Intern Id: 322, 263, 242

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 + Link Shortener)</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 0; }
    .btn-green { background: #25a244; }
    .btn-green:hover { background: #188038; }
    .btn-blue { background: #1976d2; }
    .btn-blue:hover { background: #145ba1; }
    .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; }
    .shortlink { font-size: 1.13em; background: #f2f2f2; padding: 9px; border-radius: 5px; word-break: break-all; }
  </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>

<!-- Link Shortener -->
<div class="tool-container">
  <div class="logo">Custom Link Shortener</div>
  <input type="text" id="targetUrl" placeholder="Paste your target link (e.g. https://google.com)">
  <input type="text" id="alias" placeholder="Custom alias (e.g. youtube)">
  <button class="btn-green" onclick="createShortLink()">Create Short Link</button>
  <div id="resultLink"></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","Ꭺ","Á","À","Â","Ä","Ã","Å","Ă","Ą"],
  "B": ["B","Ᏼ","Ḃ","Ḅ","Ɓ"],
  "C": ["C","Ⅽ","Ç","Ċ","Ĉ","Ć"],
  "D": ["ԁ","ⅾ","Ď","Đ","Ḋ"],
  "E": ["E","Ꭼ","Ɛ","É","È","Ê","Ë","Ē","Ĕ","Ė","Ę","Ě"],
  "F": ["Ϝ","Ғ","Ḟ"],
  "G": ["G","Ԍ","Ğ","Ġ","Ģ"],
  "H": ["H","Ĥ","Ḧ","Ħ"],
  "I": ["I","Ⅰ","Í","Ì","Î","Ï","Ĩ","Ī","Ĭ","Į","İ"],
  "J": ["J","Ĵ"],
  "K": ["K","Ⲕ","Ķ","Ḱ"],
  "L": ["L","ㅣ","Ĺ","Ļ","Ľ","Ł"],
  "M": ["M","Ⅿ","Ḿ"],
  "N": ["N","Ñ","Ń","Ņ","Ň","Ṅ"],
  "O": ["O","Ο","0","Ó","Ò","Ô","Ö","Õ","Ō","Ŏ","Ő","Ø"],
  "P": ["P","Ꮲ","Ṕ","Ƥ"],
```

"Q": ["Q"],

"R": ["Г","Ŕ","Ř","Ṙ"],

"S": ["S","Ś","Ŝ","Ş","Š","Ṡ"],

"T": ["T","Т","Ť","Ţ","Ṫ"],

"U": ["U","Ú","Ù","Û","Ü","Ũ","Ū","Ŭ","Ů","Ű","Ų"],

"V": ["V","Ṽ","Ṿ"],

"W": ["W","Ɯ","Ŵ","Ẁ","Ẃ","Ẅ"],

"X": ["X","Х"],

"Y": ["Y","Ү","Ý","Ŷ","Ÿ","Ỳ","Ẏ"],

"Z": ["Z","Ź","Ż","Ž","Ẓ"],

"a": ["a","á","à","â","ä","ã","å","ă","ą"],

"b": ["Ь","ƀ","ɓ","ḃ"],

"c": ["c","ç","ċ","ĉ","ć"],

"d": ["d","ԁ","đ","ḋ"],

"e": ["e","é","è","ê","ë","ē","ĕ","ė","ę","ě"],

"f": ["ғ","ƒ","ḟ"],

"g": ["ĝ","ğ","ġ","ģ"],

"h": ["ʜ","ĥ","ḧ","ħ"],

"i": ["i","í","ì","î","ï","ĩ","ī","ĭ","į","ı"],

"j": ["j","ĵ"],

"k": ["ĸ","к","ķ","ḱ"],

"l": ["ӏ","ĺ","ļ","ľ","ŀ"],

"m": ["м","ḿ"],

"n": ["п","ñ","ń","ņ","ň","ṅ"],

"o": ["o","о","ó","ò","ô","ö","õ","ō","ŏ","ő","ø"],

"p": ["p","ρ","ṕ","þ"],

"q": ["q"],

"r": ["r","ŕ","ř","ṙ"],

"s": ["s","ś","ŝ","ş","š","ṡ"],

"t": ["т","ţ","ť","ŧ","ṫ"],

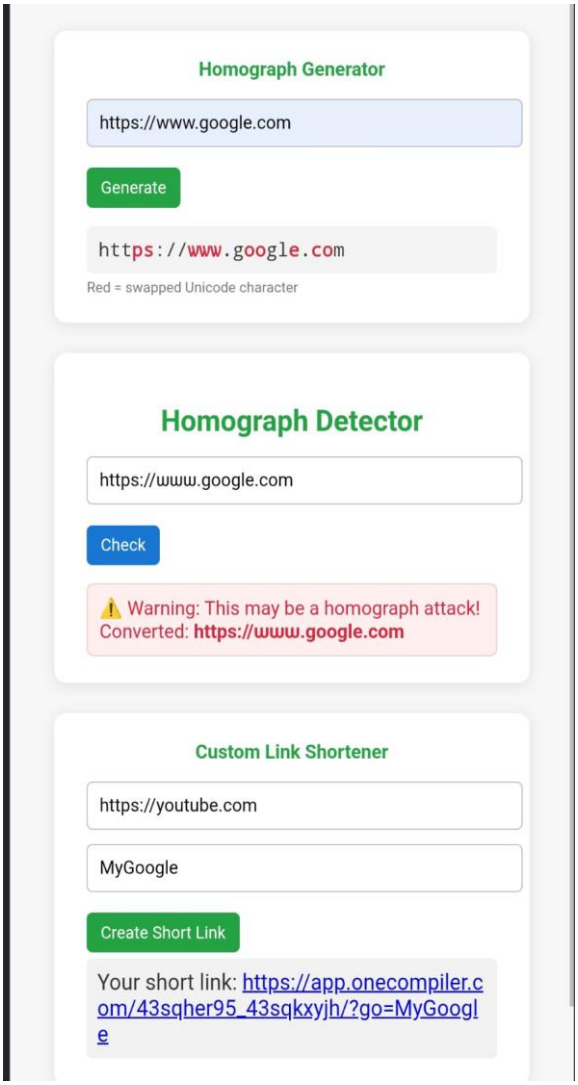
"u": ["u","ú","ù","û","ü","ũ","ū","ŭ","ů","ű","ų"],

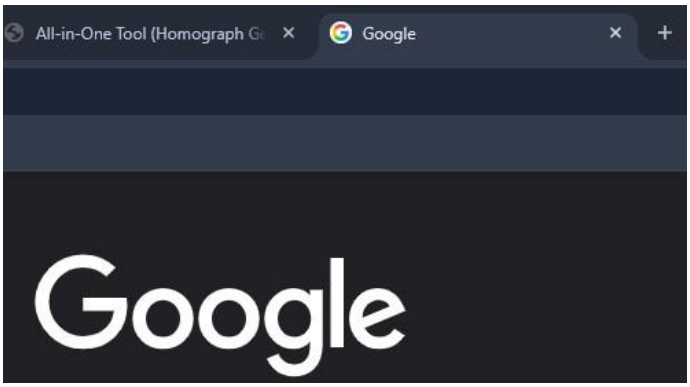
```
  "v": ["v","ṽ","ṿ"],
  "w": ["w","ŵ","ẁ","ẃ","ẅ"],
  "x": ["χ","x"],
  "y": ["γ","ý","ŷ","ÿ","ỳ","ẏ"],
  "z": ["ζ","ź","ż","ž","ẓ"]
};

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>;
}
```

```
/* ---------- Link Shortener ---------- */
function createShortLink() {
  const target = document.getElementById('targetUrl').value.trim();
  const alias = document.getElementById('alias').value.trim();
  const result = document.getElementById('resultLink');

  if(!target ||!alias) {
    result.innerHTML = "<span style='color:#d7263d;'>Please provide both the link and alias.</span>";
    return;
  }
  let base = window.location.origin + window.location.pathname;
  let shortPath = alias.replace(/[^a-zA-Z0-9]/g, '');
  if (!shortPath) {
    result.innerHTML = "<span style='color:#d7263d;'>Alias must have at least one letter or number.</span>";
    return;
  }
  let links = JSON.parse(localStorage.getItem('shortLinks') || '{}');
  links[shortPath] = target;
  localStorage.setItem('shortLinks', JSON.stringify(links));
  result.innerHTML =
    `<div class="shortlink">
      Your short link:
      <a href="${base}?go=${encodeURIComponent(shortPath)}" target="_blank">${base}?go=${encodeURIComponent(shortPath)}</a>
    </div>`;
}

window.onload = function() {
  const params = new URLSearchParams(window.location.search);
```

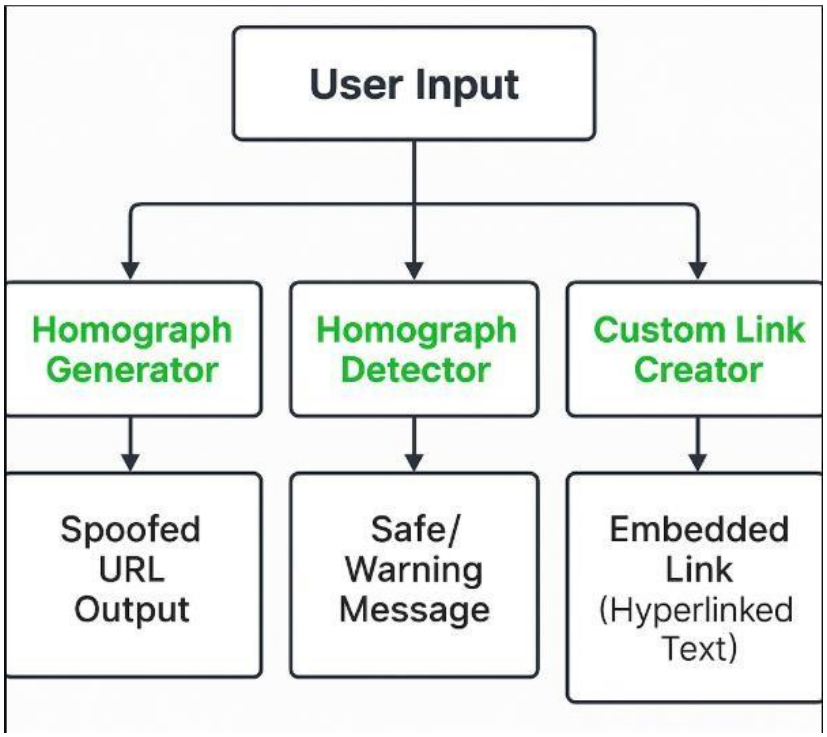
```
    if(params.has('go')) {
      let shortPath = params.get('go');
      let links = JSON.parse(localStorage.getItem('shortLinks') || '{}');
      if(links[shortPath]) {
        window.location.href = links[shortPath];
      }
    }
  }
 </script>
</body>
</html>
```

11.**Output :**





12. Workflow diagram:



https://app.onecompiler.com/43sqher95_43sqkxyjh/?go=MyGoogle

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.