IDN homograph attack

From Wikipedia, the free encyclopedia

[Jump to navigation](https://en.wikipedia.org/wiki/IDN_homograph_attack#mw-head)[Jump to search](https://en.wikipedia.org/wiki/IDN_homograph_attack#searchInput)

|  |  |
| --- | --- |
|  | **This article contains**[**special characters**](https://en.wikipedia.org/wiki/Help:Special_characters)**.** Without proper [rendering support](https://en.wikipedia.org/wiki/Help:Special_characters), you may see [question marks, boxes, or other symbols](https://en.wikipedia.org/wiki/Specials_(Unicode_block)#Replacement_character). |

[An example of an IDN homograph attack; the Latin letters "e" and "a" are replaced with the Cyrillic letters "е" and "а".](https://en.wikipedia.org/wiki/File:IDN_homograph_attack_1.svg)

An example of an IDN homograph attack; the Latin letters "e" and "a" are replaced with the Cyrillic letters "е" and "а".

The [**internationalized domain name**](https://en.wikipedia.org/wiki/Internationalized_domain_name) (**IDN**) **homograph attack** is a way a malicious party may deceive computer users about what remote system they are communicating with, by exploiting the fact that many different [characters](https://en.wikipedia.org/wiki/Grapheme) look alike (i.e., they are homographs, hence the term for the attack, although technically [homoglyph](https://en.wikipedia.org/wiki/Homoglyph) is the more accurate term for different characters that look alike). For example, a regular user of [example.com](https://en.wikipedia.org/wiki/Example.com) may be lured to click a link where the [Latin](https://en.wikipedia.org/wiki/Latin_alphabet) character ["a"](https://en.wikipedia.org/wiki/A) is replaced with the [Cyrillic](https://en.wikipedia.org/wiki/Cyrillic_script) character ["а"](https://en.wikipedia.org/wiki/A_(Cyrillic)).

This kind of [spoofing attack](https://en.wikipedia.org/wiki/Spoofing_attack) is also known as **script spoofing**. [Unicode](https://en.wikipedia.org/wiki/Unicode) incorporates numerous writing systems, and, for a number of reasons, similar-looking characters such as Greek [Ο](https://en.wikipedia.org/wiki/Omicron), Latin [O](https://en.wikipedia.org/wiki/O), and Cyrillic [О](https://en.wikipedia.org/wiki/O_(Cyrillic)) were not assigned the same code. Their incorrect or malicious usage is a possibility for security attacks.[[1]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-tr36-1)

The registration of homographic domain names is akin to [typosquatting](https://en.wikipedia.org/wiki/Typosquatting), in that both forms of attacks use a similar-looking name to a more established domain to fool a user. The major difference is that in typosquatting the perpetrator attracts victims by relying on natural typographical errors commonly made when manually entering a [URL](https://en.wikipedia.org/wiki/URL), while in homograph spoofing the perpetrator deceives the victims by presenting visually indistinguishable [hyperlinks](https://en.wikipedia.org/wiki/Hyperlink). Indeed, it would be a rare accident for a web user to type, for example, a Cyrillic letter within an otherwise English word such as "citibаnk". There are cases in which a registration can be both typosquatting and homograph spoofing; the pairs of l/I, i/j, and 0/O are all both close together on keyboards and, depending on the typeface, may be difficult or impossible to distinguish.



**Contents**

* [1History](https://en.wikipedia.org/wiki/IDN_homograph_attack#History)
* [2Homographs in ASCII](https://en.wikipedia.org/wiki/IDN_homograph_attack#Homographs_in_ASCII)
* [3Homographs in internationalized domain names](https://en.wikipedia.org/wiki/IDN_homograph_attack#Homographs_in_internationalized_domain_names)
  + [3.1Cyrillic](https://en.wikipedia.org/wiki/IDN_homograph_attack#Cyrillic)
  + [3.2Greek](https://en.wikipedia.org/wiki/IDN_homograph_attack#Greek)
  + [3.3Armenian](https://en.wikipedia.org/wiki/IDN_homograph_attack#Armenian)
  + [3.4Hebrew](https://en.wikipedia.org/wiki/IDN_homograph_attack#Hebrew)
  + [3.5Chinese](https://en.wikipedia.org/wiki/IDN_homograph_attack#Chinese)
  + [3.6Other scripts](https://en.wikipedia.org/wiki/IDN_homograph_attack#Other_scripts)
  + [3.7Accented characters](https://en.wikipedia.org/wiki/IDN_homograph_attack#Accented_characters)
  + [3.8Non-displayable characters](https://en.wikipedia.org/wiki/IDN_homograph_attack#Non-displayable_characters)
  + [3.9Known homograph attacks](https://en.wikipedia.org/wiki/IDN_homograph_attack#Known_homograph_attacks)
* [4Defending against the attack](https://en.wikipedia.org/wiki/IDN_homograph_attack#Defending_against_the_attack)
  + [4.1Client-side mitigation](https://en.wikipedia.org/wiki/IDN_homograph_attack#Client-side_mitigation)
  + [4.2Server-side/registry operator mitigation](https://en.wikipedia.org/wiki/IDN_homograph_attack#Server-side/registry_operator_mitigation)
* [5See also](https://en.wikipedia.org/wiki/IDN_homograph_attack#See_also)
* [6References](https://en.wikipedia.org/wiki/IDN_homograph_attack#References)
* [7External links](https://en.wikipedia.org/wiki/IDN_homograph_attack#External_links)

History[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=1)]

An early nuisance of this kind, pre-dating the Internet and even [text terminals](https://en.wikipedia.org/wiki/Text_terminal), was the confusion between "l" (lowercase letter "L") / "1" (the number "one") and "O" (capital letter for vowel "o") / "0" (the number "zero"). Some [typewriters](https://en.wikipedia.org/wiki/Typewriter) in the pre-computer era even combined the [L and the one](https://en.wikipedia.org/wiki/Typewriter#Typewriter_conventions); users had to type a lowercase L when the number one was needed. The zero/o confusion gave rise to the tradition of [crossing zeros](https://en.wikipedia.org/wiki/Slashed_zero), so that a [computer operator](https://en.wikipedia.org/wiki/Computer_operator) would type them correctly.[[2]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-tha-2) Unicode may contribute to this greatly with its combining characters, accents, several types of [hyphen](https://en.wikipedia.org/wiki/Hyphen), etc., often due to inadequate [rendering](https://en.wikipedia.org/wiki/Rendering_(computer_graphics)) support, especially with smaller font sizes and the wide variety of fonts.[[1]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-tr36-1)

Even earlier, [handwriting](https://en.wikipedia.org/wiki/Handwriting) provided rich opportunities for confusion. A notable example is the etymology of the word "[zenith](https://en.wikipedia.org/wiki/Zenith)". The translation from the Arabic "samt" included the scribe's confusing of "m" into "ni". This was common in medieval [blackletter](https://en.wikipedia.org/wiki/Blackletter), which did not connect the vertical columns on the letters i, m, n, or u, making them difficult to distinguish when several were in a row. The latter, as well as "rn"/"m"/"rri" ("RN"/"M"/"RRI") confusion, is still possible for a human eye even with modern advanced computer technology.

Intentional look-alike character substitution with different alphabets has also been known in various contexts. For example, [Faux Cyrillic](https://en.wikipedia.org/wiki/Faux_Cyrillic) has been used as an amusement or attention-grabber and "[Volapuk encoding](https://en.wikipedia.org/wiki/Volapuk_encoding)", in which Cyrillic script is represented by similar Latin characters, was used in early days of the [Internet](https://en.wikipedia.org/wiki/Internet) as a way to overcome the lack of support for the Cyrillic alphabet. Another example is that [vehicle registration plates](https://en.wikipedia.org/wiki/Vehicle_registration_plate#Europe) can have both Cyrillic (for domestic usage in Cyrillic script countries) and Latin (for international driving) with the same letters. Registration plates that are issued in [Greece](https://en.wikipedia.org/wiki/Greece) are limited to using letters of the [Greek alphabet](https://en.wikipedia.org/wiki/Greek_alphabet) that have homoglyphs in the Latin alphabet, as [European Union](https://en.wikipedia.org/wiki/European_Union) regulations require the use of Latin letters.

Homographs in ASCII[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=2)]

ASCII has several characters or pairs of characters that look alike and are known as *homographs* (or [*homoglyphs*](https://en.wikipedia.org/wiki/Homoglyph)). [Spoofing attacks](https://en.wikipedia.org/wiki/Spoofing_attack) based on these similarities are known as **homograph spoofing attacks**. For example, 0 (the number) and O (the letter), "l" lowercase L, and "I" uppercase "i".

In a typical example of a hypothetical attack, someone could register a [domain name](https://en.wikipedia.org/wiki/Domain_name) that appears almost identical to an existing domain but goes somewhere else. For example, the domain "rnicrosoft.com" begins with "r" and "n", not "m". Other examples are *G00GLE.COM* which looks much like *GOOGLE.COM* in some fonts. Using a mix of uppercase and lowercase characters, *googIe.com* (capital *i*, not small *L*) looks much like *google.com* in some fonts. [PayPal](https://en.wikipedia.org/wiki/PayPal) was a target of a phishing scam exploiting this, using the domain [PayPaI.com](https://en.wikipedia.org/wiki/PayPaI). In certain narrow-spaced fonts such as [Tahoma](https://en.wikipedia.org/wiki/Tahoma_(typeface)) (the default in the address bar in [Windows XP](https://en.wikipedia.org/wiki/Windows_XP)), placing a c in front of a j, l or i will produce homoglyphs such as cl cj ci (d g a).

Homographs in internationalized domain names[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=3)]

In [multilingual](https://en.wikipedia.org/wiki/Multilingual) computer systems, different logical characters may have identical appearances. For example, [Unicode](https://en.wikipedia.org/wiki/Unicode) character U+0430, [Cyrillic](https://en.wikipedia.org/wiki/Cyrillic_script) small letter a ("а"), can look identical to Unicode character U+0061, [Latin](https://en.wikipedia.org/wiki/Latin_alphabet) small letter a, ("a") which is the lowercase "a" used in English. Hence wikipediа.org (the Cyrillic version) instead of wikipedia.org (the Latin version).

The problem arises from the different treatment of the characters in the user's mind and the computer's programming. From the viewpoint of the user, a Cyrillic "а" within a Latin string *is* a Latin "a"; there is no difference in the glyphs for these characters in most fonts. However, the computer treats them differently when processing the character string as an identifier. Thus, the user's assumption of a one-to-one correspondence between the visual appearance of a name and the named entity breaks down.

[Internationalized domain names](https://en.wikipedia.org/wiki/Internationalized_domain_name) provide a backward-compatible way for domain names to use the full Unicode character set, and this standard is already widely supported. However this system expanded the character repertoire from a few dozen characters in a single alphabet to many thousands of characters in many scripts; this greatly increased the scope for homograph attacks.

This opens a rich vein of opportunities for [phishing](https://en.wikipedia.org/wiki/Phishing) and other varieties of fraud. An attacker could register a domain name that *looks* just like that of a legitimate website, but in which some of the letters have been replaced by homographs in another alphabet. The attacker could then send e-mail messages purporting to come from the original site, but directing people to the bogus site. The spoof site could then record information such as passwords or account details, while passing traffic through to the real site. The victims may never notice the difference, until suspicious or criminal activity occurs with their accounts.

In December 2001 [Evgeniy Gabrilovich](https://en.wikipedia.org/wiki/Evgeniy_Gabrilovich) and [Alex Gontmakher](https://en.wikipedia.org/w/index.php?title=Alex_Gontmakher&action=edit&redlink=1), both from [Technion](https://en.wikipedia.org/wiki/Technion), [Israel](https://en.wikipedia.org/wiki/Israel), published a paper titled "The Homograph Attack",[[2]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-tha-2) which described an attack that used Unicode URLs to spoof a website URL. To prove the feasibility of this kind of attack, the researchers successfully registered a variant of the domain name [*microsoft*](https://en.wikipedia.org/wiki/Microsoft)*.com* which incorporated Cyrillic characters.

Problems of this kind were anticipated before IDN was introduced, and guidelines were issued to registries to try to avoid or reduce the problem. For example, it was advised that registries only accept characters from the Latin alphabet and that of their own country, not all of Unicode characters, but this advice was neglected by major [TLDs](https://en.wikipedia.org/wiki/TLD).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

On February 7, 2005, [Slashdot](https://en.wikipedia.org/wiki/Slashdot) reported that this exploit was disclosed by 3ric Johanson at the [hacker](https://en.wikipedia.org/wiki/Hacker_(computer_security)) conference [Shmoocon](https://en.wikipedia.org/wiki/Shmoocon).[[3]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-3) Web browsers supporting IDNA appeared to direct the URL http://www.pаypal.com/, in which the first *a* character is replaced by a Cyrillic *а*, to the site of the well known payment site [PayPal](https://en.wikipedia.org/wiki/PayPal), but actually led to a spoofed web site with different content. Popular browsers continued to have problems properly displaying international domain names through April 2017.[[4]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-4)

The following alphabets have characters that can be used for spoofing attacks (please note, these are only the most obvious and common, given artistic license and how much risk the spoofer will take of getting caught; the possibilities are far more numerous than can be listed here):

**Cyrillic**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=4)]

Cyrillic is, by far, the most commonly used alphabet for homoglyphs, largely because it contains 11 lowercase glyphs that are identical or nearly identical to Latin counterparts.

The [Cyrillic letters](https://en.wikipedia.org/wiki/Cyrillic_alphabet) [а](https://en.wikipedia.org/wiki/%D0%90), [с](https://en.wikipedia.org/wiki/%D0%A1), [е](https://en.wikipedia.org/wiki/%D0%95), [о](https://en.wikipedia.org/wiki/%D0%9E), [р](https://en.wikipedia.org/wiki/%D0%A0), [х](https://en.wikipedia.org/wiki/%D0%A5) and [у](https://en.wikipedia.org/wiki/%D0%A3) have optical counterparts in the basic Latin alphabet and look close or identical to [a](https://en.wikipedia.org/wiki/A), [c](https://en.wikipedia.org/wiki/C), [e](https://en.wikipedia.org/wiki/E), [o](https://en.wikipedia.org/wiki/O), [p](https://en.wikipedia.org/wiki/P), [x](https://en.wikipedia.org/wiki/X) and [y](https://en.wikipedia.org/wiki/Y). Cyrillic [З](https://en.wikipedia.org/wiki/%D0%97), [Ч](https://en.wikipedia.org/wiki/%D0%A7) and [б](https://en.wikipedia.org/wiki/%D0%91) resemble the numerals [3](https://en.wikipedia.org/wiki/3_(number)), [4](https://en.wikipedia.org/wiki/4_(number)) and [6](https://en.wikipedia.org/wiki/6_(number)). [Italic type](https://en.wikipedia.org/wiki/Italic_type) generates more homoglyphs: *дтпи* or *дтпи* ([д](https://en.wikipedia.org/wiki/%D0%94)[т](https://en.wikipedia.org/wiki/%D0%A2)[п](https://en.wikipedia.org/wiki/%D0%9F)[и](https://en.wikipedia.org/wiki/%D0%98) in standard type), resembling [d](https://en.wikipedia.org/wiki/D)[m](https://en.wikipedia.org/wiki/M)[n](https://en.wikipedia.org/wiki/N)[u](https://en.wikipedia.org/wiki/U) (in some fonts [д](https://en.wikipedia.org/wiki/%D0%94) can be used, since its italic form resembles a lowercase [g](https://en.wikipedia.org/wiki/G); however, in most mainstream fonts, д instead resembles a [partial differential](https://en.wikipedia.org/wiki/Partial_differential) sign, [∂](https://en.wikipedia.org/wiki/%E2%88%82)).

If capital letters are counted, [А](https://en.wikipedia.org/wiki/%D0%90)[В](https://en.wikipedia.org/wiki/%D0%92)[С](https://en.wikipedia.org/wiki/%D0%A1)[Е](https://en.wikipedia.org/wiki/%D0%95)[Н](https://en.wikipedia.org/wiki/%D0%9D)[І](https://en.wikipedia.org/wiki/%D0%86)[Ј](https://en.wikipedia.org/wiki/%D0%88)[К](https://en.wikipedia.org/wiki/%D0%9A)[М](https://en.wikipedia.org/wiki/%D0%9C)[О](https://en.wikipedia.org/wiki/%D0%9E)[Р](https://en.wikipedia.org/wiki/%D0%A0)[Ѕ](https://en.wikipedia.org/wiki/%D0%85)[Т](https://en.wikipedia.org/wiki/%D0%A2)[Х](https://en.wikipedia.org/wiki/%D0%A5) can substitute [A](https://en.wikipedia.org/wiki/A)[B](https://en.wikipedia.org/wiki/B)[C](https://en.wikipedia.org/wiki/C)[E](https://en.wikipedia.org/wiki/E)[H](https://en.wikipedia.org/wiki/H)[I](https://en.wikipedia.org/wiki/I)[J](https://en.wikipedia.org/wiki/J)[K](https://en.wikipedia.org/wiki/K)[M](https://en.wikipedia.org/wiki/M)[O](https://en.wikipedia.org/wiki/O)[P](https://en.wikipedia.org/wiki/P)[S](https://en.wikipedia.org/wiki/S)[T](https://en.wikipedia.org/wiki/T)[X](https://en.wikipedia.org/wiki/X), in addition to the capitals for the lowercase Cyrillic homoglyphs.

Cyrillic non-Russian problematic letters are [і](https://en.wikipedia.org/wiki/%D0%86) and [i](https://en.wikipedia.org/wiki/I), [ј](https://en.wikipedia.org/wiki/%D0%88) and [j](https://en.wikipedia.org/wiki/J), [ԛ](https://en.wikipedia.org/w/index.php?title=%D4%9B&action=edit&redlink=1) and [q](https://en.wikipedia.org/wiki/Q), [ѕ](https://en.wikipedia.org/wiki/%D0%85) and [s](https://en.wikipedia.org/wiki/S), [ԝ](https://en.wikipedia.org/w/index.php?title=%D4%9D&action=edit&redlink=1) and [w](https://en.wikipedia.org/wiki/W), [Ү](https://en.wikipedia.org/wiki/%D2%AE) and [Y](https://en.wikipedia.org/wiki/Y), while [Ғ](https://en.wikipedia.org/wiki/%D2%92) and [F](https://en.wikipedia.org/wiki/F), [Ԍ](https://en.wikipedia.org/wiki/%D4%8C) and [G](https://en.wikipedia.org/wiki/G) bear some resemblance to each other. Cyrillic [ӓ](https://en.wikipedia.org/wiki/%D3%92)[ё](https://en.wikipedia.org/wiki/%D0%81)[ї](https://en.wikipedia.org/wiki/%D0%87)[ӧ](https://en.wikipedia.org/wiki/%D3%A6) can also be used if an IDN itself is being spoofed, to fake [ä](https://en.wikipedia.org/wiki/%C3%84)[ë](https://en.wikipedia.org/wiki/%C3%8B)[ï](https://en.wikipedia.org/wiki/%C3%8F)[ö](https://en.wikipedia.org/wiki/%C3%96).

While [Komi De](https://en.wikipedia.org/wiki/Komi_De) ([ԁ](https://en.wikipedia.org/wiki/%D4%80)), [shha](https://en.wikipedia.org/wiki/Shha) ([һ](https://en.wikipedia.org/wiki/%D2%BA)), [palochka](https://en.wikipedia.org/wiki/Palochka) ([Ӏ](https://en.wikipedia.org/wiki/%D3%80)) and [izhitsa](https://en.wikipedia.org/wiki/Izhitsa) ([ѵ](https://en.wikipedia.org/wiki/%D1%B4)) bear strong resemblance to Latin d, h, l and v, these letters are either rare or archaic and are not widely supported in most standard fonts (they are not included in the [WGL-4](https://en.wikipedia.org/wiki/Windows_Glyph_List_4)). Attempting to use them could cause a [ransom note effect](https://en.wikipedia.org/wiki/Ransom_note_effect).

**Greek**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=5)]

From the [Greek alphabet](https://en.wikipedia.org/wiki/Greek_alphabet), only [omicron](https://en.wikipedia.org/wiki/Omicron) ο and sometimes [nu](https://en.wikipedia.org/wiki/Nu_(letter)) ν appear identical to a Latin alphabet letter in the lowercase used for URLs. Fonts that are in [italic type](https://en.wikipedia.org/wiki/Italic_type) will feature Greek alpha *α* looking like a Latin *a*.

This list increases if close matches are also allowed (such as Greek εικηρτυωχγ for eiknptuwxy). Using [capital letters](https://en.wikipedia.org/wiki/Capital_letter), the list expands greatly. Greek ΑΒΕΗΙΚΜΝΟΡΤΧΥΖ looks identical to Latin ABEHIKMNOPTXYZ. Greek ΑΓΒΕΗΚΜΟΠΡΤΦΧ looks similar to Cyrillic АГВЕНКМОПРТФХ (as do Cyrillic Л (Л) and Greek Λ in certain geometric sans-serif fonts), Greek letters κ and о look similar to Cyrillic к and о. Besides this Greek τ, φ can be similar to Cyrillic т, ф in some fonts, Greek δ resembles Cyrillic б in the [Serbian alphabet](https://en.wikipedia.org/wiki/Serbian_alphabet), and the Cyrillic *а* also italicizes the same as its Latin counterpart, making it possible to substitute it for alpha or vice versa. The lunate form of sigma, Ϲϲ, resembles both Latin Cc and Cyrillic Сс.

If an IDN itself is being spoofed, Greek beta β can be a substitute for German esszet [ß](https://en.wikipedia.org/wiki/%C3%9F) in some fonts (and in fact, [code page 437](https://en.wikipedia.org/wiki/Code_page_437) treats them as equivalent), as can Greek sigma ς for ç; accented Greek substitutes *όίά* can usually be used for *óíá* in many fonts, with the last of these (alpha) again only resembling *a* in italic type.

**Armenian**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=6)]

The [Armenian alphabet](https://en.wikipedia.org/wiki/Armenian_alphabet) can also contribute critical characters: several Armenian characters like օ, ո, ս, as well capital Տ and Լ are often completely identical to Latin characters in modern fonts. Symbols like ա can resemble Cyrillic ш. Beside that, there are symbols which look alike. ցհոօզս which look like ghnoqu, յ which resembles j (albeit dotless), and ք, which can either resemble p or f depending on the font. However, the use of Armenian is problematic. Not all standard fonts feature the Armenian glyphs (whereas the Greek and Cyrillic scripts are in most standard fonts). Because of this, Windows prior to Windows 7 rendered Armenian in a distinct font, [Sylfaen](https://en.wikipedia.org/wiki/Sylfaen_(typeface)), which supports Armenian, and the mixing of Armenian with Latin would appear obviously different if using a font other than Sylfaen or a [Unicode typeface](https://en.wikipedia.org/wiki/Unicode_typefaces). (This is known as a [ransom note effect](https://en.wikipedia.org/wiki/Ransom_note_effect).) The current version of [Tahoma](https://en.wikipedia.org/wiki/Tahoma_(typeface)), used in Windows 7, supports Armenian (previous versions did not). Furthermore, this font differentiates Latin g from Armenian ց.

Two letters in Armenian (Ձշ) also can resemble the number 2, Յ resembles 3, while another (վ) sometimes resembles the number 4.

**Hebrew**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=7)]

Hebrew spoofing is generally rare. Only three letters from that alphabet can reliably be used: samekh (ס), which sometimes resembles o, vav with diacritic (וֹ), which resembles an i, and heth (ח), which resembles the letter n. Less accurate approximants for some other alphanumerics can also be found, but these are usually only accurate enough to use for the purposes of [foreign branding](https://en.wikipedia.org/wiki/Foreign_branding) and not for substitution. Furthermore, the [Hebrew alphabet](https://en.wikipedia.org/wiki/Hebrew_alphabet) is written from right to left and trying to mix it with left-to-right glyphs may cause problems.

**Chinese**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=8)]

The [Chinese language](https://en.wikipedia.org/wiki/Chinese_language) can be problematic for homographs as many characters exist as both [traditional](https://en.wikipedia.org/wiki/Traditional_Chinese_characters) (regular script) and [simplified Chinese characters](https://en.wikipedia.org/wiki/Simplified_Chinese_characters). In the [.org](https://en.wikipedia.org/wiki/.org) domain, registering one variant renders the other unavailable to anyone; in [.biz](https://en.wikipedia.org/wiki/.biz) a single Chinese-language IDN registration delivers both variants as active domains (which must have the same domain name server and the same registrant). [.hk](https://en.wikipedia.org/wiki/.hk) (.香港) also adopts this policy.

**Other scripts**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=9)]

Other Unicode scripts in which homographs can be found include [Number Forms](https://en.wikipedia.org/wiki/Number_Forms) ([Roman numerals](https://en.wikipedia.org/wiki/Roman_numeral)), [CJK Compatibility](https://en.wikipedia.org/wiki/CJK_Compatibility) and [Enclosed CJK Letters and Months](https://en.wikipedia.org/wiki/Enclosed_CJK_Letters_and_Months) (certain abbreviations), Latin (certain digraphs), [Currency Symbols](https://en.wikipedia.org/wiki/Currency_Symbols_(Unicode_block)), [Mathematical Alphanumeric Symbols](https://en.wikipedia.org/wiki/Mathematical_Alphanumeric_Symbols), and [Alphabetic Presentation Forms](https://en.wikipedia.org/wiki/Alphabetic_Presentation_Forms) ([typographic ligatures](https://en.wikipedia.org/wiki/Typographic_ligature)).

**Accented characters**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=10)]

Two names which differ only in an accent on one character may look very similar, particularly when the substitution involves the [dotted letter i](https://en.wikipedia.org/wiki/Dotted_and_dotless_i); the tittle (dot) on the i can be replaced with a [diacritic](https://en.wikipedia.org/wiki/Diacritic) (such as a [grave accent](https://en.wikipedia.org/wiki/Grave_accent) or [acute accent](https://en.wikipedia.org/wiki/Acute_accent); both ì and í are included in most standard character sets and fonts) that can only be detected with close inspection. In most top-level domain registries, wíkipedia.tld (xn--wkipedia-c2a.tld) and wikipedia.tld are two different names which may be held by different registrants.[[5]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-5) One exception is [.ca](https://en.wikipedia.org/wiki/.ca), where reserving the plain-[ASCII](https://en.wikipedia.org/wiki/ASCII) version of the domain prevents another registrant from claiming an accented version of the same name.[[6]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-6)

**Non-displayable characters**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=11)]

Unicode includes many characters which are not displayed by default, such as the [zero-width space](https://en.wikipedia.org/wiki/Zero-width_space). In general, [ICANN](https://en.wikipedia.org/wiki/ICANN) prohibits any domain with these characters from being registered, regardless of TLD.

**Known homograph attacks**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=12)]

In 2011, an unknown source (registering under the name "Completely Anonymous") registered a domain name homographic to television station [KBOI-TV](https://en.wikipedia.org/wiki/KBOI-TV)'s to create a [fake news website](https://en.wikipedia.org/wiki/List_of_fake_news_websites). The sole purpose of the site was to spread an [April Fool's Day](https://en.wikipedia.org/wiki/April_Fool%27s_Day) joke regarding the [Governor of Idaho](https://en.wikipedia.org/wiki/Governor_of_Idaho) issuing a supposed ban on the sale of music by [Justin Bieber](https://en.wikipedia.org/wiki/Justin_Bieber).[[7]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-7)[[8]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-8)

In September 2017, security researcher Ankit Anubhav discovered an IDN homograph attack where the attackers registered adoḅe.com to deliver the Betabot trojan.[[9]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-9)

Defending against the attack[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=13)]

**Client-side mitigation**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=14)]

The simplest defense is for web browsers not to support IDNA or other similar mechanisms, or for users to turn off whatever support their browsers have. That could mean blocking access to IDNA sites, but generally browsers permit access and just display IDNs in [Punycode](https://en.wikipedia.org/wiki/Punycode). Either way, this amounts to abandoning non-ASCII domain names.

* [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) versions 51 and later use an algorithm similar to the one used by Firefox. Previous versions display an IDN only if all of its characters belong to one (and only one) of the user's preferred languages. [Chromium](https://en.wikipedia.org/wiki/Chromium_(web_browser)), and [Chromium based browsers](https://en.wikipedia.org/wiki/Chromium_(web_browser)#Browsers_based_on_Chromium), such as [New Microsoft Edge](https://en.wikipedia.org/wiki/Microsoft_Edge#Anaheim_(2019%E2%80%93present)) and [Opera](https://en.wikipedia.org/wiki/Opera_(web_browser)) also use the same algorithm. [[10]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-10)[[11]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-11)
* [Safari's](https://en.wikipedia.org/wiki/Safari_(web_browser)) approach is to render problematic character sets as [Punycode](https://en.wikipedia.org/wiki/Punycode). This can be changed by altering the settings in [Mac OS X](https://en.wikipedia.org/wiki/MacOS)'s system files.[[12]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-12)
* [Mozilla Firefox](https://en.wikipedia.org/wiki/Mozilla_Firefox) versions 22 and later display IDNs if either the TLD prevents homograph attacks by restricting which characters can be used in domain names or labels do not mix scripts for different languages. Otherwise IDNs are displayed in Punycode.[[13]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-13)[[14]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-14)
* [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) versions 7 and later allow IDNs except for labels that mix scripts for different languages. Labels that mix scripts are displayed in Punycode. There are exceptions to locales where ASCII characters are commonly mixed with localized scripts.[[15]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-15) Internet Explorer 7 was capable of using IDNs, but it imposes restrictions on displaying non-ASCII domain names based on a user-defined list of allowed languages and provides an anti-phishing filter that checks suspicious Web sites against a remote database of known phishing sites.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]
* Old Microsoft Edge converts all Unicode into Punycode.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

As an additional defense, Internet Explorer 7, Firefox 2.0 and above, and Opera 9.10 include phishing filters that attempt to alert users when they visit malicious websites.[[16]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-16)[[17]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-17)[[18]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-18) As of April 2017, several browsers (including Chrome, Firefox and Opera) were displaying IDNs consisting purely of Cyrillic characters normally (not as punycode), allowing spoofing attacks. Chrome tightened IDN restrictions in version 59 to prevent this attack.[[19]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-19)[[20]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-20)

[Browser extensions](https://en.wikipedia.org/wiki/Browser_extension) like No Homo-Graphs are available for [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) and [Firefox](https://en.wikipedia.org/wiki/Firefox) that check whether the user is visiting a website which is a homograph of another domain from a user-defined list.[[21]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-21)

These methods of defense only extend to within a browser. Homographic URLs that house malicious software can still be distributed, without being displayed as Punycode, through [e-mail](https://en.wikipedia.org/wiki/E-mail), [social networking](https://en.wikipedia.org/wiki/Social_networking) or other Web sites without being detected until the user actually clicks the link. While the fake link will show in Punycode when it is clicked, by this point the page has already begun loading into the browser and the malicious software may have already been downloaded onto the computer.

**Server-side/registry operator mitigation**[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=15)]

[ICANN](https://en.wikipedia.org/wiki/ICANN) has implemented a policy prohibiting any potential internationalized TLD from choosing letters that could resemble an existing Latin TLD and thus be used for homograph attacks. Proposed IDN TLDs [.бг](https://en.wikipedia.org/wiki/.%D0%B1%D0%B3) (Bulgaria), [.укр](https://en.wikipedia.org/wiki/.%D1%83%D0%BA%D1%80) (Ukraine) and [.ελ](https://en.wikipedia.org/wiki/.%CE%B5%CE%BB) (Greece) have been rejected or stalled because of their perceived resemblance to Latin letters. All three (and Serbian [.срб](https://en.wikipedia.org/wiki/.%D1%81%D1%80%D0%B1) and Mongolian [.мон](https://en.wikipedia.org/wiki/.%D0%BC%D0%BE%D0%BD)) have later been accepted.[[22]](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_note-22) Three-letter TLD are considered safer than two-letter TLD, since they are harder to match to normal Latin ISO-3166 country domains; although the potential to match new generic domains remains, such generic domains are far more expensive than registering a second- or third-level domain address, making it cost-prohibitive to try to register a homoglyphic TLD for the sole purpose of making fraudulent domains (which itself would draw ICANN scrutiny).

The Russian registry operator [Coordination Center for TLD RU](https://en.wikipedia.org/wiki/Coordination_Center_for_TLD_RU) only accepts Cyrillic names for the top-level domain [.рф](https://en.wikipedia.org/wiki/.%D1%80%D1%84), forbidding a mix with Latin or Greek characters. However the problem in [.com](https://en.wikipedia.org/wiki/.com) and other [gTLDs](https://en.wikipedia.org/wiki/GTLD) remains open.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

See also[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=16)]

* [Internationalized domain name](https://en.wikipedia.org/wiki/Internationalized_domain_name)
* [Homoglyph](https://en.wikipedia.org/wiki/Homoglyph), [Duplicate characters in Unicode](https://en.wikipedia.org/wiki/Duplicate_characters_in_Unicode), [Unicode equivalence](https://en.wikipedia.org/wiki/Unicode_equivalence)
* [Typosquatting](https://en.wikipedia.org/wiki/Typosquatting)

References[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=17)]

* 1. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-tr36_1-0) [***b***](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-tr36_1-1) ["Unicode Security Considerations"](http://unicode.org/reports/tr36/tr36-8.html#Bidirectional_Text_Spoofing), Technical Report #36, 2010-04-28
  2. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-tha_2-0) [***b***](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-tha_2-1) Evgeniy Gabrilovich and Alex Gontmakher, [*"Archived copy"*](https://web.archive.org/web/20200102175251/http:/www.cs.technion.ac.il/~gabr/papers/homograph_full.pdf)*(PDF). Archived from*[*the original*](http://www.cs.technion.ac.il/~gabr/papers/homograph_full.pdf)*(PDF) on 2020-01-02. Retrieved 2005-12-10.*, Communications of the ACM, 45(2):128, February 2002
  3. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-3) [IDN hacking disclosure by shmoo.com](http://www.shmoo.com/idn/) [Archived](https://web.archive.org/web/20050320020225/http:/www.shmoo.com/idn/) 2005-03-20 at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine)
  4. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-4) [*"Chrome and Firefox Phishing Attack Uses Domains Identical to Known Safe Sites"*](https://www.wordfence.com/blog/2017/04/chrome-firefox-unicode-phishing/)*. Wordfence. 2017-04-14. Retrieved 2017-04-18.*
  5. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-5) There are various [Punycode](https://en.wikipedia.org/wiki/Punycode) converters online, such as <https://www.hkdnr.hk/idn_conv.jsp>
  6. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-6) [*"Archived copy"*](https://web.archive.org/web/20150907200124/http:/cira.ca/news/ca-takes-french-accent)*. Archived from*[*the original*](http://cira.ca/news/ca-takes-french-accent)*on 2015-09-07. Retrieved 2015-09-22.*
  7. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-7) [Fake website URL not from KBOI-TV](http://www.kboi2.com/news/local/119020194.html) [Archived](https://web.archive.org/web/20110405021458/http:/www.kboi2.com/news/local/119020194.html) 2011-04-05 at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine). KBOI-TV. Retrieved 2011-04-01.
  8. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-8) [Boise TV news website targeted with Justin Bieber prank](http://www.ktvb.com/news/business/Boise-TV-news-website-targeted-with-Justin-Bieber-prank-119026184.html)[Archived](https://web.archive.org/web/20120315224532/http:/www.ktvb.com/news/business/Boise-TV-news-website-targeted-with-Justin-Bieber-prank-119026184.html) 2012-03-15 at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine). KTVB. Retrieved 2011-04-01.
  9. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-9) *Mimoso, Michael (2017-09-06).*[*"IDN Homograph Attack Spreading Betabot Backdoor"*](https://threatpost.com/idn-homograph-attack-spreading-betabot-backdoor/127839/)*. Threatpost. Retrieved 2020-09-20.*
  10. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-10) [*"Internationalized Domain Names (IDN) in Google Chrome"*](https://chromium.googlesource.com/chromium/src.git/+/master/docs/idn.md)*. chromium.googlesource.com. Retrieved 2020-08-26.*
  11. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-11) [*"Upcoming update with IDN homograph phishing fix - Blog"*](https://blogs.opera.com/security/2017/04/upcoming-update-idn-homograph-phishing-fix/)*. Opera Security. 2017-04-21. Retrieved 2020-08-26.*
  12. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-12) [*"About Safari International Domain Name support"*](https://support.apple.com/kb/TA22996)*. Retrieved 2017-04-29.*
  13. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-13) [*"IDN Display Algorithm"*](https://wiki.mozilla.org/IDN_Display_Algorithm)*. Mozilla. Retrieved 2016-01-31.*
  14. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-14) [*"Bug 722299"*](https://bugzilla.mozilla.org/show_bug.cgi?id=722299)*. Bugzilla.mozilla.org. Retrieved 2016-01-31.*
  15. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-15) *Sharif, Tariq (2006-07-31).*[*"Changes to IDN in IE7 to now allow mixing of scripts"*](http://blogs.msdn.com/ie/archive/2006/07/31/684337.aspx)*. IEBlog. Microsoft. Retrieved 2006-11-30.*
  16. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-16) *Sharif, Tariq (2005-09-09).*[*"Phishing Filter in IE7"*](http://blogs.msdn.com/ie/archive/2005/09/09/463204.aspx)*. IEBlog. Microsoft. Retrieved 2006-11-30.*
  17. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-17) [*"Firefox 2 Phishing Protection"*](http://www.mozilla.com/en-US/firefox/phishing-protection/)*. Mozilla. 2006. Retrieved 2006-11-30.*
  18. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-18) [*"Opera Fraud Protection"*](http://www.opera.com/docs/fraudprotection/)*. Opera Software. 2006-12-18. Retrieved 2007-02-24.*
  19. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-19) [Chrome and Firefox Phishing Attack Uses Domains Identical to Known Safe Sites](https://www.wordfence.com/blog/2017/04/chrome-firefox-unicode-phishing/)
  20. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-20) [Phishing with Unicode Domains](https://www.xudongz.com/blog/2017/idn-phishing/)
  21. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-21) [*"No Homo-Graphs"*](https://addons.mozilla.org/en-US/firefox/addon/no-homo/)*. em\_te. 2018-06-28. Retrieved 2020-02-18.*
  22. [**^**](https://en.wikipedia.org/wiki/IDN_homograph_attack#cite_ref-22) [IDN ccTLD Fast Track String Evaluation Completion](https://www.icann.org/resources/pages/string-evaluation-completion-2014-02-19-en) [Archived](https://web.archive.org/web/20141017110723/https:/www.icann.org/resources/pages/string-evaluation-completion-2014-02-19-en)2014-10-17 at the [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine)

External links[[edit](https://en.wikipedia.org/w/index.php?title=IDN_homograph_attack&action=edit&section=18)]

* [Homograph attack generator](http://www.irongeek.com/homoglyph-attack-generator.php)
* [Phishing with Unicode Domains](https://www.xudongz.com/blog/2017/idn-phishing/)

An[internationalized domain name (IDN) homograph attack](https://zvelo.com/network-security-malicious-threats-and-common-computer-definitions/#homograph-attack-idn) is a method of deceiving computer users about the remote computer they’re communicating with. It exploits the fact that many characters are homographs, meaning they look alike. Homographs allow a malicious party to create an IDN that appears very similar to an established domain, which can then be used to lure users to the new website. Alphabets that have the largest number of homographs with the Latin alphabet are generally the most useful for perpetrating an IDN homograph attack since most [top-level domains (TLDs)](https://zvelo.com/anatomy-of-full-path-url-hostname-protocol-path-more/) use the Latin alphabet. A variety of defenses are currently available for protecting computer users from these attacks.

Table of Contents

Overview

An IDN homograph attack is similar to another type of domain name spoofing known as [typosquatting](https://zvelo.com/network-security-malicious-threats-and-common-computer-definitions/#typosquatting). Both techniques attempt to deceive users by using a new domain name that’s similar to an established name, although they exploit different types of similarities. Typosquatting uses a new domain name that’s spelled differently from the established name, but uses the same character set. A homograph attack typically uses a domain name that contains characters from other character sets, which requires the user to click on a hyperlink of the new name. This type of attack rarely works with a manual entry of the domain name since a user is unlikely to unintentionally enter a homograph.

Some domain names can be used for both typosquatting and homograph spoofing. For example, a spoof that uses a domain name containing an uppercase “O” instead of the numeral “0” would be both types of attack. The success of this type of spoof is highly dependent on the typeface the computer uses, as these two characters are physically identical in some typefaces.

Internationalized Domain Names

An IDN is a domain name that’s displayed in at least one language-specific alphabet. They’re stored as ASCII strings in the Domain Name System (DNS), allowing IDNs to use the full Unicode character set in a backward-compatible manner. However, this approach also increases the effectiveness of homograph attacks since it expanded the character set from the relatively small number of characters in a single alphabet to the many thousands of characters used by the world’s written languages. An attacker can thus register a new domain name that looks like the domain name of a legitimate website by substituting homographs. The new domain name can then be used to direct users to the spoof site, where information such as account passwords can be collected.

Character Sets

Unicode includes many different character sets, some of which have similar-looking characters. For example, the Cyrillic, Greek and Latin alphabets all use the character “O,” although these characters are assigned different codes in Unicode. This physical similarity creates the opportunity for successful IDN homograph attacks. The character sets with the greatest value in homograph attacks include ASCII, Cyrillic, Greek and Armenian.

ASCII

The most common homographs in [ASCII](https://en.wikipedia.org/wiki/ASCII) include an uppercase “O” and the numeral “0”, and a lowercase “L” and the numeral “1”. Depending on the typeface, an “L” and “1” can also be homographs for an uppercase “I.” Some ASCII homographs are combinations of letters, such as a lowercase “R” and “N” for a lowercase “M”.

Note how similar “rnicrosoft.com” appears to “microsoft.com”, depending on the font. This similarity is especially striking in the Tahoma font, which is the default for the URL address bar in Windows XP.

Cyrillic

Cyrillic is by the most popular alphabet for IDN homograph attacks, primarily because it contains many homographs in the Latin alphabet. Lowercase homographs have the greatest value for these attacks, since most users enter URLs in lowercase. Cyrillic has seven (7) lowercase characters that are identical or virtually identical to characters in the Latin alphabet. The characters a, c, e, o, p, x and y exist in both alphabets and are physically indistinguishable in most fonts.

Uppercase homographs between the Cyrillic and Latin alphabets include the letters A, B, C, E, H, I, J, K, M, O, P, S, T and X. Furthermore, the Cyrillic characters З, Ч and б are very similar to the Latin numerals 3, 4 and 6, depending on the font.

Greek

The Greek alphabet is less common for homograph attacks since only the lowercase “o” and “v” are identical with their Latin counterparts. However, Greek also has 10 other lowercase characters that are close matches for the Latin lowercase letters e, i, k, n, p, t, u, w, x and y. Furthermore, the lowercase Greek “α” also looks like a Latin lowercase “a” in an italic font. Fourteen uppercase homographs exist between the Greek and Latin alphabets, with the letters Α, Β, Ε, Η, Ι, Κ, Μ, Ν, Ο, Ρ, Τ, Χ, Υ and Ζ being identical.

Armenian

The Armenian alphabet also has several homographs with the Latin alphabet that make it useful in homograph attacks. For example, the lowercase letters o, n and u and uppercase letters S and L are physically identical in these two alphabets for most modern fonts. The primary disadvantage of using Armenian is that many standard fonts don’t include Armenian characters, whereas most standard fonts do include Cyrillic and Greek characters.

Defenses

The defenses to homograph attacks may be classified into client-side and server side techniques.

Client Side

The general approach to defending against homograph attacks on the client side is to ensure that web browsers don’t support [Internationalizing Domain Names in Applications (IDNA)](https://www.icann.org/resources/pages/idn-2012-02-25-en) at all or allow users to disable such support. This typically means that a web browsers displays IDNs in [Punycode](https://en.wikipedia.org/wiki/Punycode), which is a method of representing Unicode characters with the smaller ASCII subset. A less common solution is to deny access to IDNA sites.

Check out this [Punycode converter](https://www.punycoder.com/) for more information.

Server Side

Server-side defenses to homograph attacks primarily rely on policies implemented by the [Internet Corporation for Assigned Names and Numbers (ICANN)](https://www.icann.org/). These policies generally prohibit internationalized TLDs from containing non-Latin characters that could cause it to resemble an existing TLD that uses Latin characters. ICANN also encourages the use of longer TLDs, making them more difficult to resemble existing Latin TLDs.

**[READ:](https://zvelo.com/network-security-malicious-threats-and-common-computer-definitions/" \t "_self)****[Network Security, Malicious Threats, and Common Computer Definitions](https://zvelo.com/network-security-malicious-threats-and-common-computer-definitions/" \t "_self)**

Conclusions

The best way to protect yourself and your organization from internationalized domain name attacks is through continuous training for all employees—as well as implementing robust web filtering designed for cybersecurity.

Remember, always check URLs in emails BEFORE you click on them. You should also always ensure you have connected to the right remote location (i.e. check the URL once you’ve arrived at the target site).

Hey guys Abhishek this side. This is my first writeup. This article is about a vulnerability I was able to find in the BugCrowd private program.

What is IDN homograph attack?

The [**internationalized domain name**](https://en.wikipedia.org/wiki/Internationalized_domain_name) (**IDN**) **homograph attack** is a way a malicious party may deceive computer users about what remote system they are communicating with, by exploiting the fact that many different [characters](https://en.wikipedia.org/wiki/Grapheme) look alike (i.e., they are homographs, hence the term for the attack, although technically [homograph](https://en.wikipedia.org/wiki/Homoglyph) is the more accurate term for different characters that look alike). For example, a regular user of [example.com](https://en.wikipedia.org/wiki/Example.com) may be lured to click a link where the [Latin](https://en.wikipedia.org/wiki/Latin_alphabet) character [“a”](https://en.wikipedia.org/wiki/A) is replaced with the [Cyrillic](https://en.wikipedia.org/wiki/Cyrillic_script) character [“а”](https://en.wikipedia.org/wiki/A_(Cyrillic)).

One day I saw interesting #bugbountytips post on twitter <https://twitter.com/musiclouderlml/status/1276987908340232193?s=19>

Then I thought why not give a try. I started hunting for this bug on Bugcrowd private program. Let’s call target.com .

Tha web application “https://target.com/forgot-password?email=" fails to properly validate the value of “email” which was used to takeover the user’s account by changing his password using IDN homograph attack.

IDN homograph attack exploits the fact that many different charachters look like a is different from á Because in that we used a acute accent which looks like exactly a, Suppose the victim’s account is abc@gmail.com , attacker ask password reset link for abc@gmáil.com, target.com’s mail system send password reset link of victim- abc@gmail.com to the attacler mail- abc@xn — gmil-6na.com, To perform this attack , attacker have to buy domain xn — gmil-6na.com

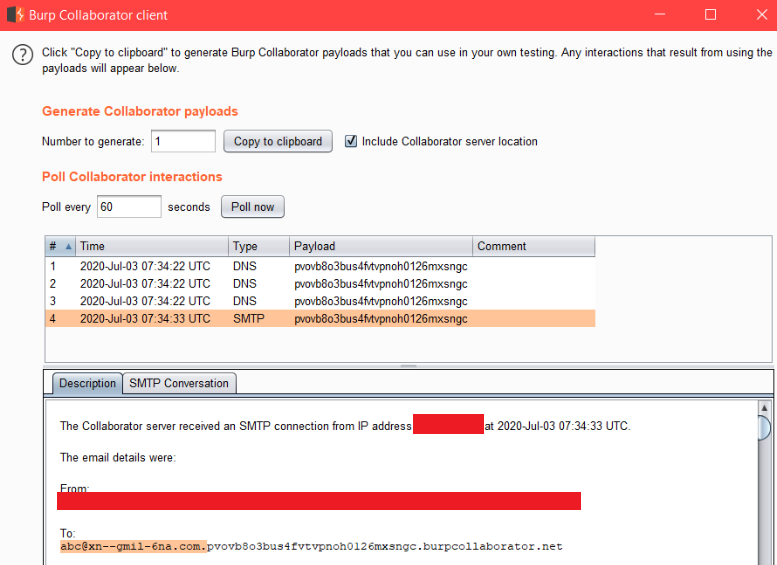
How to test without buying domain ?

* The answer is using burp collaborator client.

We have to create a account on target.com with email- abc@gmail.com.burpcollaboratorpayloadhere

So when we ask password reset link for abc@gmáil.com.burpcollaboratorpayloadhere , target.com’s send password reset link of user- abc@gmail.com.burpcollaboratorpayloadhere to the mail- abc@xn — gmil-6na.com.burpcollaboratorpayloadhere , the password reset link received on Burp collaborator client. Make sure to check in burp collaborator client , received email details: To- abc@xn — gmil-6na.com.burpcollaboratorpayloadhere.

Image for post

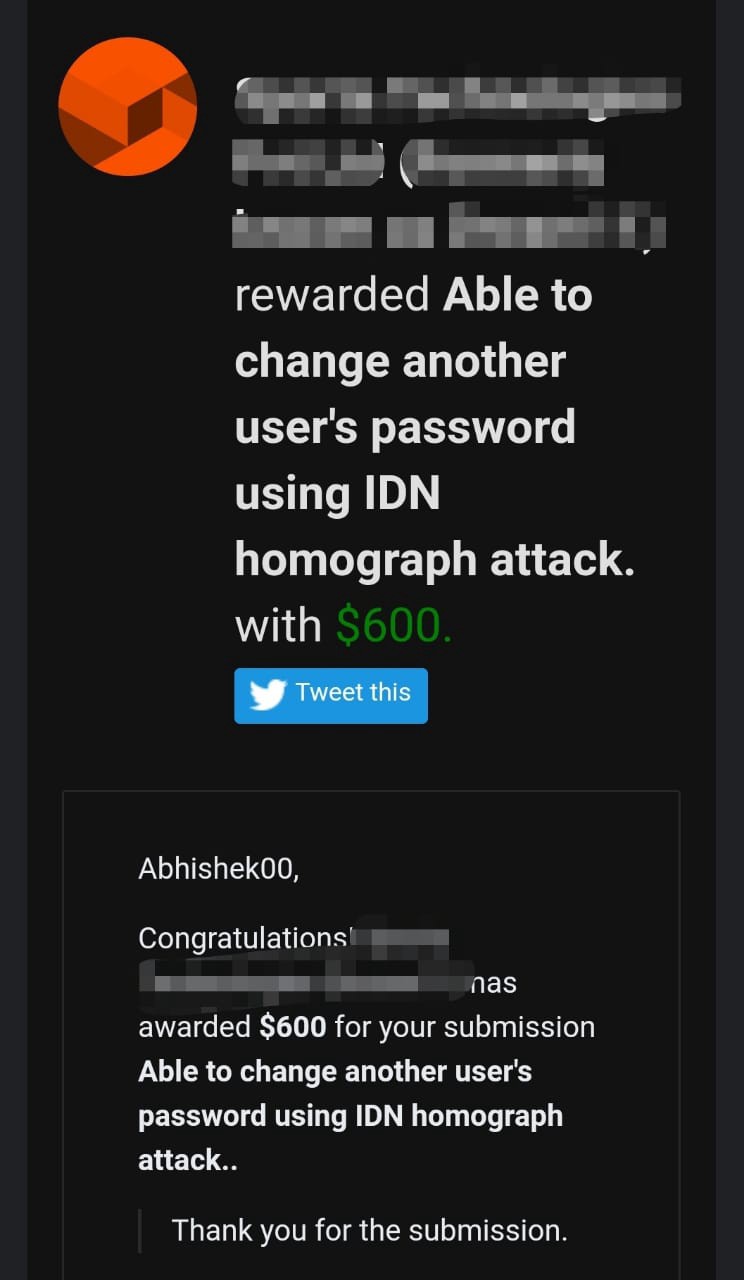


Steps to reproduce-

1. Open the burp collaborator client > Generate Collaborator payload .
2. Go to the sign up page of target.com and create a new account with email- abc@gmail.com.burpcollaboratorpayloadhere
3. Now if the target.com has email confirmation > you will receive the email confirmation link in burp collaborator client > verify the email.
4. Go to password reset page of target.com > enter email as abc@gmáil.com.burpcollaboratorpayloadhere
5. If the target.com is vulnerable then it will send password reset link to the mail- abc@xn — gmil-6na.com.burpcollaboratorpayloadhere and you will receive password reset link in burp collaborator client. Make sure to check in burp collaborator client -received email details: To- abc@xn — gmil-6na.com.burpcollaboratorpayloadhere.
6. Now you can change the password and access the victim’s account.

Result-

Image for post



Special thanks to [https://twitter.com/musiclouderlml](https://twitter.com/musiclouderlml/status/1276987908340232193?s=19) for sharing #bugbountytips.

Hope you guys enjoyed. Thanks for reading.

The internationalized domain name (IDN) homograph attack is used to form domain names that visually resemble legitimate domain names, albeit, using a different set of characters [1]. For example, the IDN "xn--akmai-yqa.com" which appears in unicode as "akámai.com" visually resembles the legitimate domain name "akamai.com". Attackers often apply IDN homograph attacks to form domain names that are used for malicious purposes, such as malware distribution [2] or phishing [3], while appearing trustworthy to victims.

The prevention of IDN homograph attacks is based on two complementary approaches: registration and access. On the client-side, major web browsers implement algorithms that attempt to identify IDN homograph attacks and present them in their true IDN form instead of unicode, to reduce their resemblance to legitimate domain names [4][5]. On the server-side, the Internet Corporation for Assigned Names and Numbers (ICANN) issued several policies over the past few years to limit the efficacy of IDN homograph attacks in ccTLD registry operators [6]. While both approaches do not eliminate the threat posed by IDN homograph attacks, they do create a tradeoff between user experience and security through supporting internationalized domain names and limiting attacks.

In this blog, we're going to explore the prevalence of user visits to domain names that were formed using IDN homograph attacks (henceforth referred to as homograph IDNs) and learn about the effectiveness of preventive approaches to the potential underlying threat. We attempt to answer these questions by applying an algorithm that classifies whether an IDN was formed by a homograph attack on Akamai's large scale DNS traffic. Then, we track and analyze the access patterns of users to every IDN that was classified as a homograph attack.

The homograph IDNs below are examples, but we caution users against searching them out and browsing them directly. While it's clear that homograph IDNs are potentially malicious, they're not all malicious; even when they're not malicious, they're almost certainly not benign.

Detecting IDN homograph attacks

The algorithm classifies an IDN as a homograph attack if two conditions are met. First, the unicode form of the IDN must resemble a legitimate and popular domain name (without the TLD). The algorithm maintains a constant list of such domain names that are likely to be spoofed by attackers, and the resemblance is measured using character replacement maps. Second, the IDN and its legitimate and popular domain name match must be registered by different owners. Examples of IDNs that were identified as homograph attacks by the algorithm are displayed in Table 1.

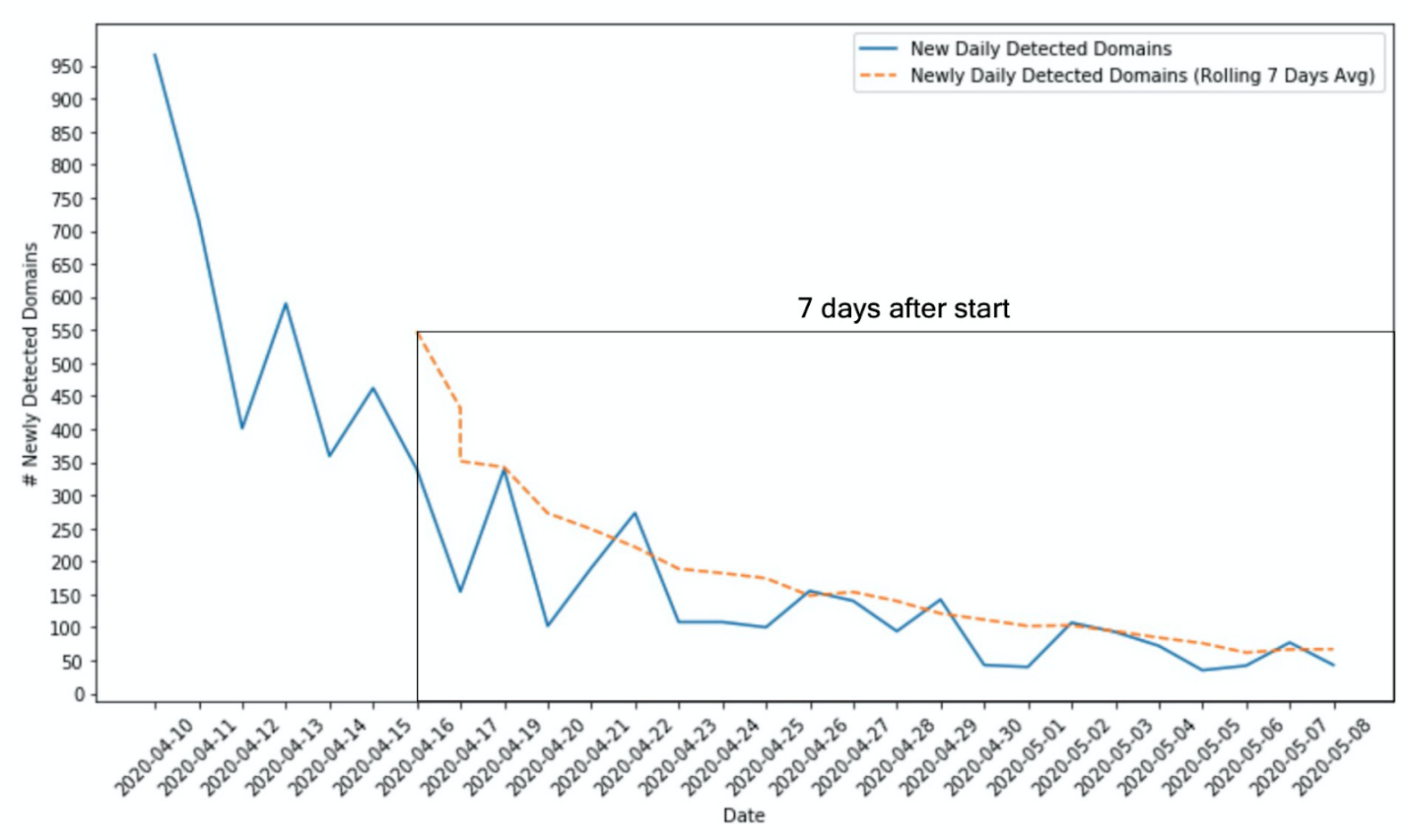
|  |  |  |
| --- | --- | --- |
| **IDN** | **Unicode** | **Legitimate match** |
| xn--alixpress-d4a.com | aliéxpress.com | aliexpress.com |
| xn--go0gl-3we.fm | go0glе.fm | google.com |
| xn--mazon-wqa.com | ámazon.com | amazon.com |

*Table 1: Example of IDN homograph attacks that were detected by the algorithm*

The Akamai DNS traffic that the algorithm processed  is compared against DNS queries made by millions of devices belonging to home users and small businesses, on more than 30 worldwide ISP networks. The volume of devices and users is estimated based on the source IP addresses. This method has the limitation of overestimation when devices use multiple IP addresses over time. We try to overcome this limitation as much as possible by looking at a small time frame of data, and conducting frequent (daily) sampling, but that should be noted by readers.

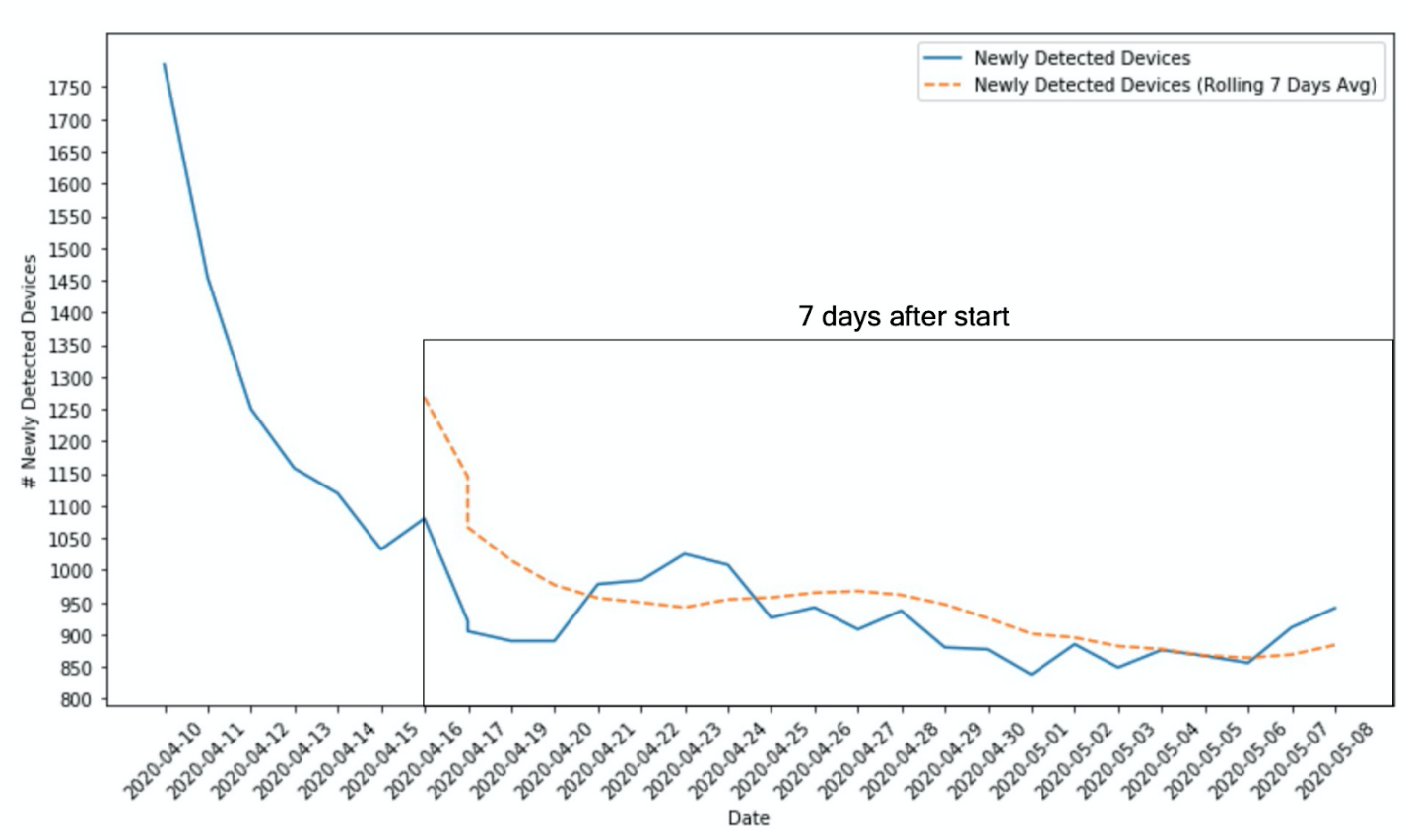
The prevalence of IDN homograph attacks

The IDN homograph detection algorithm was applied on a daily basis over the course of 32 days. Throughout this period, we identified 6670 homograph IDNs in total. We stress that every identified homograph IDN was queried in DNS traffic by at least one device in contrast to registered IDNs that are never accessed in traffic. Within 7 out of the last 32 days, we identified 67 newly-detected daily domains on average that were never seen before (as portrayed in Figure 1). Therefore, we conclude that every day at least several dozens of registered homograph IDN are accessed for the first time, despite the existence of preventive measures.



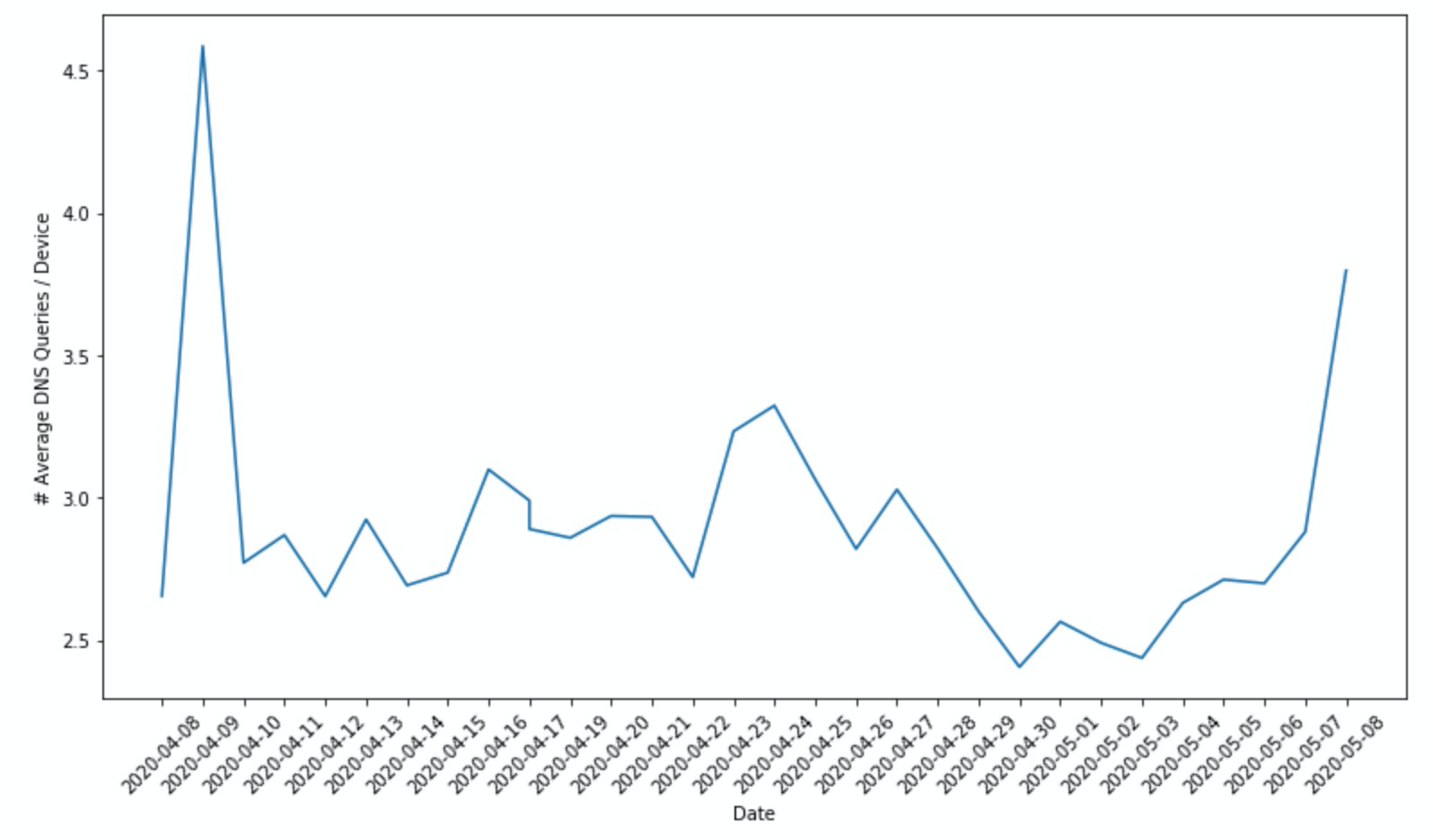
*Figure 1: The number of  newly detected homograph IDN on daily basis*

Moreover, we identified a total number of 29,071 devices that accessed at least one homograph IDN within the examined 32 days period. Within 7 out of the last 32 days, there were, on average, over 850 devices that daily accessed homograph IDNs for the first time (see Figure 2). Therefore, we conclude that every day hundreds of Internet devices unintentionally access a homograph IDN for the first time.



*Figure 2: The rate of daily device that access homograph IDN for the first time (newly visiting devices)*

Every device that made at least one access to a homograph IDN within a given day made between 2-5 access attempts (measured by DNS queries) to homograph IDNs in total (see Figure 3). Multiple access attempts might result from automatic behavior such as link prefetching [7] or user behavior such hitting a refresh button. Nevertheless, the fact that a device never makes more than 5 daily access attempts on average strengthens our claim that visits to homograph IDNs are unintentional and not recurring; otherwise, the average would be significantly higher.



*Figure 3: The average rate of daily DNS queries made by devices to homograph IDNs*

Conclusions and takeaways

IDN homograph attacks are used by attackers to form domain names that look trustworthy to victims in order to serve phishing pages and malware. Despite the emergence of preventive measures against IDN homograph attacks, the results of our analysis demonstrate that every day dozens of homograph IDNs are accessed for the first time, and hundreds of new users unintentionally access homograph IDNs for the first time. In order to best protect your device and network against phishing and malware, it is advised to use solutions that protect against IDN homograph attacks such as supporting web browsers (e.g., Chrome or Firefox) and carefully inspecting domain names for suspicious unicode characters (i.e., watch your steps). Moreover, the use of network security solutions that scan traffic to identify and block IDN homograph attacks is another layer of defense that reduces the risk of accidentally accessing these potentially malicious domains.

Akamai's Enterprise Threat Protector is a cloud based Secure Web Gateway that uses real-time threat intelligence to identify IDN homograph attacks, as well as other phishing and malware delivery techniques to protect your network and users.

References

[1] <https://en.m.wikipedia.org/wiki/IDN_homograph_attack>

[2] <https://threatpost.com/idn-homograph-attack-spreading-betabot-backdoor/127839/>

[3] <https://securityboulevard.com/2019/08/new-homograph-phishing-attack-impersonates-bank-of-valletta-leverages-valid-tls-certificate/>

[4] <https://www.chromium.org/developers/design-documents/idn-in-google-chrome>

[5] <https://wiki.mozilla.org/IDN_Display_Algorithm>

[6] <https://www.icann.org/resources/pages/announcements-2014-02-19-en>

[7] <https://en.wikipedia.org/wiki/Link_prefetching>

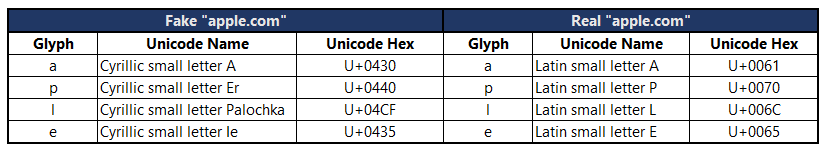
In April, Xudong Zheng, a security enthusiast based in New York, [found](https://www.xudongz.com/blog/2017/idn-phishing/) a flaw in some modern browsers in the way they handle domain names. While [Chrome](https://www.chromium.org/developers/design-documents/idn-in-google-chrome), [Firefox](https://wiki.mozilla.org/IDN_Display_Algorithm), and [Opera](https://www.opera.com/security/advisory/788) already have security measures in place to cue users that they might be visiting a destination they thought was legitimate, at that time these browsers did not flag a fake domain name that used all Latin look-alike characters taken from another foreign language. Zheng demonstrated this when he created and registered a [proof-of-concept (PoC)](https://blog.malwarebytes.com/glossary/proof-of-concept/) page for the domain, аррӏе.com, which was written in pure Cyrillic characters.

What is a homograph attack?

A homograph attack is a method of deception wherein a threat actor leverages on the similarities of character scripts to create and register phony domains of existing ones to fool users and lure them into visiting. This attack has some known aliases: homoglyph attack, script spoofing, and homograph domain name spoofing*.*Characters—i.e., letters and numbers—that look alike are called *homoglyphs* or *homographs*, thus the name of the attack. Examples of such are the Latin small letter O (U+006F) and the Digit zero (U+0030). Hypothetically, one might register *bl00mberg.com* or *g00gle.com* and get away with it. But in this day and age, such simple character swaps could be easily detected.

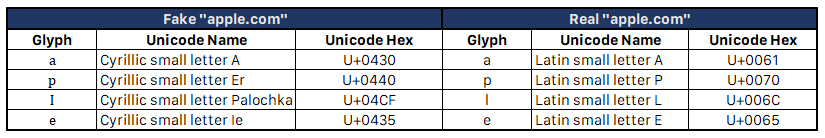
In an [internationalized domain name (IDN)](https://en.wikipedia.org/wiki/Internationalized_domain_name) homograph attack, a threat actor creates and registers one or several fake domains using at least one look-alike character from a different language. Again, hypothetically, one might register *gοοgle.com*, but not before swapping the Latin small letter O (U+006F) with the Greek small letter Omicron (U+03BF).

Zheng’s PoC is another example of an IDN homograph attack, so let’s list down each character he used to illustrate how this particular attack can be highly successful and dangerous if used in the wild. Interestingly, an operating system’s typeface of choice could make it easy or difficult for users to visually differentiate non-Latin characters from Latin ones.



*Table 1: We used*[*Segoe UI*](https://en.wikipedia.org/wiki/Segoe)*, Microsoft’s system-wide typeface, here.*

To the human eye, these Cyrillic glyphs can easily be confused with their Latin counterparts. Computers, however, read these confusables differently, as we can see from the different hex codes assigned to them.



*Table 2: We used*[*San Francisco*](https://en.wikipedia.org/wiki/San_Francisco_(sans-serif_typeface))*, Apple’s system-wide typeface, here. It’s worth noting that OSX distinguishes the Cyrillic small letter Palochka from the Latin small letter L; however, it cannot show the difference between the Latin small letter L with the Latin capital letter I, as per the text “Cyrillic small letter Ie”.*

According to [this bug report](https://bugs.chromium.org/p/chromium/issues/detail?id=683314), it seems that even the system-wide font for Linux doesn’t distinguish confusable characters either.

The use of all-Cyrillic glyphs—or any other non-Latin characters for this matter—for domain names isn’t the problem. IDN has made it possible for internet users around the globe to create and access domains using their native language scripts. The problem is when these glyphs are misused to deceive internet users.

Is this a new form of online threat?

Homograph attacks have been around for years. As far as we know, Zhang’s PoC was the first of its kind to make headlines and spark a conversation among internet users.

Below are other examples of homographed domains and how they were used:

* To raise awareness, a security consultant [highlighted the common misconception](https://www.grahamcluley.com/lloydsbank-homographic-phishing-problem/) that sometimes a Latin capital letter I (U+0049) looks similar to a Latin small letter L (U+006C) by registering a fake Lloyds Bank website and adding an SSL certificate to it to make it look as legitimate as the real one.
* A security researcher from NTT Security [shared his experience](https://www.solutionary.com/resource-center/blog/2017/01/idn-homograph-attacks/) about a friend of his who received several Google Analytics spam containing the domain, *secret[DOT]ɢoogle[DOT]com*. The “ɢ” there wasn’t the Latin capital letter G (U+0047) but a Latin letter small capital G (U+0262).
* A security researcher from NewSky Security [found an impersonated Adobe website](https://blog.newskysecurity.com/fake-adobe-website-delivers-betabot-4114d1775a18) serving the Betabot malware, pretending to be an Adobe Flash Player installer file. The threat actor used the Latin small letter B with Dot below (U+1E05) to replace the Latin small letter B (U+0062) in “adobe.com”.

How is this different from typosquatting?

Although typosquatting also uses visual tricks to deceive users, it relies heavily on users mistyping a URL in the address bar, hence, the “typo” in its name.

Are all homograph attacks just phishing attacks?

Not necessarily. Although homograph attacks usually involve phishing, threat actors could create fake yet believable websites for other fraudulent purposes or to introduce malware onto user systems, as is the case of the bogus Adobe website we mentioned earlier.

In this [in-depth report about IDN homograph attacks](https://www.symantec.com/connect/blogs/bad-guys-using-internationalized-domain-names-idns), our friends at Symantec have noted that several homographed domains they found were either part of a malvertising network, hosting exploit kits and malicious mobile apps, or generated by botnets.

How can we protect ourselves from homograph attacks?

Browser tools have been created, such as Punycode Alert and the Quero Toolbar, to aid users in alerting them of potential homograph attacks. Users have the discretion of adopting them alongside the built-in security mechanisms in today’s browsers. However, no tool can replace vigilance when browsing online and a [solid cybersecurity hygiene](https://blog.malwarebytes.com/101/2017/10/national-cybersecurity-awareness-month-simple-steps-for-online-safety/). This includes:

* Regularly updating your browser (They may be your first line of defense against homograph attacks)
* Confirming that the legitimate site you’re on has an [EVC](https://en.wikipedia.org/wiki/Extended_Validation_Certificate)
* Avoid clicking links from emails, chat messages, and other publicly available content, most especially [social media sites](https://hethical.io/homograph-attack-using-internationalized-domain-name/), without ensuring that the visible link is indeed the true destination.

Remember: Eyes open.

Stay safe!

Additional reading(s):

* [ICANN Statement on IDN Homograph Attacks and Request for Public Comment](https://www.icann.org/news/announcement-2005-02-23-en)
* Unicode Security [Considerations](https://unicode.org/reports/tr36/) and [Mechanisms](https://unicode.org/reports/tr39/)
* [The Homograph Attack [PDF]](https://www.cs.technion.ac.il/~gabr/papers/homograph_full.pdf) by Evgeniy Gabrilovich and Alex Gontmakher

Resource:

* [Punycoder](https://www.punycoder.com/)

# Introduction

The IDN (Internalized Domain Name) homograph attack, also known by the names “homoglyph” and “script spoofing,” is a method in which an attacker deceives victims by making them believe that the site they are visiting is a genuine one.

Attackers exploit this by putting up domains whose names contain more-or-less similar characters resembling the real characters: for example, using a zero instead of an O. Due to lookalike characters, a victim tends to believe they’re visiting the real site and end up giving these fake sites their credit card details, login credentials, and so on.

In a nutshell, attackers are able to register lookalike domain names by exploiting the similar appearance of certain characters in English, Chinese, Latin and Greek or other scripts.

# Leveraging Homograph Attacks

A character is differently viewed by a browser and user. This is due to the fact that computers support multilingual logical characters; hence, it is very easy to make a user get confused.

One example of such attacks is where Cyrillic characters are used. Cyrillic, whose characters resemble certain other letters in the Latin alphabet (for example, the Cyrillic letter which makes the V sound looks just like a Latin B), can easily be used to spoof domain names.

# Generating IDN Homograph Attacks

We can use many online tools to generate such lookalike domains. Most of them create homoglyphs by using lookalike Unicode characters.

MOBILE DEVICE PENETRATION TESTING

# Real-Time Attack Scenario

First, visit this URL: infosecinstitute.com. You will be probably redirected to this site’s homepage.

Now visit this URL: infοѕecinstitute.com. You will be redirected to <http://xn--nfsecnstitute-fpj5fx045a.com/>

Surprised? That’s exactly what attackers do. They simply register a new domain and then make you believe that you are on the real site. The spoof site may then get passwords and other personal details.

# Defending from Homograph Attacks

Most of the defenses against homograph attacks include the display of IDN (internalized domain names) in their Punycode format, thus drastically reducing phishing possibilities. Both Chrome and Firefox have taken adequate measure in their algorithms. ICANN has implemented a policy which prevents registering domains resembling the existing domains.

# Conclusion

Though homograph attacks have reduced now, there still remain endless possibilities for attackers to develop more complex spoofing domains. In the end, it goes down to the user to keep eyes open to any danger in the World Wide Web.

You can read what Google has to say about these attacks [here](http://www.chromium.org/developers/design-documents/idn-in-google-chrome).

Internationalized Domain Names (IDN) in Google Chrome

Background

Many years ago, domains could only consist of the Latin letters A to Z, digits, and a few other characters. [Internationalized Domain Names (IDNs)](https://en.wikipedia.org/wiki/Internationalized_domain_name) were created to better support non-Latin alphabets for web users around the globe.

Different characters from different (or even the same!) languages can look very similar. We’ve seen [reports](https://bugs.chromium.org/p/chromium/issues/detail?id=683314) of proof-of-concept attacks. These are called [homograph attacks](https://en.wikipedia.org/wiki/IDN_homograph_attack). For example, the Latin “a” looks a lot like the Cyrillic “а”, so someone could register http://ebаy.com (using Cyrillic “а”), which could be confused for http://ebay.com. This is a limitation of how URLs are displayed in browsers in general, not a specific bug in Chrome.

In a perfect world, domain registrars would not allow these confusable domain names to be registered. Some domain registrars do exactly that, mostly by restricting the characters allowed, but many do not. To better protect against these attacks, browsers display some domains in [punycode](https://en.wikipedia.org/wiki/Punycode) (looks like xn--...) instead of the original IDN, according to their own IDN policies.

This is a challenging problem space. Chrome has a global user base of billions of people around the world, many of whom are not viewing URLs with Latin letters. 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.

Chrome and other browsers try to balance these needs by implementing IDN policies in a way that allows IDN to be shown for valid domains, but protects against confusable homograph attacks.

Chrome's IDN policy is one of several tools that aim to protect users. [Google Safe Browsing](https://safebrowsing.google.com/) continues to help protect over two billion devices every day by showing warnings to users when they attempt to navigate to dangerous or deceptive sites or download dangerous files. Password managers continue to remember which domain password logins are for, and won’t automatically fill a password into a domain that is not the exactly correct one.

How IDN works

IDNs were devised to support arbitrary Unicode characters in hostnames in a backward-compatible way. This works by having user agents transform hostnames containing non-ASCII Unicode characters into an ASCII-only hostname, which can then be sent on to DNS servers. This is done by encoding each domain label into its punycode representation. This representation includes a four-character prefix (xn--) and then the unicode translated to ASCII Compatible Encoding (ACE). For example, http://öbb.at is transformed to http://xn--bb-eka.at.

Google Chrome's IDN policy

Since Chrome 51, Chrome uses an IDN display policy that does not take into account the language settings (the Accept-Language list) of the browser. A [similar strategy](https://wiki.mozilla.org/IDN_Display_Algorithm#Algorithm) is used by Firefox.

Google Chrome decides if it should show Unicode or punycode for each domain label (component) of a hostname separately. To decide if a component should be shown in Unicode, Google Chrome uses the following algorithm:

1. Convert each component stored in the ACE to Unicode per [UTS 46 transitional processing](http://unicode.org/reports/tr46/#Processing) (ToUnicode).
2. If there is an error in ToUnicode conversion (e.g. contains [disallowed characters](http://unicode.org/cldr/utility/list-unicodeset.jsp?a=%5B%3Auts46%3Ddisallowed%3A%5D&abb=on&g=&i=), [starts with a combining mark](https://unicode-org.github.io/icu-docs/apidoc/released/icu4c/uidna_8h.html#a0411cd49bb5b71852cecd93bcbf0ca2da390a6b3d9844a1dcc1f99fb1ae478ecf), or [violates BiDi rules](https://unicode-org.github.io/icu-docs/apidoc/released/icu4c/uidna_8h.html#a0411cd49bb5b71852cecd93bcbf0ca2da8a9311811fb0f3db1644ac1a88056370)), show punycode.
3. If there is a character in a label not belonging to [Characters allowed in identifiers](http://unicode.org/cldr/utility/list-unicodeset.jsp?a=%5B%3AIdentifierStatus%3DAllowed%3A&abb=on&g=&i=) per [Unicode Technical Standard 39 (UTS 39)](http://www.unicode.org/reports/tr39/#Identifier_Status_and_Type), show punycode.
4. If any character in a label belongs to [the disallowed list](https://unicode.org/cldr/utility/list-unicodeset.jsp?a=%5B%5Cu01CD-%5Cu01DC%5D+%5B%5Cu1c80-%5Cu1c8f%5D++%5B%5Cu1e90-%5Cu1e9b%5D++%5B%5Cu1f00-%5Cu1fff%5D++%5B%5Cua640-%5Cua69f%5D-%5B%5Cua720-%5Cua72f%5D+%5B%5Cu0338+%5Cu058a+%5Cu2010+%5Cu2019+%5Cu2027+%5Cu30a0+%5Cu02bb+%5Cu02bc+%5D&abb=on&g=&i=), show punycode.
5. If the component uses characters drawn from multiple scripts, it is subject to a script mixing check based on [“Highly Restrictive” profile of UTS 39](http://www.unicode.org/reports/tr39/#Restriction_Level_Detection) with an additional restriction on Latin. If the component fails the check, show the component in punycode.

* Latin, Cyrillic or Greek characters cannot be mixed with each other
* Latin characters in the ASCII range can be mixed ONLY with Chinese (Han, Bopomofo), Japanese (Kanji, Katakana, Hiragana), or Korean (Hangul, Hanja)
* Han (CJK Ideographs) can be mixed with Bopomofo
* Han can be mixed with Hiragana and Katakana
* Han can be mixed with Korean Hangul

1. If two or more numbering systems (e.g. European digits + Bengali digits) are mixed, show punycode.
2. If there are any invisible characters (e.g. a sequence of the same combining mark or a sequence of Kana combining marks), show punycode.
3. If there are any characters used in an unusual way, show punycode. E.g. [LATIN MIDDLE DOT (·)](https://unicode.org/cldr/utility/character.jsp?a=00B7) used outside [ela geminada](https://en.wiktionary.org/wiki/ela_geminada).
4. Test the label for [mixed script confusable per UTS 39](http://unicode.org/reports/tr39/#Mixed_Script_Confusables). If mixed script confusable is detected, show punycode.
5. Test the label for [whole script confusables](http://unicode.org/reports/tr39/#Whole_Script_Confusables): If all the letters in a given label belong to a set of whole-script-confusable letters in one of the [whole-script-confusable scripts](https://cs.chromium.org/chromium/src/components/url_formatter/spoof_checks/idn_spoof_checker.cc?type=cs&q=kWholeScriptConfusables&sq=package:chromium) and if the hostname doesn't have a corresponding [allowed top-level-domain](https://cs.chromium.org/chromium/src/components/url_formatter/spoof_checks/idn_spoof_checker.h?type=cs&q=allowed_tlds) for that script, show punycode. **Example for Cyrillic:** The first label in hostname аррӏе.com (xn--80ak6aa92e.com) is all [Cyrillic letters that look like Latin letters](http://unicode.org/cldr/utility/list-unicodeset.jsp?a=%5B%D0%B0%D1%81%D4%81%D0%B5%D2%BB%D1%96%D1%98%D3%8F%D0%BE%D1%80%D4%9B%D1%95%D4%9D%D1%85%D1%83%D1%8A%D0%AC%D2%BD%D0%BF%D0%B3%D1%B5%D1%A1%5D&g=gc&i=) **AND** the TLD (com) is not Cyrillic **AND** the TLD is not one of the TLDs known to host a large number of Cyrillic domains (e.g. ru, su, pyc, ua). Show it in punycode.
6. If the label contains only [digits and digit spoofs](https://cs.chromium.org/chromium/src/components/url_formatter/spoof_checks/idn_spoof_checker.cc?type=cs&q=IsDigitLookalike), show punycode.
7. If the label matches a [dangerous pattern](https://cs.chromium.org/chromium/src/components/url_formatter/spoof_checks/idn_spoof_checker.cc?type=cs&g=0&l=422), show punycode.
8. If the [skeleton](http://unicode.org/reports/tr39/#def-skeleton) of the registrable part of a hostname is identical to one of the top domains after removing diacritic marks and mapping each character to its spoofing skeleton (e.g. www.googlé.com with é in place of e), show punycode.

Otherwise, show Unicode.

This is implemented by IDNToUnicodeOneComponent() and IsIDNComponentSafe() in [components/url\_formatter/url\_formatter.cc](https://cs.chromium.org/search/?q=components/url_formatter/url_formatter.cc) and IDNSpoofChecker class in [components/url\_formatter/spoof\_checks/idn\_spoof\_checker.cc](https://cs.chromium.org/chromium/src/components/url_formatter/spoof_checks/idn_spoof_checker.cc).

Additional Protections

In addition to the spoof checks above, Chrome also implements a full page security warning to protect against lookalike URLs. You can find an example of this warning at chrome://interstitials/lookalike. This warning blocks main frame navigations that involve lookalike URLs, either as a direct navigation or as part of a redirect.

The algorithm to show this warning is as follows:

1. If the scheme of the navigation is not http or https, allow the navigation.
2. If the navigation is a redirect, check the redirect chain. If the redirect chain is safe, allow the navigation. (See Defensive Registrations section for details).
3. If the hostname of the navigation has at least a medium site engagement score, allow the navigation. Site engagement score is assigned to sites by the [Site Engagement Service](https://www.chromium.org/developers/design-documents/site-engagement).
4. If the hostname of the navigation is in [domains.list](https://cs.chromium.org/chromium/src/components/url_formatter/spoof_checks/top_domains/domains.list), allow the navigation.
5. If the user previously allowed the hostname of the navigation by clicking “Ignore” in the warning, allow the navigation. Currently, user decisions are stored per tab, so navigating to the same site in a new tab may show the warning.
6. If the hostname has the same skeleton as a recently engaged site or a top 500 domain, block the navigation and show the warning.

All of these checks are done locally on the client side.

Defensive Registrations

Domain owners can sometimes register multiple versions of their domains, such as the ASCII and IDN versions, to improve user experience and prevent potential spoofs. We call these supplementary domains defensive registrations.

In some cases, Chrome's lookalike warning may flag and block navigations to these domains:

* If one of the sites is in domains.list but the other isn't, the latter will be blocked.
* If the user engaged with one of the sites but not the other, the latter will be blocked.

Avoiding a lookalike warning on your site

**Domain owners can avoid the “Did you mean” warning by redirecting their defensive registrations to their canonical domain.**

**Example**: If you own both example.com and éxample.com and the majority of your traffic is to example.com, you can fix the warning by redirecting éxample.com to example.com. The lookalike warning logic considers this a safe redirect and allows the navigation. If you must also redirect http navigations to https, do this in a single redirect such as http://éxample.com -> https://example.com. Use HTTP 301 or HTTP 302 redirects, the lookalike warning ignores meta redirects.

Reporting Security Bugs

We reward certain cases of IDN spoofs according to [Chrome's Vulnerability Reward Program](https://www.google.com/about/appsecurity/chrome-rewards/index.html) policies. Please see [this document](https://docs.google.com/document/d/1_xJz3J9kkAPwk3pma6K3X12SyPTyyaJDSCxTfF8Y5sU/edit?usp=sharing) before reporting a security bug.

**Homograph Attack**

A [homograph](http://www.crypto-it.net/eng/attacks/homograph-attack.html) attack is based on standards of modern Internet that allow to create (and display in web browsers) URLs with characters from various language sets (with non-ASCII letters). Different languages may contain different but very similar characters. Attackers can register their own domain names that are similar to the existing web addresses. Then they can create their own websites that are, again, the same or very similar to the existing original sites (that usually belong to banks, corporations, email or news services). The phony websites are used for stealing data from users who happened to visit them.

**Simple homograph attacks**

In the simplest version of such attacks, a fake URL may consist only of simple ASCII alphanumeric characters. The intruder uses symbols that are similar to each other. Often the letter **q** may be confused with **g**, or **o** with **0**.

Such URLs may fool some less experienced users:

http://www.g00gle.co.uk  
http://bl00mberg.com

**Non-ASCII ULRs**

The ability to use non-English characters in ULR addresses was added in [2003](http://www.crypto-it.net/eng/attacks/homograph-attack.html), due to the increasing number of [non-English-speaking people](http://www.crypto-it.net/eng/attacks/homograph-attack.html) that were using Internet. The change allowed to register and use domain names that could have been understood by a much larger number of interested people. Thus it became possible to create web addresses that were combinations of ASCII and non-ASCII characters, or addresses that consisted only of national symbols:

http://россия.net  
http://газета.ру  
http://budyń.pl

All non-Latin addresses need to be encoded in a special way to be handled by DNS servers. This format is known as **Punycode** and all browsers translate non-ASCII URLs into Punycode in the background before performing a DNS lookup. A Punycode domain name always starts from xn-- and then contains ASCII characters of the original address followed by encoded Unicode data. For instance, the latter address from the example above will be encoded in the following form:

http://xn--budy-e2a.pl

Such domain names that contain letters from different alphabets are called *Internationalized Domain Names* (IDNs). They are handled in various ways by different web browsers. Usually every producer implements his own algorithms for determining the display format of requested URLs and usually one of two solutions (with some minor modifications) is preferred:

* Display all URL characters using Unicode, or
* Display all URL characters using Unicode if and only if all the characters belong to the same language that is chosen by user settings; display Punycode URL otherwise.

**Homograph attacks using non-ASCII characters**

Different languages with characters encoded in a different way, may contain some letters that look the same or at least very similar. Therefore it is possible to create URLs that consist of different characters but are indistinguishable to the human eye.

For example, Latin and Cyrillic alphabets contain a couple of letters that look the same but have completely different meaning and are encoded in a different way:

* (in Latin) **a**: U+0061, (in Cyrillic) **а**: U+0430
* (in Latin) **c**: U+0063, (in Cyrillic) **с**: U+0441
* (in Latin) **p**: U+0070, (in Cyrillic) **р**: U+0440

Given 100 000 characters supported by Unicode (many of which look alike), the intruder has great potential for creating various fake URLs and even the most careful users may be confused. At present neither DNS registrars, nor web browser vendors managed to prevent such attacks from happening.

Finally, it may be shown that an attacker can use a character that happens to look like the actual ASCII slash **/** (U+002F) - the mathematical division operator **∕** (U+2215). It allows him to set up a subdomain that looks like another real domain, using his own name server and a top-level domain. The fake URL address could look like the one below:

http://example.com∕a-top-level-domain.com/

The character located after .com is a mathematical division operator. In a web browser's address bar the character would look like a common slash and the whole URL could be easily confused with the directory a-top-level-domain.com located in the root directory under the domain example.com:

http://example.com/a-top-level-domain.com/

**Security**

The best protection against homograph attacks seems to be provided by warning or proper handlings such phony addresses by web browsers. Unfortunately this is not always the case and, what is more, the behaviour may differ depending on browser vendor.