UNIVERSITY OF CALIFORNIA SAN DIEGO

Countering financially-motivated malicious actors on the Internet

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Michael Joseph DeBlasio

Committee in charge:

Professor Alex C. Snoeren, Chair
Professor George Papen
Professor George Porter
Professor Stefan Savage
Professor Geoffrey M. Voelker

2018

The Dissertation of Michael Joseph DeBlasio is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2018

DEDICATION

This dissertation is dedicated to those who think that you have to be “really smart” to get a PhD.

You don’t—you just have to be really stubborn.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I am truly grateful to my advisor, Alex C. Snoeren. He has supported me throughout the most critical years of my graduate career with remarkable patience and quiet guidance. I am equally grateful to Geoffrey M. Voelker, who was a co-advisor in every way but on paper, and a consistent ear when I needed to talk. The best parts of this dissertation reflect Alex and Geoff's influence and support. The two of them took a chance on me when I changed research directions many years ago, and I am indebted to them both.

I am also grateful to Stefan Savage. Among many things, Stefan has opened enumerable doors for me over the last seven years, helping me build some of the most important connections in my professional life. I have learned how to navigate people and bureaucracy from him.

All three of these individuals are not just great researchers and leaders, but good people. They have constructed an environment where everyone feels welcome, and are supportive even when students want to sink substantial time into projects other than research. I have spent a lot of time working to build community in this department, and they have supported me in doing so throughout.

Chapter 3, among other projects, was only possible because of Saikat Guha. Saikat has provided me with incredible opportunities to access real data and make real impact, and has shown me what effective industry research can look like. He has a remarkable ability to *make things happen*, and I have no doubt I will work with him again in the future.

UCSD's Systems and Networking group has amazing staff without whom none of our work would be possible. Cindy Moore, Brian Kantor and Jennifer Folkestad all went out of their way to make countless problems disappear, allowing me to focus on my research. I am thankful to them all.

A key component to happiness in graduate school is surrounding yourself with good people, and I have had the privilege of working with exactly the right type of coworkers and friends over the years. They are amazing individuals who have acted as sounding boards for many bad ideas and offered sage advice from their own experiences. Among many others, I

would not be here without support from Ariana Mirian, Danny Huang, David Kohlbrenner, Dustin Richmond, Keaton Mowery, Louis DeKoven, Neha Chachra, and Tristan Halvorson.

I am also thankful for the cheerleading and love from my family throughout my graduate career. The process of a STEM PhD was new to them, and thus came with its own series of anxieties, but their support never wavered. My family set me up for success, and I would not be where I am without those innumerable efforts.

Finally, I am indebted to my partner Katie Meyers. Graduate school brought many challenges, and Katie has been a constant support throughout. She has been more patient than I could have asked as our plans slipped from five years in graduate school, to six, and then to seven. She cheerfully put up with years of me coming home and collapsing into a puddle of exhaustion, and her company has kept me going like nothing else. I can’t imagine how I would have made it through without her.

My work was supported in part by National Science Foundation grants 1237264 and 1629973, the Irwin Mark and Joan Klein Jacobs Chair in Information and Computer Science, and by generous research, operational and/or in-kind support via the UCSD Center for Networked Systems (CNS).

Chapter 2, in full, is a reprint of the material as it appears in the Proceedings of the ACM Internet Measurement Conference, 2017. DeBlasio, Joe; Savage, Stefan; Voelker, Geoffrey M.; Snoeren, Alex C.. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in Proceedings of the ACM Internet Measurement Conference, 2017. DeBlasio, Joe; Guha, Saikat; Voelker, Geoffrey M.; Snoeren, Alex C., The dissertation author was the primary investigator and author of this paper.

Chapter 4 is work that has been accepted for publication in Proceedings of the ACM Internet Measurement Conference, 2018. Khan, Mohammad Taha; DeBlasio, Joe; Voelker, Geoffrey M.; Snoeren, Alex C.; Kanich, Chris; Vallina-Rodriguez, Narseo. The dissertation author was a primary investigator and author of this paper.

VITA

2011	Bachelor of Science, Harvey Mudd College
2013	Master of Science, University of California San Diego
2013	Teaching Assistant, Department of Computer Science & Engineering University of California San Diego
2015	Teaching Assistant, Department of Computer Science & Engineering University of California San Diego
2017	Instructor, Department of Computer Science & Engineering University of California San Diego
2018	Doctor of Philosophy, University of California San Diego

ABSTRACT OF THE DISSERTATION

Countering financially-motivated malicious actors on the Internet

by

Michael Joseph DeBlasio

Doctor of Philosophy in Computer Science

University of California San Diego, 2018

Professor Alex C. Snoeren, Chair

Fraud, theft and other abuses are unfortunate realities of the modern Internet. While defenders work to stop these bad things from happening, attackers are constantly evolving to try to stay one step ahead. Experience shows that not all attacks can be prevented, so defenders must also work to detect ongoing attacks quickly to contain the inevitable damage.

As attacks evolve, detecting them requires new methods. A challenge remains on how to efficiently develop these new methods across the diverse threat landscape given that attack methods vary widely, and most current systems to detect attacks are deeply domain-specific and hard to generalize. Despite first appearances, however, many attacks do have something important in common: their underlying motivations.

These common motivations can help guide the development of new detection and mitigation methods, regardless of how the attack is ultimately carried out. For financially-driven malicious actors in particular, there is an inherent tension in evading detection: the better they maintain their illusions, the more effectively they can monetize their attack, but good evasion is expensive and cuts into the attacker's profit. As such, attackers construct illusions that are good enough only to counter expected defenses, rather than blend in perfectly. Understanding these incentives and constraints guides us not only to identify what attackers may target, but also where they are most likely to leave themselves unprotected.

In this dissertation, I demonstrate the value and applicability of this approach in three vastly different environments. I use it to detect web site compromise at scale by leveraging the incentive to perform password re-use attacks, to mitigate fraud on search engine advertising by eliminating verticals where fraudsters are forced to congregate, and to identify deceptive commercial VPN providers by identifying when providers strategically deceive users in what vantage points they offer. In each case, understanding where malicious actors would not or could not effectively mask their activity yields concrete techniques to detect and mitigate their malicious activity.

Chapter 1

Introduction

Theft, fraud and other crime are timeless problems, and while the methods people use to interact have changed, this fundamental truth still holds on the Internet today. Defenders constantly work to stop malicious actors, and so those actors necessarily change their methods to stay one step ahead. Defending against attacks in this ecosystem is hard—while defenses are essential in preventing and slowing attacks, experience shows that attacks can not be stamped out entirely, and indeed successful attacks are still commonplace. Defenders must then do their best to contain the inevitable damage.

While not all malicious activity fits this model, many malicious actors aim to remain completely under the radar during and after their attacks. Fraudsters abusing a service—like Twitter trolls or deceptive advertisers—are able to continue their abuse with impunity as long as they are not kicked off of the platform. Many types of stolen data—such as credit cards—lose their value once victims know that their information is exposed. Malicious actors with a public presence—like a purported news organization—lose much of their influence once their malicious activity is identified publicly. Detection reduces the value the malicious actors can extract from their malicious activity, and correspondingly, reduces the damage to the victims.

To enable early detection of malicious activity, defenders must identify small tells that belie the attacker's true intentions from their earliest actions. Significant work has gone into developing tools that aim to provide this early detection [31, 100], but while these systems help, they are often deeply domain-specific. It is difficult to generalize detection methods because mechanisms of attack vary so widely, and different attacks seem to lack commonalities in how they work. For instance, techniques to detect unexpected activity on a corporate network appear to have little in common with techniques to detect fraudulent activity on stolen credit cards, which traditionally share little in common with methods to identify fake accounts on a web service.

Despite appearances, these attacks do have commonalities; rather than sharing similar mechanisms, however, many attacks share similar underlying motives. In this dissertation, I advocate for a strategy to guide development of methods to detect and prevent fraud, abuse and other attacks based on an understanding of the common attacker motives. By focusing on these commonalities, strategies can apply widely across any malicious activities that share the same fundamental motives, and can remain usable regardless of who the malicious actor is.

In this dissertation, I show that identifying and understanding the constraints under which malicious actors operate can guide defenders in identifying leads on where attackers would not—or cannot—invest enough time or money to evade detection. No single approach can detect or prevent all fraud, but this approach can help provide meaningful tools to mitigate significant previously-undetected attacks across a wide range of environments.

This dissertation focuses on the subset of malicious actors that are driven by financial incentives—that is, those that want to make money. These actors are unified by their need not just to succeed in their malicious activity, but to do so profitably: the revenue gained from monetizing the activity must exceed all associated costs with it, including that of staying undetected.

Efforts to stay undetected can take many forms. Malicious actors may take additional action to act like authentic users and thus build up their disguise, or they may artificially slow or restrict their behavior to avoid suspicion. Malicious actors may also need to cover their tracks by deleting log files or otherwise destroying evidence. Finally, malicious actors must be cautious in how they implicitly announce their attacks, for instance in limiting how they sell the fruits of their labor. These methods collectively construct illusions of legitimacy to aid in staying under the radar.

There is an inherent tension for malicious actors in constructing and maintaining these illusions: doing so can be costly, and cut into their profit. As such, like magicians, attackers must instead ensure their illusions look perfect from the audience's perspective, but need not ensure that the illusions work well from other angles. Perfecting illusions from all angles may be difficult, or even impossible.

Put succinctly, while each ecosystem is different, financially driven malicious actors wishing to remain undetected operate under the mixed incentives of maintaining the illusion of legitimacy and maximizing profit of the attack. In this dissertation, I show that these trade-offs lead to imperfect illusions that can be systematically exploited to detect or mitigate the malicious behavior. I demonstrate this approach's feasibility by showing that it works across a wide variety of scenarios: when the adversary is an external attacker stealing information from a service, when the adversary is a fraudulent user of a service, and even when the adversary is a fraudulent service itself.

1.1 Understanding attacker motivation

While malicious activity can take a variety of forms, many malicious actors are driven by a relatively small number of motivations. A few common motivations are worth briefly exploring on their own to provide context on why this dissertation focuses on those wanting to make money.

Some malicious actors are driven by external, state-sponsored motivators like espionage or cyberwarfare to meet strategic national objectives. In 2016, for example, attackers affiliated with the Russian government targeted several US political organizations including the Democratic National Committee and presidential campaign of Hillary Clinton [81]. In other instances, North Korean hackers reportedly tried to steal billions of dollars from the New York Federal Reserve, and may have successfully stolen at least $81 million [85, 84]. Detecting these actors can be challenging because they may bring substantial financial and technical resources to bear to evade defenders, and may lack the constraints that other attackers may work under.

Other malicious actors are driven by fame, revenge, politics, or fun. As an example, Anonymous, a group of vigilante hackers, is notorious for defacing websites of their enemies, from ISIS to members of Congress [82, 128, 64]. In several other recent incidents, online gamers will spoof phone calls to 911-operators, claiming that a violent situation requiring a SWAT team is in progress at their victim's location. This form of online harassment to punish the enemies of the attackers can even result in accidental death [129]. Across many of these attacks, detection is an expected outcome, and may indeed be the point. While these actors also have constraints under which they operate, those constraints are not as clear cut as our next group of actors.

Finally, many malicious actors on the Internet, and the focus of this dissertation, are financially driven. These malicious actors want to make money, and attacking users or services serves that end. Examples of these actors are easy to find—for example, most Internet users are familiar with email spam attempting to sell pharmaceuticals. In another example, Dropbox experienced a breach of 68 million sets of user credentials in 2012 which later turned up for sale on underground markets [127]. A common form of malware is ransomware, which encrypts all files on a computer before demanding payment to decrypt them [22]. Services exist online to perform distributed denial of service attacks on the target of your choice for a fee [21]. Across these cases, the attackers' goal is to make money, and ultimately if the actors are unable to do so, they will go elsewhere [67]. These constraints provide an opportunity for defenders to get insight into how attackers think about a problem, which in turn dictates how the attackers will act.

It can at first glance be challenging to understand how an attacker can monetize the fruits of all attacks. When credit card information is stolen, making money may appear straightforward, but what about when, for instance, an attacker generates hundreds of accounts on a service? In these cases, stolen resources may be used to facilitate additional attacks. The modern cybercriminal ecosystem streamlines this process by commodifying the resources used in cyber attacks. In so doing, an attacker can specialize and focus on what they are good at (e.g. phishing, or attacking a particular system or a service), and rely on the secondary market to turn the stolen resource into cash [43, 77, 78]. The attackers in the Dropbox breach, for instance, did not have to use the stolen credentials themselves, but could instead directly sell those usernames and passwords to someone else. In another recent example, an attacker stole plans for a military drone and put them up for sale on underground markets [19], which allowed the attacker to focus on the initial attack, rather than knowing how to use the drone plans themselves.

Understanding these motivations of financially driven attackers can lead to insights into how to counter malicious activity, but it is still important to understand the technical details of how a malicious actor does what they do. In later chapters of this dissertation, I explore several specific problem spaces in detail and discuss the mechanisms that malicious actors use in each case.

1.2 Context

This dissertation is far from the first work to attempt to detect attacks as they occur, nor is it the first to consider attacker incentives for financial constraints. Understanding how this approach differs from past work helps to contextualize why this perspective is unique. Much of the preceding work focuses on particular problem domains and builds systems to address those needs, for instance in network intrusion detection systems [31, 100]. The technique described in this dissertation is general, and can be more widely applied.

Much work has been done in detection and mitigation specifically focusing on the cybercrime ecosystem. For instance, Levchenko *et al.* [61] present a thorough analysis of the entire "spam value chain" to identify what interventions might be most effective in combating spam. They identify bottlenecks wherein some 95% of spammers used just a handful of banks to process their payments, and that switching banks was an expensive and time-consuming process for spammers. In so doing, their work suggests that aggressively pursuing cleaning up these banks could uniquely hinder the profitability of the entire spam ecosystem. This dissertation aims to continue this style of financially-driven analysis and generalize it into a strategy for identifying and combating malicious behavior in the general case. Several other efforts [29, 106, 108] have investigated other aspects and angles of the underground ecosystems themselves.

In another work, Thomas *et al.* [109] analyzed the underground market for Twitter accounts by targeted purchasing of thousands of accounts for sale online. These accounts were then subsequently used to create a classifier to detect all such fraudulently created accounts on the platform. This paper fits quite well within the work of this dissertation in that those malicious actors chose to advertise their product widely (in hopes of increasing sales volume), and in so doing, facilitated Twitter's own detection and mitigation mechanisms.

1.3 A widely applicable approach

An advantage of the approach described throughout this dissertation is not just that it applies across types of attacks, but also that it guides defenders no matter from where an attacker attacks or where a defender tries to counter that attack. While there are exceptions, for our purposes it is useful to think of the Internet as being a series of discrete services in which users interact directly with each service. Interaction between users is indirect, and only as facilitated by the service. In a social network, for instance, users log on to a web service, posting content via that service's website in order to interact with other users. In this model, attackers and defenders can fulfill any of the roles of user, service, or third-party. In each of the following three chapters of this dissertation, I explore a distinct problem area where the attackers and defenders assume varying roles. In each case, I demonstrate the value of the approach as a whole, and identify instances where the attacker's carefully-crafted illusions break down, and where defenders can leverage the differences to counter the malicious actors.

In Chapter 2, I describe a system I developed called Tripwire that allows identification of when web services have their user databases exposed to a third party. This approach puts few constraints on how attackers compromise a site, and can be used by the service provider, users or even as a third party. This method scales well to a large number of services, has no false positives, and requires no coordination with the services under test. I do this by recognizing the financial incentive for the attacker to perform password reuse attacks. By setting up our own honeyaccounts, we detect password reuse attacks and infer when a site is breached. I describe my proof of concept implementation that detects several unknown compromises on sites large and small.

Chapter 3 describes an investigation on fraud data of search advertisements on Bing Search from the perspective of the service itself. Advertisers on the platform are users of Bing's advertising service, creating ads that run alongside results on Bing web search. Fraudulent advertisers use the prominent placement of their ads to lend an air of legitimacy, and to promote traffic to deceptive or fraudulent sites. While these advertisers spend significant effort to blend in with legitimate advertisers (e.g. by choosing bidding behavior that matches non-fraudulent advertisers' median bidding behavior), I identify several significant behavioral differences between non-fraudulent and fraudulent advertisers that the ecosystem forces fraudsters to adhere to if they want to make money. Most notably, advertisers are largely centralized among a few highly-fraudulent verticals, which Bing can target to aggressively reduce fraud on the platform.

Chapter 4 applies this same methodology to the area of malicious service providers. This study systematically evaluates commercial VPN providers' offerings for evidence of deceptive and fraudulent advertising as well as traffic monitoring and manipulation. By recognizing that the dynamics of making money as a VPN service provider strongly incentivize lying about what countries the VPN provider offers servers in, I develop a signal that identifies when these providers deceive their customers into buying a product quite different from promised.

In Chapter 5, I conclude by summarizing the contributions of the dissertation then briefly discuss the limitations of the framework and where future work may be able to expand or refine the approach.

Chapter 2

Inferring site compromise with Tripwire

In this chapter, we apply our method for developing detection strategies to the common problem of web service compromise. We provide an effective detection method that puts no restrictions on how the attacker compromises a web service. Further, the method can similarly be used by the web service, users, or even by third parties. In web service compromise, the attacker typically takes control of or extracts valuable user information from the service by exploiting a bug in the website's code. Whether email, social networks, or e-commerce platforms, web services contain valuable user information. In this chapter, we describe a technique for inferring the occurrence of such compromises by leveraging the very financial incentives the attacker uses to monetize the compromise. In particular, we focus on a monetization vector which is too sweet for attackers to ignore, but which leaves the attacker open to detection if used: password reuse attacks.

2.1 Introduction

While there are a range of deleterious effects that can occur as a result of site compromise, one of the most pernicious arises from the confluence of data breaches and password reuse. In password reuse attacks, a site is compromised such that all of its accounts and passwords (or, more commonly, password hashes) are exposed. Then, the attacker leverages this information to access other accounts a user has using the same credentials on another site. This situation is exacerbated by the widespread use of email address as a user's standard username [12, 39, 110]. In one recent study, Das et al. estimated that over 40% of users reuse passwords [24] and our own anecdotal experience with stolen bulk account data suggests that up to 20% of stolen credentials may share a password with their primary email account.

Opportunities for such attacks abound, with reports of data breaches now commonplace: in 2016 alone, reports surfaced of 117 million account credentials stolen from LinkedIn [1] and 360 million from Myspace [6]. In 2014, Hold Security reported obtaining credentials for more than a billion users from breaches on several Internet services [92]. Indeed, the market for stolen credentials is thriving, with credentials being sold in bulk for under a penny a piece [108]. The value of these credentials lies in their ability to be used across sites, enabling account compromise at sites otherwise wholly unaffected by the unrelated original site's compromise.

One of the most sensitive and important account credentials targeted are those associated with major email providers (e.g., Gmail, Live/Hotmail, Yahoo, etc.). In modern usage it is these accounts that are the foundation for one's Internet footprint. In particular, online services commonly require an email address to register, to reconfirm accounts, to communicate key information and to reset or recover passwords. Thus, access to someone's email account can be sufficient to gain access to a broad array of other services as well. Only one major email provider has had a public breach to date (Yahoo, in late 2014 [86]), but despite this, individual email accounts are routinely compromised en masse and widely available for sale in underground marketplaces, undoubtedly in large part due to password reuse.

Unfortunately, mitigating the password reuse problem is not easy. While it can be prevented with the use of password managers and two-factor authentication, adoption requires mass changes in user behavior that have proved difficult to achieve. Absent these changes, most providers behave reactively: once it becomes known that a site is compromised, the operators of the compromised site commonly reset the accounts of their users. A reset acts to protect that initially compromised site, but does nothing to protect users accounts at other sites. To mitigate this risk, some large service providers will reset or lock down the accounts of their customers known to have accounts with a compromised site (on the presumption of password reuse).

Even these imperfect responses are predicated on companies knowing of the compromise occurring and then acting on that information. In many cases such compromises may never be discovered, let alone become public. Small sites may lack the staff and instrumentation to detect compromises and even large, well-managed sites have no easy way to identify the source of a breach when their accounts are compromised via password reuse. Even when a service knows that it has been compromised, companies may choose not to inform their users of the breach for political or business reasons [83]. Further, attackers themselves are incentivized to be quiet about successful breaches: if the breach is known publicly, users and service providers can take steps to mitigate the effects, thus devaluing the attacker's cache of credentials.

Given this reality, a critical issue is being able to determine when credential breaches occur in the first place. In this chapter we describe a technique for inferring the occurrence of such breaches (both large and small) without requiring any special access to Internet sites or their hosting infrastructure. Our measurement approach detects site compromises externally by exploiting precisely the attacker's interest in the password reuse vector. In particular, by registering honey accounts at Internet sites using unique email addresses, we place our own accounts at risk, indistinguishable from any other user's account at each site under observation. By further arranging that each of these accounts shares a unique password with its corresponding email account, we create a clear password reuse attack opportunity. If any of these email accounts is ever accessed, such action provides strong and singular evidence of a compromise at the corresponding site. This approach allows a wide array of Internet sites to be efficiently monitored for compromises by an outsider and admits no false positives—presuming the email provider itself is not compromised. Further, an attacker wishing to avoid detection by this mechanism is forced to avoid using stolen credentials for password reuse attacks, which represents a significant lost revenue opportunity.

We have built a prototype system, called Tripwire, to implement this technique, which automatically crawls and registers accounts in this matter. We partnered with a major email provider to conduct a pilot study of this approach covering approximately 2,300 sites. Over a year's time, we discovered evidence of compromise at 19 sites, all but one of which were previously undisclosed to the best of our knowledge. The sites at which we detected breaches range in size from very small to a large publicly-traded company with more than 45 million active customers at the time of compromise. Moreover, by controlling the form of the passwords we can determine whether a compromise is consistent with a dictionary attack on password hashes (and thus users with strong passwords may have been protected) or whether the attacker was able to obtain the passwords in clear text (and a strong password would not have helped).

In the remainder of this chapter, we discuss previous work related to the Tripwire technique, the ethical considerations that guide our study, and our account creation, registration, and monitoring methodology. We then quantify the effectiveness of our pilot monitoring, and report our qualitative experience disclosing our findings to each of the affected sites, including how they receive such evidence and the extra-technical challenges in using it to change behavior. We conclude the chapter with a discussion of the complexities of automated account registration and the challenges in scaling such a service further.

2.2 Related work

The Tripwire measurement technique fundamentally depends on attackers stealing email account credentials on one site and then taking advantage of shared password behavior to access the stolen account on the email provider. Researchers have repeatedly found that users reuse their passwords across multiple services [24, 37], and that they have accounts on at least 25 distinct online sites [23, 44, 34], with some estimates putting the number over 100 sites [25]. Given this landscape, it seems likely attackers will continue to exploit this reuse to try to take over additional accounts that might be of greater value [53]. Indeed, there have been several recent reports of attackers taking a large list of usernames and passwords acquired at one service and trying those credentials at another [12, 39, 42].

Previous work has also shown how to detect sites vulnerable to attack [32, 35, 56], defend against those attacks [7, 9], and evaluate the risk of compromise to a site or predict whether a site will be compromised in the future [102, 117]. Yet Canali et al. found that few shared hosting providers or "add-on" security services managed to detect even simple site compromises, despite direct server access [11]. Tripwire offers an advantage over these and other schemes because it can be deployed and operated by a third party not affiliated with the websites or their users. Moreover, it is able to detect the effects of both online (e.g., key logging) and offline (e.g., external database dump) attacks, relying only on the integrity of a major, independent email provider, not of the sites being measured.

The use of a designated decoy device or account to detect attacks against a service is a classic security mechanism [15], and honeypot accounts have been used to observe recent attacker behavior in general [89] and, in particular, have been suggested as a means of detecting compromise on an online service since at least 2008 [48]. A mechanism of honeypotting has also been suggested to detect password bruteforcing [54]. All of these systems rely on some aspect of the underlying service being measured having not been compromised, and we believe Tripwire is the first use of honeypots where no part of the system under measurement needs to be trusted.

2.3 Ethical considerations

Before detailing our system and methodology, it is important to discuss the ethics and potential for harm associated with our study. First, while we obtained the full consent and cooperation of our partner email provider, we do not seek the consent of the websites that we monitor. It is both impractical and potentially damaging to the scientific validity of our study for us to seek prior consent from websites before registering accounts. In particular, sites might opt-out in a biased way (e.g., those who suspect their security to be flawed might wish to avoid being included), or choose to handle our accounts in a special fashion that would break the link between our account disclosure and site compromise.

Largely because we lack informed consent of the websites under test, we do not undertake our study glibly or without significant deliberation. In our view, there are two distinct issues: one of ethics and potential harm, the other of liability.

It is our belief that the potential direct harm we can cause to a site by attempting to register for a few (at most three) accounts that are rarely, if at all, accessed subsequently is limited to the small amounts of storage and load associated with these actions. Our automated crawler was rate-limited to attempt page loads no faster than every three seconds—and typically much slower than that due to intentional processing delays. Only three sites received more than eight registration attempts from our crawler[1]—with the overwhelming majority of sites receiving two or fewer attempts—a load unlikely to burden even tiny sites.

However, there are also indirect harms which may result to the brand or reputation of such sites if the knowledge of their breaches were to become known. For this reason, we have explicitly obscured the identify of the websites at which we detect compromise. Balanced against these potential harms is the concrete benefit to sites arising from earlier knowledge of a data breach and the benefit to consumers from earlier notification of their credentials being compromised. As detailed below, we attempted to notify the operators of all the sites where we detected compromise.

As we discuss at length later, however, our passive monitoring approach—specifically, one that provides concrete evidence of a compromise but no information regarding the exploit or mechanism employed such as pen testing or similar invasive methodologies would provide—can place notified site operators in a challenging position. Disclosing a compromise or forcing a password reset is, at least for some, perceived as a risky move that could drive users away from a service [83]. Depending on jurisdiction, however, sites may be required to notify users of any known compromise to their service [10]. In cases where a site is unable to independently corroborate a compromise, they are forced to choose between their own investigation and our evidence. Further, without being able to find the source of the compromise, they have no ability

[1] Due to crawler debugging, the three most frequently accessed sites were contacted 16, 15, and 9 times.

to assure users that future compromises will not occur.

On the legal side, we consulted extensively with general counsel and acted with the permission and knowledge of our administration. While we make no attempt to explicitly check the terms of service for each site in our study, it is quite likely that one or more aspects of our methodology are contrary to policies on some sites (e.g., sites frequently disallow "automated registration"). Even if not explicitly disallowed, bot activity is plainly discouraged by many sites through the use of CAPTCHAs and other Turing-test-like aspects of their registration processes, which we intentionally try to overcome. Moreover, if sites asked for personal information as part of the registration process, we provided fictitious details. Nevertheless, counsel advised us that the legal risk was low and outweighed by the scientific merits of the work and, moreover, that both the absence of real damages to any party and the limitations on the enforceability of terms-of-service contracts minimize even these limited risks.

Finally, we note that there are no human subject concerns in this study: all of the information we are providing is fictitious, and no human (other than perhaps an attacker who compromised a website under study) ever interacts with the email addresses or the accounts.[2]

2.4 Methodology

Conceptually, Tripwire consists of two distinct phases: account registration and monitoring. We designed an automated web crawler to register for accounts, and then partnered with a major email provider to monitor activity at the associated email accounts. Here we describe how we created and populated the email accounts used by Tripwire, the way in which we interact with the email provider, and the operation of our web crawling infrastructure.

2.4.1 Account and identity management

Tripwire ensures that each account maintains a one-to-one mapping to an identity. These identities minimally consist of an email address and password, though many sites require

[2] Aside from a handful of phone calls to the numbers associated with our accounts (see Section 2.5.2).

additional information.

Identities

Tripwire identities must not easily be distinguishable from organically created accounts so that attackers cannot selectively avoid them. Hence, we created a database of fictitious identities and associate each with an email account and password at our email provider that were designed to look as organic as possible. Tripwire identities have full names, addresses, phone numbers, dates of birth, employers, etc. We generate names from sets of real names, and addresses are syntactically and semantically valid (although not necessarily extant) US street addresses [33]. Identities have real US phone numbers under our control. No site saw the same phone number more than once.

We generate usernames and email addresses to look plausible, yet be very unlikely to be taken. We generate the local-part of email addresses in the form of an adjective, a noun, and a four-digit number (e.g., `ArguableGem8317`), and then use the first 14 characters as the username at sites that require a username distinct from the email address. (Experience shows that many sites limit the username length.) Since the email provider does not create email accounts for us when there already exists an account with that username, we use this check as a heuristic for probable availability of a given username on all other services. This allows us to reduce the complexity of the crawler (by allowing us to assume that the username is available).

Passwords

We created accounts with two types of passwords to distinguish the types of compromises that may occur. “Hard” passwords are random alpha-numeric, mixed-case, ten-character strings without special characters (e.g., `i5Nss87yf`). “Easy” passwords are eight-character strings combining a single, seven-character dictionary word with its first letter capitalized, followed by a single digit (e.g., `Website1`). Easy passwords are deliberately easy to crack in a brute-force fashion, while hard passwords are designed to be as difficult to brute-force as common password policy constraints would allow.

Nearly every website we crawled permits eight-character passwords, and many require at least eight characters. The hard password ten-character length is a balance between a desire for long, complicated passwords, and the need to support websites with short maximum password lengths. Passwords do not contain non-alphanumeric characters as, in our experience, few websites require special characters in the password, while several do not support them. These assumptions simplify our crawler by not having to consider password policy when trying to create an account.

Tripwire typically registers for multiple accounts on a site by first attempting to register an account with a hard password. If Tripwire believes that registration succeeded, it enqueues up to two additional registration attempts with differing password types. If we later detect compromises for a site only on accounts with easy passwords (or where those accounts are compromised much earlier than ones with hard passwords), it would suggest that, while the website's database was breached, the website's passwords were well hashed. If Tripwire also detects activity on accounts with hard passwords, then it suggests that the site was either storing passwords in plaintext, using a weak hash, or the compromise was able to bypass the hashing step.

2.4.2 Interaction with the email provider

We approached our email provider with the idea of Tripwire because Tripwire works best with a sufficiently attractive target email account. The email provider was only involved in providing accounts to our system, and was not aware of what accounts were used on what services.

We provided a list of identities in advance to our email provider, who then created the corresponding accounts unless they collided with a pre-existing account or violated the provider's naming policies. All email accounts were created with their corresponding name (in case an attacker sought to validate the authenticity of the accounts by checking the personal information at the email provider against the website's records) and forwarded any mail received to our own mail server, where we stored and parsed incoming messages for registration information. Since forwarding addresses are visible in the web interface of our email provider, we used forwarding addresses of accounts at one of a small number of domain names under our control who had their mail hosted by a third-party mail provider. This provider then forwarded messages to our mail server.

In addition to forwarding messages, the email provider notifies us of any successful logins in these otherwise-unused accounts. In particular, we receive sporadic dumps of login information for all of the identities we created, independent of whether they have been used to register accounts at any websites. Our provider is unaware of which accounts have been used, and which remain unassociated. The provider dumps provide timestamp, remote IP, and method (IMAP, POP, etc.) for any successful logins, but does not disclose failed attempts. We also maintain a set of control accounts that are not associated with a website, but into which we log in at our email provider from time to time. All such control login events have been accurately reported by our provider.

2.4.3 Crawler

To scale account registration, we developed a custom-built web crawler to automatically visit a given site and register for an account. The crawler attempts registrations on a 'best-effort' basis: the crawler explicitly does not attempt to support all of the site registration mechanisms encountered on the Web, as our experiment is designed only as a proof of concept and does not require complete coverage.

Registration

The crawler uses PhantomJS [49], a scriptable, headless web browser based on the WebKit engine [130]. It processes pages according to the flow shown in Figure 2.1. It attempts to identify a registration page on the site, and if successful, identify and fill each form field serially. If any stage fails, the crawler aborts with a corresponding error code. The crawler does not support any site whose registration system does not follow this basic flow, nor sites that use external account services such as those provided by Google or Facebook. The crawler relies on many hand-crafted heuristics to locate registration forms, fill them out, and submit them. These heuristics take the form of a series of weighted regular expressions and sets of DOM elements to which they apply. Our current heuristics are only designed to support sites written in English.

If an email address or password was ever shown to a site—regardless of Tripwire's assessment of the success of this submission—we "burn" the identity and forever associate it with that website. If a registration attempt fails, the identity used may be returned to the general pool to be used on another attempt only if neither the email address nor password were exposed to the website. The horizontal line in Figure 2.1 depicts the approximate point at which an identity typically becomes permanently associated with a site.

Bot-detection avoidance

The crawler bypasses some rudimentary bot-checking systems (such as CAPTCHAs or basic human-knowledge questions) by relying on a third-party CAPTCHA-solving service [30]. The crawler operates using a small network of web proxies that our group maintains solely for research purposes. These IPs are not meant to be unattributable—WHOIS records clearly state our institution name. They serve simply to decouple multiple registration attempts at the same site: websites receive at most one account registration from a given IP. We made no attempt to match the geolocation of the proxy IP to the address for a given identity, but in practice this did not seem to prevent registration.

Mail handling

Since many websites require confirmation of email addresses to create an account, our email provider forwards any messages delivered to our accounts to a mail server under our control. This server retains a copy of all messages received, and, as needed, processes incoming message contents. In particular, it processes all incoming messages to evaluate whether a message is associated with a recently-registered account, and, if so, if the message contains a validation link. If it does, the mail server loads the verification page and saves it for future debugging.

2.4.4 Interpreting account compromise

The key assumption with Tripwire is that a successful login reported by the email provider is the result of an attacker having stolen credentials from the site on which we registered with our email account. We end our methodology by arguing why we believe this assumption to be valid.

Acquiring the correct credentials requires collecting them either on a machine storing them, or in transit between credential holders. Credentials are stored in three places: within our own database, with the email provider, and with the site under measurement. We have striven to minimize the risk of leaking data from any of these sources. Our database is accessible only from a small number of servers on a small internal network, none of which provide externally accessible services. Communication between our servers is tunneled over SSH. Individual instances of the crawler have only the identity assigned to one site, so compromise of multiple identities would require full arbitrary code execution on the crawling machine.

The provider treats the email accounts used in Tripwire equivalently to their hundreds of millions of other accounts. The email provider has mechanisms to detect attempts to brute-force passwords. No known breaches of the email provider affected accounts used in Tripwire, and sensitive account credentials were only exchanged between the authors and the email provider via verified PGP.

Perhaps the most compelling evidence of the integrity of the Tripwire accounts is from the fact that no accounts were tripped that were not associated with Tripwire registrations. Tripwire has a database including more than 100,000 valid email addresses and passwords obtained from the provider that were monitored for logins, but were not registered with sites. The unused accounts conveniently serve as honeypot accounts to detect any compromise of the email provider or our own database since they are stored with the accounts used in the study, but have not been used for registrations. None of these unused accounts have ever been accessed.

For the sites under measurement, a possibility is that an attacker brute-forced our credentials without explicitly breaching a site, e.g., an attacker somehow guesses our usernames (or a site exposes them) and the site does not prevent brute-forcing attempts on its accounts. If so, then an attacker could conceivably have found the Tripwire username, brute-forced the password with the site, and then used those credentials in a password-reuse attack on the email provider. While unlikely, we consider this within the bounds of attacks that Tripwire should detect, and Tripwire would correctly declare a site as compromised in this situation.

In communicating with sites under measurement, the system used HTTPS when preferred by the site, validating certificates with a commonly accepted list of roots, and many of the tripped sites used HTTPS during the registration process. It is possible that an attacker may have actively impersonated a site during Tripwire's registration process. But we consider this threat to be an unlikely one, with only a few attacks of this kind having been seen in the wild, primarily due to targeted attacks by state-sponsored actors [18].

Finally, it is possible that a Tripwire account is stored in a sharded database on the site, and only a subset of the shards are compromised in an attack. If a Tripwire account is in an exposed shard, Tripwire indicates that a database breach occurred and still detected a significant compromise of the website under measurement. Conversely, if a Tripwire account is not in the shards exposed, then Tripwire will miss any attacks on the affected users (similar to a breach that did not result in password-reuse attacks). Registering for many additional accounts could reduce the possibility of being stored in an unbreached shard, but we consider this possibility to be remote, and additional registrations introduces ethical challenges that are not outweighed by the benefit to this rare case, especially given Tripwire's otherwise negligible false-positive rate.

2.5 Account creation

We used the Tripwire crawler to register for accounts in batches between July 2014 and July 2016, with most occurring between January and March 2015. Tripwire made 65,413 distinct registration attempts across 33,634 different sites, using a total of 8,352 identities. We detail our validation methodology below. In our best estimate, Tripwire successfully registered for approximately 3,664 accounts on around 2,302 sites.

2.5.1 Website selection

We registered accounts primarily on four occasions from December 2014 through May of 2016. We initially seeded our crawler with the Alexa top-1,000 sites [3] combined with the Quantcast top-1,000 sites [94] (with duplicates removed) in December of 2014. Subsequent registrations occurred from January through March of 2015 covering the Alexa top-25,000 sites. In late November 2015, we attempted registrations on all sites in the Alexa top-30,000. Finally, in May 2016 we manually registered for accounts at all of the eligible Alexa top-500 sites to ensure good coverage of the most popular sites. In each case, we used the most up-to-date rankings available at the start of the registration window.

In all of the automated cases, we filtered URLs through a set of regular expressions to remove sites known to use common backends—e.g., Google, YouTube, Blogger, Blogspot, etc.

2.5.2 Registration attempts

Because our infrastructure has no automated way to validate registrations after it attempts to create them, there is uncertainty in the number of accounts and sites for which the crawler successfully registered. Hence, we rely on heuristics during the registration process, email-based indicators, and manual sampling to estimate success.

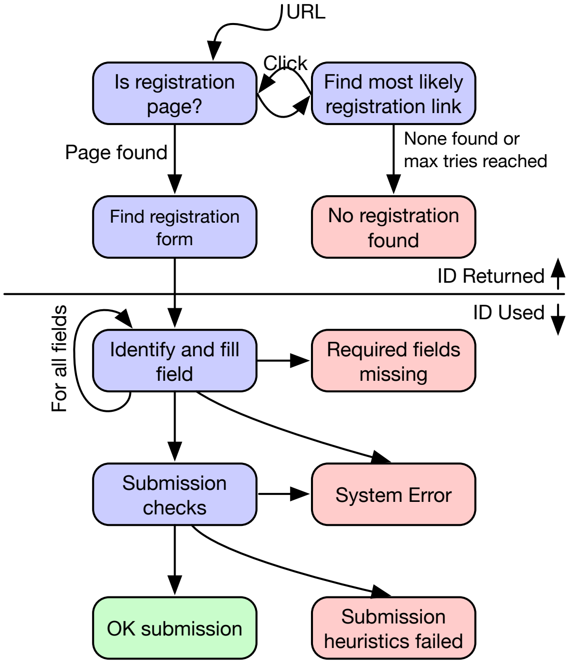


Figure 2.1. Control flow of the Tripwire crawler. Given a URL, the crawler returns the reason for termination.

Crawler termination conditions

Figure 2.1 presents the termination conditions for the Tripwire crawler's execution across various sites. "Required fields missing" indicates that the registration form did not meet the conditions for a valid form (e.g., did not ask for both password and email), or the crawler was unable to recognize a sufficient number of fields to attempt registration. "Submission heuristics failed" corresponds to the case where the crawler submitted a registration, but suspects that it did not succeed, while "OK submission" indicates its heuristics suggest it did. Finally, "System Error" represents cases where the crawler was otherwise unable to process the site. We investigate the outcomes of the crawler in Section 2.7, though we note here that crawler outcome distributions were similar across Alexa ranks.

Table 2.1. Estimates of accounts created by account status.

	Attempted					Estimated Valid			
Account Status	Hard	Easy	Total	Sites	Success	Hard	Easy	Total	Sites
Email verified	1,552	508	2,060	1,359	98%	1521	498	2,019 (55%)	1,332
Email received	128	51	179	106	82%	105	42	147 (4%)	87
OK submission	1,069	703	1,772	860	59%	631	414	1,045 (29%)	507
Bad heuristics/Fields missing	4,395	122	4,518	3,420	7%	308	9	317 (9%)	239
Manual	0	137	137	137	100%	0	137	137 (4%)	137
Total	7,144	1,521	8,666	5,882		2,565	1,100	3,665 (100%)	2,302

Out-of-band confirmation

In addition to the heuristics Tripwire uses at registration time, some sites provide further confirmation of registration via email. If an email account receives an account verification message, we label it "Email verified". If the account receives email, but we do not recognize it as a verification message, we label the account "Email received". Over 47% of "OK submission" results at registration time triggered a verification message, and 4% more triggered at least some kind of email message. Fewer than 8% of the registration attempts of the other categories resulted in any kind of email message.

While it is possible that registrations could also be verified by phone, no phone verification occurred in our sample. We did receive 18 calls from seven distinct self-identifying sources ("Hi, this is John from site X") to our phone numbers that were directly attributable to the accounts we registered. (We received several additional phone calls, but they seemed to be wrong numbers or call-center scams that cannot be conclusively tied to the phone numbers used in Tripwire accounts.) All attributable calls were sales teams following up on what appear to be free-trial accounts for which Tripwire registered.

Success estimation

After accounting for email reception, we have five distinct outcome categories shown in Table 2.1. We manually tested a random sample from each category to determine their expected success rates as a basis for estimating the number of accounts and sites for which registration succeeded.

The “Attempted” columns on the left of Table 2.1 show the number of accounts and sites according to each registration attempt. “Hard” and “Easy” correspond to accounts created with those respective password strengths, and “Total” is their sum. “Sites” corresponds to the total number of sites on which we registered accounts (some sites had multiple accounts). We estimate the success rate in each category by selecting 50 random accounts and manually attempting to log in to the corresponding site. The “Success” column shows the success rate of these login attempts. The “Estimated Valid” columns on the right then show our estimates of the true success rates by discounting the “Attempted” columns by the login success rates.

Email verified. Our highest confidence bin for automated registrations is any account that received a recognized verification email. This category consists of 2,060 automated registrations. In our manual tests of a sample of accounts, they succeeded in 98% of cases, resulting in an estimated 2,019 accounts across 1,332 domains.[3]

Email received. An additional 179 registration attempts received email, but the message did not appear to require email verification. These accounts were valid in 82% of our tests, for an additional 147 accounts on 87 domains.

OK submission. In 1,772 registration attempts, our attempt passed all heuristics for success, but no email was received. In our sampling, 59% of these accounts exist, accounting for 1,045 more accounts on 507 domains.

Bad heuristics/Fields missing. The lowest-probability-of-success outcome is that the system exposed a username and/or password, but the system triggered a heuristic signaling failure or did not attempt to submit the form. In these cases, approximately 7% of attempts still succeed, for an additional 317 accounts on 239 domains.

Manual. Finally, we manually registered accounts at the 137 English-language sites

[3] In the one “failure” case, the site in question is an app-development site partially hosted at GitHub with a local account registration. Tripwire successfully signed up for an account on GitHub instead of the site in question.

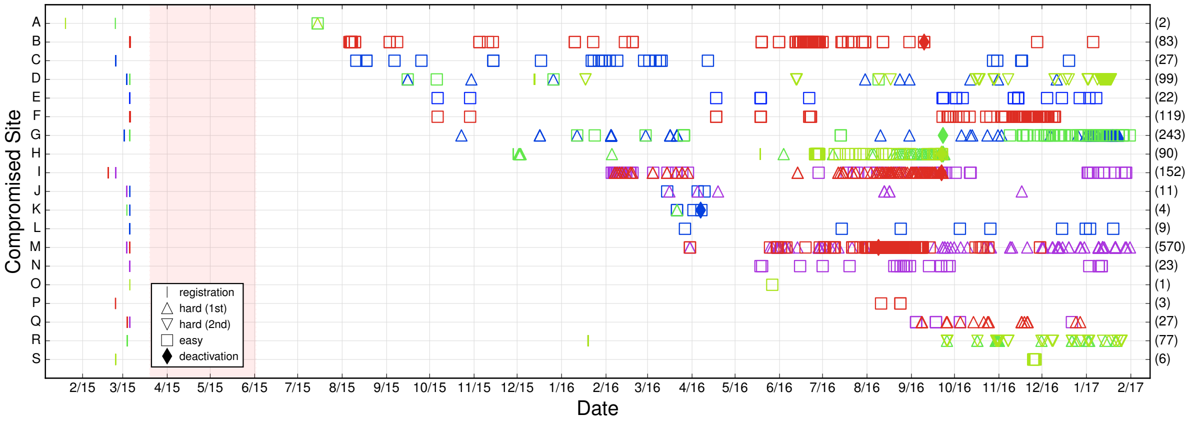


Figure 2.2. Login activity to email accounts stolen from compromised sites. Each row corresponds to a compromised site, and different colors on the same row indicate activity on different accounts at that site. The numbers along the right *y*-axis indicate the total number of logins for that site across all accounts. The shaded region in Spring 2015 corresponds to a gap in our logs.

accepting registrations among the Alexa top-500 sites (4% of all accounts registered).

2.6 Compromises detected

At various points during our study, our email provider reported any successful login activity for Tripwire email accounts. (For non-technical reasons, we were unable to collect login information on a periodic or real-time basis.) As discussed in Section 2.4.4, we interpret a successful account login as indicating a compromise of the associated site. Among the estimated 2,302 sites with successful account registrations, Tripwire detected 19 such site compromises between June 2015 and February 1, 2017.

Figure 2.2 shows the login activity to email accounts stolen from the compromised sites across time. Each row corresponds to a compromised site, vertical ticks show when we registered for accounts on the site, squares show logins to email accounts with easy passwords, and triangles show logins to email accounts with hard passwords. They are sorted from the top according to time of first account login. Numbers to the right of each row indicate the total number of account logins for that site. The shaded region in Spring 2015 corresponds to a gap in our account login data. Due to a misunderstanding of the retention limits at the email provider, login activity was lost from March 20, 2015, through June 1, 2015. Although no logins were detected for more than a month after collection resumed, it is possible that additional sites were compromised and would have tripped our system during that time.

In the rest of this section, we characterize the sites that were compromised and detected by Tripwire, as well as other compromised sites during the same time frame. We then describe our results of disclosing the compromises to the sites. Finally, we summarize the activity of attackers who accessed the stolen email accounts.

2.6.1 Sites compromised

For each site, Table 2.2 shows the approximate Alexa rank, site category, the number of accounts created and accessed, and whether an account with a hard password was accessed. We explore what site characteristics appear to correlate with their compromise, how sites manage their account databases, and which compromises we detected were also disclosed by the sites themselves.

Overall, we find that while most of detected compromises are at small sites with few staff, Tripwire has also detected compromises on large sites as well. Tripwire detected both plaintext and hashed-password breaches, and has predominantly discovered breaches that have previously been undisclosed.

Site characteristics

The compromised sites cover a wide range in terms of popularity. The detected compromised sites are distributed throughout our covered site ranking, from a top-500 site through the full range of sites selected.

The most popular site compromised is a well-known American startup with more than 45 million active customers as of the quarter they were compromised (Site `A`). Sites `E` and `F`, owned by the same parent company, are a large gaming-services company well known within online

Table 2.2. Summary of sites with detected login activity. Rank at registration time rounded up to nearest 500. No 'hard' account was registered at site P.

Site	Accounts accessed	Hard accessed	Category	Alexa rank
A	2 of 2	Y	Deals	500
B	1 of 2	N	Gaming	8500
C	1 of 2	N	BitTorrent	5500
D	3 of 3	Y	Wallpapers	20500
E	1 of 2	N	Gaming	16000
F	1 of 2	N	Gaming	18500
G	2 of 2	Y	RSS Feeds	17500
H	2 of 2	Y	Marketing	17500
I	2 of 2	Y	Horoscopes	7500
J	2 of 2	Y	Gaming	20500
K	2 of 2	Y	Classifieds	20500
L	1 of 3	N	Adult	11000
M	2 of 2	Y	Vacations	20000
N	1 of 2	N	Gaming	11500
O	1 of 2	N	Outdoors	18000
P	1 of 1	–	Adult	1500
Q	2 of 2	Y	Tourism Guide	22000
R	2 of 2	Y	Press Releases	22500
S	1 of 2	N	BTC Forum	4000

gaming communities. We also detected compromise on a top-500 site in India, the top-ranked site in its category (Site I) which claims millions of installs of their app and more than 60 million visits to their site per month. Site P, a 'tube'-style pornography site, is a top-400 site in Germany. Site Q is owned by a company with a large portfolio of travel recommendation websites, claiming 40 million views across all sites every month.[4] Finally, site S, `bitcointalk.org`, is a prominent Bitcoin discussion forum that experienced a publicly acknowledged database breach in May of 2015. Contents of that breach were reportedly sold online in 2016. While the distribution looks somewhat skewed towards lower-ranked sites, there are too few sites to observe the distribution definitively.

The compromised sites comprise a variety of site categories, although gaming (i.e., sports

[4] We did not register at any other sites owned by this company, so cannot speculate whether the compromise is limited to that site or spans across their properties.

or video games) is the most prevalent. These sites are fairly representative of sites with large user bases towards the tail of popularity.

Except sites A, E/F, Q and R, the remaining sites appear to be run by individuals or small teams. A few of the sites have not been meaningfully updated in several years, and site C has since disabled account registration. Most of the sites appear to have been created with good intentions for their stated purpose. Three sites (G, K, M), though not malicious per se, appear to have been created explicitly to generate ad revenue and offer services with little actual value.

Password management

The password strengths of the accounts provide insight into the password management practices of the sites. For sites that only store hashed passwords, easy passwords can be guessed using dictionary attacks while hard passwords remain protected. For sites that store passwords in plain text, both easy and hard passwords are vulnerable.

In eight cases (sites B, C, E, F, L, N, O, S), our system registered for both an ‘easy’ and a ‘hard’ account at a site, but logins only occurred on the ‘easy’ accounts. This behavior suggests that these sites hash passwords sufficiently to at least delay the compromise of accounts with stronger passwords, or are leaking account credentials due to large-scale brute-forcing. Despite well-known security practices, we observed logins using ‘hard’ passwords on ten sites (including site A). These sites appear to have stored account passwords in the clear or used easily-reversed hashes. (For site P, we only successfully registered an account with an easy password.)

Our methodology only registered for accounts with easy passwords after it estimated that a hard registration succeeded. This biases our results to under-report compromises, as ‘easy’ passwords are more frequently compromised. Subsequent invocations of a Tripwire system should avoid this pitfall.

Breach indicators

Of the 19 sites that we detected as being compromised, we found only three with external indications of compromise.

As mentioned above, site A is a well-known, popular American startup. Around the time of our observed logins, several of their users on Twitter complained of their accounts on site A being compromised. One publication ran a story discussing the claims, but the site denied the allegations. We can find no further reporting on the issue, but our account logins on site A corroborate these reports, show evidence of attackers using stolen account information for password reuse attacks, and serves as an example of our system providing ground-truth evidence. Section 2.6.3 details our discussions with the site when we disclosed our account compromises; from their investigations, they reported finding no internal evidence of a site compromise but could provide no explanation for our results.

We also found a post on an unrelated forum claiming to provide a link to the user database of site L, a pornographic 'tube'-style site. We were unable to verify the availability of the database or the validity of the claim, but the posting time is consistent with the login attempts we see on account l1.

Finally, we detected an account compromise on site S, `bitcoin talk.org`. This site was known to be compromised as of May 2015, and there were subsequent reports of the database of hashed passwords being for sale on underground markets in 2016. Our detections are consistent with this timeframe, and consistent with the leak of hashed passwords [20].

We could find no evidence of disclosure of any of the other compromises. We provided the usernames we used on sites A–O to several major threat intelligence companies and online service providers in possession of large collections of compromised accounts, and none of the companies found any evidence of breaches.[5] We also searched a variety of public and private sources of compromised database dumps for evidence of our breaches without success.

[5] These companies requested to not be named in exchange for their assistance.

Recovery from compromise

Although we found only three external indicators of account breaches, most of the sites appear to have either only been compromised at a single point in time, or were able to recover from the breaches. We registered for additional accounts on all sites except `C` (which no longer accepts registrations) and `O--R` (whose compromises had not yet been detected) as of mid-May 2016. To date, only our additional account at site `H` has been accessed and none others.

2.6.2 Undetected compromises

It is natural with a system like Tripwire to want to calculate the proportion of compromises that Tripwire is able to detect (i.e. recall). Unfortunately, such a calculation is not possible in practice, as it is not possible to generate an accurate number of total compromises that have occurred in the open Internet. Further, Tripwire does not attempt to detect all compromises—it merely aims to expose compromises that otherwise would have gone undetected. It is still valuable, however, to understand why our implementation fails to detect an otherwise known breach. In this section we explore why Tripwire did not detect 50 recent data breaches as listed on a site that curates public data breaches [46] and, for each, examined why Tripwire did not detect a compromise.

Missed due to scale/scope

In 22 of 50 cases, the sites involved with the compromise were ranked too low according to Alexa to make our test corpus, despite many sites being quite large. Choosing sites to target is a non-trivial problem. While Alexa provides high-quality rankings for the most popular sites, our experience has shown it to be less reliable for the long tail of the distribution.

In seven instances, the sites involved with the compromise were not in English (six were Chinese-language sites and one was Russian). In both cases, a larger-scale and language-agnostic deployment of Tripwire may have been able to register for accounts without issue.

Missed due to technical challenge

In 14 of the 50 cases, the sites were missed due to limitations of the Tripwire prototype. In five cases, the crawler failed because it was not designed to handle multi-page registration forms. In four additional cases, the crawler failed a bot-detection check (e.g., a CAPTCHA image or a free-form common-knowledge question). In one case we successfully registered, but failed to properly verify the accounts by clicking on links sent to the email address.

In four cases, the crawler was unable to locate the registration page either due to it not being clearly accessible from the home page of the site, or because the registration page was not obvious based on the text of the page (e.g., because they relied on text embedded in images). These cases represent significant technical challenges which any implementation is likely to struggle with; however, they are not fundamentally at odds with the Tripwire approach either. For this limitation, it may be possible to rely on search engines to help locate the registration pages.

Missed due to inherent limitations

In six cases, Tripwire could not have registered for accounts either because the site required payment (two cases) or because the site did not support online registrations (four cases) due to accounts being established by external means (e.g., being a customer of a bank). In one case, the site limited the length of the email address to fewer than 16 characters but the address we tried was 18 characters. These sites are largely out of scope for Tripwire. While technically possible to automate payment, it is quite difficult to scale. Sites that have login systems but that do not support purely-online registration (e.g., many banks) are similarly out of scope.

2.6.3 Disclosure

Given the significance of account breaches, we contacted all sites from which attackers had gained access to our accounts (except for one case where the breach became publicly known). We disclosed our identities, methodology, and findings, and engaged with each site to the extent that they were willing. Although each site had its own unique situation, we can summarize our findings as follows:

- Six of eighteen sites responded to our disclosures.
- Sites that did respond typically responded quickly and took the disclosures seriously.
- Only one site directly corroborated a breach, and the compromise was previously known to them. Some sites acknowledged that security was not their highest priority. No sites that disputed our claim were able to explain how our accounts could otherwise have been compromised.
- No sites have notified users to date (although one site said they would force a password reset).

Disclosure Methodology

We disclosed to sites in two batches, with most occurring on September 7th, 2016, and sites compromised after that date on November 4th, 2016. We sent similar email messages to each site, making small changes as applicable (see Appendix A for our templates).

The first message we sent identified ourselves as researchers, explained that we believed that usernames and passwords at the site had been compromised, that we were willing to discuss details with the appropriate party, and asked if such a person could respond. We chose this phrasing both to provide confidence and to encourage a response. If we received a response from the site, we sent a second message explaining our methodology and some details of the specific compromise. Subsequent messages, and phone calls if requested, were answered as needed. Verbatim responses from several sites are available in Appendix A.3.

We chose recipient email addresses by looking for contact information on the site, emailing the registrants listed in WHOIS data, and emailing common email addresses that might be relevant (e.g. ‘security@example.com’, ‘webmaster@example.com’). No site provided an obvious direct method for contacting the appropriate security contact. In each case, we emailed the complete set of addresses in case any individual address was invalid. We sent messages from the first author's institutional email address, with other authors' institutional addresses CC'd.

Sites without responses

Twelve sites — B, D, H, I, J, K, M, N, P, Q and R — did not respond to messages. Though we found additional contact information for a number of the sites, these messages also did not receive a response. A message to site I resulted in an automatic creation of an account at their internal ticketing system, but no response was ever generated. Site J had no MX record. Site M's email was forwarded to another domain that had expired and was purchased by a domain squatter. For site M, we also sent email to an email-to-SMS gateway address used as a contact in their WHOIS records.

Site A (Deals, Alexa rank 500)

The head of security at site A responded within 10 minutes of our initial notification asking for details. Per their preference, most subsequent communications with them were either PGP encrypted or by phone.

Site A asked if we were willing to sign a mutual NDA, which we declined. Per their request, the authors met over the phone with the head of security of site A, an additional engineer, and a member of their in-house counsel. During this call, their head of security asked questions to vet the process of detection and our methodology.

Site A understandably lamented the significant delay between initial compromise and notification—an artifact of our specific measurement implementation, and worse for site A than for other sites. The operators of site A reported that, after our initial disclosure, they employed a third-party incident response team to investigate. Despite both internal and third-party efforts, they were unable to find internal evidence of the breach, but did not have an alternative explanation for how our accounts were compromised. Site A did acknowledge that they were aware of the article we made reference to in Section 2.6.1.

Site C (BitTorrent, Alexa rank 5500)

Six days after notification, we received a request for more information, and the subsequent conversation provided details from the operator of site C. The operator explained that an attacker had managed to compromise the site sufficiently to create a competing clone in 2016. Our notification was the first indication to the site owner that the vulnerability had been used prior to 2016.

Site C's owner explained that until this year, passwords were simply hashed with MD5. When asked about whether they would be disclosing the attack to users, they indicated that there was no need, given that 'this information has already become public sine the hacker started a sote fork some months ago' [sic]. When asked about any technical countermeasures, the owner responded with 'sorry cannot tell. however be assured user are protected well'.

Sites E and F (Gaming, Alexa ranks 16000, 18500)

Within 30 minutes of the initial message to the owner of sites E and F, the primary author received a voicemail from the in-house counsel of the company attempting to verify our identities. Our initial notification message did not include telephone contact information for the author, but that information was readily available via online search. Shortly after the initial voice message, we received an email message asking us to confirm via phone that we had indeed sent the message, and to read their responsible disclosure policy.

We received a follow-up from the head of security at the immediate parent company explaining that they were unable to corroborate the data from our study with any of their internal information, and expressed understandable frustration that so much time had passed between event and notification.

The company was very interested in obtaining all related information available, including communicating with the email provider. We provided them with timestamps and IP addresses associated with all relevant logins. Pages on their sites list usernames, and the company asked if these could have been used by an attacker to brute-force guess passwords either at the sites or the email provider. While our email provider provides checking against brute-forcing, sites `E` and `F` do not. But if indeed this is what occurred, then it represents a compromise consistent with Tripwire's goals.

Site G (RSS Feeds, Alexa rank 17500)

The owner of site `G` responded three days after notification inquiring about our dataset. Upon explaining our data and methodology, the owner responded that, after looking for a while, he did find some SQL commands that were improperly escaped, and he knew that his server was under constant SSH brute-forcing attempts, but that he had not been aware of any prior breach. The owner also explained that he needed to update his installation of WordPress and that he would force a password reset after he had finished development. To date, a required password reset has yet to occur.

Site L (Adult, Alexa rank 11000)

The owner/admin of Site `L` explained that he had started the site in 2007. Although he personally had 'a low level of IT knowledge', in April of 2015 he got rid of his system administrators due to their cost and because he felt they were making his job harder, not easier. Before recently migrating to a cloud provider, his site ran on approximately sixty dedicated servers. Since removing his system administrators, he has been running the site himself, and that 'being thrown in the deep end is an understatement'.

By the owner's own evaluation, security had not been a priority for the site: most of the code is from 2008, and requires PHP 5.3; passwords have only been stored in a hashed ('encrypted') form since 2015, but are still unsalted; the site suffers from some known XSS vulnerabilities that he has been intending to fix. The owner speculated that the compromise could be related to a large DDoS attack he experienced around the time of compromise which lasted several days.

He explained that he plans to prioritize salting passwords and upgrading his PHP and web server versions, although he was not presently planning on notifying users of the breach.

Site O (Outdoors, Alexa rank 18000)

We received a response from site O less than 45 minutes after the initial notification was sent. This response, from the CEO of a competing site, explained that they had recently acquired site O from a major American travel-reviews company and that they had transferred accounts from site O to their own site in May of 2016 (the timeframe that our accounts were compromised). After we responded with our methodology and data for their site, the CEO responded saying that they were unaware of any account breach, but that they had performed a "lot of scripted testing" of logins onto their own site to ensure a smooth transition. Additional clarifications and questions regarding actions they planned on taking did not receive a reply, and users of site O have not been notified of the compromise.

Discussion

We believe that account information was stolen from the sites at which our registered accounts were accessed. As discussed in Section 2.4.4, we took many steps to ensure the integrity of our methodology, but we cannot categorically rule out the possibility that either the email provider or our own systems were compromised and that this was unwittingly the source of the account leaks.

However, the empirical evidence is inconsistent with the accounts being obtained via a breach other than at the sites at which they were registered. We had over 100,000 email accounts from the provider, only a subset of which were used to register accounts. Only a small number of those accounts were ever accessed, and all the ones accessed were used to register accounts. It also seems unlikely that an attacker would have defeated our operational security (or that of the email provider), obtained the account credentials, and then accessed only a fraction of the accounts acquired. Moreover, the odds that, in so doing, they would have happened to select just the accounts we used at these sites seems vanishingly small. Realistically, they would also need to know the sites at which we used each account, and have some reason to specifically target the accounts at those sites.

When engaging with the sites, only one of the sites we contacted (Site C) was able to confirm that their site experienced a breach, and in this case the breach was implicitly public since the site was illegally cloned by an attacker. Even in this case, though, the owner did not explicitly notify the site users that their account information had been stolen. All other sites were unable to confirm a breach. Yet, none of the sites were able to offer another explanation for how our account information could have been stolen, and in two cases we have other corroborating evidence (Section 2.6.1).

Given this situation, there are two immediate possibilities for why sites may not inform users about a breach. One is that the sites did not have sufficient information to corroborate the breach. Indeed, consider the perspective of the sites we contacted. The disclosures we provide inform sites that they have been breached, but do not give any information about how this occurred. Tripwire provides bounds of a compromise timeframe (between account registration and first login), but those bounds can be quite broad—in our study, this period was more than 18 months in the most extreme case. Further, while sites naturally asked to know which accounts on their service triggered detection, there is little information to be gleaned from these accounts, provided the compromise occurred after registration time. Such information provides sites with little insight on where to look for evidence of a compromise, nor how to prevent it from happening again.

Finally, even if a site believes Tripwire's evidence that a breach occurred, the specifics may not be sufficient to convince sites to incur the cost of acknowledging a breach. There are substantial potential legal and financial repercussions of publicly acknowledging a breach, particularly for sites run by businesses. The knowledge of a small number of leaked accounts, internally confirmed or not, may not constitute sufficient risk given the potential cost.

2.6.4 Attacker behavior

Lastly, we characterize the activity of attackers with the stolen email accounts [89]. In general, most attackers accessed the accounts repeatedly over the observation period. Although some accounts were shut down for sending spam, in many cases attackers have not taken active steps to use the accounts beyond siphoning email. Accounts appear to be accessed through a global network of predominantly compromised residential machines acting as proxies, typically via IMAP. Account login timing and frequency suggests that credentials are being fed into automated collection systems. We have released our data for these accesses with lightly reduced granularity, which we discuss in more detail in Secion 2.7.4.

Login frequency

Table 2.3 lists the email accounts accessed, the type of password used by the account, the total number of accesses, and the number of days between account registration and first remote access, number of days since last access (as of Feb. 1, 2017), and the number of days between the first and last accesses. 'Frozen' indicates whether the account has been frozen by our email provider due to suspicious activity. The account aliases encode the sites at which they were registered (e.g., we registered account `a1` at site `A`).

Two of the sites (`E` and `F`) show periodic, temporally aligned logins. Manual inspection revealed that these two sites were owned and operated by the same entity, and appear to use the same registration backend. Otherwise, we found no discernible pattern across accounts regarding access timing. The data shows both recurring and non-recurring logins for sites: at the most popular site `A`, both accounts were only accessed once, while account `m1` has been accessed 392 times. Accounts from several of the sites exhibit behavior consistent with ongoing observation or scraping rather than simply verifying credentials.

Table 2.3. Number and date range of login activity for compromised accounts. "Until" indicates the number of days between registration and first access. "Since" indicates the number of days since the most recent login (as of last check).

	Type	# Logins	Until	Since	Frozen	Days Accessed
a1	hard	1	175	569	N	0
a2	easy	1	141	569	N	0
b1	easy	83	153	28	Y	518
c1	easy	27	167	45	N	496
d1	hard	10	195	53	N	452
d2	easy	4	193	177	N	328
d3	hard	85	35	15	N	366
e1	easy	22	214	26	N	459
f1	easy	119	214	54	N	430
g1	hard	181	235	10	N	458
g2	easy	62	311	2	Y	385
h1	hard	42	3	132	Y	296
h2	easy	48	38	133	Y	88
i1	easy	58	345	5	N	358
i2	hard	94	353	133	Y	228
j1	easy	3	374	299	N	26
j2	hard	8	378	78	N	245
k1	easy	3	381	301	Y	16
k2	hard	1	383	318	N	0
l1	easy	9	387	14	N	298
m1	hard	207	392	2	Y	306
m2	easy	363	390	65	Y	244
n1	easy	23	439	22	N	237
o1	easy	1	447	252	N	0
p1	easy	3	533	162	N	13
q1	easy	9	548	43	N	108
q2	hard	18	553	37	N	110
r1	hard	38	571	12	N	118
r2	hard	39	250	8	N	121
s1	easy	6	639	68	N	1

Bursty logins

Although no overall pattern emerges, eleven of the accounts have bursty login behavior where multiple logins occur to the same account from different IP addresses in rapid succession of each other. In the peak case, g1 experiences 46 distinct IPs accessing the account over 10 minutes. This behavior suggests that the systems used to login to accounts are very loosely coupled and failure is common. Nine of the accounts (b1, e1, f1, g1, k1, k2, m1, m2, r2) experience bursts of logins wherein a single IP accesses the same account dozens or hundreds of times within a few seconds. In the extreme cases, this can make up more than 75% of the logins seen for an account.

Login IPs

The IP addresses originating the account logins are consistent with large-scale botnets of leased proxies. As of our final check, a total of 1316 distinct IPs logged into the our accounts across approximately 1792 login attempts. Only 181 IPs appeared more than once in the logs, with one IP appearing 58 times (to account r2).

Based on WHOIS data, the most popular countries represented are Russia (194 IPs), China (144), USA (135), and Vietnam (89), with a total of 92 countries represented. Combining manual analysis of WHOIS with DNS, the majority of these IPs appear to be residential/consumer IPs, though several higher-volume IPs map to datacenter IPs with hosts serving legitimate content, suggesting compromised servers.[6]

Account activity

Since one of the goals of site compromise is to steal accounts, it is somewhat surprising that many of the stolen accounts have been relatively idle. No email account that has been accessed has received any unexpected email messages beyond a few generic spam messages.

Eight of the 27 accounts do show suspect behavior, though. The email provider forced a password reset on one of our accounts, m1, after recognizing account compromise. Accounts b1, g2, h1, h2, i2, k1 and m2 were all deactivated by the email provider for sending spam. Prior to being shut down, account g2 had had the password changed and our forwarding address removed

[6] While we did not check extensively for spoofed reverse DNS, several spot checks suggested that reverse DNS either matched forward DNS or contained domains owned by the owner information present in WHOIS.

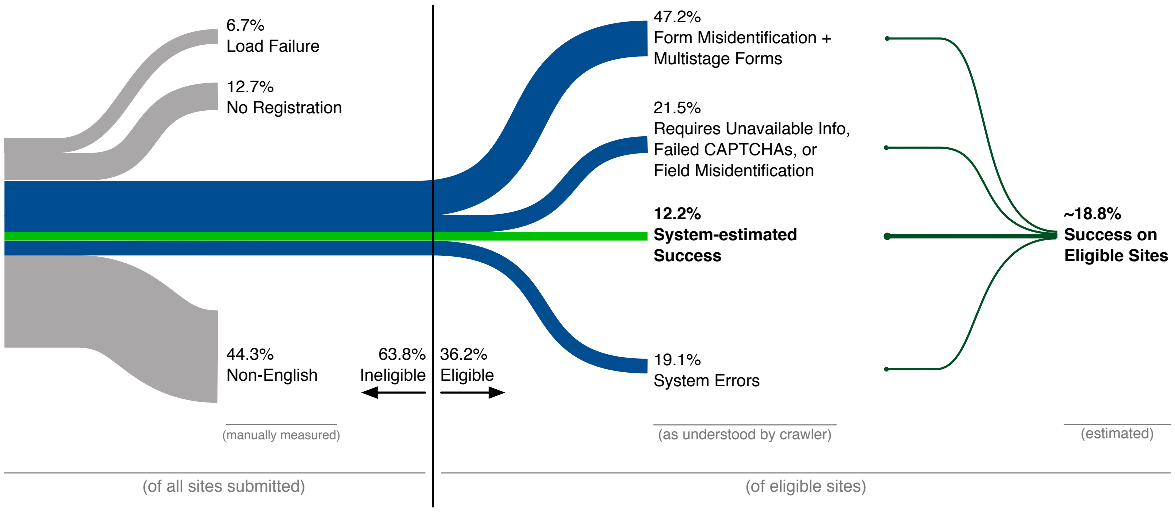


Figure 2.3. Possible outcomes of Tripwire registration attempts. The left and right thirds of the funnel are estimates, while the middle corresponds to crawler-measured outputs.

by the attacker. For the accounts where passwords have not been changed, one possibility is that attackers are stockpiling the compromised accounts for later use or sale. Another possibility is that attackers watch these accounts for messages from sites such as banks that can be leveraged for direct monetization.

2.7 Discussion

Though just a means to an end for our study, automated account registration is also potentially useful for others. We lead this section with more details on our registration results, lessons learned, and what would be required to further scale such a system.

Since a system like Tripwire must be robust against circumvention, we follow with a discussion on what would be required of an attacker to evade detection when compromising a site under Tripwire-like surveillance. Finally, we end with a brief discussion of what data and source we are making available.

Table 2.4. Registration eligibility of sites as determined by 100-site manual sample.

Start Rank	Load Failure	Not English	No Registration	Ineligible	Rest
1	3%	43%	7%	4%	43%
1,000	9%	37%	15%	6%	33%
10,000	8%	53%	16%	5%	18%
Average	6.7%	44.3%	12.7%	5.0%	31.3%
100,000	8%	43%	29%	3%	17%

2.7.1 Site eligibility

To evaluate what proportion of sites are even eligible for a Tripwire-like system, we manually visited three sets of 100 sites from the Alexa rankings, starting with Alexa ranks 1, 1,000, and 10,000, and Table 2.4 shows the results. On average, 6.7% of the pages failed to load, and 44.3% of pages rendered by default in a language other than English. Nearly 13% of them did not support any web registration, while 5% required a credit card or other information that Tripwire is unable to provide. In the end, fewer than a third of the sites were even plausible candidate sites for automated account registration.

One notable trend is the precipitous decline in the fraction of sites with viable registration pages (from 43% in the top-100 to 18% at top-10,000).[7] This trend does not affect the percentage of load failures and non-English sites, indicating that sites become decreasingly useful for registrations as one proceeds down the Alexa ranking. Although we did not use them, search engines may be an alternate source of sites to monitor.

Systematically, although we visited tens of thousands of sites across the Alexa rankings, only a fraction of them were compatible with our automated registration system. Figure 2.3 depicts the funnel of website registration attempts starting from the full set of URLs supplied to our automated account registration system on the left to the resulting set of successfully registered accounts on the right.

[7] For added scope, we also manually visited another 100 pages starting at Alexa rank 100,000 with similar results as the top-10,000.

We input sites to the crawler without any additional knowledge about the sites other than URL and Alexa rank. The crawler ignores non-English or otherwise ineligible sites. The first third of the figure breaks down the reasons that our crawler is unable to register for an account, which we estimate to be about 64% of cases (see Table 2.4). Our crawler fails to find a registration page in about 69.2% of cases. In a manual inspection of 181 of sites where it failed, we only found valid registration pages on eight of them. This finding is consistent with an estimated false negative rate of around 5%, suggesting that if a site is completely ineligible for the current version of the crawler, the crawler is unlikely to identify a registration page on that site.

Any study that relies upon registering accounts across many sites likely has a notion of "high-value" sites, such as very popular sites. Although we originally intended to solely use automated means for registering accounts, in the end we augmented that process with manual registrations for top-ranked Alexa sites (Section 2.5.1). We consider the additional manual effort for high-value sites to be well worth the cost since registrations need only occur once.

2.7.2 Extending the crawler

The middle third of Figure 2.3 visually depicts the outcomes from the crawler (omitting the proportion in which no registration was possible). The final third shows the success outcome after accounting for email verification and discounting the various categories according to our success estimation methodology (Section 2.5.2). With the present system, the automated success rate is roughly 20% even when considering only eligible sites. What steps are necessary to improve the success rate?

Non-English sites alone make up more than forty percent of all sites, none of which are presently evaluated. Supporting multiple languages would be the single greatest improvement to the crawler's coverage. More tuning could also go into the heuristics used by the Tripwire crawler. Even with this and other improvements, however, automated registration on arbitrary sites is a sufficiently ill-formed problem that additional steps would be necessary.

Bot detection. In our manual study above, 19% of sites (37% of the top-100) with registration forms used some kind of test to ensure the registration form was being filled out by a human actor. If our crawler recognizes that a field is asking for human validation, it defers to third-party CAPTCHA-solving services (or, if available, a human operator). Such solving services have non-trivial error rates [76], and the crawler has no ability to handle interactive CAPTCHA services like modern reCAPTCHA [95] or KeyCAPTCHA [58].

Multi-stage forms. Around 10% of sites with registration forms that we tested have multi-step forms, in which a user completes a portion of the form before being able to advance and complete the remainder. Our crawler makes no attempt at handling these multi-step forms, resulting in both failures to recognize the first page of some registration forms (a 'no form found' result), and to fill out subsequent pages (a 'bad heuristics'/'field missing' result).

Form and field misidentification. A common failure mode for the crawler is to misidentify the meaning of individual form fields or to not recognize a given form as a registration form. Machine learning techniques would likely more reliably identify such forms and fields instead of heuristics.

Invalid identity assumptions. We chose usernames and passwords based on common policies at sites, but a small number of sites have password policies that have uncommon requirements (e.g., require special characters). Our crawler makes no attempt at inferring these policies, and since our usernames and passwords are created ahead of time, we currently have little ability to correct for these cases.

2.7.3 Evading Tripwire

The results presented in this chapter have the advantage that no system like Tripwire (involving coordination between unrelated services to detect compromise) has previously existed, and attackers are thus unlikely to try to evade our detection. Future implementations of a similar system will not have that luxury, thus it is worth a brief discussion about what an informed attacker could do to evade Tripwire's detection. In this subsection, we assume that, at a minimum, an attacker knows that Tripwire exists, and generally how it works.

Avoiding Tripwire detection amounts to avoiding logging on to an observed email account in Tripwire. An attacker may be able to avoid this detection in a variety of ways, but each requires trade-offs. Firstly, an attacker could compromise the user database of a site not under our measurement. This is not so much an attack on Tripwire, so much as it is an acknowledgement that a system like Tripwire cannot have perfect coverage. Knowing what sites to attack requires having compromised the Tripwire operator, and thus evasion otherwise amounts to taking calculated risks on sites Tripwire was unlikely to cover. An attacker could also avoid detection by not attempting logins with the email provider, or by attempting to pick and choose which accounts to check. The odds of detection are inversely proportional to the percentage of email accounts tested. If all the attacker cares about is what approximate proportion of accounts re-use their password for the corresponding email accounts (if, for instance, the attacker was preparing the accounts for resale), then perhaps testing only a small sample may be sufficient. Alternatively, an attacker can also avoid detection by testing other accounts in lieu of testing the email account (e.g., at an OSN). As mentioned earlier, however, avoiding testing the credentials with the email provider closes off a substantial opportunity for monetization.

If the email provider for Tripwire were known, an attacker could choose to avoid checking accounts with that email provider. While effective, we chose a prominent email provider in part because a significant fraction of organic accounts on any service are likely to use this email provider, and thus this strategy is not without cost. As a happy side effect of our not disclosing our email provider partner, attackers are also left to wonder whether they must avoid checking all accounts from the largest email providers. Accounts with the largest providers, however, likely account for a significant majority of the accounts found in the breach.

An attacker may also attempt to determine specifically which accounts belong to Tripwire-like systems. Were the attacker able to determine the entire list of Tripwire accounts (for instance by compromising the Tripwire operator or the email provider), they would be able to form a complete blacklist of accounts to avoid, and completely evade Tripwire's detection. Provided that neither Tripwire's operators nor the email provider are compromised, an attacker must attempt to infer this information from signals associated with the accounts.

As discussed in Section 2.4.1, usernames, passwords and other identity information were chosen to look plausible, and thus hard to identify as part of the Tripwire system. If an attacker has access to information regarding initial registration, the attacker may be able to deduce which accounts are ours based on registration IP address. For our study, for ethical reasons and transparency, we registered for accounts with IP addresses that were clearly owned by our institution, but an operational deployment of Tripwire should use plausible user IPs to avoid this technique as a detection mechanism.

2.7.4 Data and source availability

Tripwire uses a variety of heuristics to find and fill registration forms, as well as to handle incoming email. All of these heuristics are detailed in the source code for the crawler, which is available at https://github.com/ccied/tripwire. In addition, we have provided an anonymized version of the login data at the same URL. This data consists of an entry for each login event. This record provides the account alias (e.g. 'a1'), a timestamp (rounded to the day), /24 of the accessing IP, and login method (e.g. 'IMAP'). This anonymization was chosen to balance the desires of transparency and protecting the accounts in the Tripwire sample.

2.8 Conclusions

Website security is a critical problem whose personal and financial impacts are continuing to grow. While preventing and containing site compromise and account disclosure are clearly of utmost importance, experience suggests that there will never be a time without website compromise. In this chapter, we have shown that the same incentives that give rise to this attack ecosystem can be leveraged to passively monitor for compromise at a wide range of sites and detect compromises of which site operators are either unaware or unwilling to publicly disclose. A major open question remains, however, in how much (probative, but not particularly illustrative) evidence produced by an external monitoring system like Tripwire is needed to convince operators to act, such as notifying their users and forcing a password reset.

Acknowledgments

This chapter, in full, is a reprint of the material as it appears in the Proceedings of the ACM Internet Measurement Conference, 2017. DeBlasio, Joe; Savage, Stefan; Voelker, Geoffrey M.; Snoeren, Alex C.. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Search advertiser fraud on Bing

In the previous chapter, we showed how to detect credential theft from web services by leveraging the attacker's incentive to monetize the stolen information. This incentive caused visible side effects that the attacker could not prevent without taking a substantial financial hit. But does this top-down incentive-driven approach work in other contexts? In this chapter, we shift gears to explore the case when an attacker, acting as a malicious user of a service, works to abuse that service. In particular, we explore fraudulent advertising on Bing's web search, investigating the behavior of these attackers from several angles. Ultimately, we identify a strong opportunity for mitigating fraud and abuse on the platform by focusing again on the attacker's trade offs between blending in and taking full advantage of their opportunity to make money.

3.1 Introduction

Today's Internet ecosystem delivers free access to a wide variety of services from a host of providers, much of which is underpinned by a common advertising revenue generation mechanism. Web search engines, in particular, derive much of their revenue from paid search advertisements. Unfortunately, recent reports indicate fraudulent online advertisements are often being used to mislead and even harm unsuspecting Internet users. Savvy web surfers have become rightly wary of clicking on ads, which, in the end, may bite the hand that feeds us all.

By all accounts, online advertising remains a thriving industry, with Forrester Research reporting that online advertising spend will surpass $100 billion by 2019 [98]. Search advertising depends critically, however, on the trust and participation of the public at large. If end users are unwilling to click on advertisements due to the risk of fraudulent ads, these fraudsters have the very real potential to undermine the search ad ecosystem and the services it supports.

For web search, ads display content associated with keywords that users search for. When fraudulent advertisers advertise on the platform, as with benign ads [118], their goal is to entice users to click on the ad, and thus visit their website. Once a user has clicked on an ad, the fraudulent advertisers monetize these visitors through a variety of illicit and malicious means, such as selling counterfeit goods, tricking users into paying for fake anti-virus products, over charging for technical support, or even injecting malware on unsuspecting user's machines.

Given the alarming rate at which new advertising scams seem to appear, it is an open question whether the current model of ad-supported search is sustainable. Very little is publicly known about the vitality of online search advertising as an enterprise, in particular the true costs of fraud and abuse to the search ad networks themselves. Fraudulent ads are a very real problem for search engines, who have strong motivation to defend against them. Although search engines normally profit from users clicking on ads, fraudulent ads often are not billable (if, for instance, the advertiser is using a stolen payment instrument), and, instead, search engines lose legitimate revenue from non-fraudulent advertisements that may be displaced.

We present the first characterization of the fraudulent advertiser ecosystem at Bing, one of the largest search engines, from the perspective of the ad network. Using over two years of data regarding advertiser spend and campaign management, along with user engagement (i.e., impressions, click-through rates, etc.) we measure the scale of the activity, the particular bidding behaviors exhibited by fraudulent advertisers, and the impact those behaviors have on non-fraudulent advertisers.

Our results show that despite a large fraction of new account registrations being fraudulent, Bing successfully prevents most from showing even a single ad. Further, fraudulent advertisers that do succeed in posting ads typically survive only a few hours to days, forcing many fraudulent advertisers to act aggressively. This aggressive action, where fraudulent advertisers opt to advertise more quickly, rather than more subtly, causes many fraudulent advertisers to stick out.

Further, in order to reach their customers, fraudulent advertisers are largely forced into a small set of relatively lucrative, but often dubious verticals (e.g., weight loss supplements and designer sunglasses). This targeting comes from the very nature of web advertising—advertisements that do not provide what they claim are punished by having fewer opportunities to show their advertisements. Similar to the situations seen in Chapters 2 and 4, these malicious actors on Bing can choose to blend in more—by advertising on other verticals—but doing so comes at the price of less traffic and thus lower monetization. Also similar to other chapters, this behavior provides an opportunity for Bing to defend their platform by targeting these verticals and thus limit fraudulent advertising.

3.2 Background and related work

At a high level, attackers commit fraud in the online advertising ecosystem in two ways. The first is click fraud, in which attackers generate fake 'clicks' to earn payment by posing as the publisher. Click fraud has been extensively studied in the past [26, 27, 28, 45, 55, 68, 69, 75, 103]. Efforts have studied both the infrastructure—often botnets—used to generate the clicks [4, 27, 28, 91] and the quality of the traffic so generated [103, 138]. More sophisticated forms of click fraud are emerging in the mobile space, where unscrupulous actors place ads in locations on the screen where real users are likely to accidentally click on them [63].

In this chapter, we focus instead on fraudulent advertisers who post ads to attract legitimate click traffic for a variety of malicious goals, including trying to infect the user's browser with malware (drive-by downloads) [36, 62, 97, 136], stealing email and bank account credentials with fake pages (phishing) [62], collecting personal information to sell to third-party marketing companies (lead generation) [131], bundling malware with software downloads (download stuffing) [47], selling 'miracle' supplements and nutraceuticals (diet or body-building supplements, anti-aging creams, etc.) [57], or perpetrating money-making scams such as convincing the user that their computer is infected with malware and selling them fake anti-virus software [105].

In general, attackers post fraudulent ads at scale in two ways. Either they compromise the accounts of existing legitimate advertisers, or they create new accounts with fraudulent information, including names, email addresses, and credit card information (which is typically stolen). As a result, search engines have stringent account validation (such as credit card verification) for new accounts [41], and also provide tools for advertisers to better protect their accounts [73, 74]. Search engines also proactively attempt to detect fraudulent ads posted by advertisers. When new ads are created, search engines vet the site linked to by the ad at posting, and again when the search engine visits the page over time to update its search index. Search engines have a variety of heuristics to decide whether a page delivers content that tries to compromise the browser, scam the user, or is malicious in other ways.

Despite search engines' best efforts, attackers are still able to defeat such approaches, siphoning millions of dollars from search ad networks. Verified accounts are straightforward and cheap to obtain via underground markets [109]. Furthermore, since normal user accounts at Bing, Google, etc. can be converted to advertiser accounts with just additional verification (such as a credit card), there is a constant stream of accounts compromised via phishing [50], host or browser compromise [43], or other means that can be used for the purpose. Finally, attackers use 'cloaking' on the pages they advertise to evade detection by the search engine crawler. Cloaking has traditionally been used to poison search results [123, 124, 125, 135], and attackers have developed many different kinds of cloaking over the years that fraudulent advertisers now also employ. In this chapter, it is these fraudulent advertisers that we characterize on the Bing ad network.

3.3 Sources and definitions

Our analyses focus on a two-year time span in the recent past, with much of our in-depth analysis focusing on a few representative shorter time periods. We chose these windows to be sufficiently far in the past to ensure that most fraudulent actors active during that time have been identified by Bing. Where appropriate, we have verified that more recent data is in line with our analysis. We believe that the trends we report on have not changed significantly during our extended measurement period except where noted.

3.3.1 Datasets

For the purposes of this chapter, we focus on three primary data sources:

- **Customer and ad records:** This dataset contains information on each advertiser (when their account was opened, market, language, home currency, etc.), every ad (title, description, display URL and destination URL), keywords bid on, bid types and maximum amounts.

- **Ad impression and click records:** This dataset contains information on ad impressions and ad clicks. In each case, Bing records ad information (advertiser, ad, keywords, etc.), some basic matching information (why the ad matched the query, how much Bing charged the advertiser, etc.), as well as some basic user and query information (search query, market, etc.). This dataset forms the basis for determining how effective a fraudulent advertiser is relative to other advertisers.

- **Fraud detection records:** This dataset represents actions taken by Bing to shut down fraudulent accounts, generated by both Bing's algorithms and by manual review. It covers the entire lifetime of the accounts, from creation through long-term monitoring.

3.3.2 Fraud under measurement

For the purposes of this chapter, our designation of 'fraudulent' advertisers are those that Bing has shut down according to their own internal policies [70]. This group primarily includes advertisers who attempt to defraud or deceive either Bing (for instance by providing stolen payment credentials) or Bing's users (e.g., by advertising miracle-cure products or implying that the advertiser is affiliated with a person or organization with whom they are not). Each time an advertiser is shut down by Bing, information about that advertiser's identity and/or advertising campaigns may be blacklisted. Conversely, 'non-fraudulent' advertisers are the set of active advertisers that Bing has not (yet) determined to be non-compliant; it does not include the set of advertisers whose accounts have yet to be granted initial approval.

Bing uses a variety of mechanisms to apprehend fraudulent advertisers, a discussion of which is out of scope of this dissertation. Many of these mechanisms, however, involve a manual review of the advertiser account in question. This review helps Bing to avoid accidentally shutting down accounts of legitimate paying customers. In addition, in cases where a customer may be out of line with Bing policy, an individual ad or keyword may be removed or otherwise flagged without shutting down the entire account. Thus, accounts that are entirely shutdown are overwhelmingly fraudulent, with the rate of 'friendly fire' being rather low.

There is some amount of inherent subjectivity when it comes to some policies. For instance, Bing's policy forbids claiming a non-existent affiliation with companies or individuals. But determining when the line has been crossed can be blurry when endorsements are implied rather than stated. On the whole, however, we believe that this definition represents the best ground-truth data available. As with any imperfectly labeled set, our definition may introduce some bias into our analysis. Given that we believe the system minimizes false positives (that is, incorrectly shutting down advertisers), the effects we identify in this work may be slightly under reported.

Our definition also necessarily leaves out other classes of advertisers who are worthy of study. By ignoring accounts that are frozen temporarily, for instance, we may be ignoring a potentially interesting dataset. We performed a manual inspection of advertisers that have been identified by behavioral fraud detection algorithms repeatedly but subsequently allowed to continue advertising to determine if the behavior of these accounts varies from confirmed-fraudulent advertisers. While there are not many of these advertisers, by and large they were either benign or behave similarly to other fraudulent advertisers, and only manage to evade shutdown by way of narrowly avoiding policy violations. We have found no significant sets of advertisers whose behavior meaningfully differs from other advertisers in their vertical. Further, as policies and algorithms adapt and change, advertisers who previously evaded detection tend to be labeled fraudulent—the observations in Section 3.5.2 regarding third-party tech support are a good example of this evolution.

We also necessarily omit advertisers who are fraudulent even by current policy standards, but are wholly undetected by Bing's detection methods. While some amount of undetected fraud is inherent in such analyses, we believe that there are several factors that combine to guard against large swaths of undetected fraudulent activity:

- **Bing accepts manual reporting**: Bing accepts complaints from users regarding illegitimate ads. These reports are investigated. If a fraudulent advertiser was not being detected by Bing's internal mechanisms, a user is likely to complain given sufficient activity.

- **Payment fraud detection is high**: For the portion of fraudulent advertisers who use illegitimate payment mechanisms, fraud is often detectable in the form of chargebacks or other indications from the payment network. Moreover, once an advertiser is detected as fraudulent, they may find their payment instruments blacklisted. This restriction effectively forces advertisers defrauding users into also committing payment instrument fraud, as unless the advertiser has access to a large number of genuine payment instruments, payment fraud is necessary to continue operating within the network.

- **We report on activity in the past**: Experience shows that fraudulent advertisers rarely walk away from working accounts. As a result, it is safe to assume that most fraudulent advertisers with meaningful amounts of activity that have not been detected by Bing will remain active until their detection, but also it is likely that Bing will detect this ongoing activity given sufficient time. We take advantage of this by running our analyses on data that is at least 6 months old, and typically much older. By including the oldest data in our analysis, we permit time for Bing to detect as many fraudulent advertisers active in that time period as possible.

Lastly, our data sources also limit insight into fraudulent actors who are unable to successfully open a Bing advertiser account in the first place due to Bing's immediate detection of potential fraud. Given that these actors, by definition, do not show ads and are not visible to users or other advertisers, we consider them to have negligible impact on the ecosystem.

3.3.3 Subset definitions

To make our analyses more tractable, in many instances we consider subsets of advertisers (both fraudulent and non-fraudulent) to represent the whole. In each case, we choose approximately 10,000 advertisers from the complete pool of advertisers active during the time period.

Fraudulent subsets

We construct four types of fraudulent subsets: a uniformly random selection across all fraudulent advertisers who were alive at any point during the measurement window (labeled 'Fraud'), a uniformly random selection across all fraudulent advertisers whose ads received any clicks during the measurement window ('F with clicks'), and two subsets with weighted probability of inclusion. In the spend-weighted subset ('F spend weight'), fraudulent advertisers are chosen for inclusion with probability proportional to how much money they spend on Bing during the measurement window. The volume subset ('F volume weight') has advertisers chosen with probability proportional to the number of clicks received during the measurement window.

Non-fraudulent subsets

We use a total of seven types of non-fraudulent subsets. Four are defined similarly to their fraudulent counterparts ('Nonfraud', 'NF with clicks', 'NF spend weight', 'NF volume weight')—their selection is designed to represent the non-fraudulent advertisers as a whole, and are used when investigating the effects of fraud on legitimate advertisers. The remaining three subset types are designed to facilitate comparisons between fraudulent and non-fraudulent advertisers of similar ilks and correct for differences in the demographics of fraudulent and non-fraudulent advertisers that would otherwise make behavioral comparison difficult. Each advertiser selected for inclusion is chosen to most closely resemble a corresponding advertiser in the matched subset (that is, chosen to minimize the difference between their corresponding metrics).

'NF spend match' comprises non-fraudulent advertisers chosen to match the fraudulent advertisers in the 'F spend weight' set, where similarity is defined according to amount of money spent. 'NF volume match' corresponds to 'F volume weight' according to click volume. Finally, 'NF rate match' corresponds to a subset wherein non-fraudulent advertisers are chosen to match members of 'F volume weight' according to the rate at which the advertisers receive clicks during measurement. In both cases, this rate is defined as the number of clicks received during the measurement window divided by the period of time that the advertiser could have been generating activity during that window. That period stretches from the later of the start of the measurement window and the account creation, until the earlier of the measurement window ending or the account being frozen (if applicable).

3.4 Scale and scope

We begin by quantifying the scale of the fraudulent advertising problem at Bing. We start our analysis with account registration, as from the point of view of the search ad network, accounts represent the unit of accountability.

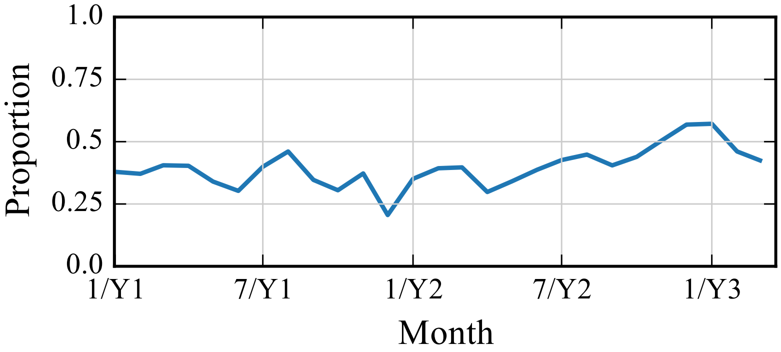


Figure 3.1. Proportion of active advertisers subsequently marked as fraudulent over time, labeled by end of measurement period.

Table 3.1. Top-five countries for fraudulent advertisers, as indicated at account registration. We consider four different subsets of fraudulent accounts.

	#1	#2	#3	#4	#5
all fraud	US	IN	GB	BR	AU
%	50.3	17.2	14.3	2.5	1.8
with clicks	US	IN	GB	BR	CA
%	58.1	14.3	12.3	2.4	1.9
volume weight	US	IN	GB	BR	DE
%	59.5	15.1	8.7	2.6	1.9
spend weight	US	IN	GB	CA	DE
%	60.4	15.1	11.5	1.8	1.7

3.4.1 Account registration

While a single fraudulent actor may register for multiple accounts, an advertiser account is the natural unit of accountability. By this metric, fraudulent advertisers represent a significant challenge for Bing. As shown in Figure 3.1, during the two years we study, generally more than a third—and near the end more than half—of new account registrations each day are eventually discovered to be fraudulent. The overwhelming majority of fraudulent advertisers have languages, currencies and registered home countries suggesting that fraudsters are based in English-speaking countries—primarily the US and India. Table 3.1 shows the top-five countries from four different

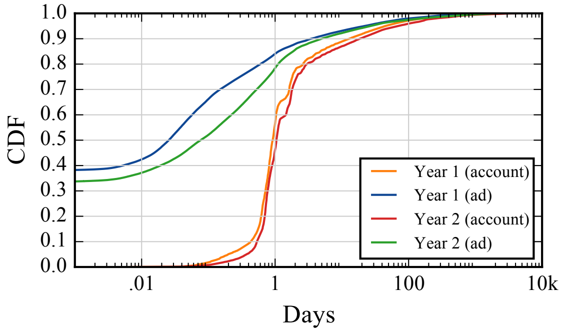


Figure 3.2. Fraudulent account lifetimes, as measured either from account registration or first ad creation. All fraudulent accounts detected as fraud in first and second year of measurement are shown.

populations of fraudulent advertisers, each weighted according to the indicated account factor.

Given the high rate of fraudulent account registration, Bing must be vigilant in identifying and acting upon signs of fraudulent activity, which can occur at almost any stage of the account lifetime. 35% of all account shutdowns, however, occur before the advertiser account is able to display even one ad, with the median fraudulent account surviving less than a day from account creation. Of those accounts that are successful in posting any ads at all, most will be shut down within eight hours of beginning to post advertisements, and 90% of all account shutdowns happen within four days of initial ad posting. Figure 3.2 shows the cumulative density function (CDF) of fraudulent account lifetimes, measured both from account registration and first ad creation. We find that lifetimes are similar in both years of our study.

3.4.2 Advertiser effectiveness

Despite their relatively short lifespans, fraudulent accounts are able to generate a non-trivial amount of traffic on Bing each month. Over the two-year period of our study, traffic regularly averaged tens of millions of clicks, and over ten million USD losses to Microsoft.

Figure 3.3 plots the total amount of billable activity or ‘spend’ and clicks generated each week by the fraudulent accounts present on Bing that week. We break the activity into

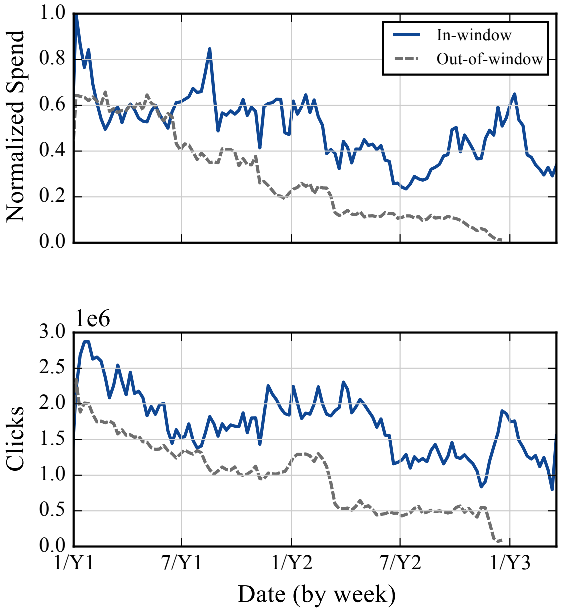


Figure 3.3. Weekly aggregate fraudulent activity over time. 'In-window' indicates the account detected as fraudulent within 90 days of the given date. 'Out-of-window' indicates accounts that were discovered after more than three months. Spend is normalized by maximum value.

two categories based upon account. The 'in-window' line includes the activity from accounts detected as fraudulent within a 90-day rolling window starting from the date of activity. The values have been normalize by the maximum value. We observe that fraudulent activity has nearly halved during the period of study.

In contrast, the 'out-of-window' line accounts for activity that was determined to be fraudulent by the end of our study, but not within 90 days of occurrence. We present this line not because it is an accurate accounting of this fraction; indeed, it cannot be: it decreases and necessarily stops approximately 3 months before the end of the figure, as the number of days, and thus opportunities for shut down, approaches zero. Rather, we show it to suggest that our analyses represent a substantial, but unavoidable, under-reporting of fraudulent activities—potentially by a factor of two or more.

A result of the rapid capture of most fraudulent advertisers is that success is centralized among the top few who dominate the rest. In most time periods, the top 10% of advertisers, as

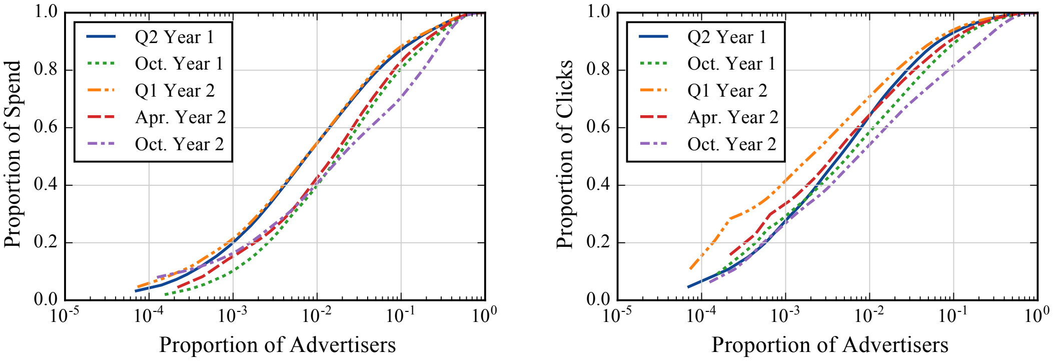


Figure 3.4. Cumulative proportion of total fraudulent spend/clicks per advertiser for five distinct time periods. Advertisers are in decreasing order of spend.

ordered by number of clicks received, collectively account for more than 95% of all fraudulent clicks received. In terms of spend, the situation is similar: the top 10% of advertisers make up 80–90% of spend. Figure 3.4 shows the distributions of spend and clicks across advertisers over several measurement windows.

Click-through rate (CTRs), the probability that a random user will click an advertiser's ad, provides the primary mechanism for demonstrating ad quality and relevance to search queries. Another metric, cost-per-click (CPC), provides the average amount spent by an advertiser to receive a click. Ad performance, as measured by CTR and CPC, heavily influences whether an ad is shown at all, as well as where the ad appears on the page. While one might expect better performance from fraudulent ads over legitimate advertising, click-through rates for fraudulent advertisers tends to be slightly lower than their non-fraudulent counterparts, only being slightly higher for the highest-spending fraudulent advertisers. Put differently, most fraudulent ads are less alluring to users than legitimate ads, except for the most successful few among them. The fraudulent advertisers that spend the most do so in part because they pay more per click than almost everyone else, existing almost entirely in the upper end of the distribution for cost per click, with CPCs regularly in the several tens of dollars. Many of these advertisers sell products costing more than $100, perhaps permitting such high click costs.

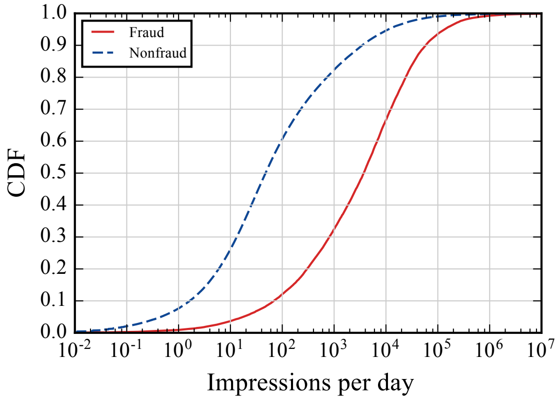


Figure 3.5. Impression rate witnessed during Year 1 Q2.

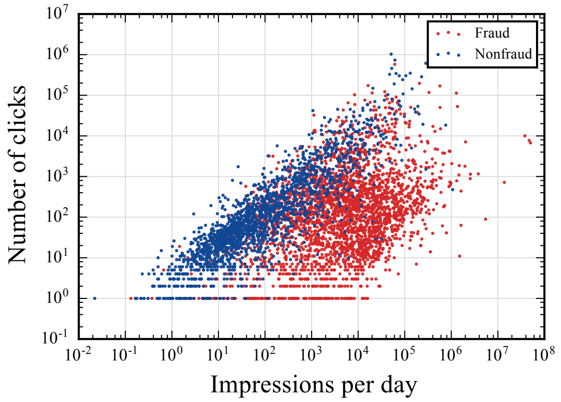


Figure 3.6. Impression rate vs. number of clicks received in Year 1 Q2.

3.5 Advertiser behavior

Fraudulent advertisers work hard to attract as many clicks to their ads as possible before they are detected. In doing so, they must be careful when choosing their ads and keywords. Effectively-targeted ads will increase the likelihood that a user will click on the ad, which causes Bing to show the ad more often. Conversely, however, having many ads and keywords that make clear what a fraudulent advertiser is offering provides greater surface area for Bing to detect dubious activity.

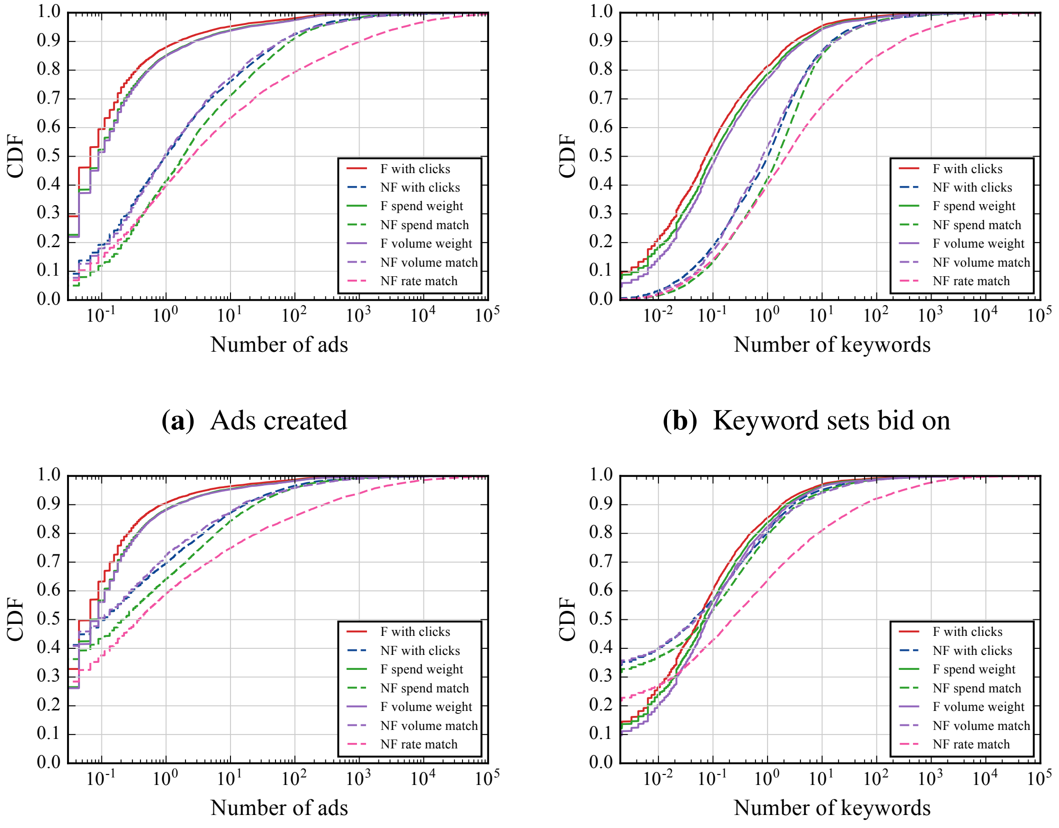
3.5.1 Rates

Figure 3.5 shows the distribution of impression rates over a representative measurement window. As one might expect, fraudsters show ads more rapidly than their legitimate counterparts. Several reasons contribute to this phenomenon; in addition to fraudsters attempting to gain as much traffic as possible prior to detection, illegitimate advertisers may have no intention of paying their bill to Bing (e.g. payment instrument fraud), and may operate in affiliate programs that pay out per-click, and are not discerning about their traffic. Click and spend rate distributions (not shown) have similar shifts with respect to non-fraudulent accounts.

The differences in rate between fraudulent and non-fraudulent disappear when one focuses on prolific advertisers, however. Figure 3.6 shows the number of clicks received as a function of impression rate during the Year 1 Q2 measurement window. The important observation is that, while there is noticeable separation between fraudulent and non-fraudulent advertisers at lower click volumes, higher valued non-fraudulent advertisers are substantially more likely to have rates roughly equaling the performance of similarly-prolific fraudulent accounts. As a result, while rate checks are effective for detecting many low-volume fraudulent users, the most successful fraudulent users blend in with their non-fraudulent counterparts.

3.5.2 Targeting

Bing determines how often to show ads in part by the performance of the advertisement when shown [8]. As a result, targeting an ad too broadly results in lower relevance to the search queries, which often hurts performance. We find that successful fraudulent advertisers target their audiences similarly to legitimate advertisers [119] (e.g. advertisers that bid on terms such as ‘YouTube’, ‘videos’ or ‘news’ offer ads for sites designed to look like video or news sites), but with the added challenge of evading blacklisted keywords or ad copy that is likely to trigger filters. Moreover, because adding ads and keywords only increases the ways in which the advertiser can be identified (both for current accounts and in the future), fraudulent advertisers are pressured to



(a) Ads created

(b) Keyword sets bid on

(c) Ads modified

(d) Keyword sets modified

Figure 3.7. The distribution of ads and keyword sets added or modified as a function of advertiser type from Year 1 Q2. Normalized by median number of creations from ‘NF with clicks’.

keep the number of ads and keywords low to reduce the probability of detection.

Figure 3.7 shows the distribution of the number of ads and number of keywords created or modified per account. The total numbers of ads created and keywords on which fraudulent advertisers bid are each more than an order-of-magnitude less than their non-fraudulent counterparts, with the differences greatest when compared to advertisers posting at similar rates to fraudulent advertisers. This is true even though fraudulent advertisers appear to maintain their ads and keyword sets at rates similar to other advertisers. This effect is even more pronounced when compared against non-fraudulent advertisers with similar rates of posting ads, consistent with fraudulent advertisers pushing to receive as much traffic per-ad as possible.

Table 3.2. Example ads from selected popular categories.

Category	**Ad Title** Ad Body
`techsupport`	**Install Printer** Call Our Helpline Number. Online Printer Support By Experts.
`downloads`	**Discord Free Download** Latest 2017 Version. 100% Free! Instantly Download Discord Now!
`luxury`	**75% Off C0ACH Factory Outlet** Enjoy 75% Off & High Quality C0ACH Bags & Purses. Winter Sale Limited Time...
`wrinkles`	**Best Anti Wrinkle Cream** Premium Skin Care Product! Removes Wrinkles in Weeks! Clinically Proven
`impersonation`	**Target - Online Shopping** Store Hours & Locations. Go To Target.com Online Shopping Now.

Verticals

Fraudulent ads span a wide array of topics. The significant majority of fraudulent advertisers, however, appear to participate in pay-per-click or pay-per-action affiliate programs. These programs are a popular choice among fraudulent advertisers because they are quick and easy to join and require little sophistication to begin monetizing. Many advertisers involved with the easier-to-join programs advertise for several programs simultaneously, using one advertising account across their campaigns. The highest spending accounts, however, tend to be more focused on fewer, more specialized and lucrative verticals.

The top categories in terms of clicks are typically sites dedicated to offering downloads of popular software. These range from heavily ad-laden sites providing unmodified copies of open source software to sites spreading malware bundled with cracked versions of commercial software. A common strategy is offering open source software bundled with ad-injecting installers.

Figure 3.8 shows some of the most popular verticals targeted among the most prolific advertisers (in terms of spend) periodically from year 2. Fraudulent advertisers target hundreds of distinct verticals; these verticals were chosen for their prevalence in at least one month. Table 3.2 provides sample ads for some prominent categories.

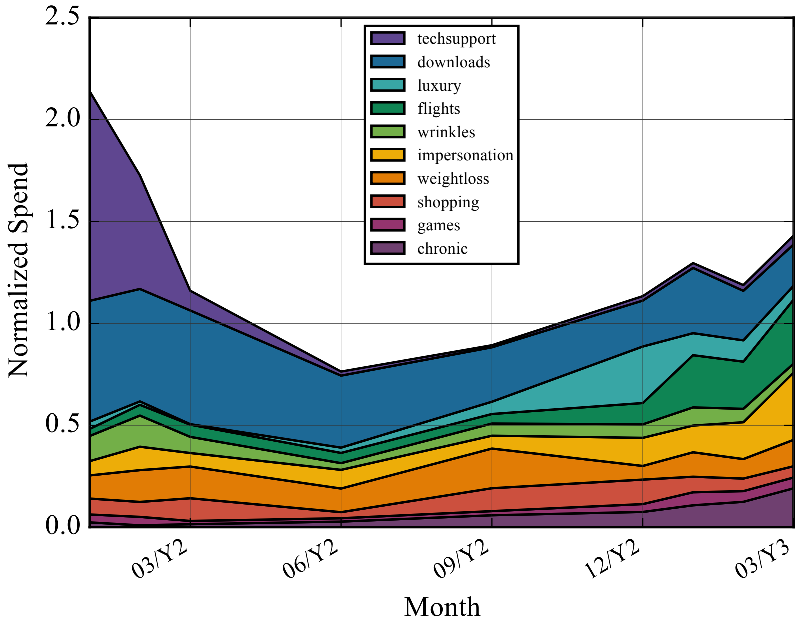


Figure 3.8. Primary verticals targeted by fraudulent advertisers, manually labeled from ad copy on all advertisers with more than $2000 spend in a month. Amounts are aggregated per-month with data points every three months, with monthly frequency at edges of date range. Data normalized by same value as in Figure 3.3.

The first quarter of the measurement period offers a particularly interesting example of targeted intervention. During that quarter, ‘`techsupport`’ was by far the vertical with the most fraudulent spend. In this model, advertisers offer technical support for business accounting software, printers, routers, antivirus products or other technology, and work by encouraging users to call a phone number, where users pay hundreds of dollars for a single support call.

This vertical was dominated by a few especially prolific advertisers; in the first quarter of year 2, just fourteen advertisers survived long enough to spend more than $100,000, and 134 spent more than $10,000. Of these, 11 of the 14 and 81 of the 134 were selling third-party tech support. In contrast, no other category received more than one advertiser in the top 14, and the second-place holders (a three-way tie) made up just 7 of the top 134.

The precipitous drop-off corresponds to a policy change in which Bing prohibited the marketing of third-party technical support services [71]. Prior to this change, Bing only prohibited advertisers from inaccurately suggesting an affiliation with other companies.

Phishing

One vertical deserves special mention, given its recent prominence in the popular press: phishing. By the numbers, phishing-type scams historically make up only a small percentage of the total fraudulent advertising activity on Bing. When we manually inspected all advertisers in Year 2 Q1 who managed to spend more than $10,000, only one account was used for phishing; most phishing accounts are shut down quickly. Similarly, manual inspection of the most prolific advertisers at various points throughout the measurement period yielded few instances of traditional credential phishing, though there was a noticeable uptick towards the end of our measurement period.

We suspect phishing is somewhat less prevalent than one might expect due to aggressively targeted machine learning and blacklisting. Like all blacklisting, Bing's blacklisting is most effective for high-value targets (like banks) with unique names, as the fraudster must name the institution in order to impersonate it. The blacklisting is less effective, however, in a few cases: when legitimate advertisers may purchase ads targeting the site (e.g. an ad may point to a user's YouTube channel), when the bare company name aliases with a term that isn't easy to blacklist, and where the institution is too small to have yet been added to the blacklist.

Indeed, much of the phishing we observe during the period of study targets small financial institutions and services in non-English-speaking markets where the blacklist is not as developed, and against services whose names cannot be effectively blacklisted. Fraudsters targeting these small institutions can be effective in evading detection for a time, but as blacklists grow, the fraudsters may run out of sufficiently-attractive targets. Impersonation of non-blacklistable companies does pose an ongoing problem.

A superset of phishing that we see commonly is impersonation. Impersonation encompasses any time a site attempts to mimic a larger site to attract clicks. Many sites in this category are not attempting to get a user to reveal private information, but are attempting to piggyback

Table 3.3. Country distribution of fraudulent clicks from a typical sample day. '% of Country' indicates the portion of clicks in that country that are to fraudulent accounts.

Country	% of Fraud	% of Country
US	61%	$< 2\%$
BR	10%	$< 6\%$
DE	10%	$< 3\%$
CA	5%	$< 2\%$
GB	3%	$< 1\%$
FR	3%	$< 1\%$
IN	2%	$< 2\%$
MX	2%	$< 1\%$
AU	1%	$< 2\%$
SE	1%	$< 2\%$

on the reputation of the more prominent sites. Streaming sites, rival search engines, large retail establishments, and social networks are all popular impersonation targets. Visitors to these sites may be greeted by any number of scams and low quality advertising.

Geography

Table 3.3 shows the countries receiving the most fraudulent clicks. The US is by far the most attractive target, but the country with the greatest proportion of fraudulent traffic is Brazil. Interestingly, the UK and France are significantly cleaner overall than other major Western nations. An equivalent breakdown by language yields a very similar result. These results mirror the fraudulent accounts' stated home countries/languages at registration, and by and large, accounts target ads in their own country.

This distribution is likely due to a combination of factors: Bing's differing market share across different markets, local regulation, market forces, relative tuning of detection algorithms and language spoken of analysts, as well as cultural and other factors likely all play a role. We were unable to locate any clear correlations between fraudulent click behavior and country, but we speculate that the size and relative wealth of the markets in each language accounts for much of this distribution.

Blacklist evasion

Bing maintains blacklists of words and patterns (such as phone numbers and some trademarks) that are not permitted in ad text or keywords. By and large, successful fraudulent advertisers rely on phrasing in ads and keywords bid on that are not easily blacklisted outright: e.g., terms like ‘news’, ‘download’ or ‘skin care’ are used by legitimate and illegitimate advertisers alike.

Occasionally, fraudulent advertisers are motivated to circumvent these blacklists, and we see every combination of words using look-alike characters (e.g. ‘O’ for ‘0’, diacritics). A typical example is the prohibition of phone numbers in ads (since users calling a phone number circumvent Bing’s billing mechanisms by not requiring a click). Advertisers often try to avoid detection in these cases by injecting text into parts of phone numbers or presenting numbers in odd formats (e.g. ‘CALL 1-800 (USA) 555 1000’).

Bing also maintains a fairly aggressive blacklist of domains used in fraudulent activities. As a result, the URLs witnessed in fraudulent ads (either as the URL displayed or as the destination URL after a click is received) are typically unique to that account. The most common domains that are shared between fraudulent advertisers are third-party services which also serve non-fraudulent traffic, including URL shortening services (e.g. `bit.ly`) and affiliate programs (e.g. MaxBounty).

The most clicked-on domains are nearly universally unique to individual advertisers (with a few affiliate programs added in). Fraudulent advertisers, however, often use more than one URL. While 74% of fraudulent advertisers use a single domain in their advertisements, and 96% use 3 or fewer, most accounts are shutdown so quickly that these figures are misleading. Predicating on accounts that have multiple ads moves the mean case to 3 domains, with the 90th percentile having nearly 20.

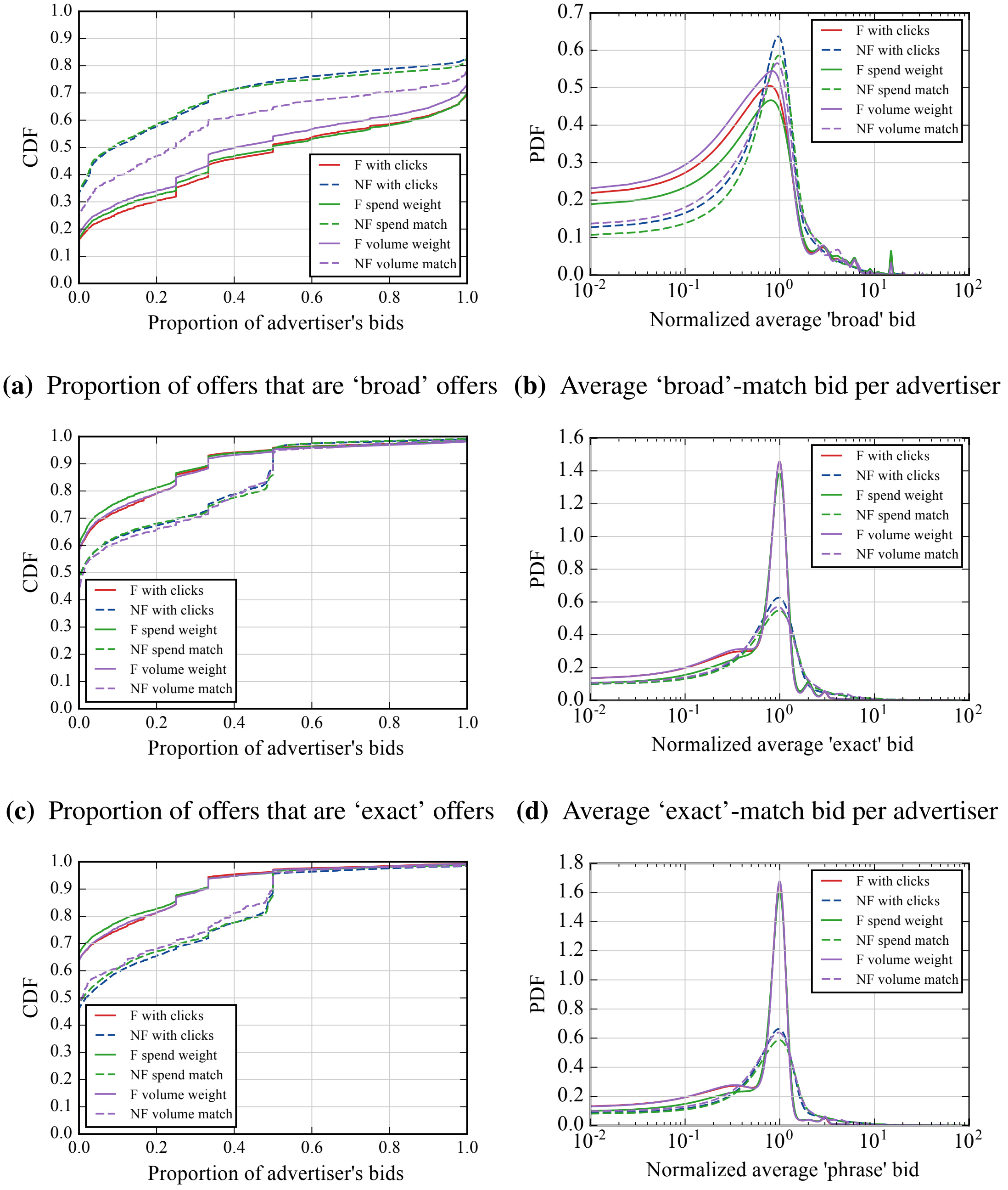
3.5.3 Bidding style

In Bing's ad platform, advertisers choose a matching method alongside choosing keywords to bid on. During a search, Bing assembles a list of ads that are eligible to be shown using this match method (or 'type') to determine whether the keywords match the search query. Bing supports three distinct types of matches that pair a search query with a given keyword phrase.

An 'exact' match occurs when the keywords chosen by the advertiser occur as the exact search query, with no changes to ordering or additional words. A 'phrase' match occurs when the keywords occur in the right order, but optionally with additional words preceding or following the keywords. Finally, a 'broad' match occurs when the keywords, or any keywords that Bing determines to be similar, occur in the query, regardless of order or existence of other words in the query [72]. Across all match types, Bing normalizes for misspellings, plurals, acronyms and other minor grammatical variations.

For many fraudulent advertisers, the strongest incentive is to ensure that their ad is seen as many times as possible before their deception is uncovered by the advertising network—precise targeting is less important, so long as users still engage. As such, fraudulent advertisers skew away from precision matching, preferring broader matches. While a quarter of legitimate advertisers use exact matches at least a third of the time, only about 10% of fraudulent actors use exact matches that frequently. 60% of fraudulent advertisers do not have even a single exact bid (compared to about 50% of legitimate advertisers). Proportions are similar for phrase matching. In contrast, legitimate advertisers use broad matching less than 10% of the time, while the median fraudulent advertiser uses phrase matching in half of cases. See Figures 3.9(a), 3.9(c), and 3.9(e) for the full distributions.

Table 3.4 shows the distribution of clicks received according to the matching method employed by fraudulent advertisers. As expected from their targeting strategies, the proportion of clicks from exact matches is lower than in the non-fraudulent population. Interestingly, however, phrase matching is considerably over-represented compared to the non-fraudulent population.



(a) Proportion of offers that are ‘broad’ offers **(b)** Average ‘broad’-match bid per advertiser

(c) Proportion of offers that are ‘exact’ offers **(d)** Average ‘exact’-match bid per advertiser

(e) Proportion of offers that are ‘phrase’ offers **(f)** Average ‘phrase’-match bid per advertiser

Figure 3.9. Advertisers’ use of ad match types in Year 1 Q2. Bid values normalized by Bing’s US default maximum bid amount.

Table 3.4. Match type distribution of clicks received on fraudulent ads on a typical sample day, as compared to non-fraudulent advertisers.

Type	% of Fraud	% of Type	Non-fraudulent %
Exact	61.62%	0.90%	67.88%
Phrase	31.05%	1.32%	23.32%
Broad	7.33%	0.83%	8.80%

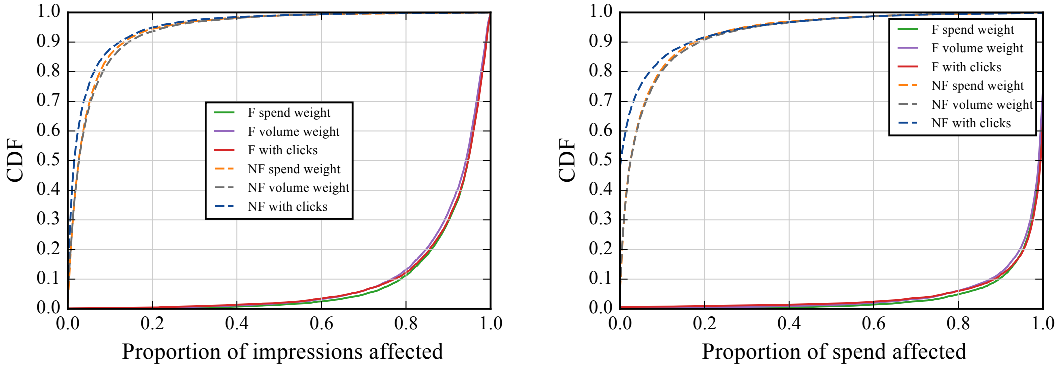
Advertisers may also specify a different maximum bid for each match type and keyword combination. Contrary to our initial expectations, bidding prices among fraudulent advertisers are not significantly different than non-fraudulent advertisers (with the exception of some quickly-caught advertisers). Across all bid types, for both fraudulent and non-fraudulent advertisers, the median maximum bid is the same as the default amount in US markets. The most notable trend in bid pricing for fraudulent advertisers is that only 17% of such advertisers are bidding more than the default on both exact- and phrase-type matches, while non-fraudulent advertisers are roughly double that. See Figures 3.9(b), 3.9(d), and 3.9(f) for the full distributions.

3.6 The Impact of Fraud

The final question we explore is the extent to which fraudulent advertisers impact other advertisers competing in ad auctions. One might expect that because fraudulent advertisers are often not spending their own money, they would be more profligate in their bids. As a result, competing advertisers may lose auctions more often, or potentially have to pay more for ads.

We consider advertisers to be competing with fraud when their ads are shown alongside ads from fraudulent advertisers. We ignore ads that compete in the auction, but are not shown. We believe this to be a safe simplifying assumption as many ads that participate in auction are still shown to the user (at a lower ranking), and in many cases, no ad is shown. Further, the impact of losing bidders is limited to the prices offered by the ads shown at the lowest ad positions.

In most markets, well less than 2% of search queries result in a fraudulent ad being shown,



(a) Impressions

(b) Revenue

Figure 3.10. Proportion of impressions and revenue affected by fraudulent competition, per advertiser from Year 1 Q2.

but the effects are not uniformly distributed. Most verticals have no overlap with fraudulent advertising at all, so advertisers within these verticals are essentially unaffected by fraudulent advertisers. For most advertisers, then, fraudulent advertising has little impact. While there are a few markets with much larger rates of fraud, the percentage of fraud tops out at about one in twenty ads shown (see Table 3.3).

3.6.1 Frequency of competition

Figure 3.10(a) shows the distributions of the proportion of an advertiser's ad impressions that compete with fraudulent ads. The median legitimate advertiser will have less than 0.6% of their ad impressions shown with a fraudulent ad, and the 95th percentile legitimate advertiser has less than 20% of their ads shown alongside fraudulent ads. Even in samples of advertisers with substantial keyword overlap with the most prolific fraudulent advertisers (not shown), less than 2% of the advertiser's impressions were shown alongside a fraudulent ad in the median case. And when non-fraudulent advertisers do encounter fraudulent competition, they are almost always faced with only a single fraudulent ad.

Fraudulent advertisers often focus on niche areas, and there is substantial competition among advertisers in those verticals. Figure 3.10(a) also shows the equivalent distributions for fraudulent advertisers competing with other fraudsters. For the median fraudulent advertiser, more than 90% of their ads will be shown adjacent to a different fraudulent advertiser's ad, and the 95th percentile fraudulent advertiser has nearly all of their impressions in competition with other fraudulent advertisers. Further, in the significant majority of cases, fraudulent advertisers are competing with more than one fraudulent ad shown beside their own (not shown).

The distribution of proportion of spend affected, shown in Figure 3.10(b), is similar to the impression distribution with one major difference: a disproportionate amount of the money spent by fraudulent advertisers occurs during heavy competition with other fraudulent advertisers. That is, fraudulent advertisers waste most of their money competing with each other. About 99% of spend is affected by competition from other fraudulent advertisers for the fraudster, compared with just 92% of impressions.

3.6.2 Impact of competition

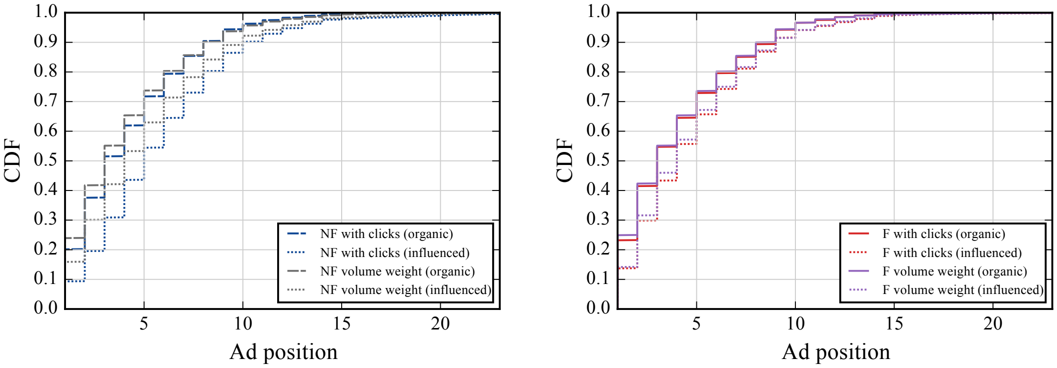
While legitimate advertisers do not frequently compete with fraudulent advertisers, when they do, the competition does negatively impact legitimate advertisers—especially when the fraudulent advertisers are operating at volume.

Impact on ad position

On a search engine results page, ads can be displayed along the top of the page (the 'mainline', above traditional search results) or along the right edge of the page ('sidebar'), with the mainline traditionally receiving more clicks than the sidebar, and higher positions in the page typically providing more traffic. In this way, we define 'ad position' as the rank of an ad in the list of ads shown on the page, from the top of the mainline down to the bottom of the sidebar.[1] For ease of comparison, we only examine the effects on ads that were displayed on the first page of results (though fraudulent competing ads may appear on subsequent pages).

Figure 3.11(a) shows the impact on ad position of non-fraudulent advertisers competing

[1] While the '1' slot is always the most valuable position, the number of ads in the mainline and sidebar is dynamic. A particular ad position does not correspond to a particular slot on the page.



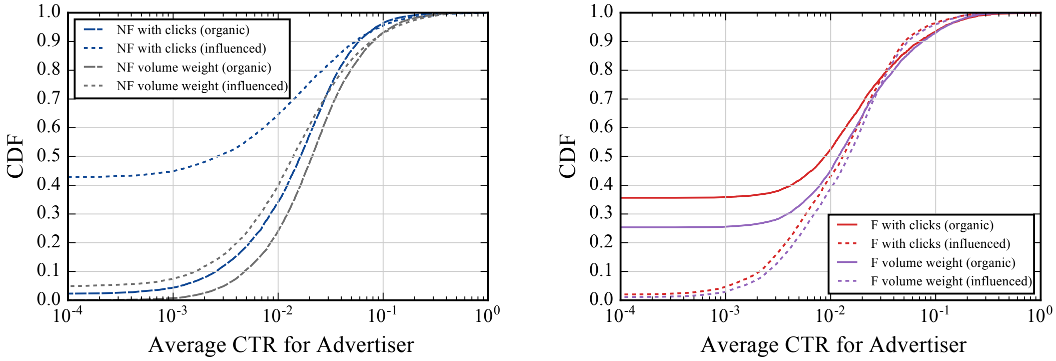
(a) Effects for non-fraudulent advertisers

(b) Effects for fraudulent advertisers

Figure 3.11. Effects of fraudulent competition on ad position for fraudulent and non-fraudulent advertisers from Year 1 Q2

with fraudulent advertisers for two sets of non-fraudulent advertisers. Other non-fraudulent advertiser subsets show similar effects. Both subsets are from a typical day in our second-quarter Year-1 sample period.

While fraudulent advertisers are only about 5% more likely to achieve the top ad position as compared to their non-fraudulent counterparts absent competition from other fraudulent advertisers, non-fraudulent advertisers are considerably less likely to achieve the top ad position when competing with fraudulent advertisers. The median non-fraudulent advertiser is likely to achieve the top position about 20% of the time without interference (labeled 'organic'); this probability drops to about 10% when competing with fraud (labeled 'influenced'), and similar drops are present throughout the distribution. Put in other terms, competing with a fraudulent advertiser typically costs the legitimate ad about one position. Figure 3.11(b) shows the same distributions for fraudulent advertisers, with similar impacts. When fraudulent advertisers compete with each other, however, their probability for reaching the top position drops by about 10%.



(a) Non-fraudulent advertisers

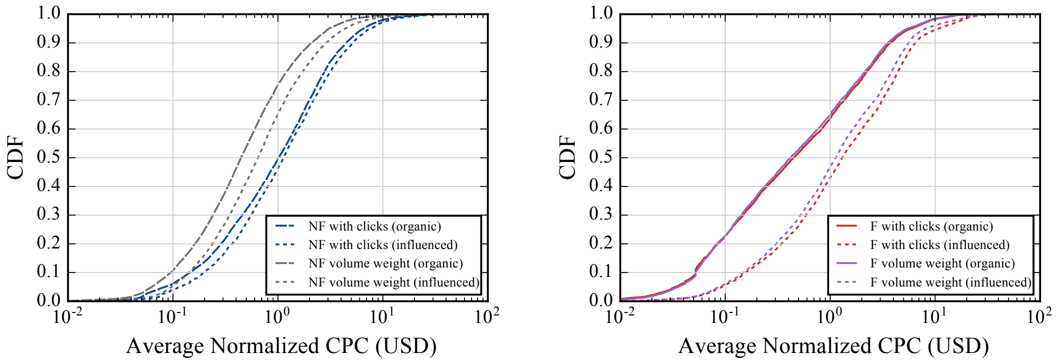
(b) Fraudulent advertisers

Figure 3.12. Effects of non-self fraudulent influence on CTR for during Year 1 Q2 in dubious verticals

Impact on CTR

Competing with fraud has a devastating impact on click through rates (CTRs) among advertisers with lower than median performance. Figure 3.12(a) compares the CTRs for non-fraudulent advertisers when competing against other non-fraudulent advertisers (‘organic’), and when competing against fraudulent advertisers (‘influenced’). While few non-fraudulent advertisers have close-to-zero CTRs when competing on a level playing field, the proportion jumps to 50% when competing with fraud. Even among high-volume non-fraudulent advertisers, CTRs drop by a factor of two in the median case.

A similar, but smaller effect occurs for fraudulent advertisers competing amongst themselves. Figure 3.12(b) shows similar CTR distributions for fraudulent advertisers, with and without competition with fraud. Without competition, only a few percent of fraudulent advertisers have near-zero CTRs, but this value jumps to nearly a third with competition. The median case, though, does not experience nearly as significant of a change. Fraudulent advertisers are accustomed to working in a high-fraud-competition environment.



(a) Non-fraudulent advertisers

(b) Fraudulent advertisers

Figure 3.13. Effects on average CPC per advertiser for advertisers in dubious verticals. Normalized by median CPC for 'NF with clicks (organic)'

Impact on CPC

While competition with fraud results in a significant increase in the cost per click (CPC) of legitimate ads across the board for the dubious verticals where they compete, this effect is unevenly distributed. Figure 3.13(a) compares the CPC distributions for non-fraudulent advertisers with and without competition with fraudulent advertisers, and Figure 3.13(b) shows the same CPC distributions for fraudulent advertisers. High-volume advertisers in these dubious verticals see increases in CPC around 30% in the median case, while randomly chosen advertisers see impacts less than 5%. Fraudulent advertisers bear an even greater increase. For fraudulent advertisers, CPC increases by around a factor of two when competing with fraud across all subsets of fraudulent advertisers.

3.7 Discussion

Throughout this chapter, we have explored search advertiser fraud on Bing's search engine platform, quantifying the scale of fraud, the dynamics of being a fraudulent advertiser, and how fraudulent advertisers impact other advertisers in the ecosystem. In this section, we speculate on the implications of our findings for Bing, other search engines, and the advertising ecosystem more broadly.

At a high level, Bing's fraud detection strategies are certainly necessary (e.g. up to a half of new account registrations are fraudulent) and also are effective—fraudulent advertisers who do succeed in evading detection sufficiently long to see non-trivial volume have to operate under considerable constraints and have been relegated to niche aspects of the advertising keyword space. Despite this, fraud is still a significant concern, costing millions of dollars a month, with adaptable adversaries constantly probing defenses. So what can Bing, and potentially other mature ad networks, do to further undermine fraudulent advertisers?

Given the mature state of Bing's defenses, new anomaly detection strategies are likely to have diminishing returns. Though many fraudulent advertisers are quickly detected, those that remain have chosen their behavior carefully to look similar to existing non-fraudulent advertisers. For instance, while certain keywords clearly indicate that a market segment is high risk (and thus perhaps warrants greater scrutiny), it is not the case that successful fraudulent advertisers have keyword or ad copy choices that are sufficiently different from all non-fraudulent advertisers. In terms of bidding behavior, most of the fraudulent advertisers simply look average with respect to bidding types. Where fraudulent advertisers have higher distributions than their non-fraudulent counterparts (for instance in the proportion of bids that are for a broad match), thresholding is effective to detect some fraud, but the spread among fraudulent advertisers is wide enough to be inconclusive.

The ad strategies employed by the prolific fraudulent advertisers are diverse. In some cases, we see advertisers running one or two campaigns at a time, discontinuing old campaigns before starting new ones; in others, we see advertisers constantly adding new ads, allowing the old campaigns to continue uninterrupted. The most prolific fraudulent advertisers even reliably pay their substantial bills over months or years, indicating that it is unlikely that they are using stolen payment instruments. In effect, Bing's defenses over long time periods have coaxed fraudulent advertisers into behaving similarly to legitimate advertisers, precisely to evade anomalous detection.

At this stage of the ad network ecosystem, the most dramatic impacts that Bing (and perhaps other ad networks) can make are by providing targeted and enforced policy changes aimed at the most prevalent verticals targeted by fraud. Fraudsters must target their advertisements fairly explicitly in order to achieve substantial traffic from the platform, and fraudsters target the verticals they do not purely by convenience, but by necessity. Eliminating these verticals from the ecosystem will challenge these fraudsters into new, likely less profitable, verticals.

Though we can only speculate on the fraudulent ecosystem experienced in other advertising networks, little of the high-level behavior we have described throughout this chapter is likely to be unique to Bing. Bing's policy change to explicitly prevent advertising of third-party support services had the single most dramatic effect on fraudulent advertiser behavior that we witnessed over two years. Similar policy changes in the future (e.g. on misleading celebrity branding) are likely to continue to be the most effective instruments of fraud prevention as it requires fraudsters to completely retool and change what they sell, rather than simply tweaking around the edges to avoid detection.

Acknowledgments

This chapter, in full, is a reprint of the material as it appears in Proceedings of the ACM Internet Measurement Conference, 2017. DeBlasio, Joe; Guha, Saikat; Voelker, Geoffrey M.; Snoeren, Alex C., The dissertation author was the primary investigator and author of this paper.

Chapter 4

Identifying malicious VPN providers

In the two preceding chapters, we explored two different areas where malicious actors attacked or abused web services, but what about when the service itself is malicious? In this chapter, we explore precisely that case by examining the ecosystem of commercial VPN providers. In particular, we investigate a large number of providers looking for evidence of traffic manipulation, traffic monitoring, or fraudulent claims about their service offerings. While we find evidence of a variety of malicious behaviors, we detect our largest findings by leveraging this dissertation's key insight: malicious actors optimize for making money, and not for perfect deception. By observing that VPN providers are incentivized to provide many geographical vantage points, but that actually having servers in many countries is expensive, we develop a test that identifies several providers defrauding their users by lying about their server locations.

4.1 Introduction

Virtual private network (VPN) providers have positioned themselves to play an integral role in users' attempts to secure their Internet freedom amid concerns of privacy, security and censorship. Likely spurred by the large number of media articles recommending their use [133, 134], VPN usage has grown dramatically in recent years. According to a recent market research report [104], commercial VPNs are currently a 15-billion dollar industry expected to grow 20% by 2022. Originally developed as a technology to privately send and receive data across public networks, VPNs are now marketed broadly as a privacy-preserving technology that allows Internet users to obscure their personal information and web browsing history from third parties such as Internet service providers (ISPs) and governments.

VPNs are widely used not only to preserve privacy and evade Internet censorship, but also to access content that may be unavailable in a user's home country due to copyright or licensing agreements. Compared to sophisticated tools like Tor [111], commercial VPN services purport to provide individuals a faster and easier solution to access blocked content and achieve their privacy goals [60, 87, 113]. Commercial VPN services are often preferred by average users over services like Tor due to improved performance and perceptions of enhanced usability.

Unfortunately, while many VPN services claim to operate robust and secure infrastructure and ensure users' privacy by not logging data, the VPN ecosystem is opaque and their claims are difficult to verify. Outside of the mobile space [51, 137], there is little independent research to systematically audit these security and privacy claims, and few practical tools for users to protect themselves. As a first step toward providing users with a rigorous, third-party evaluation of VPN service claims, we develop a test suite to do just that and apply it to a diverse set of 63 VPN providers. We seek to not only highlight the issues surrounding the lack of transparency in the commercial VPN ecosystem but also provide an active-measurement methodology that others can use to evaluate and audit arbitrary VPN services.

Our work finds evidence that at least 10% of VPNs are intercepting user traffic. This number is surely a lower-bound, as VPN operators are well situated to passively monitor user traffic in ways that are difficult or impossible to detect as an end user. We also observe similar amounts of fraud from VPN providers making false claims about their infrastructure. This deception is motivated by VPN providers needing a constant feed of new subscriptions to survive and grow. Attracting new customers requires standing out in a crowded market, and one of the few ways that providers can differentiate themselves is by offering promises of geographical diversity—VPN providers typically advertise that their service can make user traffic route through, and appear to originate from, many distinct vantage points across the globe.

While advertising this capability is easy, actually providing service from potentially hundreds of distinct countries is expensive and complicated. Further, most customers have no practical way of verifying the location of a vantage point aside from using third-party geo-location services. This lack of verification permits disreputable VPN providers that wish to save money by placing vantage points in more convenient locations, and then attempting to deceive geo-location services to mask their vantage points' true locations.

We develop a methodology to identify inconsistencies in purported vantage point locations by relying on the packet round-trip time to a variety of hosts with known locations. We were able to detect that around 10% of VPNs fail to deliver on their promises of geographical diversity, and show that the practice of 'virtualizing' vantage points—i.e. masking the true location of VPN end points—is far more prevalent than VPN services let on. In the most extreme case, we find a VPN provider claiming vantage points in more than 190 countries yet hosting servers in what appears to be fewer than 10 distinct data centers.

4.2 Background

VPN users have been driven to VPN services out of fear of surveillance and/or censorship from ISPs, governments, or others. Some of this fear is well founded. Even in the United States, lawmakers used the Congressional Review Act (CRA) in March of 2017 to repeal privacy rules developed by the Federal Communications Commission (FCC) [133, 120]. The rules would have required ISPs to seek permission from customers for collecting and sharing sensitive personal information such as Internet browsing history. In response to the repeal, some privacy advocates have pointed to VPNs as a viable way to regain control over users' private information.

VPN services advertise the ability for users to access content through services such as Hulu and Netflix that limit access to some content based on user location. Leading VPN providers must often work to continually evade blocking of their service from content providers [121]. VPN services also advertise themselves as a means to download copyrighted material without users needing to fear for legal repercussions, with the theory being that their traffic will originate from countries with lax copyright enforcement, and providers offering minimal logging will be unable to discern who is sharing files even if they wanted to.

While users flock to commercial VPN services, the effectiveness and security guarantees of most providers remains unclear. Users must largely take VPN providers at their word, and there is a dearth of independent user-friendly transparency tools to evaluate providers' claims. A few VPN providers have attempted to provide some assurances to their users—Tunnelbear released a third-party security audit in 2017 [115], and several providers offer tools to detect DNS and/or IPv6 leakage [88]—but these tools are limited and few. Not only is there a lack of widespread positive evidence of security, there is substantial anecdotal evidence to the contrary. In one recent instance, the Center for Democracy and Technology (CDT) filed a complaint against HotSpot Shield VPN [13], an Android VPN service, that was actively redirecting user traffic to partner websites. In another recent instance, IPVanish provided detailed activity logs to the Department of Homeland Security despite having a stated policy of not maintaining such logging [112].

Fierce competition between providers combined with the lack of objective metrics to evaluate them in an unregulated market leads many VPN providers to attract clients through deception. Several top-ranked VPN review websites are currently supported by affiliate marketing and services. With strong economic incentives to funnel users to particular VPNs, it is hard to be confident in the impartiality of the review websites. One popular VPN review site, vpnMentor [122], lists over 250 VPN services with none scoring below a 4 out of 5. These services funnel users to services that provide the best payoff to the affiliate site, rather than to the best provider. Taken together, these factors suggest a strong need for comprehensive and transparent evaluations of commercial VPNs.

Table 4.1. VPN review websites crawled and their use of affiliate marketing

Website	Affiliate Based Link
360topreviews.com	✓
bbestvpn.com	✓
best.offers.com	✓
bestvpn4u.com	✓
freedomhacker.net	✓
ign.com	✓
pcmag.com	✓
pcworld.com	✓
reddit.com	✗
securethoughts.com	✓
techsupportalert.com	✓
thatoneprivacysite.net	✗
tomsguide.com	✓
top10fastvpns.com	✓
torrentfreak.com	✓
trustedreviews.com	✓
vpnfan.com	✓
vpnmentor.com	✓
vpnsrus.com	✓
vpnservice.reviews	✓

4.3 Methodology

To get a comprehensive view of VPN providers active on the internet, we started by creating a list of VPN services. While it is not possible to enumerate every VPN service provider available online, we attempted to be as thorough as possible. To create this list, we combined results from several sources:

- **Review sites:** To imitate how a user might find VPN providers, we started with a simple web search. We examined the first 50 Google search results found using the keywords *"top VPN services"*. These results included both links to individual VPN providers as well as VPN review sites listing additional VPN providers. We extracted all VPNs listed, both in the initial search results and from those review sites.

Table 4.2. Number of VPNs extracted from each source of providers. Note that there is substantial overlap between sources.

VPN Selection Category	# of VPNs
Popular Services (from review websites)	74
Reddit Crawl	31
Personal Recommendations	13
“The One Privacy Site”	78
“vpnMentor”	156
Total Selected	200

- **Reddit crawl:** Many users may rely on recommendations or may not trust a simple web search. We augmented our list with VPNs found by crawling Reddit’s subreddit for VPNs [96]. This active forum sees individuals seeking out recommendations and discussion of issues related to VPN setup and configuration. In July 2017, we extracted all VPN recommendations from two main ‘mega-threads’ about VPN recommendations, as well as VPNs mentioned in regular posts from the prior three months.

- **Personal recommendations:** To ensure we included VPNs common in censored countries, we reached out on the Open Technology Fund [90] mailing list requesting recommendations for services known to be regionally popular in countries with censorship.

Table 4.2 lists the number of VPNs that we collected from each source. As these individual lists overlap, in total, we collected 200 unique commercial VPN services.

4.3.1 VPN selection

Testing VPN services is a manual and time-intensive task. For each service, you must first create and verify an account, likely pay money for any service, install the software, and verify that it works. For VPN services with custom software (rather than using a standard VPN protocol), manual intervention is also needed periodically during testing to switch between vantage points. Our full test suite takes nearly 45 minutes to execute per vantage point, and there is limited ability to parallelize as each VPN consumes the machine it is running on during testing.

Because of this manual labor and time, as well as the financial cost, it was not operationally feasible to perform measurements on all 200 VPN services we identified. Instead, we created a stratified sample of the VPN services in the ecosystem, ensuring that we had sufficient representation for a number of criteria:

- **Popular VPN services:** Nearly all of the review sites participate in some form of affiliate marketing, and so we did not use their rankings for determining relative popularity. However, we do use the prevalence of a VPN across sites as an imperfect proxy for popularity. Using this metric, we selected the 15 most prevalent VPN services for evaluation.
- **VPNs with free or trial versions:** Two websites, vpnMentor [122] and That One Privacy Site [107], have details of VPN services including cost, features and number of vantage points. We chose 30 VPN services because they included free-level or trial-level options. Doing so both selected for services likely to be popular, and simplified our sign-up process.
- **Random selection:** From the remaining providers, we chose 18 services at random to diversify our selection.

In total, we tested 63 VPN providers. For each provider, we aimed to test at least five vantage points with as much geographic diversity as possible. The actual number of vantage points tested per VPN varied between one (when only one vantage point was available) to nearly 150 (when automated).

4.3.2 Environment and setup

We performed our testing in a macOS virtual machine running across several physical systems. We restored the VM to a known state before testing a new VPN service to ensure a uniform and consistent environment across our tests. macOS provided a UNIX environment while still being popular enough to have client software available from most major VPN providers. In instances where there was no client software available, we tested the provider using OpenVPN when available.

To test each provider, we first registered for a subscription, then downloaded and installed the client. The test suite logs results for each experiment as well as traffic traces for passive analysis. Several VPN servers disconnected repeatedly during the testing phase and hence required partial re-collection of data, or omission from the dataset entirely. A general trend notable across VPNs was the irregularity in vantage point connectivity—while North American and European vantage points typically had few issues, other locales fared far worse. Multiple initially-selected VPN services also had buggy or non-functional clients that prevented us from testing. In this case, we replaced the VPN service with another from our list. Manual and automated collection combined, we collected data from 868 vantage points.

4.3.3 Tests run

VPN services are tested along two main axes: identification of traffic manipulation or surveillance by the provider, and deceptive practices in VPN infrastructure. In each case, we ran this complete set of tests against every vantage point connected to.

Traffic interception and manipulation

We ran several tests to detect potential indicators of traffic interception or manipulation:

- *Web page modification and content injection:* To identify providers who inject content or otherwise modify web traffic in-flight to users, we recorded our own accesses to two specially-crafted ‘honeysites’ using the methodology of Tsirantonakis, et al. [114] and Ikram et al. [51]. Both sites contained fixed content, but one site included ad-inclusion JavaScript code from several major ad networks to provide opportunities for ad injection or hijacking. This code used invalid publisher identifiers to ensure that no ads ran under normal circumstances and there were no financial impacts on ad networks or advertisers.

The pages loaded in a fully-featured Selenium-controlled [101] Chrome browser, recording information about every resource that loaded and the reason for loading it (e.g. requested by a particular iframe, etc.), as well as a complete copy of the final DOM.

- *TLS interception and downgrades:* To reveal any unexpected redirections or TLS stripping, we recorded redirection and TLS information from more than 200 distinct domains. In each case, our test connected to the domain over HTTP, followed any HTTP-based redirects, and recorded all URLs and response headers along the way. The test also recorded and validated every certificate seen against Mozilla's CA trust store. Domains were chosen from the Alexa top 100k sites to cover a wide variety of potentially sensitive topics (including news, politics, pornography, government websites, defense contracting, etc.) in a variety of languages and home countries. The set also contains a mix of sites using HTTPS and non-HTTPS by default.
- *Transparent proxies:* Any transparent proxy present on a VPN may observe or selectively modify traffic in a way not triggered by our previous tests. To try to detect this behavior, we made a series of slightly malformed web requests to an HTTP echo server under our control, and then compared the request sent with the request that the server received. Transparent proxies will often either inject additional headers, or slightly modify requests forwarded, both of which are detected by this test.

For the latter two tests, we also periodically collected known-unmodified data from a university IP several times per day during our study. This dataset was then subsequently used as a ground truth reference in identifying inconsistencies. All inconsistencies were manually investigated.

These tests are not perfect, and cannot detect a variety of techniques. Any purely passive observation techniques (e.g. recording network traffic) is entirely invisible to tests. Similarly, transparent proxies that do not modify requests will not be detected by our proxy identification test. Finally, any traffic modification that is selectively targeted may not be triggered by our significant, but ultimately limited, set of domain accesses. As a result, our results should be considered an underestimate.

VPN infrastructure inference

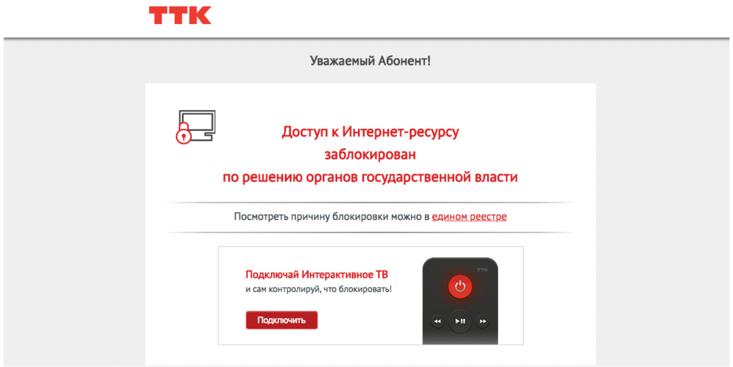
We also collected data on the performance and the purported geographic location of each vantage point:

- *Round-trip access times:* To help aid in estimating the true location of VPN vantage points, our tests collected a variety of latency information. We collected traceroutes to anycast-capable public DNS resolvers (Google Public DNS and Quad9) and well-distributed DNS roots (D, E, F, J, L), as well as ping-based RTT information to 50 RIPE Atlas anchors [99].
- *IP geo-location:* To compare claimed geo-location with IP database's geo-location, our tests collected both the coordinates and reverse geo-coding from the Google Maps API [40]. In addition, egress IPs were subsequently looked up in both IP2Location Lite [52] and MaxMind's GeoLite2 [66] databases.

It is possible to spoof latency measurements from ping, and so our estimates of deception based on RTT should be considered a conservative estimate. Sanity checking as well as additional testing using TCP-based RTTs suggest that such deception is not common place.

4.4 Results

VPN providers largely sell themselves as being neutral data brokers that securely transport your network traffic from your computer to a network located in another location. In this section, we investigate the network behavior of the selected VPN services (Section 4.3.1) using the tests described in Section 4.3.3, and find that a meaningful proportion of VPN providers are not as neutral as their users might hope.



TTK

Уважаемый Абонент!

Доступ к Интернет-ресурсу
заблокирован
по решению органов государственной власти

Посмотреть причину блокировки можно в единном реестре

Подключай Интерактивное ТВ
и сам контролируй, что блокировать!

Figure 4.1. TTK blocked content page in Russia.

Table 4.3. Destination Domains of URL Redirections

Destination Domain	VPNs	Country
`http://195.175.254.2`	8	Turkey
`http://[www.]warning.or.kr`	5	South Korea
`http://fz139.ttk.ru`	4	Russia
`http://zapret.hoztnode.net`	2	Russia
`http://warning.rt.ru`	1	Russia
`http://blocked.mts.ru`	1	Russia
`http://block.dtln.ru`	1	Russia
`http://blackhole.beeline.ru`	1	Russia
`https://www.ziggo.nl`	1	Netherlands
`http://213.46.185.10`	1	Netherlands
`http://103.77.116.101`	1	Thailand

4.4.1 Traffic manipulation and monitoring

While we have limited ability to detect when traffic is transparently monitored (as this can be done passively without leaving side effects), these tests can still detect some forms of monitoring and traffic manipulation. We find that about 10% of VPNs tested monitor traffic for an unknown purpose. In contrast, we find little evidence of traffic manipulation or tampering from providers, with most manipulation we witnessed originating from country-level censorship, rather than from the VPN itself.

Website blocking, redirection and HTTPS downgrades

Our tests detect when unauthenticated HTTP requests unexpectedly redirect to an unrelated domain[1] or the redirections failed to upgrade a connection to HTTPS when expected.

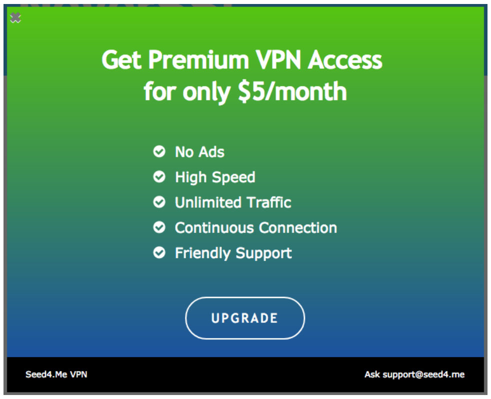
Table 4.3 enumerates the final destination of every instance of VPN-attributable HTTP redirect that we encountered. In each instance, these redirections are upstream country-level website blocking in Turkey, South Korea, Russia, Netherlands or Thailand. We only detected redirection on endpoints claiming to be in the expected countries. Typically, redirections were caused by `HTTP 302` status codes. The types of content experiencing the most blocking was pornography (Turkey, South Korea, Thailand, Russia), followed by file sharing (Turkey, Russia, Netherlands). Additionally, Turkey blocked Wikipedia and Russia blocked `jw.org` and `linkedin.com`. An example blocking page from a Russian ISP is shown in Figure 4.1.

We also identified several instances where websites appear to block known VPN traffic systematically. In these cases, sites often responded with `HTTP 403` response codes, though we also encountered `HTTP 200` response codes with a page explaining the block. Several of these sites perform blocking prior to the initial redirect to HTTPS, and as a result, were identified by our downgrade detection. We identified no VPN providers systematically stripping TLS or causing malicious redirections, though we take the signals we do identify as validation of our technique.

Traffic injection/modification

Our ‘honeysite’ test identified one instance of a VPN provider systematically injecting content into pages. All pages loaded under a free trial account of `Seed4.me` via HTTP were injected with an overlaid advertisement served by custom JavaScript hosted on a subdomain of the VPN provider’s site, as shown in Figure 4.2. We identified no other instances of HTTP injection.

[1] We considered two subdomains to be related if they shared a registered domain (according to the public suffix list [79]), their registered domains differed only by public suffix (e.g. `http://a.example.com` to `http://b.example.org`), or if they were manually determined to be related.



Get Premium VPN Access
for only $5/month

- No Ads
- High Speed
- Unlimited Traffic
- Continuous Connection
- Friendly Support

Figure 4.2. Advertisement injected by `seed4.me` trial VPN service.

Header-based proxy detection

In addition to the content injection from the previous section, we also detected transparent proxies in use in five VPNs: AceVPN, F-Secure's FREEDOME VPN, SurfEasy, CyberGhost and VPN Gate. In each case, proxies did not inject additional headers, but consistently modified our existing headers in ways consistent with header parsing and subsequent regeneration.

The presence of a proxy does not inherently mean that the VPN is doing something nefarious—VPN Gate's server software, for instance, appears to systematically proxy requests (though we can find no documentation of this), but because VPN Gate's server software is provided as an academic project to avoid censorship, we presume malicious monitoring was not the intent. That said, there is little reason to proxy requests, few VPN software stacks do this by default, and so we do not expect legitimate proxying to be common-place. It is also possible that additional services use transparent proxies that were not detected using this method.

4.4.2 Geographic distribution

VPNs make a variety of claims about where their vantage points are located geographically. They provide this list of locations as a selling point for performance, to evade geographic-based blocking on services (such as used by Netflix), and to 'enhance privacy'. While most providers work hard to provide vantage points in a variety of physical locations, some VPN providers take steps to deceive geo-IP databases about the physical locations of their vantage points. The industry occasionally calls these ‘virtual locations’.

Whether or not these virtual locations are a problem depends on the users’ intended use case. Users interested in evading state-level censorship or jurisdictional arbitrage (e.g. for downloading copyrighted content) may care deeply about a vantage point actually being located in a particular country. Other users may only care that a vantage point’s apparent location successfully fools a given web service. So long as VPN providers are open and clear about the use of ‘virtual’ locations, the service is not defrauding the user, but problems arise when VPN providers fail to ensure that users understand what the service is providing.

Claimed location vs. geo-IP databases

To get a handle on the level of disagreement between claimed VPN vantage point location and perceived location, we compared the claimed location of 626 vantage points against the locations provided by three geo-IP databases: IP2Location Lite [52], MaxMind’s GeoLite2 [66], and Google’s location service as seen from the Google Maps API [40]. The first two services are freely available, and the third can be used in a limited fashion to geo-locate a requester’s IP address.

In all cases, there were significant disparities between claimed and IP-geolocated locations, with the greatest agreement coming from the freely available databases. Google was able to provide a prediction of location in 541 cases, of which only 377 (70%) agreed with the VPN’s claimed location. IP2Location and MaxMind each provided estimates for 612 vantage points, but these locations matched the predicted locations in 552 instances (90%) and 583 instances (95%) respectively.

For all databases, about one third of the inconsistencies were the database claiming a vantage point was hosted in the US when the claimed location was elsewhere, with wide varieties of claimed locations. The difficulty of geolocating IP addresses is well documented [38], and errors exist in each dataset. However, the relative agreement among the free services, while

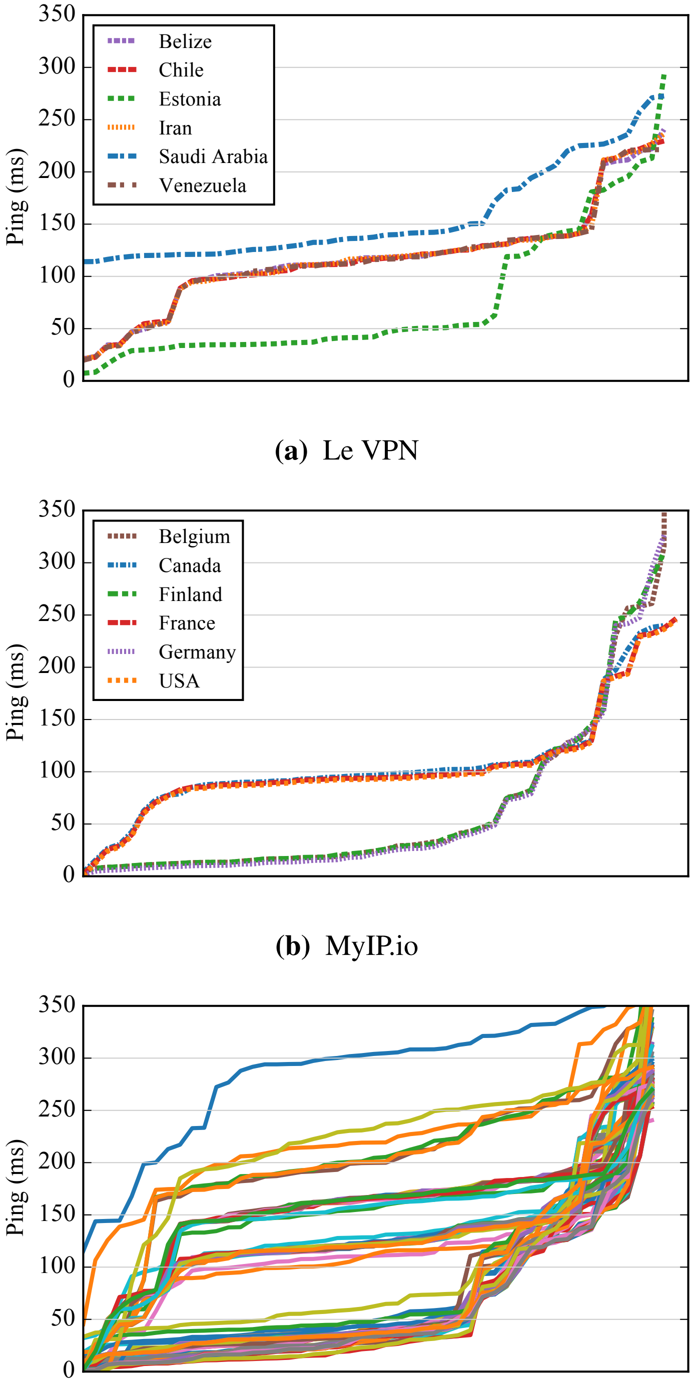
Google's service disagreeing nearly a third of the time suggests significant manipulation of IP location databases by VPN providers, who can often change their geolocation in free databases by contacting the database provider. In contrast, Google's data suggests that 'virtual' vantage points may be even more common than we identify in the next section.

Identifying 'virtual' vantage points

Our data allows us to identify several previously-unknown instances of 'virtual' vantage points by identifying vantage point co-location through comparison of round trip times between the vantage points and known hosts.

In total, we identified discrepancies in vantage point locations from six of our 63 VPN providers: HideMyAss, Avira, LeVPN, Freedom-IP, MyIP.io and VPNUK. In some cases, we were able to identify obvious discrepancies from discrepancies in ping times. As an example, Avira's 'US' vantage point registers ping times to known hosts in Germany, Luxembourg, and the Netherlands in less than 9 ms each, while ping times to known US hosts ranged between 113 and 173 ms. Such ping times indicate that this vantage point is actually in Europe.

Often, however, discrepancies are harder to spot by individual measurements alone. In these cases, we identify abnormalities by comparing measurements across multiple vantage points. For each vantage point, the series of ping-based RTT times to known hosts forms a sort of fingerprint that reflects the network properties of the particular data center in which the vantage point is hosted. In addition to suggesting geographic location, comparing these fingerprints reliably detects co-located hosts sharing a data center. MyIP.io, for instance, reports having five vantage points located in the US, France, Belgium, Germany and Finland. While the egress IPs of the US and French vantage points share the same /24 IP prefix, and the Belgian, German and Finnish vantage points share another, this is not by itself a guarantee of co-location—some VPNs that manage their own IP space heavily subdivide their ranges into very small chunks across locations. Comparison of fingerprints, however, reveals a strong match between the US and French vantage points, likely located in Montreal. Similarly, Belgium, Germany and Finnish



(a) Le VPN

(b) MyIP.io

(c) HideMyAss

Figure 4.3. Distributions of RTTs to web hosts, ordered from lowest RTT to highest, for various vantage points of three different VPN providers. Note the strong correlation between some series—these vantage points are co-located despite claims of being in separate countries. The bottom graph plots 148 different series corresponding to distinct HideMyAss vantage points.

locations appear to reside together, likely in the UK.

Figure 4.3 demonstrates co-location of vantage point at three VPN providers. In each case, the graph shows a distribution of round-trip times to our chosen hosts, ordered from lowest RTT to highest. Though not shown, the same hosts appear in the same order across vantage points. In the case of the European vantage points of MyIP.io, ping times to a reference host varied by less than 1.5ms across different vantage points.

The provider offering the most 'virtual' vantage points by far is HideMyAss. An analysis of nearly 150 of their endpoints reveals relatively few physical locations. Dozens of locations in North, Central or South America, for instance, appear to be based out of the Seattle area, while dozens more vantage points appear to be based out of Miami, Prague, London and possibly Berlin. While HideMyAss specifically advertises having 'virtual' locations—even listing them separately from physical locations—many supposedly physical locations (like North Korea) are listed as physical while still clearly virtualized.

4.5 Related work

Appelbaum *et al.* identified the theoretical security risks and vulnerabilities of commercial and public VPN services [5], but did not measure the prevalence of any potential malicious activity. Perta *et al.* [93] analyzed the behavior of 14 popular VPN services, identifying widespread IPv6 leakage opportunities for severe DNS hijacking leading to a full failure of the VPN's ability to protect traffic against a strong adversary. This study did not fully investigate the possibility of a malicious VPN provider. Al-Fannah studied the privacy risks of the introduction of WebRTC APIs in modern web browsers [2], demonstrating how WebRTC API can compromise user's anonymity by leaking a range of client IP addresses to a visited website, even if a VPN is in use. This chapter aimed to extend these works to systematically measure other aspects of commercial VPN services.

The closest studies to our own have focused on the mobile space. A study by Ikram *et* *al.* [51] analyzed more than 200 Android VPN apps, identifying instances of malware, traffic leakages, traffic manipulations and even TLS interception. Among other results, Ikram's work revealed that HotspotShield VPN actively injected JavaScript code to redirect users to partner companies [13]. Finally, Zhang *et al.* investigated security vulnerabilities in 84 OpenVPN-based Android applications [137], confirming the findings of previous studies in the mobile space, by reporting instances of vulnerable VPN mobile apps due to insecure custom modification and developer-induced misconfigurations. Our work is the first to our knowledge to measure the prevalence of malicious VPN providers in the broader commercial VPN ecosystem outside of the mobile space.

As VPNs claim to provide access to vantage points remotely, a number of academic efforts leverage VPN services for censorship analysis [16, 65, 80, 113], and network analysis of ISPs [17, 116]. The findings of this project confirm that running measurements over VPN services require caution due to their frequent traffic manipulation practices.

Related studies have also investigated other mechanisms for routing traffic for censorship evasion or anonymization [59, 126]. Some amount of traffic manipulation and monitoring also occurs in these other types of network relays, including open HTTP proxies [114] and even anonymity networks [14, 132]. The work by Tsirantonakis *et al.* studied header manipulations performed by over 65,000 open HTTP Proxies [114], finding around 5% of tested proxies performing unwanted or malicious modification or injection such as ad injection, redirection, as well as JavaScript injection for tracking and user fingerprinting purposes. Though our dataset is smaller, and VPN service providers operate under different incentives, our results are largely in line with these results.

4.6 Discussion & Conclusions

The results of prior studies of Android VPNs [51] and open web proxies [114] suggest that the VPN ecosystem might contain a substantial fraction of providers looking to take advantage of their privileged position between the Internet and their customers. In particular, prior work shows that significant numbers of vantage points intercepted and/or manipulated traffic. In this respect, our findings are broadly similar with these adjacent ecosystems—we find around 10% of VPN providers manipulating or monitoring traffic. Despite this, there are two significant differences between the broad commercial VPN ecosystem and the other two that we might expect would make commercial VPNs less malicious. Firstly, there are fewer mechanisms available to VPN providers to intercept or manipulate traffic than are available to, for instance, mobile VPN providers—many of the VPN services we consider rely on standardized protocols and client software. This standardization prevents providers from as easily performing some forms of malicious activity (e.g., TLS interception or surreptitious routing of client traffic through other clients). Perhaps more importantly, the vast majority of the VPNs we measure are paid services. These services therefor have less incentive to monetize their customers' traffic directly when compared to free services.

The greatest incentives for paid VPN services are not to manipulate customer traffic, but rather to attract new customers. One of the few user-visible ways VPN services have to differentiate themselves to potential customers is through the set of countries in which they provide vantage points. As a result, VPN providers are incentivized to inflate this set of vantage points as much as possible. It is this observation that leads us to search for, and ultimately identify, providers misleading users on what vantage points the services offer. Our measurement technique is unique in that evading it is hard—shy of placing each vantage point in the claimed country, it is very difficult to avoid latency-based artifacts that betray the provider's deception. Consistent with the theory, we see clear evidence of around 10% of providers artificially inflating their numbers by using 'virtual' vantage points rather than a physical presence in a given country. These 'virtual' vantage points are drastically cheaper for providers, and likely go unnoticed by many users, especially given that most VPN providers are not upfront about their use.

VPNs are positioned to not only observe, but modify traffic as it passes through their services, and decisions VPN providers make (such as vantage point placement) can have real consequences for users who rely on those services. As the VPN popularity continues to grow, the ability for users to verify the claims made by providers only becomes more vital. Techniques like those presented in this chapter provide a critical tool to keep providers honest in the large unregulated ecosystem of modern commercial VPNs. Though we have evaluated a large sample of VPN providers, we have not evaluated every VPN provider—our techniques can, however, be used by anyone wishing to evaluate the integrity of their own provider's claims. Especially given that our analysis is necessarily an underestimate, we have found substantial evidence that VPN providers are not always honest, and that caution before providing them with trust is well deserved.

Acknowledgments

This chapter is part of work that has been accepted for publication in Proceedings of the ACM Internet Measurement Conference, 2018. Khan, Mohammad Taha; DeBlasio, Joe; Voelker, Geoffrey M.; Snoeren, Alex C.; Kanich, Chris; Vallina-Rodriguez, Narseo. The dissertation author was a primary investigator and author of this paper.

Chapter 5

Conclusion

Abuse, attacks and fraud are a reality of the modern Internet. With no reason to expect this dynamic to go away, developing common tools and methods for finding fraud and abuse can help make the task of countering these actors more tractable. While problem areas and abuse mechanisms are constantly evolving, attackers' motives thankfully stay mostly static. It is these invariants that give us a mechanism on which to base our common defensive strategies.

This dissertation focuses on detecting activity of financially motivated attackers, permitting us to leverage the constraints of these actors: they must make money. These actors have to make choices—invest more money in better defenses to stay undetected, or pocket that cash. When faced with these options, these actors understandably invest only where, and only as much, as they think they have to to stay under the radar. By limiting their investment, they leave cracks in their armor—opportunities for defenders to take advantage. As attackers methods' evolve, defenders can use this approach again and again—the fundamental trade-offs on which it relies do not go away. Each successive iteration increases the costs to the malicious actor, slowly but surely decreasing their interest in attacking at all.

Throughout this dissertation, I demonstrate that this approach can help guide defenders to detect and mitigate financially-motivated attacks, even in substantially different problem spaces. In Chapter 2, we apply this technique to identify when web services are compromised by external attackers by looking for password re-use attacks. Password re-use provides an attractive mechanism for attackers to monetize their initial compromises, but evading our detection requires leaving password re-use gains unrealized. In Chapter 3, we suggest using targeted policy changes to shut down search advertising fraud. In particular, the method suggests targeting the verticals that fraudulent search advertisers are forced into. Fraudsters can evade these policy changes by targeting less profitable verticals, or by trying to hide what they are advertising but in so doing, they battle an algorithm that demotes such behavior, reducing profit. Finally, in Chapter 4, we use this technique to identify when VPN service providers are misleading their users by lying about vantage point locations. We detect this deception by inferring when vantage points are co-located based on similar round-trip times to other hosts. Evading this detection would mean either trying to spoof RTTs (which we could detect), or upsetting their paying customers by adding latency to all users. In each case, our framework provides guidance on finding mechanisms that can detect attacks, fraud, and abuse despite considerable differences in problem domain. Few tools offer such wide applicability.

This approach has a few substantial shortcomings that are left for future work. Firstly, this dissertation frames problems by looking at the malicious actors' incentives, but it does not provide guidance on where to look for weak illusions to leverage. Poor illusions seem to occur frequently at the interface between malicious actors and other non-victim actors, but future work is needed to investigate this idea or provide other guidance on where to look for weak illusions.

This framework also offers no guidance for non-financially-driven malicious actors. While nation-state and socially-motivated attackers operate under incentive-based constraints, their incentive structures are somewhat harder to influence. Despite this, there may yet be room to extend this work to address these diverse actors.

While work remains, this dissertation provides promising direction towards detecting and mitigating financially motivated malicious activity. Further, it demonstrates these techniques in new, timely and interesting domains, further contributing to the growing body of work undermining financially motivated attacks, fraud, and abuse.

Appendix A

Tripwire Disclosure Emails

Below are examples of email messages sent to each site. We modified a common template as needed to accommodate details for each site, and for clarity. Subsequent messages in a thread were written as-needed.

A.1 Initial Email Message

Hello,

We're contacting you because of your affiliation with SITE X, and because we believe that usernames and passwords on that site have been compromised at some point. Our earliest evidence of this compromise is from around DATE, though the actual compromise may have occurred earlier.

We are researchers at UC San Diego conducting research on detecting compromised websites, and your site is among several thousand that we have been studying for more than a year.

We would be happy to share more details of our monitoring and discovery with the appropriate party. Please have them contact us by replying to this email.

A.2 Follow-up Email Message

Our detection technique works by registering for an account on your site with a fresh email address uniquely associated with that account. Accounts we create share their passwords with the corresponding email addresses, but nowhere else. The accounts created are never used, and the email address is only ever used on that one site, and never exposed to any other site.

With the assistance of a major email provider, we monitor the email accounts for successful logins to the email accounts. Any successful login indicates a password reuse attack on the associated account on the monitored site, and thus a compromise to the service. Our study allows us to monitor any site with a login system without requiring any knowledge of the site's implementation.

In this case, we registered for two accounts on SITE X. One account had an eight-character password in the form of a capitalized word followed by one number (e.g. 'Website1'). The other account has a password that was a randomly generated ten character mixed-case alphanumeric string (e.g. 'i5Nss87yf').

These differing passwords allow us to differentiate between attacks where the attacker received only well-hashed passwords or whether the attacker received plain-text (or poorly hashed) passwords.

On your site, we witnessed successful logins to both accounts, implying that the attacker had access to full passwords from the accounts.

We registered for the accounts on your site in MONTH, YEAR. We have seen accesses to the corresponding email addresses starting on DATE, so the compromise occurred between those dates.

Unfortunately, due to our agreement with our email provider, we're unable to disclose the exact accounts that were accessed (as this would expose who our email provider is).

We wanted to let you know of our findings in case the compromise timeframe is useful to you. We'd be interested to know if you were aware of any compromise in that timeframe, or any other details you'd be willing to share. I'm also happy to try to answer any questions you have, insofar as I'm able.

Best,

A.3 Disclosure Responses

Below are a few highlights from emails received in response to our disclosures. These are included to highlight the wide variety of sites vulnerable to Tripwire-detectable compromises.

A.4 Response from Site C

joe your system is simple and amazing :)
i can share the following information with you. January 2016 I noticed that somebody found a vulnerability on the site and downloaded mysql database. I notcied it because attacjer created a clone site in some days. Later hacker confirmed that he downloaded database and I found vulnerability and closed it. But vulnerability expisted before and somebody else could have our db but use it in private (your case).

Then I must add that passwords are hashed, and until January 2016 it was just an md5 so easy passwords could be bruteforced.

Conclusion:

Useful information from you for me: vulenrability has been used only only in 2016 but in 2015 as well. Sad :(Useful information from me to you: your system works and realy effecient to track data leaks ;) Excellent :)

If you have any interesting information to add please let me know.

A.5 Response from Sites E/F

We've been unable to correlate the compromise of the canary from your study with any other findings. It's unfortunate that we weren't able to begin this investigation last fall when the canary was found to have been compromised but I do very much appreciate your disclosure and we are continuing to investigate.

Would your mail provider be willing to bring us into the know on your relationship and the identity of the canary accounts on a TLP Amber or Red basis? I'm a member of the Operations Security Trust information sharing group, if that helps them broker a trust relationship through their peers of mine. Can you or they share any other indicators that you observed in your work? We're interested in any threat actor observables or details like the datetime of access, source IP, source Geo, source reputation, TTPs, intent, profiles, etc.

SITE E and SITE F expose usernames on a very discoverable page (e.g., URL ON SITE E) and do not have sophisticated checker or credential brute force controls: might that have exposed your 8 character password canary account to attack against our infrastructure? Do your methods account for the existence of bruters and checkers?

Do these IP addresses we found having had logged into more than one account mean anything to you:

- [3 IPs in the same /16 belonging to DigitalOcean, Inc.]

Also: many sites only retain the current email address associated with an account. You may be able to further obscure your mail provider by returning to the sites you registered on and changing the addresses to be one you control, perhaps using subaddresses?

A.6 Response from Site G

I spent some time looking for signs of how it might have been compromised. I did not find any thing that indicated access. I did find a couple of sql commands that were not properly protected/escaped. I also made sure the server was up to date. The server was moved from a dedicated host to an AWS ec2 about a year ago. It might have been on the old server that it was compromised.

I see, and have seen for a very long time indication of people trying to access the server. Both in ssh password guessing and in sql injection attempts. I did not see anything that made me think either was successful. I'm not 100% sure I would though.

My plan is to force a password reset when users login. I need to do some more development for that. I also need to update wordpress. I don't believe it is up to date.

Thoughts?

Thanks,

A.7 Responses from Site L

Your work sounds fascinating, certainly a very low level way of detecting without giving the game away.

If I give you a little history of he sites you'll get the idea of how it works. I started it in 2007 with a low level of IT knowledge. Most of the php orriginates from around 2008 with various add ons over time, the fact we are limited to php 5.3 gives that away. We did used to employ sysadmins who cost a lot of money and made it their purpose in life to frustrate every possible upgrade I tried to complete so I threw them out in April 2015 and took over the sites myself, being thrown in the deep end is an understatement.

> Our previous setup was 60 servers, 30 for media streaming and 30 for back end made up mostly of database and web servers. These servers used the public network to communicate and were pretty exposed, the fact we hadn't been compromised over that 8-9 year period surprises me. I've since been moving stuff to cloud and only in the last few months have back end servers been taken to a private network. Believe it or not the passwords were only encrpyted since I took over the sites, they were plain text before. I had plans to add salt to the encryptiion, I will now accelerate that. After that code such as php will need updating as well as some XSS vulnerabilites we have.
>
> In terms of the likely time, I'm trying to find my notes but we suffered a huge DDos around March, it went on for days. It seemed to be a linux vulnerability checker and the traffic was coming through TOR nodes. When I blocked the TOR network they moved to digitalocean etc Leaseweb also suffered from a very high number of DDos at the time, they are not normally affected by them, here is a email from aroud that time:
>
> Between 01:10 and 01:25 CEST we experienced a large DDoS attack in part of our network. You might have noticed packet loss or increased latency during that time. We mitigated the attack and traffic levels are now fully restored.
>
> I know what I've provided isn't a massive help but let me know if there is anything specific you would like, I'm happy to help in return for your help alerting me :)
>
> Thanks,

We subsequently inquired whether he would be notifying users and/or taking any technical countermeasures. In response, we received the following email.

> Not planning to say anything at the moment, we are currently salting the passwords, test script is being run tomorrow. in the meantime we've been

adding as much protection as we can, the obvious elephant in the room is our version of php/httpd, both are old.

Thanks,

Bibliography

[1] Hackers selling 117 million LinkedIn passwords. http://money.cnn.com/2016/05/19/technology/linkedin-hack/.

[2] Nasser Mohammed Al-Fannah. One Leak Will Sink A Ship: WebRTC IP Address Leaks. 2017.

[3] Alexa Top 500 global sites. http://www.alexa.com/topsites.

[4] Sumayah A. Alrwais, Christopher W. Dunn, Minaxi Gupta, Alexandre Gerber, Oliver Spatscheck, and Eric Osterweil. Dissecting Ghost Clicks: Ad Fraud Via Misdirected Human Clicks. In *Proc. Annual Computer Security Applications Conf.*, December 2012.

[5] Jacob Appelbaum, Marsh Ray, Karl Koscher, and Ian Finder. vpwns: Virtual Pwned Networks. In *USENIX FOCI*, 2012.

[6] Cluster of 'megabreaches' compromises a whopping 642 million passwords. http://arstechnica.com/security/2016/05/cluster-of-megabreaches-compromise- a-whopping-642-million-passwords/.

[7] Prithvi Bisht and VN Venkatakrishnan. XSS-GUARD: precise dynamic prevention of cross-site scripting attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 23–43. Springer, 2008.

[8] Matt Bisson. Bing Ads Auction Explained: How Bid, Cost-per-Click and Quality Score Work Together. https://advertise.bingads.microsoft.com/en-us/blog/post/september-2013/bing-ads-auction-explained-how-bid,-cost-per-click-and-quality-score- work-together.

[9] Stephen W Boyd and Angelos D Keromytis. SQLrand: Preventing SQL injection attacks. In *International Conference on Applied Cryptography and Network Security*, pages 292–302. Springer, 2004.

[10] CA Civil Code Section 1798.80-1798.84. http://www.leginfo.ca.gov/cgi-bin/displaycode?section=civ&group=01001-02000&file=1798.80-1798.84.

[11] Davide Canali, Davide Balzarotti, and Aurélien Francillon. The Role of Web Hosting Providers in Detecting Compromised Websites. In *Proceedings of the 22nd International World Wide Web Conference*, pages 177–188, 2013.

[12] Carbonite. Carbonite Accounts Targeted In Password Reuse Attack. https://www.carbonite.com/en/cloud-backup/business/resources/carbonite-blog/carbonite-password-attack/.

[13] Complaint, Request for Investigation, Injunction, and Other Relief. AnchorFree, Inc. Hotspot Shield VPN. https://cdt.org/files/2017/08/FTC-CDT-VPN-complaint-8-7-17.pdf, Apr 2017.

[14] Sambuddho Chakravarty, Georgios Portokalidis, Michalis Polychronakis, and Angelos D Keromytis. Detecting Traffic Snooping in Tor Using Decoys. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2011.

[15] Bill Cheswick. An Evening with Berferd. In *Proceedings of the Winter USENIX Conference, San Francisco*, pages 20–24, 1992.

[16] Shinyoung Cho, Rishab Nithyanand, Abbas Razaghpanah, and Phillipa Gill. A Churn for the Better. In *Proceedings of the ACM Int. Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2017.

[17] Taejoong Chung, David Choffnes, and Alan Mislove. Tunneling for Transparency: A Large-Scale Analysis of End-to-End Violations in the Internet. In *Proceedings of the ACM Int. Measurement Conference (IMC)*. ACM, 2016.

[18] NSA disguised itself as Google to spy, say reports. https://www.cnet.com/news/nsa-disguised-itself-as-google-to- spy-say-reports/.

[19] Researchers find stolen military drone secrets for sale on the dark web. https://www.cnet.com/news/researchers-found-stolen-military-secrets-for-sale-on-the-dark-web/.

[20] Lester Coleman. Hacked BitcoinTalk.org User Data Goes Up For Sale On Dark Web. https://www.cryptocoinsnews.com/hacked-bitcointalk-org-user-data- goes-up-for-sale-on-dark-web/, June 2016.

[21] Hire a DDoS service to take down your enemies. https://www.csoonline.com/article/3180246/data-protection/hire-a-ddos-service-to-take-down-your-enemies.html.

[22] What is ransomware? How it works and how to remove it. https://www.csoonline.com/article/3236183/ransomware/what-is-ransomware-how-it-works-and-how-to-remove-it.html.

[23] No wonder hackers have it easy: Most of us now have 26 different online accounts - but only five passwords. http://www.dailymail.co.uk/sciencetech/article-2174274/No-wonder-hackers-easy-Most-26- different-online-accounts-passwords.html.

[24] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The Tangled Web of Password Reuse. In *Proc. Network and Distributed System Security Symposium*, February 2014.

[25] Dashlane. [INFOGRAPHIC] Online Overload — It's Worse Than You Thought. https://blog.dashlane.com/infographic-online-overload-its-worse- than-you-thought/.

[26] Neil Daswani and Michael Stoppelman. The Anatomy of Clickbot.A. In *Proceedings of the First Workshop on Hot Topics in Understanding Botnets*, 2007.

[27] Vacha Dave, Saikat Guha, and Yin Zhang. Measuring and Fingerprinting Click-Spam in Ad Networks. In *Proceedings of the ACM SIGCOMM Conference*, August 2012.

[28] Vacha Dave, Saikat Guha, and Yin Zhang. ViceROI: Catching Click-Spam in Search Ad Networks. In *Proceedings of the ACM Conference on Computer and Communications Security*, Berlin, Germany, November 2013.

[29] Emiliano De Cristofaro, Arik Friedman, Guillaume Jourjon, Mohamed Ali Kaafar, and M Zubair Shafiq. Paying for Likes?: Understanding Facebook Like Fraud Using Honeypots. In *Proceedings of the Internet Measurement Conference*, 2014.

[30] DeCaptcher — CAPTCHA solving service, math CAPTCHA bypass, hard CAPTCHA recognition. http://de-captcher.com.

[31] Dorothy E. Denning. An Intrusion-Detection Model. *IEEE Transactions on software engineering*, pages 222–232, 1987.

[32] Adam Doupé, Ludovico Cavedon, Christopher Kruegel, and Giovanni Vigna. Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner. In *Proceedings of the of the 21st USENIX Security Symposium*, pages 523–538, 2012.

[33] Fake Name Generator. http://fakenamegenerator.com.

[34] Dinei Florencio and Cormac Herley. A Large-scale Study of Web Password Habits. In *Proceedings of the 16th International World Wide Web Conference*, pages 657–666, 2007.

[35] José Fonseca, Marco Vieira, and Henrique Madeira. Evaluation of Web Security Mechanisms Using Vulnerability & Attack Injection. *IEEE Transactions on Dependable and Secure Computing*, 11(5):440–453, 2014.

[36] Sean Ford, Marco Cova, Christopher Kruegel, and Giovanni Vigna. Analyzing and Detecting Malicious Flash Advertisements. In *Proc. Annual Computer Security Applications Conf.*, December 2009.

[37] Shirley Gaw and Edward W. Felten. Password Management Strategies for Online Accounts. In *Proceedings of the Second Symposium on Usable Privacy and Security*, pages 44–55, 2006.

[38] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. A Look at Router Geolocation in Public and Commercial Databases. In *Proceedings of the ACM Int. Measurement Conference (IMC)*. ACM, 2017.

[39] GitHub. GitHub Security Update: Reused password attack. https://github.com/blog/2190-github-security-update-reused- password-attack.

[40] Google Maps Platform – Geolocation APIs. https://cloud.google.com/maps-platform/.

[41] Google. Verify your Google Account. https://support.google.com/accounts/answer/63950.

[42] GoToMyPC. GoToMyPC Password Issues: Incident Report for GoToMyPC System Status. http://status.gotomypc.com/incidents/s2k8h1xhzn4k.

[43] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. Manufacturing Compromise: The Emergence of Exploit-as-a-Service. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 821–832, Raleigh, NC, October 2012.

[44] Online passwords: keep it complicated. https://www.theguardian.com/technology/2012/oct/05/online-security- passwords-tricks-hacking.

[45] Hamed Haddadi. Fighting Online Click-Fraud Using Bluff Ads. *SIGCOMM Comput. Commun. Rev.*, 40(2):21–25, April 2010.

[46] Have I Been Pwned — Pwned Websites. https://haveibeenpwned.com/PwnedWebsites.

[47] Lowell Heddings. Here's What Happens When You Install the Top 10 Download.com Apps. http://www.howtogeek.com/198622/heres-what-happens-when-you-install- the-top-10-download.com-apps/, January 2015.

[48] Cormac Herley and Dinei Florêncio. Protecting Financial Institutions from Brute-Force Attacks. In Sushil Jajodia, Pierangela Samarati, and Stelvio Cimato, editors, *Proceedings of the 23rd International Information Security Conference*, pages 681–685, 2008.

[49] Ariya Hidayat. PhantomJS. http://phantomjs.org.

[50] Jason Hong. The State of Phishing Attacks. *Communications of the Association for Computing Machinery*, 55(1):74–81, January 2012.

[51] Muhammad Ikram, Narseo Vallina-Rodriguez, Suranga Seneviratne, Mohamed Ali Kaafar, and Vern Paxson. An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps. In *Proceedings of the ACM Int. Measurement Conference (IMC)*, 2016.

[52] Free IP Geolocation Database. https://lite.ip2location.com/, May 2018.

[53] Blake Ives, Kenneth R. Walsh, and Helmut Schneider. The Domino Effect of Password Reuse. *Commun. ACM*, 47(4):75–78, April 2004.

[54] Ari Juels and Ronald L. Rivest. Honeywords: Making Password-cracking Detectable. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pages 145–160, 2013.

[55] Ari Juels, Sid Stamm, and Markus Jakobsson. Combating Click Fraud via Premium Clicks. In *Proceedings of 16th USENIX Security Symposium*, 2007.

[56] Stefan Kals, Engin Kirda, Christopher Kruegel, and Nenad Jovanovic. SecuBat: A Web Vulnerability Scanner. In *Proceedings of the 15th International World Wide Web Conference*, pages 247–256, 2006.

[57] Mohammad Karami, Shiva Ghaemi, and Damon Mccoy. Folex: An Analysis of an Herbal and Counterfeit Luxury Goods Affiliate Program. In *IEEE eCrime Research Summit*, San Francisco, CA, September 2013.

[58] KeyCAPTCHA — Innovative Anti-Spam Solution. https://www.keycaptcha.com/.

[59] Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen M Swanson, Steven J Murdoch, and Ian Goldberg. SoK: Making Sense of Censorship Resistance Systems. *Proceedings of the Int. Privacy Enhancing Technologies Symposium (PETS)*, 2016.

[60] Sheharbano Khattak, Mobin Javed, Syed Ali Khayam, Zartash Afzal Uzmi, and Vern Paxson. A Look at the Consequences of Internet Censorship Through an ISP Lens. In *Proceedings of the ACM Int. Measurement Conference (IMC)*. ACM, 2014.

[61] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Márk Félegyházi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *Proceedings of the IEEE Symposium and Security and Privacy*, pages 431–446, Oakland, CA, May 2011.

[62] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising. In *Proceedings of the ACM Conference on Computer and Communications Security*, Raleigh, NC, October 2012.

[63] Bin Liu, Suman Nath, Ramesh Govindan, and Jie Liu. DECAF: Detecting and Characterizing Ad Fraud in Mobile Apps. In *Proceedings of the 11th ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, April 2014.

[64] Anonymous says it took down Oakland police, city websites. http://www.latimes.com/local/lanow/la-me-ln-anonymous-oakland-police-city-websites-20141210-story.html.

[65] Anuradha Mathrani and Massoud Alipour. Website Blocking Across Ten Countries: A Snapshot. In *PACIS*, 2010.

[66] GeoIP2 Databases. https://www.maxmind.com/en/geoip2-databases, May 2018.

[67] Damon McCoy, Hitesh Dharmdasani, Christian Kreibich, Geoffrey M. Voelker, and Stefan Savage. Priceless: The Role of Payments in Abuse-advertised Goods. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 845–856, Raleigh, NC, October 2012.

[68] Ahmed Metwally, Divyakant Agrawal, and Amr El Abbadi. DETECTIVES: DETEcting Coalition hiT Inflation attacks in adVertising nEtworks Streams. In *Proceedings of the of the International Web Conference (WWW)*, pages 241–250, 2007.

[69] Ahmed Metwally, Fatih Emekçi, Divyakant Agrawal, and Amr El Abbadi. SLEUTH: Single-pubLisher attack dEtection Using correlaTion Hunting. *PVLDB*, 1(2):1217–1228, August 2008.

[70] Microsoft. Ad policies and guidelines - bing ads. https://advertise.bingads.microsoft.com/en-us/resources/bing-ads-policies.

[71] Microsoft. Bing ads policies change log. https://advertise.bingads.microsoft.com/en-us/resources/policies/bing-ads-policies-change-log.

[72] Microsoft. Keyword match type options. https://advertise.bingads.microsoft.com/en-us/resources/training/keyword-match-options.

[73] Microsoft. Recent Changes to Improve Account Security in Bing Ads. http://advertise.bingads.microsoft.com/en-us/blog/27853/recent-changes-to-improve-account- security-in-bing-ads, September 2013.

[74] Microsoft. 5 Best Practices When Signing in to Bing Ads with a Microsoft Account. http://advertise.bingads.microsoft.com/en-in/blog/28115/5-best-practices- when-signing-in-to-bing-ads-with- a-microsoft-account, January 2014.

[75] Brad Miller, Paul Pearce, Chris Grier, Christian Kreibich, and Vern Paxson. What's Clicking What? Techniques and Innovations of Today's Clickbots. DIMVA, pages 164–183, 2011.

[76] Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. Re: CAPTCHAs-Understanding CAPTCHA-Solving Services in an Economic Context. In *USENIX Security Symposium*, volume 10, page 3, 2010.

[77] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. An Analysis of Underground Forums. In *Proceedings of the ACM Internet Measurement Conference*, pages 71–80, Berlin, CA, November 2011.

[78] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. Dirty Jobs: The Role of Freelance Labor in Web Service Abuse. In *Proceedings of the USENIX Security Symposium*, pages 203–218, San Francisco, CA, August 2011.

[79] Public Suffix List. https://publicsuffix.org/, May 2018.

[80] Zubair Nabi. The Anatomy of Web Censorship in Pakistan. In *USENIX FOCI*, 2013.

[81] 12 Russian Agents Indicted in Mueller Investigation. https://www.nytimes.com/2018/07/13/us/politics/mueller-indictment-russian-intelligence-hacking.html.

[82] Behind a Veil of Anonymity, Online Vigilantes Battle the Islamic State. https://www.nytimes.com/2015/03/25/world/middleeast/behind-a-veil-of-anonymity-online-vigilantes-battle-the-islamic-state.html.

[83] Defending Against Hackers Took a Back Seat at Yahoo, Insiders Say. http://www.nytimes.com/2016/09/29/technology/yahoo-data-breach-hacking.html.

[84] North Korea Said to Be Target of Inquiry Over $81 Million Cyberheist. https://www.nytimes.com/2017/03/22/business/dealbook/north-korea-said-to-be-target-of-inquiry-over-81-million-cyberheist.html.

[85] The World Once Laughed at North Korean Cyberpower. No More. https://www.nytimes.com/2017/10/15/world/asia/north-korea-hacking-cyber-sony.html.

[86] Yahoo Says Hackers Stole Data on 500 Million Users in 2014. http://www.nytimes.com/2016/09/23/technology/yahoo-hackers.html.

[87] Daiyuu Nobori and Yasushi Shinjo. VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2014.

[88] NordVPN. DNS Leakage test. https://nordvpn.com/features/dns-leak-test/, 2018.

[89] Jeremiah Onaolapo, Enrico Mariconti, and Gianluca Stringhini. What Happens After You Are Pwnd: Understanding the Use of Leaked Webmail Credentials in the Wild. In *Proceedings of the Internet Measurement Conference*, 2016.

[90] The Open Technology Fund. https://www.opentech.fund/, 2018.

[91] Paul Pearce, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. Characterizing Large-Scale Click Fraud in ZeroAccess. In *Proceedings of the ACM Conference on Computer and Communications Security*, Scottsdale, Arizona, November 2014.

[92] Nicole Perlroth and David Gelles. Russian Hackers Amass Over a Billion Internet Passwords. http://www.nytimes.com/2014/08/06/technology/russian-gang-said-to-amass-more- than-a-billion-stolen-internet- credentials.html, August 2014.

[93] Vasile C Perta, Marco V Barbera, Gareth Tyson, Hamed Haddadi, and Alessandro Mei. A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN Clients. *Proceedings of the Int. Privacy Enhancing Technologies Symposium (PETS)*, 2015.

[94] Quantcast — Top Ranking International Websites. https://www.quantcast.com/top-sites.

[95] reCAPTCHA: Easy on Humans, Hard on Bots. https://www.google.com/recaptcha.

[96] Reddit: VPN. https://www.reddit.com/r/VPN//, 2018.

[97] Charles Reis, Steven D. Gribble, Tadayoshi Kohno, and Nicholas C. Weaver. Detecting In-Flight Page Changes with Web Tripwires. In *Proceedings of the 5th ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, April 2008.

[98] Forrester Research. US Digital Marketing Forecast, 2014 To 2019. https://www.forrester.com/report/US+Digital+Marketing+Forecast+2014+To+2019/-/E-RES116965.

[99] RIPE Atlas. https://atlas.ripe.net/.

[100] Martin Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX Conference on System Administration*, LISA '99, 1999.

[101] Selenium — Web Browser Automation. https://www.seleniumhq.org/, 2018.

[102] Kyle Soska and Nicolas Christin. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *23rd USENIX Security Symposium*, pages 625–640, August 2014.

[103] Kevin Springborn and Paul Barford. Impression Fraud in On-line Advertising via Pay-Per-View Networks. In *Proceedings of the USENIX Security Symposium*, August 2013.

[104] Size of the virtual private network (VPN) market worldwide by type in 2014 and 2019 (in billion U.S. dollars). https://www.statista.com/statistics/542797/worldwide-virtual-private-network-market-by-type/, 2018.

[105] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. Understanding Fraudulent Activities in Online Ad Exchanges. In *Proc. Internet Measurement Conf.*, November 2011.

[106] Gianluca Stringhini, Gang Wang, Manuel Egele, Christopher Kruegel, Giovanni Vigna, Haitao Zheng, and Ben Y. Zhao. Follow the Green: Growth and Dynamics in Twitter Follower Markets. In *Proceedings of the Internet Measurement Conference*, 2013.

[107] That One Privacy Site. https://thatoneprivacysite.net/, 2018.

[108] Kurt Thomas, Danny Huang, David Wang, Elie Bursztein, Chris Grier, Thomas J. Holt, Christopher Kruegel, Damon McCoy, Stefan Savage, and Giovanni Vigna. Framing Dependencies Introduced by Underground Commoditization. In *Workshop on the Economics of Information Security*, Delft, The Netherlands, 2015.

[109] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In *Proceedings of USENIX Security Symposium*, August 2013.

[110] ThreatPost. No Simple Fix for Password Reuse. https://threatpost.com/no-simple-fix-for-password-reuse/118536/.

[111] Tor Project. https://www.torproject.org/, 2018.

[112] IPVanish No-Logging VPN Led Homeland Security to Comcast User. https://torrentfreak.com/ipvanish-no-logging-vpn-led-homeland-security-to-comcast-user-180505/.

[113] Michael Carl Tschantz, Sadia Afroz, Anonymous, and Vern Paxson. SoK: Towards Grounding Censorship Circumvention in Empiricism. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2016.

[114] Giorgos Tsirantonakis, Panagiotis Ilia, Sotiris Ioannidis, Elias Athanasopoulos, and Michalis Polychronakis. A Large-scale Analysis of Content Modification by Open HTTP Proxies. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[115] TunnelBear Completes Industry-First Consumer VPN Public Security Audit. https://www.tunnelbear.com/blog/tunnelbear_public_security_audit/, Aug 2017.

[116] Gareth Tyson, Shan Huang, Felix Cuadrado, Ignacio Castro, Vasile C Perta, Arjuna Sathiaseelan, and Steve Uhlig. Exploring HTTP Header Manipulation In-The-Wild. In *Proceedings of the of the International Web Conference (WWW)*, 2017.

[117] Marie Vasek and Tyler Moore. Identifying Risk Factors for Webserver Compromise. In *International Conference on Financial Cryptography and Data Security*, pages 326–345, 2014.

[118] Bhanu Vattikonda, Vacha Dave, Saikat Guha, and Alex C. Snoeren. Empirical Analysis of Search Advertising Strategies. In *Proceedings of the ACM Internet Measurement Conference*, Tokyo, Japan, October 2015.

[119] Bhanu C. Vattikonda, Santhosh Kodipaka, Hongyan Zhou, Vacha Dave, Saikat Guha, and Alex C. Snoeren. Interpreting Advertiser Intent in Sponsored Search. In *Proceedings of the ACM SIGKDD Conference*, August 2015.

[120] S.J.Res. 34: A joint resolution providing for congressional disapproval under chapter 8 of title 5, United States Code, of the rule submitted by the Federal Communications Commission relating to Protecting the Privacy of Customers of Broadband and Other Telecommunications Services. https://www.govtrack.us/congress/bills/115/sjres34/text, March 2017.

[121] 5 Best VPNs Guaranteed to Beat Netflixs Block in April 2018. https://www.vpnmentor.com/blog/5-best-vpns-netflix-actually-work/, April 2018.

[122] VPNMentor. https://www.vpnmentor.com/, 2018.

[123] David Wang, Matthew Der, Mohammad Karami, Lawrence Saul, Damon McCoy, Stefan Savage, and Geoffrey M. Voelker. Search + Seizure: The Effectiveness of Interventions on SEO Campaigns. In *Proceedings of the ACM Internet Measurement Conference*, Vancouver, BC, Canada, November 2014.

[124] David Wang, Stefan Savage, and Geoffrey M. Voelker. Cloak and Dagger: Dynamics of Web Search Cloaking. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 477–490, Chicago, IL, October 2011.

[125] Yi-Min Wang and Ming Ma. Detecting Stealth Web Pages That Use Click-Through Cloaking. Technical Report MSR-TR-2006-178, Microsoft Research, December 2006.

[126] Yuzhi Wang, Ping Ji, Borui Ye, Pengjun Wang, Rong Luo, and Huazhong Yang. GoHop: Personal VPN to Defend from Censorship. In *Proceedings of the Int. Conference on Advanced Communication Technology (ICACT)*. IEEE, 2014.

[127] Hacked Dropbox login data of 68 million users is now for sale on the dark Web. https://www.washingtonpost.com/news/the-switch/wp/2016/09/07/hacked-dropbox-data-of-68-million-users-is-now-or-sale-on-the-dark-web.

[128] Online activist group Anonymous posts what it says are private contact details for 22 GOP members of Congress. https://www.washingtonpost.com/news/the-switch/wp/2017/08/17/online-activists-anonymous-posts-what-it-says-are-private-contact-details-for-22-gop-congressmen/.

[129] Two rival gamers allegedly involved in Kansas ‘swatting’ death plead not guilty in federal court. https://www.washingtonpost.com/news/morning-mix/wp/2018/06/14/two-rival-gamers-allegedly-involved-in-kansas-swatting-death-plead-not-guilty-in-federal-court/, June 2018.

[130] WebKit. http://webkit.org.

[131] Wikipedia. Squeeze page. https://en.wikipedia.org/wiki/Squeeze_page.

[132] Philipp Winter, Richard Köwer, Martin Mulazzani, Markus Huber, Sebastian Schrittwieser, Stefan Lindskog, and Edgar Weippl. Spoiled Onions: Exposing Malicious Tor Exit Relays. In *Proceedings of the Int. Privacy Enhancing Technologies Symposium (PETS)*. Springer, 2014.

[133] If you want a VPN to protect your privacy, start here. https://www.wired.com/2017/03/want-use-vpn-protect-privacy-start/, 2017.

[134] The attack on global privacy leaves few places to turn. https://www.wired.com/story/china-russia-vpn-crackdown/, 2017.

[135] Baoning Wu and Brian D. Davison. Detecting Semantic Cloaking on the Web. In *Proceedings of the 15th International World Wide Web Conference*, pages 819–828, May 2006.

[136] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. The Dark Alleys of Madison Avenue: Understanding Malicious Advertisements. In *Proceedings of the ACM Internet Measurement Conference*, November 2014.

[137] Qi Zhang, Juanru Li, Yuanyuan Zhang, Hui Wang, and Dawu Gu. Oh-Pwn-VPN! Security Analysis of OpenVPN-based Android Apps. In *Proceedings of the International Conference on Cryptology And Network Security (CANS)*, 2017.

[138] Qing Zhang, Thomas Ristenpart, Stefan Savage, and Geoffrey M. Voelker. Got Traffic? An Evaluation of Click Traffic Providers. In *Proceedings of the WICOM/AIRWeb Workshop on Web Quality (WebQuality)*, pages 19–26, Hyderabad, India, March 2011.