Measuring and Analyzing Typosquatting Toward Fighting Abusive Domain Registrations

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
in Department of Electrical and Computer Engineering

Janos Szurdi

B.S., Electrical & Computer Engineering,
Budapest University of Technology and Economics
M.S., Electrical & Computer Engineering,
Budapest University of Technology and Economics

Thesis Committee:
Nicolas Christin, Chair
Manos Antonakakis
Lujo Bauer
Vyas Sekar

Carnegie Mellon University
Pittsburgh, PA

July 2020

Keywords: Typosquatting, DNS, Online Crime, Domain Registrations, Domain Name, Abuse, Measurement, Network security, Ethics, Economics, Policy, Computer Security

For my daughter, wife, family and friends.

Acknowledgments

I am grateful for the many years of support from my advisor Nicolas Christin. He provided guidance in ensuring the technical soundness of my research projects and helped to improve my writing and presentation skills.

I also want to thank my committee, Manos Antonakakis, Lujo Bauer and Vyas Sekar, for their valuable feedback to improve this thesis. The work presented in this thesis could not have been achieved without the help of my collaborators: Nicolas Christin, Gabor Cseh, Mark Felegyhazi, Chris Kanich, Balazs Kocso, Brian Kondracki, Meng Luo, Nick Nikiforakis, Jonathan Spring.

Initially, lectures by Levente Buttyan sparked my interest in Computer Security, and it was Mark Felegyhazi who introduced me first to the world of security research and why I chose to become a researcher.

I am thankful to the members of CyLab for always being there for a conversation or a good laugh. It was my closest friends who made my life outside of Ph.D. fun all these times. From mischiefs to serious conversations, I am lucky that I could always count on Sruti Bhagavatula, Aymeric Fromherz, Aaron Harlap, Mahmood Sharif and Josh Tan.

My family has always been there for me, providing the support I needed in any shape or form: my father Andras, mother Juli, big sister Andrea and big brother Miklos. My nieces and nephews Ivan, Anna, Emma, Mate and David always proved to be a great source to cheer me up.

Most importantly, it is my wife and best friend Orsi who helped the most all these years and I am truly thankful for her. We had many adventures together, and I am very excited about the newest chapter of our life with our daughter to be born soon.

The research in this thesis was partially supported by the National Science Foundation grant NSF CMMI-1842020, the NSF CNS-1223762 grant and by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). I was fortunate to be supported by the Ann and Martin McGuinn Graduate Fellowship.

Abstract

Inexpensive and simple domain name registrations foster a wide variety of abuse. One of the most common abusive registration practices is typosquatting, where typosquatters register misspelled variants of existing domain names to profit from users' typing mistakes. Making the matter worse, typosquatters frequently rely on advertisement networks to monetize user traffic, often exposing users to malicious and illicit content. Leveraging multifaceted large-scale measurement infrastructures, we demonstrate in this dissertation that typosquatting is a widespread issue which plays an important role in concert with other illicit traffic sources in exposing users to malice. Based on our measurement studies, we show how we can develop detection tools and leverage registration policies to reduce typosquatting and other abusive domain registrations.

Supporting our assertions about the extent and abuse of typosquatting, we design and implement three measurement infrastructures that lead to novel findings about typosquatting and related malicious domain registrations. First, to understand the extent of typosquatting, we study typosquatters who target less popular domain names. We find millions of typosquatting domains missed by previous research. Building on our findings, we create a classifier which can decide if a potentially typosquatting domain name is truly typosquatting or if it is just accidentally close to a target domain.

Second, we study how typosquatters send users to advertisement networks for profit. To gain a deeper understanding of the advertisement infrastructure redirecting users to malicious landing pages, we build a system that can emulate different types of users, can understand cloaking and blocking behavior and can reconstruct redirection chains. We find that typosquatters often share monetization strategies with ad-based URL shortening services and illicit movie streaming sites by redirecting users to the same malevolent landing pages. We also observe that miscreants differentiate users based on the device used and that using too few IP addresses can significantly decrease the number of abusive pages discovered. We develop a classifier, not specific to typosquatting and based only on features related to the redirection chain traversed by users, that can be leveraged to show warnings to users when a redirection is likely dangerous.

Furthermore, as DNS abuse is not specific to the HTTP protocol, we study how users' private emails are exposed to typosquatters. We find that 1,211 typosquatting domains receive in the vicinity of 800,000 emails per year and that millions of registered typosquatting domains have MX records pointing to only a handful of mail servers potentially enabling the collection of emails on a larger scale.

Finally, we develop a policy analysis framework based on the domain registration ecosystem finding that domain registration policies could have an essential role in complementing current detection based approaches to fight typosquatting and malicious domain registrations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Motivation

In the past few decades, the Internet became an important and crucial part of our lives, and with it came a plethora of online abuse. An essential part of the Internet is the Domain Name System (DNS) translating human-readable domain names (e.g., cmu.edu) to machine-readable Internet Protocol (IP) addresses (e.g., 128.2.42.10). The potential economic value, low cost and simplicity of registering domain names fosters a large number and wide variety of abuse. Miscreants use domain names to evade blacklisting, for accounting and business agility, to confuse users, to abuse residual trust in domains or to siphon traffic from legitimate domains. One of the most common abusive domain registration practices is typosquatting, where typosquatters register misspelled variants of existing domain names to profit from users' typing mistakes. While many attackers rely on programming bugs, protocol weaknesses or system misconfigurations to succeed in their various endeavors, typosquatters exploit human errors whilst typing domain names. These typing errors can happen in many different settings, including browsing [106, 128], sending emails [55, 126], setting up server configurations [136] or writing software code [22].

Typosquatters just like the rest of the Internet is primarily supported by the online advertisement industry [106]. In typical advertisement ecosystems, websites presenting ads to users, called ad publishers, leverage a complex system of advertisement networks to connect them with the most profitable advertisers. A standard economic model for online advertising is for ad publishers to receive compensation when users express interest in ads by clicking on them; called pay-per-click (PPC). This complex web of hundreds of millions of websites, multitudes of ad brokers and advertisers is an excellent cesspool for criminals to operate undetected. Traffic distribution systems (TDSs) are advertisement networks where both PPC and automatic redirection of users called pay-per-redirect (PPR) are common. In a TDS, actual advertisements are not always explicitly present, particularly in the case of automated redirections. Thus, we adopt a terminology often used in the context of TDSs [35, 92] to describe three entities playing an essential role in redirecting users. Traffic sources are pages visited by users for the content (free movie) or service provided (shortened URLs) or by mistake (typing mistake made). These traffic sources present advertisements to users (PPC) or automatically redirect them (PPR) to destination pages (advertisers). As there are many traffic sources and destinations, traffic brokers provide economic value by

matching sources with the highest bidding destination pages. Typically in TDSs, automatic redirection of users is often coupled with a variety of malicious destination pages.

Miscreants try to abuse many aspects of this advertisement oriented online ecosystem. Spammers leverage abusive advertisement tactics by sending billions of unsolicited emails, messages, posts, comments and tweets. To evade blacklisting, spammers use a large number of domain names both for their email addresses and for the spam advertised websites. Illicit free movie streaming websites often leverage less popular advertisement networks to profit from user visits. While these smaller advertisement networks accept traffic from illegitimate sources, they also frequently send users to malicious landing pages such as phishing and scam webpages. In phishing and scam attacks, domain names (e.g., facebo0k.com) similar to brands' domain names are used to fool users into sharing their personal information or into sending money to the perpetrators. When an expired domain is acquired with malicious intent, it can be used to exploit users' and programs' trust in the domain names, for example, for phishing attacks or to infect machines. While typosquatting can be used for phishing or to deliver attacks, in this thesis, we primarily focus on how typosquatting is used to profit from advertisements, frequently exposing users to malicious content.

Problem Statement At first glance, while typosquatting might seem simple and not particularly harmful, many typosquatting domains rely on a complex ecosystem shared with other illicit websites, often exposing users to malicious content. For a long time, the extent and monetization strategies of typosquatters were not well understood due to the scale and complexity of the problem. Five main challenges make it particularly hard to gain a clear picture of typosquatting and to understand the type and amount of malice users are exposed to. Without an adequate understanding of the typosquatting ecosystem, we cannot expect to protect users and to stop this phenomenon.

I. Problem Scale. First, the domain name space is enormous, consisting of hundreds of millions of domain names with tens of millions lexically close enough to potentially denote typosquatting. Thus the scale of the problem requires any comprehensive measurement study to cover a large number of domain names and to use significant computational and human resources.

II. Measurement in an adversarial environment. Second, any measurement of typosquatting is done in an adversarial setting, where the adversary's goal is to send users to malicious landing pages but hide their activity from security researchers and law enforcement to avoid blacklisting. These techniques to hide malicious content on web pages is often referred to as cloaking. To make the matter worse, all advertisement networks try to block automation. Thus measurement infrastructures need to become more sophisticated in realistically emulating user behavior.

III. User differentiation. Third, advertiement networks differentiate users, and thus we cannot expect to understand the abuse present by visiting these pages only as one kind of user. Therefore the measurement apparatus needs to emulate different types of users to observe the malicious contents they are exposed to.

IV. Multi-protocol problem. Fourth, DNS and typosquatting are not specific to the Web. Thus measurement across multiple application layer protocols relying on DNS is necessary to truly understand the threats users face.

V. Policy challenge. Finally, DNS abuse is a global problem, and local detection and blacklisting methods are only partially successful in defending users. The alternative option of taking down domains is cumbersome, faces legal issues and currently not scalable. Therefore it cannot keep up with fast-paced registration practices. As a result, complex analysis driven policies are necessary to address typosquatting and other abusive domain registrations.

By taking the first steps to address the challenges mentioned, we gain new insights into the typosquatting ecosystem and the threat it presents to users.

This thesis shows that millions of typosquatting registrations, often targeting less popular domains, foster a wide variety of abuse and regularly rely on shared and malicious advertisement networks; furthermore, accurate classification of typosquatting and prediction of malicious redirections can be achieved even in the face of user differentiation and cloaking. Supporting the detection of domain registration abuse, the thesis provides a framework to analyze how registration policies affect the utility of malicious domain name registrants.

Web typosquatting and malicious advertisements We find that previous research [40, 106] focused only on less than 5% of potential typosquatting domains that are targeting the most popular domains on the Internet. To address the first challenge and to understand the extent of typosquatting, we study millions of potential typosquatting domains [128] targeting Alexa's [1] top 1 million domain names and samples of .com domain names. We collect DNS, Whois and web content information about these potential typosquatting domains to understand the infrastructure supporting them. We find points of concentrations in DNS records and the registrars used, including infrastructure exclusive to typosquatting. Leveraging our observations, we develop an accurate classifier which can decide if a potential typosquatting domain name is truly typosquatting or if it was just accidentally close to the target domain. Using our classifier, we find that there are millions of true typosquatting domain names registered and that the number of typosquatting domain names is steadily increasing over time.

While our first study focuses on understanding the extent of typosquatting and the infrastructure supporting it, it does not aim at deeply understanding the malicious content users are exposed to. Our study compares typosquatting domains to existing blacklists, however blacklists lack both coverage and precision in finding malicious content on typosquatting domain names. Therefore in our complementary study, we focus on better understanding the frequency and type of malicious landing pages users are redirected to and the role typosquatting plays in malicious advertisement networks. To achieve these goals, we need to address the second and third challenges, as typosquatters might only present malicious content to certain types of users, and only if they do not suspect automation.

We develop an infrastructure that can help us understand how users are redirected to malicious pages. Our infrastructure allows us to emulate phone users and desktop users in order to understand user differentiation. We take steps to address cloaking based on the HTTP request header, the browser's properties, the IP address used and proxy detection. For example, we run measurements both with and without proxies or using one IP address versus using 240 IP addresses. Furthermore, to understand the role typosquatters play in

malicious advertisement networks, we compare them to illicit movie streaming sites, illicit online pharmacies and ad-based URL shortening services.

We find that typosquatting domains, ad-based URL shortening services, and copyright infringing websites often rely on the same traffic distribution systems to monetize traffic by sending users to the same illicit and malicious landing pages. Our analysis shows that these traffic sources up to 44 percent of the time use the same traffic brokers' domains. Additionally, malicious advertisement networks redirect users to the same kind of landing pages, and nearly half of the different types of malicious activities we find are present in all three of the typosquatting, copyrighting infringing, and ad-based URL shortening ecosystems. Examples of malice included technical support scams, deceptive surveys, deceptive downloads, and other scams. At the same time, certain types of abuse are prominent at only one traffic source. For example, copyright-infringing sites invoke users' social media activities without permission including tweets and shares. Ad-based URL shortening services advertise crypto-currency related scams. Typosquatting domains redirect to fake identity protection phishing sites. Our results indicate that these complex malicious ecosystems, similarly to advertisement networks, differentiate phone users, and attempt to block automated crawlers. Phone users are redirected to malicious landing pages desktop users never see, and desktop users are exposed to malicious content phone users would never experience. Visiting pages as five different types of users, we find 81 percent more malicious landing pages and 96 percent more suspicious landing pages, compared to visiting pages as only the one user who experienced the most malice.

Additionally, TDSs try to cloak their malicious activity leveraging IP address-based reputation and HTTP header fields. We find that using 240 IP addresses, we experienced more than twice as many malicious landing pages compared to using only one IP address. We discover that a daily blacklist of URLs (like Google's Safe Browsing list) is not just significantly delayed, as found by previous research [116], but it is not appropriate to describe the malice in these shady advertisement networks due to the dynamic nature of redirections. We provide a classifier that can be used with high precision to stop users from landing on malicious pages, relying only on features available at the time of redirection.

Email typosquatting Abuse of domain names is not confined to the Web, and subsequently studying typosquatting in the context of other applications is necessary to understand the threats users face from these illicit domain registrations. Our research is the first in-depth study on *email typosquatting* [126], in which miscreants could leverage typosquatting domain names to collect emails sent to the wrong address due to user typing mistakes.

We register 76 email typosquatting domains and collect data from these domains for more than seven months (June 4, 2016–January 15, 2017). Working in concert with our Internal Review Board (IRB), we design a protocol to process the emails we receive to determine the potential harm email typosquatting might inflict on users, as well as its potential benefits to attackers. Based on active data collection, and the examination of the whole ecosystem, we conclude that the profitability of a typosquatting domain depends on three main factors: the popularity of the target domain, the edit distance from the target domain, and the visual distance from the target domains. Among the emails received, we found users accidentally sending us emails containing highly sensitive personal data.

Additionally, we discover that a several actors already have the infrastructure necessary to collect private user emails in bulk from tens of thousands of typosquatting domain names. We also observe that some registrants own thousands of email typosquatting domains and that these domains support SMTP. Furthermore, some of the name servers (and registrars) used by tens of thousands of typosquatting domains appear to be cesspools, with a 5–10 higher typosquatting domain ratio than normal.

Extrapolating from our observations through regression analysis, we find that setting up the necessary infrastructure costs typosquatters only in the order of a couple of cents per email and that they can expect to receive hundreds of thousands of emails over a few months. However, by actively sending "honey emails" containing credentials, we discover, that even though a lot of these emails are accepted, we cannot evaluate by ethical means if these emails were used for malicious purposes.

Defense and mitigation During our research, we develop tools to detect typosquatting and to identify redirections that send users to malicious landing pages. However, detection-based blacklisting has four drawbacks. First, blacklists only cover a fraction of malicious and illicit domain names. These false negatives are partially introduced to avoid mistakenly blacklisting a good domain name. Second, despite their efforts, blacklists can also produce false positives. Third, blacklists can only protect users who use them, and subsequently, a malicious domain can remain valuable for criminals even after it is blacklisted. Finally, it takes time for domain names to appear on blacklists presenting a window of opportunity for miscreants to profit from these domain names. Some of these issues can be addressed by improving detection methods. However, we find that combining domain registration policies with detection could be crucial in making malicious domain registrations unprofitable.

As we are the first to focus on registration policies [127], our goal is to understand the limitations of different policy proposals. Studying the whole domain registration ecosystem, we strive to understand which ones of the candidate policies proposed would be effective against abusive domain registrations without having a significant negative impact on benign entities. We identify two domain registration behavior by miscreants that is substantially different from benign use. First, miscreants often need numerous domain names to evade blacklisting, whilst usually benign users only need a couple of domain names. Building on this difference in need, we can construct policies that would penalize the ownership of large amounts of domain names, contrarily to current practices that are rewarding users for bulk registrations. Second, miscreants register look-a-like and typosquatting domain names to fool users or to monetize typing mistakes. A characteristic feature of these squatting domain names is that they are similar to already existing domains. Leveraging this observation, registries and registrars could monitor, remove and harden potential squatting registrations depending on the certainty of abuse.

In addition to these patterns of abusive registration, policies aiming at making domain suspension and takedown efforts more effective would have a massive impact on the profitability of abusive domain registrations. As an example, Chachra et al. [43] shows that spam domain registration would become economically non-viable if these names were removed instead of just blacklisted. While millions of abusive and malicious domain registrations exist, they are rarely taken down or suspended. Policy strategies directly

impacting the extent and speed of takedowns and suspensions would be potent in making abusive registrations less profitable. As an alternative, we focus on incentivizing registrars and registries to remove malicious registrations. More specifically, we could increase or decrease the per-domain fee they pay based on the number of domains blacklisted from the ones registered with them, similar to how Dutch and Swedish Registries incentivize registrars to adopt DNSSEC [7].

We find that leveraging all three of the approaches together could benefit the domain registration ecosystem the most.

Future work While we take steps to better understand the extent of illicit typosquatting registrations in Chapter 3, the malice users are exposed to in malicious advertisement networks in Chapter 4, the threats of email typosquatting in Chapter 5 and how we can leverage registration policies to aid detection in Chapter 6, we still do not have a complete understanding of user differentiation in the ad ecosystem and how applications other than web and email might be affected. Studying these problems further can help us understand the effects of abusive registration and to protect users from harm.

Chapter 2

Background

Popularity attracts speculation, and typosquatting showcases this observation in the Internet ecosystem. Typosquatting is still one of the most common domain registration abuse even in the face of continuous efforts to diminish its impact. In this chapter, we present a general overview of the typosquatting ecosystem and intervention attempts providing background for the rest of the thesis.

2.1 Overview of Web Typosquatting Techniques and Monetization

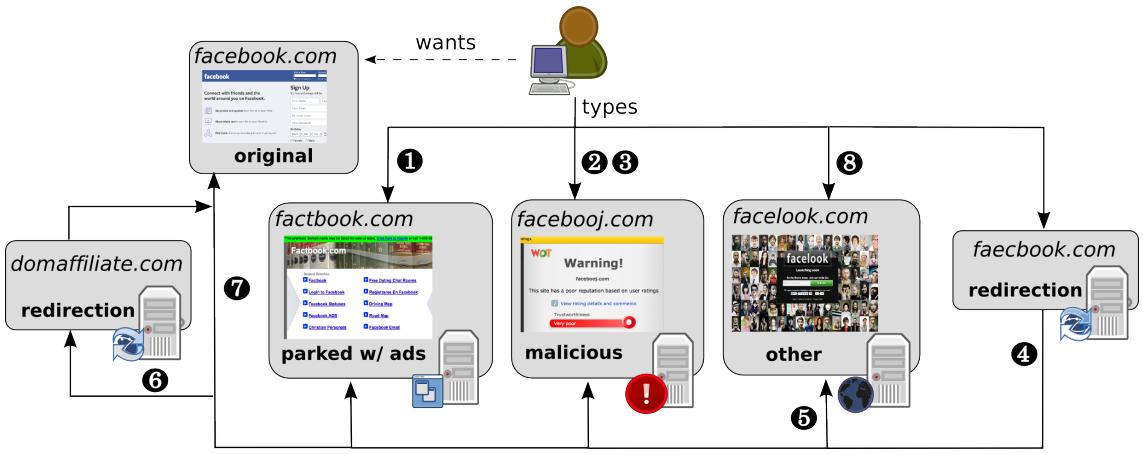


Figure 2.1: The typosquatting ecosystem with various monetization techniques.

Typosquatters register domain names that are similar to those used by other websites in hope of attracting traffic due to user mistakes. The most frequent occurrences of mistyping are those that involve a one-character distance, also called the Damerau-Levenshtein (DL) distance one, from the correct spelling both in free text [49] and in case of domain

names [40].[1] In this thesis, we focus on typosquatting domains of Damerau-Levenshtein distance one (DL-1) that are generated using the most common operations: addition, deletion, substitution of one character, transposition of neighboring characters [49]. We extend this to include deletion of the period before the "www" commonly prepended to web server domain names [106]. We note that a special case of DL-1, called fat finger distance (FF distance), is considered when the mistyping occurs with letters that are adjacent on a US English keyboard. The rationale of this metric is that users are more likely to mistype letters in close proximity.

Typosquatters use various techniques to monetize their domain name registrations. The typosquatting domain can be *parked* and serve third-party advertisements to monetize the incoming traffic (❶ on Figure 2.1). The domain can also be set up to impersonate the intended domain for instance to host a phishing page [131] (❷), serve malware (❸), or perpetrate some other scam on the user [50, 142]. Many monetization techniques can also involve redirection to another domain (❹), the *landing domain*, that might employ the previously mentioned techniques. Speculators can also redirect visitors to *competitor domains* (❺) causing a direct loss to the owner of the original domain. Conversely, the typodomain owner can redirect traffic to the intended site, and monetize this traffic via *affiliate marketing* (❻). The original domain owner can also perform *defensive registrations* of typos for their main domain name and set up the redirections themselves (❼). Finally, in some cases, the typo domain owner can serve content that is unrelated to the original domain (❽).

2.2 Typosquatters Leveraging Traffic Distribution Systems

To understand how an illegal but seemingly harmless typosquatting domain registration can lead to malicious content, next, we explain in-depth how traffic monetization through advertisement (❶) and domain redirection (❹) works.

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. [112] for an extensive description of the advertising ecosystem.

Clicks require active user participation. A much more aggressive technique for typosquatters is to instead *automatically* redirect users to a target destination website – in such a context, ad publishers are compensated through pay-per-redirect (PPR). For example, when

[1] Although some researchers have found that for longer original domains a small number of typosquatting domain names with larger DL distances exist [106].

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 [33, 110].

Using terminology from the literature [35, 92], 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 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, such as typosquatters, are agnostic to the techniques TDSs use to bring traffic to their websites. Indeed, these malicious operators are merely customers of the traffic distribution systems. 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 program (PUP).

Early research of traffic distribution systems has focused on malicious advertising in Alexa top domains [95, 96, 146]. 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 [90, 106, 110, 114], 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 [134] that focused on investigating traffic broker domains to find more malicious destination pages. Conversely, our goal is to study and compare traffic sources, quantify the effects of user differentiation and cloaking techniques.

2.2.1 Other Illicit Traffic Sources Relying on TDSs

To gain a clear picture of the malicious advertisement ecosystem used by typosquatters, we study three other traffic sources known to rely on TDSs: 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 and their suspected similarity to typosquatting pages.

URL shortening services transform complex URLs with user-friendly shorter variants. Nikiforakis et al. [110] 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 download, 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 watching or trying to load movies. Researchers have focused on the infrastructure supporting the sharing of pirated content [63], but have not investigated which abusive pages users are redirected to. Closer to our research, Rafique et al. [114] 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 [90, 92, 93, 98, 100, 139] have investigated how unlicensed online pharmacies acquire traffic, through email spam or search poisoning. They did find early evidence of cloaking (e.g., HTTP header and cookie-based). Very 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.

2.3 Overview of Anti-cloaking Techniques Used in Related Work

Typosquatters—and the TDS operators they rely on—often engage in "cloaking" to hide their malicious activity from security researchers and law enforcement. In trying to determine how the literature addresses cloaking, we surveyed 22 measurement papers [33, 40, 46, 63, 78, 79, 81, 83, 90, 92, 95, 96, 104, 106, 110, 114, 121, 128, 133, 134, 140, 146] that engage in active crawling of Web content from TDSs, illicit traffic source or destination pages. We search these papers for evidence of what steps the authors took to either address or study cloaking. There are two main limitations to this analysis of related work. First, we trust what the authors say was done correctly and is true. Second, we only consider what is written in these papers. Thus, if the authors do not mention steps they took to address cloaking, then we consider it not being done.

Table 2.1: Comparison of active measurement papers based on the anti-cloaking steps considered by the authors.

	Active Web Measurement Studies			
Anti-cloaking measures	**2011 - 2016**	**2016 and after**	**All**	**ODIN**
User-agent field-based	4	9	13	X
HTTP header field	3	2	5	X
Browser fingerprinting		2	2	X
IP address type		1	1	X
IP rate limitation	1		1	X
Proxy detection			0	X
Basic Crawler	2	1	3	
No cloaking detection	8	1	9	
Total	**12**	**10**	**22**	**1**

We find six main anti-cloaking methods in the papers we study, including changing the user-agent, setting an HTTP header field, mitigating browser fingerprinting, considering the IP address type used, addressing IP rate limitation and avoiding proxy detection. Table 2.1 provides an overview of how frequently the aforementioned anti-cloaking techniques were

used in related work, and how they compare to our data analysis and collection platform ODIN (Observatory of Dynamic Illicit ad Networks) that we will introduce in Chapter 4. It is easy to detect simple crawlers that are not able to handle cookies or cannot execute JavaScript code. In table 2.1 we noted such crawlers as "Basic crawler" (if a paper do not mention the crawler used, then we assumed it was not a basic crawler).

We summarize our results in Table 2.1. With the exception of Wang et al. [139], 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. [139]) 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 [78, 81, 134] combine multiple defenses. ODIN combines all of these techniques to mitigate cloaking attempts. As online criminals evolve, we see researchers adopting to it and developing more sophisticated measurement systems addressing cloaking.

2.4 Web and Email Typosquatting Related Work

Most typosquatting papers have focused on web typosquatting, which targets users who make a mistake while typing an URL in their browser. In 2003, Edelman undertook the first case study of one typosquatter who registered, at the time, thousands of domains [51]. Subsequently, a number of efforts [39, 40, 44, 140] proposed methods to detect typosquatting domains targeting popular websites, as ranked by the Alexa service [1], and to differentiate legitimate domains from typosquatting domains [128]. Some of these studies suggest that monetization is achieved through domain parking – the act of monetizing otherwise empty web pages with advertisements.

Moore and Edelman [106] discussed monetization of typosquatting, and showed that miscreants might be relying on Google AdWords to select which typosquatting domains to register. Along the same lines, Agten et al. [33] provided a longitudinal study of monetization strategies of typosquatting targeting Alexa's top 500 domains. More recently, Khan et al. [80] quantified the harm of typosquatting caused to *users*, and found that a typical user loses 1.3 seconds on average when visiting a typosquatting domain.

Different from this entire body of work, we broaden the scope of investigation to email typosquatting, which, from a technical standpoint shares many similarities with web typosquatting (low barrier to entry, low sophistication), but whose monetization strategies ought to be completely different—whereas web typosquatting primarily profits from advertisements, through "parking pages [137]," email typosquatting is likely to benefit from capturing credentials or sensitive information.

To the best of our knowledge, only one white paper looked at domain typosquatting beyond web typosquatting [55]. The authors registered domains that were similar to existing *subdomains*, with the exception of a missing dot—e.g., `caibm.com` as opposed to `ca.ibm.com`. They claim to have collected 120,000 mis-directed emails over six months,

but do not report on the number of domains they registered, and do not discuss whether they filtered out spam. Our work attempts to provide a far more detailed picture of email typosquatting in the wild; in particular, we will observe that filtering out spam email is a crucial step in providing credible measurements of the attack's impact. We also investigate whether typosquatters *act upon* emails they receive.

2.5 The Domain Registration Ecosystem

This section provides the background necessary to understand an array of available policy tools complementing detection-based approaches. First, we provide a high-level overview of the domain registration ecosystem to examine the relationship between online criminal activities and domain name registration. We then turn to a discussion of the "WHOIS debate," which is germane to the problem at hand.

As the Internet grew from a few hosts to millions of domains, the Domain Name System, in charge of mapping IP addresses to human-memorable strings, evolved from a simple translation file ("HOSTS.TXT," back in the days of the ARPANET) to one of the largest, if not the largest, hierarchical distributed systems in existence. Internet domain names have become so important that they are frequently interchangeable with brands, and it is not uncommon for valuable domain names to be resold for millions of dollars [143].

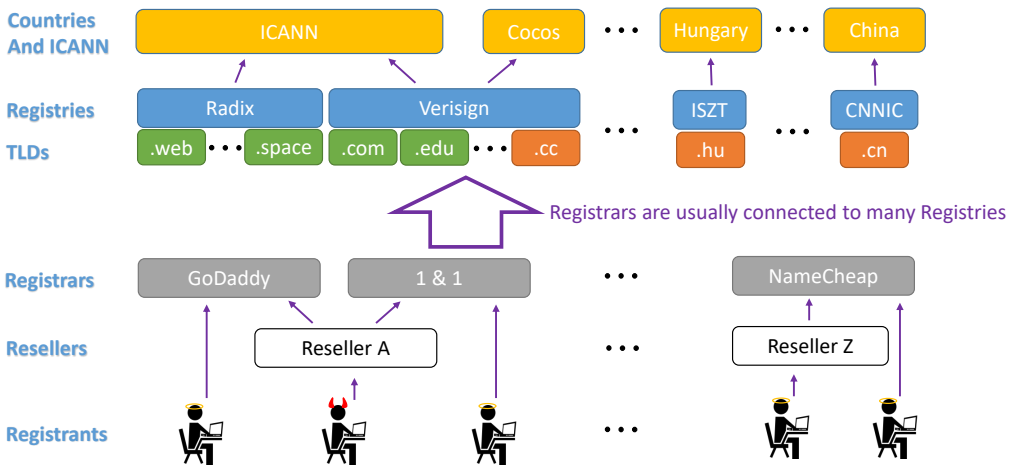


Figure 2.2: A simplified view of the domain registration ecosystem. gTLDs are in green; ccTLD in orange. Purple arrows denote administrative ownership dependencies—i.e., how money flows from registrants to domain administrators.

Figure 2.2 depicts a simplified view of the most important entities in the domain registration ecosystem. ICANN was created to manage the Internet's numerical addresses and domain names. Individual top-level domains (TLDs) are operated by *registries*. There are two kinds of TLDs: generic TLDs (gTLDs) and country-code TLDs (ccTLDs). Registries wishing to operate gTLDs need to be approved and follow ICANN's policies. As an example

in Figure 2.2, Radix has an agreement with ICANN to operate gTLDs such as `.fun` and `.space`. [74]. On the other hand, registries operating ccTLDs have varying levels of cooperation with ICANN: agreements are handled on a purely voluntary basis. For example, the Hungarian registry ISZT has an agreement with ICANN about the `.hu` ccTLD, but the Chinese registry CNNIC has no such agreement. Furthermore, some registries (such as Verisign) can operate multiple gTLDs and ccTLDs, where they need agreements with ICANN and multiple countries at the same time.

Registrars are the entities selling domain names to *registrants* (users registering domain names). *Registered domains* are the part of fully qualified domain names (FQDNs) that registrants can buy.

Besides the myriad registrants with whom registrars have agreements, registrars usually have an agreement with registries to be able to sell their domain names. To directly access gTLDs, registrars need to be accredited by ICANN. Some domain resellers, usually hosting companies, further act as middlemen, selling domains to users, and buying them from registrars.

The purple arrows in Figure 2.2 depict how money is distributed when a user acquires a domain name. For example, when a user buys `example.com` at reseller A, part of the payment is divided between reseller A, 1&1, Verisign and ICANN. If another user buys `example.cc` at Godaddy, then GoDaddy, Verisign and the Cocos Island government all profit from this transaction.

2.5.1 A Survey of Abusive Domain Registrations

Besides benign registrants, the rise in popularity of the Internet unfortunately attracted domain speculators and miscreants trying to profit from the relative ease of registering domains. Speculators buy domain names for cheap, in hope to profit from users accidentally visiting their sites, or hoping that they can resell some of their domains for a large profit margin. While domain speculation is an unintended byproduct of Internet domain registration policies, it remains legal as long as speculators are not infringing on existing trademarks or supporting criminal activities.

Understanding differences in the registration patterns and behavior between malicious and benign users is important to design a policy which affect the former but not the latter. Table 2.2 lists the major categories of online frauds, and summarizes how domain registration plays in the furtherance of each fraudulent activity. We focus on two main registration patterns that we can leverage. First, miscreants frequently need to register a large number of domain names to conduct their activities. Second, some domains have distinctive lexical features related to a target domain.

Miscreants use domain names for four main reasons. First, criminals need to evade blacklisting of their domain names and IP addresses, which often leads them to register a large number of domains. Second, they use domains for accounting and business agility (e.g., traffic distribution systems [95]) when offering their services to other miscreants. Third, crooks frequently use domain names to fool users into believing they are representing an official brand or company. Finally, criminals can register specially crafted domain names to siphon traffic from legitimate domains.

Table 2.2: Malicious domain name registration patterns

	High demand for domains	Distinctive lexical features	Role of domains	Are domains substitutable?
Spamming	yes	no	Evade BL	easy to BL
Generic Phishing / Scams	yes	usually	Evade BL Fool users	easy to BL
Targeted Phishing / Scams	no	usually	Fool users	less effective
Botnets	yes	no	Evade BL	possible
Malvertisement	yes	no	Evade BL	easy to BL
Illegal pharmacies	yes	no	Evade BL	easy to BL
Drive-by-downloads	yes	no	Evade BL	easy to BL
Illegal streaming	yes	no	Evade BL	easy to BL
Squatting variants	no	yes	Siphon trf. Fool users	no

Domain squatting, typosquatting, and variants. In domain squatting and its variants, profit stems from the domain name itself. Domain squatting (also known as cybersquatting) [101] is the act of registering domain names of brand names in hope to sell them to the brand owners for profit. More notorious domain squatters used to redirect visitors to adult pages to extort money from brand owners [54, 99].

Typosquatters, as discussed earlier, register domain names lexically close to a target domain to profit from users mistyping the target domain name [128]. Soundsquatting domains are domains that sound similar to the target domain [109]. All these squatting techniques are illegal in the U.S., where the Anti-cybersquatting Consumer Protection Act (15 USC x1125(d)) can be used to protect brand owners. Internationally, ICANN provides a Uniform Domain-Name Dispute-Resolution Policy (UDRP) to mediate domain registration disputes.

Typosquatting and combosquatting domains are also often used for phishing and scam attacks [83, 104]. Combosquatting domain names contain the name of a brand to make the fraudulent domain look like a domain owned by this brand (e.g., `famousbrand-security.com`).

Domain squatting differs from the majority of other online criminal activities, since here domain names are the means to an end: Domain squatters can be driven out of business entirely by targeting their domain registrations.

Spamming is defined as unsolicited bulk messaging. The most common form of spamming is email spam, but spammers often target blog comments, tweets, and other messaging systems. Spammers especially need a lot of domain names to evade blacklisting of their

email address domains and their spamvertised domains. While spammers could choose to use IP addresses directly instead of domain names, it would raise suspicion leading to blacklisting, since IP addresses are extremely rare as part of URLs in legitimate emails.

Phishing and scamming targeting users. Phishing emails and webpages try to trick users into sharing their personal information with miscreants. Miscreants collect this personal information to sell it to other online criminals who can monetize this information. This personal information includes usernames, passwords, addresses, SSN numbers, identification documents, credit card numbers and other financial information.

Scam operations are very similar to phishing, but instead of tricking users into sharing personal information, scammers try to directly extort money from users.

General phishing and scam attacks try to reach as many users as possible and thus they exhibit similar patterns of domain name usage as spammers to avoid blacklisting.

However, spear phishing attacks and targeted scam attacks use only a couple of carefully selected domain names for a single attack campaign, making registration policies ineffective.

As discussed, typosquatting and combosquatting domains are often used for phishing and scams. Alternatively to these domains, criminals could use an IP or a domain unrelated to the targeted brand name and obfuscate the URL sent in the email or shown in the browser.[2] Using IPs would decrease the success of these attacks just like in the case of spam. Luckily, researchers have created detection systems, such as PhisDef [88], which made URL obfuscation outdated.

Botnets are a collection of infected users' machines controlled by botmasters. Botmasters rent out these machines to be used for a plethora of other illicit online activities.

Botnet operators use techniques called fastflux and doubleflux to hide the location of their command and control centers (C&C). These techniques involve changing the domain names used and changing the NS and A records of these domain names frequently.

Botnet operators are using many other approaches that do not involve domain names to hide their location. These approaches include hard coding IP addresses (often encrypted and obfuscated in binary) or using legitimate cloud service providers' servers to host their C&C. However, these approaches have significant drawback compared to using domain names. If a piece of malware contains hard-coded IPs and is reverse engineered, then all samples of the malware can be deactivated. If a piece of malware is using a cloud service provider, then either the cloud service provider will be blacklisted after a while, or this provider will clean up the malicious activity on their servers. Thus, botnet operators keep enjoying the flexibility and simplicity provided by the domain name system for a low cost.

Malvertisement. Malvertisers post malicious advertisements on benign ad networks to infect, phish or scam users for profit.

Ad network owners such as Google and Facebook continuously try to detect and block malicious advertisements; facing constant blocks, malvertisers thus need a large number of domain names to conceal their activity.

[2]The goal of URL obfuscation is to trick users into believing that they are visiting a known brand's or company's website.

Illegal online pharmacies and other counterfeit stores frequently rely on domain names to provide a veneer of legitimacy to their businesses, making them particularly vulnerable to blacklisting.

Drive-by-downloads try to infect the victim's browser or computer upon visiting a webpage.

Domains hosting drive-by-download pages are frequently blacklisted (e.g., by Google Safe Browsing and others) as they try to infect users' machines. Drive-by-download pages are also using redirection chains and domain names (Traffic Distribution Systems) to evade blacklisting.

Copyright infringement. When pirated content is shared, online criminals hope to profit from users visiting their website, either through extensive advertisement, or, worse, by infecting user machines or running different scams or phishing schemes [114].

Pages offering pirated content are often blacklisted and taken down. Hence the operators of these pages need domain names to evade blacklisting and are affected similarly to spammers.

In general, if online criminals want user traffic, then they need to either advertise themselves via spamming malvertisement, or malicious search-engine optimization; or siphon traffic via a squatting technique. A common property of these methods is these activities are much easier to block when the bad actors do not rely on domain names, but, e.g., on IP addresses. The only other way for criminals to reach users without domain names is to penetrate a legitimate service's server and carry out the attack on the users of the compromised service.

2.5.2 The WHOIS Debate

The domain registration database (WHOIS) provides an important tool to fight online crime, but the collection of user data also raises privacy concerns. In this section, we summarize how his tension sparked a decade-long debate concerning the WHOIS system and how Chapter 6 builds on it.

Brief history of congressional hearings. Since 1998, the U.S. Congress has held more than twenty hearings about ICANN and policies regarding the domain name system [31]. At the first hearing participants discussed the transfer of management of the domain name system to ICANN. Later on, some of these congressional hearings turned into a clash between different stakeholders [28, 30]. On the one hand, the law enforcement and intellectual property communities argued for easier access to WHOIS records, enforcement of accurate WHOIS information and potentially penalizing registrars for allowing malicious registrations. On the other hand, civil right groups would have liked to restrict access to WHOIS information to protect registrants' privacy, to protect political activists, and to protect registrants from spammers and phishing. ICANN's Security and Stability Advisory Committee (SSAC) established in their "Blind men and an elephant" report [67] the need for a better understanding of why WHOIS is needed, what registration information is needed, and who should be able to access certain information.

The first proposed solution. To solve the tension between different stakeholders Operational Point Of Contact (OPOC) was proposed by the ICANN community [65]. The goal of OPOC was to provide a third-party point of contact for registrants and thus shield their personal information from online criminals and provide them a degree of privacy. This proposal achieved a certain balance between privacy and usability. However, the OPOC proposal became quite complex and different stakeholders could not achieve consensus. Therefore ICANN's Expert Working Group decided not to pursue the OPOC solution and instead initiated studies to better understand WHOIS misuse.

The importance of WHOIS. Maintaining accurate WHOIS data is important for several reasons as noted by SSAC [69] and stakeholders [28, 30]. This data is used to pursue violations of intellectual property such as copyright and trademark infringement. Law enforcement agencies frequently use WHOIS to investigate online crime. Security researchers use WHOIS to understand domain ownership and to contact domain owners to clean up compromised websites. Finally, WHOIS can be used by consumers to look up who they are conducting business with on a given domain.

The problems with open WHOIS access. The drawback of free and unlimited access to WHOIS information is that it can be used by spammers and for more elaborate scams or phishing schemes [28, 30]. This was confirmed by Leontiadis and Christin [89], when they found that WHOIS information is leveraged for spamming the registrant's email address, postal addresses, and phone numbers. Furthermore, some registrants might not want to have their personal data available to the public due to privacy considerations; for instance, activists may not want their identities linked to their websites. Inaccuracies can also occur because some registrants mistype their information for the WHOIS database. Finally, malicious registrants do not want to have their real personal data in the WHOIS database to evade law enforcement and legal investigations.

WHOIS privacy and proxy services. All these lead to a significant number of registrants either using WHOIS privacy services or entering fake data as their WHOIS records. Clayton et al. [47] studied in depth the use of WHOIS privacy and proxy services. They found that both benign and malicious registrants often use WHOIS privacy services.[3] In general, registrants that do not use privacy services often cannot be reached via the phone number provided, and, unsurprisingly, malicious registrants can almost never be reached via phone. The Fraudulent Online Identity Sanctions Act (FOISA) was specially created to deter malicious registrants from providing fake WHOIS information [29, 32]. The act doubles the maximum imprisonment if false WHOIS information was provided while committing a felony offense.[4]

ICANN on WHOIS data validation. More recently ICANN's Security and Stability Advisory Committee (SSAC) published a report discussing options for registration data validation [69]. The authors of the document focused on the reasons for WHOIS inaccuracy and the taxonomy of validation. Their taxonomy consists of three levels of validation: syntactic, operational, and identity validations. Syntactic validation refers to making sure the format of the registrant's data is correct. Operational validation means that the contact

[3]Malicious registrants use privacy services more often than benign registrants.

[4]At most, FOISA increases maximum imprisonment by seven years.

data provided actually works, for example, emails are received at the provided email address. The goal of identity validation refers to checking if the data provided corresponds to the real world identity of the registrant.

As of 2013, ICANN requires registrars to perform syntactic and operational validation of registrants' data [68, 75]. However even as of today registration data is often not valid syntactically or operationally [70]. The focus of our research is on identity verification and we assume that syntactic and operational validation is relatively easy and cheap to do well.

ICANN's current proposed solution. Currently, ICANN is working on a new Registration Directory Service (RDS) [70] that would replace WHOIS for new gTLDs. This proposal is still at an early stage where many questions are still under evaluation [76]. What data should be asked from registrants? Who should be able to access what registration data and on what scale? How should different data fields be validated? One proposal under evaluation is to offer partially public and partially gated access (tiered access) to different entities. Another proposal is to use pre-validated identities at registration time maintained by validators.

Connection to our work. Our work has both a different goal and approach compared to the discussion and research around the WHOIS service. Our goal is to systematically find a composition of policy tools that can hurt malicious registrants but not benign registrants. Contrarily, the WHOIS debate is focused on how to provide accurate registration data for security researchers and at the same time provide some privacy guarantees for registrants.

For our proposals, we assume the existence of a registration data service which solves the tensions in the WHOIS debate by providing tiered access and at least operation level validation of data, while in practice this might be challenging to achieve. On the other hand, we explore questions such as how we can provide privacy for sensitive registrants and what are the trade-offs of identity validation. A couple of our proposed policies are closely related to ICANN's new RDS. Related, we discuss the benefits and costs of different identity validation approaches ranging from no identity validation to strict identity validation. More details can be found in section 6.2.

2.6 Defenses and Policy Interventions

In this section, we discuss available defense techniques and analysis to thwart domain registration abuse. There are three main approaches defenders can utilize: reactive detection of malicious use, prediction of future misuse and domain registration policies. All of these approaches have drawbacks and advantages.

Detection. Reactive detection of abuse is the most common approach out of the three as deployment is relatively easy and flexible (contrary to registration policies), and researchers had better success at keeping false positive classification low (compared to prediction). Detection suffers from various problems. First, nearly all detection algorithms suffer from false positives or negatives. Second, the action taken upon detection is crucial. In the case of blacklisting, detection is only useful where it is deployed. Therefore it does not significantly affect the utility of many types of abuse. However, if abusive domains are

taken down, then all users benefit from the detection. Furthermore, detection is reactive and thus delayed providing cybercriminals a window of opportunity for their enterprise.

There have been some efforts to provide technical tools to mitigate typosquatting, notably the Microsoft Strider Typopatrol system which protects trademarks and childrens' sites [140]. At the user level, the OpenDNS has a typo correction feature which corrects major TLD misspellings [111] and the Mozilla URLFixer Firefox plugin [25] can suggest corrections to typed URLs. A common property of these solutions is that they only cover a relatively small set of typos, typically those that target the most popular domain names. Our solution discussed in Chapter 3 is based on an extensive set of investigated domain names and hence provides significantly better coverage to detect typosquatting. Moreover, our extended set of detection features allows for more accurate detection of typosquatting than solutions in previous work.

Prediction. Prediction of future domain abuse is hard, and most approaches have a too high rate of false positives [60, 119] to be used for direct blacklisting. As an alternative, they can be used to warn users or to keep an eye on suspicious domains for future abuse. Researchers have been working on building domain reputation systems with two goals in mind: 1) to decrease the time it takes to blacklist a domain name and 2) to increase both the precision and recall of these systems. Antonakakis et al. [37] built one of the first reputation systems for DNS which leverages the characteristics of domain usage specific to online crime. Their system was able to detect malicious usage weeks earlier than traditional blacklists. Hao et al. [60] showed how registration time features can be leveraged to proactively blacklist domain names further decreasing the time to blacklist domains.

Policy intervention. Domain registration policies can suffer from similar problems as detection and prediction approaches. Too strict policies might inflict collateral damage on benign users which is analogous to false positives. On the other hand, too lax policies might not deter crime analogous to false negatives. Coverage issues also appear if the policies are not adopted globally, allowing miscreants to migrate their infrastructure to locations under different governance.

Blacklisting approaches are made harder by the lack of identity verification and the abundance of cheap domain registration options for users. ICANN recently started its new gTLD program to increase the available options to users for domain name registrations. Halvorson et al. [58, 59] found that new gTLDs have a significantly higher rate of speculative and abusive registrations compared to other TLDs. "Taken together, our findings suggest that new gTLDs, while accruing significant revenue for registrars, have yet to provide value to the Internet community in the same way as legacy TLDs"[58]

Liu et al. [97] analyzed the effects of intervention at a single registry, CNNIC in China. They found that it will help to push abuse from that registry's TLD, .cn but it will not affect criminal endeavors in the long-term. Chachra et al. [43] found that 88% of spam domains are blacklisted in less than two days and thus their revenue is effectively limited. However, blacklisted spam domains continue to monetize because of the high demand for advertised goods, non-universal blacklisting, and delay in deployment. Their economic analysis has shown that the per-domain cost would need to be at least a $100 to make these domain registrations unprofitable. At the same time if domains were to be shut

down totally instead of blacklisted less than $3 per-domain cost would be sufficient to deter these registrations. Korczynski et al. [85] studied metrics to characterize abuse at TLDs, they found that the size of the TLD and pricing are positively correlated with abuse and DNSSEC deployment is negatively correlated with abuse. Additionally, they found that TLDs with restricted registration policies are less frequently used for phishing.

Research so far studied how to blacklist domain names more effectively, what affects abuse in TLDs or studied the effects of a couple of registration policy intervention attempts that occurred in the past. Chapter 6 is different in that we systematically study how multiple potential registration policy strategies would affect the most important entities in the domain registration ecosystem. By doing this we hope to pinpoint directions that are worthwhile to further explore in the grand battle against online criminals.

Policy intervention is more effective when targeting the registration process either at a national scale for specific TLDs or on a registrar level [97]. One can also mount an effective defense by targeting the monetization infrastructure [94, 97]. Unfortunately, the agility of domain speculators in registering new domains and the difficulty of determining their ill intent makes this a difficult prospect.

Legal intervention. Typosquatting exists within a legal and moral gray area; consequently, intervention has traditionally been weak to reduce the effect of typosquatting. ICANN provides the Uniform Domain-Name Dispute-Resolution Policy (UDRP) to mediate domain registration disputes for a relatively small filing fee. Unfortunately, cheap domain registration allows for mass typo-domain registrations and this gives a significant advantage to speculators. Against mass registrations of typo-domains UDRP mitigation becomes infeasible. Companies have initiated legal procedures in cases where cybersquatting and trademark infringement was applicable (see for example [130] on a recent court order against `twtter.com` and `wikapedia.com`, and a more recent court order against typosquatters of `facebook.com` [129]). The Anti-cybersquatting Consumer Protection Act (ACPA) (15 USC §1125(d)) offers legal protection to push such cases to court.

Chapter 3

The Long "Taile" of Typosquatting Domain Names

In this chapter,[1] we start our research by exploring the extent of typosquatting and studying the main methods leveraged by typosquatters to profit from user traffic. While previous research has focused on typosquatting domains which target popular websites, speculators also appear to be typosquatting on the "long tail" of the popularity distribution: millions of registered domain names appear to be potential typos of other site names, and only 6.8% target the 10,000 most popular `.com` domains.

We investigate the entire typosquatting distribution targeting .com domains as it can give a more complete understanding of this phenomenon. Our methodology helps us to significantly improve upon existing solutions in identifying typosquatting domains and their monetization strategies, especially for less popular targets. We find that about half of the possible typo domains identified by lexical analysis are truly typo domains. From our zone file analysis, we estimate that 20% of the total number of `.com` domain registrations are true typo domains and their number is increasing with the expansion of the `.com` domain space. This large number of typo registrations motivates us to review intervention attempts and implement efficient user-side mitigation tools to diminish the financial benefit of typosquatting to miscreants.

3.1 Introduction

Thousands of new domain names are registered daily that at first glance do not have completely legitimate uses: some contain random characters (possibly used by miscreants [94]), are a composite of two completely unrelated words (possibly used in spam [53]), contain keywords of highly-visible recent events (ex. `hillaryclingon.com` for political phishing in 2008 [117]) or are similar to other, typically well-known, domain names (ex. `twtter.com` [111, 130]). Domain purchasers use this final technique, often called "typosquatting," to capitalize on other domain names' popularity and user mistakes to drive traffic to their websites.

[1]This chapter is primarily based on our paper published at the 2014 Usenix Security Symposium [128]

Many old and new domain names alike do not ever show up in search engines, spam traps, or malicious URL blacklists, yet still maintain a web server hosting some form of content. However, maintaining the domain registration, DNS, and web server expends resources, even if these domain registrations do not serve an obvious purpose. Investigating the purpose of domain registrations in the "long tail" of the popularity distribution can help us better understand these enterprises and their relationship to speculative and malicious online activities. In this paper, we specifically consider the hypothesis that typosquatting is a reason for many of these registrations, and scrutinize different methods for committing malice or monetizing this behavior.

In the Internet economy, monetizing on user intent has been a very profitable business strategy: search display advertising is effective because relevant ads can be shown based on user search queries. DNS is similar, as domain registrations provide ample opportunities for monetization through direct user navigation rather than search. Domain name front running, domain tasting and typosquatting domain names can all monetize this phenomenon. [2] [48] According to [66], domain tasting was nearly eliminated in the generic TLDs by the 2009 policy changes by ICANN. In addition, [48] reports that the anecdotes about domain name front running by major registrars do not seem to hold. But typosquatting, the most prevalent speculative domain name registration behavior to date, continues apace.

Typosquatting wastes users' time and no doubt annoys them as well. As we show in Section 3.4.5, less than two percent of all domains we identify as "typo domains" redirect the user to the targeted domain, and the lion's share instead serve advertisements which previous research has shown to be profitable. [52, 106] These ad-filled pages give no clear indication to the user that they have typed the domain incorrectly; without a descriptive error, the user may abandon their task rather than double check their spelling. By monetizing these pages with advertisements, the typosquatter does a disservice both to the user and the victim web site. Protecting users from typosquatters can lessen the damage as well as disincentivize typosquatting by decreasing the squatters' profits.

If a typosquatter hosts a site that impersonates the legitimate brandholder it is certainly malicious and in some jurisdictions illegal. Such overt violations have been mitigated via legislation in the US and policy by ICANN [51, 64, 125]. For example, Facebook recently extracted a $2.8 million judgement against typosquatters impersonating their website; this successful litigation should serve as a strong deterrent against this form of malicious typosquatting against entities with the resources to litigate [57]. Several reports by commercial security teams have cited typosquatting domains' use in malicious campaigns for quiz scams [38], spam survey sites [142], in an SMS micro-payment scam [50], offering deceptive downloads or serving adult content [103], or in a bait-and-switch scam offering illegal music downloads [118]. However, until this paper, evidence regarding the extent of malicious typosquatting problems has not been available.

[2] *Domain name front running* is when registrars register domains that users have been looking for in order to monetize on their registration potential. *Domain tasting* is speculative behavior abusing the five-day grace period after domain registrations in some TLDs. This liberal registration policy gave refunds within a few days if the registrant wanted, however this policy resulted in short domain registrations en masse. ICANN has since changed policy, limiting the behavior [48, 66].

Typosquatting has been studied in depth in related work. In his first paper, Edelman points to the typosquatting phenomenon and discusses possible incentives for both squatters and defenders [51]. Wang *et al.* include a typo-patrol service in their Strider security framework that focuses on generating typo domains for popular domains and protect visitors from offending content [140]. Moore and Edelman revisit the problem in [106] pursuing a more thorough study of the original thesis of Edelman. They explore various monetization methods and suggest intervention options. They pessimistically conclude that the best intervention options are hampered by misaligned incentives of the participants. Banerjee *et al.* [40] make another attempt to design a typosquatting categorization tool. Their method works well for a small set of sample domain names. These analyses have focused on active measurement of typosquatting sites which target the most popular domains – considering no more than 3,264 unique `.com` domain names. However, we find that no more than 4.9% of all lexicographically similar name registrations target these popular domains. While typos for the most popular domains likely account for a significant amount of typo traffic, it is unclear whether the long tail also supports a significant amount of typo traffic.

Here we present a systematic study of domain name registrations focusing on typosquatting perpetrated against the long tail of the popularity distribution. We design a set of algorithms that can effectively identify typosquatting domains and categorize the monetization method of its owner. We also design and implement tools to improve user experience by allowing them to reach their intended destination. Although various user tools exist in the wild, most are inaccurate and focus only on a limited set of targeted domains. Our typo identification algorithms combined with the user protection tools provide improved protection against being misled by typosquatting, even when it is perpetrated against less popular sites.

Section 2 provides background on typosquatting and the most common tricks used by typosquatters. Section 3.2 presents our data collection methodology and describes our typo categorization framework. Section 3.4 presents a characterization of the extent, purpose, trends, and malice involved in the perpetration of typosquatting. We present mitigation tools and intervention options in Section 3.5. Section 3.6 concludes.

3.2 Methodology

This section presents our data collection and domain categorization framework in detail as illustrated it in Figure 3.1.

Terminology. Throughout this paper, we will refer to domains available for direct registration under a public suffix as *registered domains*, for instance *example.com* or *example.co.uk*. Generated typo domains, or *gtypos*, are domain names which are lexically similar (e.g. at DL-1) to some set of target domains. Candidate typo domains, or *ctypos*, are the subset of registered domains within the *gtypo* set which have been registered. Below we describe both how we select the target set and how we generate the gtypo set.

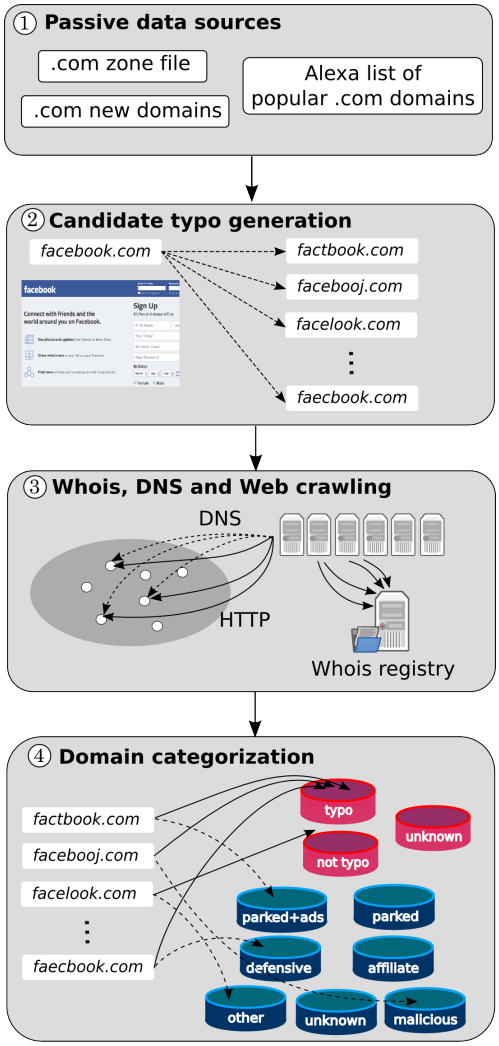


Figure 3.1: The data collection and typo categorization framework. The framework uses (①) large domain lists (zone file, Alexa popular domains list), (②) derives candidate typos based on lexical features and registration data in the zone file, (③) acquires additional information using active crawlers (Whois, DNS, Web), and finally (④) decides about typo domains and assigns them into typosquatting categories.

3.2.1 Data sources and scope

`.com` zone file. We leverage a variety of data sources to infer the prevalence of typosquatting in domain registrations. Our primary source is the `.com` zone file, which contains records of every domain registered within that TLD. As a popular generic domain name, the `.com` zone file contains millions of registered domain names `.com` and is available to researchers making it an ideal candidate for a representative investigation of typosquatting. Our comprehensive study is based on the March 15, 2013 version of the zone file provided by Verisign Inc containing approximately 106 million domain names. For trend analysis we collected the daily newly added and deleted domains from the zone file from October 01, 2012 to February 20, 2014.

Alexa list. The Alexa list of the top 1 million sites from March 15, 2013 serves as a benchmark for popularity [1], out of which 523,960 domains belong to the `.com` TLD, with 488,113 unique registered domains five characters long or more. For our study, we split the Alexa list into three categories: *Alexa top* containing domains ranked higher than 10,000, *Alexa mid* containing domains ranked 10,000-250,000, and *Alexa tail* containing the remaining `.com` domains ranked below 250,000. While Alexa cautions that rankings below 100,000 are not statistically significant, we are not concerned with exact comparative ranking or traffic counts for these domains but consider the Alexa list rather as a rough indicator of popularity. We also collected the Alexa top 1 million for the October 01, 2012 to February 20, 2014 period for trend analysis.

Domain blacklists. To shed light on the malicious use of typo domains, we check the typo domains from the `.com` zone file against twelve different domain name blacklists. The black lists come from abuse.ch's list of Zeus and SpyEye servers, malwaredomainlist.com, malwaredomains.com, malwarepatrol.com, Google Safe Browsing, and a commonly used commercial list. We also derive lists of malicious domains from recorded requests to DNS-based black lists (DNSBL). This method does not capture the complete list, but rather only includes domains actively marked as malicious and looked up by users during the collection time frame.

3.2.2 Generating candidate typos

We generated a list of all possible typo domains using the most common typo operations: addition (*add*), deletion (*del*), substitution of one character (*sub*), transposition of neighboring characters (*tra*), and supplement this set with a "." deletion operation specific to "www." domain names (e.g. a user typed (*wwwexample.com*). We define this list as the "generated typo" or *gtypo* list. The subset of the gtypo list which was registered within the `.com` TLD includes approximately 4.7 million domains, which we refer to as "candidate typos" or *ctypos*.

3.2.3 Typosquatting definitions

To define the scope of our work, we provide a concise definition of typosquatting.

Definition 1 *A candidate typo domain is called a* typosquatting domain *if (i) it was registered to benefit from traffic intended for a target domain (ii) that is the property of a different entity.*

It is important that both conditions have to be met simultaneously. Typosquatting domain names are registered with the parasitic intent to reap the mistyped traffic of popular domains belonging to someone else. This includes parked domains serving ads, phishing domains, known malicious domains, typo domains redirecting to unrelated content and affiliate marketing. Arguably, these conditions cannot always be checked with confidence, for example ownership information could be disguised.[3]

[3]For example, the name servers `*.aexp.com` of `americanexpressl.com` belong to American Express Inc., but that is the only indicator of ownership. This can only be marked using manual inspection.

According to our definition, parked domains that do not serve ads are excluded from our definition of typosquatting, because they are not making any visible profit from parking. We still consider them as typos until it becomes clear if they are performing typosquatting on the target or serving unrelated content. Candidate typo domains that are defensively registered by the original domain owner are also excluded from typosquatting, because the owner of the typo domain and the original domain are the same. Although defensive typo registrations cannot be considered as typosquatting, they are born as an unwanted consequence of typosquatting.

We define *true typo* domains as follows.

Definition 2 *We call the union of typosquatting domains, parked domains not serving ads and defensive registrations the* true typo *domain set.*

Finally, all candidate typos that are at DL-1 from an original domain yet have unrelated content are considered as incidental registrations, although they can surely benefit from the lexical proximity.[4]

3.2.4 Active crawling

We developed a set of active crawlers to collect additional information about the ctypo domains.

Whois crawler. First, we collect registration data from the WHOIS global database. We restrict our crawler to the thin whois information as provided by Verisign Inc. for the `.com` domains. From the thin whois record, we use the registrar and registration date information.

DNS crawler. We collect DNS data to explore the background infrastructure serving these domains. Our crawler queries separately for A, AAAA, NS, MX, TXT, CNAME, and SOA records for each domain. The crawler then tests for random strings under the registered domain to infer whether wildcarding is present. Wildcarding is the practice when a name server resolves any subdomain under the domain belonging to its authority in the DNS hierarchy.

Web crawler. We use a web crawler to obtain the rendered DOM of each page, along with any automatic redirections that take place during the page load. This crawler uses the PhantomJS WebKit automation framework to provide high volume, full fidelity web crawling with javascript execution, cookie storage, and page rendering capabilities [62]. The crawler follows JavaScript redirections even when they may be obfuscated or contained in child iframes; it then reports the method of redirection and the destination for intermediate and final redirections. We also collect rendered screenshots of a subset of pages for manual inspection.

[4]Here we face another uncertainty presented by scam pages that generate legitimately looking random content. We observed several such cases for suspiciously looking webshops. We make a conservative assessment and categorize them as other (O) in spite of their questionable content

3.2.5 Clustering and categorization

Clustering. We group domains together according to various attributes obtained from available datasets and active analysis. Our goal with this clustering is twofold: to identify typo domains that might have been registered for the same purpose and to point to infrastructure elements that host a large number of typo domains. First, we identify domain sets that are at DL-1 distance from each other, forming a cluster of *typo neighbors*.

Understanding the infrastructure support and the content of the typo domains is required to make an informed decision about their real purpose. To characterize the infrastructure support for typosquatting, we cluster the candidate typo domains based on their registration and hosting information. In particular, we identify the major registrars and name servers (NSs) that host candidate typo domains.

3.3 Features used for domain categorization

Feature description	**Priority**	**Comment**
Lexical attributes		
domain length	M	[106]
highest-ranked neighbor's operation	M	diff. from the most popular original domain
is any neighbor at fat finger distance one?	M	FF typos are more likely to be true typos [106]
nr. of neighbors	L	
nr. of neighbors with *op*	L	where *op*={*add*,*del*,*sub*,*tra*,*www*}
Popularity (Alexa) attribute		
Alexa rank of original domain	H	
Zone file attributes		
total nr of ctypo-s on NS	M	
ctypo/alldomain ratio on NS	H	
total nr. of domains on the NS in the zone	L	
parked keywords in NS domain	H	
Whois attributes		
total nr of ctypo-s at registrar	M	
registration date	L	
DNS attributes		
NXDOMAIN wildcarding	H	
TXT google auth	L	Google ads affiliate auth
total nr of ctypo-s on IP address	M	[40]
Content attributes		
Parked	H	by RE keywords
Serving ads	M	by RE keywords
Total redirection length	M	# of redirections [40]
Domain redirection length	H	# of redirections between registered domains
DERPContent size	M	[40]
Affiliate marketing	M	[106]

Table 3.1: Domain and infrastructures features to categorize candidate typo domains. The column Priority indicates the relative importance in idenfitying typosquatting behavior.

Domain features. We derive a feature set including lexical, infrastructure and content features of the candidate typos as shown in Table 3.1. We selected the features after carefully considering related work, collecting 40+ features in various attribute categories,

and focusing only on relevant ones. To assess the efficiency of the selected feature set, we perform a systematic evaluation based on manual sampling in Section 3.4.1 and we use the results of this evaluation as a benchmark.[5]

Among the chosen features, domain length is a key indicator for typosquatting behavior as longer ctypo domains are more likely to indeed typosquat on the original domain they are close to [106]. Intuitively, the Alexa rank of the original domain indicates that more popular domains are more likely a target of typosquatting. Based on the zone file, we are able to observe the ratio of ctypo domains versus all domain names on a given NS and we deem hosting a large of proportion of potential typo domains suspicious for an NS. Similarly, if the registered domain of the NS contains keywords indicating parking behavior, then ctypo domains hosted on this NS are more likely to belong to typosquatting domains. NXDOMAIN wildcarding is used by major parking service providers to serve ads for web requests regardless of the subdomain. It has been shown that NXDOMAIN wildcarding is a precursor of suspicious behavior and quite often indicates parked typosquatting domains [34, 141]. Thus, we also consider it an indicator for typosquatting when the page content matches some collected parking keywords.[6] Finally, several redirections usually imply suspicious behavior, and we deem them important if the redirection targets a registered domain different from the typo domain and the target domain. The features we selected resulted in a significant improvement over existing methods in identifying typosquatting domains across the whole range of `.com` domains. We leave a more complex feature set selection and parameter calibration using machine learning techniques as future work.

Categorization. Using these features, we attribute typosquatting to candidate typo (ctypo) domains by assigning the tag *typosquatting (T)*, *not typosquatting (NT)* or *unknown (U)*. Unknown is typically used when the domain returns an HTTP or DNS error which prevents successfully downloading the page. We also tag the usage type of the typosquatting domains according to the monetization categories presented in Figure 2.1. We also present the novel approach of categorizing domains based on their monetization strategy. Hence, we tag ctypo domains which do not redirect the user to the target site as *parked (P)* without ads (not on Figure 2.1), *parked serving ads (PA)* (❶ on Figure 2.1), employing a *phishing (PH)* scam (❷), or serving *malware (M)* (❸). When redirection is used, then the ctypo domain can be tagged as *defensive (D)* registration (❼), defensive registration using *affiliate (A)* marketing (❻) in addition to the previously mentioned categories. When a ctypo domain redirects to another domain, then we tag it as *other (O)* (❹, ❺) no matter if it is a competitor or a completely unrelated site.[7] Finally, we mark all uncategorized domains as *unknown (U)*, a set that typically contains unreachable domains.

[5]Manually generated datasets are widely used as indicators for malicious behavior; for example, the PhishTank phishing list is a major component of SURBL, the leading domain blacklist. [12].

[6]Here, we improve on the techniques used by [34] and [59] to find parking services and parked domains

[7]Determining domain competitors is beyond the scope of this work; we summarized redirections to third-party domains independently of the typosquatter's intent. While these redirections might simply be to other parked sites, any redirection away from the original site is a traffic loss for the original domain owner.

3.3.1 Checking Maliciousness

To analyze how the typo domains are used, 12 black lists are checked for an indication that the domains are malicious. To check a black list, we look for anything that was on that list during the first quarter of 2013. A "match" is a second-level domain match, since this is the relevant typo label.

To perform a check, a superset of all the domains for Q1 2013 per list was made, and the typo and Alexa domains were compared against that superset. For Google Safe Browsing, due to Google's technical constraints, the each set of domains was checked using the provided python client against data for May 1, 2011 to July 31, 2013. The results are presented in subsection 3.4.6.

3.4 Analysis

In this section, our goal is to characterize the current state of typosquatting. For this purpose, we use the `.com` zone file as the most popular and versatile TLD for domain registrations.

3.4.1 Typosquatting distribution

Experts believe that most newly registered domains are speculative or malicious. Paul Vixie posits that "most new domain names are malicious" [138]. The subset of registered typo domains from the generated typo domains is widely accepted as true typo domains ([106, 140]), and [106] has shown that this assertion mostly holds for the top 3,264 `.com` domains in the Alexa ranking.

We believe, however, that this assertion does not necessarily hold if we extend our scope to less popular domains. In order to investigate this possibility, we first perform a manual sampling from various sets of the `.com` zone file to systematically control the accuracy of typosquatting identification and also to provide a credible ground truth for investigation. We conduct a manual inspection of four thousand domain names because the typosquatting definitions in the academic literature [106, 140] are very crude. Moreover, we present our mitigation tool analysis in Section 3.5, and in so doing also discuss the limitations of existing defense tools that typically only focus on correcting typos for a limited set of popular domain names.

We first take a sample of 1000 ctypo domains randomly with uniform distribution from the Alexa top domain list to match the sampling methodology of [106]. We then complete this with three additional samples of 1000 ctypo domains each derived from the `.com` zone and the Alexa domain list. Our four sample sets are thus the following: ctypos of the the Alexa top/mid/tail domains (recall their description from Section 3.2.1) and ctypos of a random sample taken over the whole `.com` zone file. With these multiple sets, our goal is to check whether the conclusions from prior work regarding the frequency of typosquatting hold for less popular domains.

Typosquatting domains are notoriously difficult to identify. In several cases, only a careful investigation shows the potentially speculative behavior. We performed manual

verification to establish a ground truth for identifying typosquatting domains. Clearly, manual classification is not perfect, but it allowed us to go in depth at domains that were ambiguous. In manual classification, we go beyond simple rules, like identifying simple one-hop defensive redirections and consider the environment, like the owner of name servers (`ns*.aexp.com` indeed belongs to American Express Inc) or potential relation between brands (Oldnavy is a subsidiary of GAP and thus `oldnavy.com` redirects to `oldnavy.gap.com`). We could further establish a ground truth based on crowdsourcing typosquatting identification. This would remove the bias introduced by the mindset of the authors, yet it could introduce significant inaccuracies due to the lack of experience and understanding of typosquatting by the crowd.

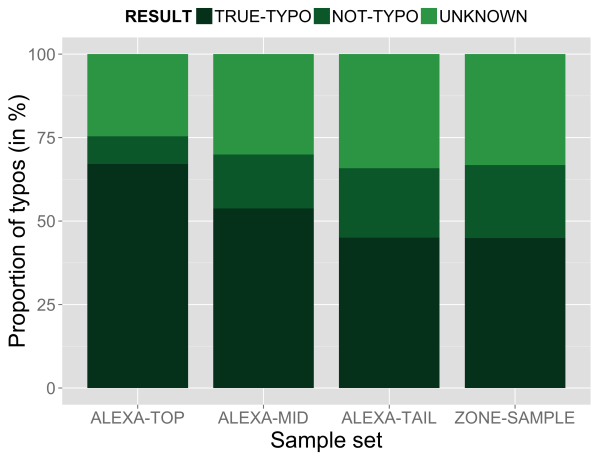


Figure 3.2: The prevalence of true typo domains in the four sample sets drawn popular and less popular `.com` domain names. The domain sets are ctypo samples of the Alexa top/mid/tail domains and the domains in the `.com` zone file. The number of true typo domains decreases with the Alexa rank of original domains, yet their ratio in the whole population remains high.

According to our manual inspection, a majority of the ctypo domains registered against the Alexa top domains are true typo domains (as shown in Figure 3.2). This result confirms the finding of [106]. We note here that there is a significant number of ctypo domains for which we cannot reliably decide if they are typo domains or not (U). This is mostly due to the fact that domains return "not accessible" responses for DNS or HTTP queries. The number of true typo domains steadily decreases when we perform the same experiment for the Alexa mid and tail domains, yet it remains high (around 50% within the set of all ctypo domains). While this indicates that thousands of domains are indeed typosquatting on less popular domains, to present defenses we need to develop a more reliable strategy to predict whether a domain is involved in typosquatting.

3.4.2 Accuracy of identification

We developed an automatic categorization tool based on the domain features presented in Section 3.2.5 called *Yet Another Typosquatting Tool (YATT)*. YATT has three modes. In the *passive mode, YATT-P* uses the information readily available from static files, such as lexical features, zone information and Alexa information. In the *DNS mode, YATT-PD* includes Whois and DNS features collected from the active crawler infrastructure, and finally in the *content mode, YATT-PDC* content features obtained via crawling are added to the categorization. The complexity of the algorithms increases from YATT-P to YATT-PDC. We expect that YATT-PDC will show the best performance in categorizing typo domains, but the other variants can still provide useful information if one wants to avoid the tedious work of collecting content features.

As presented before, we fine-tuned the parameters of YATT, but further improvement might be possible with additional features and a more complex feature selection process. At the moment, this optimization is left as future work.

In addition to YATT, we tested notable typosquatting identification methods from related work. First, we consider the method in [106], which showed that most ctypo domains of DL-1 are indeed true typos. Their primary feature is the domain length so we repeat their experiment for DL-1 and we name their method *AllTypo*. Then, we implemented the most important features of the *SUT-net* algorithm in [40] and compared it to various modes of YATT.

In Figure 3.3, we compare the accuracy of the typo identification methods in related work and the three modes of YATT to the established benchmark of manual evaluation. We perform this accuracy evaluation on the four ctypo domain samples described in Section 3.4.1. In Figure 3.3, we see that all five algorithms mark ctypo domains as positives in the Alexa top dataset. This assertive categorization results in a good true positive (TP) rate, a reasonably small number of false positives (FP) and with almost no false negatives (FN). Only the full YATT-PDC can identify a small set of true negatives (TN) in the population. In the Alexa mid, the agressive typo identification of AllTypo and SUT results in a high FP number whereas YATT keeps the FPs low while correctly identifying TNs (with YATT-PDC being the most accurate as expected). For the Alexa tail and zone datasets, the number of true typos further decreases and both AllTypo and SUT overwhelmingly categorize these domains as typos resulting in a very large false positive rate. All versions of YATT keep the FPs low and correctly categorize TNs at the expense of a small number of FNs. It is clear that perfect categorization is difficult to do, but YATT does not sacrifice much precision as the number of non-typo domains get introduced.

Next, we study the accuracy of the YATT-PDC to identify parked domains and other typosquatting indicators based on our manual sampling in Table 3.2. Note that related work on typosquatting identification usually focuses on typo identification and leaves the categorization aside. Only the active mode of the algorithm can perform this categorization, because it requires content features. YATT-PDC uses regular expression-based matching for the identification of parking domains. It matches these domains with about 85% precision, the error stemming from the incompleteness of the set of regular expressions we use. YATT-PDC still finds the majority of the parking sites and lists a significantly larger number of

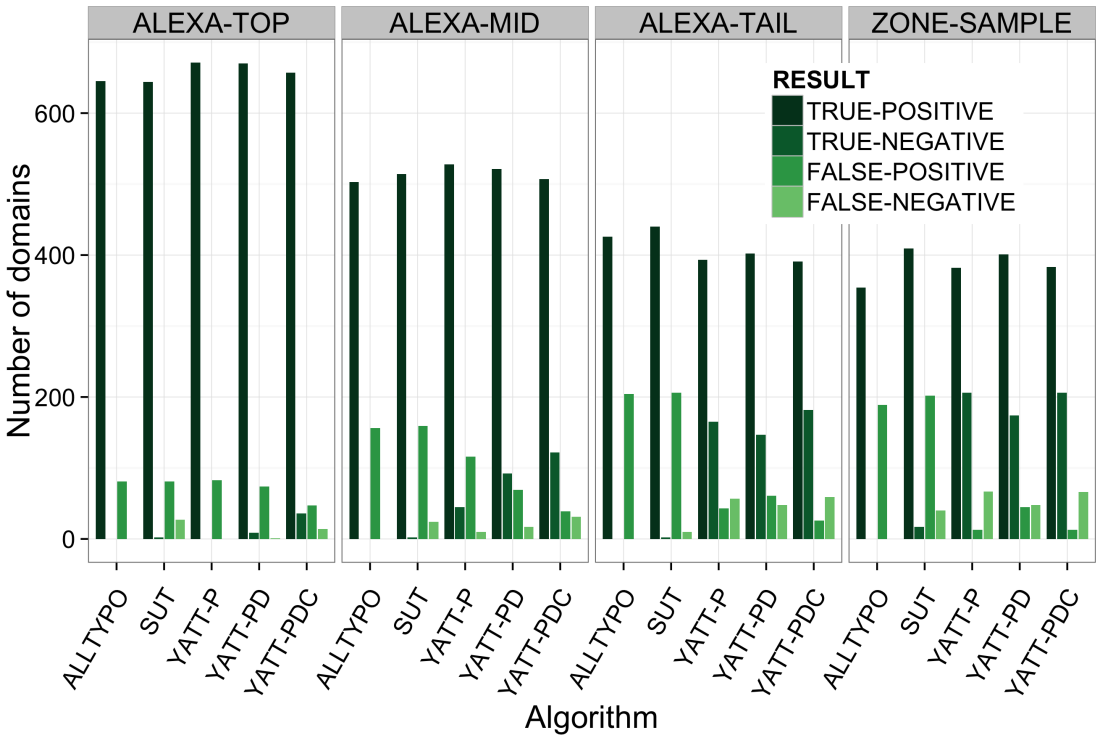


Figure 3.3: Accuracy of four typosquatting prediction tools. We tested (a) AllTypos, (b) SUT-net-based content features, (c) YATT-P, (d) YATT-PD, and (e) YATT-PDC for the four ctypo domain sample sets of (1/2/3) the Alexa top/mid/tail domains and (4) the domains in the `.com` zone file.

	PARKED			DEFENSIVE			AFFILIATE		
	False Positive	True Positive	False Negative	False Positive	True Positive	False Negative	False Positive	True Positive	False Negative
Alexa top	3	402	76	0	39	15	0	27	1
Alexa mid	3	358	50	0	18	3	0	15	0
Alexa tail	1	295	59	0	9	3	0	0	0
Zone	0	265	43	1	7	4	0	0	0

Table 3.2: The accuracy of YATT to identify parked, defensive and affiliate registrations across the sample datasets.

parking sites than methods in related work [34, 59]. For the defenisve domain registrations, YATT-PDC fares worse. It only finds 60-85% of the defensive registrations. This is due to the complexity of defensive registration patterns that can mostly be caught by a human eye. Finally, for affiliate registrations, YATT-PDC performs quite well, correctly categorizing almost all domains. We also checked the existence of malicious and phishing domains in our sample dataset, but we could not find any in such a small sample. Our results from

more rigorously checking for maliciousness in typo domains is described in subsection 3.4.6, however maliciousness was not used to classify typo domains as typos.

YATT results in an accurate prediction of true typo domains and domain categories for the whole range of the domain population and hence its results can be used as a basis for intervention attempts and tools. Using YATT, we compile a typosquatting blacklist and use it in a set of mitigation tools (see Section 3.5).

3.4.3 Presence of typosquatting registrations

Having designed an accurate typosquatting identification tool, we now study the existence of typosquatting in current domains registrations. We first obtained 4.7 million ctypos targeting the `.com` domains in the Alexa top 1m domain list and existing in the `.com` zone file using the methodology described in Section 3.2. Recall, that we split the original domains according to their Alexa rank into the Alexa top/mid/tail categories.

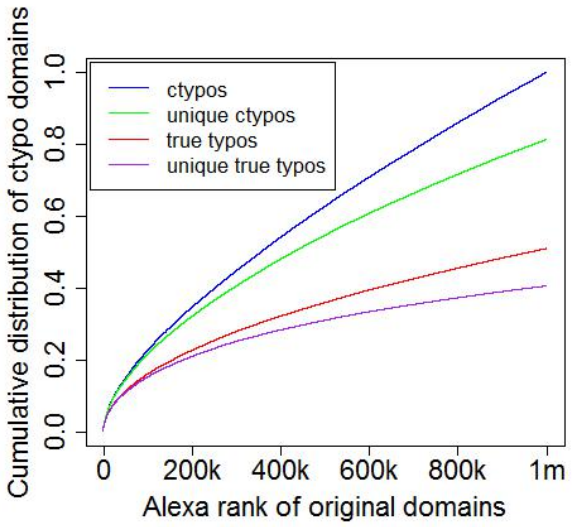
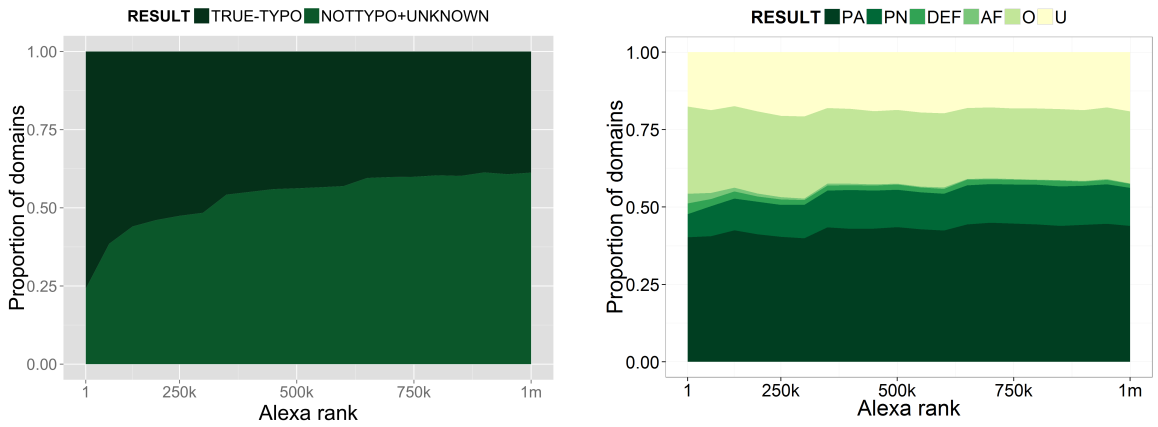


Figure 3.4: The cumulative distribution of true typo domains in ctypos and unique ctypos as a function of the Alexa rank of the original domains.

The first and foremost question is the extent of typosquatting targeting the Alexa domain set. We use YATT to determine typosquatting behavior and partition ctypo domains into the categories described in Section 3.2.5. In Figure 3.4, we plot the cumulative distribution of ctypo domains as a function of the originals' Alexa rank, and we also plot the cumulative distribution of true typo domains. We see that the number of true typos steadily increases as the Alexa rank decreases, although at a slower pace than the number of ctypos. In addition, we also plot the cumulative distribution of unique ctypos and true typo domains.

We then show the fraction of true typos in the population of ctypos in Figure 3.5(a). We calibrated YATT to make a decision about each ctypo and thus it conservatively categorizes the majority of unknown domains as not typos. For Alexa top sites, the fraction of true



(a) (b)

Figure 3.5: The existence of typosquatting domains targeting the Alexa domain set. The fraction of (a) true typo domains and (b) various typo categories in the true typo population.

typos is higher, but for lower Alexa ranks the number of nottypo and unknown domains increases. This is consistent with our benchmarking results in Figure 3.2. Finally, in Figure 3.5(b), we present the typosquatting categories as a function of the original domains' Alexa rank. We observe that the bulk of the true typo registrations profits from parked domains with advertisements. The number of defensive and affiliate registrations is higher for the Alexa top sites, but then then the affiliate registrations disappear as we head to the Alexa tail while the defensive registrations persist. Finally, there is a significant number of non-typo domains incidentally close to the domains in the Alexa domain list.

Projecting our results to the total number of `.com` domains in the zone file, we estimate that about 53% of them are candidate typo domains and hence 20% of the total domain set are true typo domains. Based on our results, we estimate that about 21.2m domains are true typo domains in the `.com` zone file.

3.4.4 Trend analysis

We analyzed trends in typo domain registrations for a period of approximately one year (from 2012-10-01 to 2013-10-15). We considered domains from four datasets: domains from the `.com` zone file, ctypos from the `.com` zone file, ctypos targeting the whole Alexa list and ctypos targeting the Alexa top list.

For the purposes of our analysis, we use visibility into the `.com` zone file as a proxy for domain registration. Because the actual registration and registration lapse events are not visible to us, we use presence in the zone file as a proxy for registration events. We define a registration event as one where a domain was not in a daily zone dump, and was present in the subsequent day's zone file, and vice-versa for a registration lapse, or deregistration.

We looked at the change in domain registrations over time. Figure 3.6 plots the cumulative changes in the number of domains registered in the above mentioned domain sets. While the overall registration rate is steady, the difference between the rate of

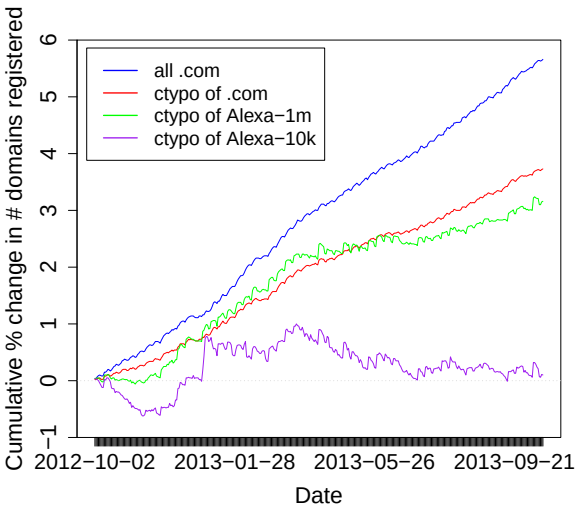


Figure 3.6: Cumulative change in the total number of domains registered over time.

Alexa-10k targeted and Alexa-1m targeted typos suggests that, through enforcement or typosquatter preference, the overall increase in registrations targeting popular domain typos is far smaller even though many DL-1 typos of popular domains are still available. It is also interesting to note that the spike centered on January 1 2013 is due to four organizations (sedoparking, 1and1.com, dsredirection, and graceperioddomain.com) registering a large number of domains: these four account for 87% of all domains registered at that time.

Our next analysis focuses on the amount of speculation present within the market for typosquatting domains between 2012-10-01 and 2014-02-20. Table 3.3 shows the percentage of stable domains, the average uptime, and the percent of domains experiencing at least one reregistration event during our measurement time period. As might be expected, random domains are purchased and left to lapse very often, with less than one third being reregistered after being abandoned. Domains which are a typo of a popular domain, however, experience almost twice as much interest, although they are not active for significantly more time. This trend suggests that the information asymmetry of the typosquatting marketplace is such that new speculators register old typos at a much higher rate than random domains.

3.4.5 Typosquatting redirections

In this section, we discuss our first encounter with typosquatting redirections. Chapter 4 builds on this initial encounter and provides an in-depth analysis of how typosquatters leverage advertisement networks to redirect users.

Now, we scrutinize the affiliate redirections via third-parties. This third-parties can be legitimate brand protection companies, but more frequently they are typosquatting affiliates collecting type-in traffic from a large number of typo domains.

	Stable	Mean uptime	Reregs
Alexa-1m ctypo	72.3%	458 days	49.5%
Alexa-10k ctypo	71.0%	454 days	49.5%
Alexa-1m	93.3%	501 days	67.1%
Alexa-10k	99.0%	506 days	86.8%
Random sample	70.4%	440 days	28.5%

Table 3.3: Speculation trend analysis between 2012-10-01 and 2014-02-20. Alexa list and zone file used was from 2012-10-01. The "stable" column indicates what proportion were registered throughout the dataset. "Reregs" indicates how many domains experienced at least one lapse in visibility at the zone file, indicating that the domain was decommissioned and then reactivated. "Random sample" is a selection of 2 million random domain names from the `.com` zone file of 2012-10-01.

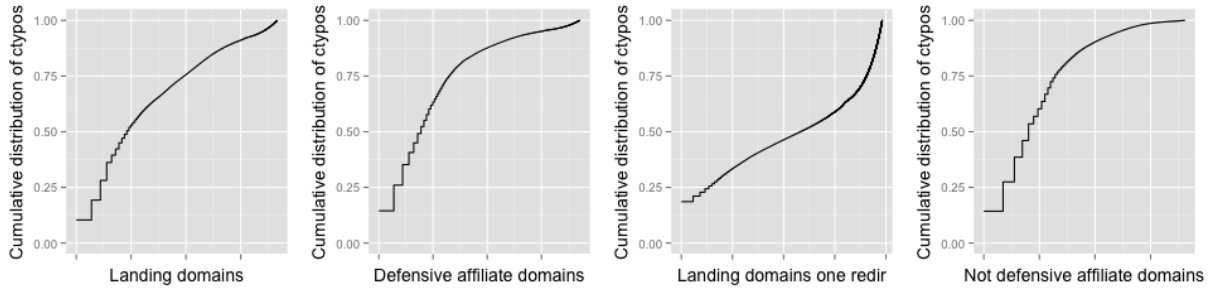


Figure 3.7: The leftmost figure shows the cumulative distribution of landing pages targeted from ctypo domains. The second figure shows the cumulative distribution of intermediate domains in case of defensive redirections. The third figure is when the length of the domain redirection chain is one. Finally, the rightmost figure shows the cumulative distribution of intermediate domains in case of redirections targeting a third party.

Domain redirections that lead back to the targeted original domains without intermediate domains are considered defensive registrations, as explained in Section 2.1. If the redirection leads back to the target domain via a third-party, then we call it an affiliate defensive registration. In Figure 3.7 the first graph shows that in the cumulative distribution of third party landing pages, eleven domains (less then 0.1 percent of all of these landing pages) get redirections from more than 50 percent of ctypos redirecting to a third party domain. The second graph in Figure 3.7 shows defensive affiliate domains, where the landing pages is the original domain, but the traffic goes through an intermediate affiliate domain. 18 such intermediate domains (1.3 percent of all domains) are responsible for more than 80 percent of defensive affiliate marketing. Even though this set has a very small overlap with the non-defensive affiliate domains, a small fraction of affiliate domains are controlling 80 percent of the affiliate market.

Finally, if the redirection leads to a third-party domain, that is away from the original target, then this is considered truly speculative. The third graph in Figure 3.7 shows redirections to third-party pages with only one redirection. Here the domains are more

	# Mal-ware Hits	% of List Marked Malware	# Phish Hits	% of List Marked Phish
Alexa	9990	1.907%	27	0.005153%
ctypos	17485	0.3716%	272	0.005781%
ttypos	3720	0.1585%	125	0.005329%

Table 3.4: Google Safe Browsing results for domains in Alexa, *ttypos*, and *ctypos*.

widely distributed: there is only one big landing domain hugedomains.com which receives traffic from more than 21 percent of this type of redirection. The last graph shows the cumulative distribution of all affiliate domains participating in third-party redirections with a non-defensive purpose. That means that these affiliate domains lead away the users from the targeted original sites. 41 of these non-defensive affiliate domains (0.4 percent of all such domains) control the traffic originating from more than 80 percent of candidate typo domains. This means that, here too, a relatively small set of domains control the majority of such traffic going to a few landing pages.

3.4.6 Maliciousness of Typo Domains

In order to test the assertion that typo domains are more malicious than other domains, the candidate typo (ctypo) and true typo (ttypo) domains extracted from the `.com` were checked against a variety of available black lists. These results are compared against the same test on the Alexa domains. By using 12 available black lists from various sources fluctuations due to the idiosyncrasies of any individual list can be controlled.

The Alexa top 488,133 `.com` domains (all the `.com` domains in the top 1m) are more likely to appear on black lists than the typos of them, either *ctypos* or *ttypos*. This result is consistent across all 12 black lists investigated. In each case, the Alexa domains are more likely to host malicious activity. The percentage of .com domains from the Alexa list on each black list is always higher than the percentage of *ttypo* domains on the same list.

Google's Safe Browsing list requires a different checking method, due to their storage method. The list also distinguishes between a match due to malicious content or attempts at phishing. However, the results show a similar trend. The Alexa domains are more likely to be purveyors of malicious software. Table 3.4 shows the results for Google Safe Browsing checking for any listing from May 1, 2011 – July 31 2013.

There are several possible causes for this pattern, and several of them would be uninteresting. A possibility is that there is a pocket of malicious activity using typos, but that most of it is benign. The first place to look for this would be the name servers hosting predominantly typo domains. There are 10 name servers for which most of the domains they host are typos of other domains—for these name servers, between 20-80% of their domains are typos.

The typo domains hosted on these 10 name servers seem to be even less likely to appear on a black list. The average percentage of these name servers' domains on any of the black

lists is 0.051%, and the maximum percentage of typo domains hosted by one of these name servers on any one list is 0.27%. Both of these numbers are below those both for typos generally as well as the results for the Alexa domains.

3.5 Intervention options

Just as defining typosquatting remains one of the grey areas of domain name security, developing effective intervention techniques is similarly difficult. So far, most intervention attempts remain ineffective. In the following, we present viable typosquatting mitigation options and present a set of practical tools to prevent typosquatting from negatively effecting users.

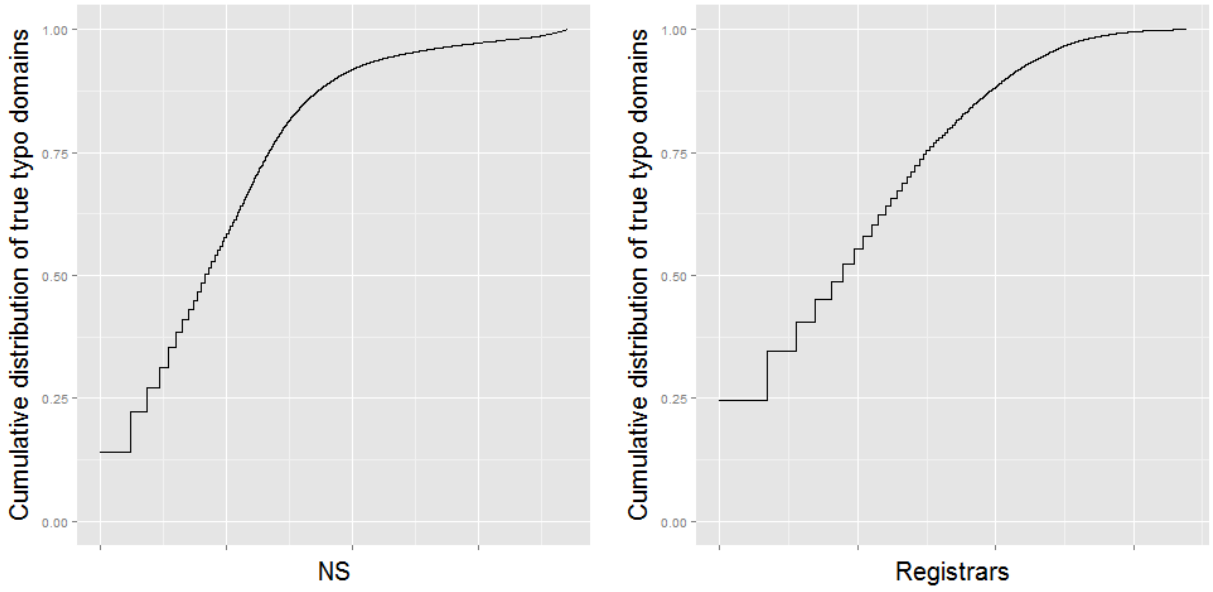
3.5.1 Policy intervention

Much of the effort to crack down on typosquatting focuses on policy options. Two major tools exist for policy intervention. The first is the UDRP arbitration framework provided by ICANN [64]. Unfortunately, only a small fraction of typosquatting domains enters the UDRP procedure [106], although domains are claimed by their trademark holders very often.

The Anti-cybersquatting Consumer Protection Act (ACPA) (15 USC §1125(d)) offers an alternative to the UDRP through legal action. The act "was designed to thwart cybersquatters who register Internet domain names containing trademarks with no intention of creating a legitimate web site, but instead plan to sell the domain name to the trademark owner or a third party." While originally aimed at preventing cybersquatting, in May 2013 Facebook successfully litigated a case including typosquatting domains, earning a US $2.8 million judgement [57]. As with any legal action, the enforcement of this act is costly and only major trademark holders have exercised their legal rights [103, 129, 130]. Additionally, the bad faith of typosquatting registrations is difficult to prove and hence the legal action might not always be efficient [125]. Unfortunately, even vigilant companies seem overwhelmed by the number of typosquatting domains targeting their brands, motivating them to litigate; even so, many of their domains are still controlled by typosquatters.

3.5.2 Infrastructure support

Another option for intervention is to motivate registrars and hosting providers to scrutinize domain name registrations when they happen (with a mandatory light-weight UDRP procedure for example). Let us now look at the potential of registration intervention at the infrastructure side. Figure 3.8 shows the distribution of typosquatting domains (a) as a function of the registrars and (b) as a function of the supporting NSs (setting the x axis to a log scale to improve visibility). We observe that most true typo domains cluster at major registrars and are hosted at a few NSs. In particular, 12 NSs and 5 major registrars are responsible for hosting 50% of the true typo domains. Forcing these major registrars to



(a) (b)

Figure 3.8: Intervention potential at domain registrars and hosting companies. We present the distribution of typosquatting domains (a) as a funtion of the registrars and (b) as a function of the supporting NSs (while setting the x axis to a log scale for better visibility)

enforce prudent registration practices with respect to typosquatting may be a viable policy option.

NS	True typos	All domains	Typo ratio
`a0f.net`	5221	6332	82%
`citizenhawk.net`	8819	12004	73%
`easily.net`	18281	36890	50%
`domainingdepot.com`	51854	132864	39%
`next.org`	9426	30252	31%
`domainmanager.com`	23493	90929	26%

Table 3.5: Worst offender NSs in true typo hosting with at least 5000 true typo domains. All NSs in the top list have higher than 25% of true typo / all domain ratio.

Based on the `.com` zone file, we are also able to collect the ratio of true typo domains to the total number of domains. Table 3.5 presents the top offenders with at least 5000 true typo domains hosted. Interestingly, there are only 65 NSs with such a high number of true typo domains. We see that the worst offenders almost exclusively host true typo domains, and none of them belong to the major hosting companies.[8] Further investigating these typo domains we found two interesting results. First, out of the 6 name servers with the highest

[8]An interesting case might be `citizenhawk.net`, a brand protection company who probably registered a large number of domain names for protecting their customers.

true typo ratio, 5 have domains that are privately registered and only `citizenhawk.net` is not, showing that the others are aware that their monetization strategy is questionable. Second, we found that on the average 24.5 percent of the domains hosted by these NSs is in the top Alexa, which is 2.5 time higher number than for the rest of the name servers. This indicates that these name servers are more effectively targeting popular typo domains than major hosting services who are not focusing on typosquatting. These hosting companies with an unusually high number of true typo domains could be regulated to effectively decrease the effect of speculative typosquatting.

Infrastructure intervention is promising if it can be enforced globally by ICANN on the supporting providers. Unfortunately, it is unlikely that such a global action will emerge as this is counterproductive for the domain registrars, and thus miscreants can always shift their businesses to negligent or accomplice providers who are financially motivated to assist their businesses. Registrar- and hosting-level intervention remains ineffective against spammers [94, 97] and it is unlikely that it will be effective against typosquatting. Registrars and hosting companies do not suffer from typosquatting, thus there is little economic incentive for them expend resources to defend against it.

3.5.3 Mitigation tools

The last option to counter typosquatting is the application of technical tools to reduce the impact of typosquatting. There exist mitigation tools to this end, but most tools suffer from either trivial errors or from small coverage of typosquatting domains.

Related work. Wang et al. developed *Strider Typopatrol*, a tool to automatically discover typo domains of popular domains [140]. They focus on a small subset of the Alexa top domain list [1], phishing targets, and childrens' websites. OpenDNS [111] provides typosquatting correction in their DNS services, but only for major TLDs. A similar tool called *URLFixer* [25] was introduced in the Adblock Plus advertisement blocking tool. The URLFixer tool includes misspellings of top Alexa domains, but fails to correct less popular domain names and includes some short domain names leading to false corrections. Chen et al. [44] develop a browser plugin to check typo domains based on a user-customized local repository. Banerjee et al. [39, 40] propose *SUT*, a method to identify typosquatting domains mostly based on HTML properties. Finally, the autocomplete feature of most major browsers can also decrease the instance of typos, albeit only for previously visited sites.

Initial tests show that most existing solutions are limited in scope (the most popular domains or most frequent typos), in features (only TLD correction or HTML features) or in the information used (search typing or local browser history) and consequently these tools are missing a large set of typosquatting domains.

The YATT framework. We developed a typosquatting categorization tool, YATT, that uses an extended domain feature set to provide accurate typosquatting identification. Based on the output provided by YATT, we implemented three typosquatting detection and protection services. The first service is a typosquatting blacklist (YATT-BL) compiled from the output of one of the versions of the YATT tool. As a DNS based blacklist, this access method is quick and lightweight. The tool works similarly to major domain blacklists

such as URIBL [24], SURBL [19] or the Spamhaus DBL [18] and it can be used to filter out typo domains from live traffic. The DNS server uses RPZ [138] to efficiently distribute the typo list.

Second, we implemented a Firefox browser plugin and a corresponding typo protection server to protect users from typosquatting domains. Our plugin contacts the typo protection server each time a user types in a domain and raises a warning if the domain typed by the user is found on the typosquatting domain list. The user is provided with the option of accepting the automatic correction or rewriting it to her needs. The typo protection server uses YATT-BL DNS blacklist described above.

Third, we are in the process of implementing a YATT DNS server for organizations that want to avoid typosquatting yet do not want to expose their DNS traffic to a third party server. Using this tool, a company could periodically download an updated typosquatting blacklist and query it locally.

3.6 Conclusion

Typosquatting has caused annoyances for Internet users for a long time. Since users lack effective countermeasures, speculators keep registering domain names to target domains and exploit the traffic arriving from mistyping those domain names. Existing studies of typosquatting focused on popular domain names and thus have only shown the tip of the iceberg. Similar to traditional cybercrimes like spamming or financial credential fraud, typosquatting has minimal transparency, allowing what may be an unprofitable activity to continue because new entrants see its effects and attempt to become profitable typosquatters themselves. Investigating such speculative, "gray area" behavior longitudinally can give us insights which might generalize to traditional cybercrime and cybercriminals.

In this paper, we performed a thorough study for an extensive set of potential target domains. We found that 95% of typo domains are targeting less popular domains. We designed an accurate typo categorization framework and find that typosquatting using parked ads and similar monetization techniques not only exists for popular domains, but a whole range of domain names in the Alexa domain list. We showed that a large number of incidental domain registrations exist with close lexical distance to the target domains. Our conservative estimates indicate that as much as 21.2 million `.com` domain registrations are confirmed true typo domains, which accounts for about 20% of all `.com` domain registrations. Additionally, we found that the typosquatting phenomenon is only continuing to thrive and expand.

The difficulty of categorizing typosquatting domains partially explains the inefficiency of existing mitigation techniques. Much like typosquatting itself, mitigation is a gray area: one cannot easily classify a new registration as an example of typosquatting based on the name alone. As such, typo domains rarely appear on blacklists. To counter this problem, we designed several defense tools that rely on a broad range of features. We provide a typosquatting blacklist and a corresponding browser plugin to prevent mistyping at the user side. While typosquatting will likely continue to exist, these analyses and tools may improve user experience and further decrease the profit available to typosquatters.

Chapter 4

The Role of Typosquatting Domains in Malicious Advertisement Networks

In Chapter 3, we discussed how to detect typosquatting domain names in general and the two most common monetization techniques: pay-per-click and pay-per-redirect advertisement. In this chapter, we dive deeper into understanding the pay-per-redirect advertisement ecosystem in which users are automatically redirected to advertisers without clicking on an advertisement. To acquire user traffic, typosquatters together with illicit website owners frequently rely on traffic distribution systems (TDSs) operated by less-than-scrupulous advertising networks. While a number of case studies on various TDSs or the businesses they serve (e.g., illicit pharmacies) have been described s [35, 63, 92, 96, 110, 114, 128], we still lack an understanding of how different illicit activities frequently leverage the same advertisement networks and, subsequently, the same malicious advertisers. Studying these advertiser/TDS ecosystems is challenging because they try to cloak their malicious activity from researchers, and target users based on the device used requiring researchers to emulate different user profiles. To address these challenges and building on previous research [33, 77, 90, 96, 104, 114, 128], we design ODIN (Observatory of Dynamic Illicit ad Networks) to study four types of traffic sources: typosquatting, copyright-infringing movie streaming, ad-based URL shortening and illicit online pharmacy websites.

ODIN collected data from 78,668 webpages over two months (June 19, 2019–August 24, 2019), posing as six different types of users (e.g., mobile, desktop and crawler) to address cloaking and user differentiation. Accumulating 874,494 scrapes and over 2TB of screenshots, browser events and archived HTTP communications. We observed 81 percent 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 percent of the time and all of them often expose users to the same malicious advertisers. 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. Indeed, the advertisement networks we study redirect mobile users to entirely different advertisers compared to desktop users. In response, we design a classifier, which can make precise predictions about the likelihood of a user being redirected to a malicious advertiser.

4.1 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 [33, 35, 81, 90, 92, 104, 110, 114, 128] without user consent. Dubious redirections of visitors also frequently expose them to malicious content, including deception, phishing, scams and malicious downloads [33, 77, 81, 96, 104, 108, 110, 114, 134]. While the research community has documented a number of abusive practices through specific case studies [33, 40, 46, 63, 77, 81, 83, 90, 92, 95, 96, 104, 106, 108, 110, 114, 121, 122, 124, 128, 133, 134, 140, 146], we still lack a general understanding of how malicious advertisement ecosystems interact with each other, and of the specific roles different entities assume.

This chapter 1) describes a measurement infrastructure called ODIN (Observatory of Dynamic Illicit ad Networks), 2) discusses results from at-scale data collection using ODIN, and 3) introduces possible countermeasures based on these findings.

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 manually label tens of thousands of screenshots of pages ODIN collects, and use these labels to perform a series of automatic analyses of page contents to better understand the threats these TDSs pose.

We seed ODIN with four distinct types of traffic sources: (i) "typosquatting sites" [128] (e.g., `yotube.com`), (ii) copyright-infringing sites, that stream pirated movies [63], (iii) ad-based URL shortening services that shorten URLs in return for exposure to potentially malicious ads [110], and (iv) unlicensed online pharmacies [90]. 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 [90] hinted at vertical integration, at least in the pharmaceutical ecosystem; we revisit whether this finding still holds nearly a decade later.

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.

Result highlights. Using ODIN, we collected data from 78,668 webpages over two months (June 19, 2019–August 24, 2019), scraping them 874,494 times in total and accumulating 2TB of data. Posing as six different types of users, ODIN finds 81 percent more malicious landing pages and 96 percent more suspicious landing pages, compared to visiting pages using the user profile which experienced the most malice. 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 do find evidence 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.

From a criminal ecosystem standpoint, we find evidence of TDS reuse *across* illicit activities. Some traffic source pairs share 44 percent 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 [104, 122], deceptive surveys [46, 81], deceptive downloads [33, 134], 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.

We discover that users are often differentiated based on the device they use. Mobile users are exclusively targeted with deceptive surveys and illicit adult content tailored to them. Conversely, desktop users are exposed to certain technical support scam pages and deceptive downloads that mobile users would never see. Unfortunately, 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.

This analysis leads us to design a proof-of-concept classifier, which relies only on features available at the time of redirection, and can be used with high precision to stop users from landing on malicious pages.

4.2 Data Collection: ODIN

Our data collection must fulfill several objectives. Our 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* or evasion by TDS operators.

To meet these objectives, we designed the collection infrastructure represented in Figure 4.1. 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

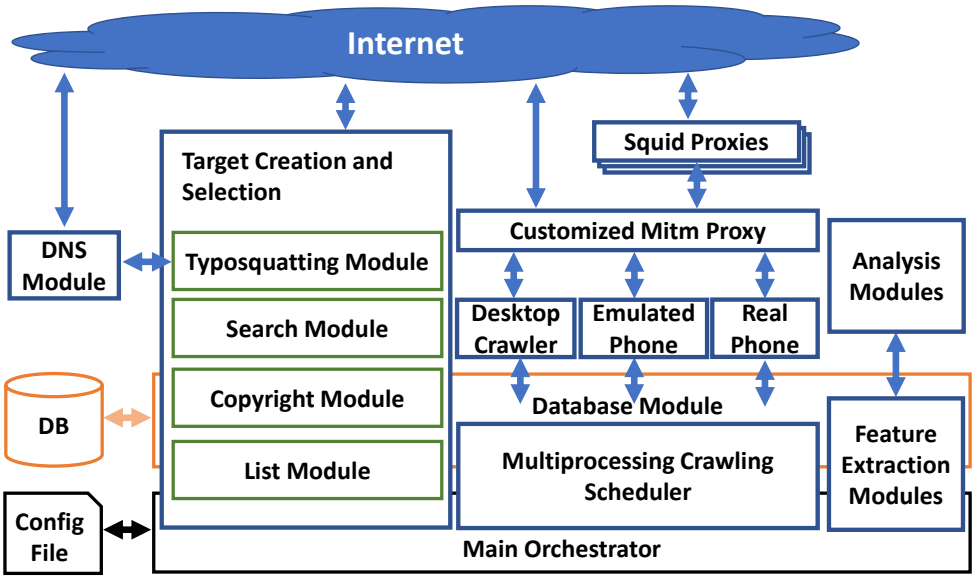


Figure 4.1: High-level overview of ODIN.

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.

4.2.1 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 domains. The main typosquatting dataset *typo-main* consists of all possible Damerau-Levenshtein distance one [49] variants of Alexa's top 500 most popular domain names. Using the DNS module, we select only those domain names that responded with valid `NS` and `A` records. We use typosquatting domain names targeting popular domains as we conjecture that they are more frequently used for malicious redirections.

We also generate three sets of typosquatting domains targeting less popular Alexa domains based on previous work [128]. The *typo-top*, *typo-mid* and *typo-tail* sets consist of typo domains targeting Alexa domains ranked from 1 to 10,000, 10,000 to 250,000, and 250,000 to 1,000,000 respectively.

Finally, the *pharma-typo* dataset is similar to the typosquatting dataset, except that we use Alexa's top 100 pharmacy domains as targets. We only study popular pharmacy domains as we are mainly interested in what connections we can find between the pharma-typo, typosquatting, and illicit pharmacy datasets.

For each set 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

Table 4.1: Summary of user profiles.

User Profile	Device Used	User-agent	Mobile Emulation	Referrer	Proxy
Vanilla Desktop	Linux Server	Windows Chrome	No	None	Yes
Referrer Desktop	Linux Server	Windows Chrome	No	Google	Yes
No-Proxy Desktop	Linux Server	Windows Chrome	No	Google	No
Google Bot	Linux Server	Google Bot	No	None	Yes
Emulated Phone	Linux Server	Android Chrome	Yes	Google	Yes
Real Phone	Nexus 6P Android	Android Chrome	–	Google	Yes

off between the limited number of target URLs that our crawling infrastructure can visit 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 [90, 92]. We freshly generate these URLs and select a maximum of 2,000 for each experiment we run.

Copyright-infringing websites. We collect URLs from `softonic.com`, a site crowd-sourcing answers and rankings of answers to all sorts of user questions. The site's statistics claimed that tens of thousands of users voted up or down 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 copyright-infringing streaming sites.

Blacklists. We also use URLs from known blacklists. We select a random sample of a thousand URLs from PhishTank [12]. We also select a random sample of 1,000 URLs for each type of SURBL lists [19], including spam and abuse sites, phishing sites, malware sites and cracked sites. For each blacklist we select a new sample for every run of the multi-proxy experiment.

Alexa domains. Finally, we select 2,000 random URLs from Alexa's top 1 million domains [1] as a baseline comparison for our other datasets.

4.2.2 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 cloaking techniques miscreants use.

More specifically, we scrape each URL target six times using the six different user profiles, as shown in Table 4.1. 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. We used the most common Windows Chrome `User-agent`. To combat referrer header based cloaking as observed by previous work [90, 92], 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 possible [135]), we also use a Nexus 6P 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 [90]. To observe how TDSs react when encountering a search engine crawler, we set the `User-agent` to Google's crawler.

4.2.3 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 the domains they point to 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-agent`s 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 tools available is vast, thus we cannot defeat technologically advanced and motivated attackers, e.g., attacker leveraging canvas fingerprinting [41].

IP rotation. Miramirkhani et al. [104] 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 smaller and less well-known VPS provider [13]. We do not leverage residential IP addresses

to avoid ethical quandaries [102], and because recent research [79] 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 [1] 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.

4.2.4 Experiments

In this chapter, we use ODIN to collect data through two large-scale 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 this Main experiment, ODIN visited pages 438,354 times during the 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.

This experiment only uses a (large) subset or the targets described above: it does not use the *typo-top*, *typo-mid* and *typo-tail* sets, or the blacklists and Alexa URLs.

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 one IP address only, while the second proxy rotates through 240 different IP addresses.

This experiment presents a couple of other differences compared to the 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 4.2.2. On the other hand, we do use the *typo-top*, *typo-mid* and *typo-tail* sets, as well as the Blacklists and Alexa URLs.

By repeating the IP-Cloaking experiment three times between June 24, 2019 and August 19, 2019, we visited pages 436,140 times to find that using multiple IP addresses, we observe significantly more malicious destination pages.

4.3 Data Labeling

Altogether, to understand potential infrastructural overlap between different illicit activities, and user differentiation in TDSs, ODIN visited 78,668 webpages over two months (June 19, 2019–August 24, 2019). Posing as different "users" (crawlers, desktop, and mobile

[1] We could equivalently have run Docker containers everywhere, but this would have caused additional complexity in data aggregation, for a questionable benefit.

Table 4.2: Summary of labels and label classes.

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

users) over different IP addresses, ODIN ended up performing 874,494 separate URL visits, from which it collected 931,551 pages,[2] which produced over 2TB of screenshots, browser events, and archived HTTP communications.

Unfortunately, we have no reliable labels telling us which pages are malicious, abusive, or illicit. To address this problem, we start by manually labeling tens of thousands of pages into fine-grained categories. We then automatically extrapolate the manual labels to the remaining 829,625 pages and create certain labels automatically. 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.

4.3.1 Original Labels

Table 4.2 summarizes the labels we use to classify destination pages ODIN visits. We use these labels to express all the different abuse we encountered during our study. We organized our labels into five label 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 [90] and storefront hosting as "illicit pharmacy." When we visit these same pages posing as a Googlebot, they often show as a page stuffed with keywords, and then we label them as "keyword stuffed." Keyword stuffing is one technique for black

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

hat SEO to manipulate search engine ranking algorithms and attract more visitors. Sites abusing affiliate programs by automatically redirecting users to advertisers are labeled as "affiliate abuse."

Regrettably, in a couple of cases we are redirected to illegal adult pages. 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 a suspicious activity, for example an otherwise empty page is asking us to enable notifications, then we 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 telling 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.

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."

We label pages where users are presented deceptive warnings or error messages, but the malicious use case is not clear, as "other malicious."

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.

4.3.2 Clustering and Data Labeling

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

We start by leveraging several approaches to cluster pages together. These methods include grouping pages by matching text, perceptual hashing [5], and clustering using the k-nearest neighbor algorithm. The k-nearest neighbor clustering uses the last layer of DenseNet 201 model trained on the ImageNet dataset from the Keras library [4] as features. Additionally, we use regular expressions based on previous work [128] 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 labelled 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 error, blocked, parked or empty labels. Such pages often have very 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 and under-developed pages.

4.3.3 Tag Extrapolation

After our manual labeling, we still have 388,168 pages in the Main experiment and 441,457 pages in the IP-Cloaking experiment that remain unlabeled. To label these pages we compare two classifiers, trained on our labeled data.

The first classifier is a one-layer neural network (NN) building on the DenseNet features introduced above. We find that the NN classifier performs best with a learning rate of 0.001, a batch size of 64,704 units in the hidden layer, and with Adam as the optimizer [82]. The second classifier is a RF (Random Forest) classifier. We compile a list of features both from related work [137] and from our domain experience. The features include perceptual hashes of the screenshots, DOM-related features (e.g., number of HTML tags, frames, and outgoing links) and text/content-related features (e.g., the ratio of text within links to total amount of text in the page). We find the Random Forest classifier performs best with `n_estimators` = 32 and `min_samples_split` = 2.

Both classifiers exhibited good performance when evaluated on a 10% validation set. The NN and RF classifiers had 97.6% and 97.7% average precision over our classes respectively. After using both models to predict labels in our unlabeled datasets, we evaluate each of them on a maximum of a hundred random samples for each label from the previously unlabeled dataset. The average precision for the NN and RF model in this second test set dropped to 88.0% and 94.9% respectively. The RF classifier performed better because it uses features other than the screenshot of the landing page allowing for a more generative model that works well on content that might look different but behaves the same way. For

the rest of the chapter we use the combination of our manual labels and results from the RF model's predictions.

4.3.4 Automatic Labeling Methodology

We next describe additional specialized classifiers and heuristic rules we use to label pages.

Illicit pharmacies. As discussed in Section 4.3.1, we consider a pharmacy to be illicit if it relies on compromised websites for hosting their storefront or automatically redirecting to the storefront. We design a separate Random Forest classifier to classify pages as illicit pharmacies. Our classifier builds on observation by previous work [90] that illicit pharmacies will respond with different web content to HTTP queries from our different user profiles. The classifier's features include the total, unique, and ratio of pharmacy related keywords, the number of domains in the redirection chain, the ratio of external content, the number and ratio of external links, the link to text ratio, the error code of the landing page, and the number of domains providing content for the page. Our classifier's precision is 99.1% and the recall is 90.8% on our manual labels. Using 200 sample pages from the predicted pages our classifier's precision remains high at 97% and the recall is 93.1%.

Defensive registrations and affiliate abuse. We label typosquatting pages as defensive if they directly redirect us to the brand owner's original domain or the domain name is hosted on a known brand protection company's name server. The number of defensive registrations is underrepresented in our dataset as we pre-filtered domains hosted on certain brand protection companies' name servers including MarkMonitor and CitizenHawk. This does not affect our results as our aim is not to characterize typosquatting brand protection. Leveraging the methodology from Chapter 3, 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 [33, 128].

File downloads. ODIN automatically downloaded 3,013 file samples with 893 unique SHA256 hashes from web pages that automatically open the browser's download prompt. We upload each file to Virus Total [26] and label the file as malicious if it appears on at least one Virus Total reported blacklist. We label a web page as a "malicious download" if we download a malicious file from that page and if we had not assigned it a different malicious label during previous labeling.

Forced social media actions. 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 then label a redirection chain in our dataset as a "forced social" if it contains at least one of these URLs.

Forced Google search analysis methodology. In Section 4.3.1, we mentioned that keyword stuffing is a known technique for black hat SEO. We find that TDSs discretely redirect users to search engines (e.g., Google) with specific search queries. We hypothesize that it is done for black hat SEO, allowing for large numbers of searches to be performed for a particular site or brand. We only consider these redirections to be "black hat SEO," if the search terms contains a domain name or a brand name together with other terms.

Impersonating pages. We labeled 1,339 pages as "potentially impersonating" based on the visual appearance of the page during manual labeling. We manually verified if the content is coming from the perceived entity or from some unknown third party by inspecting ODIN's collected HTTP(S) logs. If the content was coming from a third party, then we labeled the page as "impersonating." This leaves us with 132 manually tagged "impersonating" pages, which we then extrapolate to 1,556 pages by matching each landing URL's perceptual hash and domain.

4.3.5 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.

Redirection 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).

Domain and URL based features. 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. The 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 calculate the previously described features for the last four hops of the redirection chain. We also calculate the sum, mean, and maximum of these features across the entire redirection chain where this calculation applies.

Training a random forest classifier. Using these features, we train a random forest classifier. We train the classifier on our 103,456 manually labeled samples. We used Scikit learn Python library's random forest classifier [17] with a maximum depth of 45, maximum features of 40, minimum sample split of eight and 300 estimators.

4.4 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 on blacklist lag. Finally, we evaluate whether our proactive classifier successfully predicts when a redirection chain leads to a malicious destination page.

4.4.1 Tag 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 4.3. Tables 4.3 and 4.4 summarize the number of pages found per label class. A detailed version of our results per individual

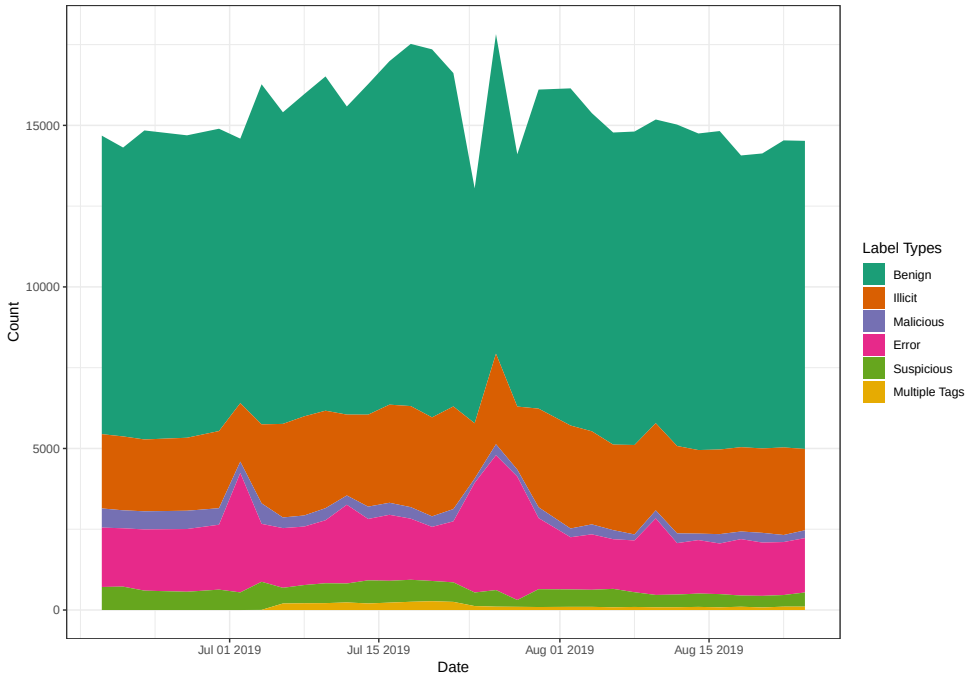


Figure 4.2: Page count, grouped by label, over time (stacked plot).

Table 4.3: Label categories per traffic source.

	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

labels can be found in Table 4.5. After removing errors, we find that 26.5% of all collected pages are malicious (2.6%), suspicious (4.0%) or illicit (20.0%).

Figure 4.2 presents the daily number of pages found per label category. A couple of error spikes induced by collection mishaps aside, label distribution is roughly constant over time.

Malice in our datasets. Figures 4.3a and 4.3b 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 4.3a clearly 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

Table 4.4: Label categories per crawl profile.

	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

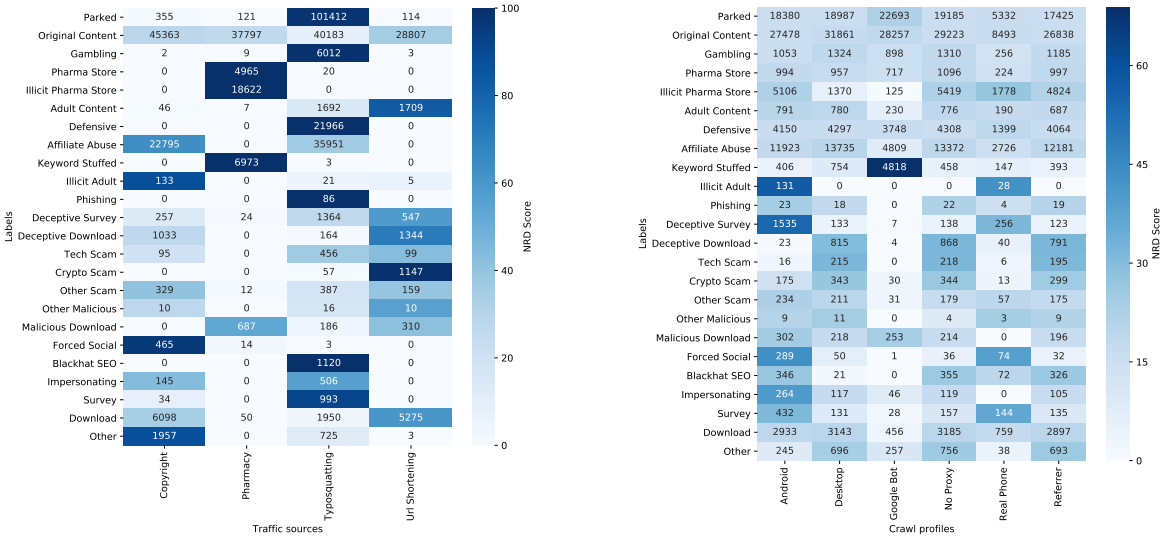
Table 4.5: Number of labels per different traffic sources and crawl profile (for data between June 19, 2019 and August 24, 2019).

	Copyright	Pharma	Typosquatting	URL Shortening	Android	Desktop	Google Bot	No Proxy	Real Phone	Referrer	All
Error	1,677 (1.85%)	4,864 (6.12%)	31,789 (11.8%)	2,995 (5.79%)	8,373 (9.16%)	8,567 (9.09%)	10,400 (11.7%)	5,661 (6.01%)	1,247 (4.27%)	7,077 (7.69%)	41,325 (8.43%)
Blocked	3,483 (3.84%)	1,641 (2.07%)	2,312 (0.86%)	5,533 (10.7%)	1,403 (1.54%)	1,710 (1.81%)	6,280 (7.06%)	1,627 (1.73%)	418 (1.43%)	1,531 (1.66%)	12,969 (2.65%)
Crawl Error	1,657 (1.83%)	1,552 (1.95%)	7,633 (2.85%)	1,245 (2.41%)	804 (0.88%)	473 (0.5%)	1,010 (1.14%)	291 (0.31%)	4,803 (16.4%)	4,706 (5.11%)	12,087 (2.47%)
Empty	4,828 (5.32%)	2,104 (2.65%)	11,034 (4.12%)	590 (1.14%)	3,307 (3.62%)	2,827 (3.0%)	3,693 (4.15%)	4,392 (4.66%)	623 (2.13%)	3,714 (4.03%)	18,556 (3.79%)
Parked	355 (0.39%)	121 (0.15%)	101,412 (37.8%)	114 (0.22%)	18,380 (20.1%)	18,987 (20.1%)	22,693 (25.5%)	19,185 (20.3%)	5,332 (18.2%)	17,425 (18.9%)	102,002 (20.8%)
Original Content	45,363 (49.9%)	37,797 (47.5%)	40,183 (14.9%)	28,807 (55.7%)	27,478 (30.0%)	31,861 (33.8%)	28,257 (31.7%)	29,223 (31.0%)	8,493 (29.0%)	26,838 (29.1%)	152,150 (31.0%)
Gambling	2 (0.0%)	9 (0.01%)	6,012 (2.24%)	3 (0.01%)	1,053 (1.15%)	1,324 (1.41%)	898 (1.01%)	1,310 (1.39%)	256 (0.88%)	1,185 (1.29%)	6,026 (1.23%)
Pharma Store	0 (0.0%)	4,965 (6.25%)	20 (0.01%)	0 (0.0%)	994 (1.09%)	957 (1.02%)	717 (0.81%)	1,096 (1.16%)	224 (0.77%)	997 (1.08%)	4,985 (1.02%)
Illicit Pharma Store	0 (0.0%)	18,622 (23.4%)	0 (0.0%)	0 (0.0%)	5,106 (5.59%)	1,370 (1.45%)	125 (0.14%)	5,419 (5.75%)	1,778 (6.09%)	4,824 (5.24%)	18,622 (3.8%)
Adult Content	46 (0.05%)	7 (0.01%)	1,692 (0.63%)	1,709 (3.31%)	791 (0.87%)	780 (0.83%)	230 (0.26%)	776 (0.82%)	190 (0.65%)	687 (0.75%)	3,454 (0.7%)
Defensive	0 (0.0%)	0 (0.0%)	21,966 (8.2%)	0 (0.0%)	4,150 (4.54%)	4,297 (4.56%)	3,748 (4.22%)	4,308 (4.57%)	1,399 (4.79%)	4,064 (4.41%)	21,966 (4.48%)
Affiliate Abuse	22,795 (25.1%)	0 (0.0%)	35,951 (13.4%)	0 (0.0%)	11,923 (13.0%)	13,735 (14.5%)	4,809 (5.41%)	13,372 (14.2%)	2,726 (9.34%)	12,181 (13.2%)	58,746 (11.9%)
Keyword Stuffed	0 (0.0%)	6,973 (8.78%)	3 (0.0%)	0 (0.0%)	406 (0.44%)	754 (0.8%)	4,818 (5.42%)	458 (0.49%)	147 (0.5%)	393 (0.43%)	6,976 (1.42%)
Illicit Adult	133 (0.15%)	0 (0.0%)	21 (0.01%)	5 (0.01%)	131 (0.14%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	28 (0.1%)	0 (0.0%)	159 (0.03%)
Phishing	0 (0.0%)	0 (0.0%)	86 (0.03%)	0 (0.0%)	23 (0.03%)	18 (0.02%)	0 (0.0%)	22 (0.02%)	4 (0.01%)	19 (0.02%)	86 (0.02%)
Deceptive Survey	257 (0.28%)	24 (0.03%)	1,364 (0.51%)	547 (1.06%)	1,535 (1.68%)	133 (0.14%)	7 (0.01%)	138 (0.15%)	256 (0.88%)	123 (0.13%)	2,192 (0.45%)
Deceptive Download	1,033 (1.14%)	0 (0.0%)	164 (0.06%)	1,344 (2.6%)	23 (0.03%)	815 (0.86%)	4 (0.0%)	868 (0.92%)	40 (0.14%)	791 (0.86%)	2,541 (0.52%)
Tech Scam	95 (0.1%)	0 (0.0%)	456 (0.17%)	99 (0.19%)	16 (0.02%)	215 (0.23%)	0 (0.0%)	218 (0.23%)	6 (0.02%)	195 (0.21%)	650 (0.13%)
Crypto Scam	0 (0.0%)	0 (0.0%)	57 (0.02%)	1,147 (2.22%)	175 (0.19%)	343 (0.36%)	30 (0.03%)	344 (0.37%)	13 (0.04%)	299 (0.32%)	1,204 (0.25%)
Other Scam	329 (0.36%)	12 (0.02%)	387 (0.14%)	159 (0.31%)	234 (0.26%)	211 (0.22%)	31 (0.03%)	179 (0.19%)	57 (0.2%)	175 (0.19%)	887 (0.18%)
Other Malicious	10 (0.01%)	0 (0.0%)	16 (0.01%)	10 (0.02%)	9 (0.01%)	11 (0.01%)	0 (0.0%)	4 (0.0%)	3 (0.01%)	9 (0.01%)	36 (0.01%)
Malicious Download	0 (0.0%)	687 (0.86%)	186 (0.07%)	310 (0.6%)	302 (0.33%)	218 (0.23%)	253 (0.28%)	214 (0.23%)	0 (0.0%)	196 (0.21%)	1,183 (0.24%)
Forced Social	465 (0.51%)	14 (0.02%)	3 (0.0%)	0 (0.0%)	289 (0.32%)	50 (0.05%)	1 (0.0%)	36 (0.04%)	74 (0.25%)	32 (0.03%)	482 (0.1%)
Black hat SEO	0 (0.0%)	0 (0.0%)	1,120 (0.42%)	0 (0.0%)	346 (0.38%)	21 (0.02%)	0 (0.0%)	355 (0.38%)	72 (0.25%)	326 (0.35%)	1,120 (0.23%)
Impersonating	145 (0.16%)	0 (0.0%)	506 (0.19%)	0 (0.0%)	264 (0.29%)	117 (0.12%)	46 (0.05%)	119 (0.13%)	0 (0.0%)	105 (0.11%)	651 (0.13%)
Survey	34 (0.04%)	0 (0.0%)	993 (0.37%)	0 (0.0%)	432 (0.47%)	131 (0.14%)	28 (0.03%)	157 (0.17%)	144 (0.49%)	135 (0.15%)	1,027 (0.21%)
Download	6,098 (6.72%)	50 (0.06%)	1,950 (0.73%)	5,275 (10.2%)	2,933 (3.21%)	3,143 (3.34%)	456 (0.51%)	3,185 (3.38%)	759 (2.6%)	2,897 (3.15%)	13,373 (2.73%)
Other	1,957 (2.16%)	0 (0.0%)	725 (0.27%)	3 (0.01%)	245 (0.27%)	696 (0.74%)	257 (0.29%)	756 (0.8%)	38 (0.13%)	693 (0.75%)	2,685 (0.55%)
All	90,762 (100.%)	79,442 (100.%)	268,041 (100.%)	53,507 (103.%)	91,654 (100.%)	94,694 (100.%)	89,006 (100.%)	94,653 (100.%)	29,254 (100.%)	92,491 (100.%)	491,752 (100.%)

pages. Surprisingly to us, ODIN downloads a significant amount of malicious files while visiting pharmaceutical-related URLs, which has not been reported by previous research. Figure 4.3b shows pretty 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 4.3 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.

Confirming previous findings [33, 128], 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



(a) Labels organized by traffic source

(b) Labels organized by crawl profile

Figure 4.3: Label counts and NRD score heatmap. The Normalized Relative Descriptive (NRD) score shows which labels are most characteristic of traffic sources (a) or crawl profiles (b).

pages, including forced Google searches, surveys (not deceptive), and financial phishing pages (further discussed in Section 4.4.3).

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

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 4.4.2, we dig deeper in whether these similar malicious landing pages are part of the same campaigns. On the other hand, the pharmaceutical ecosystem appears to be largely non-overlapping with these other activities.

Phone versus desktop users. Figure 4.3b 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.

Cloaking and bot detection. When ODIN poses as a Googlebot, it experiences very little malicious, suspicious or illicit content. This provides as 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. Only the malicious download providers seem not to try to hide their activity from crawlers. Additionally, we discover that ad-based URL shortening services frequently use Google's reCAPTCHA to stop automated crawlers, presumably in an effort to prevent discovery of questionable redirects.

We find no evidence of cloaking based on proxy detection. While not using proxies resulted in a lower error rate, this is due to errors caused by the proxies themselves. 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 [90, 92], 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 malicious pages discovered. The only exception is black-hat SEO activity, which almost always requires a referrer header field.

Table 4.6: Label categories for comparing the usage of multiple proxies versus one proxy.

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

IP-Cloaking experiment. In Table 4.6, 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 less 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. Also, 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.

Table 4.7: Domain redirection chain lengths for different labels

Label	Median	Mean	Max	St. dev.
Error	1	1.36	20	0.95
Benign	1	2.05	89	1.54
Illicit	3	3.65	31	1.85
Suspicious	4	4.56	35	2.14
Malicious	3	3.51	13	1.54
Multiple Tags	3	3.50	16	1.85

4.4.2 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.

Table 4.7 compares the difference in redirection chain lengths across label classes. Like Li et al. [96], we observe that on average users landing on a malicious, suspicious, or illicit page, have been redirected through 71% to 122% longer chains compared to when landing on a benign page.

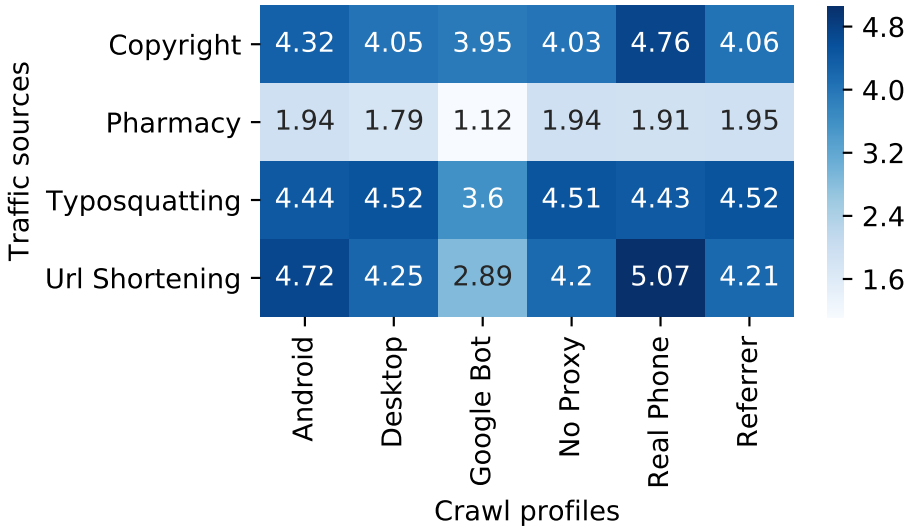
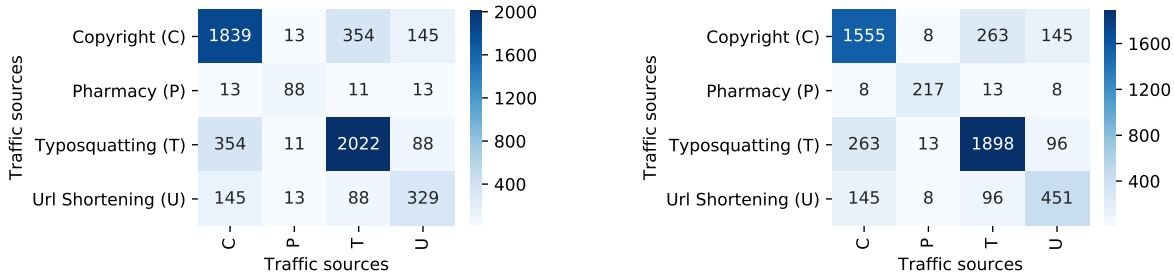


Figure 4.4: Average domain redirection chain length for different crawl profiles and traffic sources between June 19, 2019 and August 24, 2019

To better understand how our analysis of malicious TDSs is affected by different scrape and target types, we plot the average redirection chain length in Figure 4.4. The pharmacy dataset shows a much shorter average redirection chain length. Unlicensed pharmacies most of the time redirect users directly to the store from a compromised webpage. This is in

contrast with our other datasets where users are usually redirected through multiple hops to their landing page.

The Googlebot crawler experiences significantly less redirections than other agents. Conversely, phone crawlers are redirected more than the desktop crawlers. However, we do not find a significant difference in redirection chain length across different desktop crawlers.



(a) Traffic broker domain overlap

(b) Landing domain overlap

Figure 4.5: Overlap of unique malicious, suspicious or illicit traffic broker and landing registered domain names between different traffic sources.

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

In Figure 4.5a 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 conclude that often the same entities are redirecting users to malicious landing pages as **we observe a 19.2% to 44.1% traffic broker domains overlap between non-pharmacy TDSs**. We note that the overlap we observe is a lower bound of the true overlap between traffic sources as we sample URLs from them.

In Figure 4.5b 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, the typosquatting, copyright infringing and the URL shortening TDSs overlap with each other 16.9% to 32.2% of the time. **The overlap between these traffic sources are four to eight times higher compared to their overlap with illicit pharmacies.** Therefore, a significant portion of the malicious landing content is served by the same entities for all these TDSs.

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 [92, 95], 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 0.61 days), error pages (3.52 days vs. 2.84 days, $p \leq 0.01$, effect size 0.53 days), and, most interestingly, malicious pages (5.06 vs. 2.46 days,

$p < 0.01$, effect size 0.70 days), where **intermediate domains are active for more than twice as many days 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.

Table 4.8: Top malicious traffic broker domains.

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 4.9: Top malicious landing domain names. *Google only appears in this list as it is used as a target for forced searches.

Domains	Out edges	In edges	Mal. out rate	Mal. in rate	#days
`google.com`*	346	4,464	0.0	0.3004	35
`gloyah.net`	0	2,398	NaN	0.4462	21
`getadditionaloffer.com`	7	967	0.0	0.9917	3
`dwrfslsqpdfqfwy.net`	46	3,054	0.0	0.1817	35
`xterca.net`	0.0	927	NaN	0.4563	9

Top malicious domains. We list in Table 4.8 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 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.

The three short lived domains `forwrdnow.com`, `7lyonline.com` and`gonextlinkch.com` are main hubs redirecting users to deceptive downloads and scam pages. Differently, `136.243.255.89` primarily redirects users to forced Google searches. `odysseus-nua.com` redirects *only* our phone profile to deceptive surveys and illicit adult pages for the entire duration of our study.

Table 4.9 presents the top malicious destination domains. `google.com` only appears in this list because forced search engine queries redirect to the Google homepage. `gloyah.net` frequently presents users technical support scam pages, deceptive download pages and pages offering malicious downloads. `getadditionaloffer.com` mainly shows users deceptive downloads. Surprising to us, even though `dwrfslsqpdfqfwy.net`. is randomly generated, and it is a frequent participant in malicious TDSs, it has not been taken down during our study.

Table 4.10: Most malicious traffic broker domain names

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 4.11: Most malicious landing domain names. Twitter's presence is due to forced social media interactions.

Domains	Out edges	In edges	Mal. out rate	Mal. in rate	#days
getadditionaloffer.com	7	967	0.0	0.9917	3
getawesome2.com	0	486	NaN	0.9280	29
twitter.com*	2	599	0.0	0.4791	35
xterca.net	0	927	NaN	0.4563	9
gloyah.net	0	2,398	NaN	0.4462	21

High malicious rate domains. Some traffic broker and landing domains seem to entirely serve malicious redirections as shown in Table 4.10 and Table 4.11. 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.

User differentiation based on the device used. Figure 4.6 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 4.6 shows **the importance of studying user differentiation, as users visiting the same URLs might end up in very different pages depending on whether they use a phone or a desktop for browsing.**

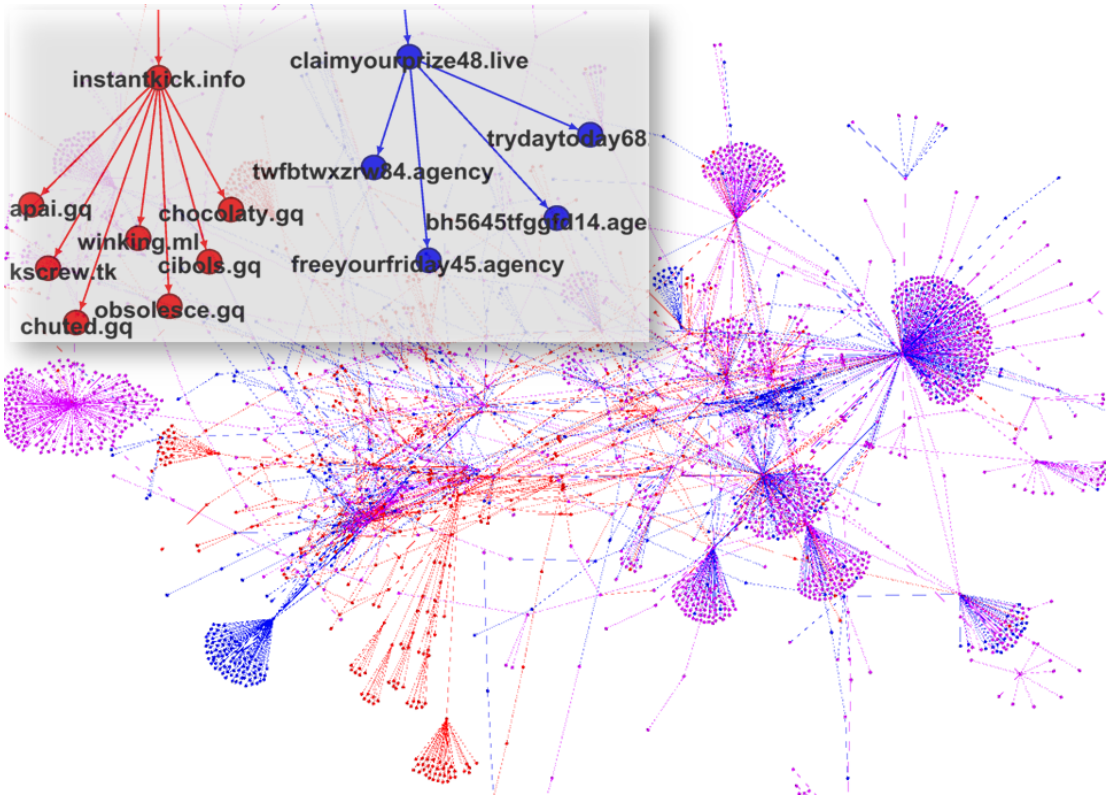


Figure 4.6: Malicious TDS redirection chains. Nodes are domain names, edges denote redirections between two domains. Blue domains were visited by our Android crawler, red domains were visited by our desktop crawler, and purple domains were visited by both crawlers.

4.4.3 Further TDS abuse analysis

Forced Google searches. In our dataset, we encounter 2,831 cases of forced Google searches through redirections. Of these, we label 2,569 or 90.75 percent as black hat SEO. Forced Google searches containing only one of the top 5 domains make up 70.4% of all queries. Domains recorded in our forced Google search dataset are not malicious and for the most part correspond to large retail and online vendors.

One interesting campaign we observe involved increasing the visibility of a damaging news story of a large corporation. Of all Google search queries recorded, 7.9 percent originated from this campaign, including key words such as "rip off," "stock," and "report." Manual search for any news regarding this corporation did not turn up any results, leading us to the conclusion this campaign was launched by somebody attempting to disparage this corporation's public image.

Impersonating pages and blacklist coverage. To determine the current coverage of phishing blacklists for traffic distribution systems, we used PhishTank [12] to scan each URL in all redirection chains we labeled as "impersonating." Out of 1,344 unique URLs, only 3 are reported as phishing, 4 are reported incorrectly as not phishing and 1,337 are not

present in PhishTank's database. Our findings show that the dearth of knowledge about traffic distribution systems negatively affects the coverage blacklists such as PhishTank provide on the URLs served by these systems.

Phishing attempts. We also discover 80 phishing pages in our dataset. We record 28 scrapes that redirects to the domain `CreditScoreAlerter.com` and 52 leading to `RewardsProgram.Onlinetpdaydea.com`. Similarities in the redirection chains suggests that these are part of the same phishing campaigns. These redirections stem from 11 typosquatting domains of several large financial institutions. Destination pages show users a warning message specific to the typosquatted financial institutions, that a data breach occurred, and they should click the link provided to ensure their information was not affected. Additionally, both domain name registrations are privacy protected suggesting that the owners do not want to be identified.

Forced social media actions. Each of the three social media sites we look at (Twitter, LinkedIn, Facebook) allow for many actions to be performed through `GET` requests. However, the actions we recorded are limited in scope. All Twitter actions attempt to post tweets, 98.6% of Facebook actions attempt to share articles, and all LinkedIn actions also attempt to share articles. Almost all actions attempt to leverage the user's social media account to advertise an illegally hosted movie on a copyright infringing website.

Table 4.12: Percentage of malicious files per URL target type

Target URL Type	**Downloads**		**Malicious**	
	Total	Unique	Total	Unique
Pharmacy	1,437	221	14 (0.97%)	14 (6.3%)
Typosquatting	968	538	682 (70.5%)	500 (92.9%)
URL Shortening	407	122	391 (96.1%)	119 (97.5%)
Copyright Infringing	18	16	13 (72.2%)	13 (81.3%)
Total	**2,830**	**897**	**1,100 (38.9%)**	**646 (72%)**

Malicious downloads. Table 4.12 displays the percentage of malicious files (according to VirusTotal) we download by target URL type. In each of the target URL categories, we find that downloaded files are malicious in the majority of the time, with 72% of all unique file downloads being malicious. URL shortening services provide the highest percentage of malicious files with 97.5% of unique files gathered from those pages reported as malicious. Pharmacy web pages provide the largest amount of downloaded files, however only a small fraction of those files are found to be malicious.

Table 4.13: The comparison between our tagging and GSB. (June 19, 2019 and July 04, 2019)

Type	Tagging	GSB				Both				FP (0 day)			
		0	+7	+30	+60	0	+7	+30	+60	Source	Inter.	Landing	Any
GoogleBot	111(0.56%)	15	18	103	120(0.60%)	0	0	0	0	15	1	6	15
Vanilla	846(3.98%)	21	250	771	785(3.70%)	3	204	398	398	15	2	7	18
Referrer	750(3.68%)	19	188	564	575(2.82%)	2	143	297	297	15	1	7	17
Referrer, no-proxy	874(4.12%)	19	257	776	793(3.74%)	4	214	402	402	15	1	5	15
Android	1,165(5.64%)	18	103	273	275(1.33%)	3	83	157	158	14	2	5	15
Total	**3,746(3.62%)**	**92**	**816**	**2,487**	**2,548(2.46%)**	**12**	**644**	**1,254**	**1,255**	**74**	**7**	**30**	**80**

4.4.4 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 [8, 14] 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. Table 4.13, shows see that only 92 pages are detected as malicious by GSB on the day of the scrape, while we label 3,746 pages as malicious. **Even with a 60-day window (i.e., we allow GSB to have up to a two-month delay), GSB finds 32% less malicious pages than we do.** While the majority of pages found to be malicious by GSB match our labels, a significant fraction of them are incorrectly classified based on our labels. The reasons for the GSB false positive classification include the dynamic nature of traffic distribution systems, which redirect users to different destination pages at each visit, and the destination pages themselves changing over time. GSB false positives fall into the error, original content, parked and gambling labels, with a secondary manual analysis confirming these results. Shortly stated, these pages change over time—they redirect to something else or became unavailable.

Delay in blacklisting. While GSB does not cover a large fraction of malice, the time it takes for a malicious page to appear on the list is also significantly delayed. This echoes findings observed in other contexts [116]. Here, we observe that approximately only a third of malicious pages are found by GSB in the first seven days, and it took 20 days for the vast majority to appear on the list. In Figure 4.7, we look at the number of days it takes a page we found to be malicious to appear on GSB's list. We found an average of seven-day delay for GSB to find our malicious pages.

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 GSB poorly understands online threats that mobile users encounter daily, while miscreants are selectively catering more malicious content towards mobile users.

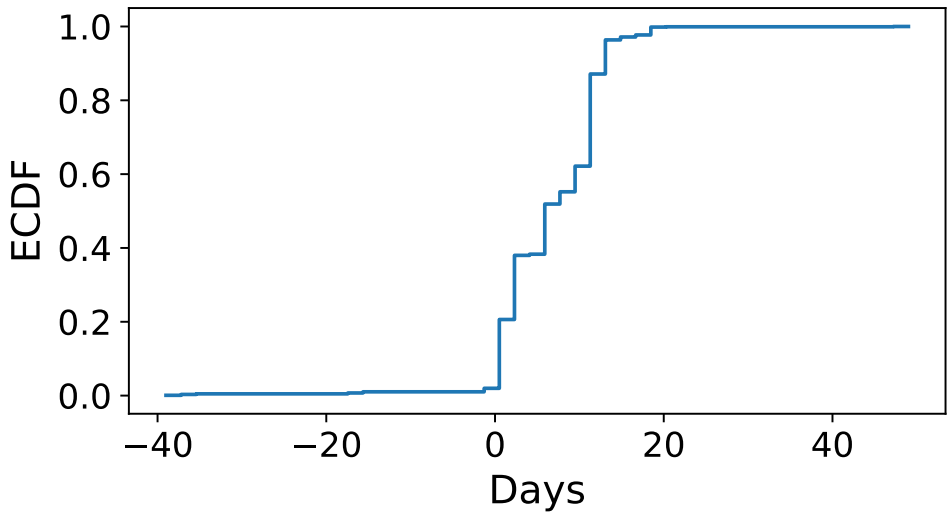


Figure 4.7: Empirical Cumulative Distribution Function of time delay for GSB detection compared to our tagging (June 19–July 04, 2019).

4.4.5 Classifier Performance

In light of the poor blacklist coverage we observed, we evaluate whether our attempt to predict a redirection chain will lead to a malicious page, using, as discussed in Section 4.3.5, a random forest classifier, holds promise.

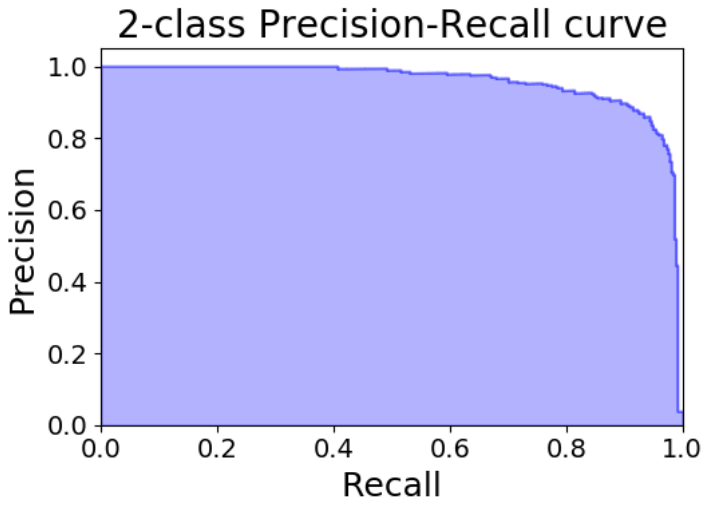


Figure 4.8: Precision-recall curve of our classifier.

We find our classifier achieves a 99.1% accuracy in labeling redirection chains as malicious or benign. However, our dataset is imbalanced with a vast majority of benign samples, so we need to to look at precision and recall as well. Figure 4.8 shows that the classifier has a large "area under the curve," with in particular a good trade-off at $(0.89, 0.9)$.

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 very 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 $(0.41, 1.0)$ 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 passing through, 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.

Even though potentially useful, WHOIS data about domain registration was not available for us, because ICANN allowed domain registrars to redact registrant information due to GDPR. Currently ICANN is in the process of developing policies to allow law enforcement and security researchers to access this data in a more controlled manner in the future [10, 11]. Future work could also consider using infrastructure based features such name server or IP address used, however these would be potentially easier for an adversary to evade.

In short, our proposed classifier achieves reasonably good performance and should be reasonably robust to evasion; as a result, it appears to be a plausible complement to blacklisting, especially considering the poor coverage of existing blacklists.

4.5 Conclusion

We introduced ODIN, a measurement infrastructure to study malicious traffic distribution systems. We ran ODIN for two months to study four different traffic sources: typosquatters, copyright-infringing movie streaming sites, ad-based URL shortening services and illicit online pharmacy websites.

While we found that each of these traffic sources send users to abusive content specific, they also often integrate their business model and send users to the same TDSs and malicious destination pages. We observed a significant amount of user-agent, referrer header field, and IP address-based cloaking. We also discovered that phone and desktop users are redirected to different malicious landing pages. Altogether, when visiting URLs posing as six different

types of crawlers, ODIN can unearth 81 percent more malicious landing pages compared to using only the most efficient crawler by itself. We also discovered popular blacklists, including GSB, strongly lack coverage of malicious pages especially those targeting mobile users.

Finally, we proposed and evaluated a classifier that can be used to prevent users whose traffic is redirected by a TDS from visiting malicious destination pages. The classifier exhibited good performance, while being reasonably hard to evade.

Chapter 5

Email Typosquatting

While we show in Chapters 3 and 4 that typosquatting is widespread and frequently exposes users to malicious content, so far in the thesis, we solely focused on web typosquatting. However, any application (e.g., email, ftp,...) relying on the domain name system for name resolution is equally vulnerable to domain typosquatting, and consequences may be more dire than with website typosquatting.

This chapter[1] presents the first in-depth measurement study of email typosquatting. Working in concert with our IRB, we registered 76 typosquatting domain names to study a wide variety of user mistakes, while minimizing the amount of personal information exposed to us. In the span of over seven months, we received millions of emails at our registered domains. While most of these emails are spam, we infer, from our measurements, that every year, three of our domains should receive approximately 3,585 "legitimate" emails meant for somebody else. Worse, we find, by examining a small sample of all emails, that these emails may contain sensitive information (e.g., visa documents or medical records).

We then project from our measurements that 1,211 typosquatting domains registered by unknown entities receive in the vicinity of 800,000 emails a year. Furthermore, we find that millions of registered typosquatting domains have MX records pointing to only a handful of mail servers. However, a second experiment in which we send "honey emails" to typosquatting domains only shows very limited evidence of attempts at credential theft (despite some emails being read), meaning that the threat, for now, appears to remain theoretical.

5.1 Introduction

Domain typosquatting is the act of registering a domain name very similar to an existing, legitimate, domain, in an effort to capture some of the traffic destined for the original domain. Domain typosquatting exploits the propensity of users to make typographical errors when typing domain names—as opposed to clicking on links—and is frequently used for financial profit. For instance, somebody registering `googe.com` would immediately

[1]This chapter is primarily based on our paper published at the 2017 Internet Measurement Conference [126]

receive large amounts of traffic meant for `google.com`. That traffic could then in turn be monetized, by showing ads or setting up drive-by-downloads. Domain typosquatting has been shown to be profitable [52, 106], while requiring no technical skill.

In some jurisdictions, domain typosquatting is considered illegal, and may trigger trademark infringement cases.[2] In 1999, ICANN, the authority which regulates domain names on the Internet, created the Uniform Domain Name Dispute Resolution Policy (UDRP) as a solution for trademark owners to claim cybersquatting or typosquatting domain names [64].

Thus far, most of the studies in the related literature have solely focused on web typosquatting, that is, domain typosquatting used to illicitly acquire "page views." However, domain typosquatting can be equally used with other target applications—`ssh`, `ftp`, email, and so forth.

This paper is the first in-depth study to focus on *email typosquatting*, in which miscreants could register domain names mimicking those of large email providers to capture emails. Even though typing mistakes may be fairly rare, typosquatting a large email provider (e.g., `gmail.com`) could remain a profitable endeavor by virtue of the number of emails passing through the service. Indeed, while most emails illicitly received would be of limited use to the attacker, some could contain sensitive information that could yield large payoffs for the attacker, and cause considerable losses to the victim.

We put this hypothesis to the test in this paper. Specifically, we register 76 email typosquatting domains, collect data from these domains for more than seven months (June 4, 2016–January 15, 2017), and—working in concert with our Internal Review Board (IRB)—design a protocol to process the emails we receive to determine the potential harm email domain typosquatting might inflict on users, as well as its potential benefits to attackers (Section 5.3). We discover that a number of actors already have the infrastructure necessary for bulk email domain typosquatting (Section 5.4). Extrapolating from our observations through regression analysis (Section 5.5), we find that setting up the necessary infrastructure costs attackers only in the order of a couple of cents per email, and that they can expect to receive hundreds of thousands of emails over a few months. However, by actively sending "honey emails" containing credentials, we discover, that even though a lot of these emails are accepted, they are not actually read (Section 5.6), meaning that email typosquatting does not appear, for now, to be monetized.

5.2 Terminology

Typosquatting actually involves a number of different concepts, which we discuss here.

Distance metrics We use two metrics to characterize the distance between various domain names. The *Damerau-Levenshtein distance* [49] is the minimum number of operations (deletion, addition, substitution, or transposition of two neighboring characters). Papers on typosquatting often rely on Damerau-Levenshtein distance of one ("DL-1") to detect typosquatting domains. Moore and Edelman define the *fat-finger distance* as "the minimum

[2]See, e.g., in the U.S., the Federal Trademark Dilution Act, or FTDA, and the Anti-cybersquatting Consumer Protection Act, ACPA.

number of insertions, deletions, substitutions or transpositions using letters adjacent on a QWERTY keyboard to transform one string into another." [106] A fat-finger distance of one (FF-1) implies a DL-1 distance. The *visual distance* measures how different the mistyped character *looks* compared to the original character. We use a set of heuristic rules to compute the visual distance, which incorporate how confusing alphabet letters with numbers (e.g., "o" and "O," "1" and "l") is more likely to happen than confusing two (different) letters or numbers.

Typosquatting domains The *target domain name* refers to any domain name targeted by typosquatters. Previous work on web typosquatting usually relies on Alexa rankings [1] to identify target domains.

We adopt Szurdi et al.'s taxonomy [128] to clearly differentiate lexically close domains from true typosquatting domain names. *Generated typo domains* ("gtypos") are "domain names which are lexically similar (e.g. at DL-1) to some set of target domains." *Candidate typo domains* ("ctypos") are "the subset of registered domains within the gtypo set which have been registered." Finally, *typosquatting domains* are candidate typo domains that "(i) [were] registered to benefit from traffic intended for a target domain," and "(ii) that [are] the property of a different entity."

Misdirected email taxonomy Typosquatting of email domains allows an attacker to capture a number of different emails. First, *receiver typo* emails are simply sent to the wrong address by the sender mistyping the recipient's email address. We only focus on typos in the domain name, and leave the issue of typos in the recipient name to future work.[3]

We also consider *reflection typo* emails. Those emails are the result of users mistyping their email address when registering for an online service. As a consequence, emails from the service are subsequently sent to the wrong address. While the harm caused would be likely negligible in the case of an online raffle, providing the wrong address to a financial services company might lead to leaks of confidential or sensitive information.

Last, we capture a completely different type of error with *SMTP typo* emails, which result from a user mistyping their SMTP settings in their email client. This type of error is pernicious, as *all* emails sent by the victim may be intercepted until the typo is fixed.

5.3 In the Shoes of a Typosquatter

In this section we describe a seven-month experiment (June 4, 2016–January 15, 2017), during which we acted as email typosquatters ourselves, in an effort to gain insights into whether email typosquatting could be a potential problem or not. The idea is simple: by registering typosquatting domains, we can simply count the number of emails these domains—which we absolutely did not advertise or otherwise use, to avoid measurement confounds—receive, and infer whether email typos occur frequently or not, and if so, which kind of typos seem more prevalent than others. This analysis will later be useful in

[3]For instance, we consider `alice@gmial.com`, but not `aliec@gmail.com`.

attempting to derive more general projections, beyond the set of domains we registered, on the potential magnitude of the problem overall.

Because we are ultimately acting as attackers, our experimental setup is driven by ethical considerations. We start with a discussion of these ethical objectives, before turning to how our collection methodology attempts to fulfill these objectives. We then analyze the results of our data collection.

5.3.1 Ethical challenges and how to address them

Registering a set of typosquatting domain names ourselves provides a very precise view of the type of information users may accidentally leak. At the same time, 1) we need to tread carefully with possible trademark infringement, and, even more importantly, 2) we can potentially receive users' personal information.

Both issues are very serious and led us to design our protocol with the collaboration of our university's Internal Review Board (IRB), in an effort to minimize the risk to users, and to ourselves. The protocol was approved by our IRB, and our sponsor's IRB, before we started our experiments.

The trademark infringement part—which actually does not impact any users but us—was relatively quickly settled. We agreed to surrender any domain we registered to the legitimate owner of a trademark it could potentially infringe upon simple request. To date, we have not received any such requests.

While we elected to keep emails accidentally sent to our domains to carry out deeper analyses than could be done by simply keeping headers, we take three measures to protect the users who sent these emails. First, our storage infrastructure consists of a hardened server accessible only from our university network. Second, we automatically remove sensitive information using regular expression matching prior to storage. Finally, we encrypt all emails prior to storing them, using an encryption key kept *separately* from the server (i.e., on removable storage). To result in potential harm, accidental disclosure of the contents of the server would need to be accompanied by a leakage of our encryption key.

Even though our IRB protocol allows us to look at the content of the emails we receive, provided that we do our best effort to automatically sanitize personal identifiers prior to doing so, we wanted to minimize as much as possible such interactions. Initially, we were hoping to be able to derive the content of these emails purely programmatically—i.e., inferring the presence of leaks from regular expression matching on the body, classification of attachment names, etc. However, we received an enormous amount of spam email, which made it important to fine-tune and evaluate the spam filtering system we used. We eventually settled on looking at a small sample of 103 emails (out of several millions we received overall) that were classified as non-spam to evaluate the performance of our spam classifier, which is absolutely crucial to the rest of our analysis due to the imbalance of our dataset.

In other words, we adopted a utilitarian ethics view—while it is undesirable (but permitted by our IRB-approved protocol) to look at some of these email contents, we were satisfied that the small minority of emails we were manually analyzing would 1) not result in any risk to the users who sent (or were meant to receive) these emails, while 2) giving us

stronger confidence in our results. We re-emphasize that potentially sensitive information (e.g., credit card numbers) was automatically scrubbed *prior* to our looking at these 103 emails.

5.3.2 Collection methodology

We next turn to discussing how we selected a set of domains to register, before delving into the details of our collection infrastructure. We then explain how we post-processed the data we acquired by presenting the layered filtering system we built to remove spam from our corpus.

Domain registration

Table 5.1: Target domains and their attributes

Domain type	company	Target domain	2014.07.13 Global Alexa R.	2016.09.19 Global Alexa R.	2014.07.13 Company's Email Alexa R.	2016.09.19 Company's Email Alexa R.
banking	Chase	smtp.chase.com	171	122	NA	NA
banking	Paypal	smtp.paypal.com	39	50	NA	NA
email service	10minutemail	10minutemail.com	7917	14552	28	22
email service	Aol	mx.aol.com	121	198	NA	NA
email service	Aol	smtp.aim.com	20623	35653	NA	NA
email service	Aol	smtp.aol.com			NA	NA
email service	GMX	mail.gmx.net	291	316	7	12
email service	GMX	smtp.gmx.com	NA	6430	7	12
email service	GMX	smtp.gmx.net			7	12
email service	Google	gmail.com	145	5023	1	1
email service	Google	imap.gmail.com	145	5023	1	1
email service	Google	imap.gmail.com	145	5023	1	1
email service	Google	smtp.gmail.com			1	1
email service	Hushmail	smtp.hushmail.com	7891	16097	27	25
email service	Mailchimp	mailchimp.com	126	300	4	4
email service	Mailchimp	smtp.mailchimp.com			4	4
email service	Microsoft	hotmail.com	13268	44247	3	3
email service	Microsoft	imap.live.com	12	11	3	3
email service	Microsoft	imap-mail.outlook.com	645	3165	3	3
email service	Microsoft	mx1.hotmail.com			3	3
email service	Microsoft	outlook.com			3	3
email service	Microsoft	smtp.live.com			3	3
email service	Microsoft	smtp-mail.outlook.com			3	3
email service	Myway	smtp.myway.com	6762	126	NA	NA
email service	Rediffmail	smtp.rediffmailpro.com	15812	5983	6	5
email service	Sendgrid	sendgrid.com	3845	9097	17	16
email service	Yahoo	am0.yahoodns.net	NA	NA	2	2
email service	Yandex	smtp.yandex.com	2132	1492	NA	NA
email service	Yopmail	yopmail.com	13696	16063	36	24
email service	Zohomail	mx.zohomail.com	NA	NA	NA	NA
email service	Zohomail	smtp.zoho.com	591	413	NA	NA
email service	Zohomail	zohomail.com			NA	NA
Generic	Apple	mail.icloud.com	1000	341	443	389
Generic	Apple	mail-in2.apple.com	47	389	443	389
Generic	Apple	mail-in4.apple.com	47	389	443	389
ISP	Att	seg.att.com	465	376	NA	NA
ISP	Centurylink	smtp.centurylink.net	5945	4266	NA	NA
ISP	Comcast	comcast.com	846	6048	NA	NA
ISP	Comcast	mx1.comast.com	846	6048	NA	NA
ISP	Comcast	smtp.comcast.net			NA	NA
ISP	Cox	smtp.cox.net	1512	2040	NA	NA
ISP	TWC	email.rr.com	1744	1969	NA	NA
iSP	Verizon	outgoing.verizon.net	3459	14758	NA	NA
ISP	Verizon	smtp.verizon.net			NA	NA
ISP	Verizon	verizon.com			NA	NA

Table 5.2: Attributes of typosquatting domains registered by us

Domain type	Company	Target domain	Typosquatting domain	Is FatFinger?	IS BitSquatting?	Protocol tested	Place of error	DL typo operation
banking	Chase	smtp.chase.com	smtpchase.com	T	F	smtp	smtp	deletion
banking	Paypal	smtp.paypal.com	smtppaypal.com	T	F	smtp	smtp	deletion
email service	10minutemail	10minutemail.com	10inutemail.com	T	F	any	rcvr,smtp,imap,pop3	deletion
email service	Aol	mx.aol.com	mxnaol.com	F	T	mail	smtp,imap,pop3	substitution
email service	Aol	smtp.aim.com	smtpaim.com	T	F	smtp	smtp	deletion
email service	Aol	smtp.aol.com	smtpaol.com	T	F	smtp	smtp	deletion
email service	GMX	mail.gmx.net	mailgmx.net	T	F	mail	smtp,imap,pop3	deletion
email service	GMX	smtp.gmx.com	smtpgmx.com	T	F	smtp	smtp	deletion
email service	GMX	smtp.gmx.net	smtpgmx.net	T	F	smtp	smtp	deletion
email service	Google	gmail.com	gmai-l.com	F	F	any	rcvr,smtp,imap,pop3	addition
email service	Google	gmail.com	gmaiql.com	F	F	any	rcvr,smtp,imap,pop3	addition
email service	Google	imap.gmail.com	imaplgmail.com	T	F	imap	imap	substitution
email service	Google	imap.gmail.com	imapngmail.com	F	T	imap	imap	substitution
email service	Google	smtp.gmail.com	smtpngmail.com	F	T	smtp	smtp	substitution
email service	Google	smtp.gmail.com	smtplgmail.com	T	F	smtp	smtp	substitution
email service	Hushmail	smtp.hushmail.com	smtphushmail.com	T	F	smtp	smtp	deletion
email service	Mailchimp	mailchimp.com	nmailchimp.com	T	F	any	rcvr,smtp,imap,pop3	addition
email service	Mailchimp	smtp.mailchimp.com	smtpmailchimp.com	T	F	smtp	smtp	deletion
email service	Microsoft	hotmail.com	ho6mail.com	T	F	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	hotmail.com	hovmail.com	F	T	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	imap.live.com	imapllive.com	T	F	imap	imap	substitution
email service	Microsoft	imap.live.com	imapnlive.com	F	T	imap	imap	substitution
email service	Microsoft	imap-mail.outlook.com	imap-mailoutlook.com	T	F	imap	imap	deletion
email service	Microsoft	mx1.hotmail.com	mx1hotmail.com	T	F	mail	smtp,imap,pop5	deletion
email service	Microsoft	mx1.hotmail.com	mx1nhotmail.com	F	T	mail	smtp,imap,pop6	substitution
email service	Microsoft	mx2.hotmail.com	mx2hotmail.com	T	F	mail	smtp,imap,pop7	deletion
email service	Microsoft	mx2.hotmail.com	mx2nhotmail.com	F	T	mail	smtp,imap,pop8	substitution
email service	Microsoft	mx3.hotmail.com	mx3hotmail.com	T	F	mail	smtp,imap,pop9	deletion
email service	Microsoft	mx3.hotmail.com	mx3nhotmail.com	F	T	mail	smtp,imap,pop10	substitution
email service	Microsoft	mx4.hotmail.com	mx4hotmail.com	T	F	mail	smtp,imap,pop11	deletion
email service	Microsoft	mx4.hotmail.com	mx4nhotmail.com	F	T	mail	smtp,imap,pop12	substitution
email service	Microsoft	outlook.com	o7tlook.com	T	F	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	oetlook.com	F	T	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	ohtlook.com	T	F	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	ou6look.com	T	F	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	outlo0k.com	T	F	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	outmook.com	F	T	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	ouulook.com	F	T	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	outlook.com	ouvlook.com	F	T	any	rcvr,smtp,imap,pop3	substitution
email service	Microsoft	smtp.live.com	smtpllive.com	T	F	smtp	smtp	substitution
email service	Microsoft	smtp.live.com	smtpnlive.com	F	T	smtp	smtp	substitution
email service	Microsoft	smtp-mail.outlook.com	smtp-mailoutlook.com	T	F	smtp	smtp	deletion
email service	Myway	smtp.myway.com	smtpmyway.com	T	F	smtp	smtp	deletion
email service	Rediffmail	smtp.rediffmailpro.com	smtprediffmailpro.com	T	F	smtp	smtp	deletion
email service	Sendgrid	sendgrid.com	sendgri.com	T	F	any	rcvr,smtp,imap,pop3	deletion
email service	Yahoo	am0.yahoodns.net	am0nyahoodns.net	F	T	mail	smtp,imap,pop3	substitution
email service	Yandex	smtp.yandex.com	smtpyandex.com	T	F	smtp	smtp	deletion
email service	Yopmail	yopmail.com	yopail.com	T	F	any	rcvr,smtp,imap,pop3	deletion
email service	Zohomail	mx.zohomail.com	mxnzohomail.com	F	T	mail	smtp,imap,pop3	substitution
email service	Zohomail	smtp.zoho.com	smtpzoho.com	T	F	smtp	smtp	deletion
email service	Zohomail	zohomail.com	zohomial.com	T	F	any	rcvr,smtp,imap,pop3	transposition
email service	Zohomail	zohomail.com	zohomil.com	T	F	any	rcvr,smtp,imap,pop3	deletion
Generic	Apple	mail.icloud.com	mailnicloud.com	F	T	mail	smtp,imap,pop3	substitution
Generic	Apple	mail-in2.apple.com	mail-in2napple.com	F	T	mail	smtp,imap,pop3	substitution
Generic	Apple	mail-in4.apple.com	mail-in4apple.com	T	F	mail	smtp,imap,pop3	deletion
ISP	Att	seg.att.com	segnatt.com	F	T	mail	smtp,imap,pop12	substitution
ISP	Centurylink	smtp.centurylink.net	smtpcenturylink.net	T	F	smtp	smtp	deletion
ISP	Comcast	comcast.com	coicast.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Comcast	comcast.com	comaast.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Comcast	comcast.com	comca3t.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Comcast	comcast.com	comcas5.com	T	F	any	rcvr,smtp,imap,pop3	substitution
ISP	Comcast	comcast.com	comcasu.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Comcast	comcast.com	comcawst.com	T	F	any	rcvr,smtp,imap,pop3	addition
ISP	Comcast	mx1.comast.com	mx1ncomast.com	F	T	mail	smtp,imap,pop4	substitution
ISP	Comcast	smtp.comcast.net	smtpcomcast.net	T	F	smtp	smtp	deletion
ISP	Cox	smtp.cox.net	smtpcox.net	T	F	smtp	smtp	deletion
ISP	TWC	email.rr.com	emailnrr.com	F	T	mail	smtp,imap,pop3	substitution
iSP	Verizon	outgoing.verizon.net	outgoingverizon.net	T	F	smtp	smtp	deletion
ISP	Verizon	smtp.verizon.net	smtpverizon.net	T	F	smtp	smtp	deletion
ISP	Verizon	verizon.com	evrizon.com	T	F	any	rcvr,smtp,imap,pop3	transposition
ISP	Verizon	verizon.com	ve5izon.com	T	F	any	rcvr,smtp,imap,pop3	substitution
ISP	Verizon	verizon.com	vebizon.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Verizon	verizon.com	vepizon.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Verizon	verizon.com	verhzon.com	F	T	any	rcvr,smtp,imap,pop3	substitution
ISP	Verizon	verizon.com	verizo0n.com	T	F	any	rcvr,smtp,imap,pop3	addition
ISP	Verizon	verizon.com	vermzon.com	F	T	any	rcvr,smtp,imap,pop3	substitution

When deciding on which domain names to register, we had a number of constraints to satisfy, and three main objectives in mind.

Constraints Our first constraint is budgetary. While registering individual domains is reasonably cheap, in the order of $8–$20 per year depending on the registrar and top-level-domain being used, it is potentially time-consuming, and we have to limit ourselves to at most a couple of hundred domains. Our second constraint, which is far more serious, is that of availability. Unfortunately a number of the most interesting typo domains are already registered (either by the trademark owners themselves, or by typosquatters), so that we were forced to choose from what is available. However, the set of gtypos is a powerset of the set of target domains. In particular, for the top 10,000 domains according to Alexa rankings, there are millions of gtypos. Even though hundreds of thousands are already registered, we are still able to select a few dozen typosquatting domains that can hopefully produce representative outcomes.

Objectives When we undertook this study, we had absolutely no idea of the amount of emails we would receive. Our first goal was thus to find typo domains that could be trusted to provide a representative, and measurable signal, if anything was to be measured. Our second goal was to compare different DL-1 typing mistakes (e.g., deletion and substitution), to be able to reason about respective impact of such mistakes. Third, we wanted to register a corpus of domains that would allow us to measure the different kinds of typos (receiver, SMTP, reflection) we had identified.

Strategy To maximize the probability of receiving emails, we aimed to register typo domains targeting some of the most popular domains. To that effect, we selected target domains with a small Alexa rank in the email category (i.e., popular domains for email). To prune down the list of domains we register, most of the typo domains we generated have a fat-finger distance of one from the target domain.

This led us to select domains targeting top email providers such as Google, Microsoft, Yahoo, Apple, and Mailchimp. We complemented this list with some of the "second tier" e-mail providers such as Rediffmail Pro, GMX, AOL, Hushmail and ZohoMail.

We hypothesized that we would see more reflection typos on domains that advertise "disposable," instant email addresses. Accordingly, we registered typos of the 10 Minute Mail (`10minutemail.com`) and YOPmail (`yopmail.com`) domains.

To assert the risks linked to SMTP typos, we also registered typos linked to some of the most popular Internet Service providers which offer SMTP service to their users: AT&T, Comcast, Cox, TWC and Verizon.

We chose Paypal and Chase as potential sensitive (financial) domains and registered a few domains targeting SMTP typos on these domains.

For each of the target domains, we registered multiple typo domains to compare how different typing mistakes impact the amount of email received.

The complete list of 76 domains we registered, as well as additional information about these domains and the targeted brand domains are shown in Tables 5.1 and 5.2.

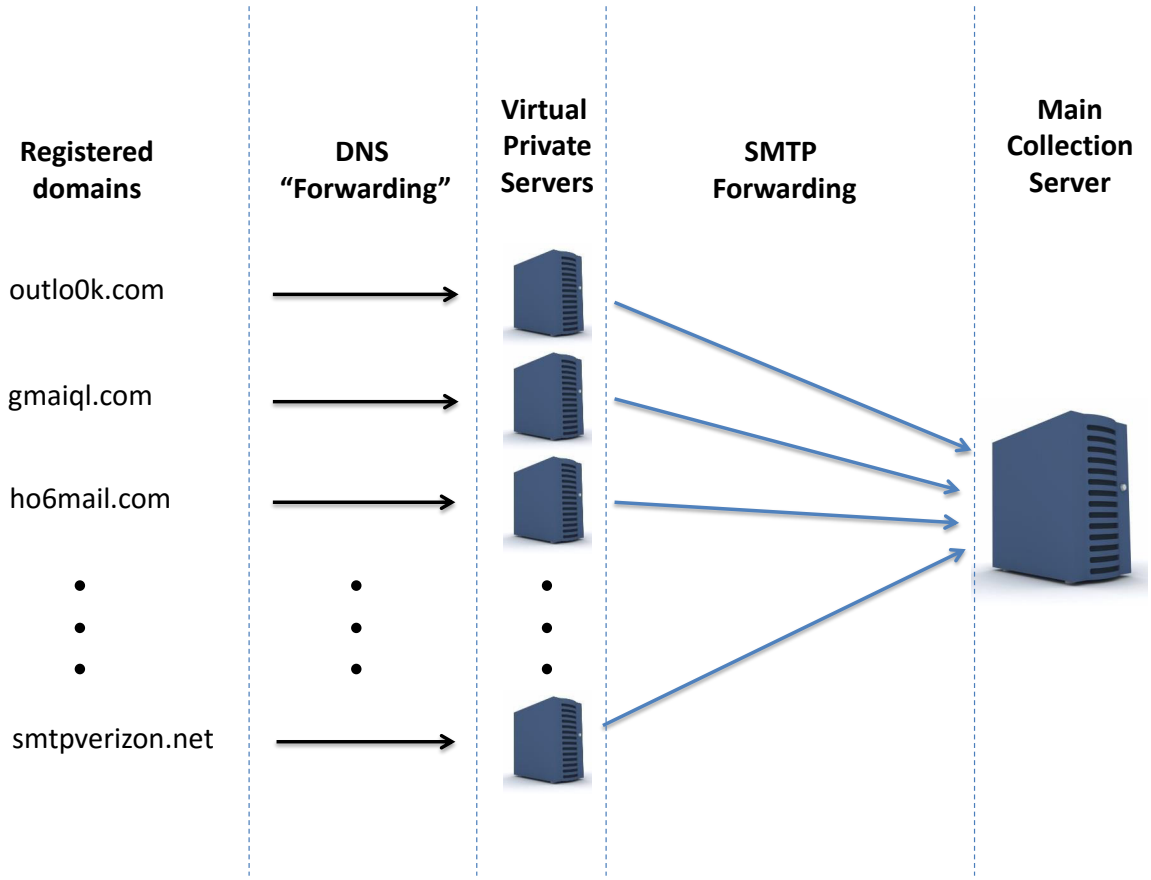


Figure 5.1: The design of the typo email collection infrastructure

Collection infrastructure

Figure 5.1 shows a high-level overview of our data collection infrastructure. Each typo domain is assigned a different Virtual Private Server, which in turn forwards the data to our main collection server. This allows us to eschew a potential (but unlikely) issue, of people spamming us from looking up domains and flagging us as security researchers. In addition, to distinguish between different SMTP typo mistakes, we used a one-to-one mapping of our domain names to virtual private server IP addresses. This is because the SMTP protocol does not require the domain name of the SMTP server contacted to be included in the headers. We thus have to differentiate domains by IP addresses.

Table 5.3 shows our DNS settings for each domain we registered. We include wildcard subdomains to collect typo domains sent to any subdomains of the domains we registered. We run Postfix on our main collection server, which we configure to accept any email sent to any email address. The username and the domain name can thus both be random strings. Our collection server never sends any email out, but ultimately forwards these emails to a processing and storage server (not represented in the picture).

Table 5.3: DNS settings for an example typo domain.

FQDN	TTL	TYPE	priority	record
*.exampel.com.	300	MX	1	exampel.com.
exampel.com.	300	MX	1	exampel.com.
*.exampel.com.	300	A	NA	1.1.1.1
exampel.com.	300	A	NA	1.1.1.1

Email processing pipeline Figure 5.2 describes this email processing pipeline. When we receive an email we first feed it into SpamAssassin [2]. We do not discard email identified as spam, and instead simply flag it as such. We then tokenize the email into header, body and attachments, save header information, and run both the body and any attachments through a text extraction module (Textract [20]), which operates on a variety of different file formats, even performing optical character recognition on some image files.

Filtering out sensitive information We send the text output into a filtering system based on regular expression matching. The idea is to flag when sensitive information is found in an email, while immediately discarding it to protect user privacy. We use the HIPAA list of personal identifiers [9] as a baseline for our set of sensitive information. We replace personal identifiers by salted hashes whenever possible; as an added precaution, we replace *all* digits in the text by zeroes.

Table 5.4: Precision and Sensitivity of our regular expression based filtering module.

Sensitive info	F1-score	Prec.	Sens.
Credit card number	0.96	0.93	1.00
Social Security number	0.88	0.78	1.00
Employer id. number	0.94	0.89	1.00
Password	0.50	0.33	1.00
Vehicle id. number	1.00	1.00	1.00
Username	0.74	0.59	1.00
Zip	1.00	1.00	1.00
Identification number	0.67	0.75	0.60
Email address	0.99	1.00	0.98
Phone number	0.89	0.83	0.95
Date	1.00	1.00	1.00

We use the public Enron email corpus [6] (May 7, 2015 version) to test how well our regular expression matching heuristics are performing. Table 5.4 shows the precision (ratio of true positives over true and false positives) and sensitivity (ratio of true positives over true positives and false negatives) for each type of sensitive information. In our context, these metrics are more useful than the more widely used "accuracy" metric. Indeed, because the number of emails containing private identifiers is small overall (and indeed, this is also

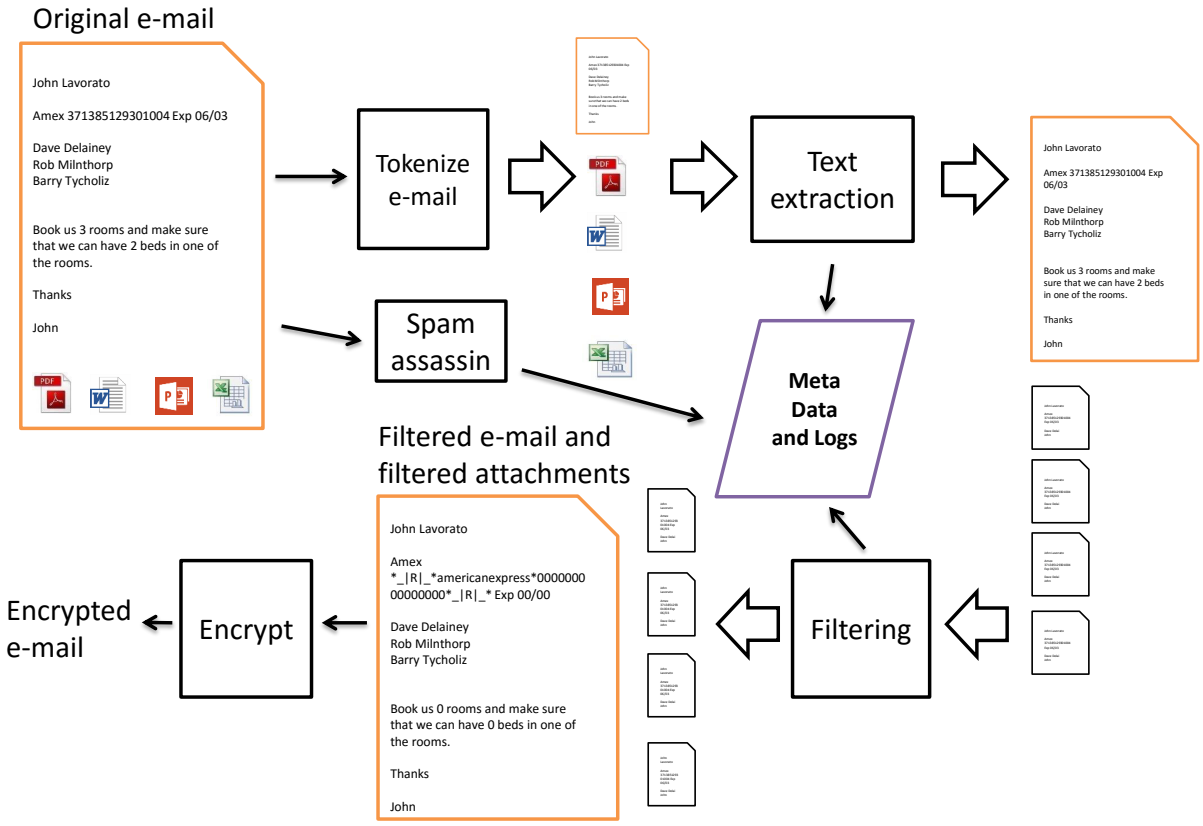


Figure 5.2: The typo email filtering system used.

true of the Enron corpus), we have an imbalanced dataset; as a result, an algorithm that always outputs "no sensitive information was found" would have a high accuracy.

Each score in Table 5.4 is computed based on sampling 20 random emails per type of sensitive information found in the dataset (except for social security numbers, for which we only had 13 examples available), manually labeling them, and comparing them to what our algorithm produced. The results show a high recall for most sensitive information, except for Identification numbers. The sensitivity for identification numbers is low, because our definition of an identification number is very broad. To validate our results further (beyond the biased sample produced by our algorithm), we sampled an additional 100 random emails from the Enron dataset and manually labeled them. Due to the imbalanced nature of sensitive data, we only found phone numbers, emails and dates in this sample. The sensitivity remains high however—0.91, 1.00 and 0.98 for phone, date and email respectively.

Once all of this processing is done, we encrypt each part (header, body, attachment) and most of the log files for storage on our collection server.

5.3.3 Email classification

After running our experiment for a few days, it became obvious we were receiving very large amounts of spam, which would completely bias any analysis if left unfiltered. Spam can come from miscreants noticing our servers accept any email (even though they don't relay to any party but our collection server), or from users mistyping their own email address (reflection typo) and being subsequently added to promotional lists. Some of our domains might have also been previously registered, and could still appear in certain promotional lists.

We thus turned to building a filtering and classification module, which not only filters out spam, but also classifies reflection typo emails that result from a single typo (e.g., making a typo while signing up for a mailing list). Our classification module consists of five layers, which act as a funnel: each email marked as spam in a given layer is not further considered.

Layer 1: Detecting erroneous header fields Emails in which the name of the SMTP server relaying the mail to our collection server does not match the name of one of our registered domains is immediately classified as spam. The sender's address should also not belong to one of our domains, since we do not send any email. Conversely, spammers often pose as sending from the same domain as the intended recipient. Thus, any email in which the sender appears to be one of our domains is classified as spam. In receiver or reflection typo emails (but not in SMTP typo emails), the recipient's email address should belong to one of our typo domains.

Table 5.5: Evaluation of Spamassassin on four datasets

Dataset	Precision	Recall
TREC [21]	0.98	0.79
CSDMC [3]	0.98	0.87
SpamAssassin [2]	0.97	0.84
Untroubled [23]	–	0.23

Layer 2: SpamAssassin We run SpamAssassin on all incoming email. Table 5.5 shows our evaluation of SpamAssassin in local mode with the default thresholds on four different datasets. While precision is good, the low recall indicates we need additional filtering. We immediately remove all emails with ZIP or RAR attachments and consider them as spam—we indeed receive large amounts of such emails, and every single one of them we manually inspected was spam.

Layer 3: Collaborative spam filtering If a sender sends us spam once, we consider all of the emails from that sender, across all of our domains, to be spam. Furthermore we apply bag-of-words analysis to the email body. If the analysis yields more than 20 words, we flag all other emails with a matching bag-of-words as spam. This filtering step should have high precision, because it is highly unlikely that two emails would be spam and ham, respectively, if both emails use the same corpus of words.

Layer 4: Detecting reflection typos Emails that have survived the first three layers might not be spam, but still be the product of automated systems. For instance, a user might have made a typo while signing up for a certain service, and subsequently received notifications to that erroneous address. We automatically classify these emails, using a set of regular expression heuristics. If an "unsubscribe-list" header field is present; "bounce" or "unsubscribe" appears in the `Sender:`, `From:`, or `Reply-To:` fields; or if any two of `From:`, `Reply-To:`, or `Return-Path:` have different values, we classify the email as a reflection typo. We additionally search for strings including "unsubscribe," "remove yourself," and other similar content in the body to flag email containing such strings as reflection typos. Finally, we also filter out emails sent from system users, e.g., "postmaster," "root," or "admin."

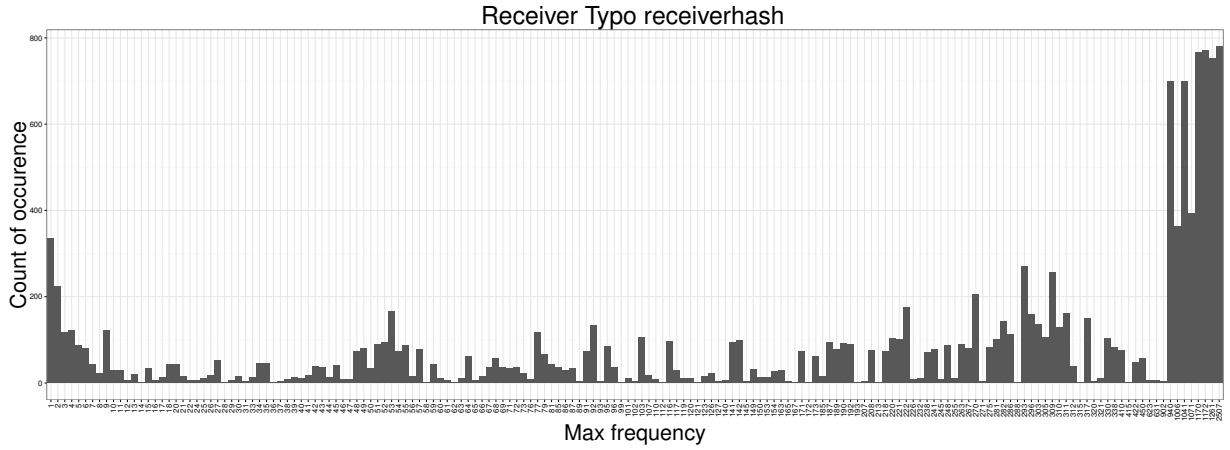


Figure 5.3: X axis shows the frequency of a receiver hash in our email corpus. Y axis show the number of receiver typo emails which has a receiver hash with that frequency.

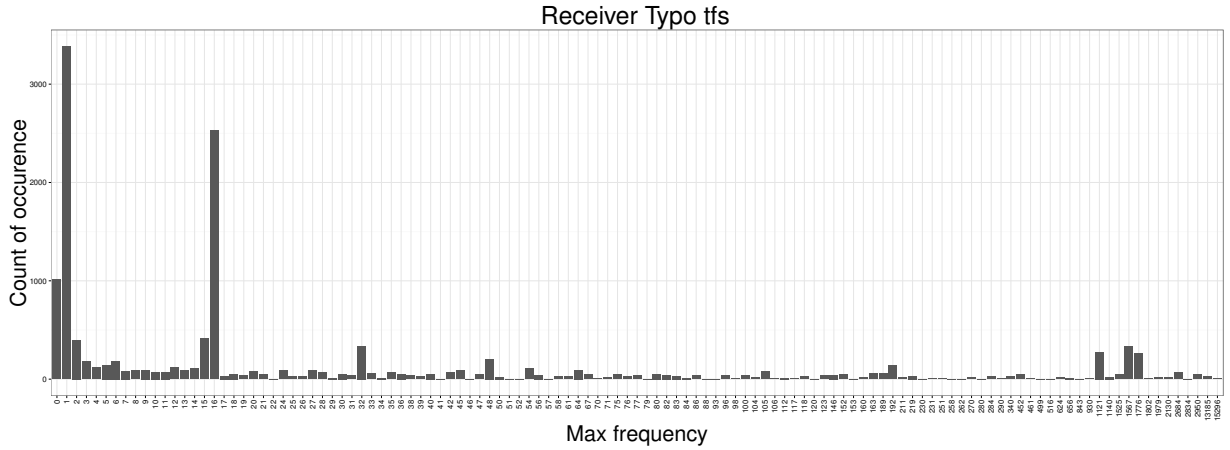


Figure 5.4: X axis shows the frequency of a bag of word models in our email corpus. Y axis show the number of receiver typo emails which has a receiver hash with that frequency

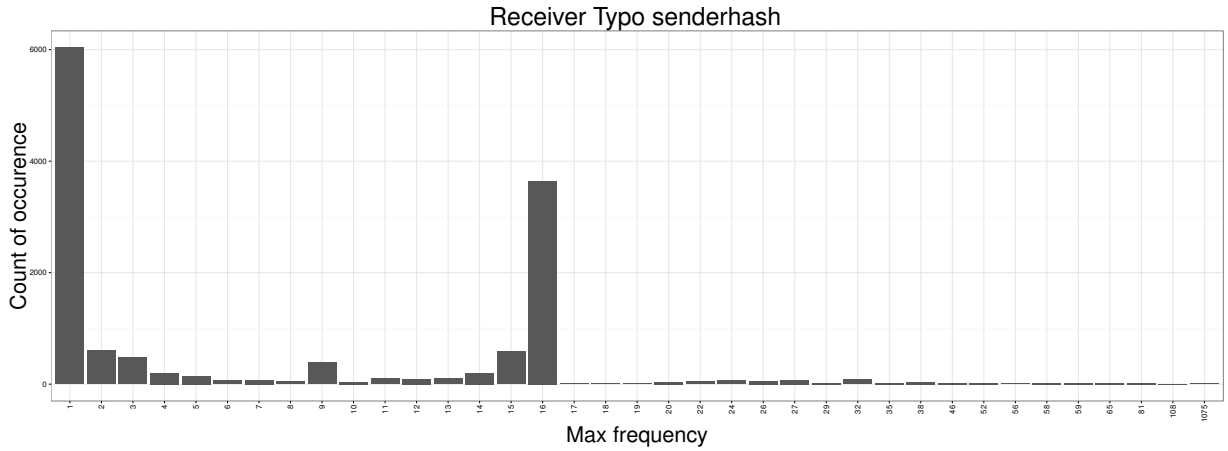


Figure 5.5: X axis shows the frequency of a sender hash in our email corpus. Y axis show the number of receiver typo emails which has a sender hash with that frequency

Layer 5: Frequency-based filtering Finally, the last layer filters out receiver typo emails (but not SMTP typos) for which the sender address, the recipient email address, or the email body appear too often in our corpus. The insight here is that true typo emails ought to be unique, rare instances. We selected thresholds for these frequencies based on the distribution of these features to include the most common frequencies and to exclude outliers. We set the receiver address frequency threshold to be 20, and both the sender address and content thresholds to 10. Details of these distributions (which motivate these thresholds) are shown on Figures 5.3, 5.4 and 5.5.

Table 5.6: Overview of our spam filtering system for candidate receiver typo emails

Domain	Typo type	Total emails	Header filtered	Spamassassin	Zip	Collaborative	Auto filtered	Freq filtered	Not filtered	Corrected	isFF
10inutemail.com	receivertypo	1245	292	177	652	62	0	60	0	0	TRUE
comcawst.com	receivertypo	7352	1302	2792	945	1195	395	700	22	4	TRUE
comaast.com	receivertypo	64292	15047	35215	7110	3095	395	3422	7	7	FALSE
comcasu.com	receivertypo	29742	10957	5777	6840	2710	95	3355	7	7	FALSE
comcas5.com	receivertypo	1322	280	567	275	142	15	35	7	7	TRUE
coicast.com	receivertypo	987	0	325	0	45	602	7	7	7	FALSE
comca3t.com	receivertypo	735	220	332	0	92	90	0	0	0	FALSE
gmaiql.com	receivertypo	131805	7717	57367	6125	35592	16222	8427	352	352	FALSE
gmai-l.com	receivertypo	2340	2340	0	0	0	0	0	0	0	FALSE
hovmail.com	receivertypo	14856485	4976770	454020	6686530	1915950	12575	808815	1825	1095	FALSE
ho6mail.com	receivertypo	160230	17452	122215	8530	5315	2005	4467	245	147	TRUE
nmailchimp.com	receivertypo	27	0	27	0	0	0	0	0	0	TRUE
ohtlook.com	receivertypo	232667	9192	71395	10720	96780	31197	12062	1320	1320	TRUE
outlo0k.com	receivertypo	92215	5860	25235	9412	30282	11915	8340	1170	1170	TRUE
outmook.com	receivertypo	114530	8090	73272	7247	12757	4665	7957	540	324	FALSE
ouulook.com	receivertypo	40062	5747	20922	5487	3955	1082	2730	137	137	FALSE
oetlook.com	receivertypo	11515	555	8835	470	1057	322	170	105	84	FALSE
ouvlook.com	receivertypo	9497	2240	2720	2575	952	130	855	25	25	FALSE
o7tlook.com	receivertypo	28370	450	11452	105	7970	4370	4002	20	20	TRUE
ou6look.com	receivertypo	7082	37	2867	32	2892	952	292	7	7	TRUE
sendgri.com	receivertypo	395	20	47	0	92	235	0	0	0	TRUE
verizo0n.com	receivertypo	6502	1757	1777	2077	427	172	280	10	10	TRUE
evrizon.com	receivertypo	7547	2507	1242	2480	742	195	372	7	0	TRUE
verhzon.com	receivertypo	1375	1127	115	87	30	0	10	5	5	FALSE
ve5izon.com	receivertypo	487	97	72	265	25	0	25	2	0	TRUE
vebizon.com	receivertypo	757	50	627	0	7	50	22	0	0	FALSE
vermzon.com	receivertypo	72	0	67	0	5	0	0	0	0	FALSE
vepizon.com	receivertypo	70	0	70	0	0	0	0	0	0	FALSE
yopail.com	receivertypo	406027	101215	150297	41312	53677	29915	28980	630	504	TRUE
zohomil.com	receivertypo	470	0	67	0	177	207	2	15	15	TRUE
zohomial.com	receivertypo	270	2	67	0	192	5	0	2	2	TRUE
Total	**receivertypo**	**16233730**	**5173187**	**1060247**	**6801112**	**2177550**	**118172**	**896200**	**7260**	6041	**-**

Table 5.7: Overview of our spam filtering system for candidate receiver typo emails received by SMTP typo domains

Domain	Typo type	Total emails	Header filtered	Spamassassin	Zip	Collaborative	Auto filtered	Freq filtered	Not filtered	isFF
mailgmx.net	receivertypo	1645	422	760	242	77	25	110	7	TRUE
mx1hotmail.com	receivertypo	455	0	257	0	57	45	2	92	TRUE
mx2hotmail.com	receivertypo	2452	692	717	500	170	42	115	215	TRUE
mx3hotmail.com	receivertypo	1692	287	500	250	217	70	127	240	TRUE
mx4hotmail.com	receivertypo	945	105	360	142	47	50	50	190	TRUE
smtpcomcast.net	receivertypo	2465	7	1857	0	525	22	12	40	TRUE
smtppaypal.com	receivertypo	65	0	62	0	0	0	0	2	TRUE
Total	**receivertypo**	**16233730**	**5173187**	**1060247**	**6801112**	**2177550**	**118172**	**896200**	**7260**	-

Table 5.8: Overview of our spam filtering system for candidate SMTP typo emails

Domain	Typo type	Total emails	Header filtered	Spamassassin	Zip	Collaborative	Auto filtered	Freq filtered	Not filtered	isFF
comcasu.com	smtptypo	1282	97	1172	0	2	2	5	2	FALSE
gmaiql.com	smtptypo	5744310	5736857	5880	2	450	887	225	7	FALSE
hovmail.com	smtptypo	37303395	37240415	2460	35740	23970	310	440	60	FALSE
outmook.com	smtptypo	6784992	6782992	1967	12	7	0	7	5	FALSE
ouulook.com	smtptypo	3097	510	1622	937	10	5	7	5	FALSE
ohtlook.com	smtptypo	4768800	4766150	2152	2	277	87	127	2	TRUE
o7tlook.com	smtptypo	4623945	4621962	1617	0	252	27	82	2	TRUE
sendgri.com	smtptypo	2640	77	1320	1082	22	62	67	7	TRUE
vermzon.com	smtptypo	1425	102	1157	0	50	30	75	10	FALSE
vebizon.com	smtptypo	1667	80	1447	0	25	27	80	7	FALSE
ve5izon.com	smtptypo	1685	80	1475	0	25	27	72	5	TRUE
verizo0n.com	smtptypo	389970	384792	2322	600	315	1860	77	2	TRUE
zohomial.com	smtptypo	11982	72	2737	5995	815	105	2242	15	TRUE
zohomil.com	smtptypo	5177	57	1895	545	1852	362	460	5	TRUE
mailgmx.net	smtptypo	3230	142	2687	0	305	2	27	65	TRUE
mxnzohomail.com	smtptypo	955	65	880	0	2	0	5	2	FALSE
outgoingverizon.net	smtptypo	1337	77	970	0	212	0	35	42	TRUE
smtpaol.com	smtptypo	163792	70	162582	0	1057	27	50	5	TRUE
smtpcenturylink.net	smtptypo	3995835	90	1460875	0	2534825	2	27	15	TRUE
smtpcomcast.net	smtptypo	3366790	77	3366147	2	462	20	20	60	TRUE
smtpcox.net	smtptypo	3542	127	2382	152	835	5	22	17	TRUE
smtpgmx.com	smtptypo	900	40	840	0	0	0	5	15	TRUE
smtpnlive.com	smtptypo	2437	82	2022	0	20	30	280	2	FALSE
smtprediffmailpro.com	smtptypo	3557	72	2807	5	532	0	120	20	TRUE
smtpverizon.net	smtptypo	4292	80	4162	0	27	2	7	12	TRUE
smtpyandex.com	smtptypo	3942	67	3832	0	7	7	12	15	TRUE
smtpzoho.com	smtptypo	1280	70	1177	0	22	0	5	5	TRUE
Total	**smtptypo**	**102661230**	**94923222**	**5114237**	**45770**	**2566902**	**5127**	**5555**	**415**	-

Performance analysis To ensure that our spam filtering performs decently, we conducted small manual analysis of receiver typo emails. We randomly selected 5 emails (collected between June 6 and September 16, 2016) for each domain name where we expected to receive receiver typo emails. One researcher analyzed the emails to decide whether they are spam emails or not. In total, the researcher labeled 77 emails and found that 80% of them were not spam emails. Further results from our filtering system can be found in Tables 5.6, 5.7 and 5.8. We additionally analyzed 26 emails that arrived by domains where we did not expect to receive anything but SMTP typos, yet, were classified as receiver typos by our system. 25 of these 26 emails turned out to have been correctly identified as receiver typos.

5.3.4 Analysis

We next turn to the analysis of the emails our infrastructure collected over more than seven months. In this entire discussion, we report numbers projected over a full year. Indeed, there were minor differences in data collection period for each domain (due, e.g., to the infrastructure being partially overwhelmed on certain days), so that we need to normalize all numbers to a common scale. Given that the study was over seven months, we hypothesize that any daily, weekly, monthly, and most seasonal effects are accounted for in our collection. In short, when collect x emails, we report the number $y = x \cdot 365/d$ where d is the number of days we actually collected data for that domain.

Email volume

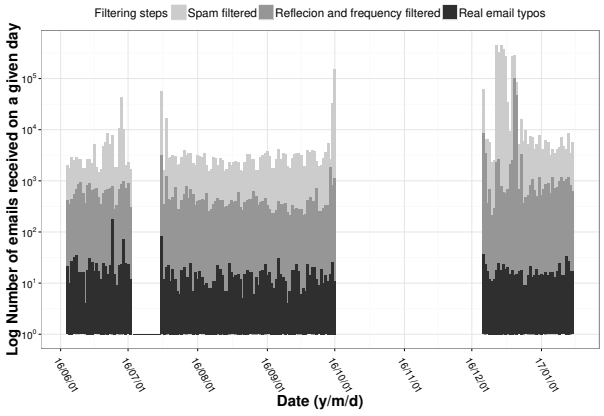


Figure 5.6: The number of receiver typo emails received daily during our data collection. Emails are in three categories: spam, auto and frequency filtered emails, and true typo emails. The plot is *not* stacked, and is in logarithmic scale on the y-axis.

Figure 5.6 and 5.7 represent the total email count, per day, we received during our collection, broken down between receiver typos (Figure 5.6) and SMTP typos (Figure 5.7). Collection gaps correspond to times during which our infrastructure was malfunctioning (in particular due to being overwhelmed with spam, and crashing as a result, with little hopes of recovering two months worth of data). We receive SMTP typo emails sparsely in small

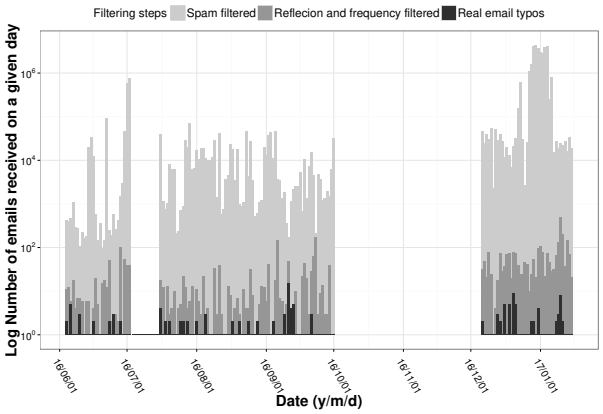


Figure 5.7: The number of SMTP typo emails received daily during our data collection. Emails are in three categories: spam, auto and frequency filtered emails, and true typo emails. The plot is *not* stacked, and is in logarithmic scale on the y-axis.

batches which perfectly characterizes what we expected. Users rarely make SMTP typo mistakes and when they do, then they quickly recognize the error and correct it. On the other hand, receiver typos occur with a near-constant rate.

Projecting from the seven months of data collection, our infrastructure receives 118,894,960 emails per year. Based on the email header, 16,233,730 are candidates to be receiver or reflection typo emails and 102,661,230 are candidates to be SMTP typo emails.

However, most of these emails turn out to be spam—only 7,260 emails per year pass all of our filters. Correcting, based on our manual analysis, would bring that number further down to 6,041 emails/year being either receiver or reflection typos.

For SMTP typo candidates we found that 5,147 emails/year are sent to us by automated agents; 5,555 of the candidate SMTP typos emails per year are frequency filtered and 415 are not. However SMTP typos, by their very nature, may lead a single user to send large amounts of email (if only for a short time), which could lead frequency filtering to produce false positives. Hence we estimate our infrastructure receives between 415 and 5,970 SMTP typo emails/year.

Surprisingly to us, we received a non-negligible number of receiver typo emails (over 700 emails/year) to domains that we had specifically designed to catch SMTP typing mistakes (for instance, `mx4hotmail.com`). These emails do not appear to be spam (as discussed above, we looked into 26 of them), but we are not sure what is causing this behavior.

Per-domain analysis

We next turn to discussing whether some domains receive more typos than others, and why.

A small fraction of domains received most of the receiver typos Out of the 31 domains registered to collect receiver typo emails, 27 domains targeted email providers, excluding temporarily email address providers (10minutemail.com and yopmail.com) or bulk email sending services (sendgrid.com and mailchimp.com). Figure 5.8 shows that out

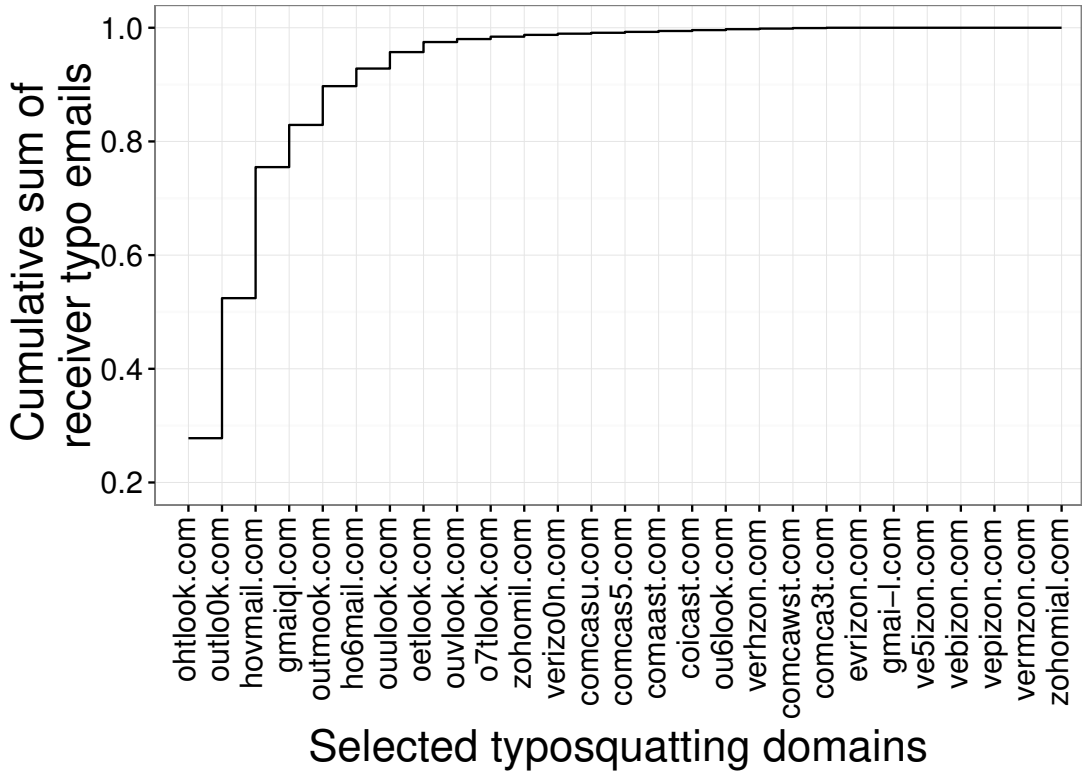


Figure 5.8: Cumulative sum of emails received by our typosquatting domains.

of these 27 domains only two domains received the majority of the total receiver typo emails and 12 domains received 99% of all emails. This finding reinforces our intuition that some typosquatting domains are orders of magnitude better than others.

SMTP typos are infrequent compared to receiver typos We receive an order of magnitude less SMTP typo emails than receiver typo emails. So, SMTP typosquatting has questionable profitability, compared to what receiver and reflection typo mistakes could offer. However, there is no harm, to the typosquatter, in simply collecting these emails on domains they would have already registered.

We define as the *persistence* of an SMTP typo for a given user the time difference between the first and last email received from that particular user. For 70% of our users, we received only one email due to a SMTP typo mistake, so that the persistence is undefined (i.e., taken to be equal to zero by convention). 83% of SMTP typos lasted less than a day and 90% less than a week. The maximum persistence was 209 days. When an SMTP typo persisted for this long it can be for one of two reasons: the same user made the same mistake several times, or these emails were spam our filtering system did not catch. 90% of SMTP mistakes caused the users to send four or less emails to our servers. As discussed earlier, emails filtered out during the frequency filtering step might include SMTP typo mistakes; however without manual inspection of their content, we cannot draw conclusions about these emails.

Visual distance, target popularity, and keyboard distances are important features Typosquatting domains targeting more popular target domains (`gmail.com`, `outlook.com`, `hotmail.com`), unsurprisingly receive significantly more receiver and reflection typo emails. More interestingly, for a given target domain, FF-1 domains always receive the most emails if the typing mistake is not totally obvious (`evrizon.com`, `ohtlook.com` and `outlo0k.com`). In other words, visual distance seems more important than keyboard distance. Figure 5.8 confirms that the top two domains are DL-1 and FF-1 typos of two of the three most popular email providers, with low visual distance from the real domain.

We only found a statistically significant correlation between the popularity of the target domain and the number of reflection and receiver typo domains received. This is not surprising since the popularity of the target domain outweighs the other attributes, and without an explanatory variable we cannot expect to see significant correlation with other attributes of the target domain.

What does a typosquatter receive?

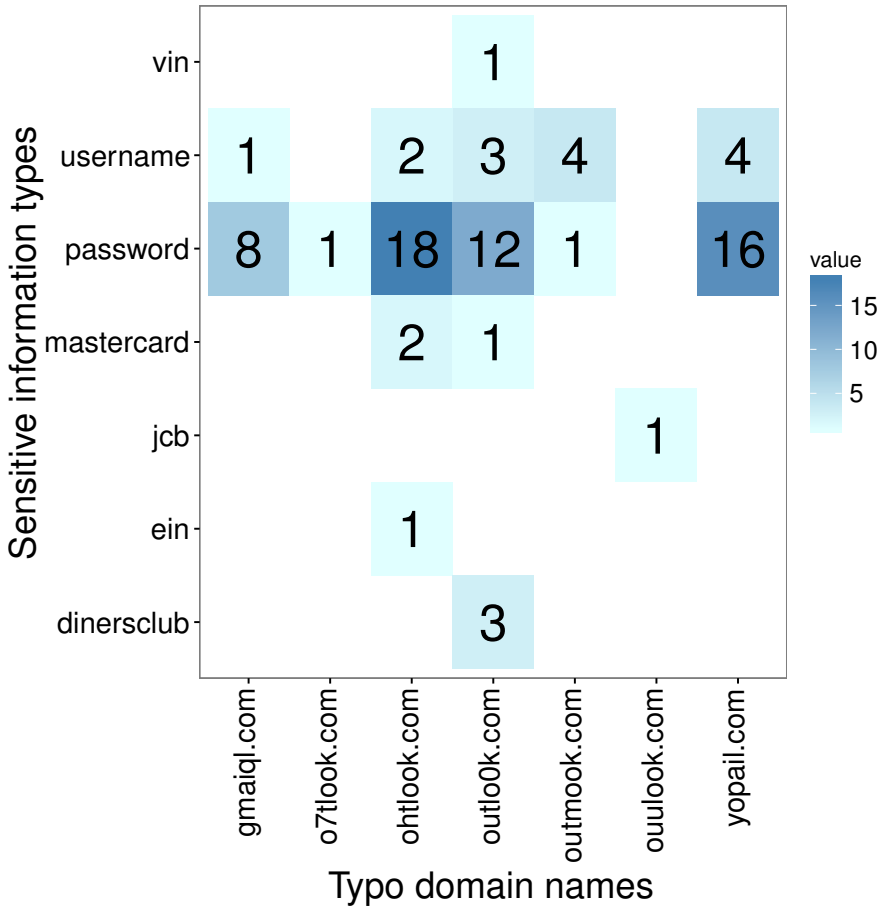


Figure 5.9: Heatmap of sensitive information of real typo emails. The heatmap shows the frequency of a sensitive information type for a given typosquatting domain.

Figure 5.9 shows among the true typo emails which ones received what kind of sensitive information. Unsurprisingly, `yopmail.com` typo domains to receive a fair amount of usernames and password since their emails are often used for temporary registration.

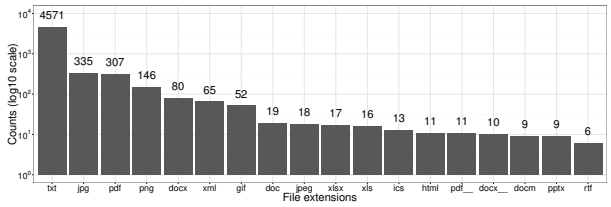


Figure 5.10: Frequency of extensions among true typo emails.

Attachment analysis Figure 5.10 shows the attachment extensions' distribution for all receiver typo emails we received. The distributions of extensions for spam emails and true typo emails significantly differ. Without filtering the emails we received have a significantly higher proportion of file types that are easier to exploit such as `.doc`, `.docm`, `avi`, `.xls` and `.xlsm`. (Recall we discard ZIP and RAR files during our filtering process.)

Out of a randomly selected 109,151 unique file hashes we found 323 in the VirusTotal database [26]. 304 of the hashes were found to be malicious and 17 were benign. All emails containing these malicious attachments were categorized as spam by our filtering system. (The benign hashes likely do not contain personal, sensitive information since they have already been observed elsewhere in the VirusTotal database.)

The dangers of reflection typos We found that one particular email address at `zohomil.com` received a lot of emails with CVs and work search related subjects and attachments. It turns out that somebody included a mistyped email address in various job postings on multiple pages—a nasty variant of a reflection typo.

5.4 The Email Typosquatting Ecosystem

We complement the results from our experiment playing the role of a typosquatter with a more "passive" analysis, in which we attempt to estimate whether email typosquatting does occur in the wild, and who the actors are.

5.4.1 Methodology

To gain an better understanding of the typosquatting ecosystem we first looked at the set of ctypo domains registered in the wild. We generated all possible DL-1 variations of Alexa's top one million domain on November 5, 2016 [1]. We considered the set of ctypo domains, i.e., the domains that are actually registered, and collected the MX and A records of these ctypo domain names, on November 7, 2016. The SMTP protocol specifies that, in absence of an MX record, the A record of the domain name should be used as the mail server's address [84]. We clustered ctypos together based on their DNS settings to see any

Table 5.9: SMTP support of typosquatting domains

Support status	Count	% total	% analyzed
No MX or A record found	651,439	15.5	23.7
No info	1,441,725	34.4	-
No email supp.	28,3636	6.8	10.3
Supp. email, no STARTTLS	1,693	0.0	0.1
Supp. STARTTLS with errors	257,952	6.2	9.4
Supp. STARTTLS w/o errors	1,556,773	37.1	56.6

evidence of concentration in the typosquatters' infrastructure. If there was no MX record found for a domain name we used the corresponding IP address for clustering.

We further analyze whether these domains actually run an SMTP server using data downloaded from zmap.io [27] on October 29, 2016. We checked the IP addresses obtained from requesting the A record for those domains for which an MX record was found. If there was no MX record, we used the A record directly.

We also attempted to collect WHOIS information for all ctypo domains between December 22 2016 and January 24 2017. We used PyWhois [15], and Ruby Whois [16] for querying and parsing WHOIS information. While a lot of the information is probably fake, it can nevertheless be useful in clustering domains by owners (e.g., while Mickey Mouse is unlikely to register typosquatting domains, repeatedly seeing the name Mickey Mouse as a technical contact for typosquatting domains might be evidence of common ownership).

More precisely, to cluster registrants of typosquatting domains we use an approach similar to Halvorson et al. [59]. We use six fields of the WHOIS record: registrant name, organization, email address, phone number, fax number and mail address. We consider two domain names to be registered by the same entity or group of entities, if four of the six fields match. Naturally, this means we cluster only domains for which at least four WHOIS fields were available. Using a `.com` zone file, we find domain name servers that serve a significantly higher proportion of typosquatting domains than should be expected.

5.4.2 Analysis

SMTP support for typosquatting domains Table 5.9 shows SMTP support for typosquatting domains. 22.3% of typosquatting domains are not capable of receiving emails, 34.4% did not yield any information, and 43.3% support SMTP.

Typosquatting registrants Using the clustering technique described above, Figure 5.11 shows the concentration among registrants (excluding those protected by WHOIS proxy services) who filled out at least four of their WHOIS registration fields. The x-axis is the fraction of all registrants. The top 14 registrants own 20% of typosquatting domains. A mere 2.3% of all of the registrants in appear to own the majority of typosquatting domains. At the same time, there is a heavy long tail for the ownership of the rest of the domains.

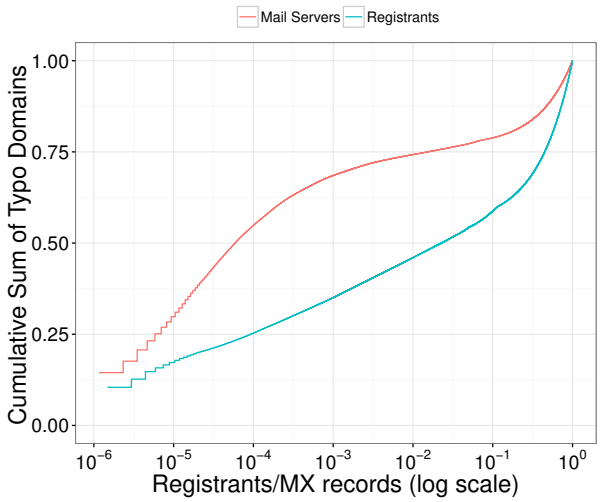


Figure 5.11: Cumulative sum of typosquatting domains by mail servers and registrants. Mail servers and registrants are ordered by the number of domains served/owned, in decreasing order.

Most of the registrants that operate a large number of typosquatting domains have SMTP servers active on most of their domains. The top three registrants are actually companies whose business appears to be holding domain names for sale. While questionable, this practice is not evidence of active malice. On the other hand, many of the other registrants do not seem to focus on domain resale, but do operate SMTP servers, which is suspicious. Table 5.10 contains a list of the top typosquatting registrants.

Suspicious name servers A number of name servers are used by a significantly higher ratio of typosquatting domains compared to benign domains. In general, the average ratio of typosquatting domains over benign domains is about 4% – par for the course for large organizations that may not be able to check very carefully the activities of all of their customers. However, a number of name servers far exceed that ratio, and can be viewed as catering to typosquatters. The candidate typosquatting ratio of all `.com` domains is as high as 89% for one such name server. Further adding to the suspicion, half of these name servers are registered behind privacy proxies, and a majority of their domains have active SMTP servers. Full details about these name servers are shown in Table 5.11.

MX record concentration As Figure 5.11 shows, not only do a lot of typosquatting domains support mail, but many of them point to only a few MX records. The top eleven SMTP servers handle mail for more than one third of typosquatting domains and 51 for the majority. Less than one percent of the SMTP servers supports more than 74% of domains. In other words, a few providers might have the chance to defend against (or be held responsible for, in case they are colluding with the miscreants) potentially dangerous and privacy invasive email typosquatting.

SMTP and mail typos Some typosquatters deliberately target SMTP subdomains (e.g., registering `smtpgmail.com` to `smtp.gmail.com`) and webmail domains (e.g., by registering `mailgoogle.com`, targeting `mail.google.com`). We found 41 SMTP and 366 mail

Table 5.10: Registrants owning a large number of typosquatting domain names

Number of Typo domains	Unknown	SMTP support Supported	Not Supported	Names	Organizations
156383	154533	1808	42	domain admin;domain admin / this domain is for sale	hugedomains.com
33333	33305	19	9	this domain for sale worldwide;this domain for sale worldwide 339-222-5132;this domain for sale toll free 866-822-9073 or 339-222-5132;rn webreg	rarenames, inc.;rarenames;buydomains.com;this domain for sale worldwide 339 222 5132
30865	30855	10	0	whois foundation;domain may be for sale, check afternic.com domain admin;domain admin;domain may be for sale, check afternic domain admin	whois foundation;megatransfert sasu
15449	15434	15	0	dns administrator	cykon technology limited
12206	13	12192	1	domain manager	eweb development inc.
9784	9783	1	0	domain administrator	domainmarket.com
8810	8398	134	278	wujianfeng;wu jianfeng;zhenjie zhang;shanghai ronghaoshiyeyouxiangongsi	wujianfeng;jianfengwu;www.jinmi.com;wu jianfeng
7273	2775	4498	0	noorinet	noorinet
5561	5538	21	2	guojianguang;guo jian guang	guojianguang;guo jian guang
5295	87	5207	1	luigi bian film;luigi bian	los angeles news;losangelesnews.com inc.;losangelesnews.com;los angelesnews.com
4735	2344	1868	523	reactivation period	
4068	3957	105	6	new ventures services, corp	new ventures services, corp
3725	1765	1957	3	jiangyilin;zhu jiang;yilin jiang;jiang yinlin	chengduyisuoxinxikeji;jiangyilin;www.yisuo.com;chengdu
3469	3195	252	22	nvc nvc-sn;new ventures services	new ventures;new ventures services;hostmonster.com
2673	968	1704	1	domain administrator	dvlpmnt marketing, inc.
2589	2507	66	16	wu wen bin;wuwen bin	xia men yin si bao hu fu wu you xian gong si;wu wen bin
2545	1210	1262	73	protection of private person	
2441	886	798	757	domain id shield service	domain id shield service co., limited
2420	2144	220	56	linshi moban;zhenguo liu;ke yang;linshimoban linshimoban	niqingsu;zhu zhi bin;zeng zhi yu;edward
2140	2037	7	96	lirong shi;shi lirong	lirong shi;shi lirong;shilirong
2081	2080	0	1	wenchao chen;song qiuxiang;songqiuxiang songqiuxiang;chen wenchao	song qiuxiang;chen wenchao
2012	2011	1	0	gregg ostrick	gno, inc.
1904	1559	345	0	gary willicott	the web group (client account)
1903	44	1859	0	domain hostmaster	protopixel pty. ltd.;protopixel pty ltd
1901	842	794	265	beth schmierer;namecheap.com for sale;namecheap.com namecheap.com;namecheap.com	vt sistemas de informatica ltda;namecheap;namecheap, inc;refruit
1817	0	1817	0	gabriella santos rodrigues	chargepal s.l.
1812	1765	46	1	kenji hiraiwa	gmo digirock, inc.
1808	1807	1	0	admin	admin
1779	1779	0	0	ryusung	ryusung
1767	180	1561	26	xserver inc. xserver;wpx xserver inc.;naoki kobayashi;xserver xserver inc.	bet inc.;xserver, inc. - sixcore;xserver inc.;bet,inc.
1704	628	1076	0	netsupport asknysite	asknysite.com llc
1624	64	13	1547	domain administrator	china capital
1546	25	1521	0	lei shi;xiaohu hu;shilei	www.jinmi.com;shilei
1469	1466	2	1	netcorp netcorp;netcorp, llc	netcorp;netcorp, llc;inquire about this domain via contact email address
1404	672	597	135	domain admin	privatewhois biz
1401	1142	259	0	yang kyung won	yang kyung won
1387	1386	1	0	kim seokjun	kim seokjun
1349	1275	32	42	weisheng hu;xiaohu hu;mibao you;yuling liu	you mibao;www.jinmi.com;zheng tao;cheng xinming
1328	1322	6	0	william coam;william william;coam, william	germanium inc.;speednames.com hostmaster
1315	1065	152	98	wuxi yilian llc	wuxi yilian llc

Table 5.11: Name servers with a high proportion of typosquatting domains.

Name Server	Total Domains	Ratio Ctypo	Ratio Ttypo	Count Ctypo	Count Ttypo	SMTP Support Unknown	SMTP Support Yes	SMTP Support No	DigiMedia.com, L.P.
gbcdn.net	1082	0.89	0.17	968	182	968	0	0	Robert Brooks
storeland.ru	5919	0.76	0.18	4476	1038	4438	38	0	Privacy protected
citizenhawk.net	6424	0.66	0.67	4220	4320	4220	0	0	Brand protection: CitizenHawk, Inc.
a0f.net	37715	0.61	0.6	22884	22691	22882	2	0	Brand protection: CitizenHawk, Inc.
orbitz.com	1528	0.54	0.52	820	792	818	2	0	Orbitz Worldwide, LLC
shutterfly.com	680	0.51	0.19	350	130	12	338	0	Shutterfly, Inc.
easily.net	77730	0.41	0.34	31588	26116	31448	140	0	Privacy protected
frays.com	1568	0.41	0.19	642	304	136	506	0	Privacy protected
liverealdeals.com	29708	0.39	0.48	11628	14174	22	11604	2	Privacy protected
lifeisatest.com	912	0.37	0.34	336	308	0	336	0	Doug Powell
dnsiz.com	12698	0.35	0.37	4434	4664	4434	0	0	jianghong
consumerinfo.com	4853	0.35	0.31	1694	1510	1692	2	0	Consumerinfo.com, Inc.
dnparking.com	15810	0.32	0.32	5047	5085	5047	0	0	Privacy protected
dnssafe.com	3639	0.3	0.26	1101	954	0	1101	0	Privacy protected
redmonddc.com	23870	0.28	0.23	6734	5442	6734	0	0	Privacy protected
slickdns.com	1105	0.28	0.01	308	7	288	20	0	John Barham
domainingdepot.com	51460	0.27	0.27	14022	13956	1168	12854	0	Privacy protected
domainca.com	17514	0.27	0.18	4643	3233	4490	151	2	Marble Internet Inc.
gettyimages.com	2936	0.27	0.24	783	693	306	477	0	Getty Images (US), Inc.
mfk1.com	7414	0.26	0.21	1910	1554	1910	0	0	Privacy protected
securedoffers.com	2640	0.26	0.22	692	590	690	2	0	Privacy protected
e43n83hd.com	2554	0.26	0	668	0	124	0	544	Privacy protected
torresdns.com	26340	0.25	0.24	6570	6416	6566	4	0	Privacy protected
domainmx.com	14884	0.25	0.22	3786	3262	3786	0	0	Privacy protected
koolwebsites.com	4344	0.25	0.26	1086	1122	0	1086	0	Ibrahim Kazanci
smtmdns.com	48062	0.24	0.24	11420	11670	11416	4	0	Privacy protected
ename.cn	1413	0.24	0.09	343	130	343	0	0	Xiamen Yi Ming Technology Co., Ltd
createsend.com	672	0.24	0.09	160	59	60	100	0	Campaign Monitor Pty Ltd
domainmanager.com	19026	0.23	0.16	4424	2990	380	4044	0	Privacy protected
digimedia.com	5834	0.23	0.21	1333	1210	1318	15	0	DigiMedia.com, L.P.

typosquatting domains registered, targeting Alexa's top 10,000 `.com` domains and Alexa's top 500 `.com` domains in the email category.

The SMTP typos include domains `smtpgmail.com`, `smtpoutlook.com` and `smtplive.com` targeting the biggest email providers. This could plausibly be defensive registrations. However, they are *privately* registered, which is inconsistent with trademark protection—in our experience, defensive registrations usually point at the legitimate owner or their agent, not at a private registration service.

5.5 Extrapolating from our Experiments

In this section, we combine the observations gleaned by through our experiment (Section 5.3 and our analysis of the typosquatting ecosystem (Section 5.4) to attempt to extrapolate our findings on an admittedly limited set of domains to the whole Internet.

5.5.1 Toward a projection

We use a seed of 25 of our typosquatting domains targeting 5 email domains: `gmail.com`, `hotmail.com`, `outlook.com`, `comcast.com`, and `verizon.com`. These domains are highly popular email services, and using the information from our small foray into typosquatting might help us best understand the potential magnitude of email typosquatting in the wild.

Specifically, we attempt to project our results to other typos of email domains. To do so, we rely on three hypotheses

(H1) Typing mistakes are equiprobable among users of different email providers.

(H2) Sending an email is a two-step process. Users type in the email address. Second, users verify the address and potentially correct any mistakes.

(H3) The number of emails sent to a typosquatting domain is proportional to the number of emails sent to the target domains.

Based on these hypotheses, we build a simple model to estimate the expected number of emails sent to a given typo domain

$$E_{ij} = E_i \cdot P_{t_{ij}} \cdot (1 - P_{c_{ij}}) \ ,$$

where E_i is the expected number of emails (over a fixed time period, e.g., a year) sent to email addresses in domain i, E_{ij} is the expected number of emails sent to email addresses in domain j, where the DL distance between i and j is either zero or one.

$P_{t_{ij}}$ is the probability of user typing j instead of i. (This includes typing the correct domain.) $P_{c_{ij}}$ is the probability of the user correcting the mistake after typing j instead of i.

Directly validating this model is impossible, because $P_{t_{ij}}$ and $P_{c_{ij}}$ are unknown, and different for different domains, even in the case of similar typing mistakes. Instead, we build on this simple model to devise a linear regression model used to predict E_{ij} based on features characterizing the process of typing mistakes.

First, we use Alexa's monthly unique visitors to estimate E_i for email domains (e.g., `gmail.com`, `outlook.com`). We assume E_i is proportional to the number of active users of domain i.[4] We add three features to incorporate $P_{c_{ij}}$ into our model: the visual distance, the length of the target domain and position of the mistake, and the fat-finger distance.

One drawback of our approach is that we were not able to register domains of popular email providers with deletion or transposition typos. Thus we used Alexa's data on typosquatting domains of the 40 most popular target domains, to estimate the difference in probability between different typing mistakes. We collected Alexa's data from October 27, 2016 to October 30, 2016 [1].

Furthermore, we removed typosquatting domains receiving outstanding traffic among typos of the same target domains, because those domains are probably not malicious, and just happen to be accidentally close to the target domain. We used the median of all absolute deviations from the median (MAD, [115]) to detect such outliers. We estimate the 95% confidence interval for the mean of the different typing mistakes to estimate how different their average traffic is. We will use these results to estimate the number of emails received by deletion and transposition typo domains.

5.5.2 Regression results

The five target domains—`gmail.com`, `hotmail.com`, `outlook.com`, `comcast.com` and `verizon.com`—are targeted by 1,211 typosquatting domains (excluding defensive registrations, and our own 25 domains).

We build a linear regression model, by transforming the dependent variable to square root space. We select the following three features: the target domain's Alexa rank (log transformed), the square root of our visual distance heuristic (between the target and the typo domain) normalized by the length of the original domain and the fat-finger distance between the target and the typosquatting domain (zero or one). The R^2 value of the fit is 0.74. Running a leave-one-out cross-validation test the R^2 value drops to 0.63.

Our model finds that the 1,211 typosquatting domains registered by others should receive approximately 260,514 emails per year, with a 95% confidence interval ranging between 22,577 and 905,174 emails per year. Figure 5.12 shows based on the AWS Alexa data collected that deletion and transposition typo mistakes are significantly more frequent than addition and substitution mistakes. Taking this information into account, our modified regression analysis yields an expected number of emails received by typosquatters equal to 846,219 with a 95% confidence interval ranging between 58,460 and 4,039,500.

Economic implications Registering a `.com` domain costs about about USD 8.5 per year. Using this price in the model above, a typosquatter owning these domains can acquire an email for less than two cents. (This computation excludes spam.) From our own experience, by keeping our five top performing typosquatting domains we could collect "legitimate," non-spam emails for less then a penny a piece (excluding marginal costs, such as those of running a server, and keeping storage).

[4]This assumption does not hold in the general case, when web popularity may be very different from email usage; but we assume it is reasonable in the case of the webmail domains we are looking at.

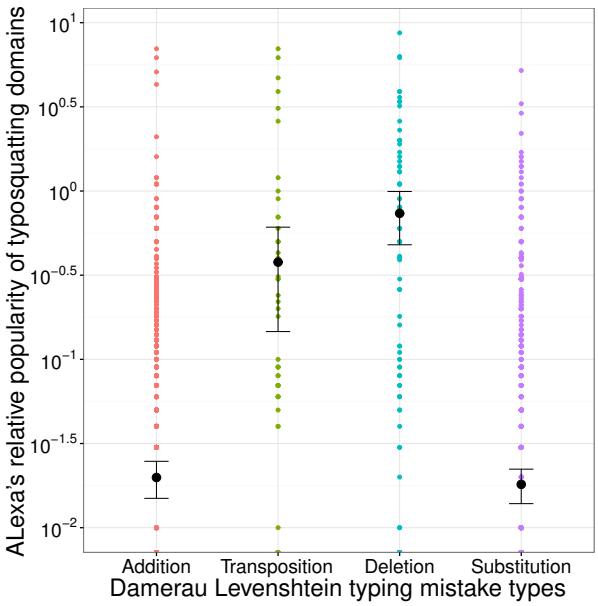


Figure 5.12: The average relative popularity of typosquatting domains separated by the type of typing mistakes: addition, deletion, substitution, transposition. We also marked the average popularity and the 95% confidence interval for each type of mistake.

However, we conjecture that the domains registered by us were mostly available, because they are less profitable than other typosquatting domains. In other words, we would not be surprised if our calculations only provided a relatively conservative estimate on the number of emails typosquatters actually receive when registering typosquatting domains targeting popular email service providers.

The very small set of emails we manually analyzed appear to contain a wide variety of sensitive information that cannot be exploited by itself, but can aid miscreants to perform targeted attacks. For instance, six of the 103 emails we analyzed manually appeared to contain digital receipts, which contain considerable personal information that could be used for subsequent spearphishing campaigns or other scams; some other emails included information (car registration, visa documents, resumes, adult side registration, medical records) that could plausibly be used for identity impersonation, spear-phishing, or even intimidation.

5.6 In the shoes of a typosquatting victim

We have discussed the potential threat of email typoquatting and the existing ecosystem that appear to support it. However, are typosquatters actually doing anything with the emails that they are able to collect? To answer this question, we run an additional experiment, in which we now play the role of a potential victim, and deliberately email known typosquatting domains with "honey emails." This experimental protocol, like the collection protocol earlier described, was vetted and approved by our IRB.

5.6.1 Experimental design

Honey email design We designed our honey emails to 1) signal back to our servers when opened and 2) to include seemingly sensitive information (e.g., login credentials), whose access we can monitor.

Our emails included a 1x1-pixel tracking image residing on a VPS we operate. HTML clients might try to download this image upon opening the email, but this is not always the case. For instance, depending on its default configuration, the Thunderbird email client may not automatically download such embedded images. Shortly stated, presence of a signal indicates that the email has certainly been opened, but absence of a signal is not proof that the email was not opened.

We included sensitive information in the form of honey tokens and honey accounts. A honey token is a file attachment that signals back upon being opened. After experimenting with both PDF and DOCX, we discovered that DOCX readers tend to allow external access by default more commonly than PDF readers.

Our honey accounts consisted of email accounts at two major email providers and a shell account on a VPS we control. The wording and headers of each email were designed to mimic real-life interactions between users. (We piloted these emails with members of our research group, to confirm they looked plausible, and were not caught by spam filters.) In total we used four different email design templates, and we made sure to send one typosquatter registrant one of each email designs exactly once. Further, we only sent one email to each typosquatting domain.

Our first email design included login information for a major email service provider. The second design included login information for a shell account under our control. The third design included a link to a tax document shared through a major document sharing service, where we could monitor accesses. Our final design had a DOCX attachment with (fake) payment information.

Sending emails We ran two measurement experiments.

Email probes. The first experiment had for objective to determine how many typosquatting domains actually accept email – the idea is that this gives us a rough idea of how many are deliberately set up as email typosquatting domains, as opposed to web typosquatting domains that happen to also target email domains. To that effect, we started with a pilot, in which we sent out a small number of 164 honey emails between May 2, 2017 and May 6, 2017. We selected a low number of target domains (and a low sending rate) to avoid alerting typosquatters to our measurements. However, most of our emails bounced, or resulted in a timeout or network error.

After this pilot, we ran a larger measurement to test how many typosquatting domains accepted *any* of our emails. To that effect, on May 15, 2017, we sent out 152,985 benign emails to 50,995 typosquatting domains, including domains of registrants owning the most typosquatting domains, domains linked with a name server frequently used for typosquatting domains, typos of the three major email domains (`gmail.com`, `hotmail.com`, `outlook.com`), and finally candidate typosquatting domains that use the most popular WHOIS privacy service.

Each domain selected listens on (some of) the SMTP server ports, according to Zmap. To verify which one, we sent three emails – one each to ports 25 (no authentication), 465 (SSL) and 587 (STARTTLS).

The emails in this experiment were designed to look like test email without any sensitive information in them. Here, we sent emails from our own virtual private servers. This allowed us to determine whether emails were actually received and/or read in a client that retrieves external resources.

Honey tokens. We then conducted a second set of measurements, in which the goal was to determine if emails were not only received, but also read and/or acted upon. Here too, the experiment started with a *honey token* pilot measurement limited to 738 domains out of these 50,995 typosquatting domains, to ensure that the infrastructure worked as it was supposed to and to run a conservative measurement unlikely to be detected by miscreants—indeed, most typosquatters, even those who operate myriad domains, received at most one email from us. We selected these 738 domains by 1) purposefully limiting ourselves to at most four domains per registrant we could identify, and 2) selecting these four domains based on their Alexa rank and the type of typing mistake. We sent out one honey email containing sensitive information to each of these typosquatting domains on May 15, 2017. All emails in this pilot were sent through a major email provider to make them less conspicuous and to avoid spam filters.

Following this pilot, on June 15, 2017, we ran a far more aggressive measurement, in which we sent all four different honey emails designs to *all* 7,269 typosquatting domains which had accepted our emails in the first set of experiments. Here, due to the size of the test (close to 30,000 emails), we used our own servers, rather than a major email provider, to send out these emails. During this test, while we only sent four (different) emails per domain exactly once, we potentially sent out the same email multiple times to the same individuals – since some typosquatters own more than one domain.

We logged access attempts to the "honey" shell account until July 1, 2017; and accesses to the other resources our honey tokens were pointing to until September 14, 2017.

5.6.2 Results

Table 5.12: Error message count received when running the initial test for the honey email experiment.

	Number of typo domains	
	Public reg.	**Private reg.**
No error	1,170	6,099
Bounce	1,567	1,160
Timeout	17,923	6,976
Network Error	7,901	6,584
Other error	93	1,522
Total	28,654	22,341

Typosquatting domains and email acceptance Table 5.12 presents the results of our first experiment, in which we monitored whether our honey emails were accepted. 1,170 publicly registered domains accepted our emails without any error message. Based on our access logs, three of these domains, including two (`outfook.com`, and `uutlook.com`) that seem to be clear typosquatting domains, appear to have read our emails. On the other hand, we experienced a large percentage of network errors and timeouts for the majority of publicly registered domains.

6,099 of our emails were accepted on domains using WHOIS privacy proxy services, which overall presented far less errors. 19 of these emails were read based on our logs. We discovered that 6 of these domains were clear typosquatting domains, 8 were legitimate domains that just happened to look like typosquatting domains, and 5 could be either way. Glancing at the time difference between emails were sent and when emails were opened seems to suggest that these emails might have been read by humans – rather than by automated processes – as it frequently took several hours before the email was opened. Furthermore, some of these emails were opened several times, sometimes days after they were first opened

Interestingly, some of these domains appear to be targeting potentially sensitive sectors, such as banking (e.g., `disvover.com`, `bankofamericqa.com`), adult sites (e.g., `nuaghtyamerica.com`), or email providers (e.g., `comcacst.com`).

Table 5.13: Distribution of the mail exchange server usage for the domains that accepted our emails.

MX domain	Total	%	CDF	Private?
`b-io.co`	3,171	43.6	43.6	Yes
`h-email.net`	1,344	18.5	62.1	Yes
`mb5p.com`	732	10.1	72.2	Yes
`m1bp.com`	635	8.7	80.9	Yes
`mb1p.com`	558	7.7	88.6	Yes
`hostedmxserver.com`	225	3.1	91.7	Yes
`hope-mail.com`	176	2.4	94.1	Yes
`m2bp.com`	94	1.3	95.4	Yes
`google.com`	61	0.8	96.2	No
`googlemail.com`	34	0.5	96.7	No

Table 5.13 shows that 95% of the domains which accepted our emails without errors rely on eight mail server domains, which are all privately registered.

Honey tokens and honey accounts While the pilot measurement – sending data to 738 domains only – did not result in any signal being sent back to us, our larger measurement to all 7,269 suspected typosquatting domains resulted in 15 emails being apparently opened and/or read by someone, and two honey tokens being accessed. Here too, we saw a lag of several hours between the time we sent emails and the time they were opened or read, suggesting human involvement.

Specifically, on June 16, 2017 a potential typosqatter read the "tax document" we had uploaded to a known document sharing service. The domain we sent this honey email to was a legitimate service once, but for the past two years it has been operating as a parked domain. Logs provided by the document sharing service indicate that the document was opened half an hour after we sent it, and was viewed for 28 seconds from Caracas, Venezuela using a Windows desktop computer. We also saw that 9 days later someone read our email from another IP also from Caracas, Venezuela and 14 days later from Orlando, Florida.

Likewise, on June 16, 2017 a potential typosquatter tried to gain access to our honey shell account from an IP in Poland. This specific email did not show up in our logs as having been viewed, presumably due to the miscreant not opening inlined images.

While interesting, we caution that these two anecdotes are far from providing evidence of systematic email collection and monetization by typosquatters—in fact, given the number of emails we sent, it seems that these practices are the (rare) exception rather than the norm.

This overall negative result may be explained by several factors. First, the risk involved with getting caught might be higher then the expected benefit of the sensitive information we sent them. Second, it is possible that typosquatters do not even realize that they are collecting these emails; plausibly, the SMTP servers could have been turned on by default, and not wilfully. After all these domains might have been registered primarily for *web* typosquatting, with email typosquatting being an afterthought, if a thought at all.

5.7 Discussion and limitations

A major limitation of this study is that it only considers domain typosquatting, and not username typosquatting. For instance `aliec@gmail.com` might receive a lot of email meant for `alice@gmail.com`. However, without the collaboration of the email service provider, doing an analysis of username typosquatting is impossible.

Our data collection experiments show that there is potential danger, but, contrary to web typosquatting, the "expected" risk to consumers is far less obvious – most of the time, the risk is probably very low, but in a few cases, depending on the specific content that is being sent, might lead to disastrous outcomes (contrary to web typosquatting).

While we have seen only scant evidence of credential abuse in the wild when we posed as victims, we have on the other hand discovered highly suspicious registration patterns. These may be a by-product of web typosquatting, but we cannot rule out that the situation will not change; the infrastructure appears to be certainly already in place, even though this may be accidental.

Web vs. email typosquatting Web typosquatting is one of the easiest attacks to carry out, because it requires almost no technical knowledge. As our measurements show, some parties are seemingly interested in exploiting typing mistakes and have the ability to collect emails from potential victims. Yet, they don't appear to act upon these emails, even though there is plenty of evidence (from our data collection) that many people actually could fall victim to this kind of attack.

Reflecting more on this negative result, web typosquatting only needs the ability to register a domain and the subscription to a parking service, and is thus accessible to any miscreant. On the other hand, email typosquatting requires deeper technical expertise. First, the collection infrastructure is not straightforward to set up. Second, spam filtering is equally complex—as we saw in our own experiment, spam filters alone might not be very reliable. To add insult to the injury, the payoff is far more uncertain (low occurrence, high payoff) than in the web typosquatting case (high occurrence, low payoff), and the risk of getting in trouble (e.g., if abusing financial credentials) is much higher.

Possible defenses What if the situation were to change, and typosquatters actually used emails received for profit? Our results in Section 5.3.4 shown that far more emails are received by typosquatting domains targeting top email service providers compared to middle sized providers. This trivially means that large providers registering their typosquatting domains defensively would have the biggest impact per defensive registration and also it would be the most cost effective per user. While for a small company it might be financially burdensome to register hundreds of domains (not mentioning the legal costs in case the domains are owned by someone else), for major companies, a few thousand dollars a year should be a negligible cost. It is not unprecedented for a large company to acquire typosquatting domains in bulk even if legal lawsuit is needed. Facebook a few years ago won a lawsuit summing to $2.8 million against typosquatters, recovering 105 typo domains. UDRP and ACPA provide frameworks for brand owners to acquire typosquatting domain names, in case they are already owned by typosquatters. Similarly these costs should be low compared to the potential harm for the financial sector such as banking domains.

Besides defensive domain registrations, typo correction tools could also help to reduce the potential harm from typosquatting. Typo correction could be integrated into any input field: at SMTP setup phase, registrations, email recipient, or when giving contact information in online forms.

Policy interventions could also be viable. For instance, the Chinese registry raised the registration price and requiring identification for `.cn` domains. Raising the cost of domain registration and requiring identification for registration would definitely drive most of the typosquatters out of business. However these intervention would potentially have a high collateral damage on legitimate domain owners. Another approach would be for ICANN and registrars to periodically remove typosquatting domains. This however is unlikely to happen due to incentive misalignments, namely that this would require a great effort from this parties who do not suffer from this activity and at the same time their revenue would decrease.

5.8 Conclusion

We conducted a measurement study of email typosquatting, based on our own data collection, and an examination of the whole ecosystem. We conclude that the profitability of a typosquatting domain depends on three main factors: popularity of target domain, edit distance from target domain, and visual distance from the target domains. We observed that receiver and reflection typo emails are an order of magnitude more frequent than SMTP typo

emails. Among the emails received we found users accidentally sending us email containing highly sensitive personal data. We also observed that some registrants own thousands of email typosquatting domains, that these domains support SMTP. Furthermore, some of the name servers (and registrars) used by tens of thousands of typosquatting domains appear to be cesspools, with a 5–10 higher typosquatting domain ratio than normal. Even though typosquatters have the infrastructure to collect private emails in bulk literally for pennies each, we found that, with very rare exceptions, they do not actually misuse sensitive information sent to them. We conjecture this may be due to incentives being in favor of web typosquatting—shortly stated, it is not worth bothering with a more complex attack with a more uncertain payoff—but cannot guarantee the situation will not change. Certainly, the potential for monetization by a determined actor is there, and proactive defenses ought to be considered.

Chapter 6

Domain Registration Policy Strategies and the Fight against Online Crime

In the previous chapters of this thesis, we explored the typosquatting ecosystem and devised techniques and tools to protect users and brands. While these defenses can be effective when used, in the past decade, we have not seen a decrease in typosquatting and other abusive domain registrations despite our and other researchers' efforts. Our hypothesis is that without reforming domain registration policies, conventional approaches to combat abusive registrations like proactive detection of the criminal activity and reactive blacklisting will not deter miscreants from abusive registrations. Thus in this chapter,[1] we take the first step to systematically study how registration policies can aid classic defenses to combat online crime.

Building on our understanding of the domain registration ecosystem, we develop a multi-stage analysis framework for registration policy proposals. As part of our framework, we discuss the biggest challenges to registration policy deployment (e.g., the complexity of the international domain registration ecosystem); when domain registration can or cannot affect online crime; and the inherent limitations of such analyses. We hope to stimulate further policy work and broaden the discussion beyond technical measures to impede online criminal activity.

Our most promising registration policy proposal comes from the observation that online criminals need far more domain names to operate effectively than benign registrants. We propose a dynamic pricing function and stricter identity verification to make bulk domain registrations expensive. Our game-theoretical analysis indicates that this proposal should have a minimal effect on benign registrants and registries while having a significant financial and operational impact on certain criminal activities. Most interestingly, we observe a synergy between blacklisting and domain registration policies, where increasing blacklisting performance disproportionately boosts policy effectiveness.

[1]This chapter is primarily based on our paper published at the 2018 Workshop on the Economics of Information Security [127]

6.1 Introduction

The Internet depends on the Domain Name System (DNS) to resolve names humans can remember to Internet Protocol (IP) addresses understood by computers. While DNS is also used for a few secondary reasons such as load-balancing and geo-targeting, its main purpose has remained to help humans to find websites (e.g., HTTP, HTTPS), communicate with other humans (e.g. SMTP, POP3, IMAP), or to find other services (e.g. FTP, SSH, Gaming servers). Some domain names became extremely valuable brands and sell for millions of dollars [143].

The value and importance of domain names brought with them a wide range of abuse aimed to profit from them. Domain squatters [101], typosquatters [128], combosquatters [83], and soundsquatters [109] hope to profit from their domain names' similarity to a brand name by passively counting on users' mistakes (e.g. typing mistakes) or by actively fooling users (phishing). Phishing and scams frequently use domain names designed to add a veneer of legitimacy. Spammers use domain names to evade blacklisting of their sender email domains or the domain names in the advertised URL. Drive-by-downloads, botnet operators, illegal content distribution sites and many other online criminals need a large number of domain names to evade blacklisting.

Existing efforts have focused on retroactively blacklisting domain names, after evidence of abuse had surfaced, or proactively detecting criminal activity, for instance, by banning domain names known to be automatically generated by bots.

In this paper, we look at the problem from a slightly different angle: can we design registration policies that make it harder for criminals to register domain names in the first place, without impeding benign registrants? Our objective is to improve existing defenses by making domain ownership more transparent, abusive domain registrations more expensive, and raising the operational risk of registering domain names at-scale for abuse.

In other words, we attempt to complement existing technical work on domain abuse detection and remedial with an exploration of the impact of domain registration policies.

Developing and analyzing an anti-abuse registration policy is challenging. First, we must consider the effects of such a proposal at least on benign users, registrars, registries and ICANN. Second, DNS is a global system deployed across political borders, thereby straddling potentially very different notions of "abuse" or "illegality."

Our contributions include:

- We summarize how domain names are used for different types of online crime, how recent research tackles abusive registrations and whether criminals have a distinctive domain registration pattern that could be leveraged to combat them via domain registration policies (Section 2.5).
- We design a framework to evaluate domain registration policies (Section 6.2).
- We discuss the potential benefits, drawbacks, and challenges of multiple registration policy proposals (Section 6.2.3).
- Using our framework and a game-theoretical model, we evaluate one of the most promising proposals to assess its potential effectiveness against online crime (Section 6.3).

6.2 Registration policy evaluation framework

The goal of our policy evaluation framework is to find potentially interesting and viable proposals for further consideration from a large set of policies. Our framework involves a multi-step process towards selecting policies to fight online crime. First, in section 6.2.1, we compile a set of important considerations for future domain registration policies to be evaluated. Second, in section 6.2.3, we systematically select and evaluate high-level policy ideas to find the ones that are likely useful against online crime and plausible to be implemented by the community. Third, it needs to be more precisely evaluated how each policy would affect different entities in the eco-system. In section 6.3, we built a game theoretical model evaluating the effects of one of our promising policies. Finally, if all the previous steps indicate that a policy could be useful then its real-life implementation should be designed and evaluated. This final stage is not in the scope of our paper, because it needs multiple stakeholders to work on it together.

Table 6.1: Summary of domain name usage

# of domains (order of mag.)	**Ref.**	**Abuse**
1,000,000	[71],[86]	Spam
100,000	[71],[86],[105]	Malware
100,000	[71],[86],[105]	Phishing
10,000	[71]	Botnet C&C
1,000,000	[128]	Typosquatting
100,000	[83]	Combosquatting

Table 6.2: Cost of online crime

Abuse	**Ref.**	**Income magnitude (USD)**
Online banking:		
- phishing	[36],[105]	100,000,000
- malware (customer)	[36],[132],[105]	10,000,000
- malware (business)	[36],[105]	100,000,000
Fake antivirus	[36],[132],[104]	10,000,000
Copyright infringment	[36]	1-10,000,000
Illegal Pharmacies	[36],[132]	10-100,000,000
Scams (other than banking)	[36],[132],[123]	10-100,000,000
Spamvertisement	[132]	10,000,000
Click fraud	[132]	10,000,000
Botnet PPI	[105]+[42]	1,000,000

6.2.1 Policy considerations

The domain name registration ecosystem includes a vast number of entities with complex interactions and connections. In this section, we outline the minimum set of entities one must consider when designing a registration policy.

At the bare minimum, a policy proposal should discuss the effects on the entities we discussed in section 2.5: registrants, registrars, registries, and ICANN.

ICANN. An overwhelming part of ICANN's revenue originates from gTLD domain sales, gTLD applications and maintenance fees [72]. Consequently, a policy intervention leading to a significant drop in the number of domain registrations or gTLDs operated would adversely impact ICANN, the main governing body of the domain registration ecosystem. At the same time, one of ICANN's goal is to ensure a secure operation of domain name registrations. "The mission of the Internet Corporation for Assigned Names and Numbers ("ICANN") is to ensure the stable and secure operation of the Internet's unique identifier systems as described in this Section 1.1(a) (the "Mission")."[73]

Registries. Registries' sole revenue is the fees from domain registrations. A drop in the number of registrations would obviously impact them negatively. Halvorson et al. [58] found that, at the time of their study, only 10% of new gTLDs were profitable. They estimated, using their most optimistic model, that 10% of new gTLDs would not become profitable even after ten years of operation. Therefore, we need to consider how stricter registration policies might make it even harder to make a TLD profitable. We also need to consider to which extent a specific TLD might contribute to the Internet community at large.

Different registries also have different incentives and rules to adhere to. Registries operating gTLDs remain profit-oriented, but they need to conform to ICANN's policies. Registries of ccTLDs are controlled (or operated, in certain cases) by their government. As such, countries more economically affected by cybercrime might have stronger incentives to adopt stronger defenses. On the other hand, some other governments might not suffer much from online crime, and at the same time may see a significant proportion of their GDP coming from domain registration fees. (Tokelau [144], governing the `.to` domain is one such example.) In short, the *economic* incentives to fight (domain registration) abuse strongly differ from country to country.

Incentives for policy change. Every registry operating a gTLD must follow their agreement with ICANN and therefore ICANN has the power to control their registrations policies. However, ICANN follows a multistakeholder model, where decisions are made based on the inputs of many entities such as governments, registrars, and registries. Countries own ccTLDs thus registries operating these ccTLDs must follow their agreement with the country for the specific ccTLD they operate. In this setting, the ICANN community and different countries have a big weight in deciding which policies will be adopted. Many countries suffer from online criminal activities and therefore they are likely to support policies targeting malicious registrations. As discussed in section 2.5.2, ICANN is already working on a new registration directory service and so it seems ICANN is also determined to work out some of the current problems with domain name registrations. And while it is possible that ICANN would tolerate some financial loss for social good, it remains unlikely they would support a proposal seriously impacting their revenue.

Registrars. Registrars are responsible for selling domain names to users and therefore registries and ICANN depend on them for their own revenue. This gives registrars an important place in ICANN's multistakeholder model. At the same time, registrars compete for users' business, which limits their profit margin on domain sales. Because of this low profit margin, many registrars use domain registrations as a gateway to increase their customer base and to cross-sell hosting services. For example, GoDaddy offers domains for $0.99, which makes their domain sales unprofitable for the first two years; GoDaddy makes up for the lost revenue by gaining customers for its hosting services. In other words, to be acceptable to registrars, a policy should not result in a decrease in customer volume, which is a more important metric than actual income from domain sales. Additionally, malicious users usually rely on separate hosting infrastructure (compromised hosts, or bulletproof servers, depending on the type of criminal activity taking place), thus a decrease in malicious registrations should only modestly affect honest registrars.

Registrants. Benign registrants value their domain names—be they indicative of a brand, or a mere vanity registration. We can assume that any change to that name, including changing the TLD, would decrease the value of the domain name for them. It is hard to estimate the exact value of a domain name to a user, but it is safe to assume that an increase in price by an order of magnitude would discourage many individual users from registering domain names. At the same time, a more modest increase, e.g. less than doubling the price, would not discourage most users from buying domain names. We discussed how malicious users depend on domain names in Section 2.5.1: different from benign users, they generally value volume over specific domains (with the exception of the various "squatting" scams).

Sensitive registrants. Many policies proposed to combat miscreants, as a side-effect, could negatively impact registrants' freedom of speech. For instance, "real name policies" used by certain entities such as Facebook, have met with significant community push back, as they can ostracize entire communities (abuse survivors, for instance).

Fortunately, the problem is not entirely unsolvable, even if we advocate for stronger identification requirements for registrants. First, privacy protection services can shield the identity of a registrant from the general public. This solution is similar to OPOC mentioned in section 2.5.2 and similar to current privacy services. However, the registrant would still own the domain name and would be responsible for its usage. Additionally, the privacy service would still need to provide data for law-enforcement agencies and security researchers.

Second, sensitive registrants might be able to register domain names at TLDs that are not operating in the jurisdiction of their government. This solution would make it hard or impossible for the registrant's government to associate them with the domain based on registration data.

Third, supporting foreign organizations could offer these users subdomains under their own domain or could even proxy ownership for them. This proposal would shield registrants fearing their own government.

Binding vs. non-binding policies. As we discussed above, the ecosystem is diverse enough that different registries will have different obligations and incentives, thus it is unlikely they would all agree to a common specific registration policy. Consequently, it is

beneficial to evaluate three levels of collaboration for each proposed policy: whether only a few registries, most registries, and all registries implement the proposed policy.

In case the proposed policy is non-binding, making abusive domain registrations harder will decrease the abuse at the adopting TLDs, but as observed by Liu et al. [97], malicious registrants will adopt and start registering domains at other TLDs. If, on the other hand, the policy is binding, that is, if ICANN mandates policy implementation, the vast majority of gTLDs will have to collaborate; individual ccTLDs may then be forced to follow suit, as the critical mass of collaborative TLDs would make it easier to blacklist malicious domains registered at shadier, non-collaborative registries.

Hacked domains versus abusive registrations. Often hacked domains and abusive registrations can be used for the same purpose. No matter how successful a domain registration policy is, it will not affect hacked domains used to support online crime. Nevertheless, a successful anti-abuse policy would *force* miscreants to primarily resort to hacked domains—which is more complicated than simply registering a domain. Recent advances in web security (e.g., predictive analytics [119]) may further increase the difficulty of compromising existing domains at scale. In conclusion, we need to tackle both malicious registrations and domain name compromises to solve the general issue with criminals using domains for malicious purposes.

Definition of abuse and illegal across borders. It is important to define the terms "abuse" and "illegal" for domain registrations. We would define a domain name registration to be abusive if it was registered for illegal purposes based on the laws of the country where the TLD's registry resides. For each TLD, the definition of abuse would be different but could have a reasonable common core, which would include illegal activities such as squatting, spamming, scams, phishing, illegal content and goods distribution, botnet operations etc. Building on this common core, registries could take actions against these malicious registrations or could introduce fines or security deposits to make criminal efforts more expensive.

6.2.2 On the potential of domain registration policies

Based on existing research, Table 6.1 summarizes the orders of magnitude of blacklisted domains or squatting domains registered every year for each type of abuse. Table 6.2 shows the estimated order of magnitude of yearly income for different types of online criminality activity.

These estimates must be treated with caution. Criminal income is in particular notoriously difficult to pinpoint and can be either overestimated or underestimated. On the other hand, the number of domains blacklisted is likely underestimated because blacklists try to minimize false positives.

Looking at Table 6.1, we can observe that abuse yields earn hundreds of millions of dollars in revenue per year, and corresponds to millions of domain names being registered each year. Straightforward averaging would yield a criminal income per domain name to be around a hundred dollars. Clearly, using the average is not suitable because the effectiveness of criminals and domain registration needs are highly variable. For example,

spear phishing campaigns or targeted scam attacks may require only a couple of domain names, each bringing in a very high revenue per domain, and therefore making the designing of policy-based countermeasures challenging. On the other hand, a number of abuses require a lot of domain names and are less effective on a per-domain name basis. In Table 6.2, spamvertisement jumps out as a potentially good candidate to be affected by stricter registration policies. Typosquatting is also a good example, where most domains would become unprofitable if the cost of malicious registrations increased. In general, previous research has – time and again – shown that online crime is a heavy-tailed business, where a few, major, actors account for the vast majority of the ecosystem [61, 91, 93]. Thus a successful registration policy proposal could decrease the number of criminals by further pushing out the less successful ones into bankruptcy.

6.2.3 High-level policy proposal discussions

We attempt to systematically build a list of policy proposal based on the tools available for registries and ICANN. These basic tools include domain pricing, level of identity verification, fees, security deposits, incentives for good behavior, lexical prediction and combinations of these policies.

Table 6.3: Table evaluating the potential effects of the policy proposals discussed.

	Malicious registrants (One reg. adopts)	(Most adopt)	**Benign registrants**	**Registrars, registries, ICANN**	**Sensitive registrants**	**Adoption probability**
Small Price Increase	local	yes	maybe	small	no	possible
Strict Identity Verification	local	yes	small	yes	yes	possible
Fines or Security Deposits	no	no	small	small	no	unlikely
Anti Bulk Registration	local	yes	small	yes	yes	possible
Large Price Increase	no	yes	yes	yes	no	unlikely
Protocol Separation	no	yes	small	yes	no	unlikely
Incentivizing Registries	local	yes	no	yes	no	possible
Anti-squatting	yes	yes	no	yes	no	possible

Proposal 1: Small increase in the registration price. For the many criminals with a small profit margins, even a small increase in pricing could be discouraging from registering domain names. The question is what price increase would impact malicious registrants but not benign registrants. Future work should attempt to provide accurate estimates of the price-sensitivity of (benign) registrants, to infer possible tuning knobs for such increases.

Proposal 2: Stricter verification requirement. Currently, the overwhelming majority of registries do not have any identity verification in place allowing criminals to register domain names with as many identities as they want. Stricter identity verification would require miscreants to use high-quality fake or stolen identities, imposing an additional cost on them. The operational risk of criminals would also increase – as procuring (a large number of) stolen identities in itself is a potentially risky endeavor.[2]

Examining further the effects of different identity verification schemes, the most important attributes to look at are their evadability, cost, and accessibility. On the one hand,

[2]Ross Ulbricht, the creator of the Silk Road website [45], actually had an initial encounter with the police, when ordering a bunch of fake driver's licenses from his own marketplace.

completely forgoing verification is cheap, accessible, but also easy to evade—the attacker does not need to take any specific precautions to do so. On the other hand, in-person verification is expensive and has limited accessibility, but it is also expensive for an attacker to defeat. SSL-extended validation style of verification is hard to defeat, but it would negatively impact most regular users, as it is both expensive and lacks accessibility.

To find the balance between cost, accessibility, and evadability, one suggestion is to use a combination of identification documents, which are hard to find on black markets, with automated face recognition and liveness detection. Such a system could be affordable and accessible for benign users, but expensive to evade. Matching credit cards and identity documents are scarce on online black markets and are more expensive than requiring non-matching documents. Adding phone number and email verification (potentially from a big email service provider) can also raise the cost of Sybil attacks. While state-of-the-art liveness detection and face recognition can be evaded [145], evasion requires higher technical skills and more investment (per identity) from the attackers. One of the biggest online identity verification provider informed us that they sell their product for $0.5-2 per identification. Our suggestion is very similar to their automated solution and also takes black market pricing (using data from [120]) into account. In other words, this approach could have a low cost, be accessible and would be potentially expensive for criminals.

Standardized registration policy and increased strictness of identity verification would also allow for better defenses detecting Sybil attacks. For example, an IP reputation system could be used to make Sybil attacks harder by tracking the number and kind of registrations from IP addresses. Luckily a lot of work has been done in this space led by tech companies such as Google, Facebook, and Jumio.

Proposal 3: Fines and security deposits. Fines are traditionally used to incent citizens to remain law-abiding. Conversely, criminals already hide their identity or operate in jurisdictions different from where they are located, making enforcement mechanisms such as fines hard to deploy. While security deposits could be useful against criminals, it might dissuade regular users from registering domains. However, fines can indirectly affect malicious registrants, by making "outsourcing" less appealing. Specifically, fines could disincent otherwise law-abiding people from registering domain names with their own identity on behalf of criminals. Finally, security deposits could be used in case of suspicious domain registrations such as typosquatting or a sudden large amount of registration attempts from a developing country.

Proposal 4: Anti-bulk registration policy. The anti-bulk registration proposal builds on the observation that spammers, botnet operators, typosquatters and many other online criminals are banking on the fact that they can access a large number of domain names cheaply to avoid reputation systems and blacklists. By making bulk registrations hard and expensive we target the abundance of cheap domain names for online criminals. Additionally, for most TLDs, the identity of registrants are not validated leading to a lack of transparency in the ownership of domain names.

The policy changes we propose are strict verification of identity at registration time, increasing domain price with the number of domains registered and optionally a security fine/deposit to thwart malicious behavior. Strict identity verification is important to make

Sybil attacks expensive and to increase transparency. Increasing domain name price as the function of domain names owned is crucial to make bulk registrations expensive and at the same time allow users to own a few domains for an inexpensive price. This policy proposal leverages the benefits of several previously discussed policy options, while it minimizes their drawbacks.

There are only a handful of legitimate reasons for a registrant to own more than a couple of domains. Domain name speculators buy large quantities of domain names in hope to sell them later for profit or earn money from incoming traffic (e.g. type-in navigation). Sometimes these domains lead to malicious content when domain owners employ more lucrative but more questionable parking services [35, 137]. As explained before, the goal of the domain name system is to give memorable names to resources on the Internet for users. Speculative domain registrations are a parasitic byproduct and are not serving the primary goal of the domain name system. A better example of benign registrations is defensive registrations. Users defensively register many variants of their brand name to protect it from domain squatting, typosquatting and other variants of name squatting. A simple algorithm could decide if a registration is defensive or not and thus a registrant could register these domains on the base price. Finally, hosting providers also often own their customers' domain names, this could be resolved by proxy ownership, where both owners are responsible for potential misuse of the domain name.

Proposal 5: Considerable increase in the registration price. This proposal's aims at making domain names less desirable for miscreants indirectly. We plan on achieving this by making the hierarchy of ownership in the domain name system deeper. Currently, the hierarchy is only two-level deep: TLDs and people registering domains under these TLDs (most often second level domains). Even though now nearly two thousand TLDs exist, only a handful of them is actually used by most Internet users. This means that domain name reputation systems basically have to work only with registered (mostly second level) domains. The proposal is to use pricing to motivate the usage of lower level domain names for domain ownership and have a different use for the different level of domain names.

More specifically, the proposal is to make domain names very expensive such that only big companies/brands/organization could afford them. This would force personal websites and small business to lower levels (mostly third level domains). We would call these domains first- and second-tier domains respectively. This proposal would make first-tier domains not economical for malicious usage. Additionally, penalties could be put in place to enforce first-tier domain owners to keep their namespace clean.

To discuss how registrants could cope with this change, consider the example of a florist from Pittsburgh named Jane. Jane would not be able to purchase the domain `janetheflorist.com`, which would be out of her price range. Instead, she could join together with small businesses in Pittsburgh and buy `pgh.com` and then use `janetheflorist.pgh.com`. Registration requirement under `pgh.com` would be strict and would require individuals to own a business in Pittsburgh,[3] and for this reason, abusive second-tier registrations would be cumbersome and rare under `pgh.com`. Free-speech advocacy organizations could buy domains such as `freedom.com` to allow anyone to have a web presence anonymously

[3]Similar, in that sense, to the policies on certain ccTLDs such as `.fr` or `.us`.

and cheaply by allowing them to use their namespace, e.g., `mypolitics.freedom.com`. To mitigate abusive second-tier domains, `freedom.com` would probably only allow a limited web presence for its second-tier domains.

The main problem with this proposal would be the transition from the current system. Many people have marketed and built out a brand around their domain names, changing them would be highly undesirable and would cause potentially financial losses to these users. In fact, the current trend in DNS – flattening of the namespace by the introduction of new gTLDs – flies very much in the face of this proposal.

Proposal 6: Combining domain registrations with SSL and protocol separation. Here, the idea is to have different levels of trust in domain names based on the level of identity verification and price paid for the domain name. Based on the level of trust, different application protocols would be accepted for different domains. As an example, email protocols would need a higher level of trust than running webpages with certain restrictions. However, to allow webpages to offer files for download and to allow these files to leave the browser's sandbox would also require a higher security level.

These security levels would aim at directly making domain names too expensive for certain types of cybercrime. For a domain to be used in a certain protocol, it would need to be priced according to the protocol's potential for malicious usage. In addition, this approach could be easily coupled with SSL domain validations. The main problem is that this proposal would need to be adopted by most users of the Internet, allowing them to decline connections from low security level domain names.

Proposal 7: Incentivizing domain registries to fight abuse. As observed by Korczynski et al.[87], we could incentivize domain registries or registrars to decrease abusive registrations based on the actual abuse found at these registries. More specifically we could increase or decrease the per domain fee they pay based on the number of domains blacklisted in their TLDs. Participants would need to agree on the definition of "abusive" and would also need to agree on which entities could decide if a domain was abusive, hence a penalty is necessary. This policy has relatively few drawbacks, making it a promising avenue for further investigation.

Proposal 8: Anti-squatting analysis. Typosquatting, combosquatting, soundsquatting and cybersquatting share that they can be identified based on lexical features with good true positive rate and moderate precision. Therefore proven typosquatting domains could be removed and new registrations for these users could be hardened. In case these registrants cannot present convincing proof of their benign intent then a security deposit could be required from them.

This approach would be highly effective against squatting and would also impact phishing and scam attacks which frequently rely on lexicographically-close domain names to fool victims. This proposal is also non-binding, thus it is effective even if only one registry implements it. Furthermore, it does not negatively impact benign registrants – making it also a promising prospect.

Summary of proposals. Table 6.3 summarizes the effects of the previously discussed policies. In general, we would like to find policies that are effective against malicious registrants but do not hurt benign registrants. Most policies would impact registrars,

registries, and ICANN because they decrease the number of domain registrations. The question is how much they would be impacted and can we counter-balance it somehow? If identity verification is required then sensitive registrants will be impacted, we discussed in Section 6.2.1 what options they have to mitigate the policy's impact on them. Last, we would like to focus on policies that are not unlikely to be implemented.

It is possible to combine multiple policy proposals to increase their effect on online criminals. For example, implementing the anti-bulk registration proposal does not mean that incentivizing registries and registrars to be more due diligent in banning malicious domain could not be effective. Additionally, adding the anti-squatting policy could help remove high-value domains that were not affected by the previous two proposals.

6.2.4 Policy proposal implementation challenges

Dependence on blacklisting. Domain registration policy efforts need to be in harmony with current blacklisting efforts. Indeed, without effective blacklisting, malicious users do not need to exhibit a registration pattern substantially different from benign users. The better the blacklisting efforts, the lower the per-domain revenue of online crime, making our registration policy proposals even more effective. At the same time, some of these registration policy proposals could also make blacklisting more effective. Making domain registrations harder for criminals would corner them into fewer TLDs and would decrease the general noise and opaqueness of the current chaotic situation of domain registrations.

Unaffected domains. Certain malicious domains are not affected by changes in pricing and verification requirement. For example, targeted scams or phishing attacks might only use one typosquatting domain name to confuse the customers of a bank. Abusive domain name registrations that do not show distinctive pattern compared to benign registrations are impossible to affect directly via registration policies. Indirectly registries and registrars could be incentivized to clean their domain more diligently.

Data management problems. The current WHOIS system is often used by researchers and security analyst to learn about the ownership of malicious domain names. As discussed in section 2.5.2, there are two main problems with the current WHOIS database. First, the data is often inaccurate since identity verification is minimal at most TLDs. Second, it has been shown that WHOIS information (when correct) is sometimes used to deliver email, mail or phone spam [89]. Implementing the anti-bulk registration proposal would allow registries to increase the accuracy of their WHOIS information. It would also allow the elimination of WHOIS spam by using pseudonyms. Using pseudonyms would still allow researchers to tie domain names together owned by the same person or company. Alternatively, a gated access to WHOIS data could be introduced as proposed by ICANN (section 2.5.2).

6.3 Game-theoretic analysis of the anti-bulk registration policy proposal

Based on our high-level analysis, we next analyze the anti-bulk registration policy proposal more in depth. The goal is to capture the benefits of increasing domain registration prices while attempting to minimize the drawbacks incurred to legitimate registrants.

6.3.1 Formal model

The game we design resembles a Stackelberg game variant. In this game, registries are the "leaders," who decide their strategy of pricing and identity verification first. The registrants are the "followers," who decide where and how many domains they want to register.

Our design is different from a classical Stackelberg game in that we have multiple leaders and followers. We model registries as two leading players. One leader is a group of collaborating registries coordinating their registration policy strategy to combat malicious registrations. The other leader is the group of non-collaborating registries, who are not impacted negatively by malicious registrations. Registries are playing a simultaneous move game. Registrants then respond to strategies selected by registries.[4]

Our analysis consists of evaluating the pure strategy Nash equilibria between registries and analyzing what would be the best response of non-collaborating registries and registrants to certain strategies chosen by collaborating registries.

Our proposed model simplifies the domain registration ecosystem by only considering registries and registrants. On the one hand, we consider registries as players because they control both registration policies and pricing for the TLDs operated by them, and thus capture the essential mechanisms in the ecosystem. On the other hand, registrar market is extremely saturated, to the point where registrars often sell domain names at or below cost. Therefore, registrars do not significantly affect the final pricing and registration policy for registrants. ICANN and governments could potentially impact registration policies set by registries. We incorporated them in our model indirectly as a parameter in the utility function of registries. ICANN, registries, and registrars have a voice in the registration ecosystem and for a policy to be implemented it is important for their revenue not to be significantly impacted. We can estimate how they are affected by the decrease in the number of domain registrations and the decrease in the number of registrants. In Section 6.2.3, we discussed how registrants with special needs would be affected and how could they be handled. As such registrants would not significantly impact the game, we assume them away and do not model them.

Players. Let us define benign registrants as $b \in B$, and malicious registrants as $m \in M$. Registries are $r \in R$. For simplicity (and without loss of generality) we assume that there are only two registry players, i.e., $R = \{r_c, r_{nc}\}$, where r_c is the group of collaborating registries and r_{nc} is the group of non-collaborating registries. With this simplification,

[4]We assume registrants are best responding to the strategies of the registries and we leverage this to calculate the registries' utilities

we do not need to model the interaction between collaborating registries as part of their strategies and utility functions.

Strategies. A malicious registrant m can decide how many domains $n_{m,r}$ they want to register at registry r and how many fraudulent (i.e., fake or stolen) identities $i_{m,r}$ they want to purchase to use at registry r. The maximum number of domains that a malicious registrant can profit from is $n_m^{\max}$, thus $\sum_{r \in R} n_{m,r} \leq n_m^{\max}$. A benign registrant b can decide how many domains $n_{b,r}$ they want to register at registry r. $i_{b,r} = 1$ for all $b \in B$ because we assume benign registrants have only one identity. Similarly to malicious registrants, the following constraint holds for benign registrants: $\sum_{r \in R} n_{b,r} \leq n_b^{\max}$.

A registry r can define its pricing function $C_r(n, i, \alpha_r, \beta_r)$ by setting the base price α_r and the discount (or penalty) for registering more than one domain β_r. The number of domains to be registered n is divided by the number of identities used i, to represent optimal fraudulent identity allocation by malicious users.[5]

$$C_r(n, i, \alpha_r, \beta_r) = \sum_{j=1}^{i} \alpha_r \cdot (\frac{n}{i})^{\beta_r}$$

The registry can also define how hard it wants to make identity verification by defining θ_r, the cost of one verification. θ_r will also define the cost of buying a fraudulent identity λ_{θ_r}.

Utility functions.

The utility function of **malicious registrants** consists of four components: the value V_m is derived from the criminal activity, the cost of registering domain names C_r (the same function as defined above), the cost of fraudulent identities F_m and $\theta_r \cdot i_{m,r}$ the cost of verification.

$$U_m = \sum_{r \in R} \Big[V_m(n_{m,r}, i_{m,r}, \gamma_{m,r}, p_{\text{bl}}) - C_r(n_{m,r}, i_{m,r}, \alpha_r, \beta_r) - F_m(i_{m,r}, \lambda_{\theta_r}) - \theta_r \cdot i_{m,r} \Big]$$

The per-domain income from perpetrating a specific type of criminal activity is represented by $\gamma_{m,r}$. For a specific criminal activity, $n_m^{\max}$ is the maximum number of domains that are useful to register.

Finally, p_{bl} is the probability of an individual domain being blacklisted. Using p_{bl} we calculate the expected number of domains blacklisted given the number of fraudulent identities used i. The formula below models how having domains blacklisted and owning too few identities leads to the blacklisting of other domains registered using the same identities. As we do not know the exact value of p_{bl}, we will evaluate a range of possible values.

$$V_m(n_{m,r}, i_{m,r}, \gamma_{m,r}, p_{\text{bl}}) = \gamma_{m,r} \cdot n_{m,r} \cdot (1 - p_{\text{bl}})^{\frac{n_{m,r}}{i_{m,r}}}$$

[5]Our simulation solves the integer version of this problem. For example, $n = 10 \wedge i = 3$ means that two identities will have three domains associated with them and one identity will have four domains associated with it.

The cost of buying a fraudulent identity is λ_{θ_r} and multiplying it by $i_{m,r}$ gives the total cost of fraudulent identities for a malicious registrant. When the value of λ_{θ_r} is unknown, we will test several interesting values.

$$F_m(i_{m,r}, \lambda_{\theta_r}) = i_{m,r} \cdot \lambda_{\theta_r}$$

The utility of a **benign registrant** consists of the value of the domain names V_b, the cost of registering the domain names C_r and $\theta_r \cdot i_{b,r}$ the cost of verification.

$$U_b = \sum_{r \in R} \Big[V_b(n_{b,r}, \gamma_{b,r}) - C_r(n_{b,r}, 1, \alpha_r, \beta_r) - \theta_r \cdot i_{b,r} \Big]$$

The average value of a domain name for a registrant is $\gamma_{b,r}$ and the maximum number of domains a registrant can profit from is $n_b^{\max}$:

$$V_b(n_{b,r}, \gamma_{b,r}) = \gamma_{b,r} \cdot n_{b,r}$$

The utility of **registries** consists of two parts: C_r the fees from domain registrations and the cost of online crime.

$$U_r = \sum_{j \in B \cup M} \Big[C_r(n_{j,r}, i_{j,r}, \alpha_r, \beta_r) \Big] - \rho_r \cdot \sum_{m \in M} \Big[V_m(n_{m,r}, i_{m,r}, \gamma_{m,r}, p_{\text{bl}}) \Big]$$

The only new parameter in this equation is ρ_r representing how important the cost of online crime is for a registry. For registries operating in countries where the cost of online crime is higher than the revenue from domain name sales, ρ_r is high. Example of high ρ_r could be countries with high GDP because these countries are more frequently targeted by online crime. In countries where the domain name fees are higher than the cost of online crime, ρ_r is low. For example, $\rho_r = 0$ for Tokelau's ccTLD .tk, because the domain name fees are a significant part of their GDP [144] while they are not affected by these criminal activities. The cost of online crime ρ_r could be influenced by ICANN for gTLDs.

Parameter estimation, simplification and assumptions.

In this section we discuss the parameters of the model and how we can estimate them or what assumptions we have to make.

$\gamma_{b,r}$ **the value of a domain name for** b**.** For most benign users we assume that they would still buy their domains, if domain prices would rise only a little bit (for example less than doubles). But they would not buy their domain, if the price would increase any more than that. This would make $\gamma_{b,r} \approx 20$ for average users. ($\gamma_{b,r}$ is expressed in dollars/domain.) In future work, we hope to estimate Alexa's top 1 million domains' traffic using the Zipf curve we fitted on Alexa traffic estimates and multiply it by how much Google pays per a thousand impressions.

$n_b^{\max}$ **the maximum number of domains that benefits** b**.** We assume $n_b^{\max} = 1$ for the sake of simplicity. An extension to the model could estimate the domain ownership distribution based on WHOIS data.

$\gamma_{m,r}$ **is the value of online crime and** $n_m^{\max}$ **is the maximum number of domains that benefits** m**.** For malicious registrants $\gamma_{m,r}$ and $n_m^{\max}$ is different for each online crime type. We have estimated these values in Section 6.3.2 for typosquatting and pay-per-install services. We also model a general online criminal with varying values per domain revenue $\gamma_{m,r}$.

θ_r **is the cost of one identity verification.** While θ_r is the choice of registry r, we might want to simplify our model and consider values based on real-life examples. Most registries do not verify the identities of users, which means $\theta_r \approx 0$. For a large identity verification service to do face recognition combined with liveness detection and document verification means $\theta_r \approx 1$. We conjecture that for the rigorous SSL extended verification $\theta_r \approx 100$ in order of magnitude. Our suggestion of combined document verification would be a modification of existing services' verification systems and it should cost approximately the same, conservatively we estimate $\theta_r \approx 4$.

λ_{θ_r} **is the cost of a fraudulent identity.** When $\theta_r = 0$ then λ_{θ_r} is also zero. However, the value of λ_{θ_r} is questionable if $\theta_r \geq 1$. We estimate its values based on online anonymous marketplace prices. However we use this estimate with caution and we test our model with multiple possible values for λ_{θ_r}.

ρ_r **is the cost of online crime for registries.** We discussed ρ_r the cost of online crime for registries earlier. A simplifying but reasonable assumption is $\rho_r = 0.1$ for the collaborative registries and $\rho_r = 0$ for the non-collaborative registries. In case of collaborative registries, we test multiple potential values of ρ_r

p_{bl} **is the probability for an individual domain to be blacklisted.** We test multiple values of p_{bl}. Setting $p_{\text{bl}} = 0$ means that we are not modeling blacklisting and criminals do not need to worry about it. $p_{\text{bl}} = 1$ means that domains are always blacklisted before crooks can profit from them. A small value for p_{bl} is reasonable because domains are often blacklisted after the online crime was already perpetrated.

The α_r **and** β_r **of the pricing function** C_r**.** To simplify our model we consider only certain values of α_r and β_r, such as $\alpha_r \in \{1, 2, 10, 100, 1000\}$ and $\beta_r \in \{0.95, 0.99, 1, 2, 3\}$. Having a finite number of strategies allows us to compute the game's payoff function based on the registrants' best response.

How registrars, ICANN, and countries are modeled in this game? In this formulation, registrars are represented as part of the registries. If fewer users are registering at a registry or the payoff of the registry decreases due to the strategies chosen by the players, would mean a decrease in the registrar's utility. ICANN and countries are represented in the choice of ρ_r for their TLD and the registrants TLD preferences $\gamma_{b,r} \wedge \gamma_{m,r}$.

6.3.2 Seeding the model with data

Estimating typosquatting domain ownership and revenue. First, we model the number of domains owned by typosquatters based on WHOIS clustering done by Szurdi et al. [126]. It is important to note that this estimate is a lower bound on the number

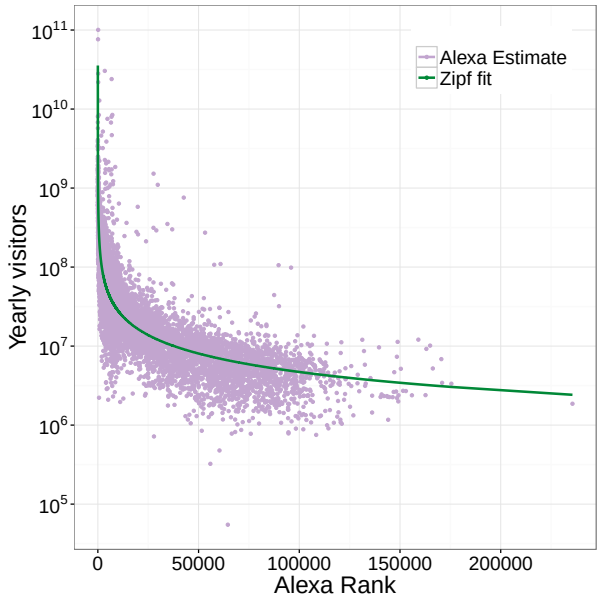


Figure 6.1: On these plots we can see Alexa's estimate of the yearly visitors at domains as the function of their Alexa rank. The green line represents the Zipf curve we fitted in log space on Alexa's estimate ($R^2 = 0.76$).

of domains owned by typosquatters because WHOIS data can be easily spoofed, thus one typosquatter might look like many entities in our clusters. We also exclude privacy protected typosquatting domains, which means the probable exclusion of some of the worst typosquatters. As a further precaution, we only consider a registrant to be a typosquatter if she owns at least ten typosquatting domain names.

We estimate the revenue of typosquatters as:

$$\gamma_{m,r} = \text{Traffic}_{orig.} \cdot Rate_{mistype} \cdot CTR \cdot PPC$$

Figure 6.1 shows our estimate of the number of visitors domain names receive ($\text{Traffic}_{orig.}$) by fitting a Zipf curve on Alexa's estimate of the traffic received by top ranked domains. We use the estimates by Moore et al. [106, 107] directly for PPC and the average percent of $\text{Traffic}_{orig.}$ going mistakenly to typosquatting domains instead of the original domain. We know that a typosquatting domain's quality depends on many factors, therefore we use Szurdi et al.'s [126] observations to estimate $Rate_{mistype}$ for individual typosquatting domains. We use Google's case study [56] to estimate CTR.

For typosquatters, we modeled V_b slightly differently compared to the formula in section 6.3.1. We took into account that their domains have significantly different values and we assumed they prioritize registering the best of their domain names.

Estimating botnet revenue per domain name. The cost of a thousand unique installs on bots cost from \$7-\$8 to \$100-\$180 [42]. For an upper bound in order of magnitude, we calculate with a cost of \$100 per a thousand bots. We conjecture that a year a machine is sold in this fashion ten times. This leads to our estimate of $R_{bot} = \$1/\text{bot/year}$ income for botnet operators. We assume that these bots are solely used for pay-per-click installs.

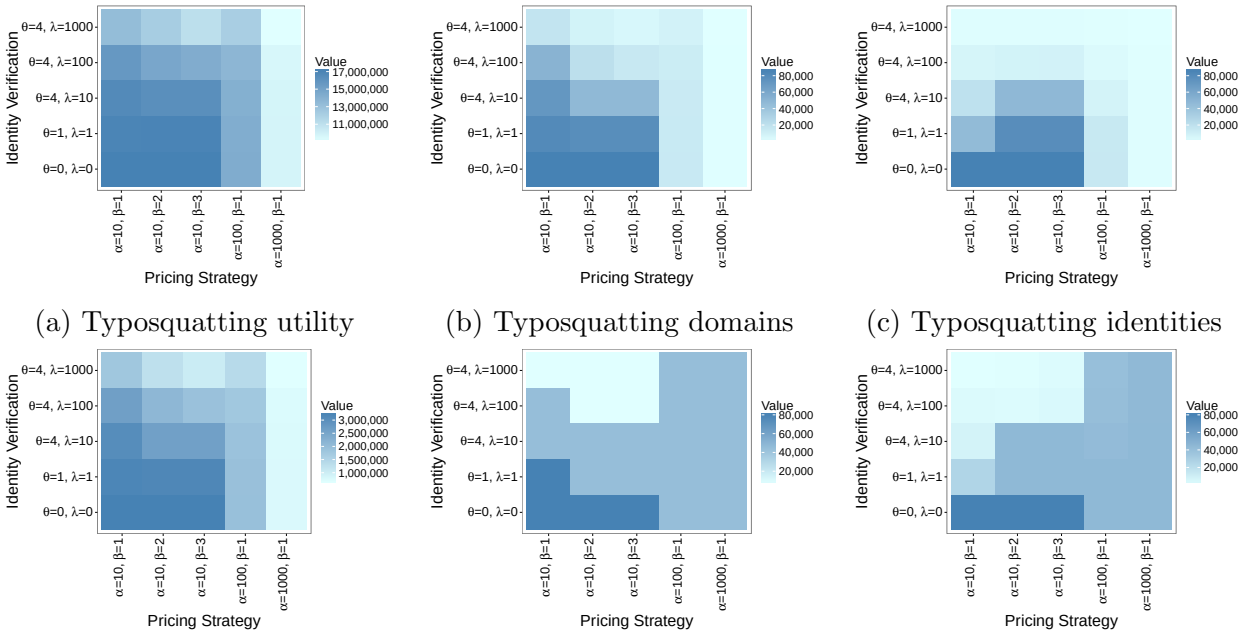
We also estimate that the time to blacklist domain names is one day on average [60, 85, 113]. This lead to the following function to calculate revenue per domain.

$$\gamma_{m,r} = N_{bots} \cdot R_{bot} \cdot \frac{min(N_{domains}, 365)}{365 \cdot N_{domains}}$$

Representation of malicious registrants. The per domain revenue $\gamma_{m,r}$ can vary by orders of magnitude for different online criminals. In our model, we represent miscreants based on how good or bad their per domain revenue is. We use criminals anywhere in the range $\gamma_{m,r} = (1, 3000)$. Criminals with low per domain efficacy include spamvertisement, small botnet operators, and typosquatters. Examples of decent $\gamma_{m,r}$ are general scam and phishing attackers and better typosquatters or botnet operators. Finally, certain criminals need only a couple of domains with high potential revenue such as spear phishing and banking trojans.

6.3.3 Analysis

We calculated the Nash Equilibria for a wide range of parameter values, which resulted in many different games and for each game potentially different sets of equilibria. First, we analyzed all these equilibria together to see if we can distill any takeaways that are true for all of them. Second, we evaluated a more precise analysis of specific scenarios. We start with a scenario we believe to be the most realistic and then we tweak the parameters to see how the results change given different scenarios.



(a) Typosquatting utility (b) Typosquatting domains (c) Typosquatting identities

(d) Botnet utility (e) Botnet domains (f) Botnet identities

Figure 6.2: Effects of different pricing and identity verification strategies on malicious registrants.

Registry nash equilibria analysis. When we evaluated our model and found the Nash equilibria, we observed that registries set α_r such that they do not lose their customers to other registries. At the same time, they also select the largest α_r where registrants still choose them. Consequently, they primarily use β_r to deter criminals.

Our model does not yield a Nash equilibrium in which the registries can discourage the largest botnet operators from registering domain names. Medium and large botnet operators decide not to register domain names if a combination of high fake identity cost and low utility from registering domain names at non-collaborating registries co-occur. Low utility of registering domains at non-collaborative registries models the situation when most or all registries are actually collaborating and thus non-collaborative registries became isolated.

The probability of blacklisting an individual domain greatly affects our model. Trivially, in the non-realistic case of $p_{\text{bl}} = 1$, no abusive registrations occur. At the other end of the spectrum, if $p_{\text{bl}} = 0$ it becomes harder for registries to do something about malicious registrations, but they are still able to affect certain criminals by increasing prices. When we increase p_{bl} it becomes easier for registries to discourage malicious domain name registrations. This captures the synergy between blacklisting and domain registration policies to combat abusive domain registrations.

Registration policy scenario analysis.

The base scenario. First, we start by making a set of assumptions about the input parameters of the model. While we choose a realistic starting point, we will also analyze the effects of changing these assumptions and the values of these parameters.

In the base scenario we assume that users prefer their current TLDs, but if the pricing increases significantly in their current TLD they are willing to switch to another TLD. This is represented by setting $\frac{\gamma_{b,rc}}{\gamma_{b,rnc}} = 10$. We assume that collaborative registries care about abusive registrations and non-collaborative registries do not care, leading to $\rho_{r_c} = 0.1$ and $\rho_{r_{nc}} = 0$. We assume that the probability of blacklisting domains is not zero, but it is low $p_{\text{bl}} = 0.01$. Finally, we assumed that benign registrants approximately register a hundred times more domains than malicious registrants. This assumption is reasonable because each year there are millions of domains registered for abusive purposes and there are hundreds of millions of domains registered by benign users.

With these assumptions, we tested 25 different combinations of pricing and identity verification strategies for collaborative registries. The values of α_{r_c} and β_{r_c} tested are shown in table 6.4. The different values of $\lambda_{\theta_{r_c}}$ for the same θ_{r_c} symbolize different possible costs for defeating the identity verification method suggested by us in section 6.2.3. We model non-collaborative registries and registrants to be best responding to the strategies of collaborative registries.

Interestingly, in this scenario, benign registrants will always register all of their domains. If $\alpha_{r_c} \in \{100, 1000\}$ then non-collaborative registries will drop their prices to get benign registrants' business. This results in a huge drop in the utility of both the collaborative registries and the benign registrants. Increasing θ_{r_c} leads to a drop in benign registrants utility, but until $\alpha_{r_c} = 10$ they will keep their domains at the collaborative registries.

Table 6.4: Pricing and verification strategies for the base scenario.

Pricing strategy	α_{r_c}	10	10	10	100	1000
	β_{r_c}	1	2	3	1	1
Identitity verification	θ_{r_c}	0	1	4	4	4
	$\lambda_{\theta_{r_c}}$	0	1	10	100	1000

Figure 6.2 shows the effects of different registrations strategies on malicious users. Setting α_{r_c} high has a significant impact on malicious registrants but it also negatively affects other entities in the ecosystem. A better solution is to keep $\alpha_{r_c} = 10$ and increase β_{r_c} and θ_{r_c}. We can see that even a small increase in θ_{r_c} can affect the utility and the domain registration behavior of malicious registrants slightly. Most interesting is setting $\theta_{r_c} = 4$ and analyzing how different possible $\lambda_{\theta_{r_c}}$ affect miscreants. We can see in Figures 6.2a and 6.2d that any value of $\lambda_{\theta_{r_c}}$ has a significant effect on the utility and domain registration behavior of crooks. Analyzing Figures 6.2b, 6.2e, 6.2c, and 6.2f, we observe that at cells corresponding to $\lambda_{\theta_{r_c}} \in \{1, 10\}$ and $\beta_{r_c} \in \{2, 3\}$ some miscreants still keep their domain names but they need to buy a lot of stolen identities decreasing their profit. When $\lambda_{\theta_{r_c}} \in \{100, 1000\}$, most criminals need to switch registries or give up their domain registrations. However, even in the most adversarial settings, the most successful criminals will continue using their domain names.

Table 6.5: The effects of $\lambda_{\theta_{r_c}}$ on malicious registrants, when $\alpha_{r_c} = 10$ and $\beta_{r_c} = 3$.

	Typosquatters			Botnets		
$\lambda_{\theta_{r_c}}$	Utility	# doms	# iden	Utility	# doms	# iden
1000	64.9	7.2	2.1	32.9	9.9	3.5
100	83.5	18.4	10.0	58.9	9.9	5.1
10	94.8	55.2	55.2	80.0	55.0	55.0
1	99.0	90.3	90.3	96.3	55.0	55.0
0	100	100	100	100	100	100

Table 6.5 shows that even a small increase in $\lambda_{\theta_{r_c}}$ has an effect on the number of domains registered and the utility of criminals. $\lambda_{\theta_{r_c}} = 100$ appears to be where the number of malicious domain registration drastically drops. Interestingly, certain combinations of personal documents available on online black markets hover around $100, based on empirical data [120]. However, even drastic drops in domain registrations do not affect the most successful criminals, and have thus a slightly more limited impact on total (aggregate) miscreant utility.

Figure 6.3 shows that only the very few top typosquatters are not affected by $\lambda_{\theta_{r_c}}$ the cost of stolen identities. Surprisingly to us, a few of the top ten typosquatters are also significantly affected by the increase in $\lambda_{\theta_{r_c}}$. It is likely that these typosquatters own many low or average quality typosquatting domain names and therefore they are increasingly

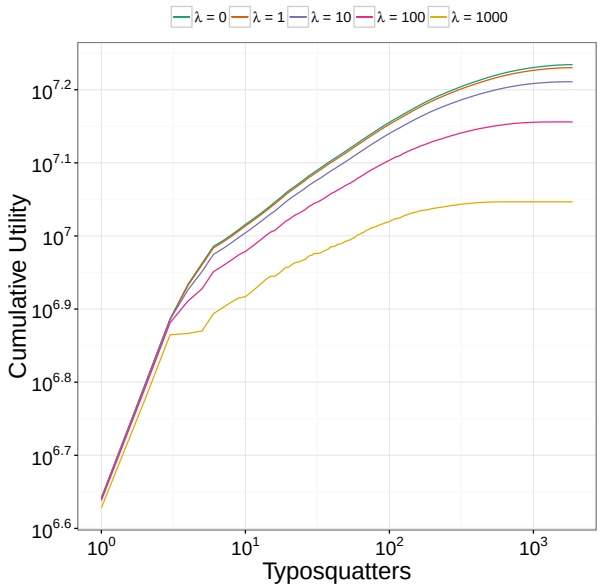


Figure 6.3: This plot shows the cumulative sum of typosquatters' utility for different values of $\lambda_{\theta_{r_c}}$ ($\alpha_{r_c} = 10$ and $\beta_{r_c} = 3$).

affected by changes in $\lambda_{\theta_{r_c}}$. The anti-squatting policy would be an effective complement against typosquatters not affected by the anti-bulk registration policy.

Changing the probability of blacklisting.

We originally assumed a low blacklisting probability. Here we answer the question of how effective registration policies are if there is no blacklisting of domain names $p_{\text{bl}} = 0$ or blacklisting will become much more effective $p_{\text{bl}} \in \{0.1, 0.5\}$. Not surprisingly, if there is no blacklisting then malicious registrants' utility is strictly higher than before. While registration policies are still effective, interestingly, the more effective a policy was in the base scenario the biggest impact $p_{\text{bl}} = 0$ had on increasing miscreants payoff. We observed the opposite effect when $p_{\text{bl}} = 0.5$. We find the synergy between the effectiveness of blacklisting and the effectiveness of registration policy strategies interesting: When a policy proposal was more effective, its effects on abusive domain registrations were disproportionally boosted by the increased performance of blacklisting.

Changing the cost of switching TLDs.

For the base scenario, we assumed that switching TLDs is costly for registrants, but if collaborative registries impose a high registration fee then registrants will switch. We test what happens if the cost of switching is higher or lower. When it does not cost anything to switch TLDs for registrants ($\frac{\gamma_{b,r_c}}{\gamma_{b,r_{nc}}} = 1$), then our results indicate that both benign and malicious registrants will switch TLDs. When $\frac{\gamma_{b,r_c}}{\gamma_{b,r_{nc}}} = 100$ registrants will not switch instead they stop registering domain name altogether when collaborative registries increase their prices.

The ratio of benign registrants.

For the base game, we assumed that there are about a hundred times more benign domains than abusive ones. We found that if the ratio of abusive registrants is higher, non-collaborative registries will be hungrier to gain the business of these malicious registrants and will drop their prices, sacrificing in the process income from regular users.

Registry utility.

As we discussed earlier, if collaborating registries set α_{r_c} high, benign registrants will not register their domain names with them. This leads to an extreme drop in the utility of registries, thus it is unlikely for them to adopt such a strategy. If instead, they increase β_{r_c} their utility also increases slightly because they decrease the utility of malicious registrants, thus they decrease the penalty weighted by $\rho_{r_c} = 0.1$. In summary, if registries are motivated (high enough ρ_{r_c}) then their best response will be to decrease malicious registrations while not hurting benign registrations. For future work, we hope to collect more data on malicious and speculative domain registrations. In the current model, we did not include speculative registrants, while the lack of their registrations is likely to significantly decrease the utility of registries if bulk registrations would become expensive.

6.4 Conclusion

We started with an overview of the domain registration ecosystem focusing on the political and financial dependencies of the most important entities. Building on this understanding, we summarized what decision-makers should consider when designing a domain registration policy. We then discussed the potential of several policy proposals. We found that a) anti-bulk registration, b) incentivizing registries and registrars, and c) anti-squatting were all potentially useful policy proposals. We believe leveraging all three of them together could potentially benefit the domain registration ecosystem the most.

We created a game-theoretical model to analyze the anti-bulk registration policy – using a variable pricing model – in more detail. The best strategies we found to fight online crime, for collaborating registries, are to increase the effectiveness of identity verification and to penalize bulk registrations. However, registries never want to increase their base price considerably because it would lead to a loss of customers. Because of the very strong asymmetry in miscreant success (where only a few miscreants succeed in earning their keep), we discovered that even the most successful domain registration policies would not significantly affect the most successful criminals and thus, may not considerably change the total revenue produced by miscreants. However, they could be particularly useful to remove from the pool the unsuccessful criminals, and drastically decrease abusive domain registrations overall. This result emphasizes the importance of combining registrations policies and to use them together with other lines of defenses.

Chapter 7

Conclusion and Future Directions

7.1 Main Findings

Our measurements have led to several novel findings about typosquatting and malicious domain registrations. We have shown that typosquatting is widespread, where typosquatters also target less popular domain names and register or renew millions of domain names every year. Additionally, we have observed that typosquatting is here to stay as the number of typosquatting domain names steadily increases over time. We have further evaluated the malicious aspects of typosquatting, finding that typosquatters often rely on complex, shared and malicious advertisement networks to profit from user visits. Typosquatters, together with illicit free movie streaming sites and ad-based URL shortening services, expose users to a diverse set of malicious landing pages, including phishing, scam and deception. As typosquatting is not specific to the web, we have also studied the threat of email typosquatting. Our analysis has shown that users send large amounts of emails containing sensitive personal information to typosquatting domains. Additionally, there is infrastructure in place for typosquatters to collect emails from tens of thousands of domain names.

To protect users, we have designed and evaluated potential countermeasures. We have developed an accurate classifier that can find typosquatting domains and can be used as a browser plugin or as a blacklist. Furthermore, we trained another classifier that can precisely identify redirection chains that would lead users to malicious landing pages before they land on those pages. Finally, we have analyzed domain registration policies and concluded that they are crucial in curbing malicious domain registrations.

7.2 Future Directions and Challenges

Building multiple measurement infrastructures to study abusive domain registrations, we expand various aspects of measurements and detection of typosquatting and malicious TDSs, including the study of various protocols, the scale of the data collection, cloaking and user emulation. Even though we can observe advancement in how online crime measurements

are performed, there is still ample space to improve our understanding of these abusive ecosystems.

How and what researchers measure limits the amount and type of malice they can discover from the data collected. Therefore, there is an opportunity for future work to better address the challenges we face.

The first challenge is achieving soundness in active measurements of malicious and dynamic ecosystems. In Chapter 4, we discussed the necessity to study cloaking and user differentiation. However, due to infrastructure limitations, we only visited pages six times to study how phone and desktop users are treated differently and to measure several aspects of cloaking. Thus, our approach was not complete, and it would be useful to study how users are differentiated based on the device vendor, the browser vendor, the browser version and the browsing history, to mention a few. Our approach was also not optimal, as different ecosystems behave differently in which cloaking techniques they use and how they differentiate users, therefore it was not necessary to visit all pages using all six different user profiles. ODIN could be improved by dynamically selecting browser profiles to use when visiting pages based on the likelihood that they will provide us with new knowledge about the page visited.

The second challenge is completeness. There are billions of webpages in hundreds of languages using hundreds of millions of domain names. Thus, considering the first challenge that visiting a page only once is not adequate to understand its behavior, tens of hundreds of billions of page visits are not feasible without vast resources. Therefore, with limited resources, we need to sample the pages carefully we would like to visit. Due to these limitations in Chapter 3, we only studied typosquatting targeting .com domain, and in Chapter 4, we only studied samples from four traffic sources. Future work could explore other traffic sources in relation to the ones we studied and could try different sampling methods to work toward completeness.

Furthermore, there is no universal definition of malice or abuse. Therefore, in our studies, we defined malice as a composite of narrow terms such as "phishing or "malicious download", the main drawback of this approach that it lacks completeness in the definition. For example, we did not study how miscreants could abuse desktop notifications. Additionally, we mainly focused on abuse targeting end-users, and it would be beneficial to also research abuse targeting traffic sources and traffic brokers.

In Chapter 3, while we found that typosquatters most frequently rely on domain parking and traffic distribution networks for profit, at the same time for many typo domains, we did not identify how their owners profit from them. Hence the question is, especially for typosquatting targeting less popular domain names, how registering these domains yields profit to their owners. Alexa's list of top domains is not a perfect estimation of popularity; thus, some less popular domains might receive more traffic in real life. It is also possible that "domainers" bulk register typosquatting domains for a lean profit per domain and that often this strategy might not be successful, or they hope to resell these domains for a margin. Second, Agten et al. [33] have shown that while at any given time only a few percent of typosquatting domains are used for abusive purposes, over a seven-month period, the vast majority of them were used for some malicious activity. This finding suggests that most typosquatting domains are there to cater to malicious use, and they rotate in usage

to avoid blacklisting. Although a subset of typosquatting domain names is registered for phishing to fool users rather than to profit from typing mistakes, our findings in Chapter 3 suggests that this number is low. While many possibilities provide a partial explanation of why these domains are registered, we still lack a complete understanding of the registrants' business model.

Finally, further opportunity for future research includes studying other applications that can be abused by typosquatters and further studying how we could leverage domain registration policies to make malicious registrations less profitable.

Bibliography

[1] 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.

[2] The apache SpamAssassin project. http://spamassassin.apache.org/.

[3] CS data minining challenge's spam dataset. http://csmining.org/index.php/spam-email-datasets-.html. Last accessed Mar 25, 2016.

[4] DenseNet, Keras. https://keras.io/applications/. Last accessed on April 18, 2020.

[5] Dhash python library. https://pypi.org/project/dhash/. Last accessed on April 18, 2020.

[6] Enron email dataset. https://www.cs.cmu.edu/~./enron/. Last accessed on April 18, 2020.

[7] Financial incentives for DNSSEC adoption. https://blog.apnic.net/2017/12/06/dnssec-deployment-remains-low/. Last accessed on April 18 2020.

[8] Google safe browsing update API. https://developers.google.com/safe-browsing/v4/update-api. Last accessed on April 18, 2020.

[9] HIPAA protected health information identifiers (45 CFR 164.14). http://www.ecfr.gov/cgi-bin/text-idx?SID=e58a563f56b8cf8e6511be534d364a64&node=se45.1.164_1514&rgn=div8. Last accessed on April 18, 2020.

[10] ICANN EPDP process. https://gnso.icann.org/sites/default/files/file/field-file-attach/epdp-gtld-registration-data-specs-final-20feb19-en.pdf. Last accessed on April 18, 2020.

[11] ICANN RDAP protocol. https://www.icann.org/rdap. Last accessed on April 18, 2020.

[12] PhishTank. http://www.phishtank.com. Last accessed on May 1, 2013.

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

[14] Python client library for google safe browsing API. https://github.com/afilipovich/gglsbl. Last accessed on April 18, 2020.

[15] Python Whois parsing tool. https://bitbucket.org/richardpenman/pywhois. Last accessed Dec, 2016.

[16] Ruby Whois parsing tool. https://whoisrb.org/. Last accessed Jan, 2017.

[17] Scikit random forest classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Last accessed on April 18, 2020.

[18] Spamhaus DBL. http://www.spamhaus.org/dbl/. Last accessed on May 1, 2013.

[19] SURBL domain blacklist. http://www.surbl.org/lists. Last accessed on May 1, 2013.

[20] Textract. https://textract.readthedocs.io/en/stable/. Last accessed on April 18, 2020.

[21] Trec spam dataset. http://trec.nist.gov/data/spam.html. Last accessed Mar 25, 2016.

[22] Typosquatting package managers. https://incolumitas.com/2016/06/08/typosquatting-package-managers/. Last accessed on April 18 2020.

[23] Untroubled.org spam archive. http://untroubled.org/spam/. Last accessed Mar 25, 2016.

[24] URIBL domain blacklist. http://www.uribl.com/about.shtml. Last accessed on May 1, 2013.

[25] URLFixer for Mozilla Firefox. http://urlfixer.org/. Last accessed on May 1, 2013.

[26] VirusTotal. https://virustotal.com. Last accessed on April 18, 2020.

[27] Zmap: Internet-wide scan data repository. https://scans.io/. Last accessed Nov 08, 2016.

[28] Congressional hearing: Internet domain name fraud - the U.S. government's role in ensuring public access to accuracte Whois data. https://babel.hathitrust.org/cgi/pt?id=mdp.39015090379986;view=1up;seq=1, 2003. Last accessed on May 21, 2018.

[29] Fraudulent online identity sanctions act (FOISA). https://www.congress.gov/bill/108th-congress/house-bill/03754, 2004. Last accessed on May 22, 2018.

[30] Congressional hearing: ICANN and the Whois database: Providing access to protect consumers from phishing. https://www.gpo.gov/fdsys/pkg/CHRG-109hhrg31537/pdf/CHRG-109hhrg31537.pdf, 2006. Last accessed on May 21, 2018.

[31] ICANN's many trips up capitol hill, part 1. https://www.bna.com/icanns-trips-capitol-b17179927466/, 2015. Last accessed on May 21, 2018.

[32] Cybertelecom: Whois Policy Summary. http://www.cybertelecom.org/dns/whois.htm, 2017. Last accessed on May 21, 2018.

[33] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS 2015)*. Internet Society, 2015.

[34] M. Almishari and X. Yang. Ads-portal domains: Identification and measurements. *ACM Transactions on the Web (TWEB)*, 4(2):1–34, 2010.

[35] S. A. Alrwais, K. Yuan, E. Alowaisheq, Z. Li, and X. Wang. Understanding the dark side of domain parking. In *Proceedings of the USENIX Security Symposium*, pages 207–222, 2014.

[36] R. Anderson, C. Barton, R. Böhme, R. Clayton, M. J. Van Eeten, M. Levi, T. Moore, and S. Savage. Measuring the cost of cybercrime. In *The economics of information security and privacy*, pages 265–300. Springer, 2013.

[37] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a dynamic reputation system for DNS. In *Proceedings of the USENIX security symposium*, pages 273–290, 2010.

[38] Avast. Misspelling goes criminal with typosquatting. `https://blog.avast.com/2012/03/23/misspelling-goes-criminal-with-typosquatting/`, Mar 23 2012. Last accessed on Apr 15, 2013.

[39] A. Banerjee, D. Barman, M. Faloutsos, and L. N. Bhuyan. Cyber-fraud is one typo away. In *Proceedings of the 27th Conference on Computer Communications (INFOCOM)*, pages 1939–1947. IEEE, 2008.

[40] A. Banerjee, M. S. Rahman, and M. Faloutsos. SUT: Quantifying and mitigating URL typosquatting. *Computer Networks*, 55(13):3001–3014, 2011.

[41] E. Bursztein, A. Malyshev, T. Pietraszek, and K. Thomas. Picasso: Lightweight device class fingerprinting for web clients. In *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 93–102. ACM, 2016.

[42] J. Caballero, C. Grier, C. Kreibich, and V. Paxson. Measuring pay-per-install: the commoditization of malware distribution. In *Proceedings of the 20th USENIX Conference on Security*, 2011.

[43] N. Chachra, D. McCoy, S. Savage, and G. M. Voelker. Empirically characterizing domain abuse and the revenue impact of blacklisting. In *Proceedings of the Workshop on the Economics of Information Security (WEIS)*, 2014.

[44] G. Chen, M. F. Johnson, P. R. Marupally, N. K. Singireddy, X. Yin, and V. Paruchuri. Combating typo-squatting for safer browsing. In *Proceedings of the Advanced Information Networking and Applications Workshops (WAINA)*, pages 31–36. IEEE, 2009.

[45] N. Christin. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd World Wide Web Conference (WWW'13)*, pages 213–224, Rio de Janeiro, Brazil, May 2013.

[46] J. W. Clark and D. McCoy. There are no free ipads: An analysis of survey scams as a business. In *Proceedings of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2013.

[47] R. Clayton and T. Mansfield. A study of Whois privacy and proxy service abuse. In *Proceedings of the 13th Workshop on the Economics of Information Security*, 2014.

[48] S. E. Coull, A. M. White, T.-F. Yen, F. Monrose, and M. K. Reiter. Understanding domain registration abuses. *Computers & security*, 2012.

[49] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176, 1964.

[50] D. Danchev. Legitimate software typosquatted in SMS micro-payment scam. blog, `http://ddanchev.blogspot.com/2009/07/legitimate-software-typosquatted-in-sms.html`, Jul 7 2009. Last accessed on Apr 15, 2013.

[51] B. Edelman. Large-scale registration of domains with typographical errors. `http://cyber.law.harvard.edu/people/edelman/typo-domains/`, Sep 2003. Last accessed on Apr 15, 2013.

[52] B. Edelman. Estimating visitors and advertising costs of typo domains. `http://www.benedelman.org/typosquatting/pop.html`, 2010. Last accessed on Jan 16, 2016.

[53] M. Felegyhazi, C. Kreibich, and V. Paxson. On the potential of proactive domain blacklisting. In *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*. USENIX Association, 2010.

[54] P. Festa. Domain squatters losing out. `https://www.cnet.com/news/domain-squatters-losing-out/`, 1998. Last accessed on May 24, 2018.

[55] Godai group. Doppelganger domains. `http://godaigroup.net/wp-content/uploads/doppelganger/Doppelganger.Domains.pdf`, Sept 6 2011. Last accessed on Feb 12, 2017.

[56] Google. Efficient Frontier's automotive clients receive twice the conversion rate as search with domain ads. `http://www.google.com/adwords/casestudies/EfficientFrontierAFDCaseStudy.pdf`, 2010. Last accessed on Feb 12, 2018.

[57] J. V. Grove. Facebook wins millions in case against typo squatters. `http://www.cnet.com/news/facebook-wins-millions-in-case-against-typo-squatters/`, 2013.

[58] T. Halvorson, M. F. Der, I. Foster, S. Savage, L. K. Saul, and G. M. Voelker. From. academy to. zone: An analysis of the new TLD land rush. In *Proceedings of the 2015 Internet Measurement Conference*, pages 381–394. ACM, 2015.

[59] T. Halvorson, J. Szurdi, G. Maier, M. Felegyhazi, C. Kreibich, N. Weaver, K. Levchenko, and V. Paxson. The BIZ top-level domain: ten years later. In *Proceedings of the International Conference on Passive and Active Network Measurement*, pages 221–230. Springer, 2012.

[60] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster. Predator: proactive recognition and elimination of domain abuse at time-of-registration. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1568–1579. ACM, 2016.

[61] C. Herley and D. Florêncio. Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. In *Proceedings (online) of the Worskshop on Economics of Information Security*, June 2009. Available from `http://weis09.infosecon.net/`.

[62] A. Hidayat. PhantomJS. `http://phantomjs.org/`, 2013.

[63] D. Ibosiola, B. Steer, A. Garcia-Recuero, G. Stringhini, S. Uhlig, and G. Tyson. Movie pirates of the caribbean: Exploring illegal streaming cyberlockers. In *Proceedings of the Twelfth International AAAI Conference on Web and Social Media*, 2018.

[64] ICANN. Uniform domain name dispute resolution policy (UDRP). `http://www.icann.org/en/help/dndr/udrp`, 1999. Last accessed on Apr 15, 2013.

[65] ICANN. Operational point of contact final report. `https://gnso.icann.org/sites/default/files/filefield_6454/icann-whois-wg-report-final-1-9.pdf`, 2007. Last accessed on May 21, 2018.

[66] ICANN. The end of domain tasting - status report on AGP measures. `http://www.icann.org/en/resources/registries/agp/agp-status-report-12aug09-en.htm`, Aug 12 2009. Last accessed on Apr 15, 2013.

[67] ICANN. SAC055, WHOIS: Blind men and an elephant. `https://www.icann.org/en/system/files/files/sac-055-en.pdf`, 2012. Last accessed on May 21, 2018.

[68] ICANN. 2013 registrar accreditation agreement. `https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en#whois-accuracy`, 2013. Last accessed on May 21, 2018.

[69] ICANN. SAC058, SSAC report on domain name registration data validation. `https://www.icann.org/en/system/files/files/sac-058-en.pdf`, 2013. Last accessed on May 21, 2018.

[70] ICANN. Final issue report on a next-generation gTLD registration directory service (RDS) to replace WHOIS. `https://whois.icann.org/sites/default/files/files/final-issue-report-next-generation-rds-07oct15-en.pdf`, 2015. Last accessed on May 21, 2018.

[71] ICANN. Domain abuse activity project report ICANN 60. `https://www.icann.org/en/system/files/files/presentation-daar-31oct17-en.pdf`, 2017. Last accessed on Feb 18, 2018.

[72] ICANN. ICANN audited fianncial statement. `https://www.icann.org/en/system/files/files/financial-report-fye-30jun17-en.pdf`, 2017. Last accessed on Feb 16, 2018.

[73] ICANN. ICANN mission statement. `https://www.icann.org/resources/pages/governance/bylaws-en/#article1`, 2017. Last accessed on May 25, 2018.

[74] ICANN. ICANN registry agreements. `https://www.icann.org/resources/pages/registries/registries-agreements-en`, 2017. Last accessed on Feb 15, 2018.

[75] ICANN. Whois primer. `https://whois.icann.org/en/primer`, 2017. Last accessed on May 21, 2018.

[76] ICANN. Registration directory services review fact sheet. `https://community.icann.org/display/WHO/Fact+Sheet`, 2018. Last accessed on May 21, 2018.

[77] L. Invernizzi, S. Miskovic, R. Torres, C. Kruegel, S. Saha, G. Vigna, S.-J. Lee, and M. Mellia. Nazca: Detecting malware distribution in large-scale networks. In

Proceedings of the 18th Annual Network and Distributed Systems Security (NDSS 2014) Symposium, volume 14, pages 23–26. Citeseer, 2014.

[78] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein. Cloak of visibility: Detecting when machines browse a different web. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pages 743–758. IEEE, 2016.

[79] J. Jueckstock, S. Sarker, P. Snyder, P. Papadopoulos, M. Varvello, B. Livshits, and A. Kapravelos. The blind men and the internet: Multi-vantage point web measurements. *arXiv preprint arXiv:1905.08767*, 2019.

[80] M. T. Khan, X. Huo, Z. Li, and C. Kanich. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, pages 135–150. IEEE, 2015.

[81] A. Kharraz, W. Robertson, and E. Kirda. Surveylance: automatically detecting online survey scams. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pages 70–86. IEEE, 2018.

[82] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[83] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 569–586. ACM, 2017.

[84] J. Klensin. Simple mail transfer protocol, Oct. 2008. IETF RFC 5321.

[85] M. Korczynski, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. van Eeten. Reputation metrics design to improve intermediary incentives for security of TLDs. In *Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 579–594. IEEE, 2017.

[86] M. Korczynski, M. Wullink, S. Tajalizadehkhoob, G. C. Moura, and C. Hesselman. Statistical analysis of DNS abuse in gTLDs final report. 2017.

[87] M. Korczynski, M. Wullink, S. Tajalizadehkhoob, G. C. Moura, A. Noroozian, D. Bagley, and C. Hesselman. Cybercrime after the sunrise: A statistical analysis of DNS abuse in new gTLDs. In *Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security*. ACM, 2018.

[88] A. Le, A. Markopoulou, and M. Faloutsos. Phishdef: URL names say it all. In *Proceedings of the 2011 IEEE INFOCOM*, pages 191–195. IEEE, 2011.

[89] N. Leontiadis and N. Christin. Empirically measuring WHOIS misuse. In *Proceedings of the European Symposium on Research in Computer Security*, pages 19–36. Springer, 2014.

[90] N. Leontiadis, T. Moore, and N. Christin. Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade. In *Proceedings of the USENIX Security Symposium*, volume 11, 2011.

[91] N. Leontiadis, T. Moore, and N. Christin. Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade. In *Proceedings of USENIX Security 2011*, San Francisco, CA, Aug. 2011.

[92] N. Leontiadis, T. Moore, and N. Christin. A nearly four-year longitudinal study of search-engine poisoning. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 930–941. ACM, 2014.

[93] 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. Click trajectories: End-to-end analysis of the spam value chain. In *Proceedings of IEEE Security and Privacy*, Oakland, CA, May 2011.

[94] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Felegyhazi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, et al. Click trajectories: End-to-end analysis of the spam value chain. In *Proceedings of the IEEE Symposium on Security and Privacy, 2011*, pages 431–446. IEEE, 2011.

[95] 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 (SP)*, pages 112–126. IEEE, 2013.

[96] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang. Knowing your enemy: understanding and detecting malicious web advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 674–686. ACM, 2012.

[97] H. Liu, K. Levchenko, M. Felegyhazi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage. On the effects of registrar-level intervention. In *Proceedings of the 4th USENIX LEET*, 2011.

[98] L. Lu, R. Perdisci, and W. Lee. SURF: Detecting and measuring search poisoning. In *Proceedings of ACM CCS 2011*, Chicago, IL, Oct. 2011.

[99] K. McCarthy. World's most notorious cybersquatter arrested. `https://www.theregister.co.uk/2003/09/04/worlds_most_notorious_cybersquatter_arrested/`, 2003. Last accessed on May 24, 2018.

[100] D. McCoy, A. Pitsillidis, G. Jordan, N. Weaver, C. Kreibich, B. Krebs, G. Voelker, S. Savage, and K. Levchenko. Pharmaleaks: Understanding the business of online pharmaceutical affiliate programs. In *Proceedings of USENIX Security 2012*, Bellevue, WA, Aug. 2012.

[101] J. D. Mercer. Cybersquatting: Blackmail on the information superhighway. *BUJ Sci. & Tech. L.*, 6, 2000.

[102] X. Mi, Y. Liu, X. Feng, X. Liao, B. Liu, X. Wang, F. Qian, Z. Li, S. Alrwais, and L. Sun. Resident evil: Understanding residential IP proxy as a dark service. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, pages 1185–1201. IEEE, 2019.

[103] Microsoft Technet. The trouble with typosquatting. http://blogs.technet.com/b/microsoft_on_the_issues/archive/2010/04/15/the-trouble-with-typosquatting.aspx, Apr 15 2010. Last accessed on Apr 15, 2013.

[104] N. Miramirkhani, O. Starov, and N. Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. In *Proceedings of the 24th Network and Distributed System Security Symposium (NDSS 2017). Internet Society*, 2017.

[105] T. Moore, R. Clayton, and R. Anderson. The economics of online crime. *Journal of Economic Perspectives*, 23(3):3–20, 2009.

[106] T. Moore and B. Edelman. Measuring the perpetrators and funders of typosquatting. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, pages 175–191. Springer, 2010.

[107] T. Moore and B. Edelman. Online appendix for measuring the perpetrators and funders of typosquatting. http://www.benedelman.org/typosquatting/pop.html, 2010. Last accessed on Feb 12, 2018.

[108] T. Nelms, R. Perdisci, M. Antonakakis, and M. Ahamad. Towards measuring and mitigating social engineering software download attacks. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, pages 773–789, 2016.

[109] N. Nikiforakis, M. Balduzzi, L. Desmet, F. Piessens, and W. Joosen. Soundsquatting: Uncovering the use of homophones in domain squatting. In *Proceedings of the International Conference on Information Security*, pages 291–308. Springer, 2014.

[110] 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*, pages 51–62. ACM, 2014.

[111] OpenDNS. There's no "i" in twtter: How to outsmart typosquatting. http://blog.opendns.com/2011/09/02/there%E2%80%99s-no-%E2%80%9Ci%E2%80%9D-in-twtter-how-to-outsmart-typosquatting/, Sep 2 2011. Last accessed on Apr 15, 2013.

[112] P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, and G. Voelker. Characterizing large-scale click fraud in zeroaccess. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 141–152, 2014.

[113] A. Pitsillidis, C. Kanich, G. M. Voelker, K. Levchenko, and S. Savage. Taster's choice: a comparative analysis of spam feeds. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 427–440. ACM, 2012.

[114] M. Z. Rafique, T. Van Goethem, W. Joosen, C. Huygens, and N. Nikiforakis. It's free for a reason: Exploring the ecosystem of free live streaming services. In *Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS 2016)*, pages 1–15. Internet Society, 2016.

[115] P. J. Rousseeuw and M. Hubert. Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):73–79, 2011.

[116] M. Sharif, J. Urakawa, N. Christin, A. Kubota, and A. Yamada. Predicting impending exposure to malicious content from user behavior. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1487–1501. ACM, 2018.

[117] C. Soghoian, O. Friedrichs, and M. Jakobsson. The threat of political phishing. In *Proceedings of the International Symposium on Human Aspects of Information Security & Assurance*. Citeseer, 2008.

[118] Sophos, Naked Security. Typosquatting - what happens when you mistype a website name? `http://nakedsecurity.sophos.com/typosquatting/`, Dec 14 2011. Last accessed on Apr 15, 2013.

[119] K. Soska and N. Christin. Automatically detecting vulnerable websites before they turn malicious. In *Proceedings of the USENIX Security Symposium*, pages 625–640, 2014.

[120] K. Soska and N. Christin. Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security'14)*, pages 33–48, Washington, DC, Aug. 2015.

[121] B. Srinivasan, A. Kountouras, N. Miramirkhani, M. Alam, N. Nikiforakis, M. Antonakakis, and M. Ahamad. Exposing search and advertisement abuse tactics and infrastructure of technical support scammers. In *Proceedings of the 2018 World Wide Web Conference*, pages 319–328. International World Wide Web Conferences Steering Committee, 2018.

[122] O. Starov, Y. Zhou, X. Zhang, N. Miramirkhani, and N. Nikiforakis. Betrayed by your dashboard: Discovering malicious campaigns via web analytics. In *Proceedings of the 2018 World Wide Web Conference*, pages 227–236. International World Wide Web Conferences Steering Committee, 2018.

[123] B. Stone-Gross, R. Abman, R. A. Kemmerer, C. Kruegel, D. G. Steigerwald, and G. Vigna. The underground economy of fake antivirus software. In *Economics of information security and privacy III*, pages 55–78. Springer, 2013.

[124] G. Stringhini, C. Kruegel, and G. Vigna. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 133–144. ACM, 2013.

[125] S. D. Sunderland. Domain name speculation: Are we playing whac-a-mole. *Berkeley Tech. LJ*, 25:465–492, 2010.

[126] J. Szurdi and N. Christin. Email typosquatting. In *Proceedings of the 2017 Internet Measurement Conference*, pages 419–431. ACM, 2017.

[127] J. Szurdi and N. Christin. Domain registration policy strategies and the fight against online crime. In *Proceedings (online) of the Fourteenth Workshop on the Economics of Information Security (WEIS)*, Innsbruck, Austria, June 2018.

[128] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich. The long "taile" of typosquatting domain names. In *Proceedings of the USENIX Security Symposium*, pages 191–206, 2014.

[129] Techcrunch.com. U.S. court rules for Facebook in its case against typosquatters on 105 domains; $2.8m in damages. `http://techcrunch.com/2013/05/01/u-s-court-rules-for-facebook-in-its-case-against-typosquatters-on-105-domains-2-8m-in-damages/`, May 1 2013. Last accessed on May 1, 2013.

[130] The Next Web. Typosquatting sites 'wikapedia' and 'twtter' have been fined $300,000 by UK watchdog. `http://thenextweb.com/insider/2012/02/16/typosquatting-sites-wikapedia-and-twtter-have-been-fined-300000-by-uk-watchdog/`, Feb 16 2012. Last accessed on Apr 15, 2013.

[131] The Register. Typosquatters set up booby-trapped High Street names. `http://www.channelregister.co.uk/2011/12/13/typosquatting_scams_target_xmas_shoppers/`, Dec 13 2011. Last accessed on Apr 24, 2013.

[132] K. Thomas, D. Y. Huang, D. W. E. B. C. GrierD, T. J. Holt, C. Kruegel, D. McCoy, S. Savage, and G. Vigna. Framing dependencies introduced by underground commoditization. In *Proceedings of the Workshop on the Economics of Information Security*, 2015.

[133] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang. Needle in a haystack: tracking down elite phishing domains in the wild. In *Proceedings of the Internet Measurement Conference 2018*, pages 429–442. ACM, 2018.

[134] P. Vadrevu and R. Perdisci. What you see is not what you get: Discovering and tracking social engineering attack campaigns. In *Proceedings of the Internet Measurement Conference*, pages 308–321. ACM, 2019.

[135] T. Vidas and N. Christin. Evading Android runtime analysis via sandbox detection. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS'14)*, Kyoto, Japan, June 2014.

[136] T. Vissers, T. Barron, T. Van Goethem, W. Joosen, and N. Nikiforakis. The wolf of name street: Hijacking domains through their nameservers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 957–970. ACM, 2017.

[137] T. Vissers, W. Joosen, and N. Nikiforakis. Parking sensors: Analyzing and detecting parked domains. In *Proceedings of the Network and Distributed Security Symposium*, 2015.

[138] P. Vixie. Taking back the DNS. `http://www.isc.org/community/blog/201007/taking-back-dns-0`, Jul 29 2010. Last accessed on Apr 15, 2013.

[139] D. Wang, S. Savage, and G. Voelker. Cloak and dagger: Dynamics of web search cloaking. In *Proceedings of ACM CCS 2011*, Chicago, IL, Oct. 2011.

[140] 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 Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, 2006.

[141] N. Weaver, C. Kreibich, and V. Paxson. Redirecting DNS for ads and profit. In *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet (FOCI), San Francisco, CA, USA (August 2011)*, 2011.

[142] Websense Security Labs. The rise of a typosquatting army. `http://community.websense.com/blogs/securitylabs/archive/2012/01/22/The-rise-of-a-typosquatting-army.aspx`, Jan 22 2012. Last accessed on Apr 15, 2013.

[143] Wikipedia. List of most expensive domain names. `https://en.wikipedia.org/wiki/List_of_most_expensive_domain_names`, 2018. Last accessed on Feb 18, 2018.

[144] Wikipedia. Tokelau. `https://en.wikipedia.org/wiki/Tokelau`, 2018. Last accessed on Feb 20, 2018.

[145] Y. Xu, T. Price, J.-M. Frahm, and F. Monrose. Virtual u: Defeating face liveness detection by building virtual models from your public photos. In *Proceedings of the USENIX security symposium*, pages 497–512, 2016.

[146] 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*, pages 373–380. ACM, 2014.