Intern ID: 242

Proof of Concept: Homograph Identification and Creation Utility

Purpose:

This utility demonstrates and explores Homograph Attacks. These involve using similar-looking Unicode symbols to imitate trusted domains google.com where the second “o” is Cyrillic).

Key uses include:
- Educating users about phishing through homograph manipulation
- Testing domain names for visual spoofing risks

Understanding Homograph Threats:

Homograph attacks leverage Unicode to mimic ASCII characters. The technique is used to create deceptive URLs that appear safe to users but direct them to malicious websites.

Example:

Authentic: www.google.com
Spoofed: www. google.com (Cyrillic 'o')

Use of Unicode & IDN:

Unicode enables characters from global scripts. Attackers exploit this to build domains that resemble legitimate ones visually but differ underneath.

Potential Impacts:

- Credential harvesting
- Malware infections
- Fraudulent login sites

Tool Functionalities:

1. Domain Generator:
- Transforms normal domains using lookalike Unicode glyphs.
- Highlights swapped characters.

2. Domain Checker:
- Reviews input domains and flags visual tricks.
- Displays original and sanitized versions.

How it Works (Workflow):

1. User provides domain input.
2. Generator uses a lookup to swap ASCII letters with Unicode equivalents.
3. Detection engine scans for lookalikes.
4. Tool outputs safe/flagged messages, and visual markers.

Security Notes:

- Users might miss subtle differences.
- Even fraudulent domains can have valid SSL certificates.
- Standard filters may not flag valid Unicode domains.

Attack Examples:

- facebook.com → facebook.com (Cyrillic "a", "e", "o")

Benefits of the Tool:

- Cybersecurity education
- Phishing simulation and training
- Web-based, no installation
- Easily integrated into awareness campaigns

Planned Improvements:

Decode punycode (e.g., xn--pple-43d.com → apple.com).

-Connect with threat reputation service

-Add browser plugin capabilities

- Enrich Unicode database

Sample Test Inputs:

Input URL and its Detection Result https://www.google.com : ✅ Safe

https://www.google.com :⚠ Warning (Cyrillic "o")!

http://facebook.com :⚠ Warning (Cyrillic mix) https://example.com :✅ Safe

Countermeasures:

- Use browser punycode rendering
- Implement safe domain allowlists
- Encourage cautious URL practices
- Use MFA to limit phishing success
- Employ email scanners that detect Unicode tricks.

Source code:

```
<!DOCTYPE html>

<html>

<head>

  <title>Homograph Generator & Detector (All-in-One)</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: #b39c4d; }
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; }

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

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

    .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 it</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>




  <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 (Upper + Lower) ---------- */    const
extendedHomographs = {

    "A": ["A","А","Α","Á","À","Â","Ä","Ã","Å","Ă","Ą"],

    "B": ["B","В","Β","Ḃ","Ɓ"],

    "C": ["C","С","Ç","Ċ","Ĉ","Ć"],
    "D": ["ԁ","d","Ď","Đ","Ḋ"],

    "E": ["Е","Ε"," ","É","È","Ê","Ë","Ē","Ĕ","Ė","Ę","Ě"],

    "F": ["Ғ","Ƒ","Ḟ"],

    "G": ["Ԍ","Ĝ","Ğ","Ġ","Ģ"],

    "H": ["Н","Ĥ","Ḧ","Ħ"],

    "I": ["І","Ι","Í","Ì","Î","Ï","Ĩ","Ī","Ĭ","Į","İ"],

    "J": ["Ј","Ĵ"],

    "K": ["К","Κ","Ķ","Ḱ"],

"L": ["L"," ו ","Ĺ","Ļ","Ľ","Ł"],

    "M": ["М","Μ","Ḿ"],

    "N": ["Ν","Ñ","Ń","Ņ","Ň","Ṅ"],

    "O": ["О","Ο","0","Ó","Ò","Ô","Ö","Õ","Ō","Ŏ","Ő","Ø"],

    "P": ["Р","Ρ","Ṕ","Ᵽ"],
```

"Q": ["Q"],

"R": ["Γ","Ŕ","Ř","Ṙ"],

"S": ["Ѕ","Ś","Ŝ","Ş","Š","Ṡ"],

"T": ["Т","Τ","Ť","Ţ","Ṫ"],

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

"V": ["Ѵ","Ṽ","Ṿ"],

"W": ["Ԝ","Ꞷ","Ŵ","Ẁ","Ẃ","Ẅ"],

"X": ["Х","Χ"],

"Y": ["Υ","Ү","Ý","Ŷ","Ÿ","Ỳ","Ẏ"],

"Z": ["Ζ","Ź","Ż","Ž","Ẓ"],

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

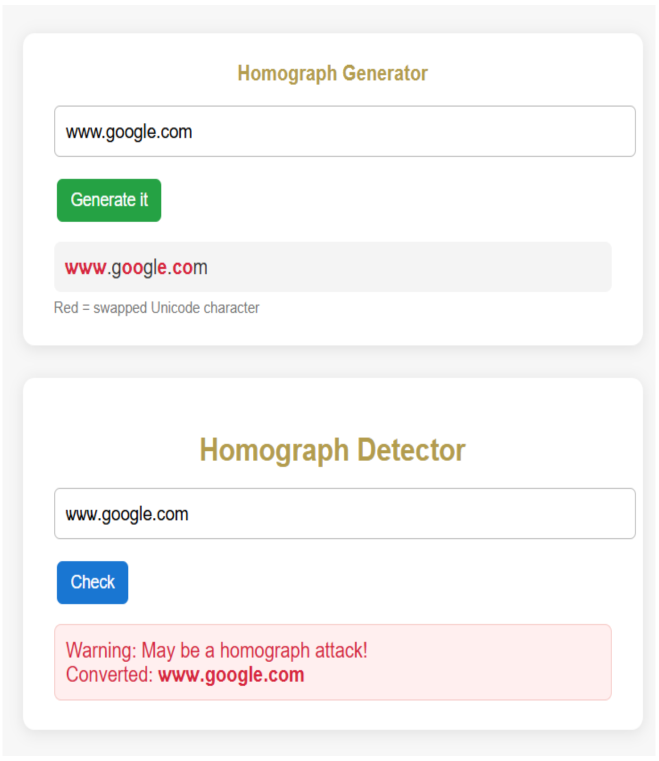
"b": ["Ь","ƀ","Ƃ","ḃ"],

"c": ["с","ç","ċ","ĉ","ć"],

```
    "d": ["d","ď","đ","ḋ"],

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

    "f": ["ғ","ƒ","ḟ"],

    "g": ["ĝ","ğ","ġ","ǵ"],

    "h": ["h","ĥ","ḧ","ħ"],

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

"j": ["ј","ĵ"],

    "k": ["κ","к","ķ","ḱ"],

    "l": ["ｌ","ĺ","ļ","ľ","ł"],

    "m": ["м","ḿ"],

    "n": ["п","ñ","ń","ņ","ň","ṅ"],

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

    "p": ["р","ρ","ṕ","ƥ"],
    "q": ["q"],

    "r": ["г","ŕ","ř","ṙ"],

    "s": ["s","ś","ŝ","ş","š","ṡ"],

    "t": ["т","т","ť","ţ","ṫ"],

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

    "v": ["ⱱ","ṽ","ṿ"],

    "w": ["w","ŵ","ẁ","ẃ","ẅ"],

    "x": ["χ","х"],

    "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: May be a homograph attack!<br>Converted: 
<strong>${converted}</strong></div>`
```

```
        : `<div class="alert safe"> This domain looks safe (no obvious homoglyphs
found...).</div>`;

    }

  </script>

</body>

</html>
```

Output:



Homograph Generator

www.google.com

Generate it

www.google.com

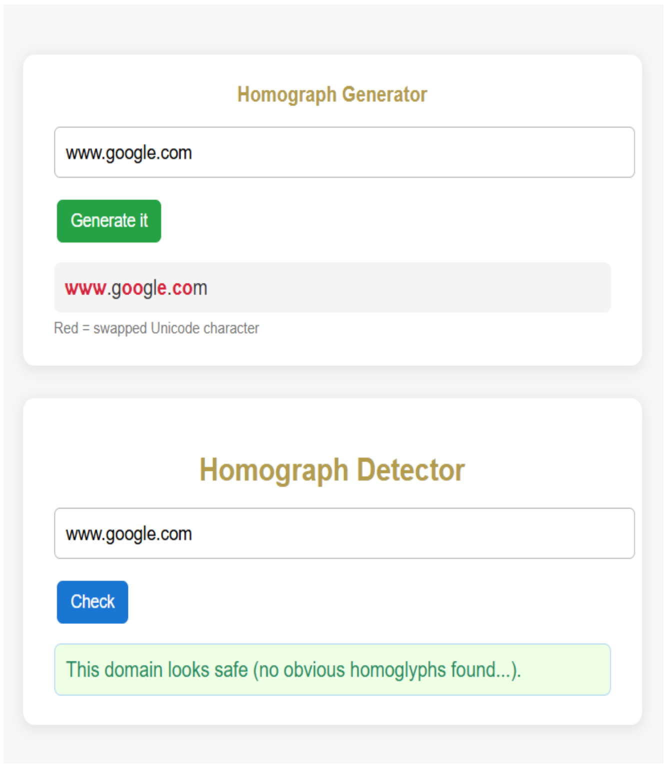
Red = swapped Unicode character

Homograph Detector

www.google.com

Check

Warning: May be a homograph attack!
Converted: **www.google.com**



Homograph Generator

www.google.com

Generate it

www.google.com

Red = swapped Unicode character

Homograph Detector

www.google.com

Check

This domain looks safe (no obvious homoglyphs found...).

Conclusion:

This utility effectively showcases how modern homograph-based phishing works and how such attacks can be flagged. It supports user training and security research applications.