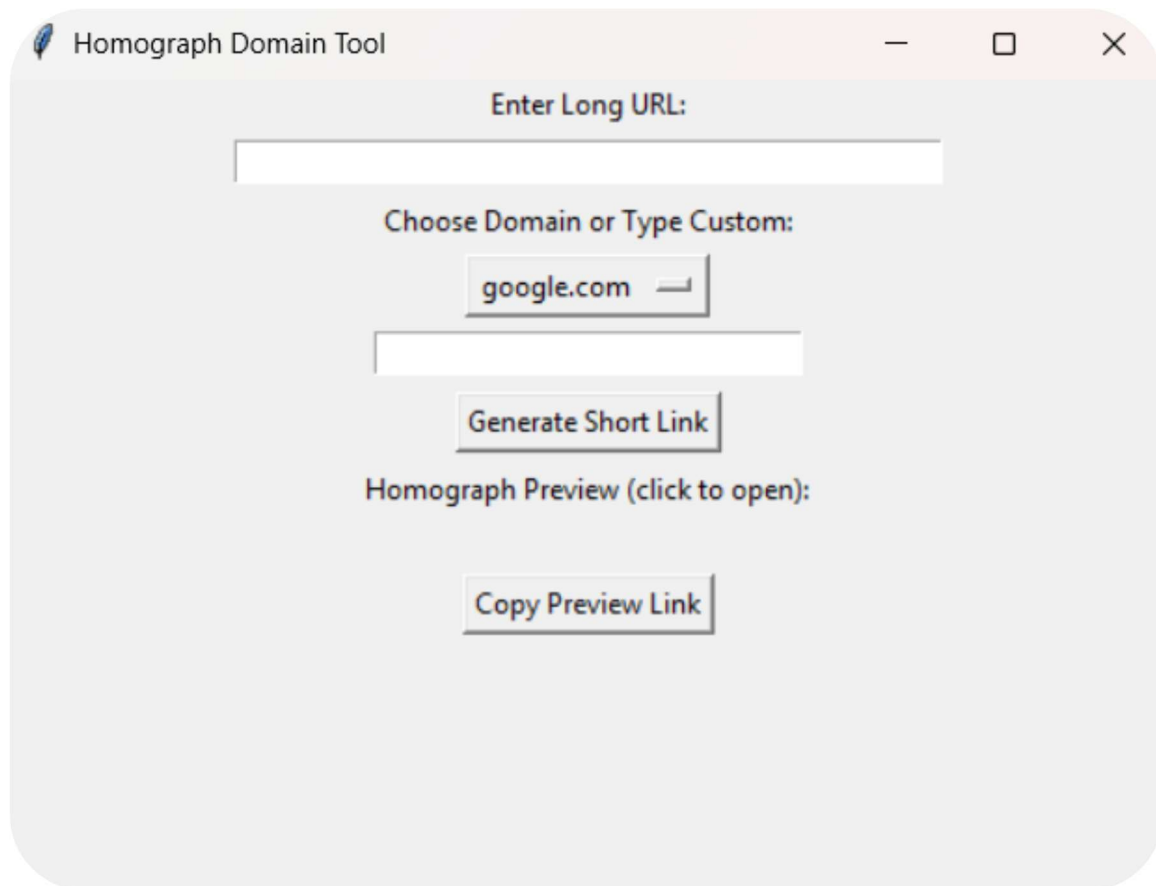
Click the preview link → Browser opens local Flask redirect → Sends you to original URL.

Step 6 — Copy Link:

Click 'Copy Preview Link' to copy the Unicode-altered domain to your clipboard.

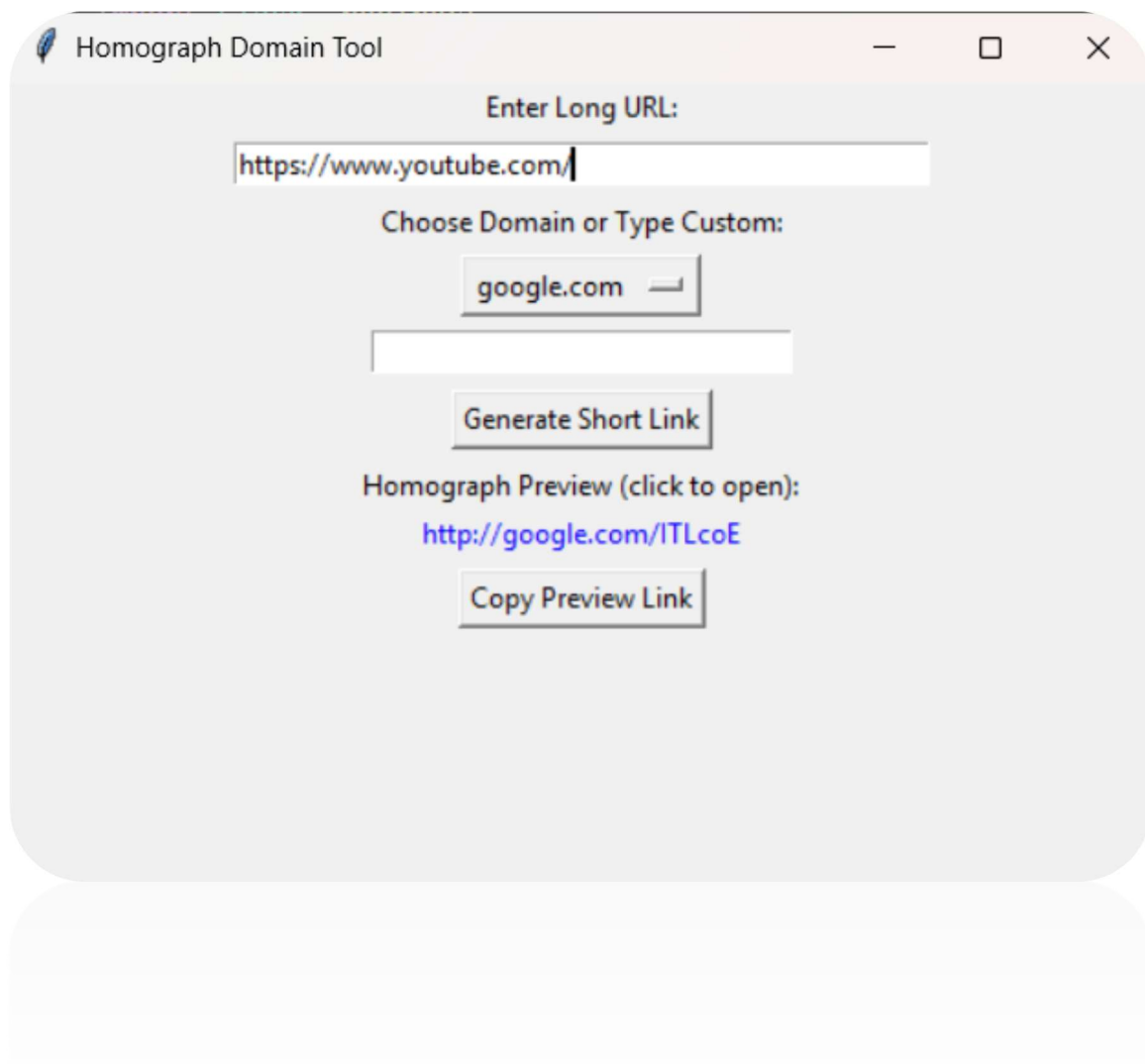
#### 4. Screenshot Placeholders

1. Tool Interface on Launch:



The screenshot shows a web application window titled "Homograph Domain Tool". The interface is light gray with rounded corners. At the top, there is a header bar with the title and standard window controls (minimize, maximize, close). Below the header, the main content area contains the following elements: a label "Enter Long URL:" followed by a text input field; a label "Choose Domain or Type Custom:" followed by a dropdown menu showing "google.com"; another text input field; a button labeled "Generate Short Link"; a label "Homograph Preview (click to open):"; and a button labeled "Copy Preview Link".

## 2. After Generating a Link:

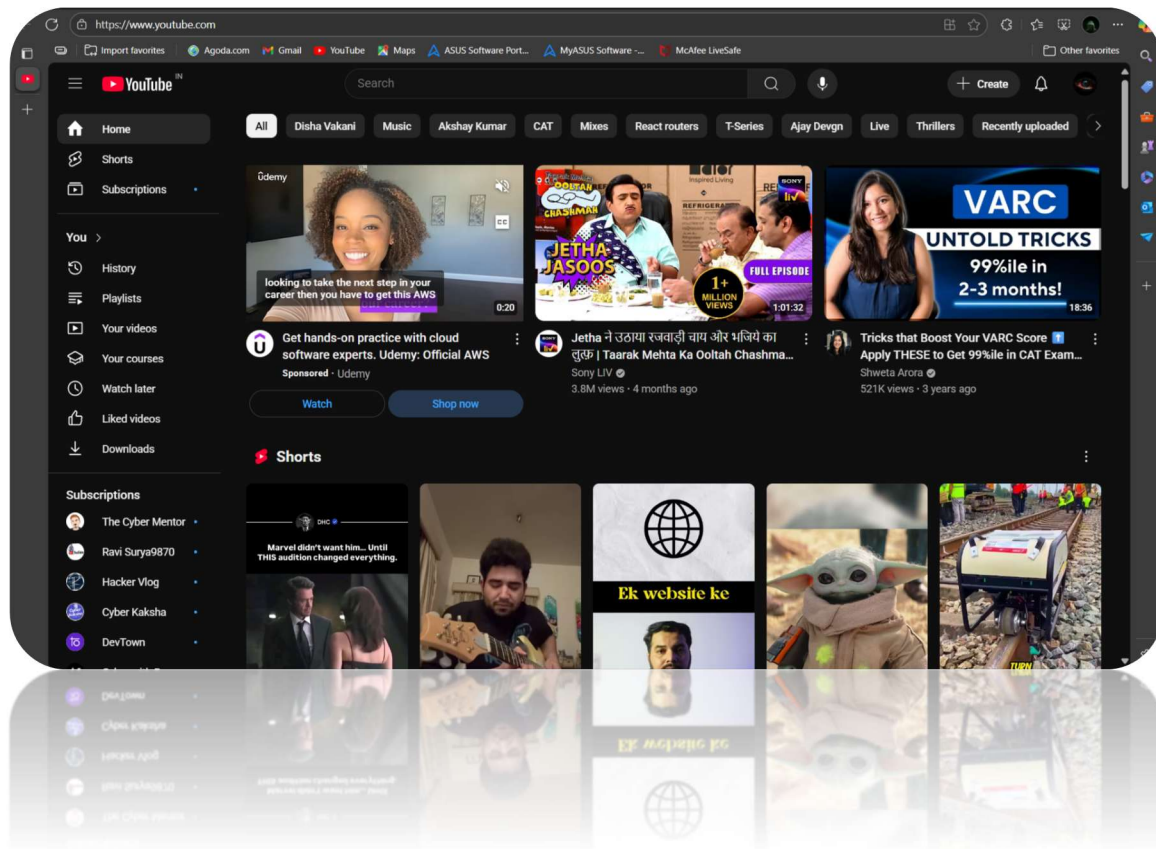


The screenshot shows a web browser window titled "Homograph Domain Tool". The interface is light gray with rounded corners. At the top, there's a header bar with the title and standard window controls (minimize, maximize, close). Below the header, the main content area contains the following elements:

- A label "Enter Long URL:" followed by a text input field containing "https://www.youtube.com/".
- A label "Choose Domain or Type Custom:" followed by a dropdown menu showing "google.com" and a small arrow icon.
- A text input field for a custom domain, which is currently empty.
- A button labeled "Generate Short Link".
- A label "Homograph Preview (click to open):" followed by a blue, underlined link "http://google.com/ITLcoE".
- A button labeled "Copy Preview Link".

The interface is clean and functional, designed for users to quickly generate homograph links.

### 3. Browser Redirect via Local Flask Server:



### 5. Conclusion

This PoC successfully demonstrates how homograph attacks can be simulated in a safe environment using Python's Tkinter + Flask. The generated fake-looking links highlight the potential for phishing campaigns and underscore the importance of user awareness & domain inspection before clicking links.