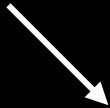


Homograph Attacks, a History

Today we're going to talk about Homograph Attacks, give some historical context and discuss why, although we first started talking about them in the 1990s, they are still relevant today.

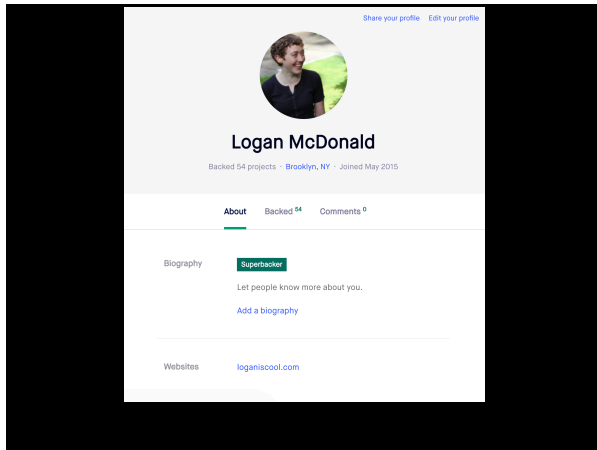
@loganmeetsworld



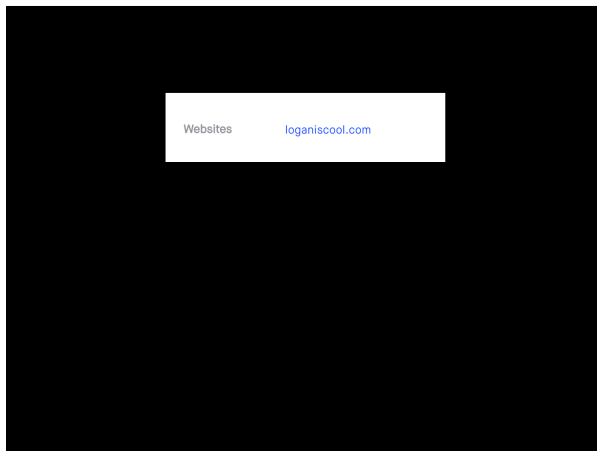
- i'm logan!
- interested in learning, want to share that with you
- this all started when I was looking at an issue in submitted to the Kickstarter bug bounty program



A bug bounty program is where an organization welcomes hacking their app in exchange for bounties for legit and fixable tickets.



If you have a site that provides profiles, you have to think about how someone is gonna use that profile to trick other people.



This hacker pointed out that our profile user url's were vulnerable to homograph attack.



And I was like "What the heck is a homograph attack?"

ICANN

**Internet Corporation for
Assigned Names and Numbers**

To do that we have to begin with ICANN

Along with performing the technical maintenance of the DNS root zone registries and maintain the namespaces of the Internet, ICANN makes all the rules about what can and cannot be a domain name.

So say you go to Namecheap to register loganisawesome.com. Namecheap uses the “extensible provisioning protocol” to verify your name with Verisign which is the organization that manages the registry for the “.com” gTLD. Verisign checks the ICANN rules and regulations for your registration attempt, tells Namecheap the result, and Namecheap tells me if I can register loganisawesome.com.

This is great. But I speak English and I use ASCII for all my awesome business on the internet. What happens to all those other languages that can't be expressed in a script compatible with ASCII?

1996–1998

**Internationalized Domain Names
Version 1**

That's the question ICANN was asking themselves when they proposed and implemented IDNs as a standard protocol for domain names. They wanted a more global internet so they opened up domains to a variety of unicode represented scripts.

Scripts

**Collection of signs in a
system**

So a script is just a collection of letters/signs for a single system; for example, Latin (supporting many languages) or Kanji (which is just one of the scripts that supports the Japanese language).

HOWEVER, there was a requirement. ICANN's Domain Name System, which performs a lookup service to translate user-friendly names into network addresses for locating Internet resources, is restricted in practice to the use of ASCII characters.

Punycode

**Puny
unicode**

The protocol for that translates names written in language-native scripts into an ASCII text representation that is compatible with the Domain Name System.

hi👋friends💖🗣️.com

xn--hifriends-
mq85h1xad5j.com

What could go wrong?

Homograph/ Homoglyph

Symbols,
words,
characters,
glyphs,
graphemes
that look alike



Well, things aren't always as they seem. And this is where homographs come in.

A homograph or homoglyph is multiple things that look or seem the same. We have many of these in English, for example "lead" could refer to the metallic substances or the past tense of "to lead."

The problem is that homoglyphs exist between scripts as well, with many of the Latin letters having copycats in other scripts, like Greek or Cyrillic.

washingtonpost.com

vs

washingtonpost.com

washingtonpost.com
vs
xn--wshingtonpost-w1k.com

BUT if you went and tried to register that second one, what would happen?

You wouldn't be able to.

2005-2008

**Internationalized Domain Names
Version 2 & 3**

ICANN told gTLD registrars they had to restrict mix scripts.

HOWEVER, while mixed scripts are not allowed, pure scripts are perfectly fine.

So we still have a problem, what about pure scripts in Cyrillic or Greek alphabets that look like the Latin characters?

```
→ ha-finder git:(master) x ruby lib/ha-finder.rb  
available domains:  
yandex.ru (yandex.ru)
```

My coworker and I had an idea to make a homograph attack detector so we made a script that :

- takes the top 1million websites
- checks if letters in each are confusable with latin/decimal
- checks to see if the punycode url is registered through a WHOIS lookup

Some of them are a little off looking (also a lot of them are porn), but we found some interesting ones you could register.

<https://paypal.com/>

[Demo]

Chrome only changed it because of Xudong Zheng's 2017 report.

user-agents IDN Display Algorithms

Chrome Algorithm

- Smart algorithm
- If it's on a gTLD and all the letters are confusable Cyrillic, show punycode in the browser
- only changed this bc of Xudong Zheng's 2017 report

Firefox Algorithm

- Way less strict than chrome

The Future of IDNs

So what, now, is our responsibility as application developers and maintainers if we think someone might use our site to phish people using a homograph? I can see a couple paths forward:


1. Advocate to Mozilla and other user-agents to make sure to change their algorithms to protect users.
2. Advocate that ICANN changes its rules around registering domains with Latin confusable characters.
3. Implement our own display algorithms. This is what we ended up doing at Kickstarter. We used Google's open-source algorithm and show a warning if it's possible that the url shown on the page is a homograph for another url.
4. Finally, we could just register these domains like PayPal so that they aren't able to be

In summary,
difficult problems are
difficult.

To summarize, this is a hard problem! That's why it's been around for two decades. And fundamentally what I find so interesting about the issues surfaced by this attack. I personally think ICANN did the right thing in allowing IDNs in various scripts. The internet should be more accessible to all.

I like Chrome's statement in support of their display algorithm, however, which nicely summarizes the tradeoffs as play:

“We want to prevent confusion, while ensuring that users across languages have a great experience in Chrome. Displaying either punycode or a visible security warning on too wide of a set of URLs would hurt web usability for people around the world.”


[github.com/
loganmeetsworld/
homographs-talk](https://github.com/loganmeetsworld/homographs-talk)


All of the notes for this talk are up here, as well as the code for find the list of available domains that can be used in a homograph attack.

Thank you,
BrooklynJS!