Proof of Concept (PoC): Homograph Attack Detection Tool

NAME : CHAITANYA MAYEKAR

INTERN ID: 253

# 1. Introduction

Homograph attacks exploit the similarity between characters from different alphabets to create deceptive URLs that visually resemble legitimate ones. For example, replacing the Latin 'a' with the Cyrillic 'а' can trick users into visiting a malicious website. This PoC demonstrates a Python-based detection tool to identify such suspicious characters in URLs.

# 2. Objective

The objective of this PoC is to:  
- Detect homograph (IDN-based) attacks by scanning URLs for suspicious Unicode characters.  
- Provide security teams with a mechanism to validate potentially malicious URLs before use.

# 3. Setup & Requirements

To run this PoC, the following environment is required:  
- Python 3.x  
- validators library (`pip install validators`)

# 4. Implementation

The detection logic is implemented in Python. The script validates whether a given string is a URL, and then scans it for homograph characters. Below is the core implementation:

import validators  
  
def is\_valid\_url(url):  
 return validators.url(url)  
  
def detect\_homographs(url):  
 suspicious\_chars = {  
 'а': 'a', # Cyrillic a  
 'е': 'e', # Cyrillic e  
 'о': 'o', # Cyrillic o  
 'і': 'i', # Cyrillic i  
 'ѕ': 's', # Cyrillic s  
 }  
 found = {}  
 for cyrillic\_char in suspicious\_chars:  
 if cyrillic\_char in url:  
 found[cyrillic\_char] = suspicious\_chars[cyrillic\_char]  
 return found  
  
# Example usage  
test\_url = "http://exаmple.com" # Note: Cyrillic 'а'  
if is\_valid\_url(test\_url):  
 matches = detect\_homographs(test\_url)  
 print("Suspicious characters found:", matches)  
else:  
 print("Invalid URL")

# 5. Test Cases

Sample test cases and expected results:

|  |  |
| --- | --- |
| Input URL | Detection Result |
| http://exаmple.com | Suspicious Cyrillic 'а' detected. |
| http://google.com | No suspicious characters found. |
| http://microsоft.com | Suspicious Cyrillic 'о' detected. |
| http://banking-рay.com | Suspicious Cyrillic 'р' detected (if added to dictionary). |

# 6. Limitations

- The script currently detects only a small set of Cyrillic characters. Other Unicode characters (Greek, Armenian, etc.) are not covered.  
- The tool does not normalize or decode punycode representations.  
- Further improvements can include an expanded dictionary and integration with threat intelligence feeds.

# 7. Conclusion

This PoC demonstrates how homograph attacks can be detected at the URL level. While effective for basic scenarios, extending the dictionary of suspicious characters and integrating punycode analysis will improve its robustness against advanced attacks.