

Intern Id: 263

POC: All-in-One Homograph Generator, Detector & Link Shortener Tool

1. Objective:

The tool is designed to demonstrate and educate about:

1. Homograph Attacks (Unicode spoofing of domains).
2. Detection of homograph-based phishing domains.
3. Creating embedded hyperlink text (link shortener style) to showcase how attackers can disguise links.

It is aimed at cybersecurity awareness, phishing training, and secure browsing practices.

2. Background

- Homograph Attacks: Exploit Unicode characters that visually resemble ASCII characters to spoof domains.
- Detection Mechanism: Identifies suspicious homoglyphs in domain names and warns users.
- Hyperlink Embedding (Link Shortener Concept): Demonstrates how links can be disguised using clickable text, often misused in phishing emails.

3. Features

- ✓ Homograph Generator: Converts normal domain text into spoofed Unicode homoglyphs.
- ✓ Homograph Detector: Scans domains for suspicious Unicode and flags phishing attempts.
- ✓ Custom Link Creator: Creates clickable text embedding a hidden link (mimicking phishing link techniques).

4. Technical Workflow

- Input Stage: User provides text/domain or inputs display text & link for hyperlink creation.
- Processing:
 - Generator: Replaces ASCII characters with Unicode homoglyphs.
 - Detector: Checks each character, converts homoglyphs to ASCII, and flags suspicious domains.
 - Hyperlink Creator: Embeds target link inside user-defined display text.
- Output:
 - Generator: Spoofed domain preview with highlighted swapped characters.

- Detector: Shows Safe/Warning message and normalized domain.
- Link Shortener: Outputs embedded clickable link.

5. Example Testing Scenarios (A)

Homograph Generator:

Input: <https://www.google.com>

Output: <https://www.google.com> (o replaced by Cyrillic o).

(B) Homograph Detector:

Input: <https://facebook.com> (Cyrillic mix)

Output: Warning: Homograph attack detected.

(C) Link Creator:

Display Text: youtube.com

Link: <https://google.com>

Output: "youtube.com" (click opens google.com).

6. Security Insight:

Link Embedding Risks: Attackers use display text (like "PayPal Login") linked to malicious URLs.

Homograph Risks: Unicode scripts (Cyrillic, Greek) used in spoofing are hard to visually distinguish.

Real-World Exploit: Combined homograph spoofing + embedded links often bypass basic user awareness.

7. Mitigation

- Always hover over links to see real URLs before clicking.
- Use browsers that display punycode (xn-- format).
- Train users via phishing simulation campaigns.
- Employ email filters that detect embedded spoofed URLs.

8. Future Enhancements

Add QR Code phishing demo (link encoded in QR codes).

Use APIs to cross-check URLs against threat intelligence databases.

Browser extension version for real-time link scanning.

9. Source Code

```
<!DOCTYPE html>

<html>
<head>

<title>All-in-One Tool (Homograph Generator + Detector + Hyperlink Creator)</title>

<style>  body { font-family: Arial, sans-serif; background: #f7f7f7; padding: 30px;
color: #333; }

.tool-container { max-width: 500px; margin: 25px auto; background: #fff; padding: 22px 28px;
border-radius: 10px; box-shadow: 0 2px 12px #e0e0e0; }  h2, .logo { text-align: center; font-
weight: bold; margin-bottom: 15px; color: #25a244; }

input { width: 100%; padding: 10px; font-size: 15px; margin-bottom: 12px; border-radius: 5px;
border: 1px solid #bbb; }

button { cursor: pointer; border: none; color: #fff; font-size: 14px; padding: 8px 12px;
border-radius: 5px; margin: 5px 3px; }

.btn-green { background: #25a244; }

.btn-green:hover { background: #188038; }

.btn-blue { background: #1976d2; }

.btn-blue:hover { background: #145ba1; }

.btn-purple { background: #6a1b9a; }

.btn-purple:hover { background: #4a148c; }

.output, .alert { padding: 10px; border-radius: 6px; margin-top: 10px; font-size: 16px; word-break:
break-word; }

.output { background: #f4f4f4; font-family: monospace; }

.swapped { color: #d7263d; font-weight: bold; }

.note { font-size: 12px; color: #777; margin-top: 5px; }

.alert.danger { background: #ffefef; color: #d7263d; border: 1px solid #f6cac7; }

.alert.safe { background: #eeffe6; color: #218c5a; border: 1px solid #b2daf7; }

</style>

</head>

<body>
```

```

<!-- Homograph Generator -->

<div class="tool-container">
  <div class="logo">Homograph Generator</div>
  <input id="inputText" type="text" placeholder="Enter text (e.g. google.com)" />
  <button class="btn-green" onclick="generate()">Generate</button>
  <div class="output" id="output"></div>
  <div class="note">Red = swapped Unicode character</div>
</div>

<!-- Homograph Detector -->

<div class="tool-container">
  <h2>Homograph Detector</h2>
  <input type="text" id="domainInput" placeholder="Enter domain (e.g. facebook.com)" />
  <button class="btn-blue" onclick="checkHomograph()">Check</button>
  <div id="result"></div>
</div>

<!-- Hyperlink Creator -->

<div class="tool-container">
  <h2>Custom Link Shortner</h2>
  <input type="text" id="displayText" placeholder="Text to display (e.g. youtube.com)">
  <input type="text" id="targetLink" placeholder="Link to open (e.g. https://google.com)">
  <button class="btn-purple" onclick="makeLink()">Create Link</button>
  <div class="output" id="linkOutput"></div>
</div>

<script>
  /* ----- Homograph Generator ----- */
  const homoglyphTable = {
    'A': 'A', 'B': 'B', 'C': 'C', 'E': 'E', 'H': 'H', 'I': 'I', 'J': 'J', 'K': 'K', 'M': 'M',

```

```

'N': 'N', 'O': 'O', 'P': 'P', 'S': 'S', 'T': 'T', 'X': 'X', 'Y': 'Y', 'a': 'a', 'c': 'c',
'e': 'e', 'i': 'i', 'j': 'j', 'o': 'o', 'p': 'p', 's': 's', 'x': 'x', 'd': 'd', 'q': 'q',
'y': 'y', 'r': 'r', 'v': 'v', 'w': 'w'
};

function generate() {
    const text =
document.getElementById('inputText').value;
    let result = "";
    for (let ch of text) {
        result += homoglyphTable[ch] ? `<span class="swapped">${homoglyphTable[ch]}</span>` : ch;
    }
    document.getElementById('output').innerHTML = result || '(No input)';
}

/* ----- Extended Homograph Detector ----- */
const extendedHomographs = {

    "A": ["A", "A", "A", "Á", "À", "Â", "Ä", "Ã", "Å", "њ", "A"],

    "B": ["B", "B", "B", "Ƀ", "B"],

    "C": ["C", "C", "Ҫ", "Ҫ", "Ҫ", "Ҫ"],

    "D": ["d", "d", "ڏ", "ڏ", "ڏ"],

    "E": ["E", "E", "Ѐ", "Ѐ", "Ӗ", "Ӗ", "Ӗ", "Ӗ", "Ӗ", "Ӗ", "Ӗ", "Ӗ"],

    "F": ["F", "F", "ڦ"],

    "G": ["G", "Ӯ", "Ӯ", "Ӯ", "Ӯ"],

    "H": ["H", "Ӯ", "Ӯ", "Ӯ"],

    "I": ["I", "I", "Ӱ", "Ӱ", "Ӱ", "Ӱ", "Ӱ", "Ӱ", "Ӱ", "Ӱ", "Ӱ"],

    "J": ["J", "ڇ"],

    "K": ["K", "K", "Ӯ", "Ӯ"],

    "L": ["L", "L", "Ӆ", "Ӆ", "Ӆ", "Ӆ"],

    "M": ["M", "M", "Ӎ"],

    "N": ["N", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ"],

    "O": ["O", "O", "O", "Ó", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ", "Ӯ"]
}

```

"P": ["P", "P", "P\u00f3", "P"],
"Q": ["Q"],
"R": ["R", "R", "R\u00e1", "R\u00e1"],
"S": ["S", "S", "S\u00e1", "S\u00e1", "S\u00e1", "S\u00e1"],
"T": ["T", "T", "T\u00e1", "T\u00e1", "T\u00e1"],
"U": ["U", "U", "U\u00e1", "U\u00e1"],
"V": ["V", "V", "V\u00e1"],
"W": ["W", "W", "W\u00e1", "W\u00e1", "W\u00e1", "W\u00e1"],
"X": ["X", "X"],
"Y": ["Y", "Y", "Y\u00e1", "Y\u00e1", "Y\u00e1", "Y\u00e1", "Y\u00e1"],
"Z": ["Z", "Z", "Z\u00e1", "Z\u00e1", "Z\u00e1"],
"a": ["a", "á", "à", "â", "ä", "ã", "å", "ă", "ä"],
"b": ["b", "b", "b\u00e1", "b\u00e1"],
"c": ["c", "ç", "c\u00e1", "c\u00e1", "c\u00e1"],
"d": ["d", "d", "d\u00e1", "d\u00e1"],
"e": ["e", "é", "è", "ê", "ë", "ě", "ë", "é", "é", "é"],
"f": ["f", "f", "f\u00e1"],
"g": ["g", "g", "g\u00e1", "g\u00e1"],
"h": ["h", "h", "h\u00e1", "h\u00e1"],
"i": ["i", "í", "í", "í", "í", "í", "í", "í", "í", "í"],
"j": ["j", "j"],
"k": ["k", "k", "k\u00e1", "k\u00e1"],
"l": ["l", "í", "í", "í", "í"],
"m": ["m", "m"],
"n": ["n", "ñ", "ń", "ń", "ń", "ń"],
"o": ["o", "o", "ó", "ò", "ô", "ö", "ó", "ó", "ó", "ó", "ó", "ó", "ó", "ó", "ø"],
"p": ["p", "p", "p", "p"],
"q": ["q"],
"r": ["r", "í", "í", "í"],
"s": ["s", "s", "s", "s", "s", "s"]

```

    "t": ["T", "τ", "t̄", "t̄̄", "t̄̄̄"],

    "u": ["u", "ú", "ù", "û", "ü", "ū", "ũ", "û", "ú", "ú̄", "ú̄̄"],

    "v": ["v", "v̄", "v̄̄"],

    "w": ["w", "ŵ", "̀w", "̀w̄", "̀w̄̄"],

    "x": ["χ", "x"],

    "y": ["γ", "ý", "ŷ", "ŷ̄", "ŷ̄̄", "ŷ̄̄̄"],

    "z": ["ζ", "ž", "ž̄", "ž̄̄", "ž̄̄̄"]
};

function checkHomograph() {
    const domain =
        document.getElementById('domainInput').value;

    let suspicious = false;
    let converted = "";

    for (let ch of domain) {
        let found = false;
        for (let [base,
            homoglyphs] of Object.entries(extendedHomographs)) {
            if (homoglyphs.includes(ch)) {
                suspicious = true;
                converted += base;
                found = true;
                break;
            }
        }
        if (!found) converted += ch;
    }

    const resultDiv = document.getElementById('result');
    resultDiv.innerHTML = suspicious
        ? `<div class="alert danger">⚠ Warning: This may be a homograph attack!<br>Converted:<strong>${converted}</strong></div>`:
        : `<div class="alert safe"> ⓘ This domain looks safe (no obvious homoglyphs found).</div>`;
}

```

```

/* ----- Hyperlink Creator ----- */

function makeLink() {

    const text = document.getElementById('displayText').value.trim();

    const url = document.getElementById('targetLink').value.trim();

    const out = document.getElementById('linkOutput');

    if (!text || !url) {

        out.innerHTML = "<span style='color:#d7263d;'>Both fields are required!</span>";

        return;
    }

    out.innerHTML = `Result: <a href="${url}" target="_blank"
style="color:#1976d2;text-decoration:underline;">${text}</a>`;

}

</script>

</body>

</html>

```

11.Output :

Homograph Generator

google

Generate

google

Red = swapped Unicode character

Homograph Detector

google

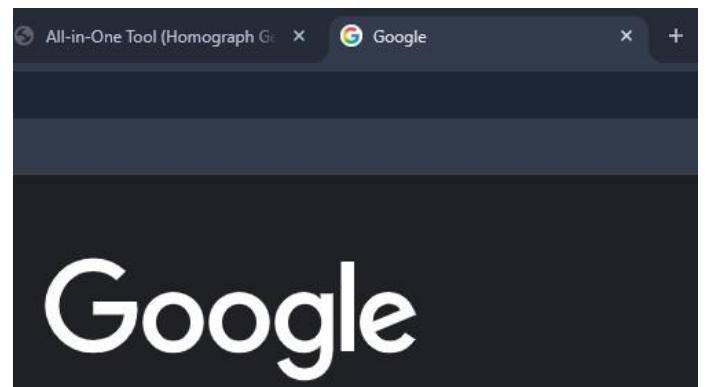
Check

⚠ Warning: This may be a homograph attack!
Converted: google

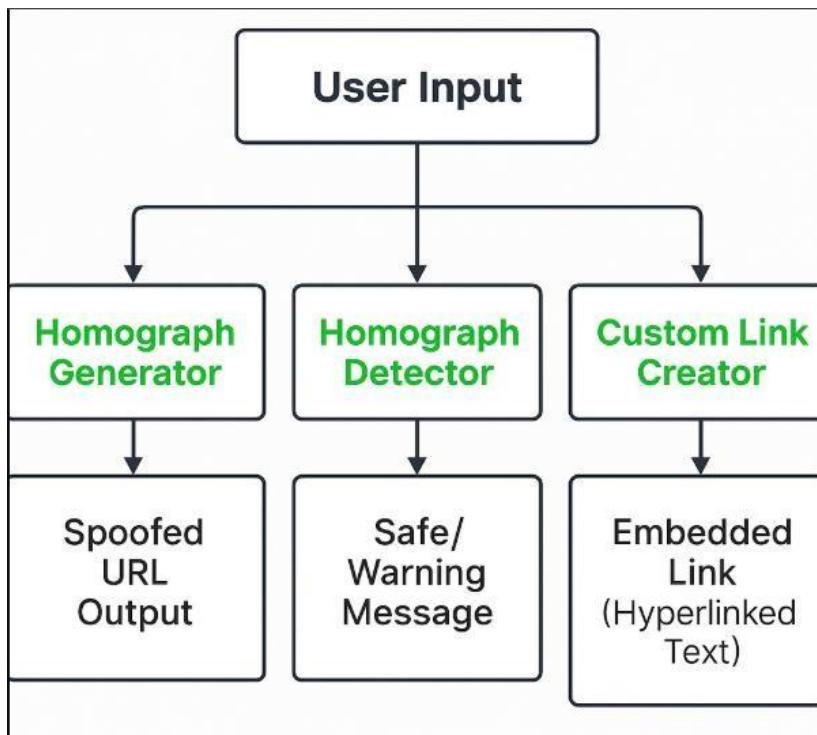
Custom Link Shortner

Create Link

Result: www.g00gle.com



12. Workflow diagram:



13. Conclusion :

This all-in-one tool effectively demonstrates three key phishing techniques (homograph spoofing, detection, and disguised hyperlinks). It is an excellent educational and practical cybersecurity awareness tool.