Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild

Ke Tian, Steve T.K, Jan, Hang Hu, Danfeng Yao, Gang Wang
Department of Computer Science, Virginia Tech
{ketian, tekang, hanghu, danfeng, gangwang}@vt.edu

ABSTRACT

Today's phishing websites are constantly evolving to deceive users and evade the detection. In this paper, we perform a measurement study on *elite* phishing domains where the websites impersonate trusted entities not only at the *page content* level but also at the *web domain* level. To search for elite phishing pages, we scanned five types of squatting domains over 224 million DNS records and identified 657K domains that are likely impersonating 702 popular brands. Then we build a novel machine learning classifier to detect phishing pages from both the web and mobile pages under the squatting domains. A key novelty is that our classifier is built on a careful measurement of evasive behaviors of phishing pages in practice. We introduce new features from visual analysis and optical character recognition (OCR) to overcome the heavy content obfuscation from attackers. In total, we discovered and verified 1,175 elite phishing pages. We show that elite phishing pages are used for various targeted scams, and are highly effective to evade detection. More than 90% of them successfully evaded popular blacklists for at least a month.

CCS CONCEPTS

• **Security and privacy → Web application security**;

ACM Reference Format:
Ke Tian, Steve T.K, Jan, Hang Hu, Danfeng Yao, Gang Wang. 2018. Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. In *Proceedings of IMC '18*. ACM, New York, NY, USA, 14 pages. https://doi.org/1111

1 INTRODUCTION

Today, phishing attacks are increasingly used to exploit human weaknesses to penetrate critical networks. A recent report shows that 71% of targeted attacks began with a spear phishing [19], which is one of the leading causes of the massive data breaches [18]. By luring the targeted users to give away critical information (*e.g.*, passwords), attackers may hijack personal accounts or use the obtained information to facilitate more serious attacks (*e.g.*, breaching a company's internal network through an employee's credential) [17].

Phishing webpages, as the landing pages for phishing messages [30, 42, 56], are constantly involving to *deceive users* and *evade detection.*

IMC '18, October 31–November 2, 2018, Boston, MA, USA
© 2018 Association for Computing Machinery.
ACM ISBN TBA…$TBA
https://doi.org/1111

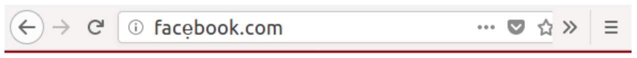


Figure 1: An example of the internationalized domain name `xn--facbook-ts4c.com` **(homograph), which is displayed as** `facẹbook.com` **in the address bar.**

Sophisticated phishing pages are constructed to impersonate the webpages of banks, government agencies, and even the internal systems of major companies [28]. In addition, phishing pages can also impersonate the domain names of trusted entities via domain squatting techniques [35, 40, 50]. For example, an attacker may register a domain that looks like `facebook.com` using an internationalized domain name to deceive users, as shown in Figure 1. While anecdote evidence suggests such "*elite*" phishing pages exist, there is still a lack of in-depth understandings of how the phishing pages are constructed and used in practice.

In this paper, we describe our efforts in searching and detecting *squatting phishing domains* where the attackers apply impersonation techniques to both the web content and the web domain. For convenience, we use the term "squatting phishing" and "elite phishing" interchangeably. Our goals are three-folds. First, we seek to develop a systematic method to search and detect squatting phishing domains in the wild. Second, we aim to empirically examine the *impersonation* techniques used by the attackers to deceive users. Third, we want to characterize the *evasion* techniques used by the squatting phishing pages and their effectiveness to avoid detection.

To these ends, we design a novel measurement system `SquatPhi` to search and detect squatting phishing domains. We start by detecting a large number of "squatting" domains that are likely to impersonate popular brands. Then, we build a distributed crawler to collect the webpages and screenshots for the squatting domains. Finally, we build a machine learning classifier to identify squatting phishing pages. A key novelty is that our classifier is built based on a careful measurement of the evasion techniques used by real-world phishing pages These evasion techniques are likely to render existing detection methods ineffective. Below, we describe each step and the discuss our key findings.

Squatting Domain Detection. We focus on 702 highly popular online services (brands) and search for squatting domains that are likely to impersonate them (*e.g.*, Facebook, Paypal). We apply five different squatting techniques [40, 51, 58] to generate candidate domains, including typo squatting, bits squatting, homograph squatting, combo squatting, and wrongTLD squatting. By analyzing over 224 million DNS records, we identified 657,663 squatting domains, and crawled both the web version and mobile version of their webpages (1.3 million pages) for 4 snapshots over a month.

A Novel Phishing Classifier. To detect squatting phishing pages among a large number of squatting domains, we develop a novel machine learning classifier. Based on a ground-truth set of 4004 user-reported phishing pages (from PhishTank [9]), we characterize common evasion techniques, and develop new features as countermeasures. Particularly, we observe that evasion techniques (*e.g.*, code obfuscation, string obfuscation, and layout obfuscation) often hide phishing related text in the source code or change the layout of the phishing pages. To this end, we apply visual analysis and *optical character recognition* (OCR) to extract key visual features from the page screenshots (particularly the regions of the login form). The intuition is that no matter how attackers obfuscate the HTML content, the visual presentation of the page will still need to look legitimate to deceive users. Our classifier is highly accurate, with a false positive rate of 0.03 and a false negative rate of 0.06.

Squatting Phishing Pages and Evasion. By applying the classifier to the 657,663 squatting domains, we identified and confirmed 1,175 squatting phishing domains (857 web pages, 908 mobile pages). Our results suggest that squatting phishing pages exist but are not highly prevalent among squatting domains (0.2%). In addition, squatting phishing pages take advantage of all five domain squatting techniques to deceive users, and are used for various targeted scams. Examples range from setting up fake Google search engines in Ukraine to scamming Uber's truck drivers or impersonating a payroll system to scam employees. Furthermore, squatting phishing pages are more likely to adopt evasion techniques and are highly effective in evading detections. More than 90% of phishing domains successfully evaded popular blacklists such as VirusTotal (70+ blacklists), PhishTank, and eCrimeX for at least a month. Our results provide key insights into how to develop effective countermeasures.

Our paper has three main contributions:

- First, we propose a novel end-to-end measurement framework `SquatPhi` to search and detect squatting phishing pages from a large number of squatting domains.[1]
- Second, we perform the first in-depth analysis on squatting phishing domains in the wild. Our results provide insights into how squatting phishing pages impersonate popular brands at both the domain and content level.
- Third, we empirically characterize the evasion techniques used by squatting phishing pages. The results indicate that existing detection methods are likely to be ineffective and need to be improved.

2 BACKGROUND & MOTIVATIONS

We start by introducing the background of phishing, and defining *elite* phishing pages that apply squatting techniques.

Phishing Web Pages. Phishing has been widely used by cybercriminals to steal user credentials and breach large networks. Typically, attackers would impersonate a trusted entity to gain the victim's trust, luring the victim to reveal important information. Phishing pages often act as the landing pages of malicious URLs distributed by phishing emails [42], SMS [56], or social network messages [30]. The phishing pages usually contain a form to trick users to enter passwords or credit card information.

As phishing attacks become prevalent [7], various phishing detection methods have been proposed, ranging from URL blacklisting [23] to visual similarity based phishing detection [47] and website content-based classification [61]. Visual similarity-based phishing detection [47] aims to compare the original webpages of popular brands to suspicious pages to detect "impersonation". Machine learning based methods [61] rely on features extracted from the HTML source code, JavaScript, and the web URLs to flag phishing websites. As phishing attacks evolve, we are curious about the potential evasion techniques used by attackers in practice.

Domain Name Squatting. Domain name squatting is the act of registering domain names that are likely to cause confusions with existing brands and trademarks. Domain name squatting has led to abusive activities such as impersonating the original websites to steal traffic, and distribute ads and malware. A squatting domain usually shares many overlapping characters at a targeted domain. Common squatting techniques include bit mutation [51], typo spelling [50] and homograph imitating [40]. Internationalized domain names (IDN) can be used for domain squatting domains, since IDNs can have a similar visual representation as the target domains after encoding (Figure 1).

Squatting domains can cause trouble to users as well as the target brands. For example, users often mis-type the domain name of the website they want to visit in the address bar (*e.g.*, typing `facbook.com` for `facebook.com`). As a result, users could be visiting a website hosted under a squatting domain. Speculators register squatting domains of popular brands and resell them with a much higher price. Sometimes, popular brands (*e.g.*, big banks) have to purchase squatting domains that targeting their websites so that they can redirect users back to the correct websites [2].

Domain Squatting for Phishing. Squatting domains are naturally powerful to conduct phishing attacks since the domain name looks similar to the domain name of the trusted entities. We refer phishing pages hosted under squatting domains as *squatting phishing pages*. More formally, a squatting phishing page (P_s) is composed of two properties: (1) the domain of the page is a squatting-based domain (S); and (2) the webpage contains deceptive phishing content (W). $P_s = S \vee W$.

2.1 Research Questions

Our goal is to search and identify squatting phishing pages in the wild. Through empirical measurements, we seek to understand how attackers perform *impersonation* to deceive users and how they perform *evasion* to avoid being detected. To achieve these goals, we face two major technical challenges.

First, *a lack of comprehensive sources of squatting domains*. It is challenging to capture a comprehensive list of squatting domains that are potentially impersonating legitimate brands and online services. More specifically, we are not looking for a specific type of domain squatting, but aim to cover all different types of squatting domains. Current phishing blacklists rarely include squatting phishing pages. Later in §4.1, we show that most of the reported phishing URLs in PhishTank [9] do not have squatting domains.

Second, *a lack of effective phishing detection tools*. Phishing pages are constantly evolving. URL blacklisting is ineffective to detect zero-day phishing pages. In addition, our preliminary analysis

[1]We open-sourced our tool at https://github.com/SquatPhish.

shows that phishing pages have adopted evasion techniques that are likely to render existing detection methods ineffective (§4.2) . An efficient and yet evasion-resilient method is needed to detect squatting phishing pages.

Our Approaches. Instead of relying on phishing blacklists, we decide to search for *previous-unknown* squatting phishing pages in the wild. To do so, we develop a set of new tools for *squatting domain detection* and *phishing page classification.* More specifically, we select a large number of popular brands which are often targeted (impersonated) by phishing pages. Then we directly search for squatting domains that are likely to impersonate these brands from hundreds of millions of DNS records. We build a tool to effectively identify known types of squatting domains including homograph squatting, typo squatting, bit squatting, combo squatting and wrongTLD squatting.

To effectively detect phishing domains from the squatting domains, we build a novel machine learning classifier that takes advantage of image analysis and optical character recognition (OCR) to overcome page obfuscation. The classifier design is driven by empirical measurements of evasion methods used in practice (based on 4000+ manually labelled phishing pages). Once the classifier is trained, we use it to search for squatting phishing pages within a large number squatting domains. In the following, we describe each of the measurement steps and our key findings.

3 MEASUREMENT METHODOLOGY

In this section, we introduce our measurement methodology to search for candidate squatting domains. Then we introduce our data collection process to obtain their webpages (both web and mobile pages).

3.1 Squatting Detection

At the high-level, we first select a large set of popular brands and online services which are the potential impersonation targets of squatting phishing pages. Then we detect squatting domains for each brand from massive DNS records.

Brand Selection. Intuitively, popular brands are attractive targets. We select domains that are ranked high by Alexa [1]. More specifically, Alexa provides 17 categories such "business", "games", "health", "finance". For each category, we select the top 50 websites (850 domains in total). Then we search for brands that are popular targets of real-world phishing attacks. Based on the statistics from PhishTank [9], we obtain 204 brands (domains). For all the domains, we then merge some of them that share the same domain names (*e.g.*, merging niams.nih.gov and nichd.nih.gov into nih.gov). We merge those that are co-listed by PhishTank and Alexa. In total, we have 702 unique brands (domain names) that cover a wide range of different online services.

DNS Dataset. Next, we search the squatting domains of the target brands within a large collection DNS records. We obtained a snapshot of 224,810,532 DNS records from the ActiveDNS project [41] on September 6, 2017. ActiveDNS project uses multiple seeds to run active DNS probing, covering a number of top-level domains (*e.g.*, COM, NET, ORG) and other lists of domain collections (*e.g.*, Alexa Top 1M, Public Blacklists). Each record is characterized by a

Domain	Type
`faceb00k.pw`	homograph
`fàcebook.com` (punycode: `xn--fcebook-8va.com`)	homograph
`facebnok.tk`	bits
`facebo0ok.com`	typo
`fcaebook.org`	typo
`facebook-story.de`	combo
`facebook.audi`	wrongTLD

Table 1: Examples of different types of squatting domains for the `facebook` brand.

domain and an IP address. We use the 224 million domain names as the base to search for squatting domains in the next step.

Squatting Domain Identification. The most challenging step is to generate squatting domains for the target brands. Unfortunately, the state-of-the-art tools such as DNSTwist [3] and URL-crazy [15] are not very applicable to our case. First, existing tools are primarily designed to generated typo squatting and bits squatting domains. They cannot effectively handle combo squatting domains or domains that change the TLD. For example, URL-crazy can generate facebookj.com based on typo squatting for facebook.com, but would miss a domain facebookj.es that exists in our DNS records.

In addition, existing tools are very incomplete in detecting homograph domains. The most important type of homograph domains is the internationalized domain names that exploit the unicode confusion [14]. We find that tools like DNSTwist fail to map the complete list of similar unicode characters. For example, there are 23 different unicode characters that look similar to the letter "*a*" [14], but DNSTwist only catches 13 of them. These limitations will seriously hurt our chance of capturing squatting phishing pages.

To these ends, we develop our own system to capture squatting domains given a target brand. Our system is extended from DNSTwist and URL-crazy with (1) a more complete detection of homograph domains, (2) a new module to detect wrongTLD domains, and (3) a module to handle combo squatting domains [40]. Below, we provide details on the 5 types of squatting domains our system can capture. We use the real-world examples shown in Table 1 to explain each squatting type. We define the 5 types to be orthogonal from each other for the ease of measurement later.

- **Homograph:** Homograph based squatting refers to squatting domains that look similar to the target domains in the *vision perception.* For example, two characters "`rn`" can be used to impersonate the character "`m`". faceb00k is a homograph squatting to facebook since 00 looks similar to oo. More advanced homograph squatting exploit internationalized domain names (IDN). IDN utilizes Punycode encoding to convert unicode characters to ASCII. For example, `xn--fcebook-8va.com` is the homograph IDN. After IDN translation, the domain is presented as `fàcebook.com` in the browser's address bar.
- **Typo:** Typo squatting aims to mimic the incorrectly typed domain names by users. There are several methods to generate typo squatting based on a given target domain, including insertion (adding a character), omission (deleting a character), repetition (duplicating a character) and vowel swap (re-ordering two consecutive characters). Insertion refers

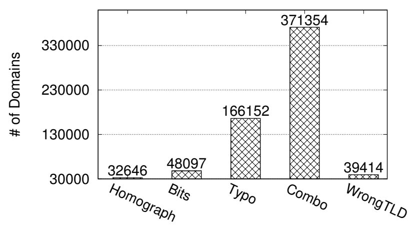


Figure 2: # of squatting domain of different squatting types.

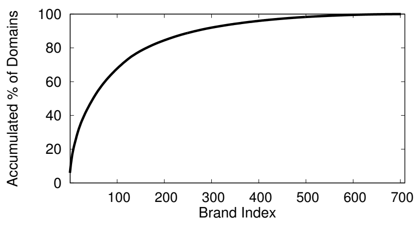


Figure 3: Accumulated % of squatting domains from top brands. Brands sorted by # of domains.

Brand	Squatting Domain	Percent
vice.com.	39,343	5.98%
porn.com.	18,149	2.76%
bt.com.	16,159	2.46%
apple.com	13,465	2.05%
ford.com.	12,163	1.85%

Figure 4: Top 5 brands with the most squatting domains.

to adding an additional character to the original domain. Omission refers to deleting a character in the domain. Repetition refers to repeating a character in the domain. Vowel swap refers to reordering two consecutive characters in the domain. For example, facebo0ok.com is a typo squatting domain by inserting 0. fcaebook.org is also a typo squatting domain by reordering *a* and *c* in the domain name.

- **Bits:** Bits squatting is to flip a bit of the domain name. A bits squatting domain is only one-bit different from the target domain. For example, facebnok.tk is bits squatting domain where one bit o is changed to n.
- **Combo:** Combo squatting is to concatenate the target domain name with other characters. The concatenation could be either attached to the head or tail. In our analysis, we particularly focus on the combo squatting with hyphens which are allowed in the domain name. For example, facebook-story is the combo squatting where new characters are attached to the tail of facebook with a hyphen.
- **WrongTLD:** All the above squatting techniques focus on the domain name but ignore the TLD. WrongTLD refers to domains that change the TLD but keep the domain name as the same. For example, facebook.audi belongs to the wrongTLD category since the original TLD com is changed to audi.

Domain Squatting Detection Results. For a given brand, we search through the DNS records to look for squatting domains. For each DNS domain, we check all 5 squatting rules against the target domain. If a match is found, we label the DNS domain with the squatting type. During the domain matching, we ignore subdomains. For example, mail.google-app.de is regarded as a combo squatting domain because the domain name google-app is a combo squatting of the target brand google.

In total, we detect 657,663 squatting domains for the 702 target brands. Figure 2 presents the distribution of different squatting types. Clearly, combo squatting is the most common type (56%). Intuitively, combo-squatting is easy to register since one can add arbitrary words to the original domain and connect them with a hyphen. Other squatting domains such as typo-squatting would be more competitive since there are only limited ways to impersonate the target domain name.

Figure 3 shows that the number of squatting domains per brand is highly skewed. More specifically, we sort the brands based on their number of squatting domains, and calculate the accumulated ratio of squatting domains that the top brands generated. We observe that the top 20 brands are responsible for more than 30% of the squatting domains. Note that the top brands here are not necessarily the most popular websites. Figure 4 presents the top 5 brands that matched the largest number of squatting domains. Typically, these domains either contains generic English word (*e.g.*, apple, vice) or the length is too short (*e.g.*, bt).

3.2 Web Crawling

To detect squatting phishing pages from a large number of squatting domains, we need to collect the web pages from each of the domains. At the high level, we aim to collect both their web version and mobile version of the pages to compare the potential differences. In addition, to assist our later classification tasks, we collect both the HTML source code and the screenshot for each page.

Crawler Design. To obtain the complete HTML content, we cannot simply query the static page using scripts like `curl`. Instead, we use headless browsers to load the dynamic content before saving the page. More specifically, we use the recently released Puppeteer [10], which is the headless Chrome. We have tested other alternative browsers such as Selenium [11, 31]. However, we find that Selenium is error-prone when crawling webpages at the million-level [5]. Given the high overhead of the large-scale dynamic crawling, we cannot exhaustively test all the possible browser versions and browser types. We choose a Chrome browser for its reliability. A potential limitation is that we might miss the cloaking websites that are specifically targeting IE explorer or other particular browsers. With Puppeteer, we build a distributed crawler to scan the 657K squatting domains and obtain the HTML content and take screenshots for the pages. Note that our crawling introduces almost no overhead to the target websites. Each website only receives 1-2 requests for each scan.

Web and Mobile Pages. For each domain, we capture both the web and mobile pages. We set "User-Agent" for iPhone 6 and Chrome 65 to obtain the mobile and web pages respectively. The data will help to analyze potential cloaking behavior or phishing pages that specifically target mobile or web users.

Redirections. Our crawler follows all the redirections when visiting each domain, and records the destination URLs. We save the HTML content and the screenshots for the webpages of the destination URLs.

Distributed Crawling. To speed up our crawling efficiency, we dispatch the crawling jobs to multiple CPU cores. The original Puppeteer does not support distributed crawling. To this end, we

Type	Live Domains	Domains w/ Redirections	Redirection Destination		
			Original	Market	Others
Web	362,545	316,620 (87.3%)	6,115 (1.7%)	10,734 (3.0%)	29,076 (8.0%)
Mobile	354,297	308,566 (87.1%)	6,486 (1.8%)	10,799 (3.1%)	28,446 (8.0%)

Table 2: Crawling statistics. We measure the redirections to the original website and those to domain marketplaces.

Brand	Domains w/ Redirection	Redirection Destination		
		Original	Market	Others
Shutterfly	32 (29%)	**76 (68%)**	3 (3%)	0 (0%)
Alliancebank	12 (35%)	**21 (62%)**	0 (0%)	1 (3%)
Rabobank	27 (33%)	**48 (61%)**	2 (3%)	2 (3%)
Priceline	135 (45%)	**157 (53%)**	1 (1%)	4 (1%)
Carfax	226 (50%)	**202 (45%)**	4 (1%)	20 (4%)

Table 3: Top brands with the highest ratio of redirections to their original websites.

Brand	Domains w/ Redirection	Redirection Destination		
		Original	Market	Others
Zocdoc	29 (19%)	3 (2%)	**118 (78%)**	1 (1%)
Comerica	58 (41%)	0 (0%)	**80 (57%)**	3 (2%)
Verizon	76 (45%)	0 (0%)	**83 (49%)**	10 (6%)
Amazon	1855 (36%)	1 (0%)	**2,168 (42%)**	1,185 (23%)
Paypal	706 (56%)	33 (3%)	**482 (38%)**	35 (3%)

Table 4: Top brands with highest ratio of redirections to domain marketplaces.

implement our own distributed crawling by allocating a kernel-level shared memory segment count. Each time, we fork a list of children processes and utilizes shmget in IPC (inter process communication) to balance the workload of each process. This allows us to the maximize the usage of CPUs for the web crawling. We run the crawler on 5 machines (24 cores, 196GB RAM) and open 20 Puppeteer simultaneously.

Web Crawling Statistics. From April 01 to April 08 in 2018, we collected one snapshot of the full 657,663 domains covering both the web and mobile pages. We use this snapshot to detect squatting phishing pages. From April 09 to April 29 in 2018, we collect three additional snapshots only for the detected squatting phishing pages (one week apart between consecutive snapshots). Table 2 provides the statistics for the full snapshot. For the web version, we find that 362,545 domains are live and reachable. For the mobile version, we obtain data from 354,297 live domains. Overall, about 55% of the squatting domains are live during the time of crawling. Among the live domains, we find that most of them (87%) have no redirection and 13% of the domains redirect the crawler to other domains.

Interestingly, 6,115 domains (1.7%) redirect the crawler to the original target domain. This indicates that the target brands indeed purchased squatting domains to redirect their users back to the correct websites. Table 3 shows the top brands whose squatting domains that the highest chance to redirect users to back to the original websites. Some of the top brands are related to sensitive services such as health (ZocDoc) and banking (Comerica, Alliancebank) These brands are likely to protect their users (and their reputation) by registering the squatting domains themselves.

In addition, we find some squatting domains will redirect users to some domain marketplaces where domain names are put out for sale (*e.g.*, marketmonitor). To measure the level of such redirection, we manually compiled a list of 22 known domain marketplaces. We find that 10,734 squatting domains (3%) would redirect users to one of the domain marketplaces. Table 4 shows top brands whose squatting domains have the highest chance to redirect users to domain markets. Not surprisingly, a large number squatting domains targeting popular brands such as Amazon and Paypal are listed on the market for sale. We find 2,168 Amazon squatting domains redirect to domain markets.

4 CHARACTERIZING EVASIONS

So far, we have collected a large set of squatting domains and their webpages. Next we aim to systematically detect squatting phishing pages. To develop an effective phishing detection system, we need to understand whether and how phishing pages are currently and actively evading common detection methods in practice. Such knowledge will help to design more reliable features to capture squatting phishing pages. In the following, we first collect and label ground-truth phishing pages and then perform a preliminary analysis their evasion techniques.

4.1 Ground Truth Phishing Pages

We first collect ground-truth phishing pages to understand evasion and train our machine learning classifiers. Here, we don't want to use any existing automated phishing detection tools to label the ground-truth since existing tools may be vulnerable to evasions. Instead, we rely on user reported and manually verified phishing pages as ground-truth. More specifically, we choose PhishTank [9] which is an online services that leverage crowdsourcing to collect phishing URLs. Any Internet users can submit phishing URLs and others can help to verify if the reported pages are truly phishing.

PhishTank Crawling. From February 2 to April 10 in 2018, we set up a crawler to crawl the phishing URLs under *all* 204 brand names provided by PhishTank. We ignore the brand named "other" since it does not specify the target brand. For each brand, our crawler checked the latest list 5 times a day to make sure we don't miss any newly submitted URLs. We focus on URLs that have been verified as phishing and URLs that are marked as "active". This allows us to *immediately* crawl the live phishing webpages under the reported URL. Same as before, for each Phishing URLs, we use a dynamic crawler to obtain its web and mobile pages and take screenshots for both pages.

In total, we crawled 6,755 unique phishing URLs under 138 brands. The other 66 brands do not have any URL submissions during our data collection period. As shown in Figure 5, the number of phishing pages per brand is highly skewed. The top 8 popular brands cover 4004 phishing URLs which counts for 59% of total phishing URLs.

Popularity and Squatting. To provide contexts for the phishing URLs, we first examine the ranking of their domains on Alexa

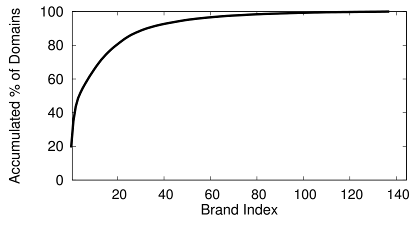


Figure 5: Accumulated % phishing URLs from top brands in PhishTank.

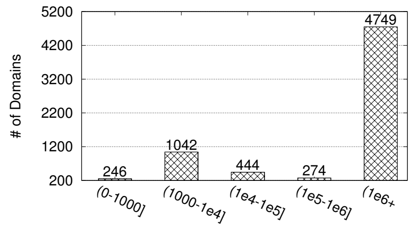


Figure 6: Alexa ranking of phishing URLs in PhishTank.

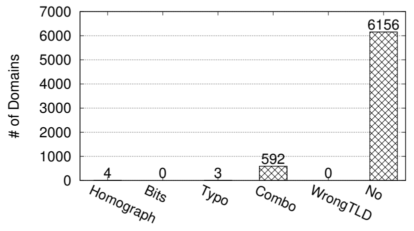


Figure 7: PhishTank squatting domains distribution.

Brand	# of URLs	Percent (%)	Valid Phishing
PayPal	1306	19.3	348
Facebook	1059	15.6	734
Microsoft	580	8.6	285
Santander UK	336	5.0	30
Google	218	3.2	95
Ebay	189	2.8	90
Aode	166	2.4	79
DropBox	150	2.2	70
SubTotal	4004	59.1	1731

Table 5: Top 8 brands in PhishTank cover 4004 phishing URLs (59.1%). Manual verification shows that 1731 pages are true phishing pages.

top 1 Million list. As shown in Figure 6, the vast majority (4749, 70%) of the phishing URLs are ranked beyond the Alexa top 1 million. This suggests most phishing pages are hosted on unpopular domains. A further analysis shows that 000webhostapp is most frequently used hosting domains for phishing pages (914 URLs) followed by sites.google and drive.google (140 URLs). The result suggests web hosting services have been abused by phishing.

We then analyze the squatting domains in the phishing URLs. As shown in Figure 7, the majority of phishing URLs are not squatting phishing — 6,156 (91%) of phishing URLs did not use squatting domains. In addition to the combo-squatting domains, we find one homograph squatting gooqle.online for google, one typo squatting paypals.center for paypal. There is no bits squatting or wrongTLD squatting in the PhishTank. This confirms that we cannot rely on phishing blacklists to study squatting phishing.

Ground Truth Labeling. Although the phishing URLs from PhishTank have been "validated", it is possible some of phishing pages have been replaced or taken-down when we crawl the pages. To this end, we cannot simply label all the crawled pages as "phishing". To obtain the ground-truth label, we select the top 8 brands (4,004 URLs, 59.1%) to manually examine the crawled pages (screenshots). As shown in Table 5, surprisingly, it turns out a large number of pages are no longer considered as phishing pages during the time of crawling. Only 1,731 out of 4,004 (43.2%) are still phishing pages. The rest 2,273 pages are no longer phishing pages (benign). Recall that our crawler has been monitoring the newly submitted URLs to PhishTank and immediately crawled their pages. The results suggest that phishing pages have a very short lifetime. Many phishing URLs have been taken-down or replaced with legitimate pages before the URLs are listed on PhishTank.

4.2 Evasion Measurement

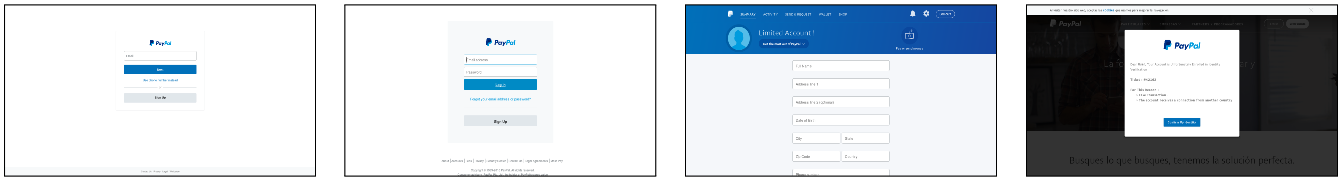
Based on the ground-truth data, we next examine the common evasive behavior of phishing pages. We will use the measurement results to derive new features to more robust phishing page detection. Our evasion measurement focuses on three main aspects: the image layout, the string text in the source code, and obfuscation indicators in the javascript code. These are common places where adversaries can manipulate the content to hide its malicious features, while still giving the web page a legitimate look. For this analysis, we focus on web version of the pages. We find that 96% of the pages on PhishTank have the same underlying HTML sources for both the web and mobile versions. This indicates that the most attackers did not show different pages to web and mobile users (*i.e.* no cloaking).

Layout Obfuscation. Many phishing detection methods assume that the phishing pages will mimic the legitimate pages of the target brands. As a result, their page layout should share a high-level of similarity [47]. Phishing detection tools may apply some fuzzy hashing functions to the page screenshots and match them against the hash of the real pages. To examine the potential evasions against page layout matching, we compute the Image hash [6] to compare the visual similarity of the phishing pages and the real pages of the target brands. The (dis)similarity is measured by the hamming distance between two image hashes.

We find that layout obfuscation is widely applied, and phishing pages often change their layout greatly to evade detection. Figure 8 shows a real example in our dataset for brand paypal. The left-most page is the official paypal page. The other 3 pages are phishing pages with different image hash distances 7, 24 and 36 respectively compared to the real pages. With a distance of 7, the phishing page is still visually similar to the original page. When the distance goes to 24 and 36, the pages look different from the original pages but still have a legitimate looking. Those pages would be easily missed by visual similarity based detectors.

Figure 9 shows the average image hash distance to the original pages for all phishing pages of different brands. We show that most brands have an average distance around 20 or higher, suggesting that layout obfuscation is very common. In addition, different brands have a different level of visual similarity, which makes it difficult to set a universal threshold that works for all the brands. These evasion steps would likely to render visual similarity based detection methods ineffective.

String Obfuscation. String obfuscation is hiding important text and keywords in the HTML source code. For example, attackers



(a) The original paypal page. (b) Phishing page (distance 7). (c) Phishing page (distance 24). (d) Phishing page (distance 38).

Figure 8: An example of page layout obfuscation of phishing pages (paypal).

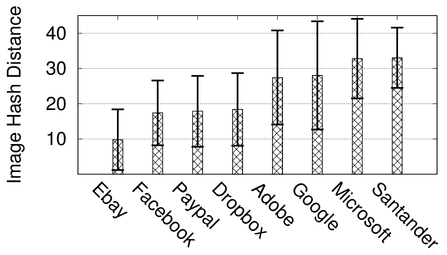


Figure 9: Average Image hash distance and standard variance for phishing pages of different brands.

Brand	String Obfuscated	Code Obfuscated
Santander	30 (100%)	4 (13.3%)
Microsoft	200 (70.2%)	127 (44.6%)
Adobe	38 (48.1%)	15 (18.9%)
Facebook	259 (35.3%)	342 (46.6%)
Dropbox	16 (22.9%)	1 (1.5%)
PayPal	61 (17.5%)	140 (40.2%)
Google	10 (10.5%)	11 (11.6%)
Ebay	8 (8.9%)	9 (10.0%)

Table 6: String and code obfuscation in phishing pages.

may want to hide keywords related to the target brand names to avoid text-matching based detection [39]. For example, in a phishing page that impersonates paypal, we find that the brand name string is obfuscated as "PayPaI", where the "l" (the lower case of "L") is changed to "I" (the upper case of "i"). Another common technique is to delete all related text about the brand name paypal but instead put the text into images to display them to users. From the users' perspective, the resulting page will still look similar.

We perform a simple measurement of text string obfuscation by looking for the brand name in the phishing pages' HTML source. Given a phishing page (and its target brand), we first extract all the texts from the HTML source. If the target brand name is not within the texts, then we regard the phishing page as a string obfuscated page. Table 6 shows the percentage of string obfuscated pages for each brand. For example, 70.2% of `microsoft` phishing pages are string obfuscated. 35.3% of facebook phishing pages are string obfuscated. This suggests that simple string matching is less likely to be effective.

Code Obfuscation. Javascript code may also apply obfuscation to hide their real purposes. This is a well-studied area and we use known obfuscation indicators to measure the level of code obfuscation in the phishing pages. Obfuscation indicators are borrowed from FrameHanger [59]. According to previous studies [38, 63], string functions (*e.g.*, `fromChar` and `charCodeAt`), dynamic evaluation (*e.g.*, `eval`) and special characters are heavily used for code obfuscation. For each phishing page, we download and parse the JavaScript code into an AST (abstract syntax tree). We then use AST to extract obfuscation indicators.

Table 6 presents the percentage of phishing pages that contain obfuscation indicators. Since we focus on strong and well-known indicators only, the results are likely to represent a lower bound of code obfuscation in phishing. For example, we find that some `Adobe` phishing pages adopt php script "action.php" for login forms. The script is invoked from a php file stored in a relative path. Automated analysis of php code (in a relative path) to detect obfuscation is a challenging problem itself.

5 MACHINE-LEARNING DETECTION

After understanding the common evasion techniques, we now design a new machine learning based classifier to detect squatting phishing pages. The key is to introduce more reliable features. Below, we first introduce our feature engineering process and then we train the classifier using the ground-truth data obtained from PhishTank. Finally, we present the accuracy evaluation results.

5.1 Feature Engineering

Based on the analysis in §4.2, we show that visual features, text-based features and javascript based features can be evaded by obfuscations. We need to design new features to compensate for existing ones. More specifically, we are examining squatting domains that are already suspicious candidates that attempt to impersonate the target brands. Among these suspicious pages, there are two main hints for phishing. First, the page contains some keywords related to the target brands either in the form of plaintext, images, or dynamically generated content by Javascripts. Second, the page contains some "forms" to trick users to enter important information. For example, this can be a login form to collect passwords or payment forms to collect credit card information.

To overcome the obfuscations, our intuition is that no matter how the attackers hide the keywords in the HTML level, the information will be *visually* displayed to users to complete the deception. To this end, we extract our main features from the *screenshots* of the suspicious pages. We use optical character recognition (OCR) techniques to extract text from the page screenshots to overcome the text and code level obfuscations. In addition, we will still extract traditional features from HTML considering that some phishing pages may not perform evasion. Finally, we consider features extracted from various submission "forms" on the page. All these features are independent from any specific brands or their original pages. This allows the classifier to focus on the nature of phishing.

Image-based OCR Features. From the screenshots, we expect the phishing page to contain related information in order to deceive users. To extract text information from a given page screenshot, we use OCR (Optical character recognition), a technique to extract text from images. With the recent advancement in computer vision and deep learning, OCR's performance has been significantly improved in the recent years. We use the state-of-the-art OCR engine Tesseract [13] developed by Google. Tesseract adopts an adaptive layout segmentation method, and can recognizes texts of different sizes and on different backgrounds. According to Google, the recent model has an error rate below 3% [12], which we believe this is acceptable for our purpose. By applying Tesseract to the crawled screenshots, we show that Tesseract can extract text such as "paypal" and "facebook" directly from the logos areas of the screenshots. More importantly, from the login form areas, it can extract texts such as "email" and "password" from the input box, and even "submit" from the login buttons. We treat the extracted keywords as *OCR features*.

Text-based Lexical Features. We still use text based features from HTML to complement OCR features. To extract the lexical features, we extract and parse the text elements from the HTML code. More specifically, we focus on the following HTML tags: h tag for all the texts in the headers, p tag for all the plaintexts, a tag for texts in the hyperlinks, and title tag for the texts in the title of the page. We do not consider texts that are dynamically generated by JavaScript code due to the high overhead (which requires dynamically executing the javascript in a controlled environment). We treat these keywords as *lexical features*.

Form-based Features. To extract features from data submission forms, we identify forms from HTML and collect their attributes. We focus on 4 form attributes: type, name, submit and placeholder. The placeholder attribute specifies a short hint for the input box. Often cases, placeholder shows hints for the "username" and "password" in the phishing pages, *e.g.*, "please enter your password", "phone, email or username". The name attribute specifies the name of the button. We treat the texts extracted from the form attributes as features. We also consider the number of forms in the HTML document as a feature.

Features that We Did Not Use. Prior works have proposed other features but most of which are not applicable for our purpose. For example, researchers of [20, 29, 61] also considered OCR and lexical features, but the underlying assumption is that phishing sites share a high level similarity with the real sites (visually or keyword-wise). However, this assumption is not necessarily true given the evasion techniques and the large variances of phishing pages (§3.1). In addition, Cantina [64] and Cantina+ [61] propose to query search engines (*e.g.*, Google) using the keywords of the suspicious web pages to match against the real sites. However, these features are too expensive to obtain given the large scale of our dataset. To these ends, the features we constructed in this paper (*e.g.*, keywords from logos, login forms, and other input fields) are more generic and light-weighted for modeling the phishing attempts

Discussions on the Feature Robustness. So far, we haven't seen any real-world phishing pages that attempt to evade the OCR engine. Future attackers may attempt to add adversarial noises to images to manipulate the OCR output. However, technically speaking, evading OCR features are difficult in the phishing contexts. First, unlike standard image classifiers that can be easily evaded [25, 32, 34, 43, 48, 54, 62], OCR involves a more complex segmentation and transformation process on the input images before the text extraction. These steps make it extremely difficult to reverse-engineer a blackbox OCR engine to perform adversarial attacks. A recent work confirms that it is difficult to evade OCR in a blackbox setting [57]. Second, specifically for phishing, it is impossible for attackers to add arbitrary adversarial noises to the *whole screenshots*. Instead, the only part that attackers can manipulate is the actual images loaded by the HTML. This means texts of the login forms and buttons can still be extracted by OCR or from the form attributes. Finally, for phishing, the key is to avoid alerting users, and thus the adversarial noise needs to be extremely small. This further increases of the difficulty of evasion. Overall, we believe the combination of OCR features and other features helps to increases the performance (and the robustness) of the classifiers.

5.2 Feature Embedding and Training

After the raw features are extracted, we need to first process and normalize the features before used them for training. Here, we apply NLP (natural language processing) to extract meaningful keywords and transform them into training vectors.

Tokenization and Spelling Checking. We first use NLTK [22], a popular NLP toolkit to tokenize the extracted raw text and then remove the stopwords [8]. Since the OCR engine itself would make mistakes, we then apply spell checking to correct certain typos from OCR. For example, Tesseract sometimes introduces errors such as "`passwod`", which can be easily corrected to "`password`" by a spell checker. In this way, we obtain a list of keywords for each page.

Feature Embedding. Next, we construct the feature vector. For numeric features (*e.g.*, number of forms in HTML), we directly append them to the feature vector. For keyword-related features, we use the frequency of each keyword in the given page as the feature value. During training, we consider keywords that are frequently appear in the ground-truth phishing pages as well as the keywords related to all the 766 brand names. The dimension of feature vector is 987 and each feature vector is quite sparse.

Classifiers. We tested 3 different machine learning models including Naive Bayes, KNN and Random forest. These models are chosen primarily for efficiency considerations since the classifier needs to quickly process millions of webpages.

5.3 Ground-Truth Evaluation

We use the ground-truth phishing pages from PhishTank to evaluate the classifier's performance. The classifier is trained to detect whether a page is phishing (positive) or not (negative). Recall that in § 4.1, there is no major difference in the HTML code for web and mobile pages, we only use the web version to perform the training.

The ground-truth dataset contains 1731 manually verified phishing pages from PhishTank. The benign categories contain 3838 webpages from two sources: the first part of 2273 benign pages were manually identified from the PhishTank dataset (§4.1); The

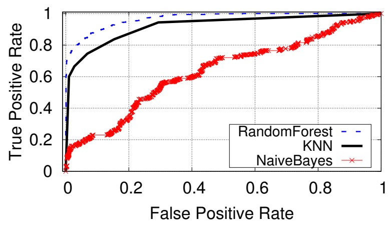


Figure 10: False positive rate vs. true positive rate of different models.

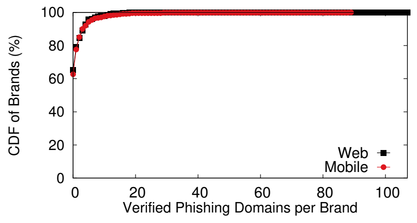


Figure 11: # of verified phishing domains for each brand.

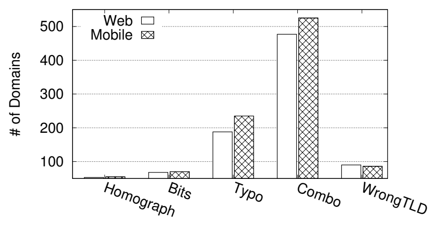


Figure 12: # of squatting phishing domains of different squatting types.

Algorithm	False Positive	False Negative	AUC	ACC
NaiveBayes	0.50	0.05	0.64	0.44
KNN	0.04	0.10	0.92	0.86
RandomForest	0.03	0.06	0.97	0.90

Table 7: Classifiers' performance on ground-truth data.

Type	Squatting Domains	Classified as Phishing	Manually Confirmed	Related Brands
Web	657,663	1,224	857 (70.0%)	247
Mobile	657,663	1,269	908 (72.0%)	255
Union	657,663	1,741	1175 (67.4%)	281

Table 8: Detected and confirmed squatting phishing pages.

second part of benign pages come from the webpages of the 1.6 million squatting domains (§3.2). We randomly sampled and manually verified 1565 benign pages. Due to the time-consuming nature of manual annotation, we only introduce the most "easy-to-confuse" benign pages (*i.e.*, those under squatting domains and those incorrectly reported as phishing). We did not include the "obviously benign pages" so that the classifiers can be more focused to distinguish the benign pages from the squatting domain set.

Table 7 shows the results of 10-fold cross-validation. We present the false positive rate, false negative rate, area under curve (AUC) and accuracy (ACC). We show that Random Forest has the highest AUC (0.97), with a false positive rate of 0.03 and a false negative rate 0.06. The classifier is highly accurate on the ground-truth dataset. Figure 10 presents the ROC curve of three algorithms. Random Forest achieves the best performance, and will be used to detect squatting phishing domains from the squatting domains.

6 SQUATTING PHISHING IN THE WILD

In this section, we apply our classifier to detect squatting phishing pages in the wild. We first describe our detection results and manually the confirm the flagged phishing pages. Then we analyze the squatting phishing pages to answer the following questions. First, how prevalent are phishing pages among the squatting domains? Second, what are the common attacks that squatting phishing pages are used for, and what types of squatting techniques are used? Third, are squatting phishing pages more evasive? How quickly can squatting phishing pages be detected or blacklisted?

6.1 Detecting Squatting Phishing Pages

We apply the Random Forest classifier to the collected web and mobile pages from the squatting domains. As shown in Table 8,

Brand	Squatting Domains	Predicted		Manual Verfied	
		Web	Mobie	Web (%)	Mobile (%)
Google	6,801	112	97	105 (94%)	89 (92%)
Facebook	3,837	21	24	18 (86%)	19 (80%)
Apple	13,465	20	22	8 (40%)	16 (72%)
BitCoin	1,378	19	17	16 (84%)	16 (94%)
Uber	5,963	16	16	11 (69%)	11 (69%)
Youtube	3,162	16	15	4 (25%)	12 (80%)
PayPal	2,330	14	17	7 (50%)	7 (41%)
Citi	5,123	10	19	8 (80%)	11 (58%)
Ebay	3,109	8	8	5 (63%)	5 (63%)
Microsoft	3,039	7	2	5 (71%)	2 (100%)
Twitter	1,378	7	5	4 (57%)	5 (100%)
DropBox	516	5	3	3 (60%)	2 (67%)
GitHub	503	6	4	5 (83%)	2 (50%)
ADP	3,305	6	7	3 (50%)	3 (43%)
Santander	567	1	1	1 (100%)	1 (100%)

Table 9: 15 example brands and verified phishing pages.

the classifier detected 1,224 phishing pages for the web version, and 1,269 phishing pages for the mobile version. Comparing to the 657,663 squatting domains, the number of squatting phishing pages are relatively small (0.2%).

Manual Verification. After the classification, we manually examined each of the detected phishing pages to further remove classification errors. During our manual examination, we follow a simple rule: if the page impersonates the trademarks of the target brands and if there is a form to trick users to input personal information, we regard the page as a phishing page. As shown in Table 8, after manual examination, we confirmed 1,175 domains are indeed phishing domains. Under these domains, there are 857 web phishing pages which count for 70.0% of all flagged web pages by the classifier. In addition, we confirmed even more mobile phishing pages (908) which count for 72.0% of all flagged mobile pages.

In Table 9, we present 15 example brands and the number of confirmed squatting phishing pages. We show the detection accuracy of the classifier is reasonably high for popular brands such as Google, Facebook, and Microsoft. However, the classifier is more likely to make mistakes on brands such as Paypal, Twitter, and Uber. Our manual analysis shows that the errors largely come from legitimate pages that contain some submission forms (*e.g.*, survey text boxes to collect user feedback) or third-party plugins of the target brands (*e.g.*, plugins for supporting payments via PayPal, Twitter "share" icons, Facebook "Like" buttons). The results suggest that classifier trained on the ground-truth dataset is still not perfect. Since the

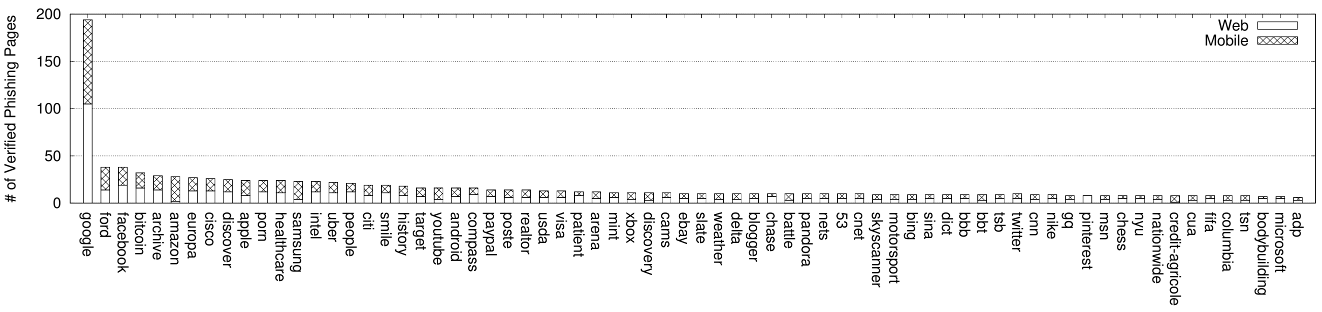


Figure 13: The top 70 brands targeted by squatting phishing pages.

Brand	Squatting Phishing Domains	Squatting Type
Google	goog1e.nl	Homograph
	gougle.pl	Homograph
	googl4.nl	Typo
	gooogle.com.uyl	Typo
	ggoogle.in	Typo
	googlw.it	Bits
	goofle.com.ua	Bits
	goofle.com.ua	Bits
Facebook	facebooκ.com	Homograph
	faceb00k.bid	Homograph
	facebouk.net ●	Homograph
	faceboook.top	Typo
	face-book.online	Typo
	fakebook.link	Typo
	faebook.ml	Typo
	faceboolk.ml ○	Typo
	facecook.mobi	Bits
	facebook-c.com	Combo
Apple	apple-prizeuk.com	Combo
Bitcoin	get-bitcoin.com	Combo
Uber	go-uberfreight.com	Combo
Youtube	you5ube.com	Typo
Paypal	paypal-cash.com	Combo
	paypal-learning.com	Combo
Citi	securemail-citizenslc.com	Combo
Ebay	ebay-selling.net	Combo
	ebay-auction.eu	Combo
Microsoft	formateurs-microsoft.com	Combo
	live-microsoftsupport.com ●	Combo
Twitter	twitter-gostore.com	Combo
Dropbox	drapbox.download	Homograph
	dropbox-com.com	Combo
ADP	mobile-adp.com	Combo
Santander	santander-grants.com	Combo

Table 10: Selected example phishing domains for 15 different brands. Note that "●" means web page only. "○" means mobile page only. The rest have both web and mobile pages.

testing data is orders of magnitude larger, it is possible that certain variances are not captured during the small-scale training. A potential way of improvement is to feed the newly confirmed phishing pages back to the training data to re-enforce the classifier training (future work).

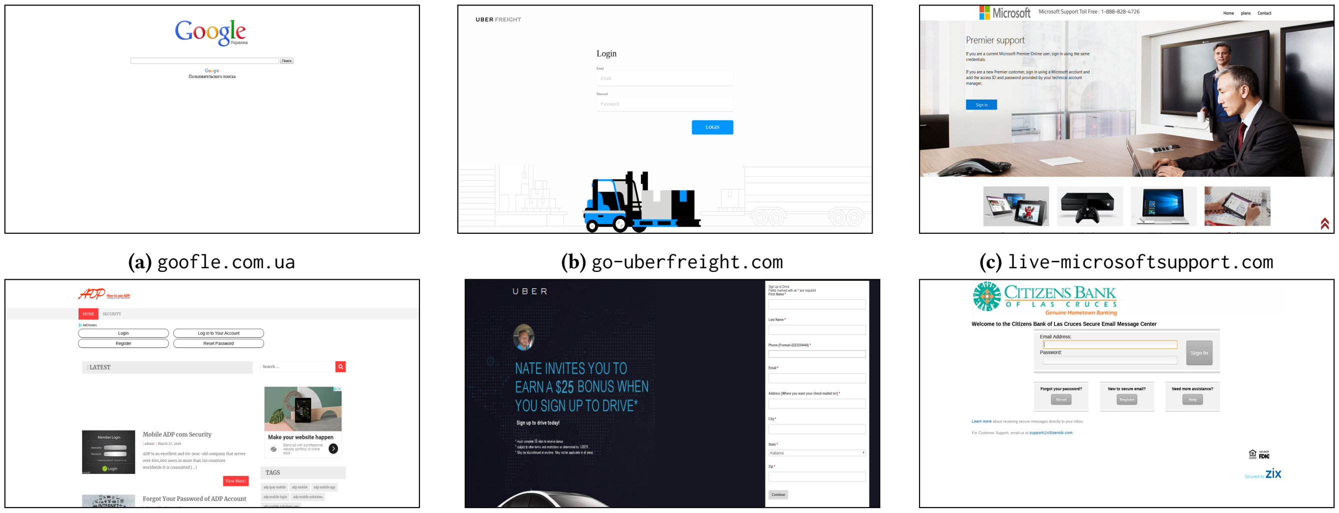
Targeted Brands. As shown in Table 8, the confirmed phishing pages are targeting 281 brands (247 brands on the web, and 255 brands on the mobile version). The rest of the 421 brands do not have squatting phishing pages under their squatting domains. Figure 11 shows the number of verified phishing pages for each brand. We show the vast majority of brands have fewer than 10 squatting phishing pages. Most brands are impersonated by tens of squatting phishing pages.

To illustrate the brands that are highly targeted by squatting phishing domains, we plot Figure 13. We observe that `google` standout as the mostly impersonated brands with 194 phishing pages across web and mobile. `Google`'s number if much higher than the second and third brands which all have 40 or below squatting phishing pages. We observe the popular brands such as `ford`, `facebook`, `bitcoin`, `amazon`, and `apple` are among the heavily targeted brands. Figure 14 shows a few example squatting phishing pages that mimic the target brands at both the content level and the domain level.

Mobile vs. Web. An interesting observation is that mobile and web does not have the same number of phishing pages. There are more mobile phishing pages. This indicates a *cloaking* behavior — the phishing websites only respond to certain types of user devices. Among the 1175 phishing domains, only 590 domains have both web and mobile phishing pages. 318 domains only show phishing pages to mobile users but not to web users; 267 domains return phishing pages to web users only. A possible reason for attackers to target mobile users is that mobile browsers do not always show the warning pages like the web browsers. During manual analysis, we used a Chrome browser on the laptop and a mobile Chrome browser to visit the confirmed phishing domains. The laptop Chrome is more likely to show the alert page compared to the mobile browser for the same domain. We also tested the laptop and mobile version of Safari and observed the same phenomenon.

As a related note, recent studies show that mobile browsers' UI design could make users more vulnerable to phishing [44, 52]. For example, mobile browsers often cannot fully display very long URLs in the address bar, and thus only show the leftmost or the rightmost part to users. This design limits a user's ability to examine the domain name of the (phishing) URL. In our case, we only a few long domain names from the 1175 phishing domains. For example, the longest domain name is "`buy-bitcoin-with-paypal-paysafecard-credit-card-ukash.com`" which has 57 characters.

IP Location. We further examine the geolocation of the IP addresses of the phishing domains. In total, we are able to look


Google
UBER FREIGHT
Login
Microsoft
Premier support
UBER
NATE INVITES YOU TO EARN A $25 BONUS WHEN YOU SIGN UP TO DRIVE*
CITIZENS BANK OF LAS CRUCES

(a) `goofle.com.ua` **(b)** `go-uberfreight.com` **(c)** `live-microsoftsupport.com`

(d) `mobile-adp.com` **(e)** `driveforuber.com` **(f)** `securemail-citizenslc.com`

Figure 14: Examples of squatting phishing pages.

up the geolocation of 1,021 IP addresses, hosted in 53 different countries. Figure 15 shows the IP distribution in different countries and we highlight the top countries with the most IP addresses. These phishing sites are widely spread all over the world. The U.S. has most of the sites, followed by Germany (DE).

Domain Name Registration. Finally, we obtain the *whois* records of the phishing domain names and examine their registration time and registrars. As shown in Figure 16, most of the squatting phishing domain names were registered within the recent 4 years. Based on the whois records, only 738 domains contain the registrar information. We find that out of 121 different registrar institutions, the most popular registrar is `godaddy.com` with 157 squatting phishing domain names.

6.2 Squatting Types & Case Studies

Next, we examine the squatting methods used by squatting phishing domains. As shown in Figure 12, there are squatting phishing pages under each every squatting method. It is not too surprising that combo squatting domains contain the largest number of phishing pages since they are less competitive to register, *i.e.*, attackers can add arbitrary strings to the target brand names. We find over 200 phishing pages within homograph squatting domains, bits squatting domains and typo squatting domains, which are more difficult to register. Table 10 shows a few examples of the phishing domains of different squatting types. We select 6 examples and present their screenshots in Figure 14, and infer the motivations behind the squatting phishing pages.

Fake Search Engine. Figure 14a presents an interesting example of bits squatting. The phishing domain "`goofle.com.ua`" is trying to impersonate Google's search engine hosted in Ukraine "`google.com.ua`", by changing one character "g". A possible motivation of this page is to perform censorship to monitor what searching queries that Ukraine citizens are performing. Another (more likely) motivation is that this website impersonates Google search to serve specific advertisements to users. Through manual examination, we find that the fake search engine not only displays more advertisements, but the advertisements are also different from those on the real site, given the same searching query (the searching results are relatively consistent).

Offline Scam. Figure 14b shows an example of combo squatting. The squatting phishing domain is "`go-uberfreight.com`", which impersonates Uber Freight, a new service of Uber to connect truck drivers with shippers. The official site is `freight.uber.com`. The purpose of the phishing page is likely to steal truck drivers' Uber accounts. Note that truck drivers' accounts are very difficult to register which takes background checks and virtual/on-site interviews. It is possible that the attacker is trying to steal truck driver's account for *offline scams*, for example, to impersonate a Uber truck driver to pick up and steal valuable goods. Another related example is shown in Figure 14e where the phishing domain "`driveforuber.com`" is impersonating the official Uber site "`drive.uber.com`".

Payroll Scam. Figure 14d shows a payroll scam on ADP. ADP offers payroll services for employees of various companies. ADP's official mobile domain is "`mobile.adp.com`". The phishing page "`mobile-adp.com`" is impersonating the mobile page of ADP through combo squatting. Interestingly, the login form on the phishing page is dynamically loaded by a JavaScript. We find that the login form will show up only if a user did not have an adblocker.

Tech Support Scam. Figure 14c shows an example of tech support scam with a combo-squatting domain. The phishing domain "`live-microsoftsupport.com`" is impersonating the online support of Microsoft "`support.microsoft.com`". The page either tries to compromise a user's Microsoft account or tricks the user to call the listed phone number. For tech support scams, scammers behind the phone often guide the victim to install malware or pay the "service fee" [49].

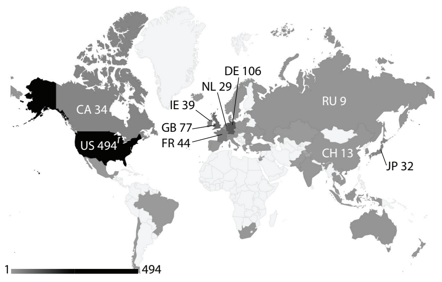


Figure 15: The location of squatting phishing websites.

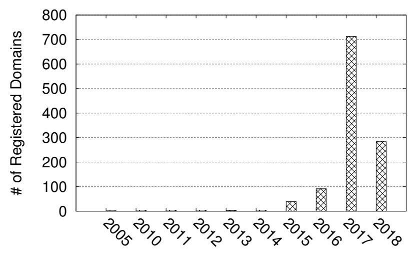


Figure 16: The registration time of squatting phishing domains.

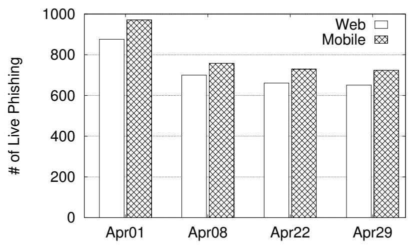


Figure 17: # of phishing pages within each snapshot.

Type	Layout Obfuscation	String Obfuscation	Code Obfuscation
Squatting-Web	28.4 ± 11.8	68.1%	34.0%
Squatting-Mobile	28.6 ± 11.6	68.2%	35.3%
Non-Squatting	21.0 ± 12.3	35.9%	37.5%

Table 11: Phishing pages that adopted evasion techniques.

Blacklist	PhishTank	VirusTotal	eCrimeX	Not Detected
Domains	0 (0.0%)	100 (8.5%)	2 (0.2%)	1,075 (91.5%)

Table 12: Detected squatting phishing pages by popular blacklists. VirusTotal contains 70+ blacklists.

Domain	April 01	April 08	April 22	April 29
facecook.mobi	Live	Live	Live	Live
facebook-c.com	Live	Live	Live	Live
face-book.online	Live	Live	Live	Live
facebook-sigin.com	Live	Live	Live	Live
faceboolk.ml	Live	Live	-	-
tacebook.ga	**Live**	**Live**	**-**	**Live**

Table 13: The liveness of phishing pages on different dates.

Stealing Payment Accounts. More commonly, squatting phishing pages aim to compromise user accounts at payment services. For example, Figure 14f is a phishing page that impersonates Citizens Bank's official page "`citizenslc.com`". The phishing domain is a combo squatting domain "`securemail-citizenslc.com`".

6.3 Evasion

We next examine whether and how squatting phishing pages would perform evasion against common detection techniques.

Squatting vs. Non-Squatting Phishing. We first examine whether squatting phishing pages are more likely to adopt evasion techniques. In Table 11, we directly compare the verified squatting phishing pages with non-squatting phishing pages labelled in PhishTank (URLs without squatting domains). As shown in Table 11, squatting phishing pages have a higher level of layout obfuscation. In addition, there is a higher ratio of squatting phishing pages that adopted string obfuscation (68.1%–68.2%) than that of non-squatting phishing pages (35.9%). Code obfuscation is less common among squatting phishing pages.

Evading Popular Blacklists. The phishing pages detected by our system are largely previous-unknown phishing pages. To examine how likely they can evade existing blacklist, we perform a quick test. As shown in Table 12, we run the list of verified squatting phishing domains against several popular phishing blacklists in May 2018. First, we checked the PhishTank and find that only 2 of our squatting phishing domains have been reported (0.1%). Then we query VirusTotal [16], which contains over 70 different blacklists. These 70 blacklists collectively marked 110 (8.2%) of squatting phishing domains. Finally, examine eCrimeX [4], a phishing blacklist maintained by the Anti Phishing Work Group (APWG). Their phishing URLs are gathered from a large number organizations around the globe. Through collaboration, we obtained 335,246 phishing URLs reported during April 2017 to April 2018. In total, eCrimeX marked 4 squatting phishing domains (0.2%). Collectively these blacklists only detected 8.4% of the squatting phishing pages, which means 91.5% of the phishing domains remain undetected for at least a month. As a comparison, a recent study [33] shows that phishing pages hosted on *compromised* web servers typically last for less than 10 days before they are blacklisted. This suggests that squatting phishing domains are much more difficult to detect.

Lifetime of Squatting Phishing Pages. We also measure the longevity of phishing pages. Recall that for domains that are classified as phishing in the first snapshot, we continue to crawl their webpages every week for a month. For each snapshot, we re-apply our classifier to their pages and examine if they are still classified as phishing. The results are shown in Figure 17. Most pages (about 80%) still remain alive after at least a month. Only a small portion of the pages has been down after 1-2 weeks. This again confirms that squatting phishing pages are difficult to detect and take-down.

Table 13 presents the liveness of 6 phishing pages that impersonate Facebook. An interesting domain is tacebook.ga. In the third snapshot, we find that the webpage under this domain has been replaced with a benign page (manually verified). However, in the fourth snapshot, the phishing page come back again.

7 DISCUSSION

Detecting Squatting Phishing in Practice. In this paper, we demonstrate a systematic approach to search and detect squatting phishing pages. With a deep-level impersonation, squatting phishing domains do not come with a large number, but are likely to be used for highly targeted attacks. Our results have shown that squatting phishing pages are difficult to detect and take down — *91.6%* of them are still alive after at least a month.

Our system SquatPhi can be used in two ways. First, any *third-party organizations* can set up a scanner to constantly monitor the squatting domains for a *broad range of brands* to capture squatting phishing domains. Crowdsourcing efforts can be introduced to speed up the manual verification process. Note that we are searching needle in a haystack by narrowing down the target from hundreds of thousands squatting domains to several hundreds phishing candidates, which are then manageable for manual investigation. Second, *individual online services* can set up their own dedicated scanner to search for squatting phishing pages that impersonate their brands. For example, Paypal can keep monitoring the newly registered domain names to the DNS to identify PayPal related squatting domains and classify squatting phishing pages. The classifier can be potentially much more accurate if it is customized for one specific brand. We have open-sourced our tool at https://github.com/SquatPhish to propel future search in the community.

Reporting Phishing Websites. In September 2018, we checked PhishTank, eCrimeX and VirusTotal again. Among the 1,175 verified squatting domains, 1,075 of them are still online, and only 60 (5.1%) of them are blacklisted. We then reported the rest verified phishing websites to Google safe browsing (under VirusTotal). Like most blacklists, Google safe browsing does not support batch reporting, and has strict rate limits and CAPTCHAs to prevent abuse. We have to submit the malicious URLs one by one manually.

Our Limitations. Our study has a number of limitations. First, our crawler only sets two profiles for a specific version of iPhone (mobile) and Chrome (web). It is possible that we might have missed phishing pages that perform cloaking, *e.g.*, those that only target Microsoft Explorer users. Second, our measurement primarily focuses on “popular brand” based on Alexa ranking. As a future work, we can extend our measurement scope to specifically cover the web domains of government agencies, military institutions, universities, and hospitals to detect squatting phishing pages targeting important organizations. Third, technically, it is difficult to evade a blackbox OCR engine while creating highly deceptive phishing pages (see §5.1). Reverse-engineering OCR for adversarial attacks is out of the cope of this paper. We leave more detailed explorations to future work. Finally, we did not directly compare our phishing classifier with existing tools such as Cantina [64] and Cantina+ [61]. This is because most existing works did not open-source their tool, and some of their features are too expensive to obtain for large-scale datasets. In this paper, we open-sourced our tool to ensure the reproducibility of the results.

8 RELATED WORK

Squatting Domains Identification. Previous works have studied different types of squatting techniques [21, 50, 58]. For example, More et al. [50] measured typo squatting by generating a list of plausible misspellings of popular domains. Nikiforakis et al. [51] measured the bit squatting by generating a single bit-flip for a valid domain. Holgers et al. [35] characterized homograph squatting through character substitutions. Kinti et al. [40] measured combo squatting by searching domain keywords from DNS records. In this paper, we focus on aggregating and improving existing squatting methods to search for squatting phishing attacks.

Phishing Webpage Detection. A plethora of research has focused on blacklisting or content-based detection methods. For example, PhishTank [9] leverages crowdsourcing to collect phishing URLs that Internet users encountered. PhishEye [33] proposed to use honeypots to monitor live phishing pages. Other detection methods are based on visual similarities [47, 60] or lexical URL properties [23, 26, 45] to detect phishing pages. For example, DeltaPhish [27] detects compromised websites by comparing the page structure similarities. Cantina and Cantina+ [61, 64] are based on the keyword frequency and page rank information. Marchal et al. [46] also use keyword frequency in the HML pages. In this paper, we show how today’s phishing pages, especially squatting phishing pages, have adopted evasion techniques that are likely to render existing detectors ineffective. A recent system Meerkat [24] uses deep learning models to analyze visual elements in webpages to detect compromised websites. Our approach is different since we use OCR to extract the text from the screenshots rather than focusing on the visual elements. Note that researchers of [20, 29] used OCR to extract keywords and query search engines to match again the real sites. However, this design still assumes phishing sites are similar/identical to the target sites, which is not necessarily true given the big variances introduced by the evasion techniques. Instead, we focus on more generic keywords extracted from logos, login forms, and other input fields to model the “phishing" attempts, which turns out to be effective.

Phishing Emails and Hosting Servers. Phishing emails are used to distribute the phishing URLs. Attackers can impersonate trusted parties to send phishing emails via email spoofing [36, 37] or email header injection [55]. In addition to registering squatting domains, attackers can also compromise existing web servers to host the phishing pages [53].

9 CONCLUSION

In this paper, we perform an extensive measurement on squatting phishing, where the phishing pages impersonate target brands at both the domain and content level. By monitoring 700+ brands and 600K squatting domains for a month, we identified 857 phishing web pages and 908 mobile pages. We show that squatting phishing pages are impersonating trusted entities through all different domain squatting techniques. Squatting phishing pages are more likely to adopt evasion techniques and are hard to catch. About 90% of them have evaded the detection of popular blacklists for at least a month.

ACKNOWLEDGMENTS

We would like to thank our shepherd Zhiyun Qian, and the anonymous reviewers for their helpful feedback. We also want to thank Manos Antonakakis and Athanasios Kountouras from Georgia Tech who kindly shared their DNS dataset with us. This project was supported in part by NSF grants CNS-1750101 and CNS-1717028, and Google Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

REFERENCES

[1] Alexa top sites by category. https://www.alexa.com/topsites/category.

[2] Cyber squatters are targeting britain's biggest banks. https://goo.gl//.

[3] DNSTwist: domain name permutation engine. https://github.com/elceef/dnstwist/.

[4] eCrimeX. https://www.ecrimex.net/.

[5] How to make selenium tests reliable, scalable, and maintainable. https://news.ycombinator.com/item?id=9925951.

[6] Image hashing. https://github.com/jenssegers/imagehash/.

[7] Lookout advances mobile phishing protection amidst growing enterprise risk. https://goo.gl/NzBkSN/.

[8] NLTK: Natural language toolkit. https://www.nltk.org/.

[9] PhishTank. https://www.phishtank.com/.

[10] Puppeteer: Headless chrome node api. https://github.com/GoogleChrome/puppeteer/.

[11] Selenium automates browsers. https://www.seleniumhq.org/.

[12] Tesseract accuracy. https://github.com/tesseract-ocr/tesseract/wiki/4.0-Accuracy-and-Performance.

[13] Tesseract: open source ocr engine. https://github.com/tesseract-ocr/tesseract.

[14] Unicode confusables summary. http://www.unicode.org/Public/security/revision-03/confusablesSummary.txt.

[15] URL-crazy: Generate and test domain typos and variations. https://www.morningstarsecurity.com/research/urlcrazy.

[16] VirusTotal. https://www.virustotal.com/.

[17] Phishing activity trends report (4th quarter 2017). APWG, 2017.

[18] 2018 data breach investigations reprot. verizon Inc., 2018.

[19] 2018 internet security threat report. Symantec Inc., 2018.

[20] Afroz, S., and Greenstadt, R. Phishzoo: Detecting phishing websites by looking at them. In *Proc. of ICSC* (2011).

[21] Agten, P., Joosen, W., Piessens, F., and Nikiforakis, N. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proc. of NDSS* (2015).

[22] Bird, S., and Loper, E. NLTK: the natural language toolkit. In *Proc. of ACL* (2004).

[23] Blum, A., Wardman, B., Solorio, T., and Warner, G. Lexical feature based phishing url detection using online learning. In *Proc. of AISec* (2010).

[24] Borgolte, K., Kruegel, C., and Vigna, G. Meerkat: Detecting website defacements through image-based object recognition. In *Proc. of USENIX Security* (2015).

[25] Carlini, N., and Wagner, D. Towards evaluating the robustness of neural networks. In *Proc. of S&P (Okland)* (2017).

[26] Choi, H., Zhu, B. B., and Lee, H. Detecting malicious web links and identifying their attack types. In *Proc. of USENIX Conference on Web Application Development* (2011).

[27] Corona, I., Biggio, B., Contini, M., Piras, L., Corda, R., Mereu, M., Mureddu, G., Ariu, D., and Roli, F. Deltaphish: Detecting phishing webpages in compromised websites. In *Proc. of ESORICS* (2017).

[28] Cui, Q., Jourdan, G.-V., Bochmann, G. V., Couturier, R., and Onut, I.-V. Tracking phishing attacks over time. In *Proc. of WWW* (2017).

[29] Dunlop, M., Groat, S., and Shelly, D. Goldphish: Using images for content-based phishing analysis. In *Proc. of ICIMP* (2010).

[30] Egele, M., Stringhini, G., Kruegel, C., and Vigna, G. Towards detecting compromised accounts on social networks. In *Proc. of IEEE Transactions on Dependable and Secure Computing (TDSC)* (2017).

[31] Englehardt, S., and Narayanan, A. Online tracking: A 1-million-site measurement and analysis. In *Proc. of CCS* (2016).

[32] Grosse, K., Papernot, N., Manoharan, P., Backes, M., and McDaniel, P. Adversarial examples for malware detection. In *Proc. of ESORICS* (2017).

[33] Han, X., Kheir, N., and Balzarotti, D. Phisheye: Live monitoring of sandboxed phishing kits. In *Proc. of CCS* (2016).

[34] He, W., Wei, J., Chen, X., Carlini, N., and Song, D. Adversarial example defenses: Ensembles of weak defenses are not strong. In *Proc of WOOT* (2017).

[35] Holgers, T., Watson, D. E., and Gribble, S. D. Cutting through the confusion: A measurement study of homograph attacks. In *USENIX ATC* (2006).

[36] Hu, H., Peng, P., , and Wang, G. Towards understanding the adoption of anti-spoofing protocols in email systems. In *Proc. of SecDev* (2018).

[37] Hu, H., and Wang, G. End-to-end measurements of email spoofing attacks. In *Proc. of USENIX* (2018).

[38] Kaplan, S., Livshits, B., Zorn, B., Seifert, C., and Curtsinger, C. "nofus: Automatically detecting" + string.fromcharcode(32) + "obfuscated ".tolowercase() + "javascript code". Tech. Rep. MSR-TR-2011-57, Microsoft Research, May 2011.

[39] Khonji, M., Iraqi, Y., and Jones, A. Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials 15*, 4 (2013), 2091–2121.

[40] Kintis, P., Miramirkhani, N., Lever, C., Chen, Y., Romero-Gómez, R., Pitropakis, N., Nikiforakis, N., and Antonakakis, M. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proc. of CCS* (2017).

[41] Kountouras, A., Kintis, P., Lever, C., Chen, Y., Nadji, Y., Dagon, D., Antonakakis, M., and Joffe, R. Enabling network security through active dns datasets. In *Proc. of RAID* (2016).

[42] Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L. F., Hong, J., and Nunge, E. Protecting people from phishing: the design and evaluation of an embedded training email system. In *Proc. of CHI* (2007).

[43] Liang, B., Su, M., You, W., Shi, W., and Yang, G. Cracking classifiers for evasion: A case study on the google's phishing pages filter. In *Proc. of WWW* (2016).

[44] Luo, M., Starov, O., Honarmand, N., and Nikiforakis, N. Hindsight: Understanding the evolution of ui vulnerabilities in mobile browsers. In *Proc. of CCS* (2017).

[45] Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. Learning to detect malicious urls. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2011).

[46] Marchal, S., Saari, K., Singh, N., and Asokan, N. Know your phish: Novel techniques for detecting phishing sites and their targets. In *Proc. of ICDCS* (2016).

[47] Medvet, E., Kirda, E., and Kruegel, C. Visual-similarity-based phishing detection. In *Proc. of SecureComm* (2008).

[48] Meng, D., and Chen, H. Magnet: a two-pronged defense against adversarial examples. In *Proc of CCS* (2017).

[49] Miramirkhani, N., Starov, O., and Nikiforakis, N. Dial one for scam: A large-scale analysis of technical support scams. In *Proc. of NDSS* (2017).

[50] Moore, T., and Edelman, B. Measuring the perpetrators and funders of typosquatting. In *International Conference on Financial Cryptography and Data Security* (2010).

[51] Nikiforakis, N., Van Acker, S., Meert, W., Desmet, L., Piessens, F., and Joosen, W. Bitsquatting: Exploiting bit-flips for fun, or profit? In *Proc. of WWW* (2013).

[52] Niu, Y., Hsu, F., and Chen, H. iphish: Phishing vulnerabilities on consumer electronics. In *Proc. of UPSEC* (2008).

[53] Oest, A., Safei, Y., Doupe, A., Ahn, G.-J., Wardman, B., and Warner, G. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In *Proc of eCrime* (2018).

[54] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against deep learning systems using adversarial examples. In *Proc. of AsiaCCS* (2017).

[55] Prashanth Chandramouli, S., Bajan, P.-M., Kruegel, C., Vigna, G., Zhao, Z., Doup, A., and Ahn, G.-J. Measuring e-mail header injections on the world wide web. In *Proc. of SAC* (2018).

[56] Reaves, B., Scaife, N., Tian, D., Blue, L., Traynor, P., and Butler, K. R. Sending out an sms: Characterizing the security of the sms ecosystem with public gateways. In *Proc. of S&P (Okland)* (2016).

[57] Song, C., and Shmatikov, V. Fooling OCR systems with adversarial text images. *CoRR abs/1802.05385* (2018).

[58] Szurdi, J., Kocso, B., Cseh, G., Spring, J., Felegyhazi, M., and Kanich, C. The long" taile" of typosquatting domain names. In *Proc. of USENIX Security* (2014).

[59] Tian, K., Li, Z., Bowers, K., and Yao, D. FrameHanger: Evaluating and classifying iframe injection at large scale. In *Proc. of SecureComm* (2018).

[60] Wenyin, L., Huang, G., Xiaoyue, L., Min, Z., and Deng, X. Detection of phishing webpages based on visual similarity. In *Proc. of WWW* (2005).

[61] Xiang, G., Hong, J., Rose, C. P., and Cranor, L. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)* (2011).

[62] Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proc. of NDSS* (2018).

[63] Xu, W., Zhang, F., and Zhu, S. Jstill: mostly static detection of obfuscated malicious javascript code. In *Proc. of AsiaCCS* (2013).

[64] Zhang, Y., Hong, J. I., and Cranor, L. F. Cantina: a content-based approach to detecting phishing web sites. In *Proc. of WWW* (2007).