**Swayam Salunke (225)**

**PoC: Homoglyph (Homographic) Domain Detector**

**Objective**

To detect potentially malicious domain names or URLs that use Unicode homoglyphs to impersonate legitimate websites. For example, attackers may use ‘ɡoogle.com’ (with script 'ɡ') to deceive users into thinking it’s ‘google.com’.

**Tools & Technologies Used**

- Python 3  
- homoglyphs library  
- Regular Expressions (re)  
- Command-line interface (CLI)

**How It Works**

1. Input: A plain text file (e.g., emails, logs, messages) containing domain names or URLs.  
2. Extraction: URLs/domains are extracted using regular expressions.  
3. Normalization & Detection:  
 - Each extracted domain is scanned for Unicode homoglyphs using the homoglyphs library.  
 - The domain is compared with its ASCII variants.  
4. Flagging:  
 - Domains that are not part of the ASCII variants of themselves are flagged as suspicious.

**Code Implementation**

import re  
import sys  
  
try:  
 import homoglyphs as hg  
except ImportError:  
 print("Please install the homoglyphs package: pip install homoglyphs")  
 sys.exit(1)  
  
def extract\_links\_from\_file(filepath):  
 with open(filepath, "r", encoding="utf-8") as f:  
 text = f.read()  
 url\_pattern = re.compile(  
 r'((?:https?://|www\.)[^\s]+|[a-zA-Z0-9-]+\.[a-zA-Z]{2,}(?:/[^\s]\*)?)'  
 )  
 return url\_pattern.findall(text)  
  
def is\_link\_suspicious(link, homoglyphs\_obj):  
 ascii\_versions = homoglyphs\_obj.to\_ascii(link)  
 return bool(ascii\_versions and (link not in ascii\_versions))  
  
def check\_links(filepath):  
 homoglyphs\_obj = hg.Homoglyphs(languages={'en', 'ru', 'el'})  
 links = extract\_links\_from\_file(filepath)  
 suspicious\_links = [  
 link for link in links if is\_link\_suspicious(link, homoglyphs\_obj)  
 ]  
  
 print(f"Links found ({len(links)}):")  
 for link in links:  
 print(" ", link)  
  
 print("\nSuspicious (potentially fake) links:")  
 if suspicious\_links:  
 for link in suspicious\_links:  
 print(" ", link)  
 else:  
 print(" None detected.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 if len(sys.argv) != 2:  
 print("Usage: python homoglyph\_checker.py your\_file.txt")  
 sys.exit(1)  
 check\_links(sys.argv[1])

**Sample Input File (sample.txt)**

Visit http://ɡoogle.com for more info.  
Here’s the legit site: https://google.com  
Suspicious: www.fаcebook.com  
Safe one: www.microsoft.com

**Sample Output**

Links found (4):  
 http://ɡoogle.com  
 https://google.com  
 www.fаcebook.com  
 www.microsoft.com  
  
Suspicious (potentially fake) links:  
 http://ɡoogle.com  
 www.fаcebook.com

**Security Relevance**

This tool is useful in:  
- Email phishing detection  
- Security log analysis  
- Safe browsing tools  
- Content filtering systems

**Future Improvements**

- Add comparison with a whitelist of top domains (e.g., Alexa top 100).  
- Highlight exact suspicious characters.  
- Create a browser extension for real-time detection.  
- Use confusables.txt from Unicode.org for an exhaustive homoglyph database.