NAME: NIKHIL BAGUL **INTERNID:277**

What is a Homograph Attack?

A homograph attack is a type of social engineering attack where a threat actor uses characters that look similar to legitimate characters to create a fake website address. The goal is to trick users into visiting a malicious website that looks like a legitimate one.

How it works:

Computers use a system called Unicode to represent characters from different languages. Some characters in different alphabets look identical, or very similar, to characters in the English alphabet. For example, the Cyrillic letter "a" looks exactly like the English letter "a".

An attacker can register a domain name using these look-alike characters. For example, an attacker could register the domain "https://www.google.com/search?q=%D0%B0pple.com" using the Cyrillic "a". To the naked eye, this looks identical to the legitimate "apple.com".

Why it's dangerous:

- **Deception:** Users can be easily fooled into thinking they are on a legitimate website.
- **Phishing:** Attackers can create fake login pages to steal usernames and passwords.
- **Malware Distribution:** The fake website can be used to trick users into downloading and installing malware.
- **Bypassing Security:** Because the domain is technically different from the legitimate one, it can bypass some security filters.

Analogy:

Imagine you receive a letter in the mail that looks like it's from your bank. The logo, colors, and font are all identical. However, the letter is actually a very convincing forgery designed to trick you into revealing your account details. A homograph attack is the digital equivalent of this.

Proof of Concept (PoC)

This PoC demonstrates how to create a homographic domain name using Python.

Objective: To create a fake domain name that looks like "https://www.google.com/url?sa=E&source=gmail&q=google.com" but uses a Cyrillic 'o'.

Code:

Python

```
# -- Homograph Attack PoC --


# The legitimate domain we want to impersonate

legitimate_domain = "google.com"


# The homograph domain using a Cyrillic 'o' (U+043E)
```

```
# This character looks identical to the Latin 'o' (U+006F)
homograph_domain = "go\u043egle.com"

# We can also represent the homograph domain using its Punycode equivalent.
# Punycode is how browsers handle internationalized domain names (IDNs).
# The Punycode for our homograph domain is "xn--ggle-5qf.com"
punycode_domain = "xn--ggle-5qf.com"



# -- Verification --

print(f"Legitimate Domain: {legitimate_domain}")
print(f"Homograph Domain: {homograph_domain}")
print(f"Punycode Domain: {punycode_domain}")

# This will show that the two strings are not the same, even though they look identical.
if legitimate_domain == homograph_domain:
   print("\nThe domains are the same.")
else:
   print("\nThe domains are DIFFERENT.")
```

Explanation:

- **legitimate_domain**: This is the real domain we are trying to impersonate.
- **homograph_domain**: This is our fake domain. We've replaced the second 'o' in "google" with the Cyrillic character 'о'. While it looks the same, the computer sees it as a completely different character.
- **punycode_domain**: This is how the browser will actually interpret our homograph domain. When you type a domain with non-ASCII characters into a modern browser, it is converted to Punycode. This is the domain an attacker would actually register.

How an attacker would use this:

1. The attacker registers the Punycode domain (xn--ggle-5qf.com).
2. They set up a website at this address that is a perfect clone of the real https://www.google.com/url?sa=E&source=gmail&q=google.com.

3. They send out phishing emails with a link to google.com.
4. A user clicks the link. Their browser converts google.com to xn--ggle-5qf.com and takes them to the fake website.
5. The user, seeing what looks like the correct domain in the address bar, enters their credentials, which are then stolen by the attacker.

Mitigation Techniques

Here are three techniques to defend against homograph attacks:

1. Browser-Level Protection (Punycode Display):

- **Description:** Modern web browsers have built-in protection against homograph attacks. If a domain name contains characters from multiple different character sets (e.g., a mix of Latin and Cyrillic), the browser will display the Punycode version of the domain in the address bar.
- **How it works:** Instead of displaying the deceptive google.com, the browser will show the much more suspicious-looking xn--ggle-5qf.com. This makes it immediately obvious to the user that they are not on the legitimate website.
- **Implementation:** This is typically enabled by default in all major browsers (Chrome, Firefox, Safari, Edge). Users should ensure their browsers are always up to date to have the latest security features.

2. Email Filtering and Security Gateways:

- **Description:** Organizations can use email security solutions to detect and block emails containing homograph domains.
- **How it works:** These systems can be configured to flag or block emails that contain links with mixed character sets or that lead to known malicious Punycode domains. They can also analyze the reputation of the sending domain and look for other signs of phishing.
- **Implementation:** This requires deploying and configuring an email security gateway or using a cloud-based email security service. The rules for detecting homograph attacks would need to be enabled and customized.

3. User Education and Awareness:

- **Description:** This is a non-technical but highly effective mitigation technique. It involves training users to be vigilant and to spot the signs of a potential homograph attack.
- **How it works:** Users are taught to:
    - **Be wary of unexpected emails:** Especially those that create a sense of urgency.
    - **Hover over links:** Before clicking, hover the mouse over a link to see the actual destination URL in the bottom corner of the browser.
    - **Manually type in sensitive URLs:** For important websites like banking or email, it's safer to type the address directly into the browser rather than clicking a link.

    - **Look for the padlock:** Ensure the website is using HTTPS, indicated by a padlock icon in the address bar. While this doesn't guarantee a site is legitimate, most phishing sites will not have a valid SSL certificate.

- **Implementation:** This is achieved through regular security awareness training, phishing simulations, and clear communication about new and emerging threats.