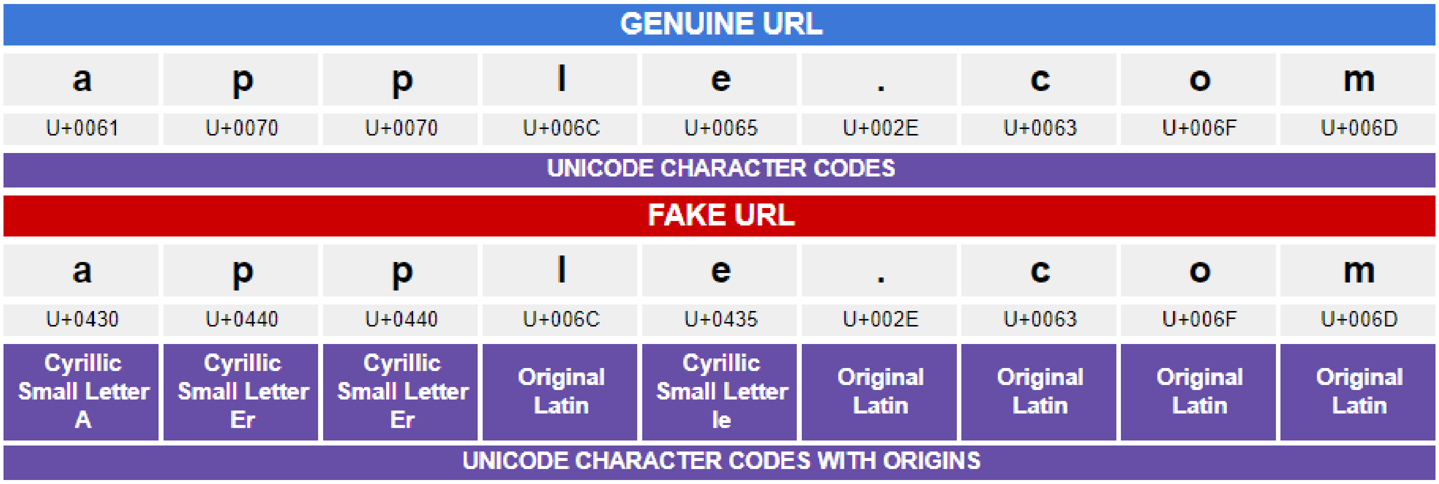
Nate Bossingham

CYSE 494

May 31, 2023

Proposal

The world of cybersecurity can be extremely confusing and difficult to understand. Most reasonably informed computer users know the basics: use an antivirus, avoid suspicious links, and don't trust emails unless you are absolutely certain that they are genuine. This is all well and good and can protect most of us a good amount of the time. Unfortunately, the attackers in the cyberspace are not so easily defeated as to give up in the face of the most baseline defenses. Every day there are new attacks developed that aim to circumvent these defenses. In the past, we have had attacks such as typo-squatting, where an attacker registers a domain name that is a slight misspelling of a popular website, and phishing, where an attacker sends out malicious emails and links that attempt to convince a victim that the links are legitimate. Both of these attacks are used to guide a user to a malicious website or file with the intent to infect the victim's system. These are the types of attacks that our commonsense guidelines are developed to combat. Unfortunately, there are now evolutions of these types of attacks that can be extremely difficult to combat. Typo-squatting and phishing have now become virtually indistinguishable from the legitimate domains that they aim to impersonate. This is accomplished by using characters from other languages in Unicode that appear virtually identical to the original URL while still technically being completely unique. This type of attack is known as the IDN homograph attack (Umawing, 2017). Figure 1 below can demonstrate how this attack is accomplished.



GENUINE URL

a	p	p	l	e	.	c	o	m
U+0061	U+0070	U+0070	U+006C	U+0065	U+002E	U+0063	U+006F	U+006D

UNICODE CHARACTER CODES

FAKE URL

a	p	p	l	e	.	c	o	m
U+0430	U+0440	U+0440	U+006C	U+0435	U+002E	U+0063	U+006F	U+006D
Cyrillic Small Letter A	Cyrillic Small Letter Er	Cyrillic Small Letter Er	Original Latin	Cyrillic Small Letter Ie	Original Latin	Original Latin	Original Latin	Original Latin

UNICODE CHARACTER CODES WITH ORIGINS

(Figure 1, Nate Bossingham)

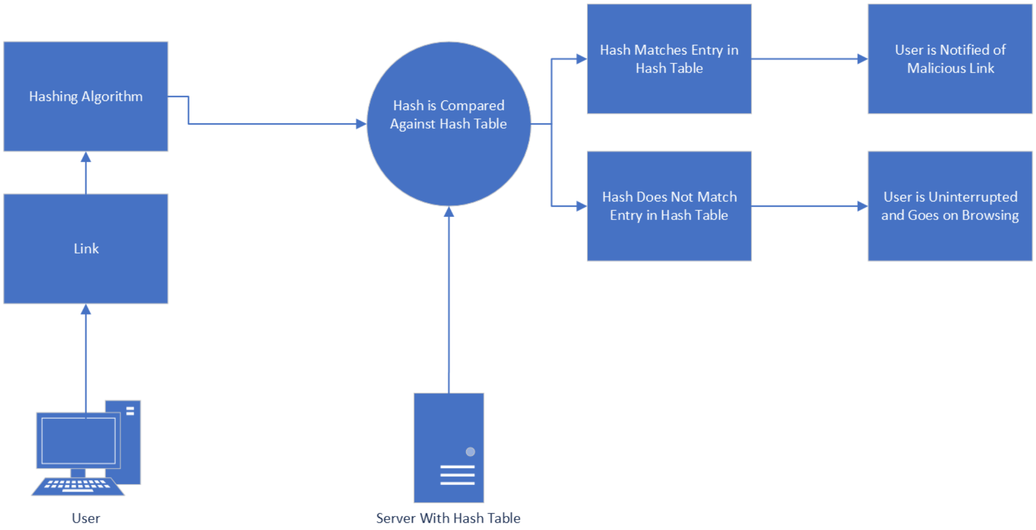
From the figure above, we can see the issue. There are several ways to address this issue and one of the best ways to do so is to set a global font that only renders characters within a set bucket of characters. This method is already utilized by many browsers and poses a few issues for international domains as many of them completely disable the feature. This leads us to the introduction of the solution we aim to develop.

Our solution to this problem is simple. We aim to develop a browser extension for all chromium-based browsers that combats these types of attacks. Along with preventing this attack, we aim to expand our browser extension to combat other modern attack vectors based on Unicode such as the Right-to-Left Override attack (Mitre, 2020). This is an attack that uses a special Unicode character to disguise **.exe** files with less suspicious file extensions such as **.txt**.

- The browser extension will scan all onscreen content and flag any links with uncommon Unicode characters.
- The browser extension will have a database of malicious links curated by our team based off of public archives and user submissions.
- The browser extension will utilize this database to block out any known malicious links.
- The browser extension will scan all text for Unicode characters such as the Right-to-Left Character and flag that content.

In developing this extension, our goal is to not impede on the use cases of a user. These Unicode characters exist to serve a genuine purpose and limiting their appearance can be detrimental to a user experience. This is why our product aims to not block these types of text outright, but to notify and flag a user whenever they appear onscreen. With our goal of not impeding on the user, we also aim to make this extension as unobtrusive as possible with different levels of notification. For those that feel confident in their ability to spot these attacks, our extension can simply add a small warning in the header of the browser notifying the user that foreign characters exist on the current page. For the opposite end of the spectrum, our extension can insert warnings on the page next to any suspicious characters as well as an alert at the header of the browser letting the user know that there is potentially unsafe links on the page. In the efforts of being as unobtrusive as possible, the extension can also handle whitelisting for any trusted domains that the user wishes. This will allow users to browse trusted websites peacefully without any obtrusions.

For the malicious link detection service, our extension will locally scan every page for any malicious links. In the effort to provide users with a secure solution without any worry of data being gathered or stolen, the extension will first search for links locally and hash those links on the machine. Those hashes will then be compared against a cloud-based hash table that contains all known malicious links hashed using the same algorithm. If the hashes match, the system is flagged for that specific link and the user is then notified by the extension that the link is a known malicious link. A diagram of this system can be found in Figure 2.


Hashing Algorithm
Link
User
Hash is Compared Against Hash Table
Server With Hash Table
Hash Matches Entry in Hash Table
Hash Does Not Match Entry in Hash Table
User is Notified of Malicious Link
User is Uninterrupted and Goes on Browsing

(Figure 2, Nate Bossingham)

This product will meet a few barriers in development. The Unicode detection system should be straightforward and can be sorted out quickly as this all happens locally on the machine. The primary barrier will be in developing the malicious link detection system. With the commitment to security and a stance against data harvesting, the encrypted system will take time to develop. The product is possible, but the infrastructure required will have a high initial cost in development time and server space. There are out of the box solutions that already exist for handling all link detections. The primary barrier this project will face is the initial investment as the development of the product is straightforward and the end product is clear in sight.

The success that this product will experience should be large. The primary target of this extension will be the business and professional sectors. By targeting these sectors, our extension can secure customers based on a subscription business model that provides extra protections. These protections will be especially enticing to these sectors due to strict government regulations regarding data leaks, and the high financial risks that come with exposure to cyber threats. According to IBM, the average cost of a data breach globally is around 4.35 million USD (IBM, 2022). This type of financial risk will incentivize customers to employ the browser extension as it will be marketed as a low-cost common-sense safeguard. This product will aim to make the internet an overall safer place to browse and, in that endeavor, will be free to all personal users. With this, our product will be enabled to gain popularity fast and increase the number of enterprise customers.

References

IBM. (2022). Cost of a data breach 2022. IBM. https://www.ibm.com/reports/data-breach

Mitre. (2020). Masquerading: Right-to-left override. Masquerading: Right-to-Left Override, Sub-technique T1036.002 - Enterprise | MITRE ATT&amp;CK®. https://attack.mitre.org/techniques/T1036/002/

Umawing, J. (2017, October 6). Out of character: Homograph attacks explained: Malwarebytes labs. Malwarebytes. https://www.malwarebytes.com/blog/news/2017/10/out-of-character-homograph-attacks-explained