You Are Who You Appear to Be

A Longitudinal Study of Domain Impersonation in TLS Certificates

Richard Roberts
University of Maryland

Yaelle Goldschlag
University of Maryland

Rachel Walter
University of Maryland

Taejoong Chung
Rochester Institute of Technology

Alan Mislove
Northeastern University

Dave Levin
University of Maryland

ABSTRACT

The public key infrastructure (PKI) provides the fundamental property of *authentication*: the means by which users can know with whom they are communicating online. The PKI ensures end-to-end authenticity insofar as it verifies a chain of certificates, but the *true* final step in end-to-end authentication comes when the *user* verifies that the website is what they expect. To this end, users are expected to evaluate domain names, but various "domain impersonation" attacks threaten their ability to do so. Indeed, if a user could be easily tricked into believing that `amazon.com-offers.com` is actually `amazon.com`, then, coupled with security indicators like a lock icon, users could believe that they have a secure connection to Amazon.

We study this threat to end-to-end authentication: (1) We introduce a new classification of an impersonation attack that we call target embedding. This embeds an entire target domain, unmodified, using one or more subdomains of the actual domain. (2) We perform a user study with the specific goal of understanding whether users fall for target embedding, and how its efficacy compares to other popular impersonation attacks (typosquatting, combosquatting, and homographs). We find that target embedding is the most effective against modern browsers. (3) Using all HTTPS certificates collected by Censys, we perform a longitudinal analysis of how target-embedding impersonation has evolved, who is responsible for issuing impersonating certificates, who hosts the domains, where the economic choke-points are, and more. We close with a discussion of counter-measures against this growing threat.

CCS CONCEPTS

• **Security and privacy → Spoofing attacks**; *Web protocol security*; *Economics of security and privacy*.

KEYWORDS

PKI; TLS; Domain impersonation; Target embedding

CCS '19, November 11–15, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6747-9/19/11…$15.00
https://doi.org/10.1145/3319535.3363188

ACM Reference Format:
Richard Roberts, Yaelle Goldschlag, Rachel Walter, Taejoong Chung, Alan Mislove, and Dave Levin. 2019. You Are Who You Appear to Be: A Longitudinal Study of Domain Impersonation in TLS Certificates. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3319535.3363188

1 INTRODUCTION

The public key infrastructure (PKI) solves a fundamental problem of online communication: it provides mechanisms by which users verify with whom they are communicating. The PKI provides end-to-end authentication in the sense that it ensures that a user's machine is able to verify the signatures in a website's certificate chain, and to check that the certificates have not been revoked [29].

However, the *true* final step in end-to-end authentication comes when the *user* ascertains who is on the other side of the connection. To this end, users have two readily available pieces of information: *First*, users are commonly presented with a security indicator, such as the well-known "lock icon" (or conversely a "Not Secure" tag) when browsing the web. Indicators such as these denote that the browser was able to successfully authenticate a website's certificate, but they do not represent whether the website is what the user *expects* it to be. *Second*, browsers present users with the website's domain name itself.

Of these, the security community has encouraged users to look for security indicators like the lock icon when sharing private information online [11]. It is well known through user studies [15, 37], however, that users misinterpret the meaning behind such security indicators and equate them not only with the authenticity of the connection, but with the trustworthiness of the site.

But to our knowledge, there has been little study on the equally critical question of whether users are effective at properly evaluating *domain names themselves*. For instance, if users could be easily tricked into believing that `amazon.com-offers.com` is actually `amazon.com`, then, coupled with the lock icon, they could believe that they have a secure ("trustworthy") connection to Amazon.

There has been a large body of work proposing and studying various forms of *domain impersonation* attacks—such as typosquatting (e.g., `youtueb.com`) [5, 32, 41, 45], homographs (e.g., `y0utube.com`) [13, 16, 17, 20, 25], and combosquatting (e.g., `youtube-videos.com`) [22]. Through various wide-scale and longitudinal studies, researchers have found that attackers appear to be using these forms of impersonation in the wild. However, we are unaware of any work performing a user-focused study of the relative successes of these attacks. Are they effective, and are there other forms of impersonation that may be even more effective?

Are these problems made more acute within the PKI, and are there players in the PKI who may be able to help mitigate them?

In this paper, we make three broad contributions:

A new classification of domain impersonation. We propose a new classification of domain impersonation which we call *target embedding*. Unlike prior schemes that alter a domain in some way, target embedded leaves the impersonated domain *unmodified*—it does this by embedding the domain by using a subdomain of the actual domain. For example, `apple.com-signin.id` embeds the target domain `apple.com`, but the actual domain is `com-signin.id`. This is a real domain: the popular certificate authority (CA) Let's Encrypt gave a certificate to this domain in October, 2018. Target embedding is differentiated from attacks such as homographs, typosquatting, and combosquatting, as it is a form of "subdomain spoofing," or an impersonation attack that is located in subdomains instead of an effective second-level domain (e2LD).

A user study of susceptibility to domain impersonation. We performed a user study of 244 users with a relatively narrow goal: to understand how thoroughly users fall for target embedding, as compared with other popular domain impersonation attacks (typosquatting, combosquatting, and homographs). Our results show that users are significantly more susceptible to target embedding than the other attacks that threaten today's browsers.

A wide-scale longitudinal study of target embedding. The bulk of our investigation analyzes all HTTPS certificates collected by Censys to understand how target embedding happens—which domains are targeted, who gives them certificates, where are they hosted, and so on—and ways to remedy it.

We summarize our longitudinal study's key findings as follows:

- Target embedding is widespread. We observe 256,045 impersonation attempts spanning 112,262 distinct domains and 7,581 distinct targets from within the Alexa top-100K most popular websites. While impersonators preferentially target more popular websites, we see a long tail of impersonation attempts.
- Impersonators take advantage of the low economic barrier of entry to impersonation attacks by using free domain registration, free certificate issuance, and free hosting.
- Three CAs were responsible for issuing 95.37% of certificates that included a target embedding domain (but issued only 80.80% of all certificates in our dataset).
- Reactive initiatives such as Google Safe Browsing [18] miss thousands of domains from large, coordinated campaigns.
- Wildcard certificates pose a large potential threat. We identify 343,336 unique wildcard domains whose wildcard is immediately followed by a TLD from an impersonated target (e.g., `*.com-deals.online`).

Roadmap §2 presents background and related work. §3 provides the results of a user study investigating how successful domain impersonation attacks are at deceiving users. §4 describes our datasets and methodology for detecting target embedding attacks, and compare target embedding to other common forms of domain impersonation. §5 presents our longitudinal study of target embedding. §6 studies how successful Google Safe Browsing is at detecting large coordinated impersonation campaigns. We present possible solutions to target embedding in §7 and conclude in §8. All of our code and data are publicly available at `https://securepki.org`

2 BACKGROUND AND RELATED WORK

Domain Impersonation Attacks A wide range of domain impersonation attacks have been identified. These include: *typosquatting*, in which the impersonating domain has a small edit distance from the target domain (faceboook.com) [5, 32, 41, 45]; *bitsquatting*, in which a bit in the ASCII representation is flipped (fagebook.com) [33]; *combosquatting*, in which the attacker includes a target's brand name alongside other string tokens (facebook-login.com) [22], *homographs*, that use "confusable" characters—often Unicode characters used in Internationalized Domain Names (IDNs) [10]—that look like the real characters (faceb00k.com) [13, 16, 17, 20, 25]; and *homophones*, domains that sound the same as a target domain when read aloud (fasebook.com) [34]. All of these impersonation attacks occur in the effective second-level domain, or e2LD (e.g., `example` in `example.com`). We introduce the umbrella term "e2LD spoofing" to describe attacks that generate a domain with a new e2LD, which impersonates a similar looking e2LD.

We expand on this work by introducing a type of impersonation that we call *target embedding*. Simply put, a target embedding domain *embeds* a complete, unmodified *target* domain, including the TLD, by using one or more subdomains of the *real* domain. The target domain is separated from the rest of the domain on the right (and optionally on the left) by either a period (.) or a hyphen (-). For example, consider the target embedding domain "www.facebook.com.user-29de84ca4bfa72.tk". The target, in this case "facebook.com", is embedded using subdomains of the actual domain, "user-29de84ca4bfa72.tk". The target's TLD can also appear in the real e2LD, such as `apple.com-login.pw`. Unlike prior domain impersonation attacks, target embedding does not operate strictly within the e2LD: in fact, it *requires* the use of at least one subdomain, as all target domains have at least one period between their e2LD and TLD.

Target embedding is part of a broader class of attacks known as "subdomain spoofing." Subdomain spoofing has been mentioned only in passing in academic literature [13, 23] and lacks a concrete definition. We define subdomain spoofing as an umbrella term that includes any attempt at domain impersonation where the *target of impersonation* is primarily contained in one or more subdomains. Snowshoe spamming [40] is a form of subdomain spoofing that prepends a target's e2LD as a subdomain on multiple domains to evade reputation filters (e.g., `LinkedIn.foo1.com`, `LinkedIn.foo2.com`, etc.). Other forms of impersonation such as URL padding can use long subdomains or e2LDs to force portions of a domain not to be rendered on a user's screen [13, 19, 23]. To the best of our knowledge, our paper is the first to empirically measure any form of subdomain spoofing at scale.

User Perception and Comprehension of URLs Much work has been conducted on drawing user attention to URLs in order to help determine a website's legitimacy. Several studies have found that some users do look at the address bar for evaluation without prompting, through interviews, think-aloud protocols, eye tracking measurements, and click-heatmaps [21, 27, 42]. In cases where users did not look at the address bar on their own, education attempts

to explicitly make them consider the address bar when evaluating a website's legitimacy have proven effective [27, 47], unlike UI changes created to indirectly alert users to the presence of a suspicious domain [42, 46, 47]. However, in all these cases, even engaged users struggled with actually identifying when a domain was spoofing a target, as users are generally unaware of modern impersonation techniques. Kumaraguru et al. developed education tools designed to teach users about impersonation techniques themselves [24]. We complement these works with a user study that measures how effective different types of domain impersonation are at confusing users.

3 DO USERS FALL FOR TARGET EMBEDDING?

To motivate our study of target embedding, we performed a user study with a solitary goal: to understand how thoroughly users fall for target embedding, as compared to other popular domain impersonation attacks (typosquatting, combosquatting, and homographs).

3.1 Study design

We designed and ran a user survey (N=244) on Amazon MTurk. After brief instructions in which we explained what a URL was, participants were presented with 48 questions. Each question presented a (possibly impersonating) URL and the name of an organization, and the participant was instructed to answer "Yes" or "No" to the question: *Do you believe that this is the organization's URL?*

Posing the question in this manner is *intentionally* unlike what users tend to experience. First, it raises their suspicion to a level higher than users are likely to experience while normally browsing [6]. Second, it draws the users' attention to the address bar (that is all our survey shows them), which browsers currently do a poor job at [42]; despite the fact that the URL is the one true indicator of a website's identity, users often make their trust decisions based on the page's content, which is easy to replicate. We believe our results capture a lower bound of users' susceptibility to domain impersonation attacks, as these two departures from reality decrease the likelihood that users will be fooled.

We tested users' ability to detect four types of domain impersonation: target embedding, combosquatting, typosquatting, and Unicode homographs. Participants were given six of the above Yes/No questions for each of these. Another six questions included the Unicode homographs converted to Punycode, an ASCII encoding of Unicode that modern browsers display to mitigate homograph attacks. For example, the homograph eβay.com in Punycode is xn–eay-sxc.com. Appendix A lists all of the domain names we used in our user study.

The remaining URLs were controls and related to measurements not discussed in this paper; more information on these, our recruitment method, compensation, ethical considerations, participant demographics, and survey protocol can be found in Appendix A. Our user study received IRB approval, and we received informed consent from all participants.

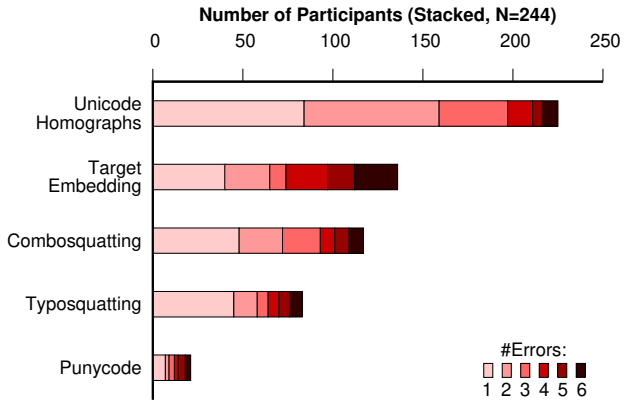

Figure 1: N=244 participants were shown 6 questions each for target embedding, combosquatting, typosquatting, and Unicode homographs. Participants were also shown the 6 homograph URLs rendered as Punycode, a common defense against that attack. The number of errors the participants committed are presented as stacked histograms.

3.2 Results

Figure 1 shows the overall responses from our survey. The total height of each bar represents the number of participants that answered at least one question in the respective category incorrectly: a taller bar represents more participants falling for an impersonation attack (at least once). Colors within each bar bin participants by how many of that category's questions the user answered incorrectly: a darker bar represents individual participants falling for an impersonation attack more often. When comparing two impersonation attacks, we use chi-square tests and report p-values, corrected with the Bonferroni method. We make several observations:

Target embedding evokes more repeated mistakes The participants collectively made 428 mistakes when classifying target embedding domain names. This is second only to homograph's 480, but this difference is not statistically significant ($p = 0.17$). Target embedding leads to more mistakes than combosquatting's 279, typosquatting's 185, and Punycode's 66 ($p < 0.001$ for each).

However, target embedding does not always have many more *users* who fall for it. A total of 136 participants fell for at least one target embedding attack. Target embedding was less effective than homograph's 225 participants, but far more effective than typosquatting's 83 and Punycode's 21 ($p < 0.001$ for all of these). Target embedding was not statistically significantly different from combosquatting, which 117 participants fell for ($p = 0.41$).

Punycode mitigates Unicode homograph attacks In an effort to mitigate homograph attacks, modern browsers convert domain names that contain Unicode (e.g., `ápple.com`) to Punycode (equivalently, `xn--pple-4na.com`) [36]. Our results show that this is an extremely effective defense: our participants were significantly less likely to fall for Punycoded domains than for any other impersonation attack ($p < 0.001$).

Target embedding is currently the greatest threat Taken together, these results show that target embedding leads to significantly more user mistakes than any other impersonation attack currently possible in modern browsers (thus excluding homographs). Moreover, the results show that if a user falls for a target embedding attack once, they are likely to fall for it multiple times—more so than with other domain impersonation attacks.

Summarized simply: to users, *domain names are who they appear to be*, and target embedding is currently the most effective means of appearing to be someone a domain is not.

3.3 Implications on the Web's PKI

Our survey results highlight why domain impersonation poses a major threat to the web's PKI. The fundamental role of the PKI is to vet websites' identities. The PKI and users both use domain names to represent identity. Since the PKI is largely automated, it correctly differentiates between `google.com-signin.com` and `google.com`—and expects users to be able to do the same. To many users, however, `google.com-signin.com` *is* `google.com`.

If an attacker can obtain a certificate for a domain d that appears to be another domain d', then in the eyes of users, the attacker is *effectively obtaining a certificate for a target website they do not own*. In other words, we view this form of domain impersonation as a *protocol-compliant* attack on the fundamental role of the PKI.

Motivated by this conflict, we now measure the prevalence of domain impersonation in the web's PKI.

4 WIDE-SCALE ANALYSIS METHODOLOGY

In the remainder of this paper we evaluate how prevalent target-embedded certificates are in the web today, who is doing the targeting, and who is being targeted. We start by introducing the datasets we use and our approach to identify target embedded domains.

Nomenclature Before presenting our methodology, we briefly overview the nomenclature used in the remainder of the paper. Consider the domain `appleid.apple.com-login.pw` (a real target embedding domain we observed in the wild), which we refer to as the *fully-qualified domain name* (FQDN). We refer to `apple.com` as the *target domain* and `com-login.pw` as the *actual domain* (where the "domain" refers to the effective 2nd level domain plus suffix).

4.1 Certificate dataset

Our primary dataset comprises all certificates collected by Censys [12] up to May 18, 2019. Censys's dataset includes a combination of active scans (they scan all IPv4 addresses and popular TLDs' zone files) and Certificate Transparency (CT) logs. VanderSloot et al. estimate that this combined dataset captures over 99% of observed certificates [44]. Accordingly, we believe this to be a highly accurate representation of all certificates on the web.

From this set, we obtain a total of 1,499,347,402 certificates, containing a total of 529,515,677 unique FQDNs. We use this dataset to evaluate the prevalence of target-embedding, combosquatting, and typosquatting domains that appear on TLS certificates.

4.2 Identifying target embedding

We now describe our methodology for identifying target embedding domains, and for filtering out false positives.

Step 1: Exact match Target embedding involves an *exact match* of the target by *using a subdomain*, followed by a dot (for instance, `apple.com.ilogin.email`) or a dash (`amazon.com-buy.site`) and preceded by nothing, a dot (`www.ebay.com--login.com`), or a dash (`secure-paypal.com.tatpk.ru`). This permits a straightforward initial filter: given a set of targets, one need only perform a simple regular expression (`[-\.]?t\.tld[-\.]`) for each target `t.tld`. Applied to our dataset, this step resulted in 468,184 unique FQDNs.

This initial exact-match pass results in domains that have a target domain t and an actual domain a. However, not all of these are necessarily impersonation. For instance, embedding should be permitted when the target is owned by the actual domain (e.g., `imdb.com.amazon.com`) or when the target is *identical* to the actual domain (e.g., `google.com.google.com`).

Step 2: Filter target ownership In the next step, we filter out all FQDNs for which it could reasonably be proved to a CA that the actual domain a has ownership or control over the target t. To infer *ownership*, we apply the same techniques as Cangialosi et al. [9] to determine if two domains are managed by the same entity. This involves obtaining WHOIS data for all t and a, extracting the administrator email addresses from the WHOIS records, filtering out privacy-preserving email addresses, and comparing the domains in the email addresses. The 468,184 FQDNs from step 1 contained 131,218 unique (t, a) pairs. We successfully obtained WHOIS data on 66,281 of these pairs, and used these to filter out 3,349 FQDNs.

Automated CAs like Let's Encrypt do not require ownership of domains to obtain a certificate; it suffices to demonstrate control over the domain's name server. To infer whether a has *control* over t, we compare the authoritative DNS name servers for both a and t, and filter out the FQDN if the name servers are "equivalent." We take a liberal approach: we consider them to be equivalent if even one of their name servers shares the same e2LD (e.g., `ns1.example.com` and `ns2.example.com`), or if they both have a name server including the substring `awsdns` (e.g., `ns-750.awsdns-29.net` and `ns-510.awsdns-63.com`). In so doing, we are likely obtaining a *lower bound* on the number of target embeddings. Following this process, we filtered out another 47,247 FQDNs. We note that an automated CA would not have to make the same approximations that we are: they could easily extend their ACME challenges to actively prove control over both a and t (and any other targets included in the domain).

The remaining FQDNs cannot be ruled out by the automated mechanisms that many popular CAs take today. The CA/Browser baseline requirements for issuance [8] would define the remaining FQDNs as *High Risk Certificate Requests*, as they "may include names at higher risk for phishing or other fraudulent usage." It further requires CAs to perform *"additional verification activity for High Risk Certificate Requests prior to the Certificate's approval"*. In an effort to emulate what CAs would be required to do as a means of additional verification, we perform two additional steps:

Step 3: Filter common subdomains Some of the Alexa top-100K websites have e2LDs that are also common subdomains. Consider, for example, if `www.com` were a top Alexa domain: this would mean that *every* domain starting with `com-` would be considered target embedding if it used the common subdomain `www`. We identified 20 domains in the Alexa top-100K whose e2LDs are also popular subdomains. These include `cpanel.com`, `mail.com`, and `mail.ru`. The popular cPanel web administration tool automatically adds the subdomains `cpanel` and `mail` to websites that it manages. We filter out all FQDNs that *begin* with these target domains, as they are likely being used in conjunction with cPanel software. This, along with the entire set of 20 target domains, filtered out an additional 24,099 FQDNs.

Step 4: Filter out web hosting providers Finally, we must account for the fact that some websites *delegate* certificate management to third parties [9]. Cangialosi et al. [9] showed that popular content delivery networks (CDNs) manage their customers' certificates and often even generate their public-private key pairs. We follow their methodology in identifying which actual domains a are likely the hosting providers for the targets t: in particular, like them, we identify the 100 most common a's (both by unique FQDNs and unique targets) and manually verify that they are services that are likely in a business relationship with one another. We note that CAs *already* perform this manual processing step: they establish business relationships with popular providers to allow them to purchase bulk certificates on others' behalves. This filters another 137,625 FQDNs.

Final dataset After the above filtering, we obtained 256,045 unique FQDNs, comprising 112,262 unique actual domains, 7,581 unique target domains, and spanning 435,717 certificates.

Ethical considerations None of this data collection involved human subjects (excluding the user study in §3), nor did it involve active probing of the web servers themselves (only queries to authoritatively resolve their name servers). We conformed to the terms and services of all of the services we used.

4.3 Comparing to prior impersonation schemes

We compare target embedding with typosquatting and combosquatting by replicating prior detection techniques on our certificate dataset:

Typosquatting We use the same methodology for identifying typosquatting as used by Agten et al. [5]. They define typosquatting as one of five mutations: add a character, delete a character, swap two adjacent characters, fat-finger replace one character, or remove the dot on a "`www.`" subdomain.

Combosquatting We follow the same methodology as Kintis et al. [22] for detecting combosquatting. Given a set of target domains, combosquatting involves checking whether the target's e2LD (e.g., "`example`" in `example.com`) is a *strict* substring of the domain in question's e2LD. For example, `youtubevideos.com` and `watch-youtube.ru` are both examples of combosquatting with `youtube.com` as the target. By definition, a domain that can be considered typosquatting is not combosquatting. Filtering the set of applicable target domains is a challenge when applying combosquatting; we explain our method next.

Target domains Unfortunately, typosquatting and combosquatting are limited in the set of target domains to which they can be applied. To bound the number of false positives, prior typosquatting work has limited analysis to target domains of at least five characters [32] and limited the number of targets to the 500–10,000 most popular websites [5, 32, 41]. Similarly, prior combosquatting work has limited their study to the Alexa top-500 most popular domains. Worse yet, detecting combosquatting requires ignoring target domains whose brands are substrings of common English words, such as `apple.com`, `att.com` (because of words like "**att**orney"), `citi.com` ("**citi**es"), and so on [22].

To perform a fair apples-to-apples comparison, we replicate the procedure used by Kintis et al. [22]. Unfortunately, several key details are elided from their paper; we describe here our good-faith effort to replicate them. Like them, we begin with the Alexa top-500 most popular sites, and remove all e2LDs of length less than four (their paper did not report on any targets of that size). We use the standard Linux dictionary in `/usr/share/dict/words` to remove all targets whose e2LDs are equal to or substrings of the dictionary's 102,305 common English words. We then add back domains that are in the Linux dictionary but also reported in their paper: `google`, `amazon`, and `yahoo`. All together, this results in 320 e2LDs, which correspond to 407 Alexa top-500 domains[1] (some domains have the same e2LD but different TLDs, such as `google.com` and `google.co.uk`). We use these 320 target e2LDs to determine whether a domain is typosquatting and combosquatting.

Fortunately, target embedding is not subject to the same limitations. In our analysis, we use the *entire* Alexa top-100K most popular websites: 1–3 orders of magnitude more target domains than could be studied in previous impersonation work [5, 7, 17, 20, 22, 32, 41, 43, 45]. As we will demonstrate, there are impersonating domains well into the least-popular websites, which other impersonation analyses could not detect. However, for this head-to-head comparison with typosquatting and combosquatting, we limit it to the 407 target domains with the same 320 e2LDs used in our typosquatting and combosquatting analysis. Also, to yield a fair comparison, we do *not* perform our additional filtering steps (after step 1) for this smaller set of domains.

Google Safe Browsing To analyze the extent to which various impersonation attacks are correlated with malicious activity, we run all of domains we identify to have performed typosquatting, combosquatting, or target embedding through Google Safe Browsing [18]. Google Safe Browsing provides a binary classification—safe or not—based on a combination of analysis by Google and user reports. When loaded in Chrome, domains flagged by Google Safe Browsing provide red-screen warnings to users. Various prior studies downloaded website content and performed their own classification of content into benign, malicious, or phishing, either manually [22] or through custom machine learning techniques [43]. We chose to instead rely on Safe Browsing's data because it permits more repeatable results, it is more scalable, and it can be applied to websites that are no longer live. This last point is particularly important, as phishing domains are typically live for only a few days [30], yet our datasets spans years.

[1] This list, along with all of our code and data, is publicly available at `https://securepki.org`

Impersonation type (# targets)	FQDNs	Flagged by Safe Browsing	
Typosquatting (407)	225,985	1,635	(0.72%)
Combosquatting (407)	1,134,106	14,801	(1.31%)
Target Embedding (407)	125,199	7,719	(6.17%)
Target Embedding (100K)	256,045	27,206	(10.63%)

Table 1: Comparison of target embedding to prior impersonation schemes. Target embedding is much more strongly correlated with unsafe domains, and scales to handle orders of magnitude more targets.

One shortcoming of Google Safe Browsing is that, although it has broad coverage, it has not necessarily classified all of the domains that we have identified. There are two reasons for this: first, it is possible that the domains appear on certificates but a corresponding website never went live. Second, it is possible that the domains were live, but that users never reported them. Unfortunately, Google Safe Browsing merely returns whether it has found the website to be unsafe, and does not note whether it has any data on the site. Thus, we cannot rely on Safe Browsing data for full coverage, but we can still use it to compare how strongly the various impersonation attacks correlate with unsafe sites.

Comparison results We present our comparison results in Table 1, from which we make two key observations. *First*, target embedding is much more strongly correlated with unsafe domains, as determined by Google Safe Browsing. When limited to the same 407 target domains as typosquatting and combosquatting, target embedding has a 8.6× and 4.7× higher ratio of unsafe domains. The higher apparent false positive rates of prior schemes is in line with previous studies, as well as our user study in §3: there is simply much more noise in typosquatting and combosquatting, which complicates detection *and* deception. Conversely, target embedding must contain the *entire unaltered target domain*. This offers a cleaner signal of intent and, as our user study showed, a more accurate means of deception. *Second*, target embedding is able to scale to a much larger set of target domains, and in so doing, is able to identify many more unsafe domains than the previous schemes. When applying target embedding to the Alexa top-100K most popular websites, we found 16.6× more impersonating domains than typosquatting and 1.8× more than combosquatting—and with 14.8× and 8.1× higher fraction of these domains being unsafe.

Taken together, these results show that target embedding is worthy of study as a unique and effective means of confusing users. It is more strongly correlated with unsafe webpages and can scale to more target domains, and thus it is a more effective lens than prior schemes to study impersonation within the web's PKI. In the remainder of this paper, we perform a thorough, longitudinal analysis of target embedding.

5 LONGITUDINAL STUDY OF TARGET EMBEDDING

We now examine the use, cause, and risk of target embedding in the web's PKI.

Domain	Unique Actual Domains	Alexa Rank
apple.com	69,362	77
paypal.com	28,449	78
icloud.com	14,911	408
runescape.com	7.135	1,822
facebook.com	6,179	3
google.com	4,572	1
naver.com	3,107	263
amazon.com	3,084	12
starwars.com	3,076	24,867
ebay.com	2,825	36
163.com	2,528	738
live.com	1,680	19
mail.ru	1,575	44
bankofamerica.com	1,513	305
ebay.co.uk	1,490	149
chase.com	1,444	183
americanexpress.com	1,146	523
tripadvisor.com	1,091	227
banorte.com	1,027	17,467
amazon.de	997	83
Others	100,920	

Table 2: Most commonly targeted domains, by count of unique FQDNs embedding the target domain.

5.1 Who is being targeted?

We begin by investigating which domains are targets for target embedding. This is important for understanding the motivations behind and ramifications of embedding targets in domains. We note that prior approaches have focused on small, hand-picked sets of potential targets—often on the order of hundreds. Conversely, we study targeting across the Alexa top-100K most popular websites (see Section 4).

In total, we observe 256,045 instances of target embedding, coming from 112,262 distinct actual domains and covering 7,581 distinct target domains from within the Alexa top-100K. Table 2 shows the most popular target domains, determined by the number of distinct FQDNs which embed them. We make several key observations:

First, some of the most targeted domains are relatively unpopular. While attackers have a preference for more popular domains, 5 of the top 20 most targeted domains have an Alexa ranking over 500. We are unaware of any prior study of domain impersonations to include, runescape.com, even though it is the fourth most targeted domain. These results show the importance of studying targets well beyond a small, hand-picked set of domains.

Second, many of the most targeted sites exhibit a clear economic incentive for an attacker. Obtaining login credentials for apple.com, paypal.com, ebay.com, or the various banking websites (chase.com, bankofamerica.com, banorte.com) can allow an attacker to make purchases in the victim's name (or potentially steal the victim's funds directly). Similarly, runescape.com allows for in-game purchases and trade, making its users' login credentials a valuable asset. We observe social networking and email services as targets, including facebook.com, google.com (e.g., for access to Gmail), mail.ru, and live.com. Such targets can be valuable pivot points for subsequent attacks against a user and the users' friends and contacts. We also observe storage services such as icloud.com; these often

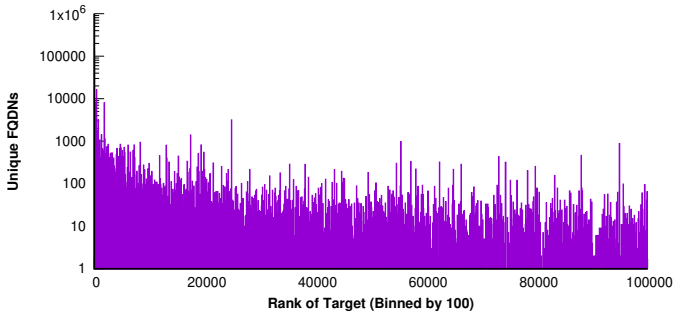

Figure 2: Number of unique target embedding domains as a function of Alexa rank (binned by 100). The long tail indicates that many domains were targeted a small number of times.

contain sensitive data. Finally, we see a large coordinated effort to target starwars.com (Alexa rank 24,867); as we discuss in Section 6, we believe this to be a campaign aiming at many targets, but for unknown reasons starwars.com is the only fully embedded target domain in this campaign.

Third, we observe a long-tail distribution in the frequency at which domains are targeted. This is shown in more detail in Figure 2, which plots the number of unique target embedding domains as a function of their targets' Alexa rankings. While the bulk of the distribution is at the head—the top 100 Alexa domains constitute 51.3% (131,416) of the unique target embedding domains—the tail extends throughout the entire range of domains we considered. This still leaves a considerable amount of target embedding from the long-tail: a nontrivial 14,527 (5.6%) domains targeted a website with an Alexa ranking over 50,000. 1,760 (23.2%) targets were targeted in only one FQDN, collectively constituting 0.7% of all target embedding attacks.

Summary Collectively, these results show that attackers are targeting a wide range of websites. Efforts to study domain impersonation must be equally broad; limiting study to, say, financial institutions or only the most popular sites, would miss a large fraction of potential attacks.

5.2 Who is doing the targeting?

Here we investigate properties of domains that are targeting others: how do attackers obtain the actual domains, and how much do they represent common domain names?

Table 3 shows the most commonly used actual TLDs in observed target-embedding domains. The table also includes the rank of how often each TLD appears in the Alexa top-1M, and the rank of how often each TLD appears in *any* domain on a certificate from the Censys dataset. Interestingly, the ranking of the TLDs where target embedding is observed is much more strongly correlated with the certificate ranking than with Alexa ranking. We make observations about two key trends:

First, several highly unpopular TLDs according to Alexa—.ga, .ml, .cf, .tk, and .gq—are among the most popular for target embedding, as well as some of the most popular across all certificates. Spamhaus has identified these as the most abused TLDs for the purposes of sending spam [3]—we believe we are the first to also demonstrate their use in target embedding campaigns. Economically, the registrars for these TLDs allow anyone to register domains under them for free. This is naturally appealing to both benevolent users and attackers (thus the high rankings). Because many of these target embedding attacks distract users from the *real* TLD, it does not matter that the TLDs may be unrecognizable to users.

Actual TLD	# Unique Domains	Alexa Rank	Censys Rank	APWG Rank [7]
.com	73,218	1	1	1
.info	16,109	11	16	7
.cf	10,909	157	7	6
.net	10,798	4	3	3
.ga	9,545	146	8	10
.ml	9,379	133	10	9
.tk	9,289	83	2	14
.gq	6,241	275	12	15
.xyz	5,740	50	15	17
.top	5,447	72	27	–
.org	5,058	2	5	4
.online	5,003	63	40	18
.us	4,371	43	29	12
.site	3,892	87	42	–
.ru	3,142	3	9	13
.me	3,095	29	39	–
.in	2,728	17	30	11
.pw	2,681	91	49	2
.bid	2,306	167	81	–
.com.br	2,176	8	11	8
Others	64,918			

Table 3: Top 20 most common actual TLDs used by target embedding domains. "Alexa Rank" ranks the TLD by how many of the Alexa top-1M websites use that TLD; similarly, "Censys Rank" ranks by how many unique actual domains from the entire Censys dataset use that TLD. Our top-20 differs from APWG's; theirs includes more country-level TLDs (`.uk`, `.it`, `.pl`, and `.ca`), while target-embedding has more TLDs that can be confused with common English words (`.top`, `.site`, `.me`, `.bid`).

Second, we observe several TLDs that are frequently used for target embedding but unpopular in both Alexa and Censys rankings, such as .online, .cc, .bid, and .pw. We hypothesize that many of these are useful to attackers because (1) they are unpopular, and thus users are unlikely to recognize them as TLDs, and (2) they appear to be relevant with respect to the overall target domain. For example, appleid.apple.com.page-signin.pw targets apple.com and purports to be a login page; the .pw TLD bolsters this by appearing to refer to the user's password. The three most common targets within a .pw domain are icloud.com, runescape.com, and apple.com—all of which benefit from obtaining users' login credentials. As another example, ebay.com-item-iphone-x-256gb-space-gray-unlocked.k7l.bid targets ebay.com; the .bid TLD bolsters this by appearing to be asking for the user's bid. Three of the five most common targets within a .bid top-level domain are ebay.com, ebay.co.uk, and ebay.de.

Summary These results demonstrate that domains engaged in target embedding strategically choose their TLD based on economic

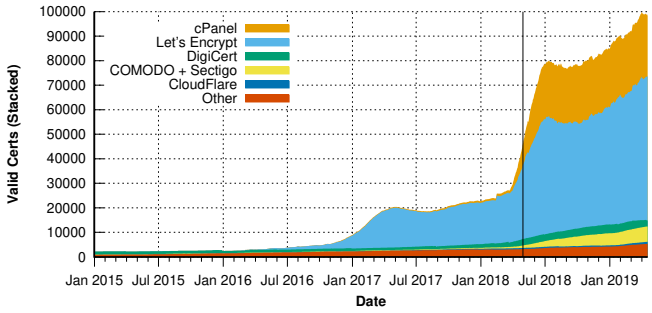

Figure 3: Stacked-plot of the number of valid certificates that include a target embedding domain, broken down by the issuing CA. The vertical line denotes when Google Chrome required all new certificates be included in CT logs.

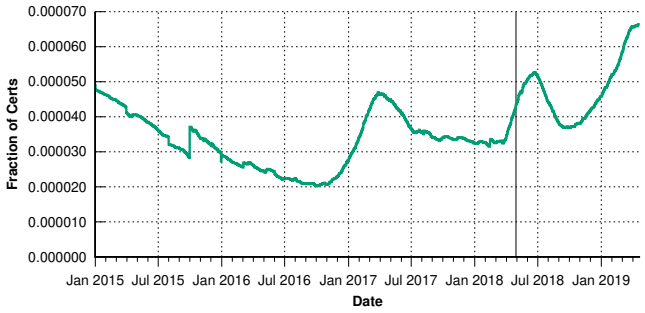

Figure 4: The fraction of all valid certificates with a target embedding domain has been increasing over time, indicating that the results in Figure 3 do not merely reflect the increased use of HTTPS.

concerns (free TLDs) and keywords relevant to the target (misleading TLDs like .bid and .pw). They also show that target-embedding domains exhibit unique characteristics when compared to Alexa-ranked domains or to all domains from Censys. Finally, these results show that attackers use a wide range of (real) TLDs. It is therefore important to use TLD-agnostic datasets, like TLS certificates, when studying domain impersonation.

5.3 Who is issuing impersonating certificates?

Our results thus far have identified hundreds of thousands of SSL/TLS certificates that contain target embedding. Next, we investigate what certificate authorities (CAs) are issuing these certificates, and how they have changed over time.

Figure 3 shows the number of valid (nonexpired) certificates issued by each CA over time, for the past four years, that contain at least one target-embedding domain. We make three key observations:

First, the use of certificates for target embedding is a relatively recent phenomenon. Prior to 2016, there were very few such certificates, the most common issuing CA being DigiCert. In 2016, with the introduction of Let's Encrypt, the ecosystem began to change drastically.

Second, over this relatively short period of time, the number of target embedding certificates has increased exponentially. At the beginning of 2016, there were only 3,154 target embedding certificates; by the end of our dataset, there are 124,432, an increase of 39.45×. This increase comprises three broad epochs: (1) late 2016/early 2017: Let's Encrypt CA starting issuing many target embedding certificates, (2) early 2017 to early 2018: COMODO was increasingly used; interestingly, during this time, the overall number of target embedding Let's Encrypt certificates remained relatively constant, and (3) since early 2018: cPanel became a common issuer of target embedding certificates, and the overall number of target embedding certificates has increased drastically across all three of these CAs.

To control for the overall increase in the number of HTTPS certificates, Figure 4 shows the fraction of *all* nonexpired certificates which included a target embedding domain over the same period of time as Figure 3. Interestingly, since the launch of Let's Encrypt, the fraction of all valid certificates with one or more target embedding domains has increased, indicating that the results from Figure 3 are not merely reflective of the increase in the PKI writ large. We do not yet understand the two spikes after January 2017; they roughly align with when Let's Encrypt was launched and when Chrome began requiring certificates be included in CT logs, but we are unable to attribute a root cause at this time.

Third, the increase can be largely attributed to CAs who offer free certificate issuance. Let's Encrypt [26] is a CA designed to foster greater adoption of HTTPS by issuing certificates in an automated fashion, for free. Users who obtained target embedding certificates quickly made extensive use of this free service; Let's Encrypt went from having a zero share of such certificates in early 2016 to issuing 61.76% of valid target embedding certificates at its peak in March 2017. These three CAs now constitute 95.37% of all target embedding certificates. By comparison, these three CAs issued 80.80% of all of the certificates in our dataset.

Recall that many of the most popular TLDs in target embedding certificates are those that can be registered for free: .ga, .ml, .cf, .tk, and .gq. In total, we identify 37,362 target embedding certificates with these TLDs. For the certificates corresponding to these five TLDs, we find that 85.91% of them are issued by Let's Encrypt, 12.13% are issued by cPanel, and only 1.40% are issued by COMODO. Collectively, 99.42% (37,144) of the domains had both free registration and free certificate issuance. In other words, there was no economic barrier of entry to register and secure these domains. Next, we investigate if there were barriers to hosting them.

Summary These results demonstrate that users who obtain target embedding certificates appear *not* to use a wide range of CAs. Rather, they prefer the small handful of CAs who provide free, automated certificate issuance. The low economic barrier of entry to target embedding has resulted in an exponential increase in the number of such certificates.

5.4 Who is hosting impersonating certificates?

Having investigated how target embedding domains and certificates are obtained, we turn to how their content is hosted. We used `curl` on each valid target embedding domain, and recorded the IP address of each site that returned a successful HTTP status code. Then, to

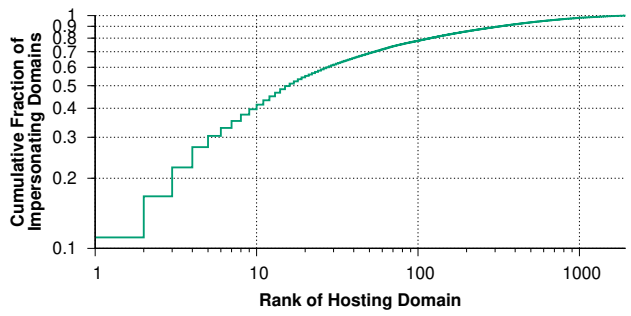

Figure 5: Cumulative fraction of target embedding domains by the providers who host them. These domains are primarily hosted on a few providers, but use a wide range of providers overall. (Note: log-scale axes.)

Hosting Provider	Unique Domains
verotel.com	7,670
namecheaphosting.com	3,852
amazonaws.com	3,817
websitewelcome.com	3,408
digitalocean.com	2,203
unifiedlayer.com	1,631
ovh.net	1,569
google.com	1,226
hetzner.de	1,205
internetx.com	1,134
Others	40,306

Table 4: Top 10 most popular hosting providers for target embedding domains.

determine who operates these IP addresses, we used the technique proposed by Cangialosi et al. [9]: we issue reverse DNS lookups for the IP addresses (many hosting providers, like Akamai, include their names in reverse lookups). If this information is not available, we then look up the IP address' autonomous system (AS) number, and report who operates that AS.

Figure 5 presents the distribution of the fraction of domains hosted by all distinct hosting domains we identified, and Table 4 shows the top 10 most popular hosting domains we observed. We make two key observations:

First, unlike the narrow distributions of CAs and registrars, we find that target embedding domains use a wide range of hosting sites. Figure 5 shows a long-tail: 0.6% (427) of the hosting domains we identified host only a single target embedding domain.

Second, however, there is a slight preference towards a small set of providers. The top 10 hosting domains in Table 4 collectively cover 42.09% of all of the target embedding domains we observe. Unsurprisingly, many of the most popular hosting domains offer options for free hosting, including, amazonaws.com (5.54%), unifiedlayer.com (2.37%), and hetzner.de (1.75%). We are unable to verify whether these target embedding domains are using the free or for-pay versions of these hosting providers. However, given our results that show preferences for free domain registration and certificate issuance, we speculate that they are using free versions for hosting, as well.

Most Common Preceding Tokens		Most Common Subsequent Tokens	
nil	64,610	`login`	4526
`www`	48,847	`account`	3655
`appleid`	38,904	`signin`	3264
`secure`	6631	`cafe`	2940
`mail`	4755	`secure`	2761
`login`	4131	`verify`	2709
`support`	3583	`id`	2459
`services`	3144	`support`	2386
`pay`	2989	`webapps`	2384
`id`	2055	`manage`	2034
Others	78,461	Others	258,110

Table 5: Top 10 most popular tokens to appear before and after targets in target embeddings.

Summary These results show that, unlike with CAs, there is a wide range of providers who host target embedding domains. The most popular hosting providers offer options for free hosting. Combined with our results showing free registration and free certificate issuance, we conclude that there is an end-to-end path by which attackers can acquire, host, and secure target embedding domains for free. As a result, currently, many such users face *no economic barrier of entry* to target embedding.

5.5 What is the structure of target embeddings?

Next, we evaluate the structure of target embeddings: what words most commonly precede and follow target embeddings? To compute this, we tokenize the FQDN at dots and dashes and extract the tokens appearing immediately before and immediately after each embedded target. In Table 5, we report on the top ten most common tokens, both before and after. *nil* denotes when the target was the first item to appear in the FQDN. We make several observations.

First, the most popular tokens—both before and after the target—are strongly concentrated around what appear to be attempts at phishing for user credentials. Terms such as `login`, `secure`, `signin`, and `account` are all associated with user logins. Similarly, `appleid`, `services`, and `pay`—three of the most popular preceding tokens—are all associated with monetary transactions.

Second, while there is a reasonably diverse distribution of most tokens—and while *nil* and `www` are expected to be popular subdomains—`appleid` stands out as a significant outlier. This not only reinforces our earlier finding that `apple.com` is one of the most highly targeted domains: it also shows *how* attackers do it: by reinforcing it with additional subdomains common to that website.

Third, we are not the first to explore words that precede or follow suspicious domain names. Netcraft [1] notes briefly that they incorporate prefixes and suffixes that are common in phishing domains when computing the domain score they use in validating certificate issuance requests. Unfortunately, their list is not public, and they only list a few: `update`, `login`, and `secure`. We also see high frequency of `login` and `secure`, but `update` is not in the top 10 for either set of tokens.

Summary The tokens that appear before and after an embedded target can yield a powerful signal as to the intent behind the embedding. This insight has been applied during some CAs' issuance [1];

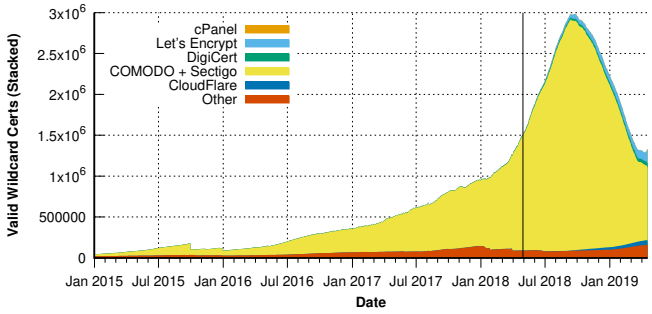

Figure 6: Stacked-plot graph representing the number of certificates valid on a given day that were issued by top CAs, for certs with wildcard domains that start with a TLD.

*.TLD	Unique Wildcard Domains	# Targets
*.blog	36,528	2
*.net	14,541	220
*.my	13,671	6
*.top	13,335	1
*.best	13,033	1
*.de	8,953	235
*.online	8,926	8
*.qa	8,149	3
*.us	6,451	11
*.live	6,389	2
Others	213,360	

Table 6: Top 10 most popular TLDs to follow a wildcard. For each, we provide the number of unique targets we observed using that TLD in target embedding; all of these targets could be embedded by one of these wildcard certificates. We limit analysis to TLDs that are used by at least one target.

however, it appears that our techniques can be used to help identify the set of suspicious tokens.

5.6 How are wildcard certificates utilized?

Recall that wildcard certificates contain at least one domain of the form `*.foo.bar.com`, allowing the certificate holder to use one compact certificate to authenticate many one-level subdomains of `foo.bar.com`. This is a powerful tool for benign website operators, as rolling out new services and subdomains does not necessarily require obtaining new certificates.

Wildcards in certificates may only be used to expand a single level of subdomain: the pattern matching the wildcard cannot include a dot [38]. For instance, `bar.example.com` is a valid completion of `*.example.com`, but `foo.bar.example.com` is not. Due to this restriction, any attempt to use a wildcard to mask the intended target of a target embedding attack *must* include the target's TLD immediately after the wildcard.

We measure all certificates where the wildcard is immediately followed by the TLD of a target seen in our targeting attacks, followed by a hyphen or a dot (e.g., `*.com-foo.bar.com` or *.co.uk.bar.com). Each such wildcard certificate has the potential to be used to target-embed *any* domain with the matching TLD. We cannot know which target(s) an attacker intends to impersonate—or if the certificates were ever used for impersonation—instead, we measure an upper bound of how many wildcard domains are *capable* of being used this way.

In total, we observe 343,336 unique wildcard domains whose wildcard is immediately followed by a TLD used in a targeted domain. Figure 6 shows how this number has changed over time, broken down by the CA who issued them.

Like with the number of target embedding certificates overall (§5.3), we see an exponential increase in the number of wildcard certificates starting with a targeted TLD. Unlike our previous results, this increase can be attributed to a single CA: COMODO. Likewise, the decline in late 2018 can also be directly attributed to COMODO. We are unable to explain this phenomenon, but we note that it is not relegated solely to wildcard certificates, nor even to target embedding writ large. Over this time, COMODO changed its name to Sectigo, and its overall number of valid certificates has gone down considerably. Let's Encrypt began issuing wildcard certificates in mid-2018. Although its share of target embedding domains has gone up, it has not done so to the extent that Let's Encrypt entered the non-wildcard ecosystem.

Next, we investigate the TLDs used in these potentially target embedding certificates. Table 6 shows the top 10 most frequent TLDs that appear immediately after a wildcard. Three of the four most common TLDs used by targets (.com, .net, and .de) show up in the top 10 TLDs following wildcards.

Table 7 shows the top 10 *actual* TLDs used in these wildcard certificates. Unlike the TLDs used in our target embedding domains (§5.2), we see a stronger concentration of the more traditionally popular TLDs, with particularly higher numbers of .de and .net. Because attackers are more likely to use TLDs that permit free domain registration, this result indicates that many wildcard certificates are likely *not* used for target embedding. However, we also see many instances of free TLDs that *are* correlated with target embedding attacks: .tk (8,626), .ga (5,892), .ml (7,037), .cf (6,046), and .gq (4,670). Finally, .stream is the third most popular real TLD for these wildcard domains. At least 90% of the 13,067 wildcard domains using .stream as their real TLD are part of the *.net- campaign outlined in Section 6.

Summary Wildcard certificates offer the possibility for the owner to perform target embedding on an unbounded number of targets. We observe a sharp increase in the number of wildcard certificates over the past couple years; with Let's Encrypt only recently offering wildcard certificates, we expect this to increase in the near future. Our results of the TLDs (both target and actual) used in wildcard certificates echo those of target embedding (Section 5.2).

5.7 Composing Impersonation Techniques

Both combosquatting and typosquatting occur in the actual domain of an FQDN, whereas target embedding requires the use of a subdomain. Due to this difference, target embedding is not mutually exclusive with other forms of domain impersonation. In this section, we investigate domains that compose methods of impersonation.

First we take the 256,045 FQDNs from our dataset of target embedding domains. Next, we see how many of those FQDNs' actual domains are either combosquatting or typosquatting, using

Wildcard TLD	Unique Domains	Alexa Rank	Censys Rank	APWG Rank
.com	103,735	1	1	1
.net	22,724	4	3	3
.de	16,400	5	4	–
.stream	13,067	145	109	–
.ru	10,330	3	9	13
.info	9,387	11	16	7
.tk	8,626	83	2	14
.live	8,369	140	52	–
.ml	7,037	133	10	9
.com.br	6,586	8	11	8
Other	137,075			

Table 7: Top 10 most common actual TLDs used by wildcard certificates that begin with a fake TLD following the `*`.

the domains collected in §4.3. We discovered 2,442 FQDNs using both target embedding and combosquatting, and 443 FQDNs using both target embedding and typosquatting.

Of the 2,875 FQDNs from the union of those two sets, 960 targeted the same target using both methods of impersonation. 1,062 targeted "apple" with one method, and "icloud" with the other method. Most of the 853 remaining FQDNs targeted seemingly unrelated pairs, such as "`www.docusign.com.amazonline.com.br`" and "`paypal.com.webapps-update-icloud.ga`".

Finally, we see if any FQDNs compose unicode homographs with target embedding. We begin by taking all domains with unicode characters, identified with the Punycode [10] prefix "xn–". For each token that includes Unicode characters, we try all combinations of ASCII characters that could be confused with the Unicode characters [28]. If this substitution results in the e2LD of an Alexa 100k target domain, we then see if that token is followed by the target's TLD[2], and that this TLD is not the real TLD of the domain. In all, we discovered 13 FQDNs that compose Unicode homographs and target embedding in this way.

We can only speculate why someone would compose impersonation methods. It may be the case that attackers feel they can maximize their chances at successfully deceiving users. Or, perhaps composing homographs with target embedding makes it easier for an impersonating domain to evade detection. Regardless, domains with multiple forms of impersonation represent a small, but present, portion of impersonating domains.

6 COORDINATED CAMPAIGNS

Our analysis in §5 identified hundreds of thousands of individual instances of target embedding. In this section, we demonstrate that many target embedding domains can be pattern-matched to uncover what appear to be *coordinated campaigns* of impersonation.

To this end, we perform a case study analysis of four large-scale campaigns that registered many unique domains with a common structure to impersonate the same target. Safe Browsing identified some of the domains in these campaigns as malicious, but using our methodology we can determine Safe Browsing's coverage of these campaigns. We summarize the results in Table 8.

[2]We also check for confusable Unicode characters in the TLD token.

Campaign	Total Domains	Flagged by Safe Browsing	
starwars.com	3,071	1,079	(35.14%)
runescape.com	4,522	854	(18.89%)
*.net-	11,765	7,439	(63.23%)
*.co-	1,926	1,409	(73.16%)

Table 8: Total number of target-embedded domains & Safe Browsing coverage for four campaigns with over 1,000 unique domains of similar structure.

StarWars This campaign had FQDNs of the form `starwars.com.p58vfa15.top` and `starwars.com.dvqdh83l6r.site`. Of the websites we detected, Safe Browsing flagged 35.14% as employing social engineering. Subject Alternate Name lists on these certificates also included website names or products and services, such as "amazon," "android-browser-update," "apple," "facebook," "microsoft," and "security-alert". Some of these certs had over 30 unique FQDNs issued to the same actual domain. While containing the e2LD of other targets, it is unknown why "`starwars.com`" was the only target whose e2LD+TLD was embedded. Interestingly, Safe Browsing only flagged domains in this campaign that used the .top and .site TLDs. Domains with the .bid TLD may not have become active in the campaign yet.

Runescape The Runescape campaign targeted `runescape.com`, a massively multiplayer online role-playing game. Examples of these domains include `oldschool.runescape.com-ds.ml` and `secure.runescape.com-kn.cf`. Domains in this campaign were issued with over 30 unique TLDs, the most common being .ml, .ga, and .cf. Safe Browsing had the lowest coverage with this campaign, flagging only 18.89% of the domains we identified.

Wildcard campaigns Our last two campaigns were discovered from our analysis of wildcard domains in § 5.6. The *.net- campaign saw domains of the form `*.net-ak78.stream` and `*.net-x69.stream`. The *.co- campaign was similar, with domains like `*.co-j26.bid` and `*.co-m76.bid`. While we do not know what these campaigns targeted, we do know that Safe Browsing had much better coverage of these campaigns than the previous two. Safe Browsing had flagged 63.23% and 73.16% of the domains fitting these structures, respectively. However, thousands of these domains were not reported as malicious and still obtained certificates.

Summary There appear to be several very large, coordinated campaigns of target embedding. Fortunately, with the global view that CT Logs provide, such campaigns can be straightforward to find through basic pattern matching. Interestingly, while Google Safe Browsing identified large percentages (18–73%) of these domains as unsafe, we are still able to find thousands that were not yet blacklisted. This indicates that our techniques for identifying and grouping together what appear to be domain impersonation attacks can be used to help improve the coverage of other tools for detecting misbehavior.

7 POTENTIAL COUNTERMEASURES

Our longitudinal study reveals several entities who play a significant role in how attackers launch target embedding attacks. In this

section, we ask: whose job should it be to help mitigate this attack? We step through each of the relevant players and discuss what role they *could* play, and the impact that their actions could have.

Browsers Modern browsers incorporate techniques to warn users about potentially harmful, misleading, or insecure websites. Google Safe Browsing [18] and other similar services, like PhishTank [35] use the *content* of the web page to determine whether it is a threat. HTTPS-only services, on the other hand, will not have any content available until they acquire a certificate. Thus, a reactive solution such as Safe Browsing inevitably misses many of the impersonation attempts. Browsers have been incredibly successful at mitigating homograph attacks by adopting Punycode (§3.2). Additional user-interface updates, or inspecting domains accessed by users for the presence of target embedding and other forms of impersonation, may help prevent users from being deceived.

Third-Party Watchdog Certificate Transparency enables third-party auditors and monitors to ensure the PKI is functioning as intended. A third party monitor could collect a body of impersonating domains on certificates, and determine if those domains are phishing or engaging in other unacceptable behavior. They could also gather a list of impersonating domains that have obtained certificates but not yet hosted any content, and repeatedly monitor these sites until they go live. A watchdog would know the instant one of these domains began hosting malicious content, and add such domains to a blacklist *before they have an opportunity to successfully attack any users.*

Facebook now offers a Certificate Transparency Monitoring service [14]; after submitting a possible target domain, Facebook issues an alert when a potentially impersonating certificate is added to a CT log. Cloudflare crawls CT logs and raises an alert when a certificate is issued for a customer's (legitimate) domain [39]. Our techniques could be incorporated into such services and alert customers when their website is the target of an impersonation attack.. However, flagging potential attacks is not enough; ideally, this information should also be shared with CAs and browsers, so that they may take action that can directly protect users.

Certificate Authorities CAs are ultimately responsible for issuing the certificates that attackers use. Before issuing certificates, CAs could ostensibly apply techniques like those presented in this paper to flag potential impersonating attacks, and then either deny the certificate request or require a more in-depth vetting process. Adoption of defenses by just three CAs could potentially address 95.37% of all target embedding attacks (§5.3). But should CAs be expected to play a role at all?

Let's Encrypt argues that CAs should not play a role in detecting phishing, as they "make poor content watchdogs" [4]. On the other hand, the CA/Browser Forum argues that the CA has a responsibility to flag "high risk" certificate requests and to follow them up with additional verification (§4). With our techniques, it is straightforward to identify the targets within a target embedding domain; a natural extension to the automated CAs of today would be to issue automated ACME challenges for *each* of the "apparent" domains within a FQDN.

These requirements leave open to interpretation the extent to which a CA must or ought to go to identify so-called high risk certificate requests. The CA/Browser forum suggests using third-party phishing repositories, in particular the Google Safe Browsing list [18] or the Miller Smiles phishing list [31]. As discussed above, these third-party services tend to use the content of a web page to determine if it is a threat, and are thus not applicable at the time of certificate issuance.

Unfortunately, wildcard certificates (§5.6) would complicate efforts to mitigate impersonation attacks at the CA. Any innocuous-appearing domain with a wildcard could ostensibly be turned into a target embedding attack with hyphens. One possibility would be for CAs to raise the bar for obtaining wildcard certificates. Perhaps the most feasible approach would be for CAs to work alongside browsers and third parties in determining when wildcard domains are used for malicious purposes and to revoke those certificates when necessary.

Summary There is no *one* entity that could fully defend against impersonation attacks by target embedding. CAs can serve a powerful role at the time of certificate issuance, but with wildcard certificates, target embedding attacks may not become evident until well after issuance. Conversely, browsers can detect impersonations when users visit a website, but browser-based initiatives like Google Safe Browsing and PhishTank are reactive, thus missing many of the impersonations.

As is typical with the PKI [48], security appears to be possible only if multiple parties work in tandem. We envision CAs submitting to CT logs, third-party watchdogs monitoring and flagging certificates using techniques like ours, and browsers incorporating these flags in their Safe Browsing-like initiatives.

8 CONCLUSION

As an unexpected result of training users to look for a "secure lock icon," users have become more likely to trust websites hosted via HTTPS [37]. In this paper, we have shown that this trust has also made users more susceptible to domain impersonation attacks. We have also identified a new classification of attack, *target embedding*, that is the most effective attack against browsers today (browsers already defend against homographs, the most effective attack). By analyzing a longitudinal certificate dataset spanning all HTTPS certificates collected by Censys, we find several alarming results: target embedding is on the rise, it is free for attackers to launch, domains include preceding and succeeding tokens indicating phishing attacks, and attackers are adapting by composing attacks together. Unfortunately, there is no one clear fix for target embedding: we argue that multiple players will have to coordinate to effectively fix this problem. To assist in this effort, we have made our code and data publicly available at: `https://securepki.org`

ACKNOWLEDGMENTS

We thank Balakrishnan Chandrasekaran, Dave Choffnes, Bruce Maggs, Nick Sullivan, Christo Wilson, our shepherd, Paul Pearce, and the anonymous reviewers for their helpful feedback. This research was supported by NSF CNS grants 1563320, 1564143, 1816802, 1900879, 1901090, and 1901325.

REFERENCES

[1] [n.d.]. Netcraft Pre-issuance Certificate Checking. https://www.netcraft.com/services-for-certificate-authorities/pre-issuance-certificate-checking/.

[2] 2017. American Community Survey (ACS) 5-year Estimates. https://www.census.gov/programs-surveys/acs/news/data-releases/2017/release.html.

[3] 2018. The Spamhaus Project: The World's Most Abused TLDs. https://www.spamhaus.org/statistics/tlds/.

[4] Josh Aas. 2015. The CA's Role in Fighting Phishing and Malware. https://letsencrypt.org/2015/10/29/phishing-and-malware.html.

[5] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse. In *Network and Distributed System Security Symposium (NDSS)*.

[6] Devdatta Akhawe and Adrienne Porter Felt. 2013. Alice in Warningland: A Large-scale Field Study of Browser Security Warning Effectiveness. In *USENIX Security Symposium*.

[7] Anti-Phishing Working Group. [n.d.]. Phishing Attack Trends Report – 4th Quarter, 2018. https://docs.apwg.org/reports/apwg_trends_report_q4_2018.pdf.

[8] CA/Browser Forum. 2018. Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates. https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.5.7-29-Apr-2018.pdf.

[9] Frank Cangialosi, Taejoong Chung, David Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. 2016. Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. In *ACM Conference on Computer and Communications Security (CCS)*.

[10] A. Costello. 2003. Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA). RFC 3492 (Proposed Standard). http://www.ietf.org/rfc/rfc3492.txt

[11] Rachna Dhamija, J. D. Tygar, and Marti Hearst. 2006. Why Phishing Works. In *ACM Conference on Human Factors in Computing Systems (CHI)*.

[12] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. 2015. A Search Engine Backed by Internet-Wide Scanning. In *ACM Conference on Computer and Communications Security (CCS)*.

[13] Yahia Elsayed and Ahmed Shosha. 2018. Large Scale Detection of IDN Domain Name Masquerading. In *APWG Symposium on Electronic Crime Research (eCrime)*.

[14] Facebook. [n.d.]. Certificate Transparency Monitoring. https://developers.facebook.com/tools/ct/subscriptions.

[15] Adrienne Porter Felt, Robert W. Reeder, Alex Ainslie, Helen Harris, Max Walker, Christopher Thompson, Mustafa Emre Acer, Elisabeth Morant, and Sunny Consolvo. 2016. Rethinking Connection Security Indicators. In *Symposium on Usable Privacy and Security (SOUPS)*.

[16] Anthony Y. Fu, Xiaotie Deng, Liu Wenyin, and Greg Little. 2005. The Methodology and an Application to Fight against Unicode Attacks. In *Symposium on Usable Privacy and Security (SOUPS)*.

[17] Evgeniy Gabrilovich and Alex Gontmakher. 2002. The Homograph Attack. *Communications of the ACM* 45, 2 (2002).

[18] Google Safe Browsing [n.d.]. Google Safe Browsing. https://safebrowsing.google.com.

[19] Crane Hassold. [n.d.]. The Mobile Phishing Threat You'll See Very Soon: URL Padding. https://info.phishlabs.com/blog/the-mobile-phishing-threat-youll-see-very-soon-url-padding.

[20] Tobias Holgers, David E. Watson, and Steven D. Gribble. 2006. Cutting through the Confusion: A Measurement Study of Homograph Attacks. In *USENIX Annual Technical Conference*.

[21] Markus Jakobsson, Alex Tsow, Ankur Shah, Eli Blevis, and Youn-Kyung Lim. 2007. What Instills Trust? A Qualitative Study of Phishing. In *Usable Security (USEC)*.

[22] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Roza Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *ACM Conference on Computer and Communications Security (CCS)*.

[23] Viktor Krammer. 2006. Phishing Defense against IDN Address Spoofing Attacks. In *International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*.

[24] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. 2010. Teaching Johnny Not to Fall for Phish. *ACM Transactions on Internet Technology (TOIT)* 10, 2 (2010).

[25] Chris Larsen. 2009. Bad Guys Using Internationalized Domain Names (IDNs). https://www.symantec.com/connect/blogs/bad-guys-using-internationalized-domain-names-idns.

[26] Let's Encrypt [n.d.]. Let's Encrypt. https://letsencrypt.org/.

[27] Eric Lin, Saul Greenberg, Eileah Trotter, David Ma, and John Aycock. 2011. Does Domain Highlighting Help People Identify Phishing Sites?. In *ACM Conference on Human Factors in Computing Systems (CHI)*.

[28] List of Unicode Confusables [n.d.]. List of Unicode Confusables. https://unicode.org/cldr/utility/confusables.jsp.

[29] Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Dave Levin, Bruce Maggs, Alan Mislove, Aaron Schulman, and Christo Wilson. 2015. An End-to-End Measurement of Certificate Revocation in the Web's PKI. In *ACM Internet Measurement Conference (IMC)*.

[30] D. Kevin McGrath and Minaxi Gupta. 2008. Behind Phishing: An Examination of Phisher Modi Operandi. In *USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*.

[31] Miller Smiles Phishing List [n.d.]. Miller Smiles Phishing List. http://www.millersmiles.co.uk/forum/index.php.

[32] Tyler Moore and Benjamin Edelman. 2010. Measuring the Perpetrators and Funders of Typosquatting. In *Financial Cryptography (FC)*.

[33] Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2013. Bitsquatting: Exploiting bit-flips for fun, or profit?. In *International World Wide Web Conference (WWW)*.

[34] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2014. Soundsquatting: Uncovering the Use of Homophones in Domain Squatting. In *International Conference on Information Security (ISC)*.

[35] PhishTank [n.d.]. PhishTank. https://www.phishtank.com.

[36] The Chromium Projects. [n.d.]. IDN in Google Chrome. https://www.chromium.org/developers/design-documents/idn-in-google-chrome.

[37] Scott Ruoti, Tyler Monson, Justin Wu, Daniel Zappala, and Kent Seamons. 2017. Weighing Context and Trade-offs: How Suburban Adults Selected Their Online Security Posture. In *Symposium on Usable Privacy and Security (SOUPS)*.

[38] P. Saint-Andre and J. Hodges. 2011. Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS). RFC 6125. https://tools.ietf.org/rfc/rfc6125.txt

[39] Ben Solomon. [n.d.]. Introducing Certificate Transparency Monitoring. https://new.blog.cloudflare.com/introducing-certificate-transparency-monitoring/amp/.

[40] Proofpoint Staff. [n.d.]. Snowshoe Spamming Brings Scale to Savvy Subdomain Phishing Attacks. https://www.proofpoint.com/us/threat-insight/post/snowshoe-spamming-brings-scale-savvy-subdomain-phishing-attacks.

[41] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The Long "Taile" of Typosquatting Domain Names. In *USENIX Security Symposium*.

[42] Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt. 2019. The Web's Identity Crisis: Understanding the Effectiveness of Website Identity Indicators. In *USENIX Security Symposium*.

[43] Ke Tian, Steve T.K. Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. In *ACM Internet Measurement Conference (IMC)*.

[44] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J. Alex Halderman. 2016. Towards a Complete View of the Certificate Ecosystem. In *ACM Internet Measurement Conference (IMC)*.

[45] Yi-Min Wang, Doug Beck, Jeffrey Wang, Chad Verbowski, and Brad Daniels. 2006. Strider Typo-Patrol: Discovery and Analysis of Systematic Typo-Squatting. In *USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*.

[46] Min Wu, Robert C Miller, and Simson L Garfinkel. 2006. Do security toolbars actually prevent phishing attacks?. In *ACM Conference on Human Factors in Computing Systems (CHI)*.

[47] Aiping Xiong, Robert W Proctor, Weining Yang, and Ninghui Li. 2017. Is Domain Highlighting Actually Helpful in Identifying Phishing Web Pages? *Human Factors* 59, 4 (2017).

[48] Liang Zhang, David Choffnes, Tudor Dumitras, Dave Levin, Alan Mislove, Aaron Schulman, and Christo Wilson. 2014. Analysis of SSL certificate reissues and revocations in the wake of Heartbleed. In *ACM Internet Measurement Conference (IMC)*.

A DOMAIN IMPERSONATION SURVEY

A.1 Survey Design

Section 3 discusses the results of a user survey we conducted to measure the effectiveness of different domain impersonation techniques. This appendix provides more detail about how our survey was designed and conducted.

Participant Recruitment 251 participants were recruited from Amazon's MTurk platform. The study was only advertised to residents of the United States (and territories) over 18 years of age, with an MTurk HIT rate over 95%. An MTurker's HIT rate is the percentage of MTurk tasks that they completed successfully and received compensation for. The survey itself was conducted through Qualtrics.

Compensation We expected the survey to take approximately 15 minutes. Participants were offered $2 USD for completing the survey, slightly above the US minimum wage for that amount of time. The mean time to completion was 7.69 minutes, with a standard deviation of 5.01 minutes.

Ethical Considerations Our study was approved by our institution's Institutional Review Board (IRB). Informed consent was obtained from participants before conducting the survey. The domains we selected and presented to users were similar in structure to impersonating domains witnessed in the wild. None of the domains hosted content at the time of the survey's creation, however we did not own the domains and it was conceptually possible that one of them could start hosting malicious content during the course of the study. To mitigate any harm in this scenario, we presented the domains to users through an image (to prevent copying and pasting them into their own URL bar), and instructed participants to not attempt to visit any of the domains described in the study.

Survey Construction Participants were asked 6 questions in each of 8 categories, for a total of 48 questions. Each question presented users with a URL and an organization, and asked the user the yes-or-no question: "Do you believe that this is the organization's URL?"? The first four categories showed domains engaged in target embedding, combosquatting, typosquatting, and utilizing Unicode homographs. The fifth category showed the same domains as the Unicode homograph category, but rendered those domains as Punycode instead of Unicode. Category 6 showed unspoofed domains paired with their appropriate organization, and category 7 showed unspoofed domains paired with an incorrect organization. Finally, category 8 showed 6 domains from the previous categories, but with Google Chrome's "not secure" warning instead of a lock icon. Our experimental design on this category was insufficient to draw conclusions on, and these results are not discussed in the paper. We will be releasing our aggregate data publicly.

Demographic Information After the survey, participants were given the option to provide basic demographic information. Table 9 includes information on those who participated. Our population overrepresented male, white, educated, and between the ages of 18-38 when compared to Census statistics from the American Community Survey [2]. Over 40% of our population reported having a technical background. We believe that those with a technical background may be better equipped to recognize attempts at domain impersonation Our results would be conservative in this respect.

Sanitization 251 participants completed the survey. Of those, 7 participants provided the same answer ("Yes") to every single question. We removed the responses from those participants from our analysis.

Participant Demographics	#	%
Gender		
Male	155	61.75%
Female	94	37.45%
Other	2	0.80%
Age		
18-29	84	33.47%
30-39	106	42.23%
40-49	33	13.15%
50-59	18	7.17%
60+	8	3.19%
No Answer	2	0.80%
Ethnicity		
White	189	75.30%
Hispanic or Latino	20	7.79%
Black or African American	23	9.16%
American Indian or Alaska Native	1	0.40%
Asian, Native Hwaaiian, or Pacific Islander	16	6.37%
Other	2	0.80%
Highest Level of Education		
Some High School Credit, No Diploma, or Equiv.	2	0.80%
H. School Graduate, Diploma, or the Equiv. (GED)	32	13.15%
Some College Credit, No degree	42	18.33%
Trade/Technical/Vocational Training	2	3.59%
Associate's Degree	22	11.55%
Bachelor's Degree	112	43.82%
Master's Degree	12	7.57%
Professional Degree	2	0.80%
Doctorate Degree	1	0.40%
Technical Background/Training?		
Yes	102	40.64%
No	148	58.96%
No Answer	1	0.40%

Table 9: Participant demographics for our user study. In addition to the information reported above, the age rage for participants was 21-70 years old, with a mean age of 35.17 and std. deviation of 10.08. We recruited participants from 45/50 states (with no participants from Hawaii, Montana, North Dakota, South Dakota, Vermont, D.C., Puerto Rico, or other US territories.)

A.2 Survey Protocol

Page One: Consent Form

⟨ Participants were presented with a consent form, affirming that they were 18 years of age or older, read and understood the consent form, and voluntarily agreed to participate in our study. If the participant answered "No" to any of the above questions, the survey would end with no further input. ⟩

Page 2: MTurk ID Verification

Before we begin, please verify your Amazon Mechanical Turk ID in the text field below. You can find your MTurk ID on your dashboard. Then click next. ⟨Text field⟩

Page 3: Survey Instructions

This survey will ask your opinion about URLs. A URL is an address on the internet that is used to indicate what website someone would like to visit. "http:/www.facebook.com" and "https://www.google.com" are URLs for Facebook and Google, respectively. You will be shown a series of 48 questions, similar to the examples below. We ask that you simply answer with your first instinct. Afterward, we will ask you several demographic questions. We do not anticipate the survey to take more than 15 minutes. Make your judgements based only on the information presented in the question; do not attempt to visit any of the websites described in this survey, and do not enter the displayed URLs into your web browser.

You will be shown the name of an organization, and a URL, as shown below:

Example 1
Organization: Google
URL: https://google.com

Example 2
Organization: Yahoo
URL: https://google.com

You will then be asked whether or not you believe this is the organization's URL. In example 1, "https://www.google.com" is Google's URL, and so the answer to this question would be "yes." Since "https://www.google.com" is not the website for Yahoo's organization, the answer to example 2 would be "no."

When you are ready, please click the arrow to continue to the survey.

Pages 4-51: Survey Questions

⟨ Participants were shown each of the following organization/URL pairs on a separate page, in a random order. For each pair, participants were asked: "Do you believe that this is the organization's URL?", and presented with "Yes" and "No" options. Note that domains in categories 1-7 were shown in a Google Chrome URL bar with a valid HTTPs lock icon, and domains in category 8 were displayed with the "Not Secure" warning Chrome displays when connecting to websites over HTTP. ⟩

Category 1: Target Embedding

- Amazon: https://www.amazon.com.order-history.com
- Apple: https://apple.com.p58vfa25.com
- Ebay: https://www.ebay.com-itm-lincoln-ranger-305-d-diesel-engine.xvp.review
- Facebook: https://facebook.com-login.pw
- Google: https://google.com-signin.com
- Paypal: https://paypal.com-ds.ml

Category 2: Typosquatting

- Amazon: https://amzon.com
- Apple: https://applee.com
- Ebay: https://eaby.com
- Facebook: https://faceobok.com
- Google: https://googgle.com
- Paypal: https://papal.com

Category 3: Combosquatting

- Amazon: https://amazon-wikis.com
- Apple: https://appleaccountuser.com
- Ebay: https://secure5-ebay.bid
- Facebook: https://facebook1234.cf
- Google: https://drive-google.com
- Paypal: https://paypal-update.ml

Category 4: Homographs

- Amazon: https://amaZon.com
- Apple: https://appɪe.com
- Ebay: https://eβay.com
- Facebook: https://faćebook.com
- Google: https://g0ogle.com
- Paypal: https://payρal.com

Category 5: Punycode

- Amazon: https://xn–amaon-ofy.com
- Apple: https://xn–appe-xhc.com
- Ebay: https://xn–eay-sxc.com
- Facebook: https://xn–acebook-2vf.com
- Google: https://xn–g0gle-kye.com
- Paypal: https://xn–ayal-9ndc.com

Category 6: Positive Control

- Amazon: https://amazon.com
- Apple: https://apple.com
- Ebay: https://ebay.com
- Facebook: https://facebook.com
- Google: https://google.com
- Paypal: https://paypal.com

Category 7: Negative Control

- Amazon: https://twitter.com
- Apple: https://bankofamerica.com
- Ebay: https://netflix.com
- Facebook: https://dropbox.com
- Google: https://yahoo.com
- Paypal: https://youtube.com

Category 8: "Not Secure" Warning

- Amazon: http://www.amazon.com.order-history.com
- Apple: http://appɪe.com
- Ebay: http://xn–eay-sxc.com
- Facebook: http://faceobok.com
- Google: http://google.com
- Paypal: http://paypal-update.ml

Pages 52: Demographic Questions

Please specify the gender with which you most closely identify.

- Male
- Female
- Other
- Prefer not to answer

Please specify your age. ⟨Numeric Entry⟩

Please specify your ethnicity.

- White
- Hispanic or Latino
- Black or African American
- American Indian or Alaska Native
- Asian, Native Hawaiian, or Pacific Islander
- Other

Please specify which US state/province you live in. ⟨Dropdown menu of US state & territory names⟩

Please specify the highest degree or level of school you have completed.

- Some high school credit, no diploma or equivalent
- High school graduate, diploma or the equivalent (for example: GED)
- Some college credit, no degree
- Trade/technical/vocational training
- Associate degree
- Bachelor's degree
- Master's degree
- Professional degree
- Doctorate degree

Have you ever received training, education, or worked in a field related to Computer Science or Information Technology (IT)?

- Yes
- No

Pages 53: Final Comments

Do you have any comments or feedback you would like to share with us regarding any aspect of the survey? These responses will remain private (will not be included in any analysis or reports), and do not affect your compensation. ⟨Free Response⟩

Pages 54: Exit Page

Thank you for participating in our survey. As a reminder, any information collected in this survey will be stored securely until the conclusion of this study, at which point all records will be destroyed (non-personally identifying results may be retained up to three years for research purposes). Your compensation will be credited to your Amazon Mechanical Turk account shortly.