

On Cloaking Behaviors of Malicious Websites

Nayanamana Samarasinghe*, Mohammad Mannan

Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada

Abstract

Malicious websites often mimic top brands to host malware and launch social engineering attacks, e.g., to collect user credentials. Some such sites often attempt to hide malicious content from search engine crawlers (e.g., Googlebot), but show harmful content to users/client browsers—a technique known as *cloaking*. Past studies uncovered various aspects of cloaking, using selected categories of websites (e.g., mimicking specific types of malicious sites). We focus on understanding cloaking behaviors using a broader set of websites. As a way forward, we built a crawler to automatically browse and analyze content from 100,000 *squatting* (mostly) malicious domains—domains that are generated through typo-squatting and combo-squatting of 2883 popular websites. We use a headless Chrome browser and a search-engine crawler with user-agent modifications to identify cloaking behaviors—a challenging task due to dynamic content, served at random; e.g., consecutive requests serve very different malicious or benign content. Most malicious sites (e.g., phishing and malware) go undetected by current blacklists; only a fraction of cloaked sites (127, 3.3%) are flagged as malicious by VirusTotal. In contrast, we identify 80% cloaked sites as malicious, via a semi-automated process implemented by extending the content categorization functionality of Symantec’s *SiteReview* tool. Even after 3 months of observation, nearly a half (1024, 45.4%) of the cloaked sites remained active, and only a few (31, 3%) of them are flagged by VirusTotal. This clearly indicate that existing blacklists are ineffective against cloaked malicious sites. Our techniques can serve as a starting point for more effective and scalable early detection of cloaked malicious sites.

1. Introduction

Websites are often used to launch social engineering attacks. For example, phishing websites exploit human weaknesses to steal sensitive user information; similarly, malware websites employ techniques to deceive users to download malware (e.g., ransomware) infecting user machines; cyber-criminals take advantage of ads hosted on low-tier networks using social engineering techniques [1]. Sophisticated phishing and malware websites hosted on *squatting* domains are deployed to deceive users by impersonating websites of high profile companies and

organizations (the so-called *elite* phishing domains [2, 3]). The domains hosting these phishing sites are subjected to typo-squatting (e.g., foxnews.com) and combo-squatting (e.g., support-apple.com-identify.us). These phishing sites impersonate trusted brand names using fake web content and typo-squatted domain names.

Additionally, phishing and malicious sites employ evasion techniques to avoid exposing malicious content to search engine crawler as opposed to human users [4, 5, 6]. The practice of displaying different content to a crawler as opposed to a browser/user is known as *cloaking*. Cloaking helps attackers to reduce the possibility of getting their services blacklisted. To discourage such practices, search engine providers also offer guidelines for website owners/maintainers—see e.g., Google [7].

There have been several studies on malware and phish-

*Corresponding author
Email address: n_samara@ciise.concordia.ca (Nayanamana Samarasinghe)

ing sites, albeit not so much on *squatting/elite* phishing and malicious domains engaged in cloaking. Past studies on cloaked malicious sites relied on specific types of websites and attacks (e.g., payment sites and phishing). Tian et al. [2] found 1175 (0.18%) phishing sites that are likely impersonating popular brands from 657,663 squatting domains (extracted from a collection of 224 million DNS records listed in ActiveDNS project [8]). They focused mainly on phishing web pages identified using specific keywords from logos, login forms, and other input fields (mostly credential phishing). Invernizzi et al. [6] studied web cloaking resulting from blackhat search engine optimizations and malicious advertising, using websites relating to luxury storefronts, health, and software. Oest et al. [9, 10, 11] used crafted PayPal-branded websites, and impersonated websites targeting a major financial service provider to study phishing. As such, the data sets used in these past studies do not cover a wide variety of malicious URLs. Our methodology also includes capturing cloaking in dynamic elements (e.g., iframes) of websites and taking semantics of web content into consideration, which were not adequately addressed in the past; e.g., Tian et al. [2] did not consider dynamically/JavaScript-generated page content due to high overhead.

We focus on understanding cloaking behaviors of a broad set of malicious sites hosted on squatting domains. These sites engage in phishing, malware distribution, and other social engineering attacks. We use DNSTwist [12] to generate already registered squatting domains that are potentially malicious. DNSTwist uses fuzzy hashing to identify malicious squatting domains by comparing its web page content with the corresponding seed domain. The squatting domains extracted from DNSTwist host content from a wide variety of possible malicious websites. To verify the ground truth of malicious squatting sites generated from DNSTwist, we adopt a semi-automated process leveraging the Symantec SiteReview [13] tool, which significantly outperformed both commercial and academic tools (e.g., VirusTotal [14], Off-the-Hook [15]) in our manual tests; cf. Vallina et al. [16].

We compare page content between a search engine crawler and browser client to detect cloaked malicious websites. For this purpose, we develop a crawler to collect page source, links, content, screenshots, headers from websites hosted on squatting domains. To distinguish be-

tween dynamic vs. cloaked pages, we employ a set of heuristics; see Section 3.3.5. To mimic a regular user browser (Chrome) and a search engine crawler (Google), we simply rely on custom browser user-agents and referrer headers. For the remainder of this paper, we use *GooglebotUA*, *ChromeUA*, *ChromeMobileUA* for search engine crawler, browser (desktop) and browser (mobile) user-agents interchangeably. Attackers may also leverage various evasion techniques to obfuscate the page-layout and HTML source, e.g., keywords in response headers to trick a search engine crawler [2], manipulate visual similarity between a phishing and a corresponding benign site [5]. Hence, we also examine the extent of such obfuscation in cloaked malicious websites.

Out of the 100,000 squatting domains (i.e., domain list category *A* in Table 1), VirusTotal flagged only 2256 (2.3%) domains as malicious—in contrast to the ground truth (74%), as verified via our semi-automated process. From the 100,000 squatting domains, we found 3880 (3.88%) as cloaked; 127 (i.e., 3.3% of 3880) of these cloaked domains are flagged by VirusTotal—in contrast to our established ground truth (80%).

On dynamic sites, we observed different types of cloaked content (e.g., technical support scams, lottery scams, malicious browser extensions, malicious links) served to users from the same domain at different times.¹ The number of cloaked sites identified from dynamic sites (861, 0.9%) is also significant, although it is certainly a lower bound as the dynamicity exhibited by these sites is inconsistent between consecutive requests.

Our results may be impacted by several factors: sites disallowing requests from automated crawlers, limitation of our heuristics, dynamicity of cloaking, and the use of SiteReview for establishing our ground-truth. Still, our findings uncover several cloaking behaviors of malicious

¹Note that serving dynamic content to GooglebotUA by a website may not necessarily be treated as cloaking. Response from a dynamic site to GooglebotUA may serve a non-dynamic version of the content that is tailored for that site (e.g., static HTML version), known as *dynamic rendering*; see: <https://developers.google.com/search/docs/guides/dynamic-rendering>. Although with dynamic rendering, a static view of a dynamic website is shown to GooglebotUA, the response content rendered to ChromeUA is dynamic. However, we consider serving significantly different content between ChromeUA and GooglebotUA as cloaking (e.g., page about cats to GooglebotUA and a page about dogs to ChromeUA).

sites and our methodology can also help detect these sites at scale.

Contributions.

1. We measure cloaking in malicious websites between a client browser (ChromeUA) and a search engine crawler (GooglebotUA) using a broader set of malicious domains with a more comprehensive methodology compared to existing work. Our technique improves the detection of cloaked malicious sites compared to past studies (e.g., cloaking in dynamically generated web content), and detect various scams (e.g., deceptive prize notices and lottery scams) and malicious content (e.g., malicious browser extensions) rendered in cloaked web pages.
2. Our methodology can identify 80% cloaked malicious domains from our ground truth; the detection rate also remained consistent between repeated measurements. For comparison, see e.g., Oest et al. [11] (detected 23% cloaked phishing sites in their full tests), Invernizzi et al. [6] (detected 4.9% and 11.7% cloaked URLs with high-risk keywords in Google advertisements and search results respectively), and VirusTotal (3.3% with our own dataset).
3. We highlight the role of domain generation engines such as DNSTwist [12], which can quickly provide a list of highly-likely malicious domains to serve as ground-truth, especially if used along with our heuristics.

2. Related Work

In this section, we compare previous work on detecting malicious sites, analyzing resiliency of blacklists, and the use of various heuristics to detect malicious sites. We also compare our methodology and results with past work.

Vadreu et al. [1] studied social engineering attacks delivered via malicious advertisements, and found 11,341 (16.1%) out of 70,541 publisher sites hosting malicious ads. Except for lottery/gift (18%) and fake software (15.4%), Google Safe Browsing (GSB) [17] detected only under 1.4% of other types of malicious ads (e.g., technical support). Tian et al. [2] studied elite phishing domains targeting desktop and mobile users, and found sites hosted on these domains were mostly used for credential phishing (e.g., impersonating of payment, payroll and

freight systems). They found 1175 out of 657,663 squatting domains were related to phishing; as the source of their domain list, they used 224 million DNS records in ActiveDNS project [8]). However, only 100 (8.5% of 1175) domains were flagged as malicious by PhishTank, eCrimeX and VirusTotal (with 70+ blacklists). They also compared evasion techniques between a desktop and a mobile client (Chrome). We study search-engine-based cloaking (ChromeUA vs. GooglebotUA), focusing on various types of malicious websites (beyond credential phishing).

Invernizzi et al. [6] studied variations in cloaking with search and advertisement URLs. They used several cloaking detection techniques based on web page features, e.g., content, screenshot, element, request tree and topic similarities; we adopt some of these techniques. In addition to static content analysis, we also analyze dynamic content. We compare screenshots of web pages between ChromeUA and GooglebotUA using OCR to find discrepancies in visual appearance (i.e., cloaking). Some of these discrepancies are not detected by simply comparing the content, but by supplementing other methods (e.g., semantics of a web page). The differences in the meaning of a page's content between the crawler and the browser (i.e., semantic cloaking) are used to deceive a search engine ranking algorithm [18], where search engine operators are more likely to be duped with the cloaked content. We use topic similarity evaluated using the LDA algorithm [19] to identify the semantic differences of web pages between ChromeUA and GooglebotUA.

Oest et al. [11] presented a scalable framework called *PhishFarm* for testing the resiliency of anti-phishing and browser blacklists, using 2,380 phishing sites deployed by the authors. Between mid-2017 and mid-2018, they found that the blacklisting functionality in mobile browsers was broken and cloaked phishing sites were less likely to be blacklisted compared to non-cloaked sites. The authors also mentioned blacklisting malicious websites remained low for mobile browsers compared to desktop browsers. We also observed a similar trend in our tests.

Rao et al. [20] used characteristics of a URL (i.e., hostname, full URL) to determine legitimate websites. Marchal et al. [15] used parts of a URL that are manipulated by a phisher (e.g., subdomains, web application path) to detect phishing sites. Panum et al. [5] reviewed highly influential past work to assess strategies with adversar-

ial robustness to detect phishing. These strategies include distinguishing between phishing and benign websites using visual similarity and leveraging URL lexical features. In our study, we use DNSTwist to generate potential malicious typo-squatting domains using lexical information of seed domains.

In summary, past measurement studies [2, 21, 6, 1] are mostly focused on specific categories of malicious websites (e.g., phishing, malware, social engineering). Each of these categories of websites may participate in cloaking. Several studies have used self-crafted URLs hosting content of particular malicious categories (e.g., phishing) or brands (e.g., PayPal) [9, 11, 2]. We use a broad set of registered squatting domains—combo-squatting (HTTPS only) and typo-squatting domains, hosting different types of potentially malicious websites to study cloaking behaviors.

3. Methodology

In this section, we explain our methodology to study cloaking behaviors in phishing and malware websites. We generate domains that may host potential phishing/malicious sites and pass them as input to our crawler. Various features (e.g., headers, links, page source/content, screenshots) are saved, and processed by an *analyzer* to identify cloaked sites and the results are stored into a database for further evaluation; see Fig. 1 for an overview of our experimental setup.

3.1. Generating squatting domains

Attackers are more inclined to impersonate popular websites, both in content and domain name, by hosting malicious sites on squatting domains [22, 23, 2]. These domains can be categorized as typo-squatting or combo-squatting. The domain lists used in our work is listed in Table 1. We generate 100,000 squatting domains (see list category A) using the following methods. The squatting domains sampled from these methods are from possible malicious domains.

3.1.1. Typo-squatting domains from DNSTwist

DNSTwist [12] takes a specific domain name as a seed, and generates a variety of potential registered phishing/malware domains. The domains generated in two con-

secutive runs of DNSTwist are not the same. This is because DNSTwist passes the seed domain provided to a function (`DomainFuzz`), which randomly generates many permutations of domain names similar to the seed domain, but with typographical errors. To determine domains hosting malicious content, DNSTwist use fuzzy hashes to identify sites serving similar content as their original domains (using the *ssdeep* option).

We provide top 1983 Tranco websites [24] as seeds to DNSTwist. From Mar. 22, 2019 to Mar. 27, 2019, we generate 277,075 already registered, unique typo-squatting domains; we then randomly choose 92,200 of these domains for our experiments (to save time). We choose the timings of the extraction of domains around the same time as the actual crawling of the sites, to ensure most of them are still responsive during crawling as typo-squatting domains can be recycled quickly [2].

The typo-squatting domains generated from DNSTwist are of the following types, explained using `google.com` as the seed domain. (1) Addition: A character is added at the end of the *public suffix*²+*l*² segment of the domain (`googlea.com`). (2) Bitsquatting: Flips one bit of the domain (`foogle.com`). (3) Homoglyph: visually similar domains, although the characters are not the same as the seed domain (`g0og1e.com`). (4) Hyphenation: A hyphen is added in between the characters of the seed domain (`g-oogle.com`). (5) Insertion: A character is inserted in between characters of the seed domain (`goo9gle.com`). (6) Omission: A character in the seed domain is removed (`goole.com`). (7) Repetition: A character in the seed domain is repeated consecutively, two or more times (`ggoogle.com`). (8) Replacement: A character in the seed domain is replaced with another character (`toogle.com`). (9) Sub-domain: A period is inserted in between any two characters of the seed domain to transform it to a sub-domain (`g.oogle.com`). (10) Transposition: Position of two characters in the seed domain is swapped (`gogole.com`). (11) Vowel-swap: A vowel character is replaced with another vowel (`goagle.com`).

²A public suffix is defined as “one under which Internet users can (or historically could) directly register names” see: <https://publicsuffix.org>.

List label	Number of domains	Experiment type
A	100,000	Cloaking is measured between ChromeUA and GooglebotUA
B	25,000	Cloaking is measured between ChromeUA and GooglebotUA for desktop environment, and between ChromeMobileUA and GooglebotUA for mobile environment. A random subset of domains from list A is used.
C	10,000	Comparison of HTTP vs. HTTPS cloaked sites (5000 each) hosted on combo-squatting domains. We use the same user-agents as in list A to identify cloaked domains.
D	5000	Comparison of user-agent vs. referrer cloaking of sites hosted on squatting domains. For referrer cloaking, we use ChromeUA with referrer header: <code>http://www.google.com</code> .

Table 1: Squatting domain lists used in our experiments

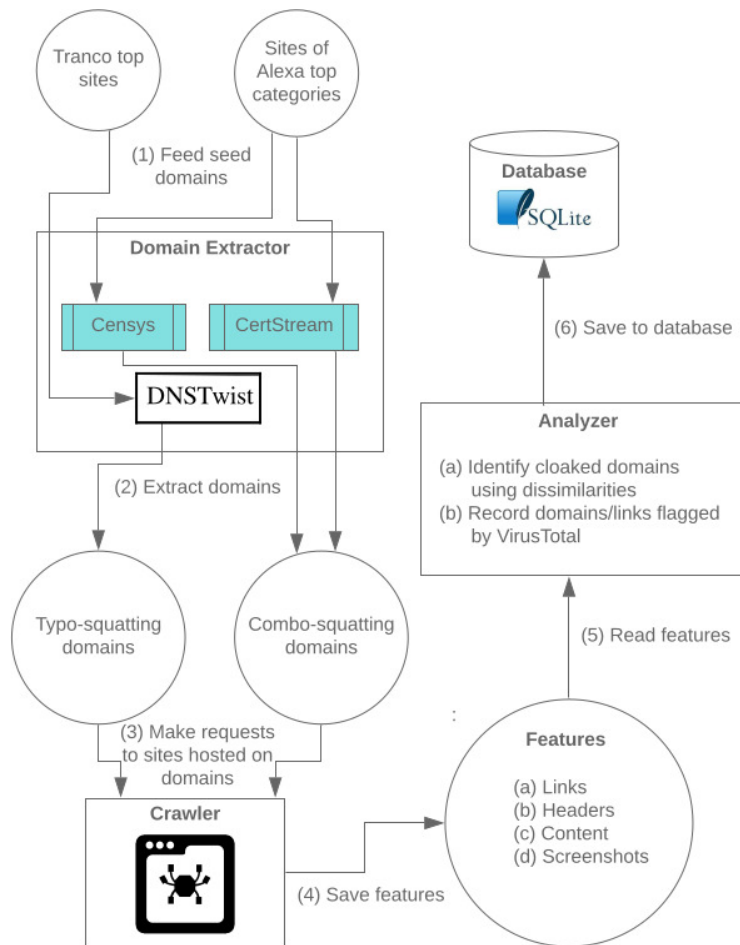


Figure 1: Our system setup.

3.1.2. Combo-squatting domains

Combo-squatting domains are concatenations of the target domain with other characters or words. These domains generally do not have spelling deviations from the target, and require active involvement from the attacker (e.g., social engineering); cf. typo-squatting is passive and relies on a user’s accidental typing of a domain name [21]. Combo-squatting domains are also used in phishing attacks [2]. Therefore, we generate 7800 combo-squatting domains as follows.

We collect top-50 sites of 16 categories (e.g., adult, business, computers, health, news, shopping, sports), and top-50 Alexa sites specific to China—a total of 850. During our preliminary manual verification, we observe a lot of phishing and malware sites are hosted in China, and thus we choose Alexa top-50 sites from China.

Then, we identify domain names that partially match any of these 850 domains from certificates that are used to host HTTPS phishing/malware sites in order to deceive legitimate users [25]. We only consider certificates issued after Jan. 1, 2019 to minimize the collection of already recycled domains. We collect combo-squatting domains that serve content over HTTPS between Apr. 4–9, 2019 using the following sources. (1) Censys: This is a search engine [26] that aggregates information of networked devices, websites and certificates deployed. We check the subject common name field of certificates against our 850 target domains. (2) CertStream: The certificate data in Certificate Transparency (CT) logs is polled over a websocket (`wss://certstream.calidog.io`) in CertStream [27]. We then check the common name field of certificates against our target domains.

To derive combo-squatting domains served via HTTP, we extract domain names from the DNS A records from Project Sonar [28]. After extracting the domains running on port 80 that return a 200 response code (i.e., non-recycled domains), we partially match them with the 850 target domains, to filter the combo-squatting domains that are derivations of the top brands.

As many combo-squatting domains are benign (e.g., `mail.google.com`), we use SquatPhish [29] to filter only those domains exploited for phishing that are derived from above sources (i.e., Censys, CertStream, Project Sonar). SquatPhish leverages a machine learning model to identify phishing pages based on the HTML source

and text extracted from images included in a web page. We use SquatPhish to filter 7800 phishing domains from 205,263 combo-squatting domains collected from the above mentioned sources. We do not consider domains that return a 4xx or 5xx response code, as those domains may already have been recycled.

3.2. Our Crawler

To identify cloaking activity, we extract features from 100,000 web pages hosted on potential malicious squatting domains, using GooglebotUA and ChromeUA (and a subset of the same websites by ChromeMobileUA). We use GooglebotUA, ChromeUA and ChromeMobileUA for our experiments by manipulating the “user-agent” field of the request header; see Appendix A for a discussion on cloaking types.³

We use Puppeteer [30] to implement our crawler. Puppeteer provides high level APIs to control the Chrome browser and can be customized to run as headless to load dynamic content before saving the web pages. Compared to other alternatives (including Selenium [31]), Puppeteer offers the flexibility of handling failed requests gracefully and is less error prone [2]. Tian et al. [2] also used a crawler based on Puppeteer. However, unlike them, our crawler renders content that is dynamically generated before saving (Tian et al. [2] chose not to consider content dynamically generated by JavaScript due to the high overhead). We believe that dynamic source files (e.g., JavaScript, Flash) may render differently based on the user-agent of a request (e.g., the list of links shown in an iframe are benign for GooglebotUA, but malicious for ChromeUA). To identify web pages with dynamic content, we request the home page of each website twice from GooglebotUA and ChromeUA. The GooglebotUA and ChromeUA are represented as C and B, and the iterations of requests from each client is 1 and 2, the sequence of requests made for a particular website is labeled as C1, B1, C2, B2.

³The user-agent string selected for GooglebotUA: “Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)”, ChromeUA: “Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36”, and ChromeMobileUA: “Mozilla/5.0 (Linux; Android 8.0.0; T A-1053 Build/OPR1.170623.026) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3368.0 Mobile Safari/537.36”.

Web servers may have heuristics to determine automated crawlers and reject their service. We incorporated few mitigation steps in our crawler to minimize these effects; e.g., we manipulate webdriver, plugins and language properties of the navigator object of the browser accordingly [32].

We crawl the sites hosted on squatting domains between April 10 to April. 13, 2019, and run the crawler on 10 Amazon EC2 instances (c5.2xlarge) setup with Ubuntu 16.04 (8 vCPU, 16GB RAM). For our experiments, we do not consider sites where the differences in content between GooglebotUA and ChromeUA are minor or benign. Some of these sites redirect to non-malicious top-1M Tranco sites [24] from ChromeUA. There are also sites that throw connection errors; see Appendix B for an overview of issues encountered during crawling.

During crawling of each site, we gather features to identify potential cloaking activity of possible malicious domains. These features include HTTP headers, page source/content (both static and dynamic), links including those generated from dynamic content (includes those in DOM objects within iframe elements) and screenshots.

3.3. Analyzer

The analyzer process applies heuristics to features of websites collected during crawling, in order to identify cloaked websites. In this section, we explain the heuristics and rules applied while processing the saved features. These heuristics are only applied if the HTML page source and screenshots are successfully saved for all C1, B1, C2, B2 visits of a website. The results evaluated by the *analyzer* are saved into a SQLite database.

3.3.1. Skipping domains with benign content

Domain name registrars (e.g., GoDaddy, Sedo) advertise domains available for sale on their landing pages. The content of such landing pages sometimes differ slightly between GooglebotUA and ChromeUA. Since such differences should not be attributed to cloaking, we skip those domains from processing. For non-English sites, we use Google translator to detect the language of those sites and translate the content to English prior to processing. For screenshots, we use the Tesseract-OCR [33] library to extract the textual content. If the extracted text from a screenshot is non-English, Tesseract-OCR library

takes a significant amount of time to process (sometimes over 30 seconds). Therefore, we call the Tesseract-OCR library for only those sites identified as cloaked with content dissimilarities method using our heuristics described in Section 3.3.5. Some domains are redirected to top-1M Tranco sites [24]. Legitimate companies may buy squatting domains to protect users (a request to the possible squatting domain is redirected to the corresponding legitimate site [22])—see Section 3.3.2. Therefore, we do not consider these domains for our experiments.

3.3.2. Eliminating squatting domains owned by popular sites

Entities owning popular domains (e.g., top Tranco sites) buy squatting domains to safeguard its clients who may accidentally browse to those sites by mistyping their URLs. These squatting domains may not always redirect a user to the original popular site. To eliminate such domains from our measurements, we use the organization owning both the squatting and corresponding popular domains using the WHOIS records [34]. If both these domains are registered by the same organization, we disregard them from our analysis. Out of all cloaked domains, only one squatting domain (`expedia.com`) is owned by the same organization (Expedia, Inc: `expedia.com`). Therefore, we eliminate the particular domain from our analysis. However, the following types of squatting domains are not eliminated from the analysis, as we cannot determine if those domains are also owned by the corresponding popular domain’s organization: 8 domains with WHOIS registrant name/organization information recorded as “REDACTED FOR PRIVACY”, and 20 squatting domains registered by *Domains By Proxy* [35] where the registrar itself is listed as the WHOIS administrative contact.

3.3.3. Domains with exceptions

We observe that some sites do not allow automated crawlers to access them. This observation holds for both GooglebotUA and ChromeUA. Unfortunately, our automation cannot determine potential cloaking activities in some sites that are prevented from accessing with GooglebotUA. Some sites display failures such as “Too many requests”, “Page cannot be displayed. Please contact your service provider for more details” and “404 - File or directory not found” when requested from our automated

crawler. We experience such failures despite the use of known techniques to avoid crawler issues with accessing websites [32]. However, upon manual inspection, we notice that some of these sites engage in cloaking.

3.3.4. Links flagged by blacklists

Target URLs hosting phishing or malicious content that are flagged by blacklists are of different forms.

Sites redirecting to websites flagged by blacklists. We record URLs redirected from squatting domains to websites that are also flagged by blacklists. We use VirusTotal to determine how many of the redirected sites are flagged as phishing or malicious.

Identify links in iframes flagged by blacklists. We traverse the Document Object Model (DOM) objects within iframes elements (including child iframes) of sites hosted on squatting domains (level 1 URLs) to find dynamically generated second level links of sites that are flagged by blacklists. A listing of such second level links appearing on an iframe of a site is shown in Figure 2b. However, most of these links show a set of related third level links when clicked on any one of them. These third level links will lead to actual sites described in link descriptions. We run 5000 link URLs from each of these 3 levels through VirusTotal on a daily basis to identify if any of those are flagged as phishing or malware. The first level URLs for this exercise is selected randomly from list category A in Table 1. This help us find the rate at which link URLs hosting phishing or malware content is detected by available blacklists.

3.3.5. Evaluate dissimilarities of website features

We evaluate the following dissimilarities based on the website features collected during crawling. These heuristics facilitate in finding websites engaged in cloaking.

Header dissimilarities. Although *title*, *keyword* and *description* are not part of the standard HTTP response header, adversaries appear to include these fields in the HTTP response headers [18]. Therefore, we compare these fields between ChromeUA and GooglebotUA to find instances of cloaking.

Link dissimilarities. We find the links in rendered web pages from GooglebotUA that are missing from ChromeUA, and vice-versa. In addition, we also identify which of those links are malicious using VirusTotal.

Content dissimilarities. We extract text surrounding *h*, *p*, *a* and *title* tags of the HTML page source following rendering of dynamic source code (e.g., JavaScript). We also consider HTML forms along with *type*, *name*, *submit* and *placeholder* attributes. Stop words (e.g., the, a, an) are removed from the extracted content.⁴ Then we evaluate the SimHash [36] of the extracted page source from GooglebotUA and ChromeUA, and compute the hamming distance between them.⁵ If the hamming distance exceeds a preset threshold ($t1=20$), we assume that the page is likely to be cloaked. We set the threshold after manual verification, where we find $t1=20$ gives optimal results after removing benign differences (e.g., pages having random session identifiers or timestamps). This threshold is also close to Tian et al. [2] (distance between 24 and 36). We set a second threshold $t2$ for pages with dynamic content. The same value (20) appears to be adequate in this case too. We define a static page if the following is satisfied: $|FH(C1) - FH(C2)| = 0$ AND $|FH(B1) - FH(B2)| = 0$; here *FH* represents FuzzyHash. A static page is possibly cloaked if the following is satisfied:

$$|FH(C1) - FH(B1)| > t1 \text{ AND } |FH(C2) - FH(B2)| > t1.$$

We also compare the semantics of a page between GooglebotUA and ChromeUA to determine if the specific page is cloaked. We identify the most prominent topic of a page (i.e., topic of the page content with highest probability) using the Latent Dirichlet Allocation (LDA) algorithm [19]. A topic in LDA is a set of related words extracted from the document with probabilities of their prominence assigned to them. If Tb and Tc are the most prominent topics corresponding to page content from GooglebotUA and ChromeUA, the static page previously identified as likely to be cloaked has a high probability of being cloaked when $Tb \neq Tc$. Similarly, a page with dynamic content is cloaked if:

$$(|FH(C1) - FH(C2)| > t2 \text{ OR } |FH(B1) - FH(B2)| > t2) \text{ AND } (|FH(C1) - FH(B1)| > t1 \text{ AND } |FH(C2) - FH(B2)| > t1) \text{ AND } Tc \neq Tb.$$

Image dissimilarities. Using the page content at source code level to determine cloaking may not be sufficient,

⁴<https://pythonspot.com/nltk-stop-words/>

⁵SimHash is a FuzzyHash that is used to identify similar documents. The difference between two documents is measured using the hamming distance—larger distance implies higher dissimilarity.

and it should be complemented with the visual differences of the page (i.e., screenshots). This is because, content rendered by dynamic source code (e.g., JavaScript, Flash) and advertisement displayed cannot be captured from the page source. Therefore, with this method, we follow the same procedure as for *Content dissimilarities*, except that we use ImageHash as the FuzzyHash to evaluate the differences of screenshots between GooglebotUA and ChromeUA. Very small color perturbations (between benign and malicious views) in the space of humans yield significant changes in the binary representation [5] of a web page screenshot.

3.4. Limitations

We exclude sites that our crawler could not reach. Also, the number of cloaked sites we identify is a lower bound due to the choice of our heuristics. According to our observations, some cloaked sites with dynamic content show distinct content at different times (cf. [11]). Therefore, our results with dynamic sites are a lower bound and is based on content rendered at the time the request is initiated from the automated crawler, where these results may differ on each request for dynamic websites.

Both academic and commercial tools available are not accurate in categorizing social engineering sites hosted on squatting domains in the wild; e.g., *Off-the-Hook* [15] gives false negatives for typo-squatting domains, SquatPhish [29] mostly detects credential phishing. However, we observe that the Symantec *SiteReview* tool detects malicious squatting domains at a comparatively higher accuracy (42.6%). SiteReview accepts the domain URL as input, but not the page content. For dynamic websites, the content viewed by our crawler may not be the same as what is analyzed by SiteReview (i.e., view of a web page may change with time due to dynamic behavior). Therefore, we have limited control in identifying the content category of a site using the SiteReview tool.

4. Ground truth

Some sites hosted on squatting domains are malicious and they may engage in social engineering attacks of various forms such as credential phishing, spear phishing, tech scams, and social engineering ad campaigns. However, most existing tools detect only particular types of social engineering attacks. For example, SquatPhish [29] is

a machine learning model to detect phishing sites with input fields (mostly credential phishing). *Off-the-Hook* [15] is a client side browser extension capable of detecting most forms of phishing pages but does not support the detection of sites hosted on squatting domains. We find Symantec’s SiteReview online tool is very effective in correctly categorizing most social engineering sites compared to other tools. However, it does not offer any API to automate the malicious domain detection. Note that SiteReview appears to use the RIPE Network Coordination Center (NCC) [37] to categorize websites.⁶

We hosted a web page on a Microsoft Azure cloud domain that closely resembles content of a malicious site, and submitted the page to SiteReview which categorized the site as *Suspicious* within 24 hours. Our web page is not shared with anyone or have any backlinks that are used for search engine optimizations (SEO). During this time, we notice requests *only* from IP addresses assigned to RIPE NCC every hour. A *Chrome* user-agent is used by all these access requests to our page.

We use SiteReview to identify categories of 3880 cloaked and 3880 non-cloaked domains. The cloaked domains are identified using the content dissimilarities method in Section 5.3. Some of these cloaked and non-cloaked domains are flagged as malicious by SiteReview. 171 cloaked and 187 non-cloaked domains were unreachable during our tests. The number of cloaked malicious domains flagged by SiteReview (1636, 44.11%) is significantly higher compared to that of non-cloaked malicious domains (1022, 27.67%); see Table 2.

We classify (**1024**) active cloaked domains (as of Oct. 15, 2019) using a semi-automated process with SiteReview to identify how many of them are malicious. During this process, we reclassify sites that SiteReview failed to classify or misclassified. This semi-automated process is used to determine the ground truth as described below.

- We found **413** sites serving content related to social engineering attacks (SEA); 383 *suspicious* sites with content that poses an elevated security or privacy risk; 23 malicious sites; 5 phishing sites; and 2 sites with potential unwanted programs. Some sites

⁶Bluecoat, the original developer of SiteReview (acquired by Symantec) is a member of RIPE NCC, see: <https://www.ripe.net/membership/indices/data/eu.blue-coat-systems.html>

Category	Cloaked domains	Non-cloaked domains
Suspicious	1550 (41.79%)	920 (24.91%)
Malicious Sources/Malnets	56 (1.51%)	71 (1.92%)
Scam/Questionable Legality	11 (0.30%)	12 (0.32%)
Phishing	13 (0.35%)	9 (0.24%)
Spam	4 (0.11%)	4 (0.11%)
Potentially Unwanted Software	1 (0.03%)	3 (0.08%)
Malicious Outbound Data/Botnets	1 (0.03%)	3 (0.08%)
Total active domains	3709	3693

Table 2: SiteReview categorization of malicious squatting domains - cloaked vs. non-cloaked

are classified into more than one of the mentioned categories.

- SiteReview was unable to classify 361 sites, labeled as “not yet rated (NYR)”. With manual inspection, we observed that some NYR sites show content similar to social engineering attack (SEA) sites. Therefore, for each of the NYR sites, we compute the SimHash [36] of the page source, and then compare the SimHash value with all SEA sites. We classify a NYR site as SEA, if the hamming distance between the SimHashes of the NYR and SEA sites is under 20, and the hamming distance is the lowest between the NYR site and any one of the SEA sites. For example, assume that the NYR site xyz shows similar content as sites in SEA categories A and B with hamming distances of 8 and 5, respectively; then we label xyz as of category B. With this approach, we could correctly classify **306** NYR sites as SEA (out of 361).
- SiteReview classified 250 sites into benign categories. With manual inspection, we found **102** false positives in this categorization (i.e., malicious sites classified as benign); 2 Chinese sites, 1 deceptive site flagged by Google Safe Browsing [17], 80 sites with iframes that include links to malicious targets, 17 sites with promotional contests (e.g., online casino), 1 shopping site and 1 site showing that the operating system (Windows 10) is infected.

From the above mentioned observations, we found a total of 821 malicious sites (413+306+102) in different social engineering categories from the 1024 cloaked sites. Therefore, the percentage of malicious sites from those

that are cloaked is 80.2%. This value may change due to the dynamicity of the content rendered from these cloaked sites (i.e., some sites alternatively show benign and malicious content during successive requests and at different times). We emphasize that SiteReview is only used to validate our ground truth, and our methodology is not dependent on SiteReview.

We also apply the ground truth analysis to sites hosted on 1500 randomly selected squatting domains generated from DNSTwist (from list category A in Table 1) and found 74% (1110 of them are malicious. These squatting domains contain both cloaked and uncloaked sites.

5. Dissimilarities

Sites with content discrepancies between GooglebotUA and ChromeUA may be cloaked, assuming differences are due to evasion techniques adopted by adversaries. In this section, we delve into such differences using the domain list category A in Table 1.

5.1. Link dissimilarities

We evaluate the number of links in web pages that appear with ChromeUA, but not with GooglebotUA, and vice-versa. We found that 21,616 distinct links appeared in ChromeUA (1557 sites), compared to 10,355 links in GooglebotUA (1235 sites); i.e., ChromeUA observed over twice the number of links compared to GooglebotUA.

Dynamic pages rendered from both ChromeUA and GooglebotUA show listings of advertisements links. These links changed on successive refreshing of the page from the same client or with different clients (e.g., ChromeUA and GooglebotUA).

5.2. Header dissimilarities

We inspect the *title*, *description* and *keywords* header fields to find the sites where the header fields are different between GooglebotUA and ChromeUA.

Header	# diff	# only with GooglebotUA	# only with ChromeUA
Title	2644	2190	3388
Description	3530	4839	1375
Keywords	265	716	408

Table 3: Header dissimilarities—the last two columns show the number of the specific header type that exists only from one user-agent (empty in the other)

Apart for the title header field, description/keywords fields in headers had significant discrepancies with GooglebotUA. Upon manual inspection, we observed that the dissimilarities in title & description header fields were benign as they mostly contained the domain name or content that relate to sale of the domain. According to Table 3, 716 sites had the keywords header field injected only with GooglebotUA (e.g., health, wellness, surgery) and its use may had an impact in improving the rank of those websites. Many keywords added to HTTP headers were sent to the crawler to perform semantic cloaking [18].

5.3. Content dissimilarities

We compare pages rendered between ChromeUA and GooglebotUA using syntactical and semantic heuristics as defined in Section 3.3.5. Sites that show benign content (e.g., website under construction) are excluded. While cloaking is prevalent in static pages, we also observed cloaking in pages with dynamic content. In the case of the latter, a significant number of sites showed cloaking behaviors at random when they were requested repeatedly.

With our automated process, we found 2183 (2.2%) sites with static content and 83 (0.08%) sites rendering dynamic content were cloaked by examining the page source/content using heuristics; see Table 6. Out of them, 1763 (1.8%) and 42 (0.04%) sites serving static and dynamic content were redirected to other URLs respectively. The top 5 target URLs where these sites were redirected (for both static and dynamic sites) were `plt2t.com` (27), `yourbigprofit1.com` (24), `www.bate.tv` (10), `yvxi.com` (8) and `www.netradioplayer.com` (7).

Failure	# Content dissimilarity	# Image dissimilarity
HTTP 404 Not Found	398	0
HTTP 403 Forbidden	349	302
“Coming soon”	244	64
HTTP 500 Server Error	14	0

Table 4: Failures from GooglebotUA

Out of these target domains, `plt2t.com` redirected to another website that showed “your computer was locked” scam message occasionally, with the aim of getting the victim to call a fake tech support number.

Most cloaked sites (361) from the squatting domain list category *A* in Table 1 had a content length difference of 1-10 KB between ChromeUA and GooglebotUA, compared to 121 cloaked domains that had a content length difference greater than 10 KB. Although this implies that in most cloaked sites, the content length difference between ChromeUA and GooglebotUA is minimal, the difference in presented content may be significant due to the use of dynamic rendering technologies (e.g., AngularJS, Puppeteer).

Phishing sites often adopt HTTPS to give a false sense of security to the victim users (see e.g., [11]). In Table 5, we compare cloaked vs. non-cloaked sites served via HTTP and HTTPS (using combo-squatting domains, category *C* in Table 1); cloaking is less apparent in HTTPS sites, where majority of the certificates (55) are issued by the free certificate provider Let’s Encrypt.

We observed the following major content differences between ChromeUA and GooglebotUA:

- Out of 100,000 squatting domains in list category *A* of Table 1, 2337 sites appeared to be dynamic only from GooglebotUA, and 2183 from ChromeUA. No overlap in domains was observed between GooglebotUA and ChromeUA. We were unable to differentiate the content of these sites between GooglebotUA and ChromeUA, as when checked manually, the most probable topic of the page content as determined by Latent Dirichlet Allocation (LDA) algorithm [19] differed drastically on each request due to dynamic nature of the sites. Among these sites, there

Protocol	Content type	With content dissimilarities		With image dissimilarities	
		Cloaked sites	Redirects	Cloaked sites	Redirects
HTTP	static	192	166	142	118
	dynamic	21	7	22	7
HTTPS	static	52	36	37	27
	dynamic	3	2	1	1

Table 5: Combo-squatting domains served via HTTP/HTTPS

were also sites displaying dynamically populated links within iframe elements from ChromeUA, while such iframes appeared to be empty from GooglebotUA. These links related to various areas of businesses (e.g., Car Insurance, Credit Cards).

- The failures with content dissimilarities in Table 4 were observed from GooglebotUA, while with ChromeUA a different view of the content was displayed. For examples, the websites that showed “Coming soon” page content from GooglebotUA, showed the actual page content when requested from ChromeUA. Malicious sites also returned error codes when they detected the visitor was not a potential victim [11] (e.g., a search engine crawler).

Figures 2 to 4 are examples of instances where cloaking was used for phishing/malware purposes.

5.4. Image dissimilarities

We also determine cloaking by comparing the differences of screenshots of web pages between ChromeUA and GooglebotUA using image dissimilarity techniques. The number of sites with static content subjected to cloaking was 1710 (1.7%), while those with dynamic content was 784 (0.8%). We observed 960 (1%) and 490 (0.5%) of these sites with static and dynamic content, respectively, were redirected to other websites. In contrast to content dissimilarity method, with image dissimilarity, we found more cloaked sites that were also dynamic.

Page content alone is insufficient to detect cloaking due to technologies used in websites (e.g., Flash) that render dynamic content. Visual identity of a benign website can be shared by a malicious website with undetectable perturbations to humans, although their binary representations are completely distinct [5]. In addition, advertisements on web pages can be more tailored to a specific

client, and may be hidden from GooglebotUA. The failures as identified from image dissimilarity technique in Table 4 were only observed from GooglebotUA. Although with image dissimilarity technique, the detection of cloaking was better, the text extracted from screenshots using the Tesseract [33] OCR library was sometimes inaccurate. For example, Tesseract reads “Coming soon” as “Coming scan”. Despite our manual efforts to minimize the impact of these inaccuracies, the inaccuracies of Tesseract may have affected the accuracy of the results in Table 4.

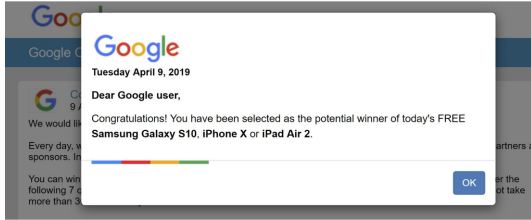
Cloaking of domains served via HTTPS giving a false sense of security to users were a fraction when compared to those domains using HTTP; static content (37, 26%), dynamic content (1, 5%). There were 38 (0.04%) cloaked sites running on combo-squatting domains with valid TLS certificates as shown in Table 5.

5.5. Comparison of results of cloaking detection techniques

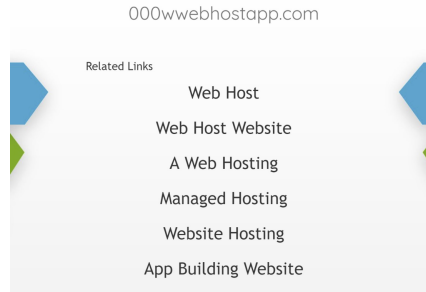
The dissimilarities techniques we use to identify cloaked sites focus on different structural elements of a web page. The results of content and image dissimilarities converge to some extent as they are applied on the syntactical and visual perspectives of the page content.

With link dissimilarities technique, we observed links are more prevalent with ChromeUA as opposed to GooglebotUA (for both static and dynamic content). The links shown in web pages hosted on domains were 209% and 140% for static and dynamic pages from ChromeUA compared to GooglebotUA. However, the links appeared in dynamic content were 6x and 9x when compared to static content with ChromeUA and GooglebotUA, respectively. This may mean that phishing/malware domains suppress links from GooglebotUA to avoid detection.

We also observed keywords in headers from GooglebotUA that were not seen from ChromeUA. These key-

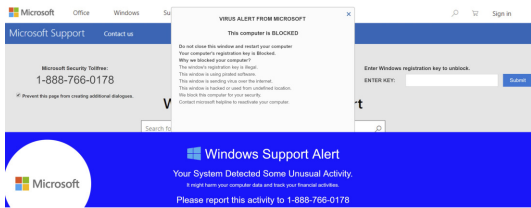


(a) ChromeUA

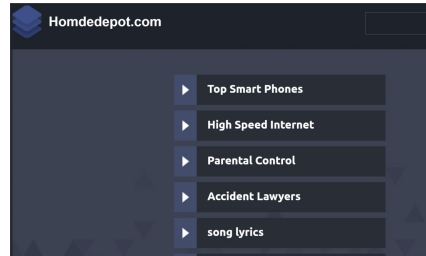


(b) GooglebotUA

Figure 2: Cloaking differences for site: 000webhostapp

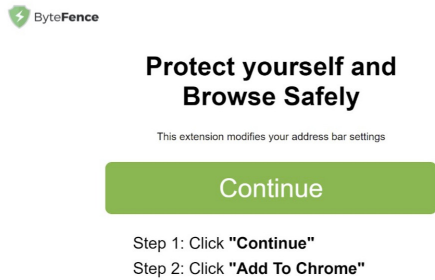


(a) ChromeUA

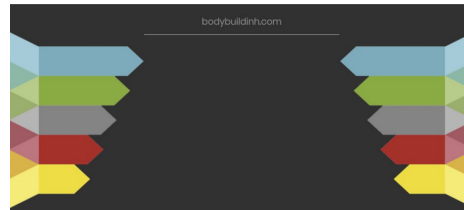


(b) GooglebotUA

Figure 3: Cloaking differences for site: homedepot.com



(a) ChromeUA



(b) GooglebotUA

Figure 4: Cloaking differences for site: bodybuildinh.com

words that were exclusive to GooglebotUA may influence the search engine ranking algorithms for corresponding sites. With header similarities technique, the keyword header fields related to specific categories of content appeared only with GooglebotUA. Therefore, these keyword header fields may possibly have been leveraged to manipulate the rankings of websites.

With content and image dissimilarities methods, we find cloaked websites from both static and dynamic websites with potential malicious content. With both content and image dissimilarity methods, we found a very small fraction (3880, 3.9%) of domains participate in cloaking. There were 880 cloaked sites that overlap between content and image dissimilarities. Out of 3880 sites 127 (3.3%) were flagged by VirusTotal. However, according to our ground truth (see Section 4), 80% of the cloaked sites were malicious. The low detection rate of malicious sites by VirusTotal highlights that blacklists are not effective in identifying a large proportion of social engineering sites. With image dissimilarities, a larger number of cloaked sites were found with dynamic content compared to content dissimilarities. Conversely, a large number of cloaked websites were identified using content dissimilarities with static content compared to image dissimilarities. Identifying dynamic content is more effective by analyzing the screenshots of web pages, as dynamic content may not be captured from the page source. Some of the cloaked sites that are dynamic, rendered different content on each refresh of the page. In some sites, benign and cloaked content were rendered alternatively when the page is refreshed multiple times. Since the dynamicity of sites depends on the time accessed, our results are a lower bound.

Manually inspecting 100 cloaked sites (from list category A in Table 1), we found 22 (22%) of them had differences in content. Few examples of differences in site content between ChromeUA and GooglebotUA are shown in Figures 2 (deceptive prize notice), 3 (technical support scam), 4 (prompting to install a malicious browser extension). The browser extension in Figure 4 (*ByteFence Secure Browsing*⁷) is a known malicious browser extension detected by reputable antivirus engines due to suspicious data collection habits and browser redirects. Most of these

sites served content that changed between subsequent requests and at times alternated between malicious and non-malicious content.

6. Discussion

We discuss below observations from our analysis in Section 5.

6.1. Dynamicity in squatting sites

We found few squatting domains (644, 0.6%) showed dynamicity in rendered content that changed between two consecutive requests with ChromeUA. Since, dynamic sites can serve different content only after multiple requests or change between static/dynamic content alternatively, our results are a lower bound. Therefore, detection of dynamic sites with cloaked content is difficult compared to that of static sites. There were 83 cloaked sites identified using content dissimilarities in Section 5.3 out of the 644 dynamic sites. These cloaked dynamic sites changed between consecutive requests to show various forms of malicious content (e.g., technical support/lottery scams, malicious browser extensions).

6.2. Malicious squatting domains generated from DNSTwist

DNSTwist [12] uses fuzzy hashes,⁸ to identify malicious sites, by comparing the fuzzy hashes between web page content of a seed domain and the corresponding typo-squatting domain. For a 100% match, the typo-squatting web page content is similar to content hosted on the corresponding seed domain (includes situations where typo-squatting domain redirects to seed domain). When the comparison returns a match of 0, the web page of the typo-squatting domain is most likely malicious. Out of 119,476 typo-squatting domains generated from DNSTwist, 76,178 (63.76%) returned a match of 0. We randomly selected 500 typo-squatting domains from list category A in Table 1, and found 187 malicious domains (37.4%) using SiteReview. Therefore, a significant proportion of DNSTwist generated typo-squatting domains are indeed malicious.

⁷<https://botcrawl.com/bytedefence-secure-browsing/>

⁸ssdeep: <https://ssdeep-project.github.io/ssdeep/index.html>.

6.3. Relevance of seed domains

We find that the number of seed domains of cloaked squatting domains with a single permutation (345, 0.3%) is considerably high compared to those with multiple permutations. There were only 229 seed domains with 2-7 permutations of cloaked squatting domains. The 7 seed domains in Figure 5 generated 8-13 permutations of squatting domains. The categories of services offered by these seed domains include government (`service.gov.uk`), gaming (`epicgames.com`), search engine (`google.com.ph`), health (`health.com`) and news sites (`cnbc.com`). We also show the number of seed domains of the generated cloaked squatting domains as a comparison in Table 6. With both content and image dissimilarities, we find the proportion of squatting domains to seed domains is higher with static content (1.89%-2.18%) compared to that of dynamic content (1.08%-1.42%).

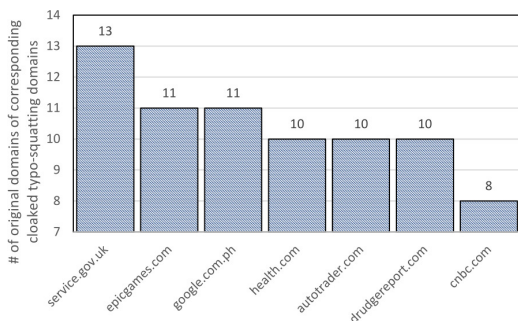


Figure 5: Top 7 seed domains of the corresponding cloaked domains with 8-13 permutations.

6.4. Detection of cloaked sites by blacklists

To study evasion of blacklists by cloaked squatting domains, we randomly selected 5000 squatting domains that are cloaked from domain list category A in Table 1, and ran them daily through VirusTotal between May 2, 2019 – June 5, 2019. At the end of this period (June 5, 2019), 92 (1.84%) were flagged by VirusTotal; phishing: 40, malicious: 41, malware: 22. Since our ground truth showed 80% of squatting domains were malicious (see Section 4), it appears that most phishing/malware squatting domains are not blacklisted. After approximately 3 months from the time of this experiment (on Aug. 26, 2019), we observed that URLs blacklisted by VirusTotal

have not changed significantly (87, down from 92). Further, on Sep. 4, 2019, we applied our methodology described in Section 3 to 2268 cloaked domains previously identified, and found 1038 (45.78%) of them were still showing cloaked content. These cloaked domains may contain malicious content although they were not flagged by blacklists. The remaining domains (1230) were either recycled or showed exceptions described in Section 3.3.3. Therefore, it appears that the rate at which these cloaked sites were detected by blacklists is extremely slow.

Typo-squatting domains hosting malicious content may get recycled more frequently. This behavior may cause delays in blocking new websites or slow reactions to domain take-downs that host malicious content [38]. We found 2256 out of 100,000 squatting domains (cloaked and uncloaked) as malicious in Apr. 2019. However, 2048 of these domains remained active as of Nov. 15, 2019, and out of those domains, 67 of them were no longer flagged by VirusTotal. These websites showed benign content that is different from when it was previously flagged by VirusTotal.

6.5. Variations of cloaking in different device types

A significant proportion of web traffic comes from mobile devices and mobile users are more vulnerable to phishing attacks [11]. We identified cloaked websites using the heuristics defined in Section 3.3.5 for 25,000 sites (category B in Table 1) hosted on squatting domains from both desktop and mobile browsers (Chrome); see Table 7. Cloaked sites with static content in mobile environment are more apparent compared to desktop environment. Similarly, redirections of sites hosted on squatting domains to target URLs are comparatively high in mobile environments. A significant number of cloaked sites overlap between desktop and mobile browsers as identified by content (326) and image (119) dissimilarity methods. The differences of the overlapping sites between desktop and mobile environments were mostly related to its layout. Tian et al. [2] found more phishing pages with mobile web browsers compared to desktop environment, and we observed a similar pattern for cloaked sites. The number of target URLs of redirections blacklisted by VirusTotal was low with mobile browsers compared to that of desktops. Oest et al. [11] observe mobile browsers (including Chrome) failed to show blacklist warnings between

Type of domain	Nature of content	With content dissimilarities			With image dissimilarities		
		Cloaked sites	Redirects	Target URLs flagged by VirusTotal	Cloaked sites	Redirects	Target URLs flagged by VirusTotal
Squatting	static	2183	1763	27	1710	960	6
	dynamic	83	42	0	784	490	3
Seed	static	1153	1012	20	985	693	6
	dynamic	77	38	0	552	382	3

Table 6: Projecting results of cloaked squatting domains to corresponding seed domains (i.e., squatting domain vs. seed)

Device	Content type	With content dissimilarities			With image dissimilarities		
		Cloaked sites	Redirects	Target URLs flagged by VirusTotal	Cloaked sites	Redirects	Target URLs flagged by VirusTotal
Desktop	static	607	498	7	484	289	3
	dynamic	20	9	0	230	135	1
Mobile	static	797	689	2	660	364	1
	dynamic	44	30	0	206	174	0

Table 7: Variation in cloaking between device types

Type	Nature of content	With content dissimilarities		With image dissimilarities	
		Cloaked sites	Redirects	Cloaked sites	Redirects
Referrer	static	9	5	4	3
	dynamic	3	2	18	15
User-agent	static	99	80	59	36
	dynamic	4	1	46	31

Table 8: Variation between user-agent vs. referrer cloaking

mid-2017 and late-2018. Although they claim that following their disclosure the protection level is comparable between mobile and desktop browsers, we noticed sites flagged by VirusTotal for mobile browsers were less than that of desktops.

6.6. User-agent vs. referrer cloaking

We compare websites identified as cloaked between user-agent and referrer cloaking. For both types of cloaking, we use the same sites in domain list category D in Table 1 that are hosted on typo-squatting/combo-squatting domains. As with our previous experiments, user-agent cloaking is measured between GooglebotUA and ChromeUA. For referrer cloaking, we use ChromeUA, but

to mimic clicks initiated through search engine results, we set the referrer header to `http://www.google.com/`. As shown in Table 8, for sites with static content, cloaked sites identified from user-agent cloaking were 11x-16x higher than that of referrer cloaking (from both content and image dissimilarities methods in Section 3.3.5).

7. Conclusions and Future Direction

Cloaked malicious sites deliver phishing, malware and social engineering content to victimize users. We found 22% of cloaked domains show malicious content (technical support scams, lottery scams, malicious browser extensions, malicious links), with significant differences

between ChromeUA and GooglebotUA. In addition, we also found cloaking behaviors in a considerable number of squatting domains hosting dynamic content at irregular time intervals. This type of cloaking in dynamic sites is harder to detect, and may go unnoticed by the detection algorithms. Some squatting domains redirect a website through multiple intermediary domains to its final destination. [22]. We found 1.8% (1805 domains in list category A in Table 1) cloaked squatting domains engaged in redirections with content dissimilarities.

We used DNSTwist to generate typo-squatting domains. The domain generation algorithms used in DNSTwist are highly successful in generating malicious domains. According to SiteReview along with our heuristics, 74% of these typo-squatting domains were malicious. Although, some of these malicious domains are short-lived, the attackers may cause harm to users during the domain life time due to slow reaction to blocking such domains.

In past studies, URLs used for crawling mostly include crafted websites or those belonging to specific malicious categories (phishing, social engineering ad campaigns). In contrast, the squatting domains we used host potential malicious content mimicking a variety of popular sites. The URLs of cloaked malicious websites we found may eventually get flagged by various blacklisting entities (e.g., VirusTotal). We observed more squatting domains and dynamically generated links identified from iframe elements are getting flagged as phishing or malicious by VirusTotal over time. The cloaked sites blacklisted by VirusTotal is a fraction (3.3%), which implies that a larger number of cloaked sites go undetected. Our ground truth showed that nearly 80% of the cloaked sites were malicious, which means nearly 77% of the malicious squatting domains were not detected by VirusTotal. Therefore, the undetected portion of cloaked malicious sites is significant. Our detection rate of cloaked malicious sites is significantly higher compared to past studies [11, 6].

According to Oest et al. [11], cloaking delays and slows down blacklisting. We found 46% of cloaked squatting domains with potential malicious content (from a sample of 2268 domains in list category A in Table 1), continue to cloak content even after 3 months, reaffirming that the techniques used by blacklisting entities are not effective for cloaked sites.

Majority of Internet traffic is originated from mobile users, and mobile browsers are prone to phishing attacks [9]. However, anti-phishing protection in mobile browsers trail behind that of desktop browsers. We observed cloaking of websites (with static content) that are potentially malicious in mobile browser (Chrome) is comparatively higher to desktop browser (Chrome). Bandwidth restrictions imposed by carriers in mobile devices is a barrier to desktop-level blacklist protection [9]. Therefore, at least over a Wi-Fi connection, the full blacklist should be checked by mobile browsers.

Since some major search engine crawlers are also owned by companies who develop browsers (e.g., Google, Microsoft), these companies can complement their existing detection techniques by comparing the views of a web page between a browser and crawler infrastructure, to tackle website cloaking. Some solutions in this aspect are already proposed in past studies [6]. Another countermeasure is to have domain registrars add extra checks in their fraud detection systems to detect domains that are permutations of popular trademarks having a higher entropy. This will facilitate registrars to request more information, if a domain registered is suspicious in carrying out malicious activities under the disguise of cloaking. A similar practice can be adopted by certificate authorities prior to issuing certificates for suspicious domains.

Cloaking may differ based on the geolocation [11] of the user or the language of web content. Also, cloaking behaviors may be different for various search engine crawlers (e.g., Bingbot, Yahoo, Baidu, Yandex) and browsers (e.g., Edge, Internet Explorer, Firefox). We leave such studies as future work.

Acknowledgment

This work is partly supported by a grant from CIRA.ca's Community Investment Program. The second author is supported in part by an NSERC Discovery Grant.

References

- [1] P. Vadrevu, R. Perdisci, What you see is not what you get: Discovering and tracking social engineering attack campaigns, in: ACM Internet measurement conference (IMC'19), Amsterdam, Netherlands, 2019.

- [2] K. Tian, S. T. Jan, H. Hu, D. Yao, G. Wang, Needle in a haystack: Tracking down elite phishing domains in the wild, in: ACM Internet measurement conference (IMC'18), Boston, MA, USA, 2018.
- [3] P. Peng, C. Xu, L. Quinn, H. Hu, B. Viswanath, G. Wang, What happens after you leak your password: Understanding credential sharing on phishing sites, in: ACM Asia Conference on Computer and Communications Security (AsiaCCS'19), Auckland, New Zealand, 2019.
- [4] J. Spooren, T. Vissers, P. Janssen, W. Joosen, L. Desmet, Premadoma: An operational solution for DNS registries to prevent malicious domain registrations, in: Annual Computer Security Applications Conference (ACSAC'19), San Juan, Puerto Rico, USA, 2019.
- [5] T. K. Panum, K. Hageman, R. R. Hansen, J. M. Pedersen, Towards adversarial phishing detection, in: USENIX Security Symposium (USENIX Security'20), Online, 2020.
- [6] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, E. Bursztein, Cloak of visibility: Detecting when machines browse a different web, in: IEEE Symposium on Security and Privacy (SP'16), San Jose, CA, USA, 2016.
- [7] Google, Webmaster guidelines, Online article (2020). <https://support.google.com/webmasters/answer/35769>.
- [8] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, R. Joffe, Enabling network security through active DNS datasets, in: International Symposium on Research in Attacks, Intrusions, and Defenses (RAID'16), Evry, France, 2016.
- [9] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, A. Doupé, Phishtime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists, in: USENIX Security Symposium (USENIX Security'20), Online, 2020.
- [10] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, G.-J. Ahn, Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale, in: USENIX Security Symposium (USENIX Security'20), Online, 2020.
- [11] A. Oest, Y. Safaei, A. Doupé, G.-J. Ahn, B. Wardman, K. Tyers, PhishFarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists, in: IEEE Symposium on Security and Privacy (SP'19), San Francisco, CA, USA, 2019.
- [12] DNSTwist, Dnstwist, Online article (2020). <https://github.com/elceef/dnstwist>.
- [13] Symantec, Webpulse site review request, Online article (2020). <https://sitereview.bluecoat.com/#/>.
- [14] VirusTotal, VirusTotal, Online article (2019). <https://www.virustotal.com/gui/home/upload>.
- [15] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh, N. Asokan, Off-the-hook: An efficient and usable client-side phishing prevention application, IEEE Transactions on Computers 66 (10) (2017) 1717–1733.
- [16] P. Vallina, V. Le Pochat, Á. Feal, M. Paraschiv, J. Gamba, T. Burke, O. Hohlfeld, J. Tapiador, N. Vallina-Rodriguez, Mis-shapes, mistakes, misfits: An analysis of domain classification services, in: ACM Internet measurement conference (IMC'20), Online, 2020.
- [17] Google Safe Browsing, Google Safe Browsing, Online article (2020). <https://safebrowsing.google.com/>.
- [18] B. Wu, B. D. Davison, Detecting semantic cloaking on the web, in: International World Wide Web Conference (WWW'06), Edinburgh, Scotland, UK, 2006.
- [19] T. D. Science, Topic modeling and latent dirichlet allocation (LDA) in Python, Online article (2018). <https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24>.

- [20] R. S. Rao, T. Vaishnavi, A. R. Pais, Catchphish: Detection of phishing websites by inspecting urls, *Journal of Ambient Intelligence and Humanized Computing* 11 (2) (2020) 813–825.
- [21] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, M. Antonakakis, Hiding in plain sight: A longitudinal study of combosquatting abuse, in: *ACM Conference on Computer and Communications Security (CCS'17)*, Dallas, TX, USA, 2017.
- [22] Y. Zeng, T. Zang, Y. Zhang, X. Chen, Y. Wang, A comprehensive measurement study of domain-squatting abuse, in: *IEEE International Conference on Communications (ICC'19)*, Shanghai, China, 2019.
- [23] Z. Zhou, L. Yu, Q. Liu, Y. Liu, B. Luo, Tear off your disguise: Phishing website detection using visual and network identities, in: *International Conference on Information and Communications Security (ICICS'19)*, Beijing, China, 2019.
- [24] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoo, M. Korczyński, W. Joosen, Tranco: A research-oriented top sites ranking hardened against manipulation, in: *Network and Distributed System Security Symposium (NDSS'19)*, San Diego, CA, USA, 2019.
- [25] Hudak, Patrik, Finding phishing: Tools and techniques, Online article (2019). <https://0xpatrik.com/phishing-domains/>.
- [26] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, J. Halderman, A search engine backed by internet-wide scanning, in: *ACM Conference on Computer and Communications Security (CCS'15)*, Denver, Colorado, USA, 2015.
- [27] Cali Dog Security, Introducing certstream, Online article (2017). <https://medium.com/calidog-security/introducing-certstream-3fc13bb98067>.
- [28] Project Sonar, Forward DNS (FDNS), Online article (2020). https://opendata.rapid7.com/sonar.fdns_v2/.
- [29] SuatPhish, Squatting-domain-identification, Online article (2018). <https://github.com/SquatPhish/1-Squatting-Domain-Identification>.
- [30] Google Chrome, Puppeteer, Online article (2019). <https://github.com/GoogleChrome/puppeteer>.
- [31] Selenium, SeleniumHQ browser automation, Online article (2019). <http://www.seleniumhq.org/>.
- [32] Intoli, It is *not* possible to detect and block Chrome headless, Online article (2018). <https://intoli.com/blog/not-possible-to-block-chrome-headless/>.
- [33] Tesseract, Tesseract-OCR, Online article (2019). <https://github.com/tesseract-ocr/tesseract>.
- [34] Michael Carter, pywhois, Online article (2010). <https://pypi.python.org/pypi/pywhois/0.1>.
- [35] Domains by Proxy, Your identity is nobody's business but ours, Online article (2019). <https://www.domainsbyproxy.com/>.
- [36] M. S. Charikar, Similarity estimation techniques from rounding algorithms, in: *ACM Symposium on Theory of Computing (STOC'02)*, Montreal, QC, Canada, 2002.
- [37] RIPE NCC, RIPE NCC, Online article (2020). <https://www.ripe.net/>.
- [38] P. Peng, L. Yang, L. Song, G. Wang, Opening the blackbox of VirusTotal: Analyzing online phishing scan engines, in: *ACM Internet measurement conference (IMC'19)*, Amsterdam, Netherlands, 2019.
- [39] H. Suzuki, D. Chiba, Y. Yoneya, T. Mori, S. Goto, Shamfinder: An automated framework for detecting IDN homographs, in: *ACM Internet measurement conference (IMC'19)*, Amsterdam, Netherlands, 2019.

- [40] K. L. Chiew, K. S. C. Yong, C. L. Tan, A survey of phishing attacks: Their types, vectors and technical approaches, *Expert Systems with Applications* 106 (2018) 1–20.
- [41] D. Y. Wang, S. Savage, G. M. Voelker, Cloak and dagger: Dynamics of web search cloaking, in: *ACM Conference on Computer and Communications Security (CCS'11)*, Chicago, Illinois, USA, 2011.
- [42] PCrisk, `ERR_NAME_NOT_RESOLVED` - How to fix?, Online article (2019). <https://blog.pcrisk.com/windows/12819-err-name-not-resolved>.
- [43] ScrapeHero, How to prevent getting blacklisted while scraping, Online article (2020). <https://www.scrapehero.com/how-to-prevent-getting-blacklisted-while-scraping/>.

Appendix A. Types of cloaking

Domain name squatting is a technique where domains are registered with typographical errors resembling websites of popular brands and trademarks [2]. This leads to abusing the good name of the original brand to attract more traffic than usual for illicit purposes. Squatting domain names are structured by tweaking the names of the original domains (e.g., using encoded Internationalized domain names [39] which are visually similar compared to the original domain). Because of these characteristics in squatting domain names, they are a powerful means in aiding social engineering attacks. Therefore, in addition to showing deceptive page content, social engineering attacks are more successful if these sites are hosted on typo squatting domains [2].

The adversaries who host phishing and malware services want to hide their activities from the search engine crawler [6]. Adversaries use Search Engine Optimization (SEO) techniques when showing fake content to search engine crawlers compared to browser clients, in order to increase the ranking of their illicit sites [40]. Adversaries can also pay advertising networks to show benign advertisements to crawlers, while users view deceptive advertisements that lead to scams and malware [6].

In order to cloak content, the adversary’s web server needs to distinguish the type of client (i.e., crawler vs.

browser) based on an identifier [41], and the choice of the identifier depends on the cloaking technique as described below.

1. In *user-agent cloaking*, the type of client of an incoming request is identified by inspecting the user-agent string. If the user-agent belongs to a crawler, benign content is shown, otherwise malicious content is displayed.
2. With *IP cloaking*, the user is identified using the client IP address of the incoming request. If the IP address of incoming request is within a well known range of public IP addresses of a search engine crawler, benign content is rendered. Otherwise, the IP address most likely belongs to a user/enterprise, in which case malicious content is displayed.
3. *Repeat cloaking* is used to victimize a user on the first visit to the website. In this case, the state of the user is saved at client side (e.g., cookie) or server side (e.g., client IP) to determine a new user visit.
4. *Referrer cloaking* uses the *Referrer* field of the request header to determine if the user clicked through a search engine query result, in which case, the user can be redirected to a scam web page. In Referrer cloaking, adversary’s objective is to target search engine users.

In practice, different types of cloaking are combined and used together.

Appendix B. Issues during crawling

In this section, we explain the errors, disallowing of requests by web servers and failures encountered during crawling of websites. The data shown in this section are based on squatting domain list category *A* in Table 1.

Errors during crawling. We crawled 100,000 sites hosted on squatting domains by imitating the ChromeUA and GooglebotUA user-agents. Out of them, 9712 (9.7%) and 9899 (9.9%) requests encountered errors during crawling from ChromeUA and GooglebotUA. Requests initiated from GooglebotUA had a slightly higher number of errors. Table B.9 shows the top 5 errors. Most errors were due to timeouts; ChromeUA (4423, 4.54%), GooglebotUA (4502, 4.5%). We set a 30 seconds timeout for each request made from the crawler, as it is a reasonable

Error	ChromeUA	GooglebotUA
Navigation Timeout Exceeded: 30000ms exceeded	4423	4502
ERR_NAME_NOT_RESOLVED	1640	1594
Execution context was destroyed, most likely because of a navigation	893	584
ERR_CONNECTION_REFUSED	820	608
ERR_CERT_COMMON_NAME_INVALID	343	341

Table B.9: Top 5 errors encountered during crawling

time interval within which a web page can load. Setting a higher timeout value not only reduces our ability to crawl a larger number of URLs within a reasonable time period, but also increase the chance of crashing the crawler. If the timeout is increased from 30 to 60 seconds, we were able to successfully crawl more sites, although adhoc crashing of the crawling automation is experienced. However, in this case, the timeout errors observed was lower than having a 30 seconds timeout; ChromeUA (3418), GooglebotUA (3745). ERR_NAME_NOT_RESOLVED are DNS related errors that are most likely to be caused by issues related to client browser issues or firewall settings [42]; ChromeUA (1640, 1.6%), GooglebotUA (1594, 1.6%). To validate this aspect, we crawled sites that resulted in ERR_NAME_NOT_RESOLVED errors from a separate residential machine located in the same city, and found a significant proportion of them didn't show this error; ChromeUA (316, 0.3%), GooglebotUA (380, 0.4%). Some of these sites even didn't return an error from the new location; ChromeUA (219, 0.2%), GooglebotUA (263, 0.3%). ERR_CONNECTION_REFUSED errors are usually caused by DNS, proxy server or browser cache issues. Some sites threw errors due to loosing of its execution context. This can happen when a web page loses its execution context while navigating from the crawler. Therefore, running a callback relevant for a specific context that is not applicable during the current navigation can throw an error. ERR_CERT_COMMON_NAME_INVALID errors signal a problem with the SSL/TLS connection where the client cannot verify the certificate.

Failures based on user-agent. Web hosting providers often block clients with unusual traffic. We observed 152 (0.2%) sites were blocked from GooglebotUA by the web

hosting provider. In order to block requests from bots (e.g., GooglebotUA), web hosting services use different techniques to identify them [43]. For example, honeypots consisting of links that are only visible to bots are used to attract crawlers, to detect and have them blocked [43]. Different types of content observed in these blocked sites are in Table B.11. Content of some of these sites are in Chinese (e.g., <http://diirk.com>). Also, during our crawling, we noticed some sites did not accept requests initiated from automated crawlers. These failures depend on the user-agent of the request. We show these failures in Table B.10. Some sites showed "Too many requests" failures when requesting a site from both GooglebotUA and ChromeUA. This behavior was consistent between ChromeUA (2640, 2.6%) and GooglebotUA (2448, 2.4%). This failure was also observed from a real browser when the site was requested repeatedly. We found "404 - File or directory not found" errors were more than six times higher with GooglebotUA (472, 0.5%) compared to ChromeUA (72, 0.07%). The robots.txt file which is in the root directory of a website can be configured to prevent automated crawlers from requesting the site [43]. However, some of these sites may not want to block popular search engine crawlers such as Google, as otherwise it will impact their site ranking. We found 19,040 (19%) websites were disallowed according to the rules in robots.txt which is significant. Our crawler is able to scrape the content of these websites. From these sites, 4722 (4.7%) showed benign content, and the rest of them mostly contained a listing of links and phishing/malware related content. Out of the sites that are disallowed from robots.txt, a smaller fraction showed "Too many requests" failures; ChromeUA (1583, 1.5%) and GooglebotUA (1460, 1.4%).

Failure	ChromeUA	GooglebotUA
Too many requests	2640	2448
Page cannot be displayed. Please contact your service provider for more details	1368	1369
404 - File or directory not found	72	472

Table B.10: Failures while crawling

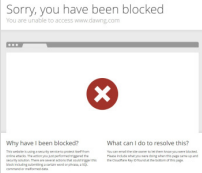

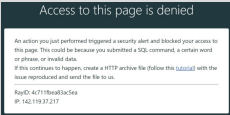
Page content	No. of sites	Example site
Sorry, you have been blocked. You are unable to access [DOMAIN] Why have I been blocked? This website is using a security service to protect itself from online attacks.	136	http://dawng.com 
Your request has an illegal parameter and has been blocked by the web-master settings!... (Chinese translation)	12	http://diirk.com 
..Access to this page has been denied.. An action you just performed triggered a security alert and blocked your access to this page. This could be because you submitted a SQL command, a certain word or phrase, or invalid data. ...	4	https://support.bed-booking.com 

Table B.11: Sites blocked from GooglebotUA

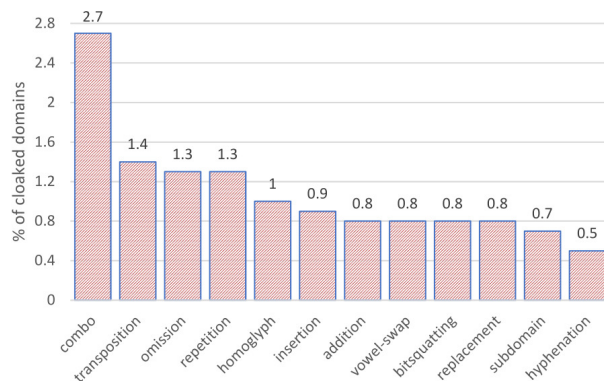


Figure C.6: Cloaking by type of squatting domain.

Appendix C. Relevance of type of squatting domains for cloaking

Most cloaked sites are hosted on combo-squatting domains as shown in Figure C.6. This may mean that combo-squatting domains are more effective in cloaking phishing and malware site content. Kintis et al. [21] find most combo-squatting domains are not remediated for a long period of time (sometimes up to 1000 days). Therefore, many occurrences of abuse happen before they are detected by blacklists.

Appendix D. Relevance of cloaking by other factors

The top 10 countries hosting the largest number of squatting domains with cloaked content were United States (1508), Germany (145), Netherlands (53), Australia (53), Seychelles (41), Canada (34), Switzerland (26), Japan (17), France (16) and British Virgin Islands (15). Therefore, most of these cloaked sites were hosted

in the United States and Germany. Tian et al. [2] observed a similar pattern where most phishing sites are spread in these countries. The top 5 registrars of squatting domains hosting cloaked content were GoDaddy (477), Sea Wasp (225), Xinnet Technology Corporation (115), Tucows, Inc. (84), Enom, Inc. (82). GoDaddy had registered the most number of cloaked domains.