The 2017 homograph browser attack mitigation survey

Tyson McElroy
Edith Cowan University

Peter Hannay
Edith Cowan University

Greg Baatard
Edith Cowan University

DOI: 10.4225/75/5a84f5a495b4d
McElroy, T., Hannay, P., & Baatard, G. (2017). The 2017 homograph browser attack mitigation survey. In Valli, C. (Ed.). (2017). The Proceedings of 15th Australian Information Security Management Conference, 5-6 December, 2017, Edith Cowan University, Perth, Western Australia. (pp.88-96).

THE 2017 HOMOGRAPH BROWSER ATTACK MITIGATION SURVEY

Tyson McElroy[1,2], Peter Hannay[1,2], Greg Baatard[2]
[1]Security Research Institute, [2] School of Science, Edith Cowan University, Perth, Western Australia
tjmcelro@our.ecu.edu.au, p.hannay@ecu.edu.au, g.baatard@ecu.edu.au

Abstract

Since their inception, International Domain Names (IDN) have allowed for non-Latin characters to be entered into domain names. This feature has led to attackers forging malicious domains which appear identical to the Latin counterpart. This is achieved through using non-Latin characters which appear identical to their Latin counterpart. This attack is referred to as a Homograph attack. This research continues the work of Hannay and Bolan (2009), and Hannay and Baatard (2012), which assessed the mitigation methods incorporated by web browsers in mitigating IDN homograph attacks. Since these works, time IDN mitigation algorithms have been altered, such as the one used in Mozilla Firefox (Gerv, 2017). This study evaluates browser homograph attack mitigation strategies in browsers released post-2011. In this study, we find a high level of effective multi-script mitigation across the browser families surveyed. Notable exceptions to this include a single version of Firefox in which the mitigation features were not present and ongoing omission of mitigation against single script attacks.

Keywords: IDN, homograph, homoglyph, internationalised domain names, browser security, phishing

BACKGROUND

Domain name resolution is a technology which allows for IP addresses to be encoded as a string of characters. In the early days of Domain Names, the technology only accepted a string of alphanumeric ASCII characters as input for domain names (Mockapetris, 1987). This limitation prevented international users from accessing certain domains in their respective languages. The introduction of IDNs allowed for specific domain names to be accessible to multiple languages through encoding domain names in the Unicode format ("Introduction to IDNs," 2016).

The proposed IDN solution made use of UTF-8 character encoding to allow for non-latin characters to be displayed. In order to enable existing DNS infrastructure to handle UTF-8 domains a system known as Punycode was developed. Punycode provides facility to represent IDNs as regular ASCII domain names, as such no changes are required for the majority of infrastructure (Costello, 2003). An example of an IDN would be the domain name ☃.com, which would be represented as xn--n3h.com when converted to punycode.

IDN's have allowed for domains to be accessible from many global users. However, the use of Unicode characters has also allowed for phishing attacks to be possible against domain names (Spaulding, Upadhyaya, & Mohaisen, 2016). The use of non-Latin characters in the domain names allowed users to enter Unicode characters which appeared identical to the Latin counterpart, an example of this concept is shown in Figure 1 (Krammer, 2006). As a method to counteract these threats, Domains adopted a system known as Punycode to translate any Unicode characters into their ASCII representation ("Introduction to IDNs," 2016). Many browsers have adopted this technology, allowing users to see when a non-ASCII character is entered into a domain name.

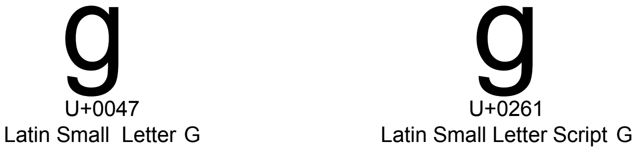

Figure 1 - Example of Homoglyph for "g"

IDN's have continued to see considerable growth and development in recent years ("Key Numbers," 2016), reported that from 2010 to 2015, the total number of registered IDN's had doubled to 6.8 million. Despite browsers incorporating mitigation actions against IDN phishing attacks, they still pose a significant threat. In 2017 browsers such as Firefox, Google, Chrome and Opera were found to be vulnerable to a Punycode exploit (Kumar, 2017). This attack was possible due to a domain registration exploit, which allows the user to register a domain name in

Punycode format with foreign ASCII characters which appear identical to legitimate domains (Tseng, Ku, Lu, Wang, & Geng, 2013).

In the Wild Phishing Attacks using IDN Homographs

There have been numerous websites aiming to educate the public about the dangers of IDN homograph attacks, including epic.com, apple.com, and google.com (Hannay, 2012; Maunder, 2017; Zheng, 2017). However, the occurrence of homograph usage in confirmed phishing campaigns has been relatively minor with only a single confirmed major incident. In August the domain adoḅe.com was registered, subsequently the domain was used to distribute the Beta Bot malware, disguised as an update to the Adobe Flash Player software (Mimoso, 2017). Links to the website were distributed via email and Skype messages, requesting that the user install the fabricated update. Post infection the malware disables security software, then steals financial data and user credentials (Kaspersky Lab, 2017).

PREVIOUS WORK

Since the introduction of IDN's, many attacks have been possible due to the use of non-ASCII characters. The following literature review gives a brief overview of the types of attacks due to the introduction of IDNs. Table 1 provides an overview of of the types of attacks, covering single script, mixed-script, and whole script spoofing attacks which can take place through character substitution.

Table 1 – Examples of Single Script, Mixed Script, and Whole Script Spoofing

Script Type	String	Punycode	Comments
Single Script	EPlC.com	EPlC.com	Lowercase L used as replacement for uppercase I
Mixed Script	epic.com	xn--pic-qdd.com	Cryllic replacement used for 'e'
Whole Script	epic.com	xn--e1awd7f.com	Cryllic replacements used for 'epic'

Mixed-Script Spoofing

One of the most prevalent attacks which have been proven possible by IDN domains is mixed-script spoofing. Mixed-script spoofing generates domain names using visually indistinguishable characters form different script groups (Krammer, 2006). These characters appear almost the same to the end user but contain different Unicode values. These characters exist within Unicode due to the writing system being used, letter and number encodings, and legacy encoding values (Davis & Suignard, 2006). These visually indistinguishable characters are known as Homoglyphs. Due to the indistinguishable nature of certain Unicode characters, attackers can easily forge domain names which appear to be visually indistinguishable from other legitimate domain names. These domain names are referred to as holographs, as they are comprised of various characters from separate scripts. Homographs can be used to trick users into going to a malicious domain and as such have been used in various phishing schemes.

Whole-Script Spoofing & Single-Script Spoofing

Whole script spoofing and single script spoofing differ from the mixed-script spoofing approach. Due to the introduction of IDNs, entire domain names can be spoofed through substituting each character in a domain with one of a different script (Krammer, 2006). This attack relies on each character in a domain name having an indistinguishable counterpart in another script. Attackers can utilise this coincidence to generate a fully indistinguishable domain name. Another attack possible is single script spoofing. This attack uses characters from the same script to visually trick users into going to the domain (Gelernter & Herzberg, 2016). These attacks are more recognisable to end users, as attackers are not substituting Unicode character from different scripts (Krammer, 2006). Instead, domain names are constructed using characters which appear somewhat identical to their counterparts. An example of this is Latin 'o' and '0', which can be used to forge the domain 'www.g00gle.com'.

DEFENCE TECHNIQUES

Various defence mechanisms were adopted to address the security issues arising from the introduction of IDNs. The most prevalent defence against homograph attacks is displaying Unicode characters in a Punycode format. Punycode is used to encode a Unicode string into its appropriate ASCII string representation (Costello, 2003). Punycode values are prefixed with xn-- to represent the Unicode string. Values which contain ASCII characters are interpreted as literal strings; however Unicode characters are transformed into their ASCII interpretation (Costello, 2003).

Web Browser Defence

As a means of defence against homograph threats, web browsers have begun implementing defence measures to notify users of potential issues with the domain name. One mitigation technique employed by Internet Explorer 7 is to display Punycode when mixed-script characters are detected in the domain name (Al Helou & Tilley, 2010). Another defensive technique which browsers incorporate is colour coding particular scripts. This technique shows characters in different colours based on the script to which they belong.

Another technique incorporated by Mozilla and Safari is to use a whitelisting approach. This security measure displays all IDNs in Punycode unless the domain is registered under a Top Level Domain (TLD) which has policies in place to prevent spoofing of the domain. As a requirement for this registering the domain with Homoglyphs, the owner must already own the western equivalent of the domain name. Mozilla Firefox still retains this whitelisting approach for handling IDNs but has since updated how the Punycode display works in 2012 (Gerv, 2017). The new algorithm employed determines if the entered domain name contains characters belonging the same script or if the characters are being pulled from one of the allowed predefined script combinations (Gerv, 2017). If the entered domain name is not within the pre-established whitelist for TLDs or if the domain name is using characters from illegal script combinations, a Punycode sample is displayed to the user. The previous whitelisting approach only remains for compatibility purposes with the domains registered with it but is no longer the primary method used for Homograph mitigation (Gerv, 2017). Some browsers such as Opera still retain the previous whitelisting approach which Firefox has since abandoned, but remains active for compatibility purposes with the registered domains.

In the Hannay and Baatard (2012) survey, various browsers were assessed to determine their effectiveness in mitigating Homograph attacks. In this previous study, numerous versions of the web browsers Mozilla Firefox, Internet Explorer, Google Chrome, Opera, and Safari were assessed regarding the mitigation techniques used against homograph attacks. The results from the previous study demonstrated that later versions certain browsers such as Google Chrome and Mozilla Firefox were highly effective in mitigating homograph attacks, while the latest versions of Internet Explorer and Safari were still vulnerable to some attacks at the time of the study.

In recent versions of browsers such as Firefox, the algorithm used to display Punycode has been altered. This modification indicates that later versions may not have the same homograph mitigation methods applied. This research updates the results seen in the Hannay and Baatard (2012) study by analysing the mitigation techniques adopted by browser versions released between 2011 and 2017. Through performing this investigation, the author answers the question of if mitigation techniques have been applied to browser versions post-2011 and if changes to the mitigation functions resulted in further vulnerabilities with certain versions.

RESEARCH METHOD

The testbed used for this investigation consists of a virtual machine running Windows 7 and an Ubuntu 14.04 Desktop virtual machine hosting sites containing single and mixed-script domain name. Various versions of the web browsers Mozilla Firefox, Chromium, Internet Explorer, and Opera were installed on the Windows 7 virtual machine. The versions which are tested consist of various versions from 2011 to 2017, which were not covered in the 2012 study. Due to issues obtaining previous versions of Google Chrome, the Chromium browser is used for this research. Given the total quantity of browser versions released per year, the versions covered in this research consist of those releases mid-year and at the end of the year. As a means of managing the browser versions installed on the tested, a snapshot is taken before any browser is installed on the environment. After each browser version is tested, the virtual environment is rolled back to the base install to prepare for the next iteration of testing.

Table 2: Test Domain Names

Domain Name Character Set	Domain Name
Single-Script	n0tasecuresite.com
Mixed-Script	notasecuresite.com

To perform this test in a controlled environment, two distinct websites were created, these were hosted on an Apache web server. The domain names of these sites were configured to use single-script and mixed-script characters. The domain names used in the test are shown in Table 2. The single-script domain uses the Latin character ‘0’ in place of an ‘o’, while the mixed-script site uses the U1086 Cryllic ‘o’ character. These sites are also using self-signed Secure Socket Layer (SSL) certificates which also correspond to the single-script and mixed-script characters used in the domain names. Finally, both the sites are configured to use Geolocation services which prompt the user to share their current location with the given domain. Through hosting websites using single-script and mixed-script character sets, It is possible to assess the mitigation techniques applied to the two domains securely.

To test the effectiveness of the mitigation methods applied to each browser version, four common attack vectors were identified corresponding to browser locations where the output is either the standard Unicode format or Punycode. This mitigation tactic is used to convey information to the user, regarding if any non-standard ASCII characters are detected. Through assessing various browser features, the research demonstrates how effective each browser is in conveying this information to the end user. These attack vectors used in this investigation are:

- The text shown in the browser’s address bar, after the “Go” (or equivalent) button has been pressed.
- The text shown in the browser’s status bar while the mouse is hovering over a hyperlink.
- The text shown when viewing prominent information about the website’s SSL certificate.
- The text shown when the user is prompted to share their location using geolocation services.

As a means to assess how effective each browser version is with mitigating homograph attacks, a revised mitigation rating table similar to those presented in the 2012 study is provided. A value of zero was assigned should the browser not support a particular attack vector, for example, geolocation services. A value of negative one is given if a browser has implemented an attack vector, but is still open to homograph attacks. A value of positive one is given if a browser has implemented an attack vector and mitigated homograph attacks. Should the browser support or not support mitigation methods when displaying text in the browsers address bar, a value of positive two or negative two is given for this value. This value is due to this vector being the most prominent location for displaying Punycode. The browsers will receive two distinct ratings for single script and mixed-script mitigation. An example of the table used can be viewed in Table 3.

Table 3: Mitigation Table Structure

Address Bar	Status Bar	SSL Certificate	Location Request
-2 (No mitigation)	-1 (No mitigation)	-1 (No mitigation)	-1 (No mitigation)
0 (No Support)	0 (No Support)	0 (No Support)	0 (No Support)
+2 (Mitigated)	+1 (Mitigated)	+1 (Mitigated)	+1 (Mitigated)

RESULTS

The post-2011 versions of Internet Explorer, as shown in Table 4, demonstrated improvements to mitigating Homograph attacks. Following the introduction of geolocation services in version 9, version 10 incorporated a Punycode mitigation method for mixed-script attacks. Internet Explorer has yet to implement a method of displaying Punycode for the SSL certificate, therefore not all mitigation methods have been implemented in the browser as of yet. Internet Explorer provides no method of mitigating against single script attacks, as no notification was given when supplementing a Latin ‘o’ and a ‘0’.

As with Internet Explorer, single-script mitigation techniques are not present in any version of Firefox shown in Table 5. Version 17 of Firefox was not capable of displaying any form of Punycode in the address bar, SSL certificate, or geolocation request. This finding was likely a bug with versions from that time span, as this issue was later fixed. The only other issue with later Firefox versions was a lack of geolocation support present in version 26.

These results of Table 6 show that the Opera browser is highly effective in detecting mixed-script IDNs and implementing appropriate mitigation methods to notify the users. As with the other browsers, no support for mitigating single script spoofing has been implemented yet.

Table 4: Internet Explorer Mitigation Table

Version & Release Date	Address Bar Mitigation	Status Bar Mitigation	SSL Certificate Mitigation	Location Request Mitigation	Mitigation Rating
10 (2012 - 09) **11 (2013 - 10)**	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-Script – Punycode	Mixed-Script – Punycode	Mixed-Script – No Mitigation	Mixed-Script – Punycode	+3

Table 5: Firefox Mitigation Table

Version & Release Date	Address Bar Mitigation	Status Bar Mitigation	SSL Certificate Mitigation	Location Request Mitigation	Mitigation Rating
13 (2012 – 06)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	+5
17 (2012 – 11)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-script – No Mitigation	Mixed-script – Punycode	Mixed-script – No Mitigation	Mixed-script – No Mitigation	-3
22 (2013 – 06)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-Script – Punycode	Mixed-Script – Punycode	Mixed-Script – Punycode	Mixed-Script – Punycode	+5
26 (2013 – 12)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	**No support***	-4
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	**No Support***	+4
30 (2014 – 06) **34 (2014 – 12)** **38 (2015 – 05)** **43 (2015 – 12)** **47 (2016 – 06)** **50 (2016 – 11)** **55 (2017 – 08)**	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-Script – Punycode	Mixed-Script – Punycode	Mixed-Script – Punycode	Mixed-Script – Punycode	+5

Table 6: Opera Mitigation Table

Version & Release Date	Address Bar Mitigation	Status Bar Mitigation	SSL Certificate Mitigation	Location Request Mitigation	Mitigation Rating
12 (2012 – 06)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	+5
15 (2013 – 07)	Single Script – No Mitigation Mixed-script – Punycode	Single Script – No Mitigation Mixed-script – Punycode	Single Script – No Mitigation Mixed-script – Punycode	No Support	-4
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	No Support	+4
18 (2013 – 11) 22 (2014 – 06) 26 (2014 – 12) 30 (2015 – 06) 34 (2015 – 12) 38 (2016 – 06) 42 (2016 – 12) 47 (2017 – 08)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	+5

Table 7: Chromium Mitigation Table

Version & Release Date	Address Bar Mitigation	Status Bar Mitigation	SSL Certificate Mitigation	Location Request Mitigation	Mitigation Rating
20.0.1123 (2012 – 05) 25.0.1323.1 (2012 – 11) 29.0.1541.0 (2013 – 06) 32.0.1700.6 (2013 – 11) 37.0.2017.2 (2014 – 05) 41.0.2243.0 (2014 – 12) 45.0.2431.0 (2015 – 06) 49.0.2593.0 (2015 – 12) 53.0.2763.0 (2016 – 05)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Mitigation	-5
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – Punycode	+5
56.0.2902.0 (2016 – 10)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Support	Single Script – No Mitigation	-4
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – No Support	Mixed-script – Punycode	+4
61.0.3153.0 (2017 – 09)	Single Script – No Mitigation	Single Script – No Mitigation	Single Script – No Support	Single Script – No Mitigation	-4
	Mixed-script – Punycode	Mixed-script – Punycode	Mixed-script – No Support	Mixed-script – Punycode	+4

The results from Table 7 show that Chromium is highly effective in mitigating against IDN homograph attacks. Like the other browsers analysed, no support for single-script mitigation has been added to any version. Chromium was shown to be highly consistent with the mitigation functions applied to each browser version. However, the most recent version of the browser does not accurately convey SSL certificate information to the user. This limitation results in the browser being unable to display Punycode for the domain name used in the SSL certificate.

The results from single-script mitigation are demonstrated in Figure 1. Given the lack of mitigation functions, each browser received a negative mitigation rating. The most consistent rating across all browsers was a -5. The versions which achieved a -4 rating were as a result of a lack of support for a given for that version.

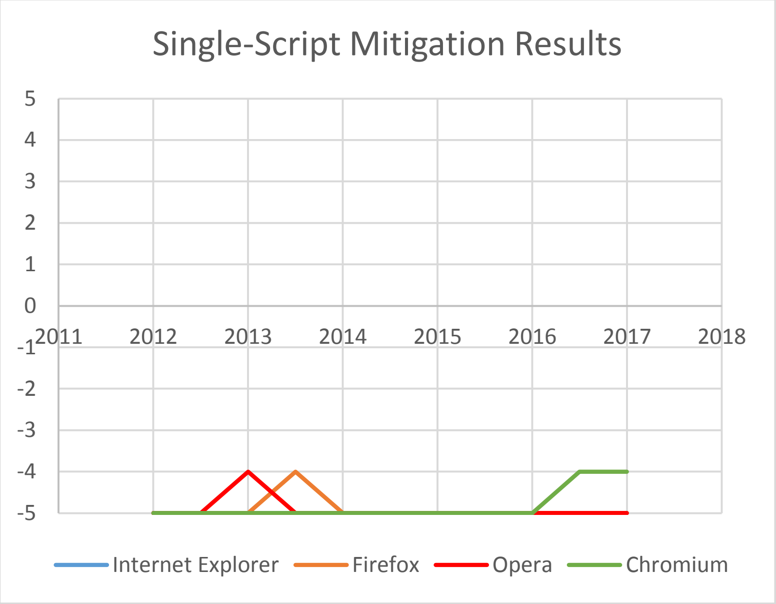

Figure 2 - Single-Script Mitigation Results

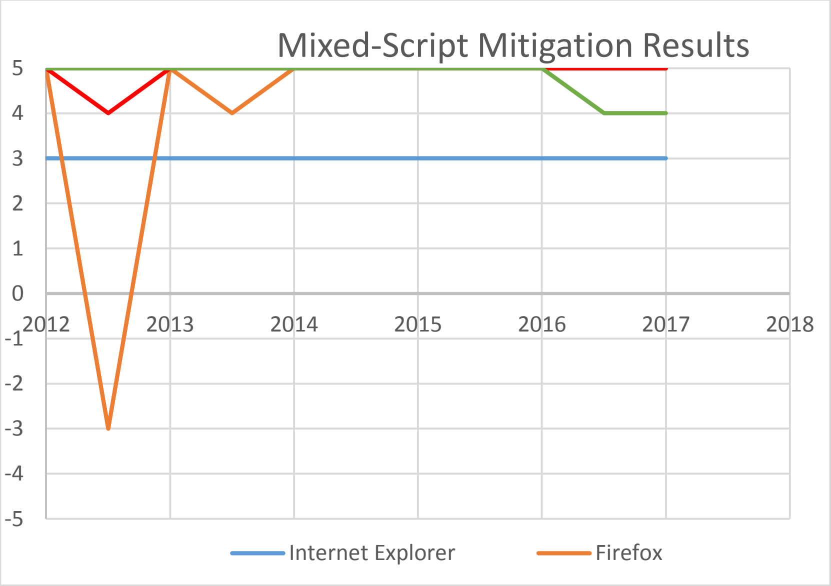

Figure 3- Mixed-Script Mitigation Results

The results from mixed-script mitigation are demonstrated in Figure 2. The majority of browser versions demonstrate consistent mitigation ratings across different releases. The drops in mitigation ratings appear to be from certain versions not supporting for a particular feature. However, the 2012 version of Firefox is shown to have the greatest drop in mitigation rating. This finding is a result of the version only displaying Punycode for the hover feature of Hyperlinks. The issues discovered in this version were later amended in later versions, with the only other drop being the lack of geolocation support in the 2013 version. Internet Explorer has demonstrated a greater mitigation rating in later versions by implementing geolocation services. Opera was shown to be the most effective in implementing mitigation features for IDN homograph attacks, as it received the rating of +5 more consistently than the other browsers tested.

The uptake rate of new browser versions plays a significant role in determining the potential exposure of any particular browser version to hostile actors. A study conducted by Ion, Reeder, and Consolvo (2015) compared the cyber security practices of expert and non-expert users. One major finding of the study was that largest point of difference between the groups, was the importance placed on installing software updates in a timely fashion (Ion et al., 2015). Examining W3Counter data on browser usage shows us that as of August 2017, that 25.84% of users were running web browsers more than three months old (W3Counter, 2017). As such we can see that the potential exposure for vulnerabilities in specific browser versions, such as those seen in Firefox in 2012, may span many months from release of the software.

CONCLUSION

The results discovered in this research are representative of how IDN mitigation techniques have been implemented in various browsers in recent years. In extending the results of the previous study by Hannay and Baatard (2012), the results found in this study appear to indicate that mixed-script homograph attack mitigation has become a standard feature for most browsers. The majority of browsers analysed in this study demonstrated to implement appropriate mitigation techniques for IDN homograph attacks. The only degree of variance in the mixed-script results appears to be with different versions of particular browsers, which could be a result of bugs or changes in the algorithm. The results for single-script mitigation demonstrate that this is not a feature commonly implemented in browsers. Of the browsers analysed, none provided any support to notify the end user about a non-standard character being used in the domain name. This research suggests that mitigation against mixed-script homograph attacks has become a common feature for most browsers, while implementation of mitigation functionality for single-script spoofing attacks has not been undertaken.

REFERENCES

Al Helou, J., & Tilley, S. (2010). *Multilingual web sites: Internationalized Domain Name homograph attacks.* Paper presented at the Web Systems Evolution (WSE), 2010 12th IEEE International Symposium on.

Costello, A. M. (2003). Punycode: A bootstring encoding of unicode for internationalized domain names in applications (IDNA).

Davis, M., & Suignard, M. (2006). Unicode security considerations: Citeseer.

Gelernter, N., & Herzberg, A. (2016). Autocomplete Injection Attack. In I. Askoxylakis, S. Ioannidis, S. Katsikas, & C. Meadows (Eds.), *Computer Security – ESORICS 2016: 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II* (pp. 512-530). Cham: Springer International Publishing.

Gerv. (2017, April 17th). IDN Display Algorithm. Retrieved from https://wiki.mozilla.org/IDN_Display_Algorithm

Hannay, P. (2012). Google Awesome Edition. Retrieved from http://xn--goole-tmc.com/

Hannay, P., & Baatard, G. (2012). *The 2011 IDN homograph attack mitigation survey.* Paper presented at the Proceedings of the International Conference on Security and Management (SAM).

Hannay, P., & Bolan, C. (2009). *Assessment of Internationalised Domain Name Homograph Attack Mitigation.* Paper presented at the Australian Information Security Management Conference.

Introduction to IDNs. (2016, July 3rd). Retrieved from http://idnworldreport.eu/introduction-to-idns/

Ion, I., Reeder, R., & Consolvo, S. (2015). *"... No one Can Hack My Mind": Comparing Expert and Non-Expert Security Practices.* Paper presented at the SOUPS.

Kaspersky Lab. (2017). What is Beta Bot? Retrieved from https://usa.kaspersky.com/resource-center/definitions/beta-bot

Key Numbers. (2016, August 16th). Retrieved from http://idnworldreport.eu/facts-figures/number-of-idns-2/

Krammer, V. (2006). *Phishing defense against IDN address spoofing attacks.* Paper presented at the Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services.

Kumar, M. (2017, April 17th). This Phishing Attack is Almost Impossible to Detect On Chrome, Firefox and Opera. Retrieved. Retrieved from http://thehackernews.com/2017/04/unicode-Punycode-phishing-attack.html

Maunder, M. (2017, April 14th). Chrome and Firefox Phishing Attack Uses Domains Identical to Known Safe Sites. Retrieved from https://www.xn--e1awd7f.com/ & https://www.wordfence.com/blog/2017/04/chrome-firefox-unicode-phishing/

Mimoso, M. (2017, September 6th). IDN Homograph Attack Spreading Betabot Backdoor. Retrieved from https://threatpost.com/idn-homograph-attack-spreading-betabot-backdoor/127839/

Mockapetris, P. V. (1987). Domain Names-Concepts and Facilities *RFC1034*: IETF.

Spaulding, J., Upadhyaya, S., & Mohaisen, A. (2016). *The landscape of domain name typosquatting: Techniques and countermeasures.* Paper presented at the Availability, Reliability and Security (ARES), 2016 11th International Conference on.

Tseng, S.-S., Ku, C.-H., Lu, A.-C., Wang, Y.-J., & Geng, G.-G. (2013). *Building a Self-Organizing Phishing Model Based upon Dynamic EMCUD.* Paper presented at the Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on.

W3Counter. (2017). Browser & Platform Market Share - August 2017. Retrieved from https://www.w3counter.com/globalstats.php?year=2017&month=8

Zheng, X. (2017). IDN Homograph Example. Retrieved from http://xn--80ak6aa92e.com/