Where are you taking me? Understanding Abusive Traffic Distribution Systems

Janos Szurdi
Carnegie Mellon University
Palo Alto Networks
jszurdi@alumni.cmu.edu

Meng Luo
Stony Brook University
meluo@cs.stonybrook.edu

Brian Kondracki
Stony Brook University
bkondracki@cs.stonybrook.edu

Nick Nikiforakis
Stony Brook University
nick@cs.stonybrook.edu

Nicolas Christin
Carnegie Mellon University
nicolasc@andrew.cmu.edu

ABSTRACT

Illicit website owners frequently rely on traffic distribution systems (TDSs) operated by less-than-scrupulous advertising networks to acquire user traffic. While researchers have described a number of case studies on various TDSs or the businesses they serve, we still lack an understanding of how users are differentiated in these ecosystems, how different illicit activities frequently leverage the same advertisement networks and, subsequently, the same malicious advertisers. We design ODIN (Observatory of Dynamic Illicit ad Networks), the first system to study cloaking, user differentiation and business integration at the same time in four different types of traffic sources: typosquatting, copyright-infringing movie streaming, ad-based URL shortening, and illicit online pharmacy websites.

ODIN performed 874,494 scrapes over two months (June 19, 2019–August 24, 2019), posing as six different types of users (e.g., mobile, desktop, and crawler) and accumulating over 2TB of data. We observed 81% more malicious pages compared to using only the best performing crawl profile by itself. Three of the traffic sources we study redirect users to the same traffic broker domain names up to 44% of the time and all of them often expose users to the same malicious advertisers. Our experiments show that novel cloaking techniques could decrease by half the number of malicious pages observed. Worryingly, popular blacklists do not just suffer from the lack of coverage and delayed detection, but miss the vast majority of malicious pages targeting mobile users. We use these findings to design a classifier, which can make precise predictions about the likelihood of a user being redirected to a malicious advertiser.

KEYWORDS

Traffic, Distribution, Web, Security, Mobile, Phone, User, Cloaking

ACM Reference Format:
Janos Szurdi, Meng Luo, Brian Kondracki, Nick Nikiforakis, and Nicolas Christin. 2021. Where are you taking me? Understanding Abusive Traffic Distribution Systems. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450071

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450071

1 INTRODUCTION

Online advertising subsidizes the World Wide Web: ads monetize user visits and pay for infrastructure. Unsurprisingly, as a lucrative business, online advertising also invites abuse. For instance, questionable or illicit sites automatically redirect users to advertisers [1, 3, 21, 24, 25, 34, 38, 45, 55] without user consent. Dubious redirections of visitors also frequently expose them to malicious content, including deception, phishing, scams and malicious downloads [1, 16, 21, 28, 34, 36, 38, 45, 59]. While the research community has documented a number of abusive practices through specific case studies [1, 4, 8, 15, 16, 21, 22, 24, 25, 27, 28, 34–36, 38, 45, 49–51, 55, 58, 59, 63, 66], we still lack a general understanding of how malicious advertisement ecosystems interact with each other, of the specific roles different entities assume, and, more generally, of how different the landscape is, between mobile and desktop web users.

Generally, (legitimate) advertising on the web works as follows. Websites include content from sources called *ad publishers*, who themselves leverage a complex system of advertisement networks to choose, on-the-fly, which ad (provided by an *advertiser*) to display for a given user, during a given browsing session. To maximize engagement ("clicks"), displayed ads are selected through a combination of behavioral user profiling and a bidding process among advertisers based on user profiles. This model is called "pay-per-click" (PPC) since ad publishers are rewarded as a function of the number of clicks generated by their website. We refer the reader to Pearce et al. [41] for an extensive description of the advertising ecosystem.

Clicks require active user participation. A much more aggressive technique is to instead *automatically* redirect users to a target destination website – in such a context, ad publishers are compensated through pay-per-redirect (PPR). Both PPR and PPC form the bedrock of the *traffic distribution systems* (TDSs) used by advertisement networks to direct traffic to advertisers. PPR, however, is far more intrusive than PPC, and is frequently observed along with malicious or abusive behavior [1, 38].

Using terminology from the literature [3, 25], TDSs connect *traffic sources*—pages visited by users for content (e.g., free movies), for services (e.g., URL shorteners), or by accident (e.g., typing mistake)—to *destination pages* (advertisers). *Traffic brokers* match traffic sources (and potentially user profiles) with the highest bidding advertiser. In the PPR model, this often involves a brief visit to one or more separate websites run by the TDS operators before reaching the destination page. This entire journey from traffic

source, to intermediate traffic brokers, to destination (or "landing") pages, constitutes a *redirection chain*.

Importantly and differently from legitimate advertisers, malicious destination page operators are agnostic to the techniques TDSs use to bring traffic to their websites. Indeed, these malicious operators are merely customers of the TDSs. These operators' own monetization strategies rest on other techniques, such as, deceiving users into sharing sensitive information, stealing funds, or serving a malicious or potentially unwanted programs (PUPs).

Contributions. This paper 1) describes a measurement infrastructure called ODIN (Observatory of Dynamic Illicit ad Networks) which allows us to compare how campaigns differentiate over multiple types of users (from different vantage points, using different browsers, hardware, etc.), 2) presents novel results from at-scale data collection using ODIN, and 3) introduces possible countermeasures based on these findings. While previous studies considered questionable ads relevant to specific ecosystems, or relying on specific techniques, the key novelty is in ODIN's ability to take a broader view, which enables us to discover how questionable advertisers perform per-user differentiation to monetize their traffic.

ODIN's goal is to offer a systematic exploration of various TDSs used by questionable content providers. To do so, ODIN collects screenshots, HTTP communications, content and browser logs. We semi-automatically label 101,926 pages that ODIN collected, and we use these labels to perform a series of automatic analyses of page contents to better understand the threats these TDSs pose.

In the past, researchers have either individually studied illicit traffic sources [1, 25, 30, 35, 38, 45, 55] or focused on a single malicious activity [8, 16, 21, 34, 44, 49]. Expanding this body of work, we study multiple traffic sources and a wide variety of malice stemming from them. We seed ODIN with four distinct types of traffic sources: (i) "typosquatting sites" [55] (e.g., `yotube.com`), (ii) copyright-infringing sites that stream pirated movies [15], (iii) ad-based URL shortening services that shorten URLs in return for exposure to potentially malicious ads [38], and (iv) unlicensed online pharmacies [24]. We choose these traffic sources as they are known to redirect users to malicious or illicit landing pages. At the same time, previous studies have generally not exhibited much overlap between these various activities, which allows us to test the hypothesis whether TDSs are "vertically integrated" (i.e., each criminal coterie uses their own TDS infrastructure) or if they cross-cut multiple segments. Earlier results [24] hinted at vertical integration, at least in the pharmaceutical ecosystem; revisiting this finding a decade later, we discover that *vertical integration no longer holds*.

The vast majority of papers before 2016 [1, 4, 8, 17, 24, 25, 27, 28, 35, 38, 45, 55, 62, 63, 66] simply did not consider cloaking. Even after 2016, most papers [15, 19, 21, 22, 34, 49, 58, 59], only accounted for a couple of aspects of cloaking. ODIN assumes all of the participants in the TDS ecosystem are malicious and attempt to cloak their activities, or evade detection through blocking. Despite this adversarial landscape, we show that ODIN can successfully reconstruct redirections. As a side-benefit, ODIN allows us to unearth a wide variety of cloaking techniques.

Crucially, ODIN emulates a variety of different profiles (web crawler, desktop users, mobile users) – using a combination of user emulation and actual mobile hardware – and compare TDS behavior across these different user profiles. ODIN also relies on various proxying techniques to examine IP address-based differentiation in TDS responses. We open sourced ODIN on GitHub [18] and make the collected and labeled data available for researchers upon request.

Results. Using ODIN, we scraped webpages 874,494 times over two months (June 19, 2019–August 24, 2019), accumulating 2TB of data. Posing as six different types of users, ODIN finds 81% more malicious and 96% more suspicious landing pages, compared to visiting pages only using the user profile which experienced the most malice. We find that mobile users are exclusively targeted with deceptive surveys and illicit adult content tailored to them. Conversely, desktop users are exposed to technical support scam pages and deceptive downloads that mobile users never see. Our experiments also show that some state-of-the-art blacklists do not include the vast majority of malicious destination pages mobile users are exposed to.

From a criminal ecosystem standpoint, we find evidence of TDS reuse *across* illicit activities. Some traffic source pairs share 44% of traffic broker domains they use. TDSs also redirect to the same kind of landing pages, and nearly half of the different types of malicious activities we found were present in the typosquatting, copyright infringing, and the URL shortening ecosystems. Shared malice includes technical support scams [34, 50], deceptive surveys [8, 21], deceptive downloads [1, 59], and other scams. At the same time, certain types of abuse are prominent at only one TDS. For example, copyright-infringing sites force users' social media activities such as tweets and shares. URL shortening services advertise crypto-currency related scams. Typosquatting domains redirect to fake identity protection phishing sites.

Miscreants still leverage IP reputation, user agent and the referrer HTTP header fields to cloak their activity. Additionally, we observe that most of the malicious entities leverage simple techniques to block or to cloak their activity, but do not appear to use more advanced techniques such as the detection of mobile phone emulation or WebRTC-based proxy detection. Comparing results obtained from a pool of 240 IP addresses with those obtained from a single vantage point, we find that, in addition to rate limiting, some TDSs attempt to escape detection by disproportionately redirecting suspected crawlers like ODIN to benign pages instead of their usual landing pages, resulting in half as many malicious pages observed.

2 RELATED WORK

Our paper extends multiple areas of research that have explored TDSs [16, 27, 28, 36, 50, 51, 59, 66], illicit traffic sources [1, 4, 15, 24, 25, 34, 35, 38, 45, 49, 55, 63] and cloaking [17, 19, 39, 62].

TDSs. Early research of TDSs has focused on malicious advertising in Alexa top domains [27, 28, 66]. While popular domains might redirect users to malicious destination pages from time to time, questionable businesses frequently redirect users to abusive or malicious landing pages. Even though researchers have studied these potentially dangerous websites [24, 35, 38, 45], there has been no research on how they constitute together a complex interconnected network supporting online crime. Closest to our work is research by Vadrevu and Perdisci [59] that focused on investigating traffic broker domains to find more malicious destination pages. Conversely, our goal is to study and compare traffic sources, while quantifying the effects of user differentiation and cloaking techniques.

Illicit traffic sources. To gain a clear picture of the malicious advertisement ecosystem, we study four traffic sources— typosquatting, ad-based URL shortening services, copyright-infringing movie streaming websites, and illicit pharmacies. We selected these sources based on the diversity of how they attract user traffic.

Typosquatters register misspelled variants of domain names, such as `yotube.com`, to profit from users' typing mistakes. Despite having been studied for over fifteen years [1, 4, 6, 11, 20, 29, 35, 42, 48, 53–56, 58, 63, 65, 67], typosquatting still occurs, with little abatement. Complementary to this body of work, we look at typosquatting as part of a broader criminal ecosystem. We also account for the impact of cloaking, as well as focus on how users are differentiated, and how they end up on malicious pages.

URL shortening services transform complex URLs with user-friendly shorter variants. Nikiforakis et al. [38] have shown that third-party ads used in ad-sponsored URL shortening services expose users to a diverse type of abusive content, including drive-by downloads, online scams, and illicit adult contents.

Copyright-infringing movie streaming sites offer pirated content to profit from users intentionally or accidentally clicking on ads while trying to watch movies. Researchers have focused on the infrastructure supporting the sharing of pirated content [15], but have not investigated abuse. Closer to our research, Rafique et al. [45] studied sport-streaming sites that expose users to malicious content similar to illicit movie streaming sites. Studying pirated movie streaming sites gives us a complementary datapoint.

A few studies [24–26, 30, 32, 62] have investigated how unlicensed online pharmacies acquire traffic, through email spam or search poisoning finding early evidence of cloaking (e.g., HTTP header and cookie-based). Interestingly, these studies all suggest that the unlicensed online pharmaceutical industry appears to be a relatively "closed" ecosystem, at least in the early 2010s. Traffic brokers serving pharmacies, in particular are (or were) rarely shared with other businesses. By complementing online pharmacies with three other traffic sources, we see that while pharmaceuticals are indeed an outlier, there is a significant amount of overlap between other types of activities.

Malice on the Web. Another body of work focused on uncovering different types of malice, such as drive-by-downloads [16, 44], phishing pages [31, 64], technical support scams [34, 49] or survey scams [8, 21]. Our research is different in that we consider a wide variety of abuse in the TDSs we study.

Cloaking. TDS operators and other miscreants often engage in "cloaking." In trying to determine how the literature addresses cloaking, we surveyed twenty-three measurement papers [1, 4, 8, 15, 17, 19, 21, 22, 24, 25, 27, 28, 34, 35, 38, 45, 49, 55, 58, 59, 62, 63, 66] that engage in active crawling of Web content from TDSs, illicit traffic sources or destination pages.

With the exception of Wang et al. [62], most papers published before 2016 did not take explicit steps to study or mitigate adversarial cloaking. On the other hand, most papers published after 2016 (and Wang et al. [62]) use a combination of one or more of the six following methods: (i) changing the user-agent, (ii) setting an HTTP header field, (iii) mitigating browser fingerprinting, (iv) changing the type of IP address used, (v) rotating through IP addresses to eschew rate limitation, and (vi) avoiding proxy detection. While most papers only consider HTTP header based cloaking techniques, a couple of papers [17, 21, 59] combine multiple defenses. ODIN combines *all* of these techniques to mitigate cloaking attempts.

3 DATA COLLECTION: ODIN

Our data collection must fulfill several objectives. The primary goal is to understand if and how disparate traffic sources are leveraging the same traffic brokers and cloaking techniques. At the same time, we cannot exhaustively search for all possible malicious activity on the web; we thus will have to focus on a subset of possible sources, that must be *diverse* and *representative*. Second, our infrastructure must be *resilient to cloaking* and evasions by TDS operators.

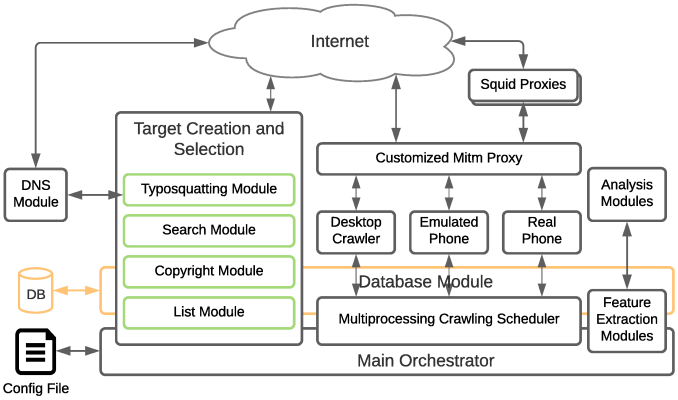


Figure 1: High-level overview of ODIN.

To meet these objectives, we designed the collection infrastructure represented in Figure 1. For each traffic source we study (typosquatting, ad-based URL shortening, illicit movie streaming, and illicit pharmacy sites) we have a separate module to select URLs that ODIN visits. These URLs are then ordered by a scheduler to avoid being detected by TDSs that are looking for multiple visits from the same IP address in quick succession. In an effort to determine differences in treatment between user types, each URL is visited by (a combination of) various collection agents: three desktop crawlers, an agent mimicking a Google bot, an emulated phone, and an actual phone. Finally, ODIN extensively relies on proxies to pretend the visits are coming from various, unrelated connections.

3.1 Target Creation and Selection

We generate a new set of target URLs for every run of an experiment. The only exception is the URL shortening dataset, where we create target URLs once, before starting the experiment.

Typosquatting. The main typosquatting dataset *typo-main* consists of all possible Damerau-Levenshtein distance one [9] variants of Alexa's top 500 domain names. Using the DNS module, we select only those domain names that responded with valid `NS` and `A` records. We generate the full list of typosquatting domains and randomly select 2,000 domains for every collection round.

URL shortening services. To create URL shortening targets, we create URLs pointing to Alexa's top 20 domain names at 15 URL shortening services. This selection is a trade off between the limited number of target URLs that our crawling infrastructure can visit

User Profile	User-agent	Mobile Emulation	Referrer	Proxy
Vanilla Desktop	Windows Chrome	No	None	Yes
Referrer Desktop	Windows Chrome	No	Google	Yes
No-Proxy Desktop	Windows Chrome	No	Google	No
Google Bot	Google Bot	No	None	Yes
Emulated Phone	Android Chrome	Yes	Google	Yes
Real Phone	Android Chrome	–	Google	Yes

Table 1: Summary of user profiles. All experiments are conducted from Linux servers, except "Real Phone" for which a Nexus 6P Android was used.

daily and the expectation that our infrastructure can reach more malicious campaigns. For each experiment, we use all 300 target URLs in our URL shortening dataset.

Illicit pharmacies. We query the Google Search API with a set of pharmaceutical-related search terms curated by Leontiadis et al. shown to produce strong coverage [24, 25]. We freshly generate and select a maximum of 2,000 URLs for each experiment we run.

Copyright-infringing websites. We collect URLs from `softonic.com`, a site crowdsourcing answers and rankings of answers to all sorts of user questions. The site's statistics claimed that tens of thousands of users voted on sites in their list of "best free movie streaming sites." We compare this site's crowdsourced solution to querying Google's search API with related keywords and movie titles. We found that Google appears to effectively scrub copyright-infringing sites from its search results as we only find a fraction of the sites listed on `softonic.com`. For each experiment, we harvest 300 URLs from approximately a hundred sites.

3.2 User Emulation

One of our key objectives is to examine how users are differentiated. To do so, ODIN emulates various types of users. As a side-benefit, our setup allows us to discover some of the cloaking techniques miscreants use. More specifically, we scrape each URL target six times using the six different user profiles, as shown in Table 1. For all these profiles, we rely on a fully-featured, headless Chrome browser, governed by Selenium.

Desktop users. The vanilla desktop crawler mimics a desktop user browsing with Google Chrome using a common Windows Chrome `User-agent`. To combat referrer-header-based cloaking as observed by previous work [24, 25], we also use a modified version of the vanilla crawler where we set the HTTP referrer header to `https://google.com` for our initial query. ODIN visits each target URL with and without an anonymous (Squid) proxy to better understand the impact of proxy usage on measurements.

Mobile Phone users. We emulate a mobile phone browser to study our hypothesis that TDSs treat phone users differently than how they treat desktop users. We use Chrome's mobile emulation option, and additionally set the correct window size, pixel ratio, and `User-agent` to emulate a popular Android phone. To understand if TDSs detect phone emulation (which has been shown to be trivial [60]), we also use a Nexus 6P phone with a modified version of Chromium. Faulty testing hardware caused the phone to crash and shut down during our experiment. As a result, we were only able to scrape around 50% of target URLs from our phone. Fortunately, due to ODIN randomizing the target URLs, this error has the same effect as random sampling.

Google Bot. Certain malicious sites hide their activity or show a search engine optimization page when visited by Google's crawler [24]. To observe how TDSs react when encountering a search engine crawler, we set the `User-agent` to Google's crawler.

3.3 Cloaking Detection and Avoidance

A particularly important feature of ODIN is to explicitly consider adversarial behavior from TDSs, and to attempt to detect, and circumvent, cloaking. This is partly done through the multiple scrapes from various user types described above, and complemented through the following assortment of techniques.

Self rate-limiting. Certain traffic sources, especially typosquatters, cloak their malicious activity after only a few visits from the same IP address. To combat IP-based cloaking, ODIN's scheduler tries to schedule related URLs as far apart in time as possible. Two URLs coming from the same traffic source are considered related; in addition, using the DNS module, ODIN determines that two URLs are related if their domains share identical `NS` or `A` DNS records. ODIN further attempts to mitigate IP-based cloaking by randomly sampling URLs from the four traffic sources to only visit at most 3,000 URLs every other day.

Anti-browser fingerprinting. Some of the simplest methods to figure out automation include the detection of `User-agents` and the lack of JavaScript execution or handling of cookies. These are already taken care of by using a full featured browser, as discussed above. To address some of the slightly more sophisticated browser fingerprinting approaches we modify properties of our browser by changing the window size, adding extensions, and adding a default language. The array of browser fingerprinting techniques that enable stateless tracking is vast and therefore our approach may not be able to cover all possible techniques, e.g. an attacker recognizing ODIN via canvas fingerprinting [5].

IP rotation. Miramirkhani et al. [34] observed that typosquatters cloak malicious activity if their pages are visited from a large datacenter's IP addresses. Thus, ODIN uses university IP addresses (one per profile) and a /24 subnet from a research-friendly, but less well-known VPS provider [43]. We do not leverage residential IP addresses to avoid ethical quandaries [33], and because research [19] has shown that using university addresses is a good alternative.

Proxy detection avoidance. The simplest way to utilize multiple IP addresses is to use proxies which can unfortunately be detected. To avoid proxy detection and since we control the proxy software deployed on the aforementioned vantage points, we scrub headers such as the `via` and the `forwarded_for` HTTP headers. To study if there are attackers who leverage more advanced proxy detection (e.g., WebRTC-based detection), we also collect pages using a crawler that does not use proxies. Additionally, we emulate mouse movements to address user behavior-based detection. In the case of sites streaming pirated movies, we also click on the play button, as a user would, to trigger stealthy HTML overlay redirections.

3.4 Experiments

In this paper, we use ODIN to collect data through two experiments.

Main experiment. In this experiment our goal is to understand the shared dependencies between the four traffic sources and differentiation of phone and desktop users. For the Main experiment, ODIN

Label Classes	Labels
Error	Crawl Error, Error, Blocked
Benign	Empty, Parked, Original, Adult, Gambling, Online Pharmacy, Defensive
Illicit	Illicit Pharmacy, Keyword Stuffed, Affiliate Abuse, Illicit Adult
Suspicious	Survey, Download, Other
Malicious	Technical Support Scam, Crypto Scam, Other Scam, Deceptive Download, Malicious Download, Deceptive Survey, Impersonating, Phishing, Forced Social, Black hat SEO, Other Malicious

Table 2: Summary of labels and label classes.

collected 490,094 pages during a two-month period. Altogether, for the Main experiment, we visited every URL six times from six different IP addresses to address user differentiation and cloaking.

IP-Cloaking experiment. The goal of our secondary experiment is to quantify and better understand IP-address-based cloaking. In this experiment we use ODIN to visit pages using two different types of anonymous proxies. The first proxy uses only one IP address, while the second proxy rotates through 240 different IP addresses.

This experiment presents a couple of other differences compared to the aforementioned Main experiment. ODIN uses only four out of the six available user emulations. We do not visit pages using a real phone, and we do not use our "No-Proxy" profile explained in Section 3.2. On the other hand, we use five other datasets sampling 2,000 benign domains from Alexa's top 1 million list [2], 6,000 typosquatting domains targeting less popular Alexa domains [55], 2,000 domains targeting Alexa popular pharmacy domains, 1,000 domains from PhishTank [40], and 5,000 domains from SurBL [52].

By repeating the IP-Cloaking experiment three times between June 24, 2019 and August 19, 2019, we collected 441,457 pages finding that when we use multiple IP addresses, we observe significantly more malicious destination pages.

4 DATA LABELING

To understand potential infrastructural overlap between different illicit activities as well as user differentiation in TDSs, ODIN altogether visited 78,668 webpages over two months (June 19, 2019–August 24, 2019). Posing as different "users" (crawlers, desktop, and mobile users) over different IP addresses, ODIN ended up performing 874,494 separate URL visits, from which it collected 931,551 pages,[1] which produced over 2TB of screenshots, browser events, and archived HTTP communications.

Unfortunately, we have no externally-provided, reliable labels telling us which pages are malicious, abusive, or illicit. To address this problem, we start by semi-automatically labeling 101,926 pages into fine-grained categories. We then automatically extrapolate the manual labels to the remaining 829,625 pages. We create specific classification rules, classifiers, or collect additional information for certain labels, including illicit pharmacies, malicious downloads and impersonating pages. As a by-product of this classification, we conclude this section by discussing the feasibility of predicting whether a user will be redirected to a malicious landing page solely based on the redirection chain traversed.

4.1 Labels

Table 2 summarizes the labels we use to classify destination pages ODIN visits. These labels express the different kinds of abuse ODIN encountered, and can be grouped into five classes: error, benign, illicit, suspicious, and malicious.

Error labels. We label errors caused by our infrastructure as "crawl error," most frequently due to one of our proxies not working. When we are explicitly blocked, then we tag the page as "blocked." All other errors are labeled as "error."

Benign labels. We label pages as "empty" when we find little or no content. For simplicity the "parked" label aggregates together pages consisting of ads, trying to sell domain names, under construction, under-developed or serving an HTTP server default page. The "adult" and "gambling" labels include any related content, for example including adult games, dating sites, and lotteries. Pharmacies that do not leverage compromised sites are labeled as "pharmacy." All benign pages with substantial content that do not fit any of the other benign categories are labeled as "original content." We label defensive registrations where brand owners proactively register the typosquatting variants of their domain name as "defensive."

Illicit labels. We label all online pharmacies leveraging compromised sites for black-hat search engine optimization [24] and storefront hosting as "illicit pharmacy." When we visit these same pages posing as a Googlebot, they often present pages full of keywords, in an effort to game search-engine rankins and attract more visitors. We label these pages as "keyword stuffed." Sites abusing affiliate programs by automatically redirecting users to advertisers are labeled as "affiliate abuse." In a couple of cases, ODIN was redirected to adult pages of dubious legality. We discard these screenshots, only keeping their hashes, and label the corresponding pages as "illicit adult."

Suspicious labels. When ODIN is redirected to suspicious pages offering a download or a survey, but there is no deception involved, then we label them as "download" or "survey" respectively. When a page is engaging in suspicious activities (for example an otherwise empty page is asking us to enable notifications) we then tag the page as "other" as we are not sure about the intent.

Malicious labels. When deception is involved, we label download and survey pages as "deceptive download" or "deceptive survey." Deceptive download pages try to scare users into downloading files telling them, for example, that their flash player is outdated or warning them that they might have vulnerabilities or even viruses. When a downloaded file is malicious, we then label the page as a "malicious download" if the page does not have another malicious label.

We label pages informing us that we have been selected to receive free products or money as "deceptive survey" or "other scam" depending on whether filling out a survey is required. Often these pages ask users to perform several tasks such as filling out surveys, asking for personal information, and downloading applications. We also label pages offering high-yield investments or high-paying jobs not requiring any specific skills as "other scam." We label pages offering free crypto currency mining or large amounts of crypto rewards as "crypto scam." We label pages that are clearly set up to steal a user's personal data as "phishing." We distinguish pages impersonating online services to trick users into sharing their credentials as "impersonating." We label pages as "tech scam" if they try to scare users into believing that their machine is infected and that paying for technical support offered on the page is necessary to clean their computer.

[1] Scraping a URL results in multiple pages and screenshots collected, if new windows are opened in the browser automatically.

Certain pages craft HTTP redirects to try to automatically initiate some user action. In particular, we label pages that attempt to force users into engagement on a social network, like tweeting or sharing, as "forced social." Other pages redirect users to a Google search to manipulate their brands' or sites' search ranking: we label these as "black hat SEO." Finally, we label pages as "other malicious" when users are presented with deceptive warnings or error messages, but where the malicious use case is not immediately clear.

Multiple tag label. URL shortening services might present users multiple different types of content. We label them as "multi tag," to avoid combinatorial explosion in the number of categories our classifiers will have to predict.

4.2 Clustering and Data Labeling

Using the labels described, we cluster and semi-automatically label 101,926 pages collected between June 19 and July 4 in 2019. These labels form the bedrock of our subsequent (automated) classification.

We start by leveraging several approaches to cluster pages. These methods include grouping pages by matching text or perceptual hash [14], and clustering using the k-nearest neighbor algorithm (KNN). The KNN clustering uses the last layer of DenseNet 201 model trained on the ImageNet dataset from the Keras library [7] as features. Additionally, we use regular expressions based on previous work [55] to classify parked pages, and simple heuristic rules based on the HTTP error code received and the text shown to users to find error pages. These enable us to label 65,276 pages.

The remaining 36,650 pages feature 14,746 unique perceptual hashes. We randomly selected a page for each different hash, and then had it manually labeled by at least two researchers. Inter-coder agreement was high, with a Cohen's kappa score of 0.81. When manual labels did not match, a third researcher broke the tie, or the label was further discussed as a group when deemed necessary. We then labeled the remaining 21,904 pages by propagating identical labels to all pages sharing the same perceptual hashes.

As a final validation check, we randomly selected a maximum of a hundred screenshots for each label, adding up to 1,607 labels, which we verified again. Only 43 screenshots (2.67%) had the wrong label. We find that 42 of these mislabeled pages consisted of error, blocked, parked or empty labels. Such pages often have little content, which causes perceptual hashing to be too coarse. However, we find this inaccuracy acceptable for our purposes, as we do not necessarily need to distinguish between error pages and under-developed pages.

4.3 Tag Extrapolation

After our manual labeling, 388,168 pages in the Main experiment and 441,457 pages in the IP-Cloaking experiment remain unlabeled. To label these pages we train a RF (Random Forest) classifier. We compile a list of features both from related work [61] and from our domain experience. The features include content and DOM-related features such as the page size, number of frames, number of unique HTML tags, number/ratio of internal/external links, text size, link to text ratio, ordinal encoded perceptual hashes of the screenshots, number of total/unique/ratio of pharmacy-related words and the number of unique words. We find the Random Forest classifier performs best with `n_estimators=32` and `min_samples_split=2`.

Label	Precision	Recall	Label	Precision	Recall
Error	0.87	0.89	Phishing	1.00	1.00
Blocked	0.99	0.98	Deceptive Survey	0.81	0.97
Crawl Error	0.97	0.94	Deceptive Download	0.94	1.00
Empty	0.99	0.91	Tech Scam	0.96	0.98
Parked	0.94	0.84	Crypto Scam	1.00	1.00
Original Content	0.86	0.59	Other Scam	0.90	0.99
Gambling	0.90	0.98	Other Malicious	0.94	1.00
Pharma Store	1.00	0.96	Download	0.95	0.93
Adult Content	0.96	0.96	Survey	0.99	1.00
Keyword Stuffed	0.90	1.00	Other	1.00	1.00
Illicit Adult	1.00	1.00	Multi Tag	1.00	1.00

Table 3: Per-class precision of our multi-class RF classifier. As our goal was to evaluate precision, we selected a hundred samples per class, which results in the recall for classes with many elements to be biased negatively. Recall appears lower for these classes as the number of positive examples is disproportionately underrepresented.

Our classifiers have a 97.7% accuracy and 97.0% average precision over our classes evaluated on a 10% validation set. After predicting labels in our unlabeled datasets, we evaluate the classifier on a maximum of 100 random samples for each label from the previously unlabeled dataset. The average precision drops to an acceptable 94.9%. Table 3 lists the per class precision of the RF model. For the rest of the paper we use the combination of our manual labels and results from the RF model's predictions.

5 AUTOMATIC LABELING METHODOLOGY

In this section, we describe additional specialized classifiers and heuristic rules we use to label pages.

We train a Random Forest classifier building on the observation by previous work [24] that illicit pharmacies will respond with different web content to HTTP queries from our different user profiles. The model relies on features calculated for all scrapes of a page, including number/ratio of external links/sources, link to text ratio, number/unique/ratio of pharmacy-related words, length of domain redirection chain, landing error code, number of external source domains. On a test set of 200 sample pages our classifier's precision is 97% and the recall is 93.1%.

A typosquatting page is labeled "defensive" if it is owned by a known brand protection company or directly redirects to the brand owner's original domain. Leveraging the methodology of Szurdi et al. [55], if a typosquatting domain name redirects to a non-malicious content through one or more different intermediate traffic broker domain names, then we label it as "affiliate abuse."

We label a page as "malicious download" if we downloaded a file from the page, the file was tagged malicious by at least one VirusTotal vendor, and the page was not previously assigned a different malicious label. We determine which URLs lead to "forced social" media actions by searching through the developer APIs of Facebook, Twitter, and LinkedIn and recording which endpoints correspond to each action. We find that TDSs discretely redirect users to search engines (e.g., Google) with specific search queries, presumably for black hat SEO. We only label a page "black hat SEO," if the search terms contains a domain name or a brand name.

Manually investigating HTTP archive files, we verified if any of the 1,339 pages labeled earlier based on the visual appearance as "potentially impersonating" are truly impersonating. This leaves us with 132 manually-tagged "impersonating" pages, which we

Features	Comment
Redir chain features	
Length of redirection chain	[4, 21, 27, 55, 57]
Length of registered domain redirection chain	[31, 55]
IP instead of domain at current hop in chain	[23, 27, 64]
Top Level Domains seen in redrection chain	[27]
Type of redirections (e.g., JavaScript, meta, HTTP)	[57, 61]
Number of IP addresses seen in redirection chain	
Domain features	
Cur/Sum/Avg/Max domain length	[23, 31, 55, 57, 64]
Cur/Sum/Avg/Max number of hyphens	[23]
Cur/Sum/Avg/Max number of dots	[23, 31, 57, 64]
URL features	
Cur/Sum/Avg/Max URL length	[21, 23, 31, 57, 61]
Cur/Sum/Avg/Max number of URL paramteres	[23, 27]
Cur/Sum/Avg/Max length of URL parameters	[23]
Cur/Sum/Avg/Max length of URL path	[23, 57]
Cur/Sum/Avg/Max number of URL path sub directories	[23]
Cur/Sum/Avg/Max length of URL filename	[23]
Cur/Sum/Avg/Max content size (except last hop)	[4, 27, 55]

Table 4: Features used for predicting malicious redirections described in Section 5.1. Cur means the value at a given redirection hop. In the comment column, we list references to papers that have used similar features often for different purposes.

then extrapolate to 1,556 pages by matching each landing URL's perceptual hash and domain.

5.1 Proactive Classification of Malicious Pages

We piggyback on the labeling effort described above to develop a prototype classifier that can identify whether a user is going to land on a malicious page. We use features purely based on the redirection chain and the URLs visited before loading the final destination page.

While researchers experimented with some variant of the features we use [4, 21, 23, 27, 28, 31, 55, 57, 61, 64], they either used features heavily relying on the page loaded or chose a graph-based approach building on the entire redirection chain graph to calculate their features. Our approach is different, as we only rely on the single redirection chain being traversed to predict if a user will land on a malicious page, and do not use a pre-computed malicious graph topology (which might change over time). Furthermore, previous work usually concentrated on one type of malice (e.g., phishing, drive-by-download), while our approach is independent of the kind of malice perpetrated.

Features. Our features include the number of URLs, IPs, and domains visited during redirections and the method of redirection (e.g., JavaScript, meta headers, and HTTP redirection codes). Our domain name features include the length of the domain name, the number of subdomains and the number of hyphens used in the domain name. URL-based features include the length of the URL, the number of URL parameters, the length of the parameters, the length of the directories, the number of sub-directories, the length of the filename, and the amount of content downloaded from the URL. We compute the previously described features for the last four hops of the redirection chain. We also derive the sum, mean, and maximum of these features across the entire relevant redirection chain. We detail the full list of 181 features used in Table 4.

Training a random forest classifier. Using these features, we train a random forest classifier. We train the classifier on our 101,926 semi-manually labeled pages. We used the random forest classifier of the Scikit-learn Python library [10] with a maximum depth of 32, maximum features of 40, minimum sample split of eight and 300 estimators.

	Copyright	Pharmacy	Typosquatting	URL Shortening	All
Error	6,817 (7.51%)	8,057 (10.1%)	41,734 (15.5%)	9,773 (18.2%)	66,381 (13.5%)
Benign	50,594 (55.7%)	45,003 (56.6%)	182,319 (68.0%)	31,223 (58.3%)	309,139 (62.8%)
Illicit	22,928 (25.2%)	25,595 (32.2%)	35,975 (13.4%)	5 (0.01%)	84,503 (17.1%)
Suspicious	8,089 (8.91%)	50 (0.06%)	3,668 (1.37%)	5,278 (9.86%)	17,085 (3.47%)
Malicious	2,334 (2.57%)	737 (0.93%)	4,345 (1.62%)	3,616 (6.76%)	11,032 (2.24%)
Multiple Tags	0 (0.0%)	0 (0.0%)	0 (0.0%)	3,612 (6.75%)	3,612 (0.73%)
All	90,762	79,442	268,041	53,507	491,752

Table 5: Label categories per traffic source.

	Android	Desktop	Google Bot	No Proxy	Real Phone	Referrer
Error	10,580 (11.5%)	10,750 (11.3%)	17,690 (19.8%)	7,579 (8.01%)	6,468 (22.1%)	13,314 (14.3%)
Benign	56,153 (61.2%)	61,033 (64.4%)	60,236 (67.6%)	60,290 (63.7%)	16,517 (56.4%)	54,910 (59.3%)
Illicit	17,566 (19.1%)	15,859 (16.7%)	9,752 (10.9%)	19,249 (20.3%)	4,679 (15.9%)	17,398 (18.8%)
Suspicious	3,610 (3.94%)	3,970 (4.19%)	741 (0.83%)	4,098 (4.33%)	941 (3.22%)	3,725 (4.03%)
Malicious	3,216 (3.51%)	2,152 (2.27%)	372 (0.42%)	2,497 (2.64%)	525 (1.79%)	2,270 (2.45%)
Multiple Tags	529 (0.58%)	930 (0.98%)	215 (0.24%)	940 (0.99%)	124 (0.42%)	874 (0.94%)
All	91,654	94,694	89,006	94,653	29,254	92,491

Table 6: Label categories per crawl profile.

6 RESULTS

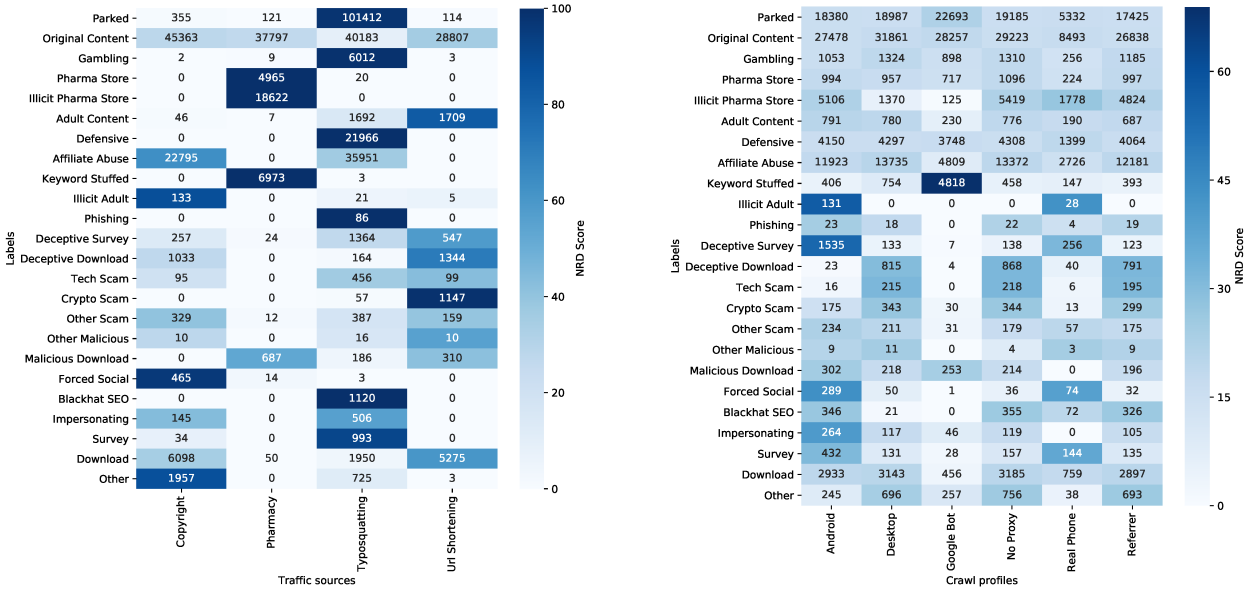
We next use our labels to describe the kinds of pages ODIN finds. Then, we discuss TDS overlap based on the redirection chains we observe. We also elaborate on abuse in these TDSs, and how blacklists perform. Finally, we evaluate our proactive classifier's performance.

6.1 Label Analysis

We start our analysis by discussing the types of content users are exposed to in the studied TDSs based on the labels described in Section 4. Tables 5 and 6 summarize the number of pages found per label class. After removing errors, we find that 26.5% of all collected pages are malicious (2.6%), suspicious (4.0%) or illicit (20.0%).

Phone versus desktop users. Figures 2a and 2b present the page count, and the associated Normalized Relative Descriptive (NRD) score for each destination page label, when sliced by traffic sources, and by crawl profile. We calculate the NRD score by first normalizing the number of occurrences for each slice separately, and then normalizing again for each label separately.

Figure 2b shows that phone users are, compared to desktop users, more often targeted by survey campaigns (e.g., promising prizes in exchange for filling out multiple questionnaires and downloading an app), by forced social media actions and impersonating pages, and, by illicit adult sites. Conversely, certain kinds of malicious contents, such as technical support scam pages and deceptive download pages, are more often shown to desktop users. One possible explanation for the absence of technical-support scams for phone users is that increasingly US adults no longer have landlines and solely rely on mobile phones for communication [46]. Making their smartphone unusable (via a barrage of pop-ups and alerts) would therefore prevent these users from being able to call the scammers and request their assistance. Because the mobile and desktop experiments were conducted at the same time and with the same infrastructure, this is the first time – to the best of our knowledge – that a study can conclusively state that mobile users are targeted by different malicious ads, compared to desktop users.



(a) Labels organized by traffic source

(b) Labels organized by crawl profile

Figure 2: Label counts and NRD score heatmap. The NRD score shows which labels are most characteristic of traffic sources (a) or crawl profiles (b).

Common malicious destination pages across traffic sources. Typosquatters appear to expose users to the same malicious content as illicit movie streaming sites and ad-based URL shortening services. Half of our malicious labels are present *in all three of these datasets*. We often observe the same technical support scams, deceptive survey and deceptive download pages. In Section 6.2, we dig deeper in whether these similar malicious landing pages are part of the same campaigns. Contrastingly, the pharmaceutical ecosystem appears to be largely non-overlapping with these other activities.

Malice in our datasets. Figure 2a shows that pharmaceutical queries present substantially different behavior compared to the other three traffic sources. We rarely observe malicious landing pages in this dataset and, as expected, we find mostly illicit pharmacies and keyword stuffed pages. Surprisingly, ODIN downloads a large number of malicious files while visiting pharmaceutical-related URLs, which has not been reported by previous research. Figure 2b shows clear differences depending on the type of user connecting: phone users (real or emulated) show different patterns than desktop (with or without proxy) users, while crawlers (GoogleBot) land on completely different pages.

Next, Table 5 shows (ad-based) URL shorteners present the highest rate of malicious URLs. These services frequently advertise adult content, crypto scams and file downloads. Among these destination pages, crypto scam advertisements were mostly unique to URL shortening pages. We found that 72% of all unique downloaded files are malicious, according to Virus Total. This number is 97.5% in the case of files downloaded from URL shortening services.

Confirming previous findings [1, 55], typosquatting domains lead us most of the time to parked pages. However, typosquatters also often engage in affiliate abuse, and in a wide variety of malicious activity. Most common malicious or suspicious content includes download pages, deceptive surveys, forced Google searches, impersonating pages, technical support scams and other scams. Certain malicious pages are specific to typosquatting pages, including forced Google searches, surveys (not deceptive), and financial phishing pages. We discovered 11 typosquatting domains hosting phishing content targeting customers of several large financial institutions. Additionally, we found one curious case of a forced Google search as part of a campaign launched by somebody attempting to disparage a corporation's public image with keywords such as "rip off," "stock," and "report."

Copyright infringing sites most commonly attempt to monetize user visits by deceiving users into downloading unwanted files. Moreover, movie streaming sites automatically force users to post on social media sites to promote their illicit activities.

Cloaking and bot detection. When ODIN poses as a Googlebot, it observes only few instances of malicious, suspicious or illicit content. This provides us with a baseline of how TDSs behave when visited by an automated crawler. We observe that automated crawlers are explicitly blocked 5% more often than other users and covertly blocked (by sending users to parked or other error pages) at least 8% more frequently.

We find no evidence of proxy detection based cloaking. While not using proxies resulted in a lower error rate, this is due to errors caused by the proxies. Similarly, it appears that cybercriminals do not attempt to detect phone emulation. The only difference between our emulated and real phone experiment is due to a measurement quirk: the phone infrastructure was not working on the dates when the crypto scam and the impersonation campaigns took place.

We confirm results by previous work [24, 25], that illicit pharmacies use the HTTP referrer header to cloak their illicit activity. Conversely, setting the referrer header seems to have the opposite effect in other TDSs, in that it slightly decreasing the number of

Label	Multi IP	Single IP
Error	56,794	62,947
Benign	148,428	144,756
Illicit	10,835	9,937
Suspicious	1,672	1,373
Malicious	2,690	1,287
Multiple Tags	411	429

Table 7: Comparing label categories of using multiple versus one proxy.

malicious pages discovered. The only exception is black-hat SEO activity, which almost always requires a referrer header field.

IP-Cloaking experiment. In Table 7, we present the results of the IP-Cloaking experiment, where we compare the difference between using 240 IP address versus only one IP address while running the same measurements. **We find that using multiple IP addresses leads us to find more than twice as many malicious pages.** We also experience fewer errors, and find more illicit and suspicious pages with multiple IP addresses. When miscreants show us a benign or error page instead of a malicious one, we face cloaking 86% of the time and are explicitly blocked only 14% of the time.

We also find that typosquatting domains are more likely to block our crawler if we use only one IP address, compared to URLs in the copyright, pharmaceutical, and URL shortening datasets. Moreover, if a malicious actor does not bother to conceal their activity from crawlers, they also do not bother performing IP-based blocking. Last, our phone crawler was proportionally less frequently blocked than the desktop crawlers.

6.2 TDS Redirection Analysis

We next discuss how the different traffic sources we selected share traffic brokers, subsequently sending users to similar malicious destinations. To that effect, we analyze TDS redirection chains.

User differentiation. Figure 3 compares how phone and desktop users might traverse entirely different parts of the TDS ecosystems. Nodes are domain names; edges signify a redirection between two domains. Blue domains were visited by our Android crawler, red domains were visited by our desktop (no-proxy) crawler; purple domains were visited by both crawlers. Red and blue clusters represent neighborhoods in the TDS ecosystem visited only by desktop users, or by phone users respectively. The zoomed example in the top left corner illustrate edges pointing to red (technical support scam) and blue (deceptive survey) domain clusters: these clusters denote landing pages. Purple clusters are source domains with only outward edges. Figure 3 shows the importance of studying user differentiation, as **users visiting the same URLs about half of the time end up on very different pages depending on whether they use a phone or a desktop for browsing.**

Ecosystem infrastructure overlap. Through our previous observations, we can conclude that different TDSs frequently serve the same malicious content to users. Next, we analyze whether these are the same entities that serve content to the different traffic sources.

In Figure 4a we present the number of unique malicious, suspicious or illicit unique traffic broker registered domains overlapping between different TDSs. Even though the illicit pharmacies overlap with other traffic sources, it is only a few domain names. We

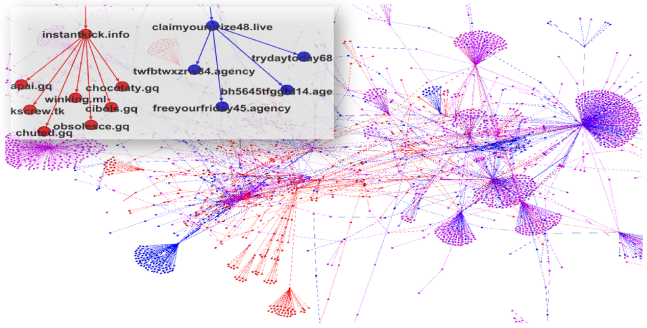


Figure 3: Malicious TDS redirection chain graph.

conclude that often the same entities are redirecting users to malicious landing pages as **we observe 19.2% to 44.1% traffic broker domains overlap between non-pharmacy TDSs**.

In Figure 4b we look at the overlap of unique landing registered domains across TDSs. We find that while the illicit pharmacy TDS overlaps only 3.7% to 4.1% of the time with the other datasets. Differently, typosquatting, copyright infringing and URL shortening TDSs overlap with each other 16.9% to 32.2% of the time. **These traffic sources overlap four to eight times more with each other than they do with illicit pharmacies.**

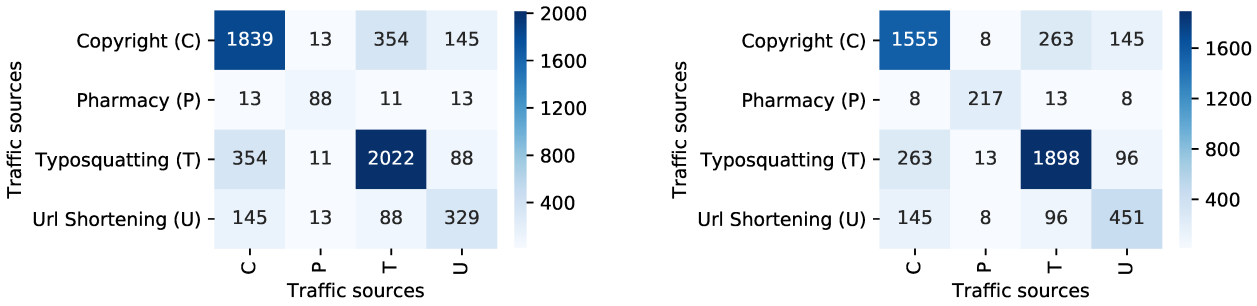
Redirection chain lengths. Like Li et al. [28], we observe that **on average users landing on a malicious, suspicious, or illicit page, are redirected through 71% to 122% longer chains compared to when landing on a benign page.**

Figure 5 illustrates the average redirection chain length for different crawl profiles and traffic sources. The pharmacy dataset shows a much shorter average redirection chain length compared to the other traffic sources, as usually they redirect users directly to the store from a compromised webpage. The Googlebot crawler experiences significantly fewer redirections than other agents. Conversely, phone crawlers are redirected more than the desktop crawlers.

Domain lifetime. As we sample a new set of URLs for every run of our experiments, we cannot directly compare the lifetime of the source domains. For the landing and intermediate domains, we can look at the number of days we see these domains as a rough proxy of relative usage lifetime in TDSs. Similar to related work [25, 27], we observe that intermediate domains (traffic brokers) are longer-lived than landing domains. Using a Mann-Whitney U-test, the difference is statistically significant for benign pages (5.62 days vs. 3.32 days, $p < 0.01$, effect size[2] 0.61), error pages (3.52 days vs. 2.84 days, $p \leq 0.01$, effect size 0.53), and, most interestingly, malicious pages (5.06 vs. 2.46 days, $p < 0.01$, effect size 0.70), where **intermediate domains are active more than twice as long as landing domains.** The difference is not statistically significant ($p > 0.1$) for the illicit (5.09 vs. 4.54 days) and suspicious (6.50 vs. 5.49 days) sources.

Top malicious domains. We list in Table 8 the top five traffic broker domains that redirect to the most malicious, suspicious or illicit landing pages. These **five domains are responsible for more than half of all the malicious redirections we encounter**. While these domains also redirect us to benign landing pages, this is generally not their primary business (only `odysseus-nua.com` could plausibly claim a majority of its traffic isn't malicious). They

[2]Effect size is calculated using the Common Language Effect size [37].



(a) Traffic broker domain overlap

(b) Landing domain overlap

Figure 4: Overlap of unique malicious, suspicious or illicit traffic broker and landing registered domain names between different traffic sources.

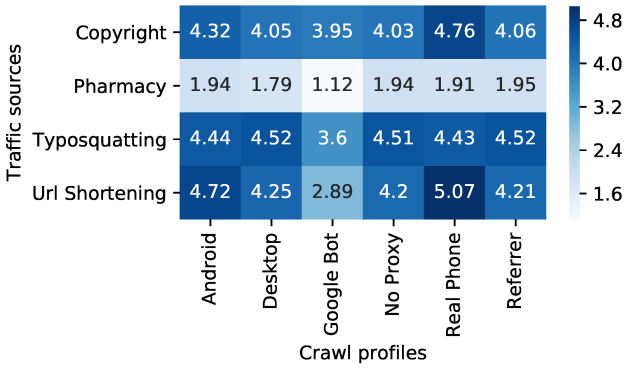


Figure 5: Average domain redirection chain length.

Domains	Out edges	In edges	Mal. out rate	Mal. in rate	#days
forwrdnow.com	2,595	2,595	0.6019	0.6019	9
7lyonline.com	1,811	1,811	0.6919	0.6919	6
136.243.255.89	2,015	2,015	0.5727	0.5727	35
odysseus-nua.com	4,612	4,621	0.2446	0.2441	35
gonextlinkch.com	912	913	0.9912	0.9901	3

Table 8: Top malicious traffic broker domains.

tend to be long lived: odysseus-nua.com and 136.243.255.89 are used undisturbed for more than two months, our full study period.

Domains	Out edges	In edges	Mal. out rate	Mal. in rate	#days
eleseems-insector.com	572	572	0.9930	0.9930	32
turtlehillvillas.com	596	596	0.9916	0.9916	35
gonextlinkch.com	912	913	0.9912	0.9901	3
7lyonline.com	1,811	1,811	0.6919	0.6919	6
addthis.com	659	794	0.7436	0.6171	34

Table 9: Most malicious traffic broker domain names

High malicious rate domains. Some traffic broker and landing domains seem to entirely serve malicious redirections as shown in Table 9. Even though they are an integral part of malicious ecosystems, it seems that many of them continue operating undisturbed. All the domains appearing for a few days only in our dataset are redirecting users to deceptive downloads. Certain domains, such as eleseems-insector.com, redirect users to technical support scam pages in the vast majority of the time; 7lyonline.com, redirects users to forced social media actions such as forced tweets. While addthis.com is a popular service, we observe that it is often used in redirection chains that automatically force social media interactions.

TLD usage in redirection chains. Analyzing how domains are utilized for redirections, we found that, on average, 2.7 times more unique domains and 2.5 more unique TLDs are leveraged for intermediate nodes in a malicious chain compared to a benign one. Additionally, it is 2.3 times likely for a malicious redirection chain to lead users to a destination page that uses a new gTLD domain[3].

6.3 Google Safe Browsing analysis

We next look into whether Google Safe Browsing (GSB) can help accurately label TDS destination pages as malicious. To do so, we compare GSB labels to our manually analyzed malicious label dataset collected between June 19, 2019 through July 4, 2019. We use the GSB Update API [12, 13] to determine if a domain or URL is deemed malicious by GSB. In a redirection chain, if *any* domain or URL is present in GSB on a given day, we label the page as malicious on that day.

Lack of coverage for malicious pages targeting phone users. While we find that mobile users are more frequently redirected to malicious landing pages than desktop users, it seems that **GSB does not include malicious landing pages shown to mobile users 76% of the time**. We conclude that not only are miscreants selectively catering malicious content towards mobile users but also that GSB currently suffers from poor coverage trying to protect mobile users when it comes to these malicious URLs.

Lack of coverage and delay in blacklisting. We find that GSB only labels 92 out of the 3,746 malicious pages on the same day we detect them. Even after 60 days, GSB finds 32% less malicious pages than we do. Additionally, a significant fraction of the GSB labels are false positives[4] due to the dynamic nature of TDSs, which redirect users to different destination pages at each visit, and the destination pages themselves changing over time. Finally, we confirm findings about the delay of blacklisting observed in other contexts [47], finding an average of seven-day delay for GSB to detect malicious pages.

[3]We consider a gTLD to be new if it was introduced after 2011.
[4]GSB false positives fall into the error, original content, parked and gambling labels, with a secondary manual analysis confirming these results.

6.4 Classifier Performance

In light of the poor blacklist coverage we observed, we evaluate our classifier predicting whether a redirection chain will lead to a malicious page holds promise. We find our classifier achieves an average 99.0% accuracy and 92.7% F_1 score (evaluated using 10-fold cross-validation) in labeling redirection chains as malicious or benign. Our classifier has a large area (0.95) under the precision-recall curve, with a particularly good trade-off at (0.89: recall, 0.9: precision).

Thus, our random forest classifier is able to identify the majority of malicious redirection chains with a decent precision before users would land on them. If a high precision is required (to accommodate base-rate issues and minimize false alarms), the classifier can still identify more than one third of the malicious pages proactively, as shown by the (0.42: recall, 0.99: precision) point.

Adversarial considerations. While the classifier performance appears satisfactory, we have to assume an adversary would spare no effort in trying to evade classification. Fortunately, features based on the redirection chain (e.g., chain length) could be economically costly for an adversary to evade. First, evading many of these features would restrict usage of TDSs, and thus, would make user traffic acquisition more costly. Second, without complex redirections, it becomes easier to automatically blacklist miscreants' domains. Similarly, our features related to the TLDs used would be a burden for an adversary to evade as these miscreants usually select TLDs, for at least some of the redirection hops, where registering domain names is cheap and convenient to decrease the cost of blacklisting.

URL and domain name-related features are moderately hard to evade as some of the URL features are inherent to the redirection hops the adversary does not necessarily control. For example, a traffic redirection service that is not particularly malicious, but that does not care about the safety of the users, might not change how it functions to aid its malicious customers. Some domain-related features might not be trivial for an adversary to evade as short domains are scarce and random domains are easier to detect. Miscreant would have to continuously generate longer but plausible sounding domain names. In short, our proposed classifier achieves reasonably good performance and should be reasonably robust to evasion.

7 DISCUSSION

Protecting users. Our machine learning model could be used, for example, as a browser extension to warn or block users before they are exposed to malice. Unlike previous work that relies on precalculated malicious graph topology [16, 27, 28, 51], our classifier only uses features observed at redirection time, making our model generalizable to any malicious activity that redirects users in a similar fashion as the TDSs we study. Such a mechanism would be a plausible complement to blacklisting, especially considering the inadequate coverage of existing blacklists.

Infrastructure take-down. Given the sharing of TDS infrastructure among different types of abusive content, our results suggest that correct prioritization of TDS take-downs by law enforcement has the potential to curb multiple kinds of abuse simultaneously.

Future of online crime research. Whether we consider academic research, security industry or law enforcement, going forward, when security practitioners attempt to discover malicious content online, they *must* deploy their crawlers from multiple vantage points, mitigate a variety of cloaking techniques and emulate different form factors (i.e., desktop and mobile). To inspire more research in this area, we open-sourced ODIN [18] and make the data collected available to researchers upon request.

8 CONCLUSION

This paper introduced ODIN, a measurement infrastructure to study for two months user differentiation, cloaking, and business integration in four different traffic sources that use TDSs. We found that these traffic sources often integrate their business model and send users to the same TDSs and malicious destination pages. Our analysis clearly demonstrates that phone and desktop users are redirected to different malicious landing pages. We also observed a significant amount of user-agent, referrer header field, and IP address-based cloaking. Altogether, when visiting URLs posing as six different types of crawlers, ODIN was able to unearth 81% more malicious landing pages compared to using only the most efficient crawler by itself. We also discovered that popular blacklists, including GSB, present limited coverage of malicious pages especially those targeting mobile users. Overall, our findings show that future studies measuring online crime *must* deploy their crawlers from multiple vantage points, address cloaking and emulate different types of users.

Acknowledgments We thank the reviewers for their helpful feedback. We are grateful to Mahmood Sharif for his advice on image clustering and to Orsolya Kovacs and Attila Kovacs for their help with data labeling. This work was supported by the National Science Foundation under grants CNS-1813974 and CMMI-1842020.

REFERENCES

[1] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of NDSS 2015*.
[2] Alexa. [n.d.]. Alexa's list of top one million popular sites. `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip`. Last accessed on April 18, 2020.
[3] Sumayah Alrwais, Kan Yuan, Eihal Alowaisheq, Zhou Li, and XiaoFeng Wang. 2014. Understanding the dark side of domain parking. In *USENIX Security 14*.
[4] Anirban Banerjee, Md Sazzadur Rahman, and Michalis Faloutsos. 2011. SUT: Quantifying and mitigating URL typosquatting. *Computer Networks* (2011).
[5] Elie Bursztein, Artem Malyshev, Tadek Pietraszek, and Kurt Thomas. 2016. Picasso: Lightweight device class fingerprinting for web clients. In *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*.
[6] Guanchen Chen, Matthew F Johnson, Pavan R Marupally, Naveen K Singireddy, Xin Yin, and Vamsi Paruchuri. 2009. Combating Typo-Squatting for Safer Browsing. In *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*.
[7] François Chollet. [n.d.]. DenseNet, Keras. `https://keras.io/applications/`. Last accessed on April 18, 2020.
[8] Jason W Clark and Damon McCoy. 2013. There are no free ipads: An analysis of survey scams as a business. In *Presented as part of the 6th {USENIX} Workshop on Large-Scale Exploits and Emergent Threats*.
[9] Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* (1964).
[10] Jérémie du Boisberranger, Joris Van den Bossche, Loïc Estève, Thomas J Fan, Alexandre Gramfort, Olivier Grisel, Yaroslav Halchenko, Nicolas Hug, Adrin Jalali, Guillaume Lemaitre, Jan Hendrik Metzen, Andreas Mueller, Vlad Niculae, Joel Nothman, Hanmin Qin, Bertrand Thirion, Tom Dupré la Tour, Gael Varoquaux, Nelle Varoquaux, and Roman Yurchak. [n.d.]. Scikit Random Forest Classifier. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`. Last accessed on April 18, 2020.
[11] B. Edelman. 2003. Large-Scale Registration of Domains with Typographical Errors. `http://cyber.law.harvard.edu/people/edelman/typo-domains/`.
[12] A. Filipovich. [n.d.]. Python client library for Google Safe Browsing API. `https://github.com/afilipovich/gglsbl`. Last accessed on April 18, 2020.
[13] Google, Inc. [n.d.]. Google Safe Browsing Update API. `https://developers.google.com/safe-browsing/v4/update-api`. Last accessed on April 18, 2020.

[14] Ben Hoyt. [n.d.]. Dhash Python library. https://pypi.org/project/dhash/. Last accessed on April 18, 2020.

[15] Damilola Ibosiola, Benjamin Steer, Alvaro Garcia-Recuero, Gianluca Stringhini, Steve Uhlig, and Gareth Tyson. 2018. Movie pirates of the caribbean: Exploring illegal streaming cyberlockers. In *Twelfth International AAAI Conference on Web and Social Media*.

[16] Luca Invernizzi, Stanislav Miskovic, Ruben Torres, Christopher Kruegel, Sabyasachi Saha, Giovanni Vigna, Sung-Ju Lee, and Marco Mellia. 2014. Nazca: Detecting Malware Distribution in Large-Scale Networks.. In *NDSS*.

[17] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean-Michel Picod, and Elie Bursztein. 2016. Cloak of visibility: Detecting when machines browse a different web. In *2016 IEEE S&P*.

[18] Janos Szurdi. 2021. ODIN public Github repository. https://github.com/jszurdi/ODIN/. Last accessed on February 7, 2021.

[19] Jordan Jueckstock, Shaown Sarker, Peter Snyder, Panagiotis Papadopoulos, Matteo Varvello, Benjamin Livshits, and Alexandros Kapravelos. 2019. The Blind Men and the Internet: Multi-Vantage Point Web Measurements. *arXiv preprint arXiv:1905.08767* (2019).

[20] Mohammad Taha Khan, Xiang Huo, Zhou Li, and Chris Kanich. 2015. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In *In 2015 IEEE Security and Privacy (SP)*.

[21] Amin Kharraz, William Robertson, and Engin Kirda. 2018. Surveylance: automatically detecting online survey scams. In *2018 IEEE S&P*.

[22] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. 2017. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of CCS 2017*.

[23] Anh Le, Athina Markopoulou, and Michalis Faloutsos. 2011. Phishdef: Url names say it all. In *2011 Proceedings IEEE INFOCOM*.

[24] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. 2011. Measuring and Analyzing Search-Redirection Attacks in the Illicit Online Prescription Drug Trade.. In *USENIX Security Symposium*.

[25] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. 2014. A nearly four-year longitudinal study of search-engine poisoning. In *In ACM CCS 2014*.

[26] K. Levchenko, N. Chachra, B. Enright, M. Felegyhazi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, A. Pitsillidis, N. Weaver, V. Paxson, G. Voelker, and S. Savage. 2011. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *Proceedings of IEEE Security and Privacy*.

[27] Zhou Li, Sumayah Alrwais, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2013. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In *2013 IEEE S&P*.

[28] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM CCS*.

[29] Alessandro Linari, Faye Mitchell, David Duce, and Stephen Morris. 2009. Typo-Squatting: The Curse of Popularity. In *Proceedings of the WebSci'09: Society On-Line*.

[30] L. Lu, R. Perdisci, and W. Lee. 2011. SURF: Detecting and Measuring Search Poisoning. In *Proceedings of ACM CCS 2011*.

[31] Samuel Marchal, Giovanni Armano, Tommi Gröndahl, Kalle Saari, Nidhi Singh, and N Asokan. 2017. Off-the-hook: An efficient and usable client-side phishing prevention application. *IEEE Trans. Comput.* (2017).

[32] D. McCoy, A. Pitsillidis, G. Jordan, N. Weaver, C. Kreibich, B. Krebs, G. Voelker, S. Savage, and K. Levchenko. 2012. PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs. In *Proceedings of USENIX Security 2012*.

[33] Xianghang Mi, Ying Liu, Xuan Feng, Xiaojing Liao, Baojun Liu, XiaoFeng Wang, Feng Qian, Zhou Li, Sumayah Alrwais, and Limin Sun. 2019. Resident Evil: Understanding Residential IP Proxy as a Dark Service. In *2019 IEEE S&P*.

[34] Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. 2017. Dial One for Scam: A Large-Scale Analysis of Technical Support Scams. In *NDSS*.

[35] Tyler Moore and Benjamin Edelman. 2010. Measuring the perpetrators and funders of typosquatting. In *Financial Cryptography and Data Security*.

[36] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2016. Towards measuring and mitigating social engineering software download attacks. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*.

[37] Robert G Newcombe. 2006. Confidence intervals for an effect size measure based on the Mann–Whitney statistic. Part 1: general issues and tail-area-based methods. *Statistics in medicine* (2006).

[38] Nick Nikiforakis, Federico Maggi, Gianluca Stringhini, M Zubair Rafique, Wouter Joosen, Christopher Kruegel, Frank Piessens, Giovanni Vigna, and Stefano Zanero. 2014. Stranger danger: exploring the ecosystem of ad-based url shortening services. In *WWW 014*.

[39] Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. 2019. PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques Against Browser Phishing Blacklists. In *PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists*.

[40] OpenDNS. [n.d.]. PhishTank. http://phishtank.com. Last accessed on April 18, 2020.

[41] Paul Pearce, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey Voelker. 2014. Characterizing large-scale click fraud in zeroaccess. In *In ACM CCS 2014*.

[42] Paolo Piredda, Davide Ariu, Battista Biggio, Igino Corona, Luca Piras, Giorgio Giacinto, and Fabio Roli. 2017. Deepsquatting: Learning-based typosquatting detection at deeper domain levels. In *Conference of the Italian Association for Artificial Intelligence*.

[43] PRGMR. [n.d.]. PRGMR VPS provider. https://prgmr.com/xen/. Last accessed on April 18, 2020.

[44] Niels Provos, Panayiotis Mavrommatis, Moheeb Rajab, and Fabian Monrose. 2008. All your iframes point to us. (2008).

[45] M Zubair Rafique, Tom Van Goethem, Wouter Joosen, Christophe Huygens, and Nick Nikiforakis. 2016. It's free for a reason: Exploring the ecosystem of free live streaming services. In *Proceedings of NDSS 2016*.

[46] Felix Richter. 2020. Landline Phones Are a Dying Breed. https://www.statista.com/chart/2072/landline-phones-in-the-united-states/.

[47] Mahmood Sharif, Jumpei Urakawa, Nicolas Christin, Ayumu Kubota, and Akira Yamada. 2018. Predicting impending exposure to malicious content from user behavior. In *Proceedings of the 2018 ACM CCS*.

[48] Jeffrey Spaulding, Shambhu Upadhyaya, and Aziz Mohaisen. 2016. The landscape of domain name typosquatting: Techniques and countermeasures. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*.

[49] Bharat Srinivasan, Athanasios Kountouras, Najmeh Miramirkhani, Monjur Alam, Nick Nikiforakis, Manos Antonakakis, and Mustaque Ahamad. 2018. Exposing search and advertisement abuse tactics and infrastructure of technical support scammers. In *WWW 2018*.

[50] Oleksii Starov, Yuchen Zhou, Xiao Zhang, Najmeh Miramirkhani, and Nick Nikiforakis. 2018. Betrayed by your dashboard: Discovering malicious campaigns via web analytics. In *In WWW 2018*.

[51] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *ACM CCS 2013*.

[52] SURBL maintainers. [n.d.]. SURBL: URI reputation data. http://www.surbl.org/lists. Last accessed on April 18, 2020.

[53] Janos Szurdi and Nicolas Christin. 2017. Email typosquatting. In *Proceedings of IMC 2017*.

[54] Janos Szurdi and Nicolas Christin. 2018. Domain registration policy strategies and the fight against online crime. *WEIS, June* (2018).

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

[56] Rashid Tahir, Ali Raza, Faizan Ahmad, Jehangir Kazi, Fareed Zaffar, Chris Kanich, and Matthew Caesar. 2018. It's all in the name: Why some urls are more vulnerable to typosquatting. In *IEEE INFOCOM 2018*.

[57] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. 2011. Design and evaluation of a real-time url spam filtering service. In *2011 IEEE S&P*.

[58] Ke Tian, Steve TK Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a haystack: tracking down elite phishing domains in the wild. In *IMC 2018*.

[59] Phani Vadrevu and Roberto Perdisci. 2019. What You See is NOT What You Get: Discovering and Tracking Social Engineering Attack Campaigns. In *In IMC 2019*.

[60] T. Vidas and N. Christin. 2014. Evading Android Runtime Analysis via Sandbox Detection. In *In ASIACCS'14*.

[61] Thomas Vissers, Wouter Joosen, and Nick Nikiforakis. 2015. Parking sensors: Analyzing and detecting parked domains. In *Proceedings of NDSS 2015*.

[62] D. Wang, S. Savage, and G. Voelker. 2011. Cloak and Dagger: Dynamics of Web Search Cloaking. In *Proceedings of ACM CCS 2011*.

[63] Yi-Min Wang, Doug Beck, Jeffrey Wang, Chad Verbowski, and Brad Daniels. 2006. Strider typo-patrol: discovery and analysis of systematic typo-squatting. In *Proc. 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*.

[64] Colin Whittaker, Brian Ryner, and Marria Nazif. 2010. Large-scale automatic classification of phishing pages. (2010).

[65] Jing Ya, Tingwen Liu, Quangang Li, Pin Lv, Jinqiao Shi, and Li Guo. 2018. Fast and Accurate Typosquatting Domains Evaluation with Siamese Networks. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*.

[66] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2014. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proceedings of IMC 2014*.

[67] Yuwei Zeng, Tianning Zang, Yongzheng Zhang, Xunxun Chen, and YiPeng Wang. 2019. A Comprehensive Measurement Study of Domain-Squatting Abuse. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*.