Parking Sensors: Analyzing and Detecting Parked Domains

Thomas Vissers*, Wouter Joosen* and Nick Nikiforakis†
* iMinds-DistriNet, KU Leuven,
{firstname}.{lastname}@cs.kuleuven.be
† Department of Computer Science, Stony Brook University,
nick@cs.stonybrook.edu

***Abstract*—A parked domain is an undeveloped domain which has no content other than automatically computed advertising banners and links, used to generate profit. Despite the apparent popularity of this practice, little is known about parked domains and domain parking services that assist domain owners in parking and monetizing unused domains.**

This paper presents and in-depth exploration of the ecosystem of domain parking services from a security point of view, focusing mostly on the consequences for everyday users who land on parked pages. By collecting data from over 8 million parked domains, we are able to map out the entities that constitute the ecosystem, thus allowing us to analyze the domain owners, parking services, and advertisement syndicators involved. We show that users who land on parked websites are exposed to malware, inappropriate content, and elaborate scams, such as fake antivirus warnings and costly remote "technicians". At the same time, we find a significant number of parked domains to be abusing popular names and trademarks through typosquatting and through domain names confusingly similar to authoritative ones.

Given the extent of observed abuse, we propose a set of features that are representative of parked pages and build a robust client-side classifier which achieves high accuracy with a negligible percentage of false positives.

I. Introduction

Up until twenty years ago, domain names were available for free on a first-come, first-serve basis. Network Solutions, the company that was contracted by the US Defense Information Systems Agency to operate the DNS registry, was imposing at the time a one domain for each person/company limitation. Because of the high demand, in 1995, Network Solutions switched to a paying model which has been preserved till today.

Apart from the rapid expansion of the web due to the fact that every person and company desired to have an online presence, the high demand for domain names was also due to people buying large amounts of domains in bulk and creating domain portfolios. These people, later called "domainers", had no interest in setting up companies and using their purchased domains. They had, instead, realized that domains would soon become a very valuable commodity. As such, they bought hundreds or thousands of domains under the premise that real companies would later pay them a large amount of money in exchange for a desirable domain from their extensive portfolios. Popular successful examples of this strategy include domains like wine.com and casino.com both sold for millions of dollars. Even with the advent of new Top Level Domains (TLDs), short and generic domains are still occasionally sold for exorbitant prices [32].

Initially, the owners of large domain portfolios did little more than wait until an interested buyer contacted them for one of their domains. Eventually, some people realized that instead of just idly waiting, domain portfolios could be put to better use. These people struck deals with other websites and included banners and "favorite links" for a flat monthly fee [22]. At the same time, however, a new type of advertising started appearing on the web, namely, Pay-Per-Click advertising. As the name implies, in this new scheme, a publisher would only get paid by the advertiser if a user would click on one of the ads.

Undeveloped domains were a natural fit for this type of ads. Services started appearing that made it easy for domain owners to incorporate ads on their domains without worrying about finding advertisers and setting up contracts. These services were called *domain parking* services, since domain owners would simply "park" their domains at these service providers, and then the rest would be done automatically. Domain parking was so successful that owners of large domain portfolios stopped worrying about selling their domains and instead started making profit simply because of the commission they got from the ads that were clicked [22].

Despite of the popularity of the domain parking phenomenon, domain parking services have received little attention from the security community. Up until recently, these services were only mentioned in papers studying cybersquatting, showing that domain parking is, by far, the most popular monetization strategy for cybersquatters [12], [20], [25], [27], [28], [30], [31]. In recent research, Alrwais et al. studied domain parking from the point of view of advertisers and domain owners [15]. They showed that the majority of domain parking companies employ shady practices, such as, hiding clicks from domain owners (thereby not sharing click commissions), conducting click fraud, and sending unrelated

NDSS '15, 8-11 February 2015, San Diego, CA, USA

http://dx.doi.org/10.14722/ndss.2015.23053

traffic to advertisers who pay for traffic with very specific demographics.

In this paper, we study domain parking services mainly from the point of view of everyday web users and thus orthogonally to the work of Alrwais et al. [15]. We identify 15 popular parking services and retrieve a corpus of more than 8 million parked domains. Through a series of automatic and manual experiments, we identify the presence of a wide range of fraud and abuse, targeting users as well as companies. Among others, we show that typical users landing on parked pages are exposed to inappropriate ads, malware, elaborate scams involving "technicians" getting access to a user's machine and scripts that detect and bypass advertising blockers. We also challenge the stance of the most popular ad syndicator who provides the domain parking ecosystem with the necessary advertising infrastructure, while at the same time telling its users that parked pages are *spam* and are thus hidden from that syndicator's search engine.

Given the extent of the discovered abuse, we design and build a robust classifier for detecting parked domains which does not rely on hard coded signatures but on distinguishing features of parked domains with a true-positive rate of 97.9% and a false-positive rate of only 0.5%. This classifier can be straightforwardly incorporated in a browser through its extension system, and alert users whenever they land on a parked domain.

Our main contributions are:

- We perform the first thorough study of domain parking services, mapping out the entire ecosystem while focusing on the abuse affecting everyday web users.
- We show that parked pages expose the user to a series of dangers including malware, scams and inappropriate content
- We verify the presence of an unacceptably large number of typosquatting domains parked with popular domain parking services, and demonstrate the lack of control when it comes to parking an obviously typosquatting domain
- We propose a performant classifier for detecting parked pages, utilizing robust features that are telling of a website's nature

The remainder of the paper is organized as follows. Section II describes the ecosystem of the domain parking industry. In Section III, we analyze the domains parked with these services and we study several of their abusive practices. Afterwards, Section IV details how we designed a classifier to detect parked pages. Thereafter, the observations and results of the paper are further discussed in Section V, followed by an overview of the related work in Section VI. Lastly, the paper concludes in Section VII.

II. Domain Parking Ecosystem

In this section, we first describe the modus operandi of domain parking services and the various entities involved in the domain parking ecosystem. We then report on the discovery of 8 million parked domains, hosted with 15 different parking services, in an effort to identify how domain parking is used today and the extent of abuse in terms of trademark infringements, typosquatting and malicious redirections.

A. What is Domain Parking?

The ecosystem of domain parking consists of four different parties: domain owners, parking services, advertisement syndicators, and advertisers. Domain owners (or domainers) usually own a large portfolio of undeveloped domain names, which they wish to monetize. As long as the monetization strategy returns more money than the cost of owning and managing a domain portfolio, this strategy probably is financially attractive.

Parking services provide hosting and generated content for domain owners who wish to monetize their domain names. As the name suggests, the domain owners "park" their domain names with domain parking services who then manage these domains and, in return, give the domain owners a share of their profits. Parking services typically collaborate with advertisement syndicators to serve purportedly relevant pay-per-click (PPC) ads from one of their advertising clients.

The operation model of domain parking is depicted in Figure 1. When a domain owner selects a service to park his domains, he configures the DNS settings of the domains to use the name servers of that parking service (1). When a web user later visits that domain, a parking page is displayed with content that is dynamically generated by the parking service. In addition, the parking service includes JavaScript code from an ad syndicator on the parked page. The syndicator's code will attempt to fetch and display ads from a plethora of advertisers, based on relevant keywords derived from the domain name (2). For example, if a user visits the parked domain cheapgas.com the parked page eventually displays ads that, in principle, are relevant to "cheap gas." If a user clicks on one of these ads, he will be sent to the respective advertiser's website, through the syndicator's tracking and redirection mechanisms (3). The advertiser will pay the syndicator for the visitor, who, in turn, will pay the parking service for the delivered click. Finally, the domain owner is given his share for supplying the domain on which the click was generated (4).

B. How do users end up on parked domains?

A detail that is missing from the above description is the process through which users end up on parked domains. Since parked services take, by definition, advantage of undeveloped domain names, it is unlikely that a user landed on a parked page by finding and following a link in a search engine, or on a trustworthy domain. As such, users end up on parked domains through alternative means.

While today's search engines are tightly coupled in a user's browser (either in a dedicated text field next to the address bar, or piggybacking on the address bar itself) that was not always the case. In the past, a user would have to either know about a specific domain name, or explicitly visit the website of a search engine and search for relevant content. During that time, browsing the web involved significantly more typing of domain names in a browser's URL bar than it does today.

For known sites, users would typically need to memorize the domain name of a website and manually type it, in full,

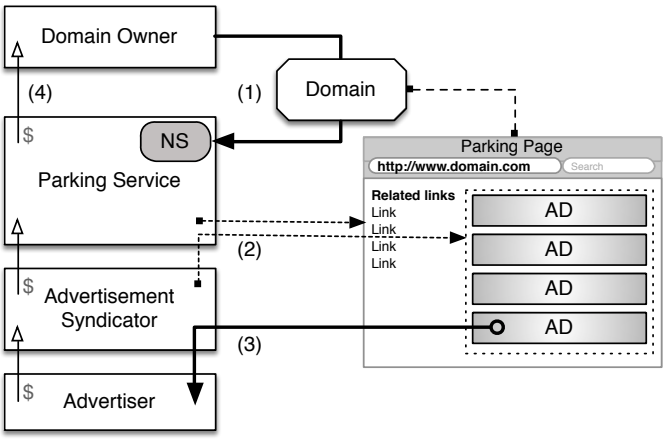

Fig. 1. Overview of the operation model of domain parking. (1) Domain owner hosts domain with parking service. (2) Parking page is dynamically generated. (3) Visitor clicks on an ad and is sent to the advertiser's website. (4) Payment flows from the advertiser to every entity up the chain.

in their browsers' address bars. Opportunists noticed that as users type in long URLs, there is a possibility of a typing mistake that will go by undetected, e.g., typing and requesting wikipdeia.org instead of wikipedia.org, and started registering these typo-including domain names. This practice was called *typosquatting* and, as prior research has shown, the preferred monetization strategy of typosquatters is parked domains [12], [25], [30], [31].

Next to typographical mistakes, users would also try to "guess" the domain names of websites relevant to their needs. Thus, a user who is interested in finding "cheap gas" could either visit a search engine and search for that phrase or, alternatively, concatenate the two words, append the most popular TLD and attempt to visit the cheapgas.com website. It is also worth noting that, at the time, browsers were trying to "help" users by appending TLDs before giving up on a domain. That is, if the user typed in cheapgas and the domain did not resolve, the browser would automatically append the .com TLD and try again [5].

In both scenarios, a user lands on a parked page by attempting to type the address of a website. The traffic resulting from this action was appropriately named "type-in" traffic and is, historically, the reason why domain parking services exist. Next to this type of traffic, researchers also recently noticed that domains belonging to malware distribution sites and C&C servers, when taken down, are usually repurposed as parked pages [23]. Thus, in addition to type-in traffic, users can land on parked pages by visiting infected sites that forward these users to a parked domain, instead of forwarding them to a browser-exploiting page

C. Gathering Parked Domains

In order to establish a representative analysis of the domain parking industry, four different sources were used to assemble a list of domain parking services: a survey amongst domainers [13], the top results of Alexa.com and Google.com when querying for *domain parking*, and a thread from a popular domaining forum [6]. Next, we select all services that are listed in more than one source, resulting in a collection of 15 services on which we focus for the rest of this paper.

service	Setting	Address
SedoParking	NS	sedoparking.com
InternetTraffic*	NS	internettraffic.com
CashParking*	NS	cashparking.com
Fabulous*	NS	fabulous.com
DomainSponsor	NS	dsredirection.com
Above[1]	NS	above.com
ParkingCrew	NS	parkingcrew.net
	A	62.116.181.25
	CNAME	parkingcrew.net
Skenzo*	NS	ztomy.com
NameDrive	NS	fastpark.net
Voodoo*	NS	voodoo.com
RookMedia	NS	rookdns.com
Bodis	NS	bodis.com
	CNAME	parking.bodis.com
DomainApps	NS	domainapps.com
TrafficZ*	NS	trafficz.com
	A	198.202.142.246
	A	198.202.143.246
TheParkingPlace	NS	pql.net
	Redirect	putoppose.net/d/domain

TABLE I. SUMMARY OF THE OBSERVED PARKING SERVICES TOGETHER WITH THEIR REQUIRED DOMAIN CONFIGURATION. ENTRIES MARKED WITH AN ASTERISK WERE FOUND THROUGH EXTERNAL ANALYSIS.

The next step is to find domains that are parked with the aforementioned domain parking services. Since it is unlikely that the services will voluntarily provide us with the domain portfolios that they manage, we must identify parked domains in an alternative way. The strategy for this task is a rather straightforward one: given any domain name, if its configuration matches the configuration of any of the 15 studied services, then that domain is indeed operated by a domain parking service.

To that extent, we registered ourselves as domain owners with each service and took note of the configuration instructions. The parking configuration of a domain usually involves the setting of DNS records to point to the parking service's servers. For the vast majority of cases, this involved pointing the appropriate NS record of your domain to their name servers. In some cases, services did not accept us as their clients. This was usually due to our domain portfolio not being "large enough". For these services, we manually identified existing parked domains and extracted the appropriate information from their DNS records. Table I summarizes the configurations we found across all services.

We started the process of collecting parked domains by searching the records of the DNS Census dataset [2], which contains about 2.5 billion DNS records gathered in 2012 and 2013. We extracted all domains that matched the configurations of Table I and subsequently queried its domain's DNS records to confirm whether they were still parked with that particular parking service. This resulted in a total of 8,064,914 actively

[1]Strictly speaking, **Above** is a *parking manager*. They determine an optimal parking strategy for each domain.

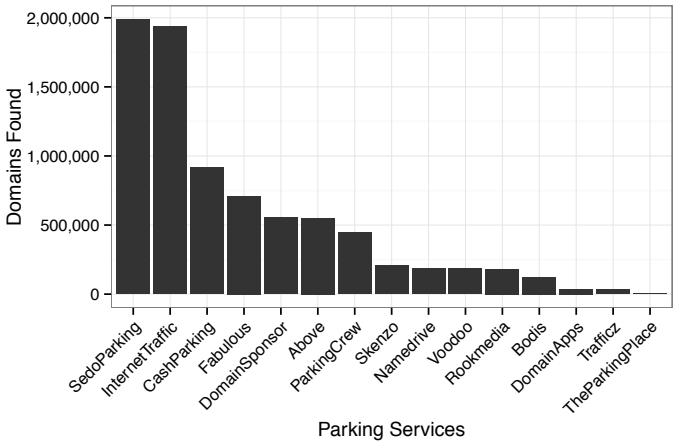

Fig. 2. Number of found domains for the 15 observed parking services.

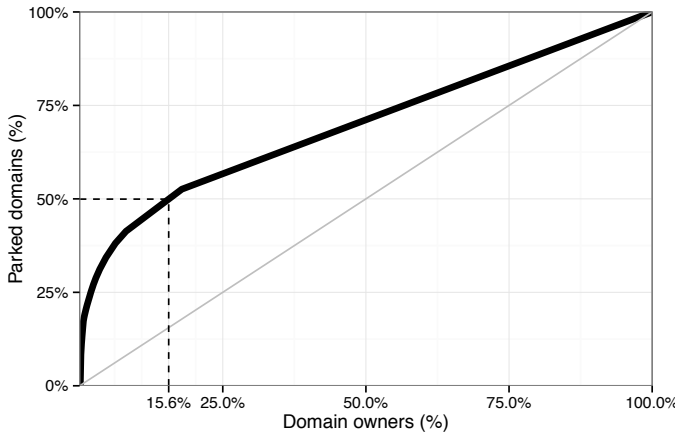

Fig. 3. The percentage of distinct owners and their cumulative share in possession of parked domains.

parked domains from the 15 observed parking services. Since the DNS Census is outdated (and likely incomplete), this means that at the time of this writing, there exist *at least* 8 million domains whose sole purpose is to serve ads when they are visited.

Figure 2 shows the distribution of the gathered domains with each service. One can quickly notice that even though many parking services exist, only a handful of them are responsible for managing the majority of parked domains. That is, 60% of the discovered domains are parked with the three most popular services: *Sedo Parking*, *Internet Traffic* and *Cash Parking*.

III. Analysis of Parked Domains

In this section we study several characteristics and practices of parked domains. We start by inspecting the 8 million gathered domain names and map out their typosquatting abuse. Next, we randomly sample several thousand parked domains which we crawl automatically using PhantomJS [7], an instrumented browser, while saving the HTML of every loaded frame, logging all HTTP requests and taking a screenshot. Furthermore, we collected WHOIS data for these domains. This data is then used to map out advertising networks, domain owners, trademark abuse, malicious redirects and ad-blocker detection mechanisms.

A. Parked Domain Owners

To get an understanding of parked domain ownership, we request WHOIS data for 3000 randomly-selected parked domains. We parse the WHOIS records using Ruby Whois [18] and filter out the records that are anonymized or unparseable. From the remaining 1,582 domains, we extract the registrant, administrator, and technical contact details and group together the domains that list the same name, email address and organization. As a final step, we manually merge clusters of domains for which we are certain belong to the same individual or organization. In total, we find 910 distinct domain owners to which the 1,582 domains belong. Figure 3, shows that a small number of them is responsible for the majority of parked domains. For instance, 50% of domains is owned by 15.6% (142) of owners. This means that next to owners who posses a couple of domains and use domain parking services, there are individuals with very large collections of domains, all of which are registered to simply serve ads.

B. Advertisement Syndicators

Advertising is the lifeblood of domain parking services. In order to generate revenue from parked domains, parking services usually serve pay-per-click (PPC) advertisements to visitors. Every time that a user clicks on an advertisement situated on a parked page, both the domain owner, as well as the domain parking service itself are paid a small commission.

While a domain parking service could, in principle, directly accept ads from people who wish to advertise their products and services and then display them on their own parked pages, it is easier and more scalable for them to use existing advertising infrastructure of third-party ad syndicators.

The integration of these syndicators involves little more than the inclusion of remote JavaScript libraries from the servers of the syndicator. These libraries are responsible for assessing the content of the page, fetch and display ads from other third-party servers, deliver the user who clicks on an ad to the advertiser who paid for that ad and register that action so that the publishing site (in this case the domain parking service) can receive the appropriate commission.

In search for these syndicators, we sampled 3,000 parked domains, crawled them, and inspected their source code for ad-related remote JavaScript inclusions. In total, we found only four advertisement syndicators that provided ad-related JavaScript code, *Adsense, Doubleclick, Media.net* and *Chango*, as shown in Figure 4. Their aggregate presence reaches 91%±1% of the parked websites. The other 9%±1% were either redirects (7%±0.9%) (described later in Section III-F) or had no identifiable advertising code (2%±0.05%). Since domain parking services depend on advertising, we assume that the latter was due to some temporal server-side misconfiguration.

Doubleclick and Adsense, both Google products, were present in 90%±1.1% and 88%±1.2% of parked websites respectively. While, given the ubiquitous nature of Google

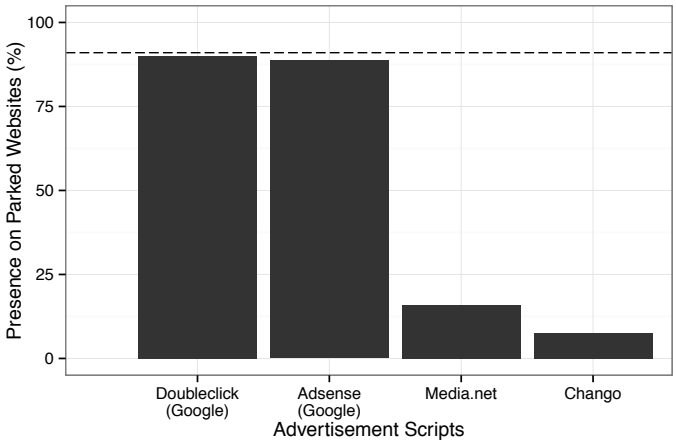

Fig. 4. Different third-party JavaScript advertisement syndicators and their presence on parked domains. The horizontal dashed line represents the amount of websites that included at least one third-party ad syndicator

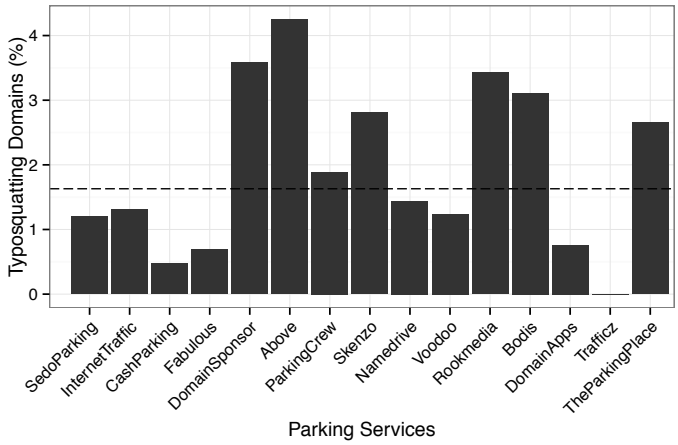

Fig. 5. Percentage of typosquatting domains present per service. The dashed line represents the relative amount of typosquatting domains present in all 8 million domains (1.63%).

in the modern web, this comes as no surprise, there is an interesting back story that is worth mentioning.

Google used to manage their own domain parking service, that was operating in a fashion similar to the 15 services studied in this paper. In 2012, Google ceased their own hosted parking service [4], possibly due to typosquatting lawsuits such as *Vulcan Golf, LLC v. Google, Inc.* [10]. In this lawsuit, the plaintiffs claimed the Google's parking program as a *"massive scheme to generate billions of advertising dollars through the parking of domain names that are the same as or substantially and confusingly similar to the plaintiffs' distinctive trade names or trademarks."* In addition to stopping their own domain parking products, Google is, at the time of this writing, stating on their *fighting spam* page that: *"Parked domains are placeholder sites with little unique content, so Google doesn't typically include them in search results."* [19]

One can easily spot the contradiction of these statements when compared to the pervasiveness of Google as an ad-syndicator for domain parking services. We thus find it hard to reconcile Google's decision to keep parked pages away from its search results, while still profiting by being the most popular advertising syndicator in known domain parking services.

C. Typosquatting abuse

Prior research investigating the phenomenon of typosquatting has established that the preferred monetization approach of domain squatters is the use of parking services [25], [31]. Yet typosquatting sites are illegal under the Anti-Cybersquatting Consumer Protection Act (ACPA), which prohibits registration and use of domain names that are identical or confusingly similar to a trademark. As such, a domain parking service should, in principle, always be wary of people trying to park typosquatting domains.

In order to measure the prevalence of typosquatting in the investigated parking services, we perform several "reverse-typo" transformations on the parked domains and attempt to discover whether these transformation result in an authoritative domain. We define an authoritative domain, as a domain that is ranked higher on the Alexa global list than its potential typosquatted version. Consider, for instance, the currently parked domain vacebook.com. This website does not belong in Alexa's top 1 million sites thus it automatically receives the lowest possible rank. One of our reverse-typo transformations will produce facebook.com, the domain belonging to the popular social network and ranked as the second most popular website of the Internet. Since this domain is ranked higher than the parked vacebook.com, we automatically mark the latter as a typosquatting domain.

Our transformation models were mainly inspired by Wang et al. [31], which introduced missing-dot, character-omission, character-permutation, adjacent character-substitution and character-insertion typo models. For some models, we had to develop the reverse operation, because of their non-bidirectional properties, such as for the missing-dot and character-insertion typo models. For the missing-dot model, we used heuristics to identify the place where a dot has to be added (e.g. add a dot in wwwfacebook.com *after* the "www" sequence of characters). Additionally, for the character-insertion model we defined a subset, namely, the character-duplication model, which only takes into account the accidental repetition of a character in the domain name. In order to reverse the character-duplication model, we had to identify a sequence of repeated characters and deduplicate them. Furthermore, we chose to discard the "reverse-typo" character-omission model due to high chances of false positives. Finally, in addition to the aforementioned models, we also introduced a TLD-substitution model which changes the top-level domain (TLD) with another popular one, for example from `com` to `us`.

We applied the reverse-typo transformations to the 8 million parked domains and found a total of 131,673 typosquatting domains (1.63%). The distribution, however, of these domains across parking services is anything but uniform. Figure 5 shows the percentage of typosquatting abuse for each studied service. *TrafficZ* is the only service with no typosquatting domains in its observed portfolio. The service with the biggest relative presence is *Above* with over 4%, followed by *DomainSponsor* and *RookMedia*. A positive finding is that the most popular parking services, as listed in Fig. 2 are also the services with less than average typosquatting abuse.

D. Parking a Typosquatting Domain

In the previous section we established that unfortunately, typosquatting domains are by no means rare on the investigated domain parking services. In this section, we approach the problem from the opposite side. Instead of trying to identify parked typosquatting domains, we want to quantify the "hurdles" that domain squatters have to go through in order to successfully park their domain names, i.e., we try to assess the parking services' selectiveness in terms of excluding obviously abusive domains.

With explicit permission of the authoritative company, we register a typosquatting domain of a high profile and well known website, *stackoverflow.com*. At the time of writing (July 2014), this website is the 53rd most popular website in the world, according to Alexa. More specifically, we registered a character-permutation typo domain, namely *stcakoverflow.com* which we then attempted to park with each service on which we managed to register an account. During this process, we discovered that the domain name was owned by a typosquatter in the past: it was registered from August 2012 until it expired a year later. During that time, the domain was parked with four of the services under analysis. Since these services had already record of this domain name belonging to a different user, they required an extra "verification" step. The verification simply involved emailing the support and optionally sending a screenshot of our registrar's account details. Even with the extra attention of a verification step involving a human operator verifying our screenshots, not a single service denied the submission of this abusive domain. Every service hosted the typosquatting domain for at least a week, until we transferred it to the next one. According to the services' statistics, the parking page was receiving visitors daily.

While our experience with the process of parking a single typosquatting domain does not necessarily generalize to the parking of hundreds of abusive domains, they are in line with the findings of the aforementioned domain-squatting studies, namely, that for domain parking services, taking advantage of squatting domains appears to be part of their everyday operations.

E. Trademark abuse

While typosquatting domains belong to the trademark abuse category, not every domain that abuses trademarks is necessarily a typosquatting domain. Consider, for example, the currently parked domain facebookonline.com. This domain clearly abuses on Facebook's trademark, yet it would never be automatically generated by the aforementioned typosquatting models. Historically, such abusive domains have been called "cousin domains" [21] and have been often associated with phishing since it is unlikely that an everyday user of the web will think that facebookonline.com is *not* associated with facebook.com.

As done in the earlier sections, we again make use of sampling to cope with the large amount of data. That is, we randomly selected 500 parked domains, manually extracted each domain's distinct keywords (e.g. the words `facebook` and `online` for the aforementioned example) and queried a popular search engine for these keywords. If the results revealed the presence of an obviously similarly named website or organization, we mark the domain as abusive.

Out of the 500 investigated parked domains, 79 (16%±3.2%) domains were clearly abusing trademarks of existing companies and websites.

More specifically, in 44 (9%±2.5%) of the cases, the domains were found to be typosquatting, while in the remaining 35 (7%±2.2%), the domains obviously contained trademarks. The percentage of typosquatting domains is significantly larger than the one that we automatically calculated in Section III-C suggesting that our typosquatting models provided a very conservative estimate of the magnitude of the problem. The reason for this discrepancy is that the typosquatting models that we used, do not cover all typographical errors abused in the wild. For instance, in our manual sample we noticed the abuse of homonyms, e.g., theheneryford.org abusing the authoritative thehenryford.org as well as character-omission typos which were excluded from our earlier analysis.

To gain insights on what type of advertisements end up on trademark-abusing domains, we also looked closely at the ads captured by our crawlers. There, we found that 29 (6%±2%) of domains abusing existing trademarks, displayed advertisements of a competitor. This means that when a user lands on such a trademark-abusive domain, not only would the trademark holder "lose" that user's visit, but the user could potentially end up on a website of a competing company. Given the presence of advertising syndicators and automatically computed advertisements based on the keywords in a domain name, we can relatively safely conclude that the blame here is with the domain parking service which did not check whether the domain was abusing trademarks, rather than with the competitor who ended up receiving the user.

F. Malicious redirections

Domain parking companies state that they provide a legitimate service that help visitors by showing them relevant advertising links [22]. While this can be true for domains without any trademark or typosquatting issues, there is another phenomenon that makes us even more skeptical about the goodwill of domain parking services.

Out of our initial 3,000 randomly sampled pages, our instrumented browser was redirected to a different domain in 7%±0.9% of the cases, a feature of parking services that is called Pay-Per-Redirect (PPR) [15]. Note that our crawler never clicks on any advertising links thus the redirection can be fully attributed to the domain parking service. In a preliminary examination of these redirections, we witness the landing on dubious websites including, among others, malware, scams, and affiliate abuse.

In order to examine this phenomenon more thoroughly, we sampled 100 domains from each parking service and setup an additional crawler which crawled them daily for a week. During this crawl, we recorded the entire redirection chain, visited and downloaded all links present on the final page, and kept track of all downloaded data and files. To assess any geographical differences, we performed this crawl in parallel from the United States and one country from Europe. Table II shows that ten out of fifteen studied services are conducting

	United States				Europe			
Service	Redirections	Malware	Scams	Adult	Redirections	Malware	Scams	Adult
Parking Service 1	0.4%	66.7%	-	-	-	-	-	-
Parking Service 2	1.3%	11.1%	-	-	0.4%	-	-	-
Parking Service 3	1.9%	-	-	-	2.0%	-	42.9%	21.4%
Parking Service 4	2.6%	44.4%	-	-	3.0%	-	-	38.1%
Parking Service 5	5.0%	-	-	(60.0%)	5.0%	-	-	(60.0%)
Parking Service 6	8.6%	3.3%	21.7%	-	2.6%	-	-	(50.0%)
Parking Service 7	12.4%	60.9%	1.2%	-	12.0%	-	26.2%	10.7%
Parking Service 8	19.4%	42.7%	6.6%	-	10.9%	-	26.3%	2.6%
Parking Service 9	34.6%	9.1%	2.1%	-	34.6%	0.4%	46.3%	0.8%
Parking Service 10	65.4%	21.0%	7.4%	-	66.0%	-	54.5%	27.7%

TABLE II. THE PERCENTAGE OF REDIRECTIONS THAT OCCURRED PER SERVICE PER REGION. OF THESE REDIRECTIONS, THE RELATIVE MALICIOUS AMOUNT IS GIVEN PER CATEGORY: MALWARE, SCAMS AND ADULT. REDIRECTIONS LEADING TO ADULT CONTENT THAT ORIGINATED FROM ADULT-ORIENTED PARKED DOMAIN NAMES ARE PUT IN PARENTHESIS.

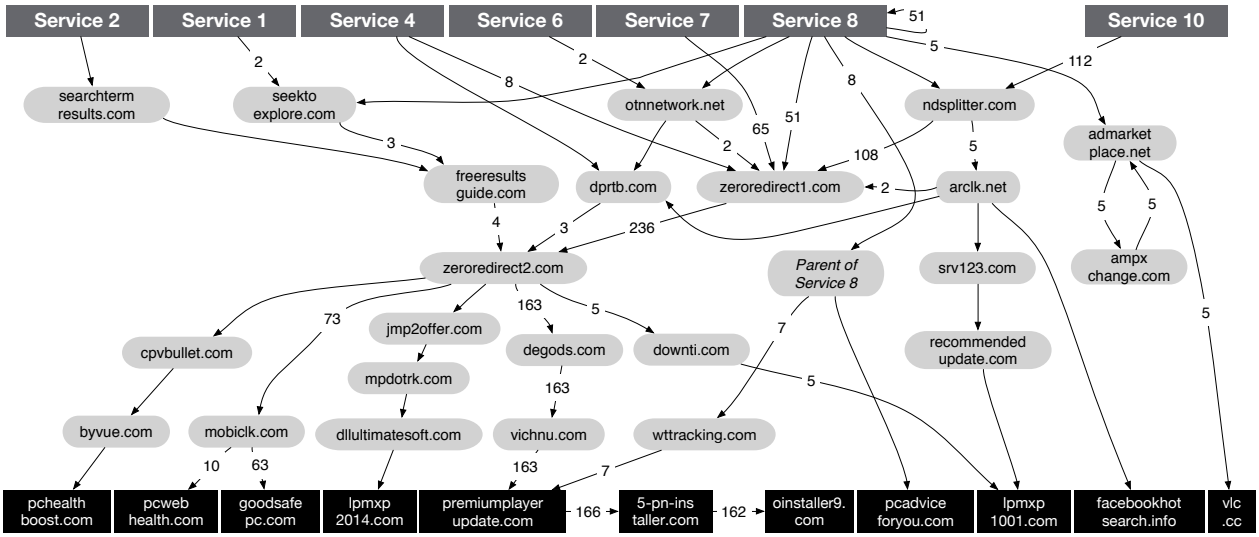

Fig. 6. Graph of redirections leading to malware recorded by the US crawler. The graph shows a page hosted by a parking service (top row), following redirections throughout various nodes, towards domains hosting malware (bottom row). The graph excludes Parking Service 9, because it had no common nodes with the other services.

redirects[2]. Parking Service 10 tops the list by redirecting in 2/3 of the cases, followed by Parking Service 9, which redirected about 1/3 of the visits. Some services redirect far more often than others, but there is no clear trend in relation to the size of the service's domain portfolio.

We distinguish between three categories of malicious redirections: malware, scams and adult material. In order to identify different campaigns, we automatically clustered malicious websites based on their screenshots through the use of perceptual hashing [24], a technique that generates distinct hash values that are robust to small image changes. After this process, we manually corrected any wrong clustering, labeled the resulting clusters, and extracted the chain of redirect domains for each campaign.

[2]Previous research has shown that malicious advertisements can occasionally be delivered by legitimate advertising networks [33]. For this reason, we decided to anonymize the names of the domain parking services which delivered malicious or inappropriate content during our measurements.

A website was classified as containing malware, if it hosted any downloadable executables that are flagged as malware by at least one antivirus engine of the VirusTotal service [9]. The redirections that contained malware were hosted on 16 different domains, most of them trying to convince the visitor to download a malicious update for their Flash player or browser. Interestingly, as shown in Table II, these malware campaigns seem to aim almost exclusively on American traffic, since we encountered them mostly on our crawler behind an IP address located in the US. One plausible explanation for this phenomenon is the high cost of advertising targeted at US traffic, compared to the rest of the world. As such, malicious advertisers may be much more aggressive so that they can recuperate their high advertising costs. Related to this is the fact that compromised machines are worth much more in the US than the rest of the world [17], [26], presumably because of their increased trustworthiness and, in extent, ability to be monetized better through spam and other malicious operations. The effect of geographical location on the type of abusive

ads was also recently observed Nikiforakis et al. [29] who studied malicious advertising in the context of ad-based URL shortening services.

Figure 6 summarizes the redirection chains of seven services that have been found to redirect to malware-laden websites. The graph shows that many intermediate parties are involved in leading a visitor to a malicious website and several nodes account for redirections of multiple services, such as `zeroredirect2.com`. Possibly, these complex chains are the consequence of a process similar to *ad arbitration*, a widely adopted practice performed by most ad syndicators [33]. During this process, the syndicator bids on available ad slots of other publishers or syndicators, allowing them to resell these slots to the next bidder. Often, ad slots are subjected to multiple iterations of this reselling process. As a consequence, ad slots are no longer under control of the syndicator that the original publisher partnered with. All these interactions and intermediate parties have the potential to blur the direct involvement of the parking service in serving malware. In some cases, however, we also see malware being delivered more directly, for example, by the parent company of Parking Service 8.

In terms of scams, we encountered several different kinds of campaigns. In one campaign, spanning eight different domains, the advertiser was trying to persuade users to hand over highly sensitive information, such as their Social Security Number, to allegedly retrieve the user's creditworthiness. In one particular case, typosquatting domains, such as banjofamerica.com, were used to increase the credibility of the scam. The displayed page had a special notice for "Bank of America visitors" warning them to urgently check their credit score because of a security breach.

Another kind of scam, of which we found two campaigns residing on three domains, claims that the visitor's computer is infected with malware. The web page insists on calling a given phone number for support. An example of this is depicted in Figure 7, which abuses the Norton logo to increase its trustworthiness. We called two of these numbers, pretending to be a clueless victim. Both "support lines" offered us assistance for the bogus infection. They required us to install a remote desktop application and inspected our machine for malware. For the purposes of this experiment, we setup a virtual machine running Microsoft Windows XP and installed a handful of popular desktop applications. Since we had just created these virtual machines for our experiment, we are confident that no malware was present on our systems.

The two "technicians" who were given remote access to our machine, used similar social engineering techniques to try to convince us that our computer was infected with malware. More specifically, they inspected our list of processes and showed us completely benign warnings from the Windows Event Viewer log, which they claimed were there because of malware. One service offered to remove this malware for 150 USD, the other asked 250 USD, plus another 250 USD to install antivirus software[3]. Other scams involved the user's participation in surveys which convince users to disclose their sensitive information by promising them high value coupons

[3]An audiovisual recording of this call can be viewed online at http://vimeo.com/101502467, with the password *NDSS2015*

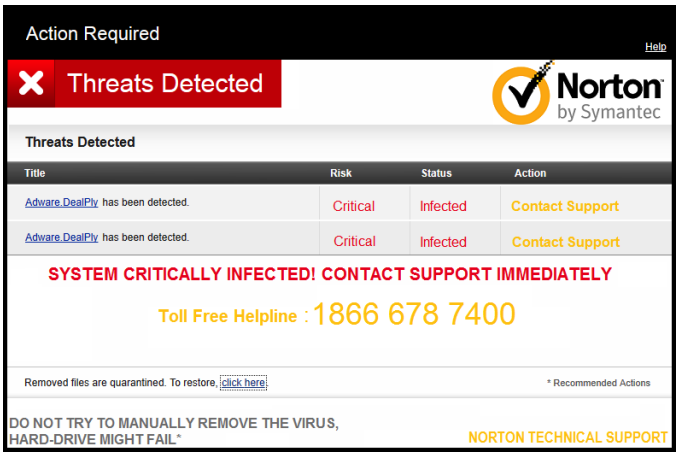

Fig. 7. Screenshot of a scam website to which our crawler was redirected after visiting a parking page.

for big box stores which, somewhat predictably, are never delivered.

When one considers both regions, all ten services were found to redirect to malware or scams at some point during those seven days.

Finally, the last category of observed malicious redirections leads to adult websites. We encountered seven domains hosting pornographic content. As seen in Table II, such redirections were most prevalent in our European crawler. Seven of the parked domains that redirected to adult material originated from adult-oriented parked domain names, specifically those parked with Parking Service 5 and 6. We assume users expect to be exposed to adult content when typing in such domain names, therefore, these redirections were considered non-malicious and put in parenthesis. However, the other 42 parked domains that automatically redirected to pages with sexually explicit content were either unrelated or even completely inappropriate, such as southwestairlanes.com and arabianmarriage.com and thus considered as malicious redirections.

Overall, while malicious advertisers are ultimately responsible for malicious ads, one cannot fully excuse the domain parking services involved in putting users in harm's way. The large percentages of discovered abuse suggest either a complete lack of countermeasures against malicious advertising, or the presence of very ineffective ones. Moreover, given the way that some of these scams were set up, we think that it would be really hard to successfully press charges against these services, since, for instance, in the scam involving virus warnings, the user voluntarily calls their support centers and gives them access to his machine. These kinds of scams show that legislation is not always sufficient to protect or compensate users, and thus highlights the necessity of technological countermeasures and user education.

G. Detecting and Bypassing Ad-Blockers

Today, some of the most popular extensions in browser markets are ad-blocking extensions. These extensions typically operate by scanning each web page against a blacklist of advertising-related regular expressions and prevent the

matched content from being loaded. Since domain parking services mostly rely on visitors clicking on, usually omnipresent, advertisements for their revenue, the adoption of these tools is a big threat to their business model. For each investigated service, we tested whether the domains parked with them tried to detect or bypass the workings of ad-blocking extensions. Note that since these services have full control of their parked domains, the discovery of a parked domain being involved in a specific practice equates to the parking service being involved in that same practice. Overall, we discovered that 2 out of the 15 studied services attempted to detect and bypass advertising blockers.

a) NameDrive: NameDrive ships their parked websites with an additional detection mechanism for ad blockers. They include an external file called *advertisements.js*. However, unlike the file name suggests, this file does not contain any code related to advertisements: it is just a single statement that sets a variable to `false`. This file is basically a honeypot for ad blockers: it deliberately attempts to trigger the blocking of resources by exposing a very obvious advertising-related filename for its JavaScript code. The file is detectable by the `"/advertisements."` present in the *EasyList* [1] blacklist, one of the most common blacklists used by a range of ad-blocking extensions, including the popular AdBlock Plus. Later on, another script included on the page verifies if the variable was actually set to `false` or not. If not, the user is automatically redirected to a completely different website for PPR monetization, as described in Section III-F, since no money can be made from this user from regular PPC advertising schemes. As such, we expect the number of redirections to be higher for users browsing with ad-blocking extensions.

b) Fabulous.com: Websites parked with Fabulous contain JavaScript code that verifies whether the JavaScript object named *google* exists. This object is normally initialized by a Google Adsense script included in the parked page. If this object does not exist, one can reasonably assume that advertisements are blocked. Subsequently, the web page reacts by creating iframes with content generated from other Fabulous pages. The generated content contains internal advertisement links, which when clicked, eventually redirect the user to the websites of advertisers.

H. Summary of Findings

By analyzing more than 8 million parked domains, hosted with 15 different services, we found that the lion's share of the domain parking ecosystem is under the control of a small fraction of the involved entities. For instance, we discovered that the majority of domains are in the hands of a small fraction (16%) of all domain owners, and that just the three most popular parking services are accommodating over 60% of the examined domains. This trend continues when it comes to the monetization through advertisements, as a single, popular, advertising syndicator provides PPC ads for 90%$\pm$1.1% of all visits to parked domains. Since only a handful of parties is responsible for the majority of the ecosystem's monetization chain, we argue that a change in policy or the business model of these large players, could drastically and effectively influence the domain parking scene.

In terms of abuse, we found that only one parking service did not include typosquatting domains in their managing portfolios, with all other services accepting and profiting from these abusive domains. Specifically, through automatic measurements, we conservatively estimated that a service's domain portfolio may contain over 4% typosquatting domains. In a more in-depth manual sample analysis, however, we found 16%$\pm$3.2% of parked domain names to be either trademark or typosquatting abuse. A reasonable assumption is that typosquatting domains receive more visitors than most generic parked domain names, thus contributing to a large extent to the profits generated in the domain parking industry. As such, foregoing the profits associated with accepting typosquatting, and other cybersquatting domains, is likely not something that these services will do voluntarily.

Furthermore, we examined the Pay-Per-Redirect (PPR) phenomenon, which is used as a secondary monetization source, performed in 7%$\pm$0.9% of the visits to parked domains. All ten services deploying this redirection strategy have been found to unexpectedly send their visitors to malware-laden websites, scams or explicit adult content. This phenomenon shows that parking services are not reluctant to deploy malicious monetization strategies at the expense of user safety.

IV. Detecting Parked Domains

The previous section analyzed the ecosystem of domain parking and mapped out the practices involved in their business model. Based on our findings, we do not consider it much of a stretch to claim that, at their current state-of-practice, domain parking services act in a parasitic way. As such, it is important to implement countermeasures in order to reduce their prevalence or at least minimize the user's exposure to them. We approach this problem by proposing and developing a classification model that is able to detect parked pages. This model is meant to be robust, meaning that it does not rely on any parking services' specifics, such as their specific name servers, but rather relies on features inherent to the conceptual operations of parked domains. Applications for this model exist on various levels, e.g., it could be part of a search engine crawler, where the detection could be used to discard parked pages from their search results, or part of a browser extension that detects and blocks parked pages when a browsing user encounters one.

In this section, we walk through the construction of the classification model by describing how the data was gathered, which features were used, how the model was tuned, and how the classifier was trained and evaluated.

A. Gathering data

We begun this process by first obtaining a random sample of 3,000 verified parked pages and 3,000 pages from the Alexa top 1 million. We automatically verified that the sampled pages from Alexa are not parked by examining their name servers and ensuring that they do not match the name servers of any of the studied parking services. Next, we crawled all 6,000 pages and collected data from several different sources. More specifically, we gathered the HTML source code of every loaded frame recursively, recorded a trace of all HTTP requests initiated by

the web page (HAR), as well as the redirection chains of the main page and every frame. Finally, we inspected properties of the domain itself, such as typosquatting occurrence and WHOIS records. From this data, we can extract discriminative features that can serve as input for our classifier.

B. Feature set

When creating features to detect parked pages, we try to target the inherent nature of the parking services' operation model. This approach results in more robust detection, as opposed to searching for traces of specific parking services or looking for fixed keywords.

We focus on detecting the omnipresence of third-party advertising, dynamic and on-the-fly page generation, lack of content, malicious redirections, and abusive domains. In total, we construct eleven HTML features, five HAR features, four frame features and one domain feature. We elaborate on the extraction of each feature and our rational for choosing them in the following paragraphs:

HTML Features: The HTML features are extracted from the source code of every loaded frame. From this code we can analyze the content and the scripts deployed on the page.

- **Average and maximum link length.** We count the number of `<a>` elements present on the page and measure the string length of the destination addresses. From these numbers, we can calculate the average and maximum link length of the page. The rationale behind this feature is that advertisement links, which usually form the majority on domain parked websites, pass more and longer parameters along with the link in order to track the click on the PPC ad. They might, for example, include the publisher's identifier, the final link destination, tokens, timestamps, etc.
- **Average source length.** Similar to the previous feature, source addresses for banners and other advertisement media, tend to pass parameters of campaigns, image dimensions, etc. We expect non-parked websites to have more static media sources and thus shorter address lengths.
- **External link and external source ratio.** We define an external link or source as one with an address pointing to a another domain. Links and media generated by third-party advertisement syndicators will generally reside on domains of that syndicator. We expect non-parked websites to have a lower ratio of external links, because they commonly also have links to pages and media hosted on the same domain.
- **Website directory presence.** Since parked domains are undeveloped websites that display content that is generated on-the-fly, it is uncommon for them to have dedicated directories on their website. We search within the HTML source and link addresses for the presence of a directory and use this as a boolean feature.
- **Link-to-global text ratio.** Many parked pages have hardly any text on their page that is not part of a link. On a typical parked page, text is either part of an ad or part of the "Related links". To assess this characteristic, we extract all text from the HTML pages with Python's Natural Language Toolkit [16], which omits the HTML tags and returns the textual content. We compare the amount of text that resides within links (`<a>` elements and their child nodes) to the global amount of text present on the page.
- **Amount of non-link characters.** To more robustly test the characteristic of the previous feature, we incorporate an additional feature that counts the actual amount of characters not belonging to any link element, instead of solely relying on the ratio.
- **Text-to-HTML ratio.** We also measure the ratio of text to the total amount of characters in the HTML file. This feature focuses more on the dynamic generation of content.
- **Redirection mechanisms** Parked pages use redirection mechanisms to lead visitors to other pages or domains. Although non-parked pages might also deploy such mechanisms, we still believe that this feature, when considered together with other ones, can assist classification. We detect two different redirection methods. One feature records the presence of JavaScript redirection code by searching for `window.location`, while the other finds HTML meta refreshes by looking for `http-equiv="refresh"`.

HAR Features: These features are derived from the HTTP archive (HAR) that is constructed while loading a page. We focus on the following discriminative characteristics of HTTP requests:

- **Third-party requests ratio.** We extract the number of HTTP requests to third-parties (other domains) and the total amount of requests. Next, we calculate the ratio between those two. This feature is motivated by the amount of third-party content and media generated on parked pages. In addition, HTTP requests conducted after redirecting to a different domain, are all considered third-party requests, with respect to the initial domain.
- **Third-party data ratio.** Similarly, we calculate the ratio between data (number of bytes) coming from third-party sources and all incoming data.
- **Third-party HTML content ratio.** This feature further incorporates the characteristics of third-party content. We expect most third-party content on regular websites to be generally JavaScript libraries and media files. Parked websites, however, are known to include `html/text` content pulled from third-party services, such as through the use of iframes generated by ad syndicators. For this reason, we include a specific feature that represents the ratio of HTML content brought in by third-party requests.
- **Initial response size and ratio.** For this feature, we first record the size of the initial response when making the first request to the web page. Next, we compare this with the total amount of received data

after completely loading the website. This feature attempts to capture the dynamic generation of content on parked pages which is a core concept of the modus operandi of domain parking services. With this feature, we expect to identify the initial lightweight page skeleton, which stands in contrast with the final amount of received data.

Frame Features: The following frame features are extracted by tracking every loaded frame on the web page.

- **Amount of frames.** While manually inspecting the structure of parked pages, we found that the presence of iframes is very common. In order to take this into account, we recursively count all frames and iframes present on a page and its child frames.
- **Main frame and iframe redirections.** The redirection chain of every frame was tracked when we crawled the domains. For every chain, we extract the number of redirections that occur on the main frame as well as all other frames. As noted in previous sections, malicious redirects initiating at parked pages contain many different traffic distributors and redirection hops, e.g., as shown in Figure 6. Thus, we expect a benign redirection chain to consist of a limited amount of intermediate steps.
- **Different final domain.** This feature checks if the main frame (i.e. the frame of which the address is visible in the browser's URL bar) was redirected to a different domain. It excludes internal redirections, such as from `www.domain.com` to `blog.domain.com`, which is a more common redirection process on regular websites.

Domain name feature: This feature focus on characteristics inferred from the domain name itself.

- **Typosquatting domain.** The current domain name is checked for typosquatting abuse with the algorithm described in III-C. Regular, authoritative websites should not be flagged by this feature.

C. Classification

Our objective is to construct a classifier that can reliably detect parked websites when visiting them. When loading a web page, the aforementioned features are extracted and are treated as the features of a particular *instance*. The classification model's goal is to take an unknown instance as input, process the features, and assign a probability of that instance belonging to a certain class. More specifically, the model calculates whether or not any given web page is likely a parked one. Given these probabilities, a threshold value can be used to actually classify the instance as either parked or non-parked. This threshold can then be appropriately varied to tune the sensitivity of the model.

In order to build the classifier, we first select an appropriate learning algorithm for our model. Afterwards, this model needs to be given a sufficient amount of parked and non-parked instances for it to learn from. Once a model is learned, we can evaluate the performance of the classifier with unseen test instances.

1) Learning method: We aim for high interpretability of our classifier, as it is important to comprehend the prediction of the classifier for further improvement and adaptability. This quality of a classifier is also important if our model is to be incorporated in a browser and be used by users. Therefore, we opt for the Random Forest algorithm, as it combines the strength of ensemble learning with the interpretability qualities of decision trees. Furthermore, decision trees tend to be robust with regard to outliers, while the ensemble technique of Random Forest protects the model against overfitting. Moreover, after the trees have been constructed in the learning phase, the classification is usually very quick in the detection phase.

2) Dataset handling:

a) Train and test dataset: The set of instances used in the learning phase of the model is referred to as the *training set*, for which we reserve 2/3 of our dataset of 6,000 pages. As follows, 1/3 of the set is isolated and used after the model is built, in order to adequately evaluate our classifier. Since these instances are used to test a classifier, they are referred to as the *test set*.

b) Data transformation: In order not to overfit the classifier, we remove outliers and extreme values from our training set using an interquartile range filter, which is configured to operate on a per feature basis.

c) Omitted features: Our initial set of features contained two additional features, which, however, were omitted after manual and algorithmic selection.

One of these features was based on the detection of WHOIS entries that make use of anonymizing services, such as WhoisGuard [11]. We expected parked domain owners to have a higher incentive to anonymize their personal association with their domains, as opposed to regular domain owners. However, when manually inspecting the features of the training set, we found that this feature was not very discriminative for parked domains. Furthermore, since it required searching for specific strings related to anonymizing services, the feature is also less robust. For these reasons, the feature was removed from our final model.

To further improve our feature selection, a backwards greedy stepwise search was conducted on the remaining feature set. The search started with the complete feature set, and removed features one by one until a better classification result was achieved. The algorithm found that another feature was reducing the performance of the classifier. This particular feature counted the number of times the parked domain name was passed along with HTTP request parameters. The reasoning behind this feature was that, often, the domain name is sent as a parameter in a request to a third-party. Parked domains do this in order to enable the parking service to return content or ads relevant to the given domain, as described by Wang et al. [31]. Nevertheless, the feature was found to confuse our classifier and was thus omitted from the model.

3) Evaluation: After transforming the data and selecting the best performing features, we built a model from our training set using Random Forest with 10-fold cross validation. As a model's performance is not determined by its effectiveness on the training set, but on its ability to classify unknown instances, our test set was used for evaluation purposes. The trained

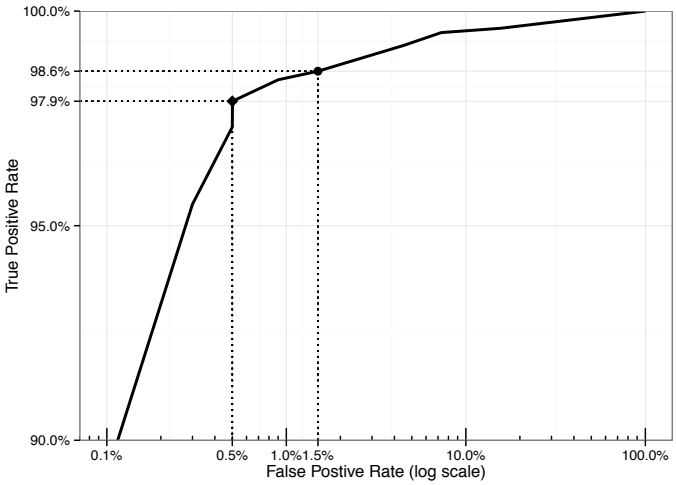

Fig. 8. ROC curve of the parking detection model on the test dataset. The model achieves 99.65% AUC. Two threshold points (0.5 and 0.7) indicated on the curve with their resulting FPR and TPR.

model processed 1,000 unseen parked and 1,000 unseen benign instances. Next, the ROC curve, Figure 8, was generated by varying the classification threshold and keeping track of the resulting True Positive Rate (TPR) and False Positive Rate (FPR). With a 99.65% area under the curve (AUC), the model proves to generalize very well, resulting in a performant classifier.

The default threshold, set at 0.5, results in a FPR of 1.5% and a TPR of 98.6%. Since we consider the cost of a false positive to be higher than a false negative (i.e., it is worse to classify a normal website as a parked one, than vice-versa), we attempt to increase the specificity of the model. As one can see from the ROC curve, it is possible to reduce the false positive rate without sacrificing too much sensitivity. With the threshold set at 0.7, a FPR of 0.5% is achieved (i.e. 5 out of 1,000 benign domains were falsely classified as parked domains) together with a TPR of 97.9%, which results in an overall classifier accuracy of 98.7%. Both thresholds and their resulting FPR and TPR are indicated on Figure 8.

Looking at the performance statistics, we consider the classifier highly capable of protecting a browser, search engine, or other system by detecting parked websites on the web.

4) Detection Evasion: It is reasonable to assume that once a classifier is deployed at a large enough scale, the affected parties will attempt to evade it. The presented classification model, however, purposely focuses on features inherent to the parking services' workings. As such, an evasion sufficient to purposefully misclassify a parking page is non-trivial. Domain parking services would face all sorts of obstacles while attempting to bypass the features of the classification model. For instance, they would need to substantially decrease the relative presence of third-party advertising by either providing a large portion of first-party content for each domain, or removing third-party syndicators all together. Both options require major refactoring of their current operations. Additionally, redirections chains and mechanisms are tracked in every frame, so the PPR monetization would need to be abandoned. Likewise, typosquatting domains could be excluded from parked domain portfolios to reduce detection, but this would be a "welcoming" evasion since it would effectively reduce the level of abuse present in domain parking services.

In other words, we opine that if the classifier would be widely deployed, the domain parking industry would either be forced to shift to a different, hopefully more legislated, business model, or lose its remaining exposure to users, both effectively mitigating current abusive monetization techniques.

V. Discussion

Although parking services have been around for over a decade, they have always been controversial, mainly because of the limited added value they contribute to the web, which stands in contrast with the millions of domains associated with them. The people who disagree with this opinion, are usually domainers and the services themselves who argue that domain parking is of legitimate help to users by assisting them in finding relevant content [22].

While the business climate for domainers was extremely satisfactory in the early 2000s and a lot of money was made with parking services, the circumstances have changed substantially since then. If we look at the annual reports of the Sedo Holding [8], the service managing the largest domain portfolio, we observe the sales revenue of their "Domain Marketing" segment dropping year after year. Since 2007, they have been losing up to 17.9% yearly, now summing up to a 34.9 million EUR (56%) decline in sales revenue in 2013. The report explains this profit shrinkage, and the overall downtrend of the parking industry, as further impacted by "Advances in browser technologies." We postulate that these "advances" might refer to the deep integration of search engines in the browser, as described in Section II-B. Most likely, this integration heavily reduced type-in traffic for parking pages. Furthermore, the adoption of ad-blocking tools is another browser technology that might be causing a huge setback for their monetization capabilities, as explained in Section III-G. Lastly, one more obstacle for the industry was introduced by an update to Google's search engine in December 2011 [3] where Google decided to exclude parked domains from Google's search results.

We speculate that parking services (and domainers) have become desperate to counter this problematic regression, and are therefore resorting to shadier or even downright malicious ways to attract and monetize traffic. It is clear that typosquatting domains are still a regular part of their business model, as these are probably one of the only remaining types of domains that reliably receive type-in traffic. Next to abuses of trademarks, we encountered parked domains that redirected to malware, scams and adult material, for which we can reasonably assume that they pay more than legitimate redirections.

A similar observation is made in concurrent work, where the involvement of parking services in click fraud towards their advertisers, and the reluctancy to distribute the rightful commission to the domain owners, is attributed to the decrease in revenue [15].

If we look at the number of services involved in fraud, malicious activities and monetization of abusive domains, we opine that domain parking has become a threat for all users on the web as well as for the (legitimate) advertising industry.

VI. Related Work

In concurrent work, Alrwais et al. [15] investigated the domain parking industry, focusing on the fraudulent practices in their monetization chain. In their work, the authors registered themselves both as domain owners and advertisers, effectively operating at both ends of the domain parking ecosystem. This allowed them to connect the dots and detect any discrepancies between what they knew to be true and what the domain parking services claimed as true. Using this method, the authors uncovered several fraudulent practices including the presence of click fraud, where their advertising campaigns were charged for receiving visitors through clicks, even though the visitors were the researchers' crawlers which were configured to never "click" on any ads.

While Alrwais et al. focus their investigation on the abuse against advertisers, we instead focus on the abuse against users landing on parked domains, as well as domain owners whose trademarks are being diluted because of the absence of trademark-infringing checks from the domain parking services. Though our work is orthogonal to their work, we both arrive at the same conclusion: Domain parking services are currently unlegislated and that allows for a lot of abuse, towards all third-parties of the domain parking ecosystem: users landing on parked pages, advertisers paying for ads, and holders of popular trademarks and domain names. To that extent, a client-side countermeasure, like the parked-page classifier we propose in this paper, can protect the user-part of this ecosystem while we hopefully transit to a more legislated domain parking industry.

In 2010, Almishari et al. [14] developed a classifier for "ads-portal" domains, which they used to identify typosquatting abuse of domain parking services. They observed that in 2008, 50% of parked pages were residing on typosquatting domains. Now, 6 years later, we witness that parking services have a significantly smaller, albeit still substantial, typosquatting portfolio. The classifier they developed leveraged several HTML features similar to the ones proposed in this paper. For instance, they focused on the dominant presence of anchor and image elements and the numerous parameters passed along with these URLs. However, they did not incorporate HAR features, which are able to identify the dynamic and external nature of parking pages. Furthermore, they did not take into account redirections and there no in-depth frame analysis was made. In terms of performance, they incorporated keyword-based features to increase the accuracy of the classifier, a strategy that we deliberately avoided in order to ensure robustness.

Domain parking was also recently mentioned by Li et al. who studied generic malicious web infrastructures [23]. Interestingly, the authors observed that domains hosting malicious Traffic Distribution Systems (TDS), were monetized through parking services after the TDS had been taken down. Even after a take down, these domains keep on receiving a large amount of traffic since numerous malicious doorways (typically compromised websites) are still redirecting users to the TDS domain.

Prior to the aforementioned research, domain parking was only mentioned in research regarding cybersquatting. One of the oldest studies of typosquatting by Wang et al. [31] proposed a series of typosquatting models and showed that, in 2006, 51% of all the possible typosquatting domains of websites in the Alexa top 10,000 were registered and active. The authors uncovered that 59% of the active typosquatting domains were hosted at 6 major domain parking companies and reported that inappropriate adult content was encountered on typos of children's websites.

In a later study, Moore and Edelman tried to identify the entities responsible for typosquatting [25]. The researchers observed that name servers belonging to parking companies have up to 4 times more typosquatting domains than average. In terms of advertising, they identified Google as the prime source for PPC ads on typosquatted domains. Furthermore, they encountered abusive domains to be serving ads for the authoritative domain (self-advertising) and for competitors. Additionally, they measured a similar concentration phenomenon as to what we noted in the domain parking ecosystem: 63% of typosquatting domains that were using Google PPC ads, belonged to only 5 different publisher IDs. In our study, we observed that not only Google maintains almost complete control of advertising in domain parking services, but that they do so while claiming that domain parking pages are spam [19].

In a very recent content-based typosquatting study, Agten et al. [12] verified that parked pages are still the most prominent monetization strategy for typosquatting domains. More specifically, ad parking was witnessed in 51% of observations. Furthermore, they detected malicious redirections towards malware, scams and adult content. Taking into account the observations from our research, it is likely that these redirections were initiated by parking services that hosted the typosquatting domains.

VII. Conclusion

Despite existing for over ten years, managing millions of domain names, and making yearly revenues of multiple millions of dollars, domain parking services have received little attention so far.

In this paper, we mapped the ecosystem of domain parking services by identifying popular parking services, the types of parked domains, the owners of these parked domains, and the advertising content that users are exposed to when they land on the pages of parked domains. In this process, we identified multiple types of abuse including malware, inappropriate advertising, and scams that could cost users their personal details and hundreds of dollars. Next to the abuse affecting users, we also discovered a significant fraction of typosquatting domains being monetized through domain parking, and witnessed the lack of controls by successfully parking an obviously abusive domain with all domain parking services on which we could get a parking account.

Motivated by the discovered abuse, we designed and built a parked-page classifier which can be used to, among others, block parked pages, or alert users that they are currently interacting with a parked page. Instead of hardcoding parking signatures, we compiled a list of generic and robust features that are characteristic of parked pages and showed that our classifier has a very high accuracy with only minimal false positives.

ACKNOWLEDGMENTS

We want to thank the anonymous reviewers for the valuable comments. For KU Leuven, this research was performed with the financial support of the Prevention against Crime Programme of the European Union (B-CCENTRE), the Research Fund KU Leuven, the IWT project SPION and the EU FP7 project NESSoS.

REFERENCES

[1] Adblock plus - easylist. [Online]. Available: https://easylist-downloads.adblockplus.org/easylist.txt

[2] Dns census 2013. [Online]. Available: http://dnscensus2013.neocities.org/

[3] Google algorithm change history. [Online]. Available: http://moz.com/google-algorithm-change#2011

[4] Hosted domains have been retired. [Online]. Available: https://support.google.com/adsense/answer/2456470?ctx=as2&rd=2

[5] Mozilla - Domain Guessing. [Online]. Available: http://www-archive.mozilla.org/docs/end-user/domain-guessing.html

[6] Namepros - domain parking companies by payout. [Online]. Available: https://www.namepros.com/parking-and-traffic-monetization/783661-domain-parking-companies-by-payout.html

[7] Phantomjs - a headless webkit scriptable with a javascript api. [Online]. Available: http://phantomjs.org

[8] Reports: Sedo holding ag. [Online]. Available: http://www.sedoholding.com/en/investors/reports/

[9] Virustotal - free online virus, malware and url scanner. [Online]. Available: https://www.virustotal.com

[10] Vulcan golf, llc v. google, inc. [Online]. Available: http://www.finnegan.com/vulcangolfllcvgoogleinc/

[11] Whoisguard: Protect your privacy using whoisguard (whois protection). [Online]. Available: http://www.whoisguard.com/

[12] P. Agten, W. Joosen, and N. Nikiforakis, "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS 15)*, 2015.

[13] A. Allemann. Survey: domainers have a new favorite domain parking company. [Online]. Available: http://domainnamewire.com/2013/02/19/survey-domainers-have-a-new-favorite-domain-parking-company/

[14] M. Almishari and X. Yang, "Ads-portal domains: Identification and measurements," *ACM Transactions on the Web (TWEB)*, vol. 4, no. 2, p. 4, 2010.

[15] S. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, and X. Wang, "Understanding the Dark Side of Domain Parking," in *Proceedings of the 23rd USENIX Security Symposium*, 2014.

[16] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*, ser. COLING-ACL '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 69–72. [Online]. Available: http://dx.doi.org/10.3115/1225403.1225421

[17] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, "Measuring Pay-per-Install: The Commoditization of Malware Distribution," in *Proceedings of the the 20th USENIX Security Symposium*, San Francisco, CA, August 2011.

[18] S. Carletti. Ruby whois. [Online]. Available: http://ruby-whois.org/

[19] I. Google. Fighting spam. [Online]. Available: http://www.google.com/insidesearch/howsearchworks/fighting-spam.html

[20] T. Holgers, D. E. Watson, and S. D. Gribble, "Cutting through the confusion: a measurement study of homograph attacks," in *Proceedings of the annual conference on USENIX '06 Annual Technical Conference*, ser. ATEC '06. Berkeley, CA, USA: USENIX Association, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267359.1267383

[21] M. Jakobsson and S. Myers, *Phishing and countermeasures: understanding the increasing problem of electronic identity theft*. John Wiley & Sons, 2006.

[22] D. Kesmodel, *The Domain Game: How People Get Rich from Internet Domain Names*. Xlibris Corporation, 2008.

[23] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. Wang, "Finding the linchpins of the dark web: A study on topologically dedicated hosts on malicious web infrastructures," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, ser. SP '13, 2013, pp. 112–126.

[24] V. Monga and B. L. Evans, "Perceptual image hashing via feature points: performance evaluation and tradeoffs," *IEEE Transactions on Image Processing*, vol. 15, no. 11, 2006.

[25] T. Moore and B. Edelman, "Measuring the perpetrators and funders of typosquatting," in *Financial Cryptography and Data Security*, vol. 6052, 2010, pp. 175–191.

[26] E. Naone, "MIT Technology Review: Get Paid to Install Malware," http://www.technologyreview.com/view/417354/get-paid-to-install-malware/.

[27] N. Nikiforakis, S. V. Acker, W. Meert, L. Desmet, F. Piessens, and W. Joosen, "Bitsquatting: exploiting bit-flips for fun, or profit?" in *Proceedings of the 22nd International World Wide Web Conference (WWW)*, 2013, pp. 989–998.

[28] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen, "Soundsquatting: Uncovering the use of homophones in domain squatting," in *Proceedings of the 17th Information Security Conference (ISC)*, 2014, pp. 291–308.

[29] N. Nikiforakis, F. Maggi, G. Stringhini, M. Z. Rafique, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, and S. Zanero, "Stranger danger: Exploring the ecosystem of ad-based url shortening services," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14, 2014, pp. 51–62.

[30] J. Szurdi, B. Kocso, G. Cseh, M. Felegyhazi, and C. Kanich, "The long taile of typosquatting domain names," in *Proceedings of the 23rd USENIX conference on Security Symposium*. USENIX Association, 2014, pp. 191–206.

[31] Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels, "Strider typo-patrol: discovery and analysis of systematic typo-squatting," in *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, ser. SRUTI'06. Berkeley, CA, USA: USENIX Association, 2006.

[32] B. Woods. 15 of the most expensive domains of all time. [Online]. Available: http://thenextweb.com/shareables/2013/08/13/15-of-the-most-expensive-domains-of-all-time/

[33] A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna, "The dark alleys of madison avenue: Understanding malicious advertisements," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014, pp. 373–380.