UNIVERSITY OF CALIFORNIA, SAN DIEGO

Understanding the Role of Outsourced Labor in Web Service Abuse

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

in

Computer Science

by

Marti A. Motoyama

Committee in charge:

Professor Stefan Savage, Co-Chair
Professor George Varghese, Co-Chair
Professor Geoffrey M. Voelker, Co-Chair
Professor Gert Lanckriet
Professor Akos Ronas-Tas
Professor Lawrence K. Saul

2011

The dissertation of Marti A. Motoyama is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

____

____

____

____
Co-Chair

____
Co-Chair

____
Co-Chair

University of California, San Diego

2011

DEDICATION

To my family, friends, and loved ones, who all made this journey a little less painful.

EPIGRAPH

Cool story, bro.
—Christopher William Kanich

A mind needs books as a sword needs a whetstone, if it is to keep its edge.
—Tyrion Lannister

I think it is the excitement that only a free man can feel, a free man starting a long journey whose conclusion in uncertain.
—Red

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First off, I'd like to thank my advisors, Professors Stefan Savage, George Varghese, and Geoffrey M. Voelker. Each has provided me with valuable research ideas, feedback and guidance. I will start with Professor Voelker, who taught me the value of conveying one's research results in a simple, understandable manner. He taught me that writing papers is not just about cramming as many graphs, figures, and tables into a paper as possible, but also about interpreting those results; without his input, my papers would be nigh incomprehensible. He also taught me that, as always, I need to increase the size of the font in my graphs. Professors Savage and Varghese both taught me the value of selling one's ideas; each is a talented storyteller, able to wrap any research problem into a compelling narrative that immediately engages the reader. Professor Varghese is always enthusiastic about his work, and that enthusiasm is very infectious. I felt much better about my research pursuits after each conversation I had with him. Professor Savage always has a story to tell, and I am always floored by the amazing breadth of knowledge he possesses. I hope to one day have that much information stored in my brain. Professors Savage and Voelker also supported me as I radically altered my research direction, and for that, I'm grateful. I'm also grateful for the many IRBs that both have submitted on my behalf.

I've had the pleasure of working with a number of spectacular postdocs and fellow graduate students. I'd like to start by first thanking Kirill Levchenko and Damon McCoy, who have both provided me with valuable guidance throughout my academic career. Both have lent me their ears when I needed alternate perspectives on solutions to problems, or when I've needed to express my frustrations with certain projects. Kirill is one smart dude, whose attention to detail and aesthetics still astounds me. Damon is the guy who can get you things; oftentimes, when he hands me something, I'd rather not ask where it came from. I also would like to thank Chris Kanich, who has kept me abreast of the latest Internet memes and provided me with valuable feedback about my research over the years. I would also like to thank Catherine Wah, who, with the same ridiculous hours as me, always stopped by to give me a little "pat pat" when I was feeling down or stressed out about work.

I have a long list of current students to thank; here they are in alphabetical order: Boris Babenko, Neha Chandra, Steve Checkoway, Tristan Halvorson, Alden King, Christos Kozanitis, Xiao Ma, John McCullough, Keaton Mowery, Radhika Mysore, Nima Nikzad, Andreas Pitsillidis, Alexander Rasmussen, Cynthia Taylor, Daniel Turner, David Wang, Kai Wang, Meg Walraed-Sullivan, Ding Yuan, Qing Zhang, and Gjergji Zyba.

I've received much needed advice from various UCSD alumni; here they are in alphabetical order: Yuvraj Agarwal, Jeanne Albrecht, Alvin AuYoung, John Bellardo, Ryan Braud, Yu-Chung Cheng, Diwaker Gupta, Calvin Hubble, Brandon Enright, Evan Ettinger, Samory Kpotufe, Dionysios Logothetis, Jeff Meister, Justin Ma, Shaan Mahbubani, Priya Mahadevan, Ramana Kompella, Marvin McNett, Leo Porter, Barath Raghavan, Frank Uyeda, Patrick Verkaik, Kashi Vishwanath, and Michael Vrable.

I'd also like to thank the other members of my thesis committee, who gave me feedback during the course of my dissertation: Gert Lanckriet, Akos Rona-Tas, and Lawrence K. Saul.

Finally, I would like to thank my family, a quirky bunch of people who have given me all the love and support a graduate student could possibly ask for. I'd also like to thank Audrey Choi, who has stuck by me and offered me moral and emotional support as I've navigated the stressful experience of grad school.

Chapters 1, 2, and 3, in part, are a reprint of the material as it appears in Proceedings of USENIX Security 2011. Motoyama, Marti; McCoy, Damon; Levchenko, Kirill; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

Chapter 1, 2, and 4, in part, are a reprint of the material as it appears in Proceedings of USENIX Security 2010. Motoyama, Marti; Levchenko, Kirill; Kanich, Chris; McCoy, Damon; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

VITA

2005	B.S. in Electrical Engineering and Computer Science, University of California, Berkeley
2009	M.S. in Computer Science, University of California, San Diego
2011	Ph.D. in Computer Science, University of California, San Diego

PUBLICATIONS

Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffery M. Voelker, "An Analysis of Underground Forums." In *Proceedings of the ACM Internet Measurement Conference*, Berlin, Germany, November 2011

Do-kyum Kim, Marti Motoyama, Lawrence K. Saul, and Geoffery M. Voelker, "Topic Modeling of Freelance Job Postings to Monitor Web Service Abuse." In *Proceedings of the ACM Workshop on Artificial Intelligence and Security*, Chicago, IL, October 2011

Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffery M. Voelker, "Dirty Jobs – The Role of Freelance Labor in Web Service Abuse." In *Proceedings of the USENIX Security*, San Francisco, CA, August 2011

Chris Kanich, Neha Chachra, Damon McCoy, Chris Grier, David Wang, Marti Motoyama, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker, "No Plan Survives Contact: Experience with Cybercrime Measurement," In *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test*, San Francisco, CA, August 2011

Marti Motoyama, Kirill Levchenko, Christopher Kanich, Damon McCoy, Geoffery M. Voelker, and Stefan Savage, "Re: CAPTCHAs – Understanding CAPTCHA Solving from an Economic Context," In *Proceedings of the USENIX Security*, Washington, D.C, August 2010.

Marti Motoyama, Brendan Meeder, Kirill Levchenko, Stefan Savage, and Geoffery M. Voelker, "Measuring Online Service Availability Using Twitter," In *Proceedings of the USENIX Workshop on Online Social Networks*, Boston, MA, June 2010.

Marti Motoyama and George Varghese, "I Seek You: Searching and Matching Individuals in Online Social Networks," In *Proceedings of the ACM Workshop on Web Information and Data Management*, Hong Kong, China, October 2009.

Marti Motoyama and George Varghese, “CrossTalk: Scalably Interconnecting Instant Messaging Networks,” In *Proceedings of the ACM Workshop on Online Social Networks*, Barcelona, Spain, August 2009.

Fang Yu, T. V. Lakshman, Marti Motoyama, and Randy H. Katz, “SSA: a power and memory efficient scheme to multi-match packet classification,” In *Proceedings of the ACM Architectures for Networking and Communications Systems*, Princeton, NJ, October 2005.

ABSTRACT OF THE DISSERTATION

Understanding the Role of Outsourced Labor in Web Service Abuse

by

Marti A. Motoyama

Doctor of Philosophy in Computer Science

University of California, San Diego, 2011

Professor Stefan Savage, Co-Chair
Professor George Varghese, Co-Chair
Professor Geoffrey M. Voelker, Co-Chair

Modern Web services are typically free and open access, often supported by advertising revenue. These attributes, however, leave services vulnerable to many forms of abuse, including sending spam via Web-based email accounts, inflating page rank scores by spamming backlinks on blogs, etc. However, many of these schemes are nontrivial to execute, requiring technical expertise and access to ancillary resources (e.g. IP diversity, telephone numbers, etc.). Thus, many scammers prefer to offload the execution of their abuse schemes onto hired labor. This desire to minimize effort has created a demand for workers to carry out malicious tasks. Meanwhile, various online labor marketplaces have emerged that connect employers with cheap, human workers. Abusers have turned to online freelancing sites to find workers willing to carry out numerous schemes. Outsourcing is an attractive option for entrepreneurial scammers, as the workers are typically cheap, technically adept, and exist in vast numbers.

In this dissertation, we investigate how outsourcing impacts the security of Web services; no longer must service providers be wary of automated tools, they must now contend with inexpensive human labor willing to do any menial task. In the first part of the dissertation, we characterize the role of freelance labor in Web service abuse, analyzing over seven years of data from the popular crowdsourcing site Freelancer.com, as well data from our own active job solicitations. We identify the largest classes of abuse work, including account creation, social networking link generation and search engine optimization support, and characterize how pricing and demand have evolved in supporting this activity. We show that scammers heavily employ outsourced labor, with abuse jobs constituting approximately 30% of the job solicitations on the site. Further, we demonstrate that workers quickly adapt their skill sets in responses to changes in demand for various abuse tasks. Lastly, the engagement portion of our study shows that workers actually deliver the promised goods, though the quality of the items is often variable.

The second part of the dissertation focuses exclusively on the role of humans in circumventing CAPTCHAs. Human CAPTCHA solving services represent a heavily commercialized, outsourced abuse task, and we perform an in-depth analysis of this industry. CAPTCHAs are an ubiquitous defense used to protect open Web resources from being exploited at scale. In response to the widespread deployment of CAPTCHAs, a robust solving ecosystem has emerged, selling real-time human labor to bypass these protections. We analyze the behavior and dynamics of CAPTCHA-solving service providers, their price performance, and the underlying labor markets driving this economy. Ultimately, our work shows that CAPTCHAs are effective at differentiating between humans and computers. However, due to the vast number of human workers willing to solve CAPTCHAsfor low wages, CAPTCHAscannot necessarily prevent widespread abuse; instead, they serve as a low-cost economic impediment to abusers.

The results from these two studies demonstrate the increasing role that outsourcing plays in abusing Web services at scale. Furthermore, they suggest that Web services

not only need to consider automated threats, but also must contend with an agile human labor pool. Lastly, they suggest one way to evaluate deployed security mechanisms, by monitoring the price and demand fluctuations for various abusively obtained products.

Chapter 1

Introduction

Advertising-based Web services continue to experience massive growth: Facebook, Twitter, and Hotmail, for example, each command over 750, 105, and 364 million active users, respectively [8, 6, 13]. To attract new customers, many of these Web services are open access, allowing users to register for free while keeping the barrier to entry low. The service can then gain a critical mass of users, subsequently becoming valuable as a platform for advertising. Meanwhile, the services achieve utility by providing consumers with the ability to contribute and share personalized content, in the form of photos, social networking status updates, blogs, etc. However, because the services are free and open-access, they become ripe for exploitation. Attackers can utilize these services in a number of malicious ways; for example, Web services can be used to send email spam [11] to advertise pharmaceutical products, to post backlinks on blogs [2] to artificially enhance search rank, and to spam online classified sites [9] to obtain higher search placement. These activities are termed *service abuse*: attackers achieve financial gains by victimizing Web services. Service abuse generally takes one of two forms: the services themselves are used in the execution of abuse schemes (e.g., sending spam from Web-based email accounts), or they can be exploited to create unsanctioned advertising channels (e.g., using blogs to post backlinks).

To prevent abuse, Web service providers will deploy a variety of defensive measures, from CAPTCHAs, to IP rate limiting, to SMS message verification. Web service providers make every effort to maintain the integrity of their sites to preserve their advertising enterprises. Attackers, however, are rarely dissuaded by technical barriers, and will continue to pursue every avenue possible to maliciously monetize a service. An "arms race" ensues, and Web service providers engage in a perpetual back-and-forth escalation with attackers.

Abusers need to constantly adapt to the increasingly sophisticated countermeasures deployed by Web service providers. Unfortunately, keeping pace with those defensive techniques requires abusers to spend money on resources and devote time to acquiring technical knowledge. To maximize profit, scammers want to minimize all costs associated with reaching their end goals. Meanwhile, the increasing prevalence of the Internet has led to the emergence of online labor marketplaces that connect employers with a globally distributed workforce. Many entrepreneurial scammers have turned to online freelancing sites to find workers willing to carry out their misdeeds. Rather than wasting their efforts on understanding the intricacies of deployed security mechanisms, the abusers can hire cheap, human labor from freelance work sites, where the workers compete with each other to secure jobs. This approach has numerous advantages, the primary being that employers who solicit abuse tasks receive cheap prices on goods. Furthermore, the countermeasures typically deployed by Web service providers are designed with mechanistic automation in mind. By "crowdsourcing" abuse tasks, the attackers bypass the fundamental assumption that software is being used to target Web services. Lastly, attackers can quickly explore different techniques to circumvent defensive measures, since the labor pool is adaptable. The workers will acquire new skills in response to changes in demand for tasks.

The goal of this dissertation is to determine the role of outsourced labor in actualizing Web abuse schemes. To this end, we take two approaches. In the first, we analyze the content of an online freelancing site that connects employers with cheap, globally distributed workers. We identify the types of abuse jobs being solicited, then characterize the pricing and demand trends for these tasks. We also engage with the workforce, assessing the quality of the abusively acquired goods that the workers deliver. In the second approach, we perform an in-depth investigation of a heavily commercialized abuse task, human-based CAPTCHA solving. During the course of this study, we evaluate the CAPTCHA-solving sites from multiple dimensions and characterize the population of workers supporting this industry. We analyze the economic benefits that compel abusers to use outsourced labor to solve CAPTCHAs rather than resorting to traditional software-based approaches. Combined, these two studies highlight the widespread usage of outsourced human labor to abuse Internet services.

Furthermore, by looking at the economic aspects of abuse jobs, we can reason about the economic value of those security mechanisms protecting Web services. Since abuse goods are generally virtual and not manufactured, the price associated with the items typically reflects the difficulty in acquiring the product. As such, not only do prices provide defenders with insight into the extractable value of their products, but they also indicate the efficacy of the defensive measures placed in front of those goods. For example, a phone-verified Craigslist account is expensive at approximately $4.50, whereas a CAPTCHA-protected Yahoo account costs as little as $0.005. The pricing trends can provide feedback to the parties responsible for shaping the defensive strategies of Web services. Without this understanding, it is challenging for companies to reason about the tradeoffs between protecting their assets and implementing increasingly stricter security mechanisms.

1.1 Contributions

In this dissertation, we evaluate the role of outsourced labor in Web service abuse, investigating the types of services commissioned by abusers, the demand for various jobs, the cost of abuse tasks, and the quality of delivered products. By investigating the retail and labor costs of abusively acquired merchandise, we can begin to assess the economic value of the security mechanisms protecting Internet "goods". Absent this knowledge, Web service providers do not have a clear understanding of the impact of their defenses. Rather than assessing the problem of abuse from a purely technical perspective, we approach security from an economic standpoint, looking at the labor costs involved in executing abuse schemes. The contributions of this dissertation are as follows:

- We provide a horizontal view of the abuse market, looking at the role of outsourced labor in facilitating service abuse by analyzing the content of an online freelancing site, Freelancer.com. We characterize the types of jobs present on the site, then investigate the demand and pricing trends for abuse job solicitations. Furthermore, we engage the workforce to assess the quality of the delivered work. We show that scammers outsource a vast and diverse set of jobs, with abusive tasks comprising approximately 30% of the project postings. Further, we show that the workers are technically adept and adaptable, modifying their skillsets in response to changes in job demand. Lastly, we show that the products delivered by the workers are of variable quality.

- We perform an in-depth, vertical analysis of the human-based CAPTCHA solving market, comprehensively characterizing several retail outfits and their corresponding human-labor backends. We describe how using cheap, outsourced labor is a much more financially sensible means of bypassing CAPTCHAs than developing software solutions. Human-based CAPTCHA solving services can provide accurate solutions within 30 seconds, for the small price of $1–2 per 1,000 CAPTCHA solves. Also, many of the services have large solving capacities, with hundreds of workers available to type in CAPTCHA solutions. We characterize the workforce supporting the retail sites, showing that the workers originate from India, China, and Eastern Europe. The conclusion of our study is that due to outsourcing, CAPTCHAs cannot necessarily prevent large scale abuse; instead, they have become an economic impediment to scammers.

Our analysis allows us to paint a broad picture of the Web abuse marketplace, giving service providers insight into emerging threats and how their defenses impact abuse markets.

1.2 Organization

The remainder of this dissertation is organized in the following manner.

Chapter 2 provides background material and related work on both freelancing marketplaces and CAPTCHAs.

Chapter 3 characterizes the role of freelance labor in Web service abuse. We extract over seven years worth of data from a popular outsourcing site Freelancer.com.

Using the data, we identify various types of solicited jobs, and document the pricing and demand trends for abuse services. We present a detailed analysis of the largest abuse classes, which include account registration, social networking link generation, and search engine optimization. Additionally, we engage with the workforce, soliciting bids for abuse jobs and commissioning tasks.

Chapter 4 explores the CAPTCHA-solving industry in depth, an outsourced job briefly touched upon in the previous chapter. We present an in-depth analysis of the industry, detailing how the robust human solving ecosystem emerged in response to the increasing difficulty in deciphering CAPTCHAs via software. We analyze the dynamics of several CAPTCHA-solving service providers, and provide an in-depth analysis of the underlying labor markets.

Finally, Chapter 5 summarizes our work. Additionally, we discuss several future directions for research.

Chapter 1, in part, is a reprint of the material as it appears in Proceedings of USENIX Security 2011. Motoyama, Marti; McCoy, Damon; Levchenko, Kirill; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

Chapter 1, in part, is a reprint of the material as it appears in Proceedings of USENIX Security 2010. Motoyama, Marti; Levchenko, Kirill; Kanich, Chris; McCoy, Damon; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

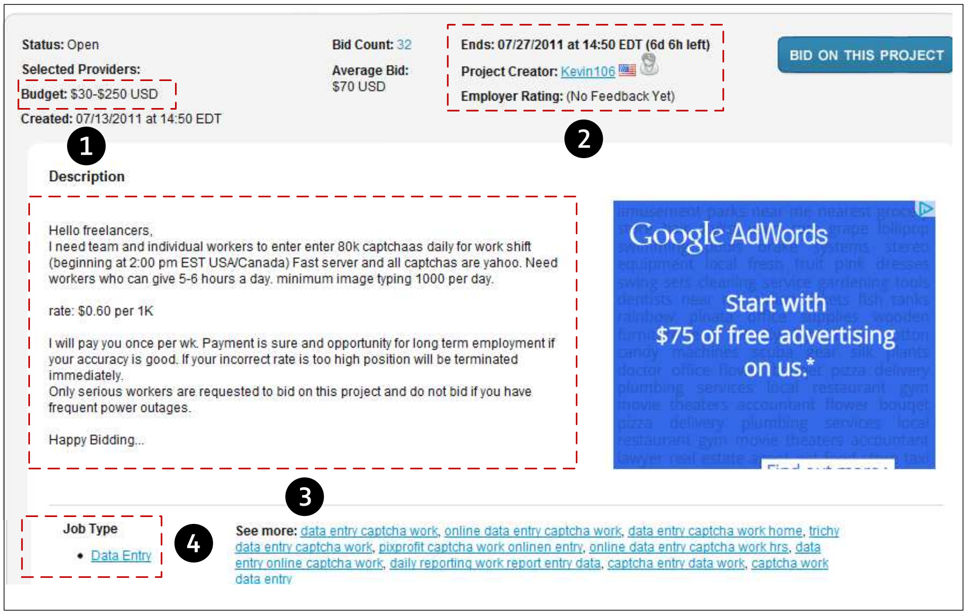
Chapter 2

Background and Related Work

In this chapter, we provide background context on online freelancing sites and the security mechanism known as a CAPTCHA. First, we document the rise of online outsourcing and detail the inner workings of Freelancer.com, a site we study in depth in Chapter 3. Many abusers hire individuals through freelancing Web sites, which connect employers with workers from low-cost labor markets. Next, we provide background details on CAPTCHAs, a defense mechanism used to deter automated abuse software. Scammers have recently begun using outsourced labor to circumvent CAPTCHAs. The human-based CAPTCHA industry exemplifies one of the most refined, commiditized abuse job types.

2.1 Freelancing Sites

The purpose of online freelancing Web sites is to connect employers (typically from high-income countries) with workers from low-wage labor regions. Freelancing sites continue to grow, with employers spending over $70 million in the third quarter of 2010 alone [7] on several top online marketplaces. The sites are very simple to use: employers post jobs and select individual workers based on their bids, skill sets, and prior experiences. Many online freelancing sites exist, with the most popular being Freelancer, Elance, RentACoder, Guru, and oDesk. We specifically examine the activity at Freelancer.com, one of the oldest and largest sites, claiming roughly two million employers and workers [24] from 234 different geographic regions and with close to



Status: Open

Selected Providers:

Budget: $30-$250 USD

Created: 07/13/2011 at 14:50 EDT

Bid Count: 32

Average Bid: $70 USD

Ends: 07/27/2011 at 14:50 EDT (6d 6h left)

Project Creator: Kevin106

Employer Rating: (No Feedback Yet)

Description

Hello freelancers,
I need team and individual workers to enter enter 80k captchaas daily for work shift (beginning at 2:00 pm EST USA/Canada) Fast server and all captchas are yahoo. Need workers who can give 5-6 hours a day. minimum image typing 1000 per day.

rate: $0.60 per 1K

I will pay you once per wk. Payment is sure and opportunity for long term employment if your accuracy is good. If your incorrect rate is too high position will be terminated immediately.
Only serious workers are requested to bid on this project and do not bid if you have frequent power outages.

Happy Bidding...

Job Type

- Data Entry

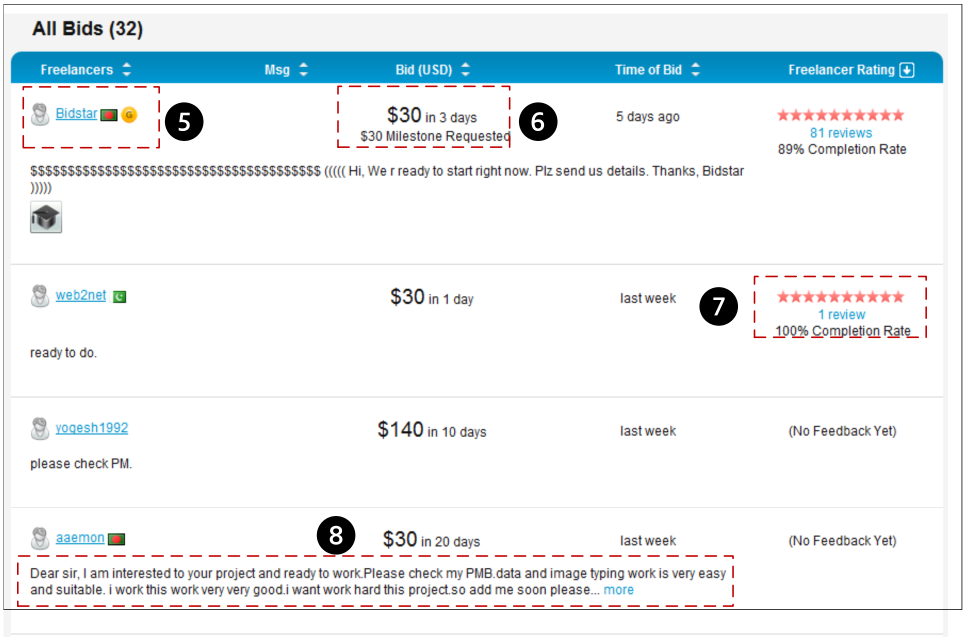
See more: data entry captcha work, online data entry captcha work, data entry captcha work home, trichy data entry captcha work, pixprofit captcha work onlinen entry, online data entry captcha work hrs, data entry online captcha work, daily reporting work report entry data, captcha entry data work, captcha work data entry

Figure 2.1: Screenshot showing a job post on Freelancer.com. The numbers correspond to the following fields: ❶–project budget, ❷–employer username and rating, ❸–project description, ❹–employer selected keywords.

nine hundred thousand projects posted on the site since 2004. We also chose Freelancer because the site offers an open API for querying information about past jobs and users. We have also gathered smaller amounts of data from most of the other large freelancer sites (i.e., via scraping) but since the activity is extremely similar across sites, we chose to focus on the one for which our data was comprehensive.

Visitors to Freelancer must register and select a handle by which they are visible to other users. The only due diligence concerning a user's identity is a requirement to have a valid email address. The site does offer "skills tests" for a fee, by which individual users may demonstrate proficiency in various skills and earn "badges" visible on their profile. There is no discrimination between employers and workers and any user can participate on either or both sides of the labor market.

To post a project on the site, the project poster, or buyer, must pay a $5 fee, which is refunded once a worker is selected. Buyers may choose to pay an additional fee of $14 to have their jobs "featured", meaning that they are listed towards the top of the job listings. Workers independently scan these jobs listings to find projects matching their



All Bids (32)

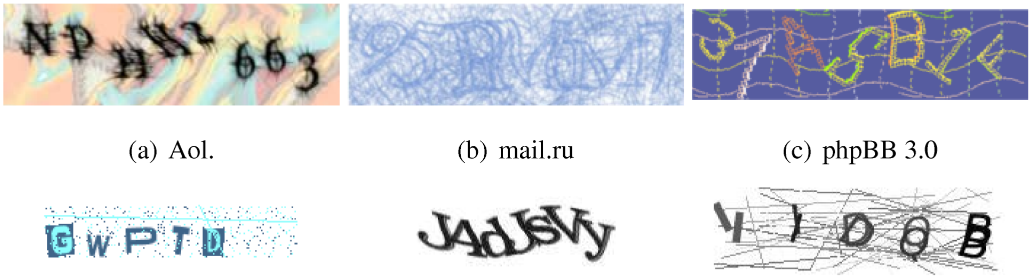
Freelancers	Msg	Bid (USD)	Time of Bid	Freelancer Rating
Bidstar	$$ (((((Hi, We r ready to start right now. Plz send us details. Thanks, Bidstar)))))	$30 in 3 days $30 Milestone Requested	5 days ago	★★★★★★★★★★ 81 reviews 89% Completion Rate
web2net	ready to do.	$30 in 1 day	last week	★★★★★★★★★★ 1 review 100% Completion Rate
yogesh1992	please check PM.	$140 in 10 days	last week	(No Feedback Yet)
aaemon	Dear sir, I am interested to your project and ready to work.Please check my PMB.data and image typing work is very easy and suitable. i work this work very very good.i want work hard this project.so add me soon please... more	$30 in 20 days	last week	(No Feedback Yet)

Figure 2.2: Screenshot demonstrating the various fields for the worker bids on Freelancer.com. The numbers correspond to the following fields: ❺–worker username, ❻–worker bid, typically not reflective of desired salary, ❼–worker rating, ❽–worker message to employer detailing skills, additional messages are sent via private message.

particular skill sets and then place bids (a combination of structured fields, such as dollar amounts, and freeform text). Buyers then select the workers who are most appropriate for their tasks. The fields for buyers and workers are detailed in Figures 2.1 and 2.2, respectively.

Once workers are chosen, Freelancer charges either $3 or 3% of the total project cost to the buyers, depending on whichever amount is higher (Freelancer acts as the middleman in the transaction, using online payment methods such as PayPal, Moneybookers and Webmoney). However, some less scrupulous buyers are reputed to simply cancel their orders and settle with workers out-of-band. Finally, while job postings are effectively "broadcast", there are a range of such posts that identify themselves as private by specifically identifying the workers they are interested in employing.

The simplicity of the system, coupled with the tremendous savings achieved by hiring globally distributed workers, makes these sites incredibly appealing. However, a less appreciated negative impact of this ecosystem is how anonymous access to cheap



(a) Aol.

(b) mail.ru

(c) phpBB 3.0

(d) Simple Machines Forum

(e) Yahoo!

(f) youku

Figure 2.3: Examples of CAPTCHAs from various Internet properties.

aggregated labor impacts the security of existing of Internet services. Indeed, the crowdsourced market for Web service abuse labor is thriving. Our results in Chapter 3 confirm this. While many abuse jobs can be found on Freelancer.com, one in particular has undergone heavy commercialization: human-based CAPTCHA solving.

2.2 CAPTCHAs

To understand human-based CAPTCHA solving, we must provide some background on why CAPTCHAs were initially created. The term "CAPTCHA" was first introduced in 2000 by von Ahn *et al.* [43], describing a test that can differentiate humans from computers. Under common definitions [16], the test must be:

- Easily solved by humans,
- Easily generated and evaluated, but
- *Not* easily solved by computer.

Over the past decade, a number of different techniques for generating CAPTCHAs have been developed, each satisfying the properties described above to varying degrees. The most commonly found CAPTCHAs are visual challenges that require the user to identify alphanumeric characters present in an image obfuscated by some combination of noise and distortion.[1] Figure 2.3 shows examples of such visual CAPTCHAs.

[1] There exists a range of non-textual and even non-visual CAPTCHAs that have been created but, excepting Microsoft's *Asirra* [22], we do not consider them here as they play a small role in the current

The basic challenge in designing these obfuscations is to make them easy enough that users are not dissuaded from attempting a solution, yet still too difficult to solve using available computer vision algorithms. CAPTCHAs are widely used because they are incredibly effective at separating humans from machines; this is valuable in preventing automated abuse at scale. The absence of a CAPTCHA leaves Web services vulnerable to software attacks, which can easily scale out using a distributed infrastructure.

2.3 Related Work

Few academic efforts have analyzed the role of outsourced human labor in abuse tasks. The closest piece of related work is a blog post covering crowdsourced abuse on Mechanical Turk. In contrast, a tremendous amount of literature has been written on the topic of CAPTCHAs. Many researchers in computer vision have developed ways to programmatically decipher CAPTCHAs. Although technically feasible, we argue in Chapter 4 that this approach is not economically viable.

2.3.1 Outsourcing Abuse Tasks

Outsourcing has long been a cost-cutting strategy in developed economies—pushing out key business processes to exploit the efficiencies or lower labor costs of third-party service providers. A more recent innovation is “crowdsourcing”, further unbundling labor from any structured organization and leveraging the broad connectivity provided by the Internet. In this model, individuals participate in the labor force as free agents, responding to open calls for work on a piecework basis. In many cases, crowdsourcing is built on free labor (e.g., for many contributors to open-source projects, or in von Ahn’s seminal ESP game [5]). However, fee-based crowdsourcing sites quickly emerged, the most famous being Amazon’s Mechanical Turk service. Using such services, employers post requests for service at a particular price, while laborers in turn can “solve” the subset of requests that appeal to them. To the best of our knowledge, the only comprehensive investigation of abuse jobs on freelancing/crowdsourcing sites analyzed the “hits” posted on Mechanical Turk [10]. Ipeirotis *et al.* analyzed just 100

CAPTCHA-solving ecosystem.

hits, far fewer than what we cover in Chapter 3. The authors crowdsource the task of categorizing the jobs using just six class types, employing other Mechanical Turk workers to complete the labeling. This methodology is flawed because the workers likely did not possess the requisite domain knowledge to properly identify the chosen categories. Furthermore, the authors categorized the jobs but did not perform any deeper analysis of the commissioned tasks.

Several other studies have looked at other concerns relating to crowdsourcing. First, as an employment vehicle, crowdsourcing is controversial, since critics claim its pure free-market approach to labor has the potential to be highly exploitative, particularly of those in developing countries; one recent analysis estimates that the average hourly wage on Mechanical Turk is $5/hour [28]. Moreover, even on the employer side of the equation, crowdsourcing can be problematic since—absent any strong reputation mechanism—there may be little incentive for workers to provide quality work-products. Consequently, third-party services, such as *crowdflower*, have emerged that trade cost for data quality by replicating work requests and voting among them [3].

2.3.2 CAPTCHA Solving

The issue of CAPTCHA usability has been studied on a functional level—focusing on differences in expected accuracy and response time [15, 41, 45, 50]—but the ultimate effect of CAPTCHA difficulty on legitimate goal-oriented users is not well documented in the literature. That said, Elson *et al.* provide anecdotal evidence that “even relatively simple challenges can drive away a substantial number of potential customers” [22], suggesting CAPTCHA design reflects a real trade-off between protection and usability.

The second challenge, defeating automation, has received far more attention and has kicked off a competition of sorts between those building ever more sophisticated algorithms for breaking CAPTCHAs and those creating new, more obfuscated CAPTCHAs in response [20, 27, 37, 39, 40, 49]. In Chapter 4 we examine this issue in more depth and explain why, for economic reasons, automated solving has been relegated to a niche status in the open market.

Finally, an alternative regime for solving CAPTCHAs is to outsource the problem to human workers. Indeed, this labor-based approach has been commoditized and today a broad range of providers operate to buy and sell CAPTCHA-solving service in bulk. We are by no means the first to identify the growth of this activity. In particular, Danchev provides an excellent overview of several CAPTCHA-solving services in his 2008 blog post "Inside India's CAPTCHA solving economy" [18]. We are, however, unaware of significant quantitative analysis of the solving ecosystem and its underlying economics. The closest work to our own is the complementary study of Bursztein *et al.* [15] which also uses active CAPTCHA-solving experiments, but is focused primarily on the issue of CAPTCHA difficulty rather than the underlying business models.

Chapter 2, in part, is a reprint of the material as it appears in Proceedings of USENIX Security 2011. Motoyama, Marti; McCoy, Damon; Levchenko, Kirill; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

Chapter 2, in part, is a reprint of the material as it appears in Proceedings of USENIX Security 2010. Motoyama, Marti; Levchenko, Kirill; Kanich, Chris; McCoy, Damon; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

Chapter 3

The Role of Freelance Labor in Web Service Abuse

Today's online Web services—search engines, social networks, and the like—create value by helping consumers find and interact with content generated by other users. While these services typically rely on advertising for their revenue, their open access and reliance on user-generated content create powerful opportunities for abusers to fabricate secondary, extremely cheap advertising channels as well. The result is well-known: Web-mail spam, polluted search results, "friend" requests from fake persons and so on. These types of service abuse exploit some feature of a public service for an attacker's financial gain at the expense of the service provider.

Each Web service provider aims to prevent such activities to preserve the integrity of their advertising enterprise. To that end, most Web sites include extensive contracts declaring limits on the way their services may be used. However, implicit threats of legal action rarely deter attackers, and so the provider must rely on a broad range of defenses and countermeasures to enforce their terms of service. While the technical details of this "arms race" are themselves interesting, they are ultimately just symptoms of this larger struggle over controlling who may monetize access to a site's users.

Thus, in this chapter we do not focus deeply on the underlying technical attacks themselves, but rather explore the human labor markets in which these capabilities are provided. Though not widely appreciated, today there are vibrant markets for such abuse-oriented services. Much of this activity occurs on freelance work sites in which buyers “crowdsource” work by posting jobs they need done, and globally distributed workers bid on projects they are willing to take on.[1]

There are multiple advantages in this approach. First, many anti-abuse countermeasures are designed to detect or deter mechanistic automation and can be bypassed through the use of low-cost human labor. Perhaps the best known example of this phenomenon is found in CAPTCHAs, human-solvable puzzles designed to be challenging for automated solvers. While these puzzles are specifically designed to prevent computer-based service abuse, a robust CAPTCHA-solving market has emerged that aggregates large amounts of cheap human labor.

A second advantage is that the crowdsourcing medium allows innovative attackers to quickly explore different schemes for evading anti-abuse defenses (due to the agility of a large contract labor pool). Finally, once a new attack scheme becomes sufficiently popular to commoditize, competitive pressures naturally drive workers to develop the most efficient means of satisfying the demand. Indeed, eventually the most popular activities (e.g., CAPTCHA-solving or phone verified accounts) can support their own branded retail services outside the scrum of the spot labor market.

In this chapter, we characterize the abuse-related labor on Freelancer.com, one of the largest and most popular freelancer sites. Using almost seven years of historic data, and a range of our own contemporary work solicitations, we examine four classes of jobs:

- Account registration and verification,
- SEO content and link generation,
- Ad posting and bulk mailing,
- Social network linking

Each of these represents a kind of service abuse, incorporating manual labor to bypass existing controls, and each is ultimately a building block in some larger, economically-driven, advertising enterprise.

We begin the remainder of this chapter with a description of the data obtained from Freelancer.com.

[1] To be clear, while the majority of such work is legitimate—anything from corporate logo design to software development—a large minority serves the online service abuse ecosystem.

3.1 Data Overview

In this section, we describe our methodology for collecting data on Freelancer job activity and categorizing the jobs into various kinds of "dirty" tasks.[2]

3.1.1 Data Collection Methodology

Freelancer.com exports an API for programmatically querying for information regarding projects and users. Using this API, we implemented a crawler to collect both contemporary and historical information about Freelancer activity. We ran the crawler from December 16, 2010 through April 6, 2011 to minimize load on the site. For historical data, we observed that Freelancer uses monotonically increasing IDs for both projects and users. To crawl all projects over time, we iterated through the entire available project ID space, which at the time ranged from 1–1,015,634. As a result, the job postings in our data set represent all of the jobs that were viewable through the API. We derived the set of user IDs based upon the set of projects, including any user associated with a project as buyer, bidder, or worker.[3]

For all crawled projects, we extracted the project details and the corresponding project bids, as well as the buyer, bidders, and selected workers who were awarded the projects (if any). For all users we encountered, we downloaded their public account metadata and feedback comments.

3.1.2 Data Summary

Starting with the earliest project posted on February 5, 2004 at 12:28 EST, we collected data through April 6th, 2011, capturing over seven years of activity. Table 3.1 summarizes this data set. During this time, 842,199 jobs were posted to Freelancer[4]

[2] While space prevents a detailed description of the oversight and ethical considerations here, our protocols were reviewed by our Human Research Protections Program and we consulted with key brand holders in advance of any active purchasing activity.

[3] Unlike projects, we did not exhaustively collect information for all two million users by crawling the user ID space since the majority of users do not appear to be active on the site.

[4] Note the discrepancy of 173,435 jobs between the maximum ID and the number of postings we obtained through the API. When crawling these IDs, Freelancer's API returned an error indicating that the ID was invalid. We assume that invalid IDs are jobs that never existed or have been deleted—which, according to complaints, happens for only a select number of jobs that egregiously violate Freelancer

Table 3.1: Summary of Freelancer activity between February 5, 2004 and April 6th, 2011.

Activity	Count	
Projects	842,199	
Projects w/ Selected Workers	388,733	(46%)
Project Bids	12,656,978	
Active Users	815,709	
Buyers Only	179,908	(22.1%)
Workers Only	590,806	(72.4%)
Buyer & Workers	44,995	(5.5%)

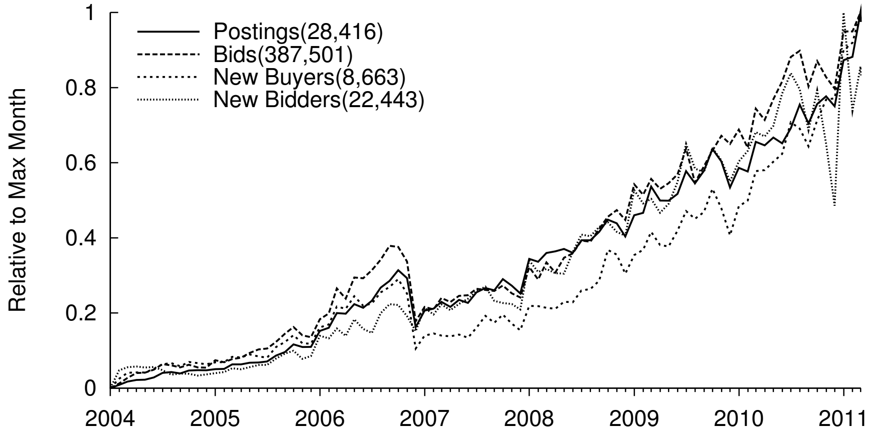


Figure 3.1: Growth in Freelancer activity over time. Numbers in parentheses in the legend denote the largest activity in a month.

and 815,709 users were active on the site. Roughly 46% of the posted jobs report a worker selected for the job. This number represents a lower bound on the number of job transactions; a buyer and a worker will sometimes use Freelancer to rendezvous, but will negotiate the transaction through private messaging and, thus, never report a selected worker. Among all users associated with at least one project, 22.1% were buyers only, 72.4% were bidders/workers only, and 5.5% served as both.

Activity on Freelancer has grown steadily over time. Figure 3.1 shows the number of jobs offered and the number of bids made per month, as well as the number of new buyers and bidders per month. To overlay and compare the curves, we normalized them to their maximum monthly value as listed in Table 3.1. The curves all show a drop

rules.

Table 3.2: Distribution of 2,000 random, manually-labeled projects into job categories. Referenced sections of the appendix include examples of jobs in the corresponding category.

Category	Job Type	Description	Count	%
Legitimate [§A.1.1]	Web Design/Coding	Create, modify, or design a Web site	769	38.5
	Multimedia Related	Complete multimedia-related task	265	13.2
	Private Jobs	Jobs designated for a particular worker	138	6.9
	Desktop/Mobile Apps	Create a desktop or mobile application	100	5.0
	Legitimate Miscellaneous	Miscellaneous jobs	177	8.8
Accounts [§A.1.2]	Account Registrations	Create accounts with no requirements	22	1.1
	CAPTCHA Solving	Requests for human CAPTCHA solving	19	0.9
	Verified Accounts	Create verified accounts (e.g. phone)	14	0.7
SEO [§A.1.3]	SEO Content Generation	Requests for SEO content (e.g., articles)	195	9.8
	Link Building (Grey)	Get backlinks using greyhat methods	53	2.6
	Link Building (White)	Get backlinks using no greyhat methods	20	1.0
	SEO Miscellaneous	Nonspecific SEO-related job postings	61	3.0
Spamming [§A.1.4]	Ad Posting	Post content for human consumption	25	1.2
	Bulk Mailing	Send bulk emails	8	0.4
OSN Links [§A.1.5]	Create OSN Links	Get friends/fans/followers/etc.	33	1.7
Misc [§A.1.6]	Abuse Tools	Tools for abuse (e.g., CAPTCHA OCR)	41	2.1
	Clicks/CPA/Signups	Get clicks, emails, zip codes, signups, etc.	32	1.6
	Manual Data Extraction	Manually scrape website content	21	1.1
	Gather Contact Lists	Research contact details for people	17	0.9
	Academic Fraud	Write essays, code assignments, etc.	10	0.5
	Reviews/Astroturfing	Create positive reviews	1	0.1
	Other Malicious	Misc. jobs with malicious intentions	35	1.8

in activity in December 2006 (the reason for which we have not been able to determine). After this point, Freelancer experiences strong linear growth in buyers and bidders and their associated posting and bidding activity. Freelancer's job market is healthy and growing. Work posted by a steadily increasing number of buyers (5,000 new buyers a month on average in 2010) has been satisfied by an equally steadily increasing supply of bidders (15,000 new bidders/month).

3.1.3 Categorizing Jobs

Our first step in understanding Freelancer activity is to categorize the types of jobs found on the site. We use a two-step process for this categorization. We first manually browse sampled projects to identify a meaningful list of job categories. We then use a combination of keyword matching and supervised learning to identify jobs from the entire Freelancer corpus that fall into the categories.

From browsing random job postings, gauging the interest level in various tasks from observed bidding activity, and incorporating awareness of the larger underground cybercrime ecosystem, we identified 22 types of jobs falling into six categories. To establish a baseline of the prevalence of these types of jobs, we manually inspected a random sample of 2,000 jobs, tagging each job with a category.

Table 3.2 summarizes the list of categories and the distribution of jobs that fall into each category from our random sample. Note that a job may be tagged under multiple categories; for example, social bookmarking jobs for search engine optimization (SEO) usually also require account creation. Legitimate projects comprise 65.4% of these jobs and primarily involve Web-related programming and content creation tasks. We include private jobs, corresponding to projects targeted to one specific user, in the legitimate job class, since we typically do not know the job details; private postings, however, will sometimes contain enough data to determine their intent. In our manually labeled corpus, we were unable to determine the intent of 5.4% of the jobs. The remaining 29.2% of the jobs correspond to various kinds of "dirty" jobs, ranging from delivering phone-verified Craigslist accounts in bulk to a wide variety of search-engine optimization (SEO) tasks.

We then focused on identifying jobs in the entire Freelancer corpus that fall into "dirty" categories. Since we could not manually classify all jobs, we used keyword matching to generate training sets and supervised learning to train classifiers for each category. We then applied the classifiers to each job to determine the dirty category it falls into, if any.

To find positive examples for each classifier, we used keywords associated with the job type to conservatively identify jobs that fall into each category. For example, to locate jobs about CAPTCHA solving, we searched job postings for the terms "CAPTCHA" and "type" or "solve". For negative examples, we randomly chose jobs from the other orthogonal job types. For features, we computed the well-known *tf-idf* score (term frequency-inverse document frequency) of each word present in the title, description, and keywords associated with jobs in the training sets. We then used *svm-light* [30] to train classifiers specific to each category.

Table 3.3 shows the results of applying these classifiers to all Freelancer jobs.

Table 3.3: Freelancer jobs categorized using the classifiers.

Class	Job Type	Count	%
Accounts	Account Registrations	6,249	0.7
	Human CAPTCHA Solving	4,959	0.6
	Verified Accounts	3,120	0.4
SEO	SEO Content Generation	72,912	8.7
	Link Building (Grey Hat)	16,403	1.9
	Link Building (White Hat)	10,935	1.3
Spamming	Ad Posting	11,190	1.3
	Bulk Mailing	3,062	0.4
OSN Linking	OSN Linking	11,068	1.3

We focus on just those dirty job categories that had at least 1,000 jobs. Although the classifiers are not perfect (e.g., some jobs placed in the “link building” categories might be better placed in the more generic “SEO” category), they sufficiently capture the set of jobs in each category and greatly increase the number of jobs we can confidently analyze. Note that we did not attempt to be complete in the categorization of the postings: there are likely jobs that should be in a category that we have missed. However, such jobs are also likely not well-marketed to workers, since they most likely lack the typical keywords and phrases commonly used in jobs under those categories.

We focus on the jobs comprising these categories in the analyses we perform in the subsequent sections.

3.1.4 Posting Job Listings

Pricing information is a crucial aspect of our study, since it represents the economic value of an abusive activity to attackers. Both job descriptions and bids contain pricing info, often at odds with each other. To determine which source of pricing info to use, we performed an experiment where we posted jobs on Freelancer and solicited bids. In the process, bidders posted public bids and, in some cases, sent private messages to our user account. These private messages occasionally reveal the external Web store fronts operated by Freelancer workers, in addition to the tools, services, and methods they use to complete each type of job. We posted 15 job listings representative of the categories for which we have classifiers. We also randomly posted half of the jobs as a "featured" listing to determine whether this increased the quantity of bids we received (which it did).

Table 3.4 summarizes the results of our job posting experiments. Of the 228 total bids we received, 47 were commensurate with market rates for these projects. Most of the remaining bids, however, were simply minimum bids used as "place holders". The actual bid amount was either included in a private message to our buyer account, or the bidder provided an email address to negotiate a price outside of the Freelancer site to avoid the Freelancer fee.

Because many prices in the public bids severely underestimate market prices, we use the prices in job descriptions by buyers in our studies in Section 3.2. Even so, we note that the pricing data has some inherent biases. They are advertised prices and not necessarily the final prices that may have been negotiated with selected workers. Further, we use prices that were systematically extracted from the job descriptions. Even with hundreds of hand-crafted regular expressions, we were only able to extract pricing data from about 10% of the jobs. Job descriptions are notoriously unstructured, ungrammatical, and unconventional. These biases notwithstanding, the pricing data is still useful for comparing the relative value of jobs, as well as trends over time.

3.2 Case Studies

This section features case studies of the four groups of abuse-related Freelancer jobs summarized in Table 3.2.

3.2.1 Accounts

Accounts on Web services are the basic building blocks of an abuse workflow. Because they are the main mechanism for access control and policy enforcement (e.g., limits on number of messages per day), circumventing these limits requires creating additional accounts, often at scale. Thus account creation has become the primary battlefield in abuse prevention.

Accounts primarily enable a wide variety of spamming and scamming. For Web

Table 3.4: Results from posting job listings to Freelancer. A “*” indicates the post was featured, the number within the “()” is the number of bids that included prices. All prices in the cost column are for the smallest unit of service (i.e., per one account, backlink, email, post, and 500-word article).

Class	**Job Type**	**Bids**	**Cost**
Accounts	Craigslist PVA	10 (4)	$4.25
[§A.2.1]	Gmail Accounts	6 (5)	$0.07
	Hotmail Accounts*	21 (12)	$0.007
	Facebook Accounts*	24 (10)	$0.07
SEO	Blog Backlinks*	10 (5)	$0.30
[§A.2.2]	Linking (White Hat)*	17 (8)	$0.81
	Forum Backlinks	12 (9)	$0.50
	Social Bookmarks*	44 (21)	$0.13
	Bulk Article Writing	29 (23)	$3.00
Spamming	Bulk Mailing	10 (5)	0.075¢
[§A.2.3]	Craigslist Posting	10 (3)	$0.60
OSN	Facebook Friends*	11 (4)	$0.026
Linking	Facebook Fans	5 (5)	$0.039
[§A.2.4]	MySpace Friends	2 (2)	$0.037
	Twitter Followers*	7 (6)	$0.02
Software [§A.2.5]	ReCaptcha OCR*	7 (1)	$2,000
	Yahoo OCR	3 (1)	$180

mail services like Gmail and Yahoo, spammers use accounts to send email spam, taking advantage of the reputation of the online service to improve their conversion rate. For online social networks like Facebook and Twitter, spammers use accounts to spam friends and followers (Section 3.2.2), taking advantage of relationships to improve conversion. For classified services like Craigslist, spammers use accounts to create highly-targeted lists, post high-ranking advertisements for a variety of scams, recruit money laundering and package handling mules, advertise stolen goods, etc. Further, accounts on some services easily enable paired accounts on related services (e.g., creating a YouTube account from a Gmail account), further extending the opportunities for spamming.

Table 3.5: Summary of the results from purchasing email accounts. The names of the account sets embed the worker countries: IN is India, UK is the United Kingdom, BD is Bangladesh, PK is Pakistan, and CA is Canada. The rating column refers to the average rating of the selected worker. *Notes:* *We purchased $\texttt{IN}_1$ in 2010, the rest in 2011. **The worker responsible for $\texttt{CA}_1$ repeatedly copied and pasted 508 accounts to meet the 5k requirement.

Name	**Rating**	**Tested**	**Valid (%)**	**Age (Days)**
$\texttt{IN}_1$*	9.8	500	100.0	0.4
$\texttt{UK}_1$	9.9	3,500	99.9	25.7
$\texttt{BD}_1$	10	6,999	99.6	24.7
$\texttt{IN}_2$	9.8	5,015	99.6	9.7
$\texttt{PK}_1$	10	4,999	99.4	78.6
$\texttt{PK}_2$	9.8	4,000	95.4	82.6
$\texttt{PK}_3$	9.9	4,013	77.3	414.7
$\texttt{IN}_3$	9.9	6,200	76.2	30.7
$\texttt{CA}_1$**	9.6	508	15.7	21.7

Account Creation Insights

In the context of another research effort, we obtained approval from a major Web mail provider to purchase fraudulently-created accounts on their service. We purchased 500 such accounts from a retail site selling accounts, gave them to the provider, and in return received registration metadata for the supplied email accounts, including account creation times and the IP addresses used to register the accounts. We later discovered that the supplier we contacted was a very active member of Freelancer.com; this worker is responsible for account set $\texttt{IN}_1$ in Table 3.5.

The supplier had bid on 2,114 projects, had been chosen as a selected worker on 147 projects, and served as a buyer on 84 projects. Interestingly, the supplier acted as a buyer for 25 jobs that involved the creation of other Web mail account types. The supplier contracted out this task at a rate of $10–20 per 1,000 accounts, and yet the supplier charged $20 per 100 accounts on the retail Web site, an order of magnitude more.

The accounts we purchased were created an average of only 2.8 seconds apart, suggesting the use of either automated software or multiple human account creation teams in parallel.[5] Such automation would be one way to earn money bidding on account jobs for this particular worker. Further, 81% of the IP addresses used to register the accounts were on the Spamhaus blacklist, suggesting the use of IP addresses from compromised hosts to defeat IP-based rate limiting of account creation.

Experience Purchasing Accounts

In 2011, we commissioned a job to purchase additional email accounts for the same Web mail provider in quantities ranging from 3,500–7,000. We selected nine different workers, of which eight ultimately produced accounts, listed in Table 3.5 after $\texttt{IN}_1$. Once given the accounts and the corresponding passwords, we logged into the accounts and downloaded the newest and oldest inbox pages (assuming the account was valid). Table 3.5 shows the results of the purchasing and account checking. Of the eight email sets, seven consisted of largely valid accounts, with over 75% of the tested email accounts yielding a successful login. $\texttt{PK}_3$ was particularly interesting; the worker previously used the email addresses to create Facebook and Craigslist logins and posts, then resold the accounts to us. Also, four of the account batches are relatively old (as determined by the date of their oldest emails), with the median age of the accounts between two months and over one year. These ages indicate that workers are likely sitting upon a stockpile of email accounts. Lastly, the worker ratings do not seem to reflect the quality of the accounts, as demonstrated by the high ratings (out of 10) achieved by those workers responsible for the $\texttt{PK}_3$, $\texttt{IN}_3$, and most notably, $\texttt{CA}_1$ account sets.

CAPTCHA Solving

To keep the barrier to participation extremely low, creating an account at an online service today requires little more than solving a CAPTCHA. CAPTCHAs are designed to be hard to solve algorithmically, and thus create an obstacle to automating service abuse. In response to their widespread deployment, human-based CAPTCHA-solving services emerged in abuse ecosystem. Such services depend on cheap human labor to provide a simple programmatic interface for solving CAPTCHAs to an otherwise

[5] We know that an effective automated CAPTCHA solver existed at this time for this Web mail provider, so automation is the likely suspect.

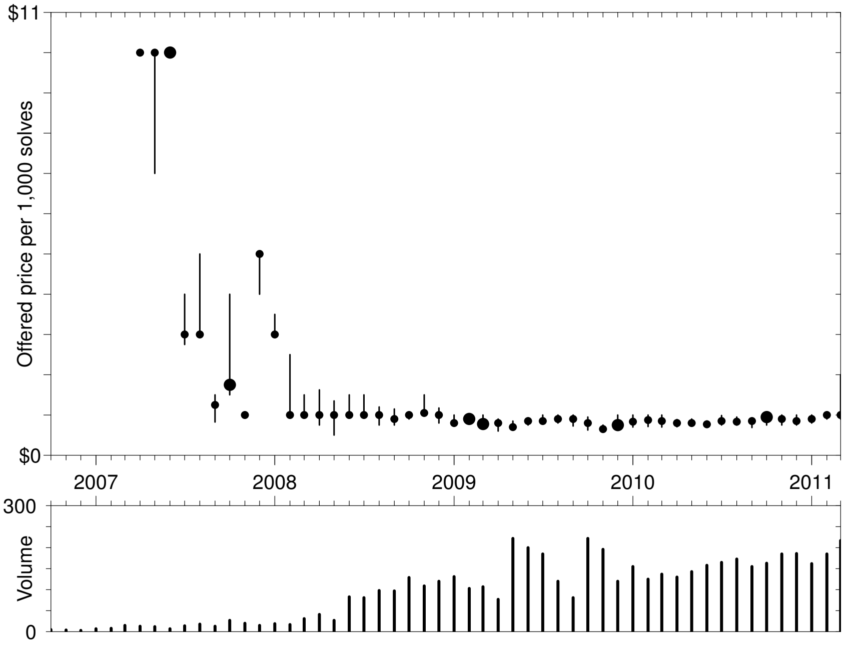


Figure 3.2: Median monthly prices offered by buyers for 1,000 CAPTCHA solves (top) and the monthly volume of CAPTCHA solving posts (bottom), both as functions of time. The solid vertical price bars show 25% to 75% price quartiles.

completely automated abuse processes chain. In a previous study [38], we described a robust retail CAPTCHA-solving industry capable of solving a million CAPTCHAs a day at $1 per 1,000 solved. Thus today, CAPTCHAs are neither more nor less than a small economic impediment to the abuser, forming the first step in the account value chain.

By their nature, CAPTCHAs are ideally suited to the Freelancer outsourcing paradigm, and indeed the Freelancer marketplace has played a key role in the evolution of CAPTCHA solving. Figure 3.2 shows the history of prices offered for CAPTCHA solving as well the demand (in number of job offers per month) since 2007. We see a rise in demand starting from their first appearance, and a corresponding drop in prices to the $1 per 1,000 price seen today, corroborating our previous findings [38].

Account Verification

Because creating a basic account—even one requiring solving a CAPTCHA—is so cheap, to curb online abuse services must necessarily take advantage of some limited resource available to a user. To increase the limits placed on a basic account, a user must sometimes undergo *account verification*, which takes a variety of forms (e.g. phone numbers, credit cards, etc.). Verification increases the user's standing within the service, giving the account holder greater access to the service and thereby increasing the value of the account. For this reason, verification is a step in the value chain of many abuse processes.

The most popular type of verified account uses phone verification. Beyond the steps for creating a basic account, phone-verified accounts (PVAs) require a working phone number as an additional validation factor in account authorization. Services will either call or message a code to the number, and the user must submit the number back to the service to complete authorization. For some services phone verification is mandatory (e.g., for posting advertisements in certain forums on Craigslist, creating multiple accounts in Gmail from the same IP address), and for other services, phone verification adds convenience (e.g., avoids CAPTCHAs with Facebook). Services typically require the phone number to be associated with a landline or mobile phone since, unlike VoIP phone numbers, it is much more difficult to scale the abuse of such numbers. Phone verification is effective: immediately after Gmail introduced phone verification to limit account abuse, for instance, prices for Gmail accounts on underground forums skyrocketed to 10 times other Web-mail accounts [4]. However, even more so than CAPTCHAs, PVAs add further delay and inconvenience to users and is the primary reason why services do not use phone verification uniformly.

Web Services Targeted

Figure 3.3 shows the distribution of services targeted in job postings for basic and verified account registrations. For ease of comparison, it shows the top 10 targeted services for both kinds of accounts, combined. For a job targeting multiple services, we count it in the total for each service mentioned. Job postings target accounts in every major category of Internet service: Web mail, social networks, as well financial and marketplace services. However the distribution of specific services differ markedly between the two types of account registration jobs, reflecting how services vary in their deployment of additional verification mechanisms (if any). Basic accounts are useful

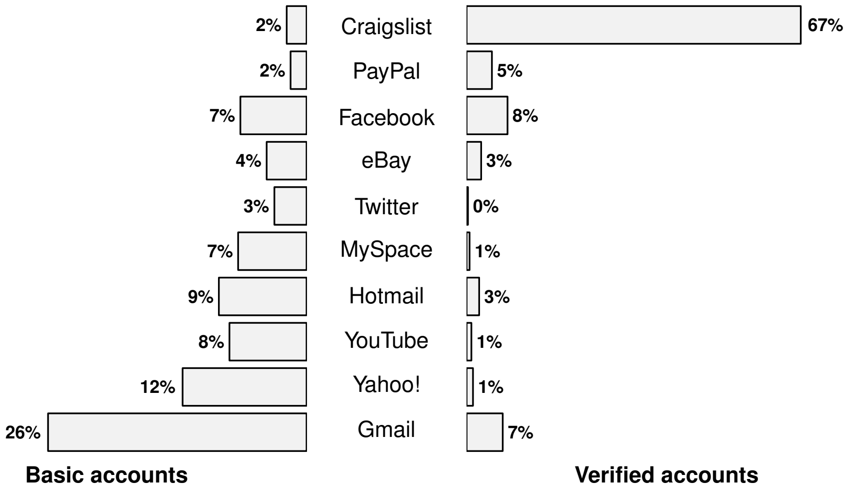


Figure 3.3: Sites targeted in account registration jobs.

for many purposes, including obtaining accounts up for other Internet services (Facebook, Craigslist, etc.), and Gmail is by far the most popular. When it comes to verified accounts, on the other hand, Craigslist is the dominant target, most certainly because Craigslist sections targeted by spammers all require PVAs.

We posted a job soliciting bids for “CraigsList Phone Verified Accounts PVA” on Freelancer.com. Of the 10 bids we received, 4 contained prices: $3, $4, $4.50, and $6. These prices are consistent with the currently observed buyer offers for Craigslist PVAs. The pricing of PVAs tells us in monetary terms the value of phone verification as a security mechanism. For Craigslist, PVAs have made account abuse extremely expensive. In contrast, retail services sell Gmail PVAs for around 25¢, a 10–20 fold price difference compared to Craigslist.

Trends

Demand for accounts through Freelancer grew dramatically starting mid-2008. Figure 3.4 shows the number of account creation jobs posted over time. Demand for basic accounts steadily increased through mid-2008, then dramatically increased until it peaked in mid-2009.

Demand for verified accounts rose greatly when Craigslist introduced phone verification for the erotic services section of their site in early March 2008 [14]. Demand

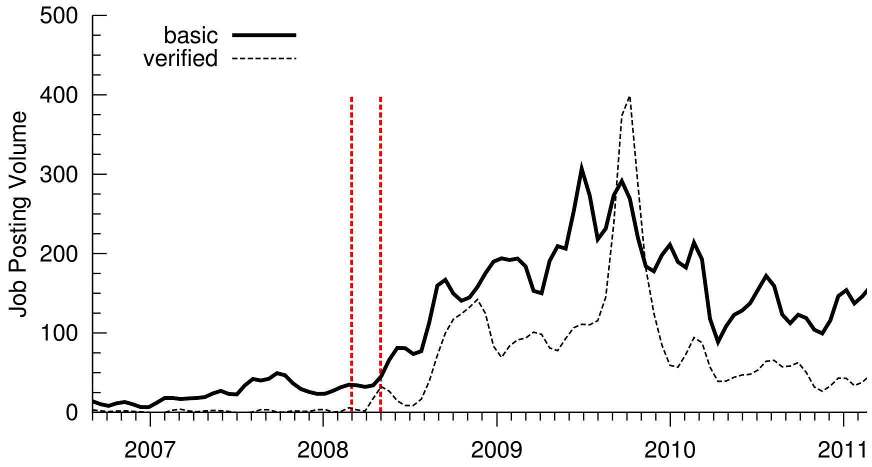


Figure 3.4: Demand for account registration jobs over time. The dashed vertical lines indicate approximate dates when Craigslist introduced phone verification for erotic services ads (March 2008) and other services (May 2008).

grew steadily until about October 2009, and then dropped. We extracted prices from the Craigslist postings, and observed that Craigslist PVAs first rose to $4 by the end of 2008 and then settled around $2. In October of 2009, prices spiked to more than $5, then hovered between $2 and $3 through 2010.

For both types of accounts—basic and verified—demand dropped during 2010. We do not know the cause; however we suspect this may be due to stricter policing on behalf of Freelancer.com; our own price solicitation for Craigslist posting was canceled by the site.

3.2.2 OSN Linking

Online social networking links can be abused in two ways: (1) as a communication channel to market to real users, which is a finished product ready to directly monetize; (2) as an intermediate product to increase the reputation—and thus influence—of accounts by adding social links to other fake accounts. Previous work has shown that online social networking spam has a higher click-through rate than traditional email-based spam [26]. Thus, OSN platforms have emerged as a lucrative marketing venue where spammers are exploiting the trust relationships that exist in social networks to improve their conversion rates. However, it is difficult for a spammer to contact users on a social networking site until they have established a *social link* with real users. These social links take many different forms, depending on the targeted social networking site, such as convincing a user to friend the spammer, follow a spammer's Twitter feed, become a fan of the spammer's page, or subscribe to the spammer's YouTube channel. Building social links to real users is analogous to gathering email addresses that will later by monetized with email spamming. Once this social link is established, the spammer has a communication channel that is both highly reliable and not subject to aggressive filtering.

Adding fake social links is a relatively inexpensive method for increasing the reputation of an account, which in turn presumably improves the success rate of establishing links to real users. This method is effective because people are more willing to establish or accept social links that are more popular in terms of the number of previously-established social links or other endorsements. If the account has many social links and, more importantly, if mutual social links exist, the likelihood increases that the targeted real user will establish or accept a social link with the spammer.

In this section we survey the Freelancer.com market for buying both real and fake bulk social links.

Characterization

There are two main categories of social networking links requested in jobs. The first are friendship relationships (e.g., MySpace and Facebook friends), where an active invitation is offered and, if accepted, targeted messages can then be delivered to a user's private inbox. The second are subscription relationships (e.g., Facebook fans, Twitter followers, YouTube subscribers) where, if a user can be induced to follow a spammer's account, messages will appear in a user's feed; depending on the site, the relationship also grants the ability to send private messages to the user. A closely related goal is to use social links to increase the perceived popularity of an object. Examples of this type of task are increasing the view count of YouTube videos, or digging links on Digg. We group all these jobs into the category of social network links and they all follow the form of increasing the reputation of an account/object or establishing a marketing channel to real users.

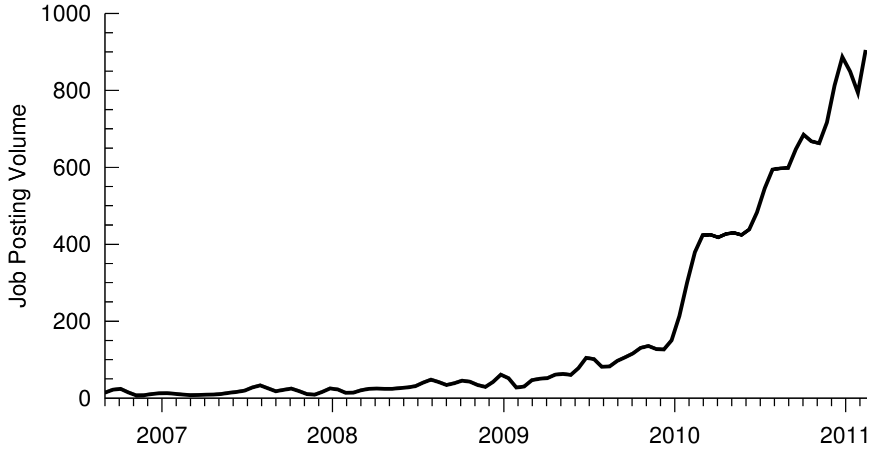


Figure 3.5: Number of job postings for social networking links.

Jobs for bulk social link building range from a few hundred to hundreds of thousands of links. Typically jobs interested in acquiring fake social links will request a relatively small number of links spread out over a large number of accounts (e.g., add 500 friends to 50 accounts). The requests for social links to real users often specify a target demographic for the links, thereby exploiting the same targeted marketing potential of using information included in a profile that legitimate advertisers on these sites also use to improve ad targeting. For example, a job might require that most social links be to male accounts in the US over the age of 18. The most targeted geographic demographics are high-income English speaking countries including the US (46%), UK (13.2%), Canada (9.5%) and Australia (6.2%). Also, based on keyword searches, females are specifically targeted in 8% of jobs and males in 3% of the jobs.

Trends

Figure 3.5 shows the demand over time for job postings for social networking links. Overall demand for social links has skyrocketed since the early part of 2010, suggesting that spammers have only recently realized the potential for monetizing social links. The social networking sites with the largest English-speaking user bases (Facebook, MySpace, Twitter, and YouTube) are targeted by 97% of the job postings for social links. Over 50% of social link jobs included words such as “real” and “active” indicating that they were seeking to buy a more finished type of social link that could be directly spammed. This percentage is a lower bound, however, as it is unclear how

Table 3.6: Summary of the social links purchased to pages for our custom Web sites. The names of the sets correspond to the selected workers' home countries, while the rating column refers to his or her average rating. The worker responsible for BD$_7$ did not complete the job in a timely manner. Country codes: BD – Bangladesh, IN – India, RO – Romania, MY – Malaysia, PK – Pakistan.

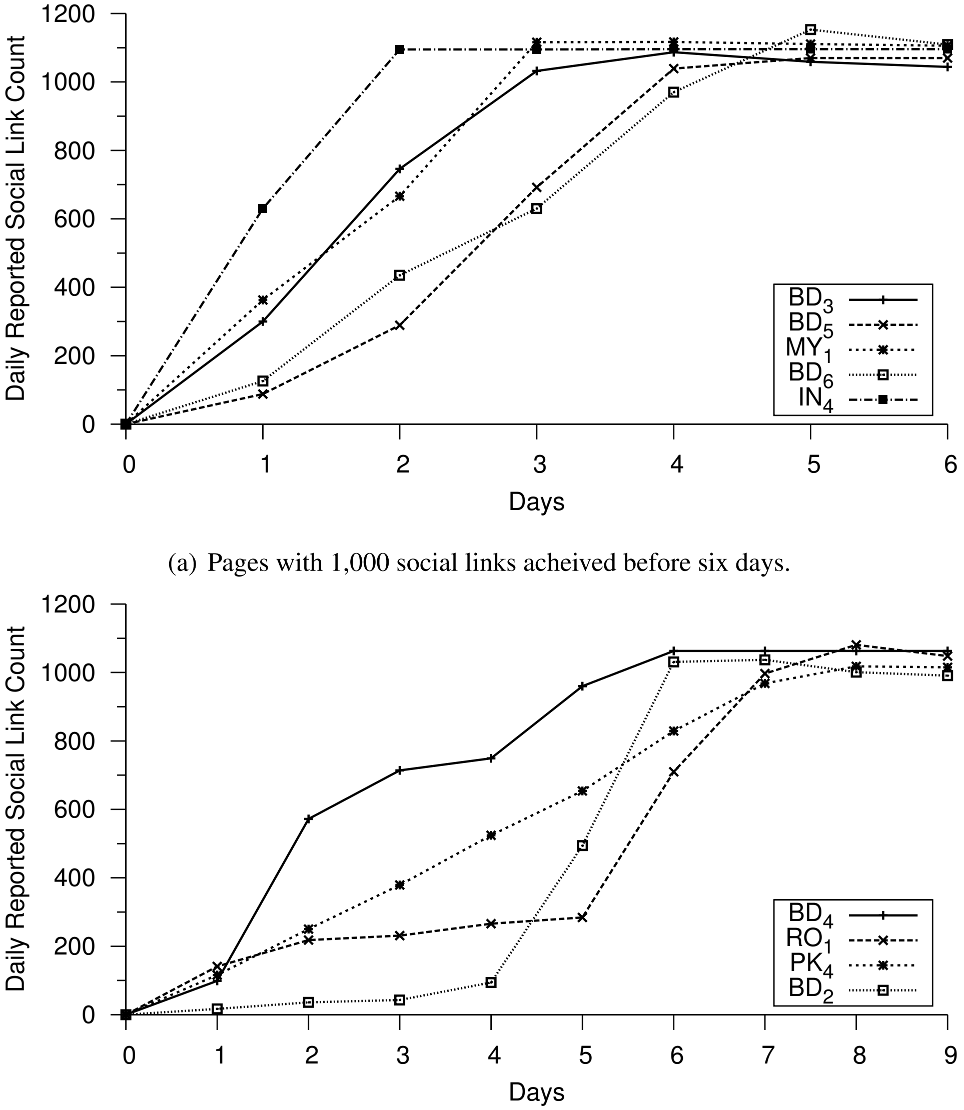
			Top Countries (%)			
Name	**Rating**	**Links**	US	IN	BD	PH
BD$_2$	9.8	1,034	26.2	13.8	5.9	7.7
BD$_3$	9.8	1,081	43.3	7.4	32.5	4.4
BD$_4$	8.4	1,063	74.5	0.3	25.2	—
BD$_5$	10	1,071	—	—	100	—
BD$_6$	10	1,145	60.0	8.7	8.4	5.3
BD$_7$*	9.8	555	30.6	10.4	10.6	8.4
IN$_4$	9.9	1,095	64.3	25.1	10.5	—
MY$_1$	9.8	1,110	99.1	—	—	0.1
PK$_4$	–	1,015	24.7	9.2	5.9	7.0
RO$_1$	10	1,058	31.8	11.0	8.8	8.4

many postings did not include these types of words but were actually seeking real social links.

Overall the median offered price in posts were $0.01 per social link, and median bids were between $0.02–0.03 per a social link. These prices were similar across all of the social networking sites. This low price point raises the interesting question of whether proposed defenses that mitigate Sybil attacks via analysis of social link structure [51, 52] might be vulnerable to adversaries that are willing to simply hire humans to create real social links.

Experiences Purchasing Social Links

In preparation for purchasing social links, we instantiated several Web sites on the topic of cosmetics consulting [53] and created separate "pages" about each site on a popular social networking service. We then commissioned a job to obtain one thousand social links for these pages. The posted job explicitly targeted users from the US, Canada, and the UK. We assigned the task to 10 different workers, each given a different



(a) Pages with 1,000 social links acheived before six days.

(b) Pages completed after six days

Figure 3.6: Social link growth rate. The first day a social link appears is counted as Day 1.

Web site to target.

Table 3.6 shows the results of this task. The name of the sets correspond to the selected workers' home countries, and the links column is the maximum reported daily number of social links. Most of the workers delivered the required number of social links in a timely manner (except for the $\texttt{BD}_7$ set); the quality of the social links, however, was quite poor. Most of the workers did not deliver social links from users that met

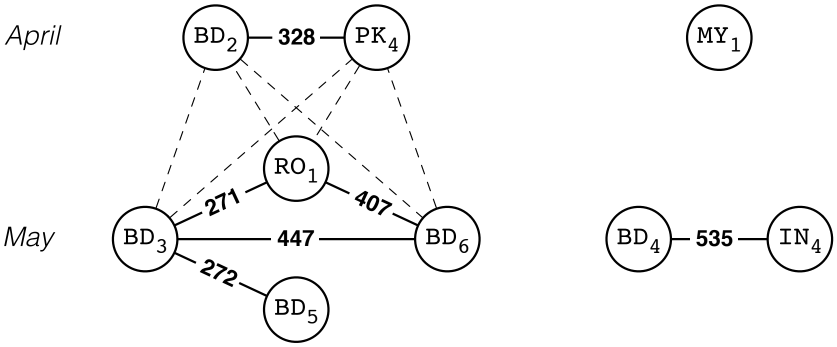


Figure 3.7: The number of user accounts common to each pair of workers hired to create social links. Labeled solid lines indicate at least 100 user accounts (out of 1,000 requested) in common, dashed lines indicate at least 10 but fewer than 100 user accounts in common. Work performed by $\mathtt{MY}_1$, $\mathtt{PK}_4$, and $\mathtt{BD}_2$ was done in April, while the remaining jobs were done roughly a month later.

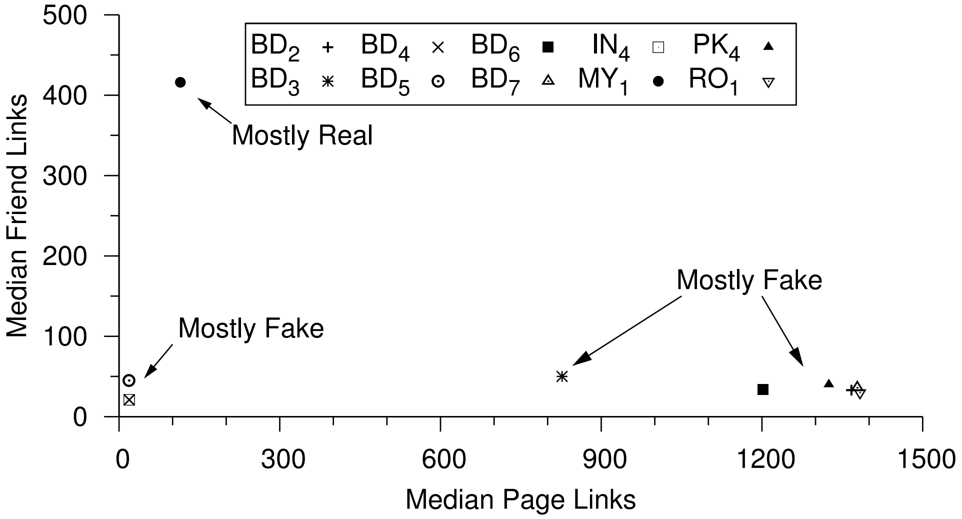


Figure 3.8: Median number of friends vs. median number of page social links for the sets of users linked to our websites.

our specifications, particularly in regards to user countries. Also, several of the workers added social links at a rapid pace, as shown in Figure 3.6. Worker $\mathtt{IN}_4$ (see Figure 3.6(a)) completed the job in as few as two days, adding social links at a rate of over 500 per day. This rapid growth rate would serve as a strong indicator that the page owners purchased their social links. Likewise, $\mathtt{PK}_4$ in Figure 3.6(b) has a linear growth rate, which is highly suspicious and indicates the use of an automated tool.

Next, we observed substantial overlap between the users linked to our target pages, shown in Figure 3.7. As many as 50% of the users (between $\mathtt{IN}_4$ and $\mathtt{BD}_4$, for example) overlapped. This overall suggests that the workers are all manipulating the same set of users to produce these social links, or even perhaps subcontracting out the task to the same groups of workers. Only one worker, responsible for $\mathtt{MY}_1$, had no overlap with any of the other sites. Again, the selected worker ratings do not reflect the quality of the delivered products; we posit that buyers who hire these workers find it difficult to evaluate social link quality.

We extracted the profiles for the OSN users who were linked to our target Web sites, and looked at the number of friends and page links listed on their profiles. Figure 3.8 shows a scatterplot of the median number of friends versus the median number of page links for these OSN users. Several clusters emerge in the graph. Within each user batch, we manually visited the profiles of those users; only one worker, $\mathtt{MY}_1$, appears to have delivered social links from legitimate users. The rest used predominately fake accounts, many of which had few friends and a large number (>1,000) of page social links.

3.2.3 Spamming

In our study, we consider spamming to be the dissemination of an advertiser's message to users by means other than established advertising networks. Spamming provides the buyer with a direct marketing channel to his targets, and as such, represents one of the most finished commodities in the advertising value chain.[6]

In our survey and classifier-based labeling (Tables 3.2 and 3.3), the class of spamming jobs is comprised of ad posting and bulk mailing.[7] Because Craigslist is the main target of ad posting jobs (82%), we treat it separately. We begin by first analyzing the pricing data for bulk mailing.

[6] The most finished commodity is actual site traffic; however, traffic of reasonable quality (with respect to conversion rate) usually requires site-specific targeting and additional advertiser-provided material ("creatives").

[7] While we found several other kinds of spam-like jobs (e.g., bulk SMS), they did not represent a significant fraction of all jobs, and are not part of our study.

Bulk Mailing

Bulk mailing is simply traditional email spam and represents 0.3–0.4% of all jobs posted on Freelancer.com. In most cases, the buyers supply their own mailing lists, although some—generally targeting larger volumes—expect bidders to supply their own address lists.

We extracted pricing data from the job descriptions of 236 postings. We averaged these prices and discovered that buyers on Freelancer.com were willing to pay approximately $5.62 to send 1,000 emails, with a median price of $1.00. The extracted prices varied wildly; thus, we manually scanned another 100 random postings. Again, we observed a wide range of prices, from one buyer willing to pay only $0.06/1,000 emails, to another buyer willing to pay $5.00/1,000 emails.

A final point of comparison is our own posting for bulk mailing services. We posted a job that involved sending bulk emails to three million individuals and received 10 responses. Of the 10 responses, five included a price, and these prices ranged from $0.30 to $2 per 1,000 messages (with a median of $0.75/1,000 emails).

Craigslist Ad Posting

Posting an ad on Craigslist is typically free, but Craigslist takes special measures to restrict the number of ads posted by a single individual (e.g., IP rate limiting, CAPTCHAs, etc.). In the context of our study, when Freelancer.com buyers create jobs to "spam" Craigslist, their goal is to obtain *repeated* ad postings from workers, usually on a daily basis. This is done to keep a buyer's ads at the top of the search results. Our classifier identified 11,190 job postings of this type, 9,096 (81%) of which contained the service name "Craigslist" or a variation thereof (in total comprising 1.1% of all jobs on Freelancer.com).[8]

Figure 3.9 shows the prices offered by buyers for a single Craigslist posting (top) and the average number of job posts per day pertaining to Craigslist ad posting (bottom). The solid circles indicate monthly median prices, and the solid bars show the 25% to 75% quartiles of the prices. In early March 2008, Craigslist added a phone

[8] Classified ad sites BackPage and Kijiji represented 6.6% and 5.5% of jobs classified as ad posting; we chose to focus on Craigslist because it dominated this job category.

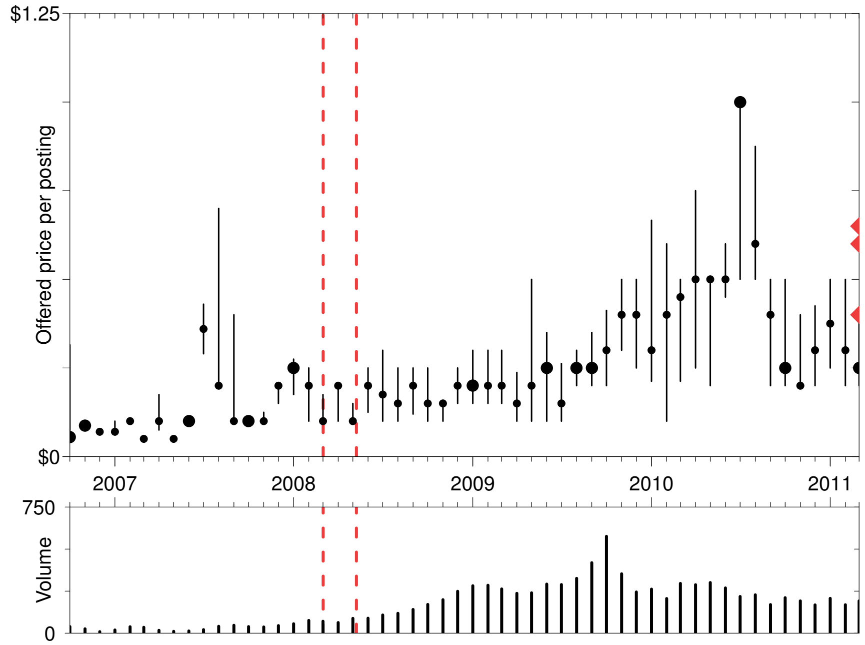


Figure 3.9: Median monthly prices offered by buyers for each Craigslist ad posted (top), and the monthly number of posts (bottom), both as a function of time. The solid vertical price bars show 25% to 75% price quartiles. The dashed vertical lines indicate approximate dates when Craigslist introduced phone verification for erotic services ads (March 2008) and other services (May 2008). The three bids received in response to our solicitation are indicated with a triangle on the right edge.

verification requirement for posting in the erotic services section [14], and later extended the requirement to posting in other parts of the site some time in early May 2008 (both dates indicated with dashed vertical lines in the graph).

Figure 3.9 illustrates that the demand for *posting* to Craigslist started growing gradually after the policy changes, and the prices offered by buyers stayed essentially unchanged until mid-2009. Recall that in mid-2009, demand for phone verified accounts (which are dominated by Craigslist) appears to drop dramatically (Figure 3.4), having increased rapidly over the past year. Note, however, that the demand for Craigslist ad *posting* continues to rise during that same time period, nearly quadrupling in price within a year.

To further compare pricing data, we posted a job description on Freelancer soliciting bids for "Experienced Craigs List Posters." We received 10 responses, with three bids of $0.40, $0.60, and $0.65 per ad; these prices are shown for comparison purposes in the top graph of Figure 3.9 as solid triangles on the right edge. These prices are roughly in accordance with the buyer offers.

3.2.4 Search Engine Optimization

Search engine optimization (SEO) represents the second major advertising channel along with spam. SEO is a multi-billion dollar industry for improving the ranking of sites and pages returned in search results on popular search engines. Improving the ranking of pages in search results increases traffic to that page. “White hat” SEO improves the search rank of pages while obeying the guidelines provided by search engine companies like Google that prevent abuse of the indexing and ranking algorithms. “Black hat” SEO abuses the indexing and ranking algorithms, sacrificing the relevance of a page with the sole goal of attracting traffic via search results.

There are three kinds of black hat SEO offerings on Freelancer.com, spanning the spectrum from least to most “finished”: content generation, link building, and search placement. Content generation increases the number of sites that contain indexable content together with links to a target page. This goal is achieved either by having writers generate unique content for sites, often by rewriting existing material, or by using a semi-automated technique known as *spinning*. Spinning often uses structured templates together with a variety of word, phrase, and sentence “dictionaries” to generate many variants of effectively the same content, and is analogous to the template-based techniques used to generate polymorphic spam that can defeat spam filters [36].

Link building is a more focused type of SEO job whose goal is to place links on pages with existing content, emphasizing placement on pages with high rank as defined by search engines. Rather than generating and distributing content across many sites as a basis for improving the ranking of a target page, link building bootstraps on existing highly-ranked pages.

The most finished kind of SEO job is search placement. The buyer does not care *how* the desired search placement is achieved, only that they place in the top search results on Google. Such jobs were relatively rare on Freelancer, and we only survey content generation and link building jobs in further detail.

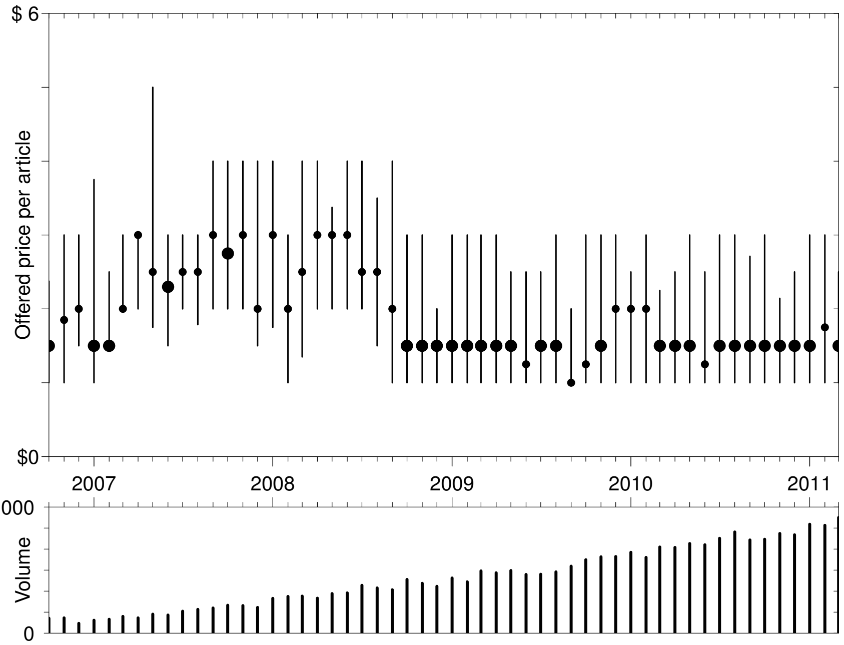


Figure 3.10: Median monthly prices offered by buyers for each article posted (top) and monthly average number of buyer posts per day (bottom), as a function of time. Vertical price bars show 25% to 75% price quartiles.

Content Generation

A popular form of abusive SEO is to post “articles” to various sites and forums. These articles contain keywords and links intended to increase the search engine PageRank of a page in search results returned from queries that use the same keywords. With proper accounts (Section 3.2.1), the posting step can be automated. However, defenses implemented by search engines can detect automatically-generated article content. Such defenses have thereby created a demand for human workers to generate sufficiently realistic articles that defeat the countermeasures. Indeed, such article writing jobs represent the most popular abusive job category by far, accounting for over 10% of all Freelancer jobs (Table 3.2).

Article job descriptions request batches of 10–50 articles at a time in grammatically correct English on a particular topic, seek articles typically 250–500 words in length, and often have a variety of requirements that reflect perceived countermeasures implemented by the search engines. A frequent requirement is sufficient “originality” (albeit often of simply rewritten text) to pass CopyScape, a popular plagiarism detection

Table 3.7: Quality of articles written by workers on the topic of skin care products. Columns *Len*, *KD*, and *CS* show how many of each worker's six articles failed the length, keyword density, and CopyScape plagiarism detection requirements. The *FKGL* column shows the Flesch–Kincaid Grade Level [34] range of each worker's text after excluding their lowest and highest scoring articles. Country codes: PH – Philippines, IN – India, BD – Bangladesh, KW – Kuwait, UK – United Kingdom, US – United States, AU – Australia, KE – Kenya.

		Failed articles			
ID	Rating	Len	KD	CS	FKGL
$\mathtt{IN}_5$	9.50	–	6	–	8.8 ± 1.0
$\mathtt{PH}_1$	9.75	4	5	–	7.7 ± 0.9
$\mathtt{BD}_8$	–	–	4	–	8.1 ± 0.7
$\mathtt{KW}_1$	9.62	–	3	–	10.0 ± 0.3
$\mathtt{IN}_6$	9.62	–	2	–	7.2 ± 0.8
$\mathtt{UK}_2$	10	–	1	2	9.0 ± 0.5
$\mathtt{US}_1$	10	–	1	–	8.6 ± 0.2
$\mathtt{BD}_9$	9.81	–	1	–	9.3 ± 0.5
$\mathtt{AU}_1$	–	–	1	–	11.0 ± 1.0
$\mathtt{KE}_1$	10	–	–	–	9.6 ± 1.0

tool; such originality counters the capability of search engines to detect and discount similar content. Other such requirements request rewritten text beyond straightforward manipulation of existing content (simple synonym substitution, transposing sentences, etc.).

Figure 3.10 shows buyer demand and offered prices over time for article content generation jobs. Growth in demand for articles has been strong, with the number of jobs offered increasing linearly, with a peak of nearly 3,000 article jobs posted in August 2010. This substantial growth in demand strongly suggests that article writing is indeed an effective form of SEO abuse. Yet prices for articles have been relatively stable over the past four years, with buyers offering $2–4/article.

Experiences Purchasing Articles

To evaluate the quality of the articles written by Freelancer workers, we solicited and employed ten workers to write six original articles on the topic of skin care products.

We required each article to contain at least 400 words, have a keyword density of at least 2%,[9] and pass the CopyScape [17] plagiarism detection system. Table 3.7 shows the results of this assignment. Workers are identified by their two-letter country and a digit. In addition to the three criteria above, we also computed the articles' Flesch–Kincaid Grade Level [34]—a measure of text readability based on word and sentence length, roughly indicating the school grade level required to comprehend the text. The FKGL column shows the score range of the work produced by each worker after excluding their lowest and highest scoring articles.

Quality of the work produced by the ten workers varied considerably. More than half of the articles produced by workers $\mathtt{IN}_5$, $\mathtt{PH}_1$, and $\mathtt{BD}_8$ did not meet our 2% keyword density requirement; in addition, $\mathtt{PH}_1$ failed to produce articles of the required length (400 words). On the other hand, half of the workers produced articles satisfying our criteria in at least five out of six cases. Unfortunately, two of the articles produced by $\mathtt{UK}_2$ did not pass the CopyScape plagiarism detection tool, and as such, would likely not be indexed by search engines.

Articles written by the workers were understandable and on topic. The Flesch–Kincaid Grade Level of the articles reveals a notable level of English composition. For comparison, five Wikipedia articles on the same topic had scores in the range 12.1 ± 0.5, while six articles from Cosmopolitan—a popular women's magazine in the US—about skin care fell in the 7.9 ± 0.8 range. Thus, at least with respect to SEO, our results show Freelancer to be a useful source of inexpensive content that would be difficult to distinguish mechanically from work produced by more highly-paid specialist writers.

Link Building

Google reports a PageRank (PR) metric for every page, accessible via the Google Toolbar. The PR ranges from 0–10, with new and least popular pages having a PR of 0 and the highest ranked pages having a PR of 10. This PageRank is a combination of the number of sites that link to the page—so-called backlinks—and the PageRanks of the

[9] "Keyword density" is the frequency of occurrence of a set of keywords provided by the bidder to be included in the text. Keyword density thresholds ensure that search engines index a Web page with respect to the specified keywords. In our experiment, we provided workers with keywords such as "dry skin moisturizer" and "exfoliating scrub".

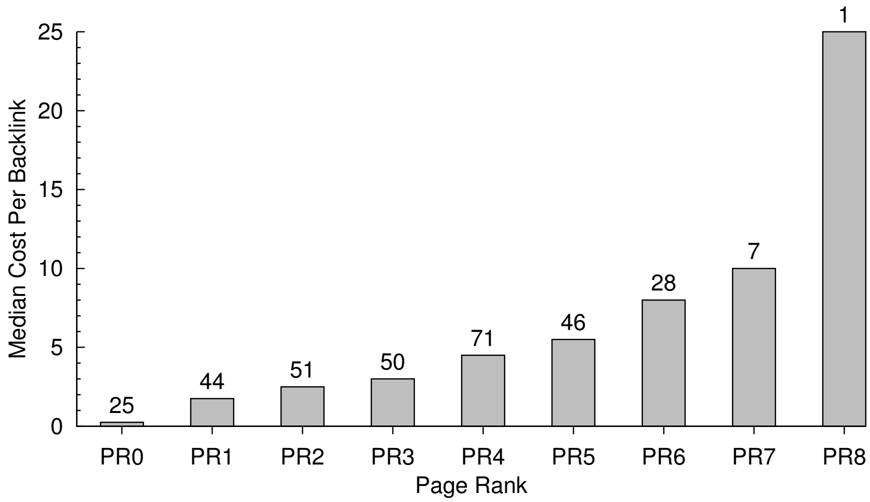


Figure 3.11: Average price buyers offered for backlinks on pages with a given PageRank (PR). Higher PRs correspond to more popular (and valuable) pages. The number above the bar corresponds to the number of jobs requesting backlinks of that PR.

pages with the backlinks. Not surprisingly, another common SEO abuse is to increase the number of sites that backlink to a page, and to have those backlinks on sites with high PageRank.

Hiring people to perform this kind of SEO task is another frequent kind of abusive job on Freelancer, accounting for over 3% of all jobs. We placed such link-building tasks into two categories, “white hat” and “grey hat”. White hat link building jobs have requirements that specifically try to avoid search engine countermeasures, such as no link farms, no blacklisted sites, no redirects or JavaScript links, links on sites with generic top-level domains, and so on. Jobs also specify the PageRank of the pages on which the backlinks will be placed, and that the buyer will validate the links created according to all of their criteria. Grey hat link building is much more indiscriminate, such as spamming blogs with links embedded in comments.

How much do people value backlinks as an SEO technique? The job postings quantify this value in economic terms. For the “white hat” link building jobs for which we could automatically extract pricing data, Figure 3.11 shows that the median price per backlink buyers offered is directly correlated with the PageRank (PR) of the page containing the backlink. One buyer offered over $25 per backlink on pages with PR8, while buyers offered nearly $5 per backlink on PR4 pages, the most popularly-requested

Table 3.8: Summary of top 10 targeted domain names for greyhat link purchasing.

Domain Name	Num. Sites	Num. Inlinks
Blogspot	316	10,028
Wordpress	213	2,402
Yahoo	147	1,187
ArticlesBase	143	747
Folkd	108	302
ArticleSnatch	107	491
Google	97	184
Squidoo	88	154
Diigo	88	277
ArticleAlley	88	471

PR.

Next, we look more closely at buyers who posted “grey hat” link building jobs, or ones that allow for such questionable SEO methods as blog commenting, forum posting, etc. For these Freelancer job postings, buyers oftentimes directly specify the URL that they are interested in using greyhat techniques on. We extracted over two thousand URLs that were present in the body of the greyhat link building posts. Using Yahoo Site Explorer [48], we checked the first 1,000 inlinks (restricted by the API) pointing to each URL. Then, we filtered URLs with more than 1,000 inlinks remaining (i.e., not retrievable via the Yahoo API), yielding 813 sites. Table 3.8 shows the top domain names for the inlinks. As expected, Blogspot and Wordpress are highly targeted for link spamming. Yahoo Answers and Groups, as well as Google Knol and Google Sites, are also targeted.

3.3 User Analysis

We end our investigation of Freelancer activity by surveying the geographic demographics and job specialization of Freelancer users. We also touch on the reputation and lifetime of Freelancer users.

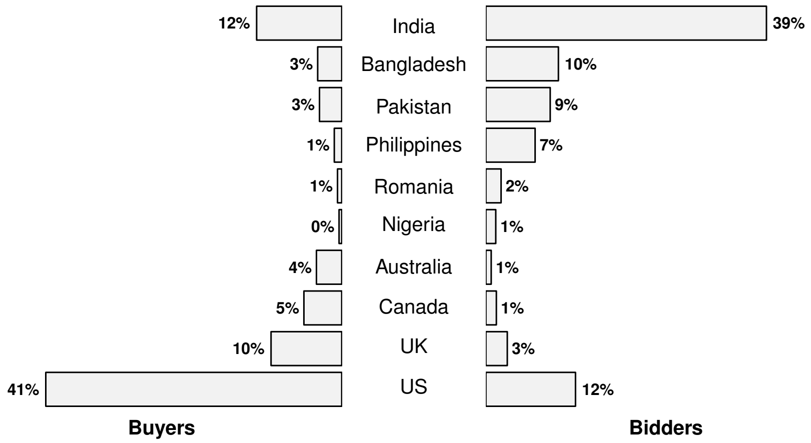


Figure 3.12: Distributions of countries for buyers and bidders.

3.3.1 Country of Origin

There are clear demographic differences between buyers and bidders. Figure 3.12 shows the distribution of countries of origin for all buyers and bidders of the abuse-related jobs categorized in Table 3.2. (The distribution for selected workers closely follows the overall bidder distribution.) We extract the country of origin for users from their profile information. We note that this information is self-reported and nothing prevents users from being dishonest; further, we have seen instances where buyers post jobs specifically avoiding bidders from India, for instance, providing a potential motive for dishonesty. Numbers for such countries are therefore a lower bound.

The largest group of buyers is from the United States, and other English-speaking countries feature prominently (UK, Canada, Australia, even India). In contrast, the largest group of bidders is from India, followed by neighboring Pakistan and Bangladesh—countries with a large cheap labor force, substantial Internet penetration, and where English is an official language or has widespread fluency.

The country of origin demographics for each category reveals yet more detail. Figures 3.13 and 3.14 show the top five countries of buyers and bidders, respectively, for each abusive job category in Table 3.2. Buyers for advertisement posting (generally targeting Craigslist, Section 3.2.3) are primarily from the United States, whereas, somewhat surprisingly, buyers for human CAPTCHA solvers are primarily from Bangladesh and India—these are buyers looking to form teams of solvers. Bidders from India and

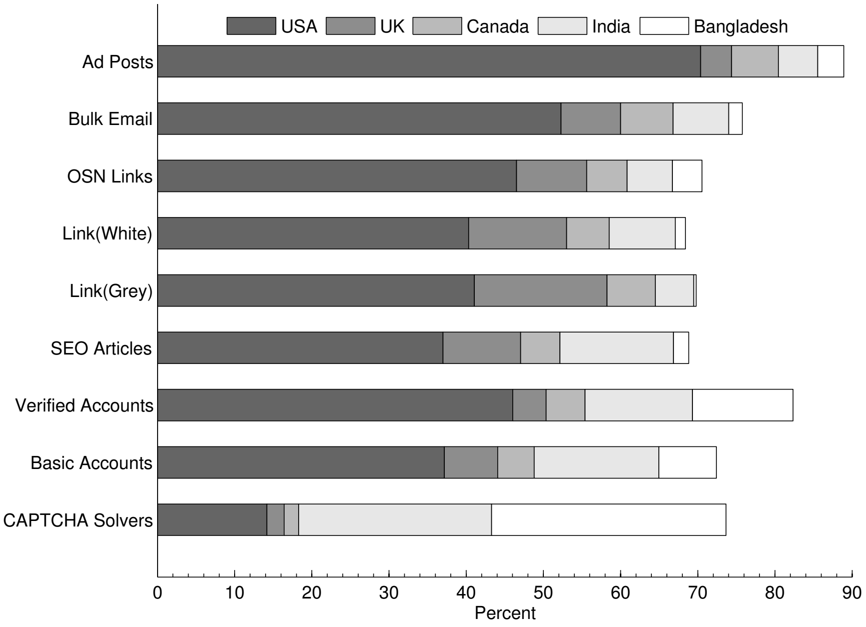


Figure 3.13: Top five countries of buyers posting abusive jobs.

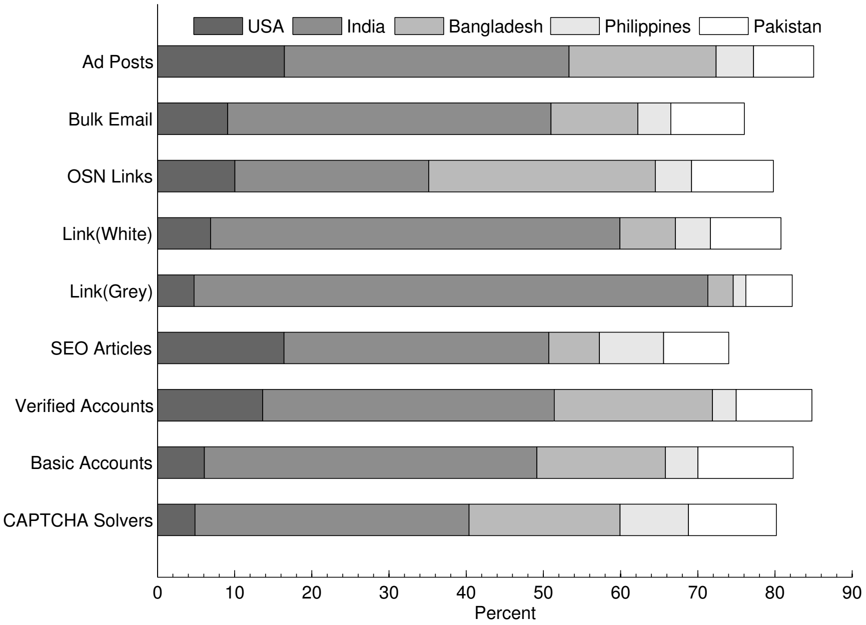


Figure 3.14: Top five countries of workers bidding on abusive jobs.

Bangladesh dominate white hat and social networking link building jobs, respectively. Bidders from the only Western country (US) in the top five target article generation, creating PVAs, and advertisement posting.

3.3.2 Specialization

Aside from some uniform basic fundamental requirements, such as understanding English and having access to and basic knowledge of the Internet, the abuse jobs posted on Freelancer essentially require unskilled labor. As a result, Freelancers need not necessarily specialize—focus solely on a particular job category—in the tasks that they undertake.

As one metric of whether specialization occurs or not, we examined whether buyers and bidders participated in more than one category of job (for those buyers and bidders who engaged in more than one job). Indeed, bidders clearly do not specialize. For all but one category, on average fewer than 5% of the jobs that bidders bid on are within the same category; the exception is article content generation, where nearly 15% of bids per bidder are on other article jobs. Moreover, not only are most bids on other job categories, but the majority of bids are on jobs that did not even fall into an abuse category in Table 3.2. In other words, for bidders who bid on at least one abuse job, 70–80% of their other bids were for a non-abuse job.

Buyers follow a similar pattern as bidders, but are slightly more focused: 10% of a buyer's jobs, on average, are for jobs in the same category, while 60–70% of a buyer's jobs were for a non-abuse job. Article content generation again is the one exception, with 30% of a buyer's jobs requesting articles.

3.3.3 Reputation and Lifetime

Briefly, we also surveyed user reputation, based upon feedback on buyers and bidders, as well as the "lifetime" of activity for a user on Freelancer.

For buyers, there was no difference in the distribution of reputation of buyers for abuse jobs and non-abuse jobs. Bidders on abuse jobs, though, had worse reputations than those on non-abuse jobs: about 20% of bidders for abuse jobs had the lowest reputation score, whereas only 10% of bidders for non-abuse jobs had such a reputation. Perhaps bidders for abuse jobs are willing to act unscrupulously more often.

Buyer user IDs have much longer lifetimes than bidder user IDs on Freelancer. Nearly 50% of buyers are active on Freelancer for at least four months, and 20% for

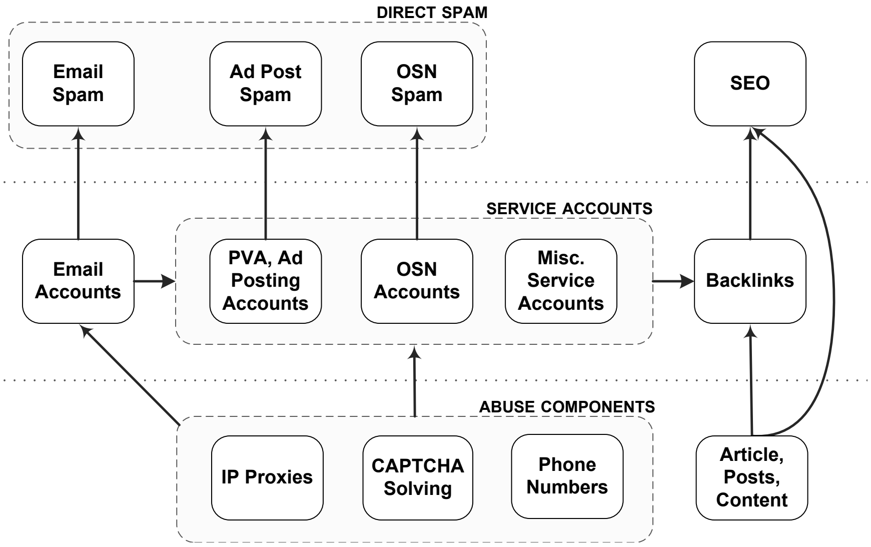


Figure 3.15: How the various elements of the market fit together

more than two years. In contrast, 50% of bidders are active on Freelancer for only a month or more, and 20% for more than half a year.

3.4 Discussion

Figure 3.15 illustrates how the various markets described in this study fit together in the Web abuse chain. At the lowest level, workers need access to Web proxies (due to account registration limits placed on IP addresses), CAPTCHA solvers/OCR packages, and phone numbers. Utilizing these components, abusers can create Web-based email accounts, the primary building blocks for service abuse. The email accounts can be used to register accounts for a number of Web services, including Craigslist, Facebook, Twitter, Digg, etc.

The abusers can then implement various monetization schemes with the accounts, most of them involving “spamming”. The most direct form of spamming utilizes the Web email accounts to send spam. Craigslist PVAs allow abusers to post repeated, daily advertisements, making a retailer’s product consistently appear near the top of the search results. Abusers can use social networking accounts in several ways, the most direct involving the creation of social links (fan, friend, follower, etc.) for marketing purposes.

The relationship between this ecosystem and SEO is subtle: the accounts on social networking sites can also be used for SEO purposes. For example, abusers may spam blogs with comments that link to a Web page to obtain more backlinks for the site. Abusers may also submit links to social bookmarking sites, or utilize forum accounts to create posts containing links (most often in the signature field). Many of these SEO jobs require content, either in the form of articles, or actual content to include in blog comments or forum posts. Lastly, abusers can also directly purchase backlinks on sites.

3.5 Summary

We have described how low-cost freelance labor enables Web service abuse. Using historical data spanning over seven years, we survey the market for such abuse-related work on Freelancer.com, a popular online market for piecework labor outsourcing. We found a broad range of such activities, including mass account creation, SEO-related tasks, and social network link building. Moreover, we witnessed a steadily increasing demand for such services matched by a highly competitive world-wide labor force.

Freelance labor markets like Freelancer.com serve as an incubator and catalyst for new kinds of service abuse. Such a general labor pool allows nascent abuse schemes to be prototyped and evaluated quickly, and, if ultimately profitable, leads naturally to the efficient commoditization of the requisite services. Mature services, such as CAPTCHA solving, eventually evolve into standalone services capable of meeting growing market demand. Modern anti-abuse defenses must, in the end, contend with sophisticated attackers having a versatile and inexpensive labor force at their disposal.

Chapter 3, in part, is a reprint of the material as it appears in Proceedings of USENIX Security 2011. Motoyama, Marti; McCoy, Damon; Levchenko, Kirill; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Understanding CAPTCHA-Solving Services in an Economic Context

In the previous chapter, we detailed the various abuse jobs supported by freelance labor. Now, we look specifically at human-based CAPTCHA-solving, in which sufficient demand for the task on freelancing sites spawned an entire retail industry. In particular, we focus on evaluating the effectiveness of CAPTCHAs as a security mechanism when abusers have access to a workforce willing to solve a single CAPTCHA for far less than a cent. CAPTCHAs were developed as a means to limit the ability of attackers to scale their activities using automated means. In its most common implementation, a CAPTCHA consists of a visual challenge in the form of alphanumeric characters that are distorted in such a way that available computer vision algorithms have difficulty segmenting and recognizing the text. At the same time, humans, with some effort, have the ability to decipher the text and thus respond to the challenge correctly. Today, CAPTCHAs of various kinds are ubiquitously deployed for guarding account registration, comment posting, and so on.

This innovation has, in turn, attached value to the problem of solving CAPTCHAs and created an industrial market. Such commercial CAPTCHA solving comes in two varieties: automated solving and human labor. The first approach defines a technical arms race between those developing solving algorithms and those who develop ever more obfuscated CAPTCHA challenges in response. However, unlike similar arms races that revolve around spam or malware, we will argue that the underlying cost structure favors the defender, and consequently, the conscientious defender has largely won the war.

The second approach has been transformative, since the use of human labor to solve CAPTCHAS effectively side-steps their design point. Moreover, the combination of cheap Internet access and the commodity nature of today's CAPTCHAS has globalized the solving market; in fact, wholesale cost has dropped rapidly as providers have recruited workers from the lowest cost labor markets. Today, there are many service providers that can solve large numbers of CAPTCHAS via on-demand services with *retail* prices as low as $1 per thousand.

In either case, we argue that the security of CAPTCHAS can now be considered in an economic light. This property pits the underlying cost of CAPTCHA solving, either in amortized development time for software solvers or piece-meal in the global labor market, against the value of the asset it protects. While the very existence of CAPTCHA-solving services tells us that the value of the associated assets (e.g., an e-mail account) is worth more to some attackers than the cost of solving the CAPTCHA, the overall shape of the market is poorly understood. Absent this understanding, it is difficult to reason about the security value that CAPTCHAS offer us.

This chapter investigates this issue in depth and, where possible, on a empirical basis. We document the commercial evolution of automated solving tools (particularly via the successful Xrumer forum spamming package) and how they have been largely eclipsed by the emergence of the human-based CAPTCHA-solving market. To characterize this latter development, our approach is to engage the retail CAPTCHA-solving market on both the supply side and the demand side, as both a client and as "workers for hire." In addition to these empirical measurements, we also interviewed the owner and operator of a successful CAPTCHA-solving service (MR. E), who has provided us both validation and insight into the less visible aspects of the underlying business processes.[1] In the course of our analysis, we attempt to address key questions such as which CAPTCHAS are most heavily targeted, the rough solving capacity of the market leaders, the relationship of service quality to price, the impact of market transparency and arbitrage,

[1] By agreement, we do not identify MR. E or the particular service he runs. While we cannot validate all of his statements, when we tested his service empirically our results for measures such as response time, accuracy, capacity and labor makeup were consistent with his reports, supporting his veracity.

the demographics of the underlying workforce and the adaptability of service offerings to changes in CAPTCHA content. We believe our findings, or at least our methodology, provide a context for reasoning about the net value provided by CAPTCHAs under existing threats and offer some directions for future development.

4.1 Automated Software Solvers

From the standpoint of an adversary, automated solving offers a number of clear advantages, including both near-zero marginal cost and near-infinite capacity. At a high level, automated CAPTCHA solving combines *segmentation* algorithms, designed to extract individual symbols from a distorted image, with basic optical character recognition (OCR) to identify the text present in CAPTCHAs. However, building such algorithms is complex (by definition, since CAPTCHAs are designed to evade existing vision techniques), and automated CAPTCHA solving often fails to replicate human accuracy. These constraints have in turn influenced the evolution of automated CAPTCHA solving as it transitioned from a mere academic contest to an issue of commercial viability.

4.1.1 Empirical Case Studies

We explore these issues empirically through two representative examples: Xrumer, a mature forum spamming tool with integrated support for solving a range of CAPTCHAs and reCaptchaOCR, a modern specialized solver that targets the popular reCaptcha service.

Xrumer

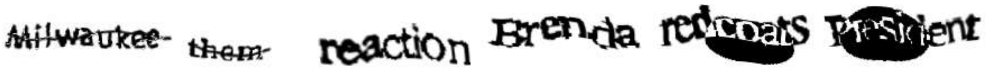
Xrumer [47] is a well-known forum spamming tool, widely described on “blackhat” SEO forums as being one of the most advanced tools for bypassing many different anti-spam mechanisms, including CAPTCHAs. It has been commercially available since 2006 and currently retails for $540, and we purchased a copy from the author at this price for experimentation. While we would have liked to include several other well known spamming tools (SEnuke, AutoPligg, ScrapeBox, etc), the cost of these packages range from $97 to $297, which would render this study prohibitively expensive.

Xrumer's market success in turn led to a surge of spam postings causing most service providers targeted by Xrumer to update their CAPTCHAs. This development kicked off an "arms race" period in Xrumer's evolution as the author updated solvers to overcome these obstacles. Version 5.0 of Xrumer was released in October of 2008 with significantly improved support for CAPTCHA solving. We empirically verified that 5.0 was capable of solving the default CAPTCHAs for then current versions of a number of major message boards, including: Invision Power Board (IPB) version 2.3.0, phpBB version 3.0.2, Simple Machine Forums (SMF) version 1.1.6, and vBulletin version 3.6. These systems responded in kind, and when we installed versions of these packages released shortly after Xrumer 5.0 (in particular, phpBB and vBulletin) we verified that their CAPTCHAs had been modified to defeat Xrumer's contemporaneous solver. Today, we have found that the only major message forum software whose default CAPTCHA Xrumer can solve is Simple Machines Forum (SMF).

With version 5.0.9 (released August 2009), Xrumer added integration for human-based CAPTCHA-solving services: Anti-Captcha (an alias for Antigate) and CaptchaBot. We take this as an indication that the author of Xrumer found the ongoing investment in CAPTCHA-solving software to be insufficient to support customer requirements.[2] That said, Xrumer can be configured to use a hybrid software/human based approach where Xrumer detects instances of CAPTCHAs vulnerable to its automated solvers and uses human-based solvers otherwise. In the current version of Xrumer (5.0.12), the CAPTCHA-related development seems to focus on supporting automatic navigation and CAPTCHA "extraction" (detecting the CAPTCHA and identifying the image file to send to the human-based CAPTCHA-solving service) of more Web sites, as well as evading other anti-spam techniques.

When compared with developers targeting "high-value" CAPTCHAs (e.g., reCaptcha, Microsoft, Yahoo, Google, etc.), Xrumer has mostly targeted "weaker" CAPTCHAs and seems to have a policy of only including highly efficient and accurate software-based solvers. In our tests, all but one included solver required a second or less per CAPTCHA (on a netbook class computer with only a 1.6-GHz Intel Atom CPU)

[2] The developers of Xrumer have recently been advertising enhanced CAPTCHA-solving functionality in their forthcoming "7.0 Elite" version (including support for reCaptcha), but the release date has been steadily postponed and, as of this writing (June 2010), version 5.0.12 is the latest.



(a) Early 2008 (b) December 16th 2009 (c) January 24th 2010

Figure 4.1: Examples of CAPTCHAs downloaded directly from reCaptcha at different time periods.

and had an accuracy of 100%. The one more difficult case was the solver for the phpBB version 3 forum software with the GD CAPTCHA generator and foreground noise. In this case, Xrumer had an accuracy of only 35% and required 6–7 seconds per CAPTCHA to execute.

reCaptchaOCR

At the other end of the spectrum, we obtained a specialized solver focused singularly on the popular reCaptcha service. Wilkins developed the solver as a proof of concept [46]. The existence of this OCR-based reCaptcha solver was reported in a blog posting on December 15, 2009 [19]. Although developed to defeat an earlier version of reCaptcha CAPTCHAs (Figure 4.1a), reCaptchaOCR was also able to defeat the CAPTCHA variant in use at the time of release (Figure 4.1b). Subsequently, reCaptcha changed their CAPTCHA-generation code again to the version as of this writing (Figure 4.1c). The tool has not been updated to solve this new variant.

We tested reCaptchaOCR on 100 randomly selected CAPTCHAs of the early 2008 variant and 100 randomly selected CAPTCHAs of the late 2009 variant. We scored the answers returned using the same algorithm that reCaptcha uses by default. reCaptcha images consist of two words, a control word for which the correct solution is known, and the other a word for which the solution is unknown (the service is used to opportunistically implement human-based OCR functionality for difficult words). By default reCaptcha will mark a solution as correct if it is within an edit distance of one of the control word. However, while we know the ground truth for both words in our tests, we do not know which was the control word. Thus, we credited the solver with half a correct solution for each *word* it solved correctly in the CAPTCHA, reasoning that there was a 50% chance of each word being the control word.

We observed an accuracy of 30% for the 2008-era test set and 18% for the 2009-era test set using the default setting of 613 iterations,[3] far lower than the average human accuracy for the same challenges (75–90% in our experiments).

Finally, we measured the overhead of reCaptchaOCR. On a laptop using a 2.13-GHz Intel Core 2 Duo each solution required an average of 105 seconds. By reducing the number of iterations to 75 we could reduce the solving time to 12 seconds per CAPTCHA, which is in line with the response time for a human solver. At this number of iterations, reCaptchaOCR still achieved similar accuracies: 29% for the 2008-era CAPTCHAs and 17% for the 2009-era CAPTCHAs.

4.1.2 Economics

Both of these examples illustrate the inherent challenges in fielding commercial CAPTCHA-solving software.

While the CAPTCHA problem is often portrayed in academia as a technical competition between CAPTCHA designers and computer vision experts, this perspective does not capture the business realities of the CAPTCHA-solving ecosystem. Arms races in computer security (e.g., anti-virus, anti-spam, etc.) traditionally favor the adversary, largely because the attacker's role is to generate new instances while the defender must recognize them—and the recognition problem is almost always much harder. However, CAPTCHAs reverse these roles since Web sites can be agile in their use of new CAPTCHA types, while attackers own the more challenging recognition problem. Thus, the economics of automated solving are driven by several factors: the cost to develop new solvers, the accuracy of these solvers and the responsiveness of the sites whose CAPTCHAs are attacked.

While it is difficult to precisely quantify the development cost for new solvers, it is clear that highly skilled labor is required and such developers must charge commensurate fees to recoup their time investment. Anecdotally, we contacted one such developer who was offering an automated solving library for the current reCaptcha CAPTCHA. He was charging $6,500 on a non-exclusive basis, and we did not pay to test this solver.

[3] The solver performs multiple iterations and uses the majority solution to improve its accuracy.

At the same time, as we saw with reCaptchaOCR, it can be particularly difficult to produce automated solvers that can deliver human-comparable accuracy (especially for "high-value" CAPTCHAS). While it seems that accuracy should be a minor factor since the cost of attempting a CAPTCHA is all but "free", in reality low success rates limit both the utility of a solver and its useful lifetime. In particular, over short time scales, many forums will blacklist an IP address after 5–7 failed attempts. More importantly, should a solver be put into wide use, changes in the gross CAPTCHA success rate over longer periods (e.g., days) is a strong indicator that a software solver is in use—a signature savvy sites use to revise their CAPTCHAS in turn.[4]

Thus, for a software solver to be profitable, its price must be less than the total value that can be extracted in the useful lifetime before the solver is detected and the CAPTCHA changed. Moreover, for this approach to be attractive, it must also cost less than the alternative: using a human CAPTCHA-solving service. To make this tradeoff concrete, consider the scenario in which a CAPTCHA-solving service provider must choose between commissioning a new software solver (e.g., for a variant of a popular CAPTCHA) or simply outsourcing recognition piecemeal to human laborers. If we suppose that it costs $10,000 to implement a solver for a new CAPTCHA type with a 30% accuracy (like reCaptchaOCR), then it would need to be used over 65 million times (20 million successful) before it was a better strategy than simply hiring labor at $0.5/1,000.[5] However, the evidence from reCaptcha's response to reCaptchaOCR suggests that CAPTCHA providers are well able to respond before such amortization is successful. Indeed, in our interview, MR. E said that he had dabbled with automated solving but that new solvers stopped working too quickly. In his own words, "It is a big waste of time."

For these reasons, software solvers appear to have been relegated to a niche status in the solving ecosystem—focusing on those CAPTCHAS that are static or change slowly in response to pressure. While a technological breakthrough could reverse this state of affairs, for now it appears that human-based solving has come to dominate the commercial market for service.

[4] We are aware that some well-managed sites already have alternative CAPTCHAS ready for swift deployment in just such a situation.

[5] Moreover, human labor is highly flexible and can be used for the wide variety of CAPTCHAS demanded by customers, while a software solver inevitably is specialized to one particular CAPTCHA type.

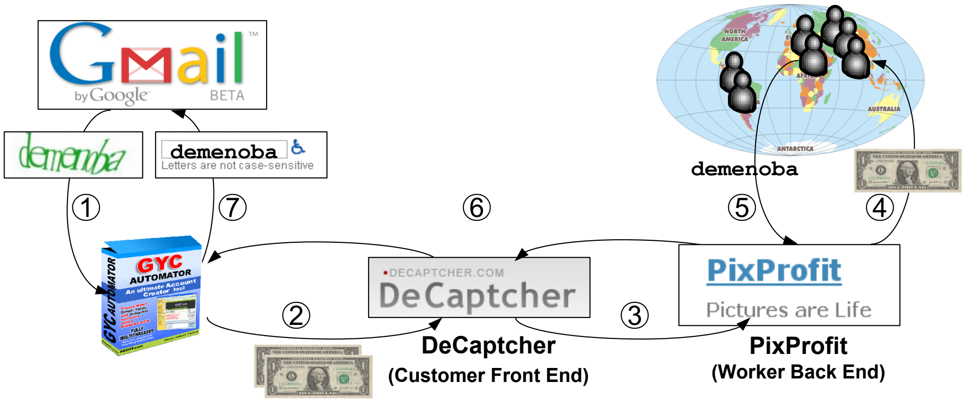


Figure 4.2: CAPTCHA-solving market workflow: ① GYC Automator tries to register a Gmail account and is challenged with a Google CAPTCHA. ② GYC uses the DeCaptcher plug-in to solve the CAPTCHA at $2 per 1,000. ③ DeCaptcher queues the CAPTCHA for a worker on the affiliated PixProfit back end. ④ PixProfit selects a worker and pays at $1/1,000. ⑤ Worker enters a solution to PixProfit, which ⑥ returns it to the plug-in. ⑦ GYC then enters the solution for the CAPTCHA to Gmail to register the account.

4.2 Human Solver Services

Since CAPTCHAs are only intended to obstruct automated solvers, their design point can be entirely sidestepped by outsourcing the task to human labor pools, either opportunistically or on a “for hire” basis. In this section, we review the evolution of this labor market, its basic economics and some of the underlying ethical issues that informed our subsequent measurement study.

4.2.1 Opportunistic Solving

Opportunistic human solving relies on convincing an individual to solve a CAPTCHA as part of some other unrelated task. For example, an adversary controlling access to a popular Web site might use its visitors to opportunistically solving third-party CAPTCHAs by offering these challenges as its own [1, 21]. A modern variant of this approach has recently been employed by the Koobface botnet, which asks infected users to solve a CAPTCHA (under the guise of a Microsoft system management task) [31]. However, we believe that retention of these unwitting solvers will be difficult due to the high profile nature and annoyance of such a strategy, and we do not believe that opportunistic solving plays a major role in the market today.

4.2.2 Paid Solving

Our focus is instead on paid labor, which we believe now represents the core of the CAPTCHA-solving ecosystem, and the business model that has emerged around it. Figure 4.2 illustrates a typical workflow and the business relationships involved.

The premise underlying this approach is that there exists a pool of workers who are willing to interactively solve CAPTCHAs in exchange for less money than the solutions are worth to the client paying for their services.

The earliest description we have found for such a relationship is in a Symantec Blog post from September 2006 that documents an advertisement for a full-time CAPTCHA solver [42]. The author estimates that the resulting bids were equivalent to roughly one cent per CAPTCHA solved, or $10/1,000 (solving prices are commonly expressed in units of 1,000 CAPTCHAs solved). Starting from this date, one can find increasing numbers of such advertisements on "work-for-hire" sites such as getafreelancer.com, freelancejobsearch.com, and mistersoft.com. Shortly thereafter, *retail* CAPTCHA-solving services began to surface to resell such capabilities to a broad range of customers.

Moreover, a fairly standard business model has emerged in which such retailers aggregate the *demand* for CAPTCHA-solving services via a public Web site and open API. The example in Figure 4.2 shows the DeCaptcher service performing this role in steps ② and ⑥. In addition, these retailers aggregate the *supply* of CAPTCHA-solving labor by actively recruiting individuals to participate in both public and private Web-based "job sites" that provide online payments for CAPTCHAs solved. PixProfit, a worker aggregator for the DeCaptcher service, performs this role in steps ③–⑤ in the example.

4.2.3 Economics

While the market for CAPTCHA-solving services has expanded, the wages of workers solving CAPTCHAs have been declining. A cursory examination of historical advertisements on getafreelancer.com shows that, in 2007, CAPTCHA solving routinely commanded wages as high as $10/1,000, but by mid-2008 a typical offer had sunk to $1.5/1,000, $1/1,000 by mid-2009, and today $0.75/1,000 is common, with some workers earning as little as $0.5/1,000.

This downward price pressure reflects the commodity nature of CAPTCHA solving. Since solving is an unskilled activity, it can easily be sourced, via the Internet, from the most advantageous labor market—namely the one with the lowest labor cost. We see anecdotal evidence of precisely this pattern as advertisers switched from pursuing laborers in Eastern Europe to those in Bangladesh, China, India and Vietnam (observations further corroborated by our own experimental results later).

Moreover, competition on the retail side exerts pressure for all such employers to reduce their wages in turn. For example, here is an excerpt from a recent announcement at typethat.biz, the "worker side" of one such CAPTCHA-solving service:

```
009-12-14 13:54 Admin post
Hello, as you could see, server was unstable
last days. We can't get more captchas
because of too high prices in comparison
with other services. To solve this problem,
unfortunately we have to change the rate,
on Tuesday it will be reduced.
```

Shortly thereafter, typethat.biz reduced their offered rate from $1/1,000 to $0.75/1,000 to stay competitive.

These changes reflect similar decreases on the retail side: the customer cost to have 1,000 CAPTCHAs solved is now commonly $2/1,000 and can be as low as $1/1,000. To protect prices, a number of retailers have tried to tie their services to third-party products with varying degrees of success. For example, GYC Automator is a popular "black hat" bulk account creator for Gmail, Yahoo and Craigslist; Figure 4.2 shows GYC's role in the CAPTCHA ecosystem, with the tool scraping a CAPTCHA in step ① and supplying a CAPTCHA solution in step ⑦. GYC has a relationship with the CAPTCHA-solving service Image2Type (not to be confused with ImageToType). Similarly, SENuke is a blog and forum spamming product that has integral support for two "up-market" providers, BypassCaptcha and BeatCaptchas. In both cases, this relationship allows the CAPTCHA-solving services to charge higher rates: roughly $7/1,000 for BypassCaptcha and BeatCaptchas, and over $20/1,000 for Image2Type. It also provides an ongoing revenue source for the software developer. For his service, MR. E confirms that software partners bring in many customers (indeed, they are the majority revenue source) and that he offers a variety of revenue sharing options to attract such partners.

However, such large price differences encourage arbitrage, and in some cases third-party developers have created plug-ins to allow the use of cheaper services on such packages. Indeed, in the case of GYC Automator, an independent developer built a DeCaptcher plug-in which reduced the solving cost by over an order of magnitude. This development has created an ongoing conflict between the seller of GYC Automator and the distributor of the DeCaptcher plug-in. Other software developers have chosen to forgo large margin revenue sharing in favor of service diversity. For example, modern versions of the Xrumer package can use multiple price-leading services (Antigate and CaptchaBot).

Finally, while it is challenging to measure profitability directly, we have one anecdotal data point. In our discussions with MR. E, whose service is in the middle of the price spectrum, he indicated that routinely 50% of his revenue is profit, roughly 10% is for servers and bandwidth, and the remainder is split between solving labor and incentives for partners.

4.2.4 Active Measurement Issues

The remainder of this chapter focuses on active measurement of such services, both by paying for solutions and by participating in the role of a CAPTCHA-solving laborer. The security community has become increasingly aware of the need to consider the legal and ethical context of its actions, particularly for such active involvement, and we briefly consider each in turn for this project.

In the United States (we restrict our brief discussion to U.S. law since that is where we operate), there are several bodies of law that may impinge on CAPTCHA solving. First, even though the services being protected are themselves “free”, it can be argued that CAPTCHAS are an access control mechanism and thus evading them exceeds the authorization granted by the site owner, in potential violation of the Computer Fraud and Abuse Act (and certainly of their terms of service). While this interpretation is debatable, it is a moot point for our study since we never make use of solved CAPTCHAs and thus never access any of the sites in question. A trickier issue is raised by the Digital Millennium Copyright Act's anti-circumvention clause. While there are arguments that CAPTCHA solvers provide a real use outside circumvention of copyright controls (e.g., as aids for the visually impaired) it is not clear—especially in light of increasingly common audio CAPTCHA options—that such a defense is sufficient to protect infringers. Indeed, Ticketmaster recently won a default judgment against RMG Technologies (who sold automated software to bypass the Ticketmaster CAPTCHA) using just such an argument [12]. That said, while one could certainly apply the DMCA against those offering a *service* for CAPTCHAsolving purposes, it seems a stretch to include individual human workers as violators since any such "circumvention" would include innate human visual processes.

Aside from potential legal restrictions, there are also related ethical concerns; one can do harm without such actions being illegal. In considering these questions, we use a consequentialist approach – comparing the consequences of our intervention to an alternate world in which we took no action — and evaluate the outcome for its cost-benefit tradeoff.

On the purchasing side, we impart no direct impact since we do not actually *use* the solutions on their respective sites. We *do* have an indirect impact however since, through purchasing services, we are providing support to both workers and service providers. In weighing this risk, we concluded that the indirect harm of our relatively small investment was outweighed by the benefits that come from better understanding the nature of the threat. On the solving side, the ethical questions are murkier since we understand that solutions to such CAPTCHAs *will* be used to circumvent the sites they are associated with. To sidestep this concern, we chose *not* to solve these CAPTCHAs ourselves. Instead, for each CAPTCHA one of our worker agents was asked to solve, we proxied the image back into the same service via the associated retail interface. Since each CAPTCHA is then solved by the *same* set of solvers who would have solved it *anyway*, we argue that our activities do not impact the gross outcome. This approach does

Table 4.1: Summary of the customer workload to the CAPTCHA-solving services.

Service	$/1K Bulk	Dates (2009–2010)	Requests	Responses
Antigate (AG)	$1.00	Oct 06 – Feb 01 (118 days)	28,210	27,726 (98.28%)
BeatCaptchas (BC)	$6.00	Sep 21 – Feb 01 (133 days)	28,303	25,708 (90.83%)
BypassCaptcha (BY)	$6.50	Sep 23 – Feb 01 (131 days)	28,117	27,729 (98.62%)
CaptchaBot (CB)	$1.00	Oct 06 – Feb 01 (118 days)	28,187	22,677 (80.45%)
CaptchaBypass (CP)	$5.00	Sep 23 – Dec 23 (91 days)	17,739	15,869 (89.46%)
CaptchaGateway (CG)	$6.60	Oct 21 – Nov 03 (13 days)	1,803	1,715 (95.12%)
DeCaptcher (DC)	$2.00	Sep 21 – Feb 01 (133 days)	28,284	24,411 (86.31%)
ImageToText (IT)	$20.00	Oct 06 – Feb 01 (118 days)	14,321	13,246 (92.49%)

cause slightly more money to be injected into the system, but this amount is small.

Finally, we consulted with our human subjects liaison on this work and we were told that the study did not require approval.

4.3 Solver Service Quality

In this section we present our analysis of CAPTCHA-solving services based on actively engaging with a range of services as a client. We evaluate the customer interface, solution accuracy, response time, availability, and capacity of the eight retail CAPTCHA-solving services listed in Table 4.1.

We chose these services through a combination of Web searching and reading Web forums focused on "blackhat" search-engine optimization (SEO). In October of 2009, we selected the eight listed in Table 4.1 because they were well-advertised and reflected a spectrum of price offerings at the time. Over the course of our study, two of the services (CaptchaGateway and CaptchaBypass) ceased operation—we suspect because of competition from lower-priced vendors.

4.3.1 Customer Account Creation

For most of these services, account registration is accomplished via a combination of the Web and e-mail: contact information is provided via a Web site and subsequent sign-up interactions are conducted largely via e-mail. However, most services presented some obstacles to account creation, reflecting varying degrees of due diligence.

For example, both CaptchaBot and Antigate required third-party invitation codes to join their services, which we acquired from the previously mentioned forums. Interestingly, Antigate guards against Western users by requiring site visitors to enter the name of the Russian prime minister in Cyrillic before granting access—an innovation we refer to as a "culturally-restricted CAPTCHA".[6] Some services require a live phone call for account creation, for which we used an anonymous mobile phone to avoid any potential biases arising from using a University phone number. In our experience, however, the burden of proof demanded is quite low and our precautions were likely unnecessary. For example, setting up an ImageToText account required a validation call, but the only question asked was "Did you open an account on ImageToText?" Upon answering in the affirmative (in a voice clearly conflicting with the gender of the account holder's name), our account was promptly enabled. For one service, DeCaptcher, we created multiple accounts to evaluate whether per-customer rate limiting is in use (we found it was not).

Finally, each service typically requires prepayment by customers, in units defined by their price schedule (1,000 CAPTCHAs is the smallest "package" generally offered). To fund each account, we used prepaid VISA gift cards issued by a national bank unaffiliated with our university.

4.3.2 Customer Interface

Most services provide an API package for uploading CAPTCHAs and receiving results, often in multiple programming languages; we generally used the PHP-based APIs. BeatCaptchas and BypassCaptcha did not offer pre-built API packages, so we implemented our own API in Ruby to interface with their Web sites. The client APIs generally employ one of two methods when interacting with their corresponding services. In the first, the API client performs a single HTTP POST that uploads the image to the service, waits for the CAPTCHA to be solved, and receives the answer in the HTTP response; BeatCaptchas, BypassCaptcha, CaptchaBypass and CaptchaBot utilize

[6] In principle, such an approach could be used to artificially restrict labor markets to specific cultures (i.e., CAPTCHA labor protectionism). However it is an open problem if such a *general* form of culturally-restricted CAPTCHA can be devised that has both a large number of examples and a low false reject rate from its target population.

this method.

In the second, the client performs one HTTP POST to upload the image, receives an image ID in the response, and subsequently polls the site for the CAPTCHA solution using the image ID; Antigate, CaptchaGateway, and ImageToText employ this approach. These APIs recommend poll rates between 1–5 seconds; we polled these services once per second. DeCaptcher uses a custom protocol that is not based on HTTP, although they also offer an HTTP interface. One interesting note about ImageToText is that customers must verify that their API code works in a test environment before gaining access to the actual service. The test environment allows users to see the CAPTCHAs they submit and solve them manually.

4.3.3 Service Pricing

Several of the services, notably Antigate and DeCaptcher, offer bidding systems whereby a customer can offer payment over the market rate in exchange for higher priority access to solvers when load is high. In our experience, DeCaptcher charges customers their full bid price, while Antigate typically charges at a lower rate depending on load (as might happen in a second-price auction). To effectively use Antigate, we set our bid price to $2/1,000 solutions since we experienced a large volume of load shedding error codes at the minimum bid price of $1/1,000 (Section 4.3.9 reports on our experiences with service load in more detail). We have not seen price fluctuations on the worker side of these services, and thus we believe that this overage represents pure profit to the service provider.

4.3.4 Test Corpus

We evaluated the eight CAPTCHA-solving services in Table 4.1 as a customer over the course of about five months using a representative sample of CAPTCHAs employed by popular Web sites. To collect this CAPTCHA workload, we assembled a list of 25 popular Web sites with unique CAPTCHAs based on the Alexa rank of the site and our informal assessment of its value as a target (see Figure 4.4 for the complete list). We also used CAPTCHAs from reCaptcha, a popular CAPTCHA provider used by many sites.

We then collected about 7,500 instances of each CAPTCHA directly from each site. For the capacity measurement experiments (Section 4.3.8), we used 12,000 instances of the Yahoo CAPTCHA graciously provided to us by Yahoo.

4.3.5 Verifying Solutions

To assess the accuracy of each service, we needed to determine the correct solution for each CAPTCHA in our corpus. We used the services themselves to do this for us. For each instance, we used the most frequent solution returned by the solver services, after normalizing capitalization and whitespace. If there was more than one most frequent solution, we treated all answers as incorrect (taking this to mean that the CAPTCHA had no correct solution). Table 4.1 shows the overall accuracy of each service as given by our method.

To validate this heuristic, we randomly selected 1,025 CAPTCHAs having at least one service-provided solution and manually examined the images. Of these, we were able to solve 1,009, of which 940 had a unique plurality that agreed with our solution, giving an error rate for the heuristic of just over 8%. Of the 16 CAPTCHAs (1.6%) we could not solve, seven were entirely unreadable, six had ambiguous characters (e.g., ‘0’ vs. ‘o’, ‘6’ vs. ‘b’), and three were rendered ambiguous due to overlapping characters. (We note that Bursztein *et al.* [15] removed CAPTCHAs with no majority from their calculation, which resulted in a higher estimated accuracy than we found in our study.)

4.3.6 Quality of Service

To assess the accuracy, response time, and service availability of the eight CAPTCHA solving services, we continuously submitted CAPTCHAs from our corpus to each service over the course of the study. We submitted a single CAPTCHA every five minutes to all services simultaneously, recording the time when we submitted the CAPTCHA and the time when we received the response. Recall that ImageToText, Antigate and CaptchaGateway require customers to poll the service for the response to a submitted CAPTCHA; we paused one second between each poll call.

Table 4.1 also summarizes the dates, durations, and number of CAPTCHA re-

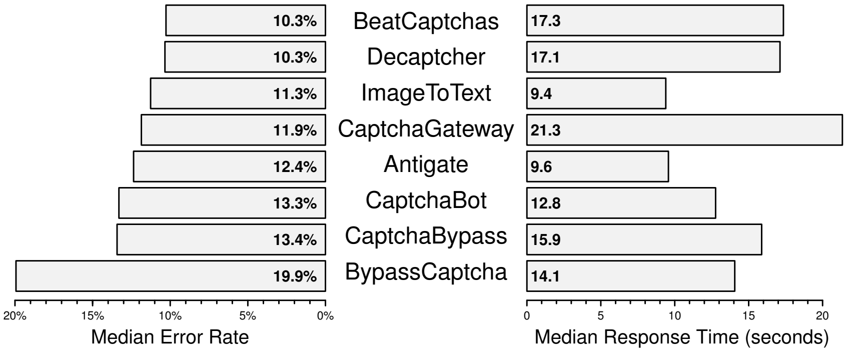


Figure 4.3: Median error rate and response time (in seconds) for all services. Services are ranked top-to-bottom in order of increasing error rate.

quests we submitted to the services; Figure 4.4 presents the error rate and mean response time at a glance for each combination of solver service and CAPTCHA type. We used each service for up to 118 days, submitting up to 28,303 requests per service during that period. We were not able to submit the same number of CAPTCHAs to all services for a number of reasons. For example, services would go offline temporarily, or we would rewrite parts of our client implementation, thus requiring us to temporarily remove the service from the experiment. Furthermore, CaptchaGateway and CaptchaBypass ceased operation during our study.

Accuracy

A CAPTCHA solution is only useful if it is correct. The left bar plot in Figure 4.3 shows the median error rate for each service. Overall the services are reasonably accurate: with the exception of BypassCaptcha, 86–89% of responses [7] were correct. This level of accuracy is in line with results reported by Bursztein *et al.* [15] for human solvers and substantially better than the accuracy of reCaptchaOCR (Section 4.1).

By design, CAPTCHAs vary in difficulty. Do the observed error rates reflect such differences? The top half of Figure 4.4 shows service accuracy (in terms of its error rate) on each CAPTCHA type. The area of each circle is proportional to a service's mean error rate on a particular CAPTCHA type. Services are arranged along the y-axis in order of increasing accuracy, with the most accurate (lowest error rate) at the top and the least

[7] The error rate is over received responses and does not include rejected requests. We consider response rate to be a measure of *availability* rather than accuracy.

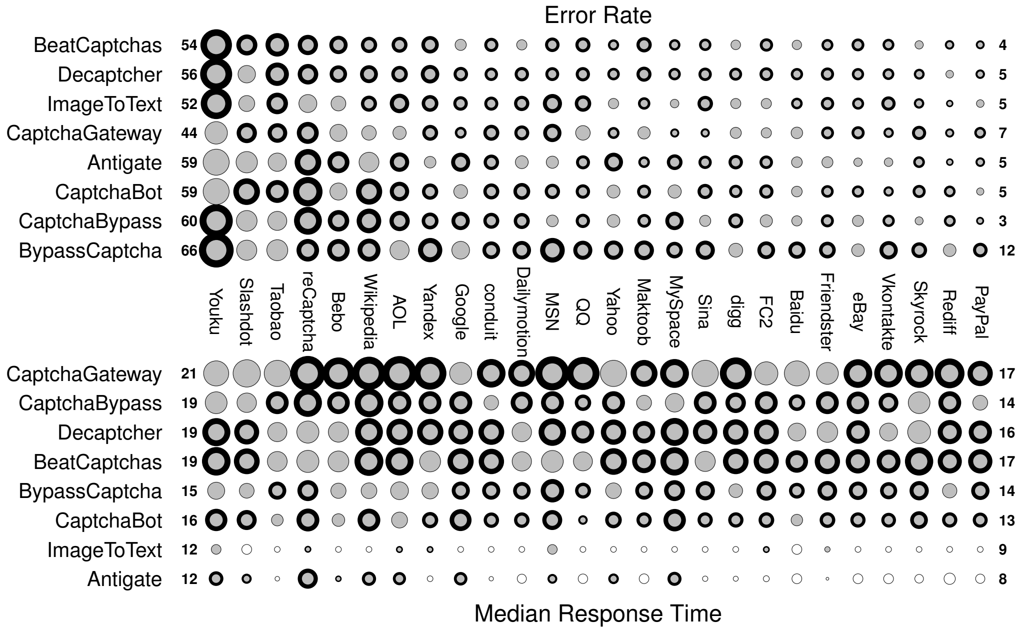


Figure 4.4: Error rate and median response time for each combination of service and CAPTCHA type. The area of each circle upper table is proportional to the error rate (among solved CAPTCHAS). In the lower table, circle area is proportional to the response time *minus ten seconds* (for increased contrast); negative values are denoted by unshaded circles. Numeric values corresponding to the values in the leftmost and rightmost columns are shown on the side. Thus, the error rate of BypassCaptcha on Youku CAPTCHAS is 66%, and for BeatCaptchas on PayPal 4%. The median response time of CaptchaGateway on Youku is 21 seconds, and 8 seconds for Antigate on PayPal.

accurate (highest error rate) at the bottom. CAPTCHA types are arranged in decreasing order of their median error rate. The median error rate of each type is also shown in Figure 4.5.

Accuracy clearly depends on the type of CAPTCHA. The error rate for ImageToText with Youku, for instance, is 5 times its PayPal error rate. Furthermore, the ranking of CAPTCHA accuracies are generally consistent across the services—all services have relatively poor accuracy on Youku and good accuracy on PayPal.

Based on the data, one might conclude that a group of CAPTCHAS on the left headed by Youku, reCaptcha, Slashdot, and Taobao are "harder" than the rest. However an important factor affecting solution accuracy (as well as response time) in our measurements is worker familiarity with a CAPTCHA type. In the case of Youku, for

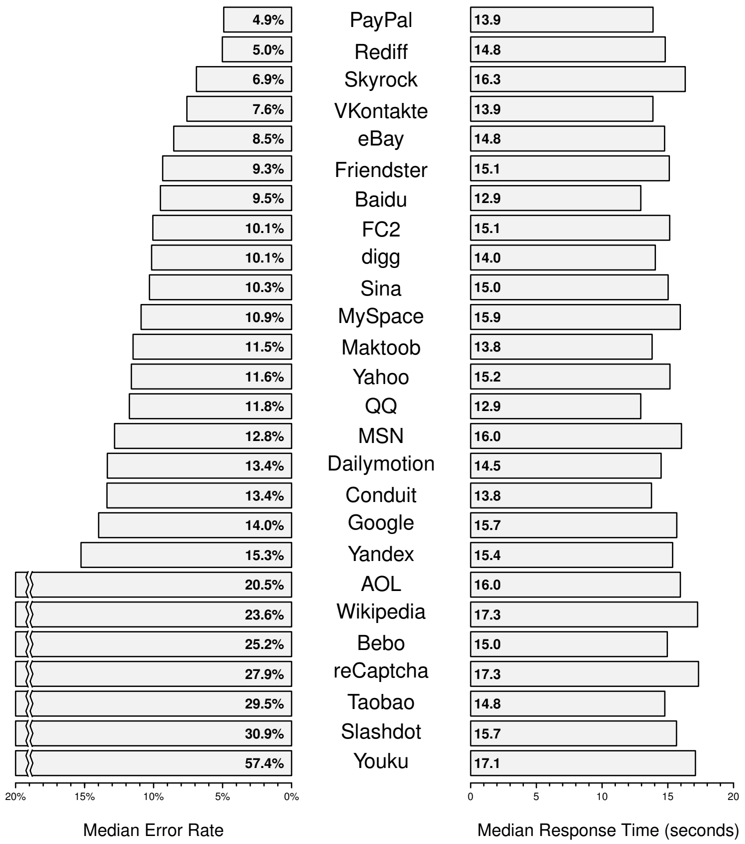


Figure 4.5: Median error rate and response time (in seconds) for all CAPTCHAs. CAPTCHAs are ranked top-to-bottom in increasing error rate.

instance, workers may simply be unfamiliar with these CAPTCHAs. On the other hand, workers are likely familiar with reCaptcha CAPTCHAs (see Section 4.4.6), which may genuinely be "harder" than the rest. As a point of comparison, MR. E reported in our interview that his service experiences a 5–10% error rate. Since his CAPTCHA mix is likely different, and less diverse, than our full set, his claim seems reasonable.

Response Time

In addition to accuracy, customers want services that solve CAPTCHAs quickly. Figure 4.6 shows the cumulative distribution of response times of each service. The curves of CaptchaBot, CaptchaBypass, ImageToText, and Antigate exhibit the quantization effect of polling—either in the client API or on the server—as a stair-step pattern. The shape of the distributions is characteristically log-normal, with a median response of

14 seconds (across all services) and a third-quartile response time of 20 seconds—well within the session timeout of most Web sites. For convenience, Figure 4.3 also shows median response times for each service. In contrast to Bursztein *et al.* [15], who used a different labor pool (Amazon Mechanical Turk), we found no significant difference in response times of correct and incorrect responses.

Services differ considerably in the relative response times they provide to their customers. Antigate (for which we paid a slight premium for priority service as described in Section 4.3.3) and ImageToText provided the fastest service with median response times of 9.6 seconds and 9.4 seconds, respectively, with 90% of CAPTCHAS solved under 25 seconds. CaptchaGateway was the slowest service we measured, with a median of 21.3 seconds and 10% of responses taking over a minute; it was also one of the two services that ceased operation during our study. The remaining services fall in between those extremes. MR. E reported that his service trains workers to achieve response times of 10–12 seconds on average, which is consistent with our measurements of his service.

DeCaptcher and BeatCaptchas have very similar distributions. We have seen evidence (i.e., error messages from BeatCaptchas that are identical to ones documented for the DeCaptcher API) that suggests that BeatCaptchas uses DeCaptcher as a back end. Antigate returns some correct responses unusually quickly (a few seconds), for which we currently do not have an explanation; we have ruled out caching effects.

Services have an advantage if they have better response times than their competition, and the services we measured differ substantially. We suspect that it is a combination of two factors: software and queueing delay in the service infrastructure, and worker efficiency. Antigate, for instance, appears to have an unusually large labor pool (Section 4.3.8), which may enable them to keep queueing delay low. Similarly, ImageToText appears to have an adaptive, high-quality labor pool (Section 4.4.4). We observed additional delays of 5 seconds due to load (Section 4.3.9), but load likely affects all services similarly.

We found that accuracy varied with the type of CAPTCHA. A closely related issue is to what degree response time also varies according to CAPTCHA type. The bottom of Figure 4.4 shows response times by CAPTCHA type. Services are listed along the y-axis

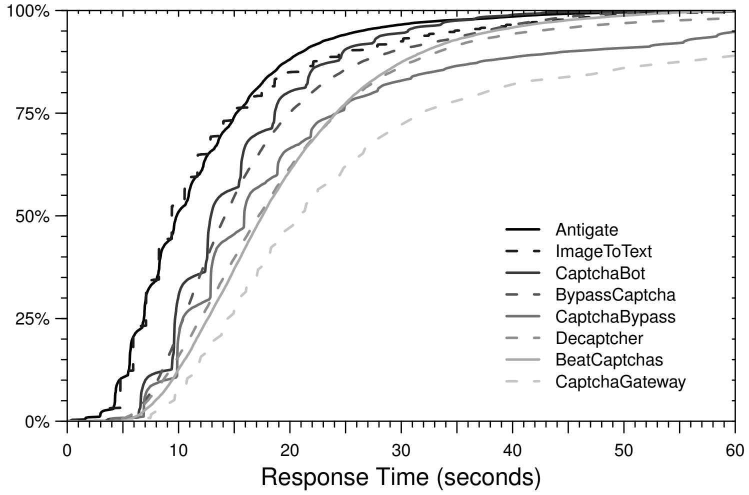


Figure 4.6: Cumulative distribution of response times for each service.

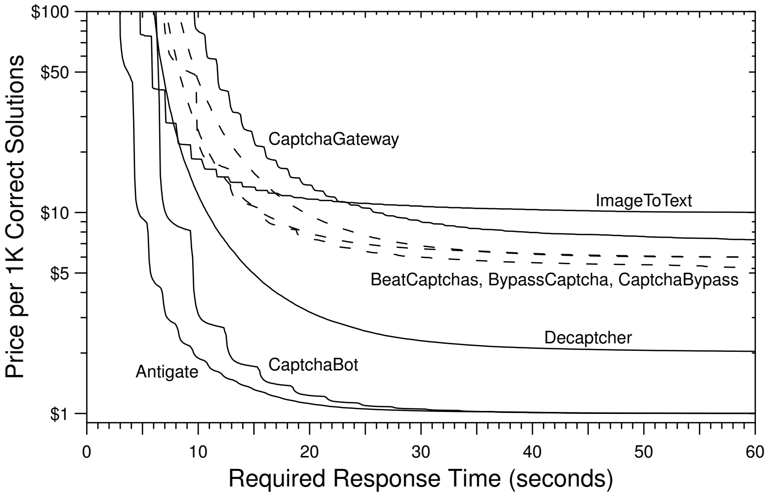


Figure 4.7: Price for 1,000 correctly-solved CAPTCHAs within a given response time threshold.

from slowest (top) to fastest service (bottom). The area of each circle is proportional to the median response time of a service on a particular CAPTCHA type *minus ten seconds* (for greater contrast). Shaded circles are times in excess of ten seconds, unshaded circles are times less than ten seconds. For example, the median response time of Antigate on PayPal CAPTCHAs—8 seconds—is shown as an unshaded circle. Note that CAPTCHA types are still sorted by *accuracy*. The right half of Figure 4.3 aggregates response times by service, showing the median response time of each.

We see some variation in response time among CAPTCHA types. Youku and reCaptcha, for instance, consistently induce longer response times across services, whereas

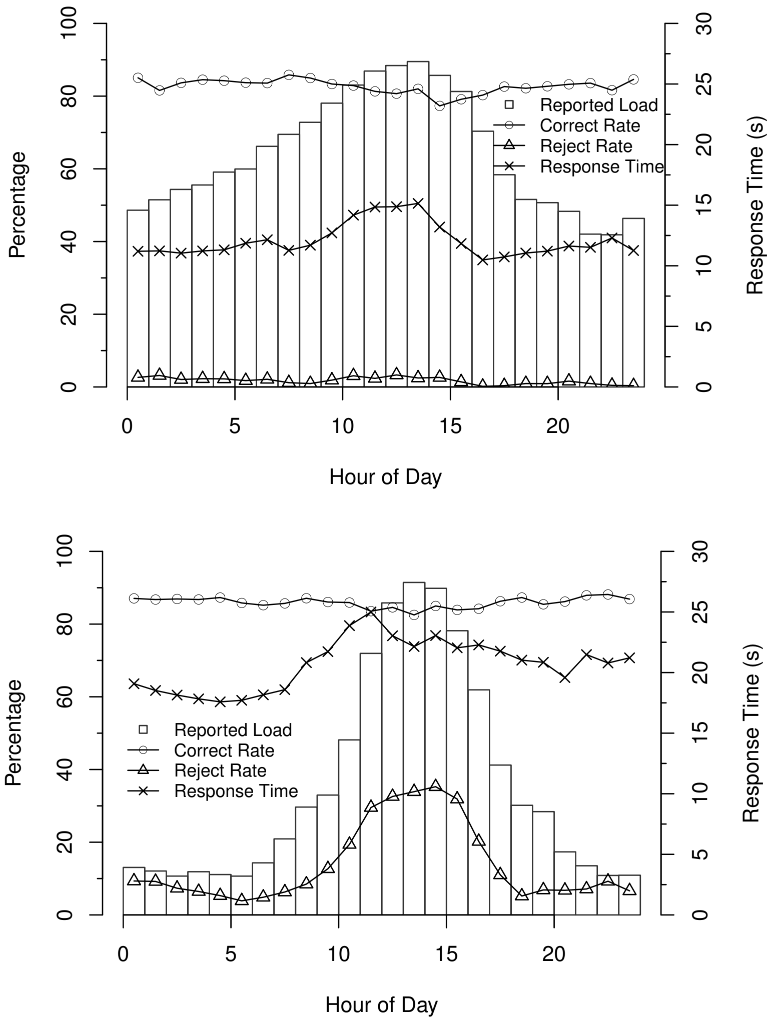


Figure 4.8: Load reported by (a) Antigate and (b) DeCaptcher as a function of time-of-day in one-hour increments. For comparison, we show the percentage of correct responses and rejected requests per hour, as well as the average response time per hour.

Baidu, eBay, and QQ consistently have shorter response times. However, the variation in response times among the services dominates the variation due to CAPTCHA type. The fastest CAPTCHAs that DeCaptcher solves (e.g., Baidu and QQ) are slower on average than the slowest CAPTCHAs that Antigate and ImageToText solve.

4.3.7 Value

CAPTCHA solvers differ in terms of accuracy, response time, and price. The *value* of a particular solver to a customer depends upon the combination of all of these factors:

a customer wants to pay the lowest price for both fast and accurate CAPTCHAs. For example, suppose that a customer wants to create 1,000 accounts on an Internet service, and the Internet service requires that CAPTCHAs be solved within 30 seconds. When using a CAPTCHA solver, the customer will have to pay to have at least 1,000 CAPTCHAs solved, and likely more due to solutions with response times longer than the 30-second threshold (recall that customers do not have to pay for incorrect solutions). From this perspective, the solver with the best value may not be the one with the cheapest price.

Figure 4.7 explores the relationship among accuracy, response time, and price for this scenario. The x-axis is the time threshold T within which a CAPTCHA is useful to a customer. The y-axis is the *adjusted price* per bundle of 1,000 CAPTCHAs that are both solved correctly *and* solved within time T. Each curve corresponds to a solver. Each solver charges a price per CAPTCHA solved (Table 4.1), but not all solved CAPTCHAs will be useful to the customer. The adjusted price therefore includes the overhead of solving CAPTCHAs that take longer than T and are effectively useless. Consider an example where a customer wants to have 1,000 correct CAPTCHAs solved within 30 seconds, a solver charges $2/1,000 CAPTCHAs, and 70$\%$ of the solver's CAPTCHA responses are correct and returned within 30 seconds. In this case, the customer will effectively pay an adjusted price of $\$2 \times (1/0.70) = \$2.86/1,000$ useful CAPTCHAs.

The results in Figure 4.7 show that the solver with the best value depends on the response time threshold. For high thresholds (more than 25 seconds), both Antigate and CaptchaBot provide the best value and ImageToText is the most expensive as suggested by their bulk prices (Table 4.1). However, below this threshold the rankings begin to change. Antigate begins to have better value than CaptchaBot due to having consistently better response times. In addition, ImageToText starts to overtake the other services. Even though its bulk price is $5x$ that of DeCaptcher, for instance, its service is a better value for having CAPTCHAs solved within 8 seconds (albeit at a premium adjusted price).

4.3.8 Capacity

Another point of differentiation is solver capacity, namely how many CAPTCHAs a service can solve in a given unit of time. In addition to low-rate measurements, we also attempted to measure a service's maximum capacity using bursts of CAPTCHA requests. Specifically, we measured the number and rate of solutions returned in response to a given offered load, substantially increasing the load in increments until the service appeared overloaded. We carried out this experiment successfully for five of the services. Of them, Antigate had by far the highest capacity, solving on the order of 27 to 41 CAPTCHAs per second. Even at our highest sustained offered load (1,536 threads submitting CAPTCHAs simultaneously, bid set at $3/1,000), our rejection rate was very low, suggesting that Antigate's actual capacity may in fact be higher. Due to financial considerations, we did not attempt higher offered loads.

For the remaining services, we exceeded their available capacity. We took a non-negligible reject rate to be an indicator of the service running at full capacity. Both DeCaptcher and CaptchaBot were able to sustain a rate of about 14–15 CAPTCHAs per second, with BeatCaptchas and BypassCaptchas sustaining a solve rate of eight and four CAPTCHAs per second, respectively.

Based on these rates, we can calculate a rough estimate of the number of workers at these services. Assuming 10–13 seconds per CAPTCHA (based on our interview with MR. E, and consistent with our measured latencies of his service in the 10–20 second range), Antigate would have had at least 400–500 workers available to service our request. Since we did not exceed their available capacity, the actual number may be larger. Both DeCaptcher and CaptchaBot, at a solve rate of 15 CAPTCHAs per second mentioned above, would have had 130–200 workers available.

4.3.9 Load and Availability

Customers can poll the transient load on the services and offer payment over the market rate in exchange for higher priority access when load is high. During our background CAPTCHA data collection for these services, we also recorded the transient load that they reported. From these measurements, we can examine to what extent services report substantial load, and correlate reported load with other observable metrics (response time, reject rate) to evaluate the validity of the load reports. Because DeCaptcher charges the full customer bid independent of actual load, for instance, it might be motivated to report a false high load in an attempt to encourage higher bids from customers.

Figure 4.8 shows the average reported load as a function of the time of day (in the US Pacific time zone) for both services: for each hour, we compute the average of all load samples taken during that hour for all days of our data set. Antigate reports a higher nominal background load than DeCaptcher, but both services clearly report a pronounced diurnal load effect.

For comparison, we also overlay three other service metrics for each hour across all days: average response time of solved CAPTCHAs, percentage of submitted CAPTCHAs rejected by the service, and the percentage of responses with correct solutions. Response time correlates with reported load, increasing by 5 seconds during high load for each service—suggesting that the high load reports are indeed valid. The percentage of rejected requests for DeCaptcher further validates the load reports. When our bids to DeCaptcher were at the base price of $2/1,000 at times of high load, DeCaptcher aggressively rejected our work requests. To confirm that a higher bid resulted in lower rejection rates, we measured available capacity at 5PM (US Pacific time) at the base price of $2 and then, a few minutes later, at $5, obtaining solve rates of 8 and 18 CAPTCHAs per second, respectively. Although not conclusive, this experience suggests that higher bids may be necessary to achieve a desired level of service at times of high load. Likewise, Antigate exhibits better quality of service when bidding $1 over the base price, though bidding over this amount produced no noticeable improvement (we tested up to $6/1,000).

As further evidence, recall that for Antigate we had to offer premium bids before the service would solve our requests (Section 4.3.2). As a result, even during high loads Antigate did not reject our requests, presumably prioritizing our requests over others with lower bids.

Finally, as expected, accuracy is independent of load: workers are shielded from load behind work queues, solving CAPTCHAs to their ability unaffected by the offered load on the system.

4.4 Workforce

Human CAPTCHA solving services are effectively aggregators. On one hand, they aggregate demand by providing a singular point for purchasing solving services. At the same time, they aggregate the labor supply by providing a singular point through which workers can depend on being offered consistent CAPTCHA solving work for hire. Thus, for each of the publicly-facing retail sites described previously, there is typically also a private "job site" accessed by workers to receive CAPTCHA images and provide textual solutions. Identifying these job sites and which retail service they support is an investigative challenge. For this study, we focused our efforts on two services for which we feel confident about the mapping: Kolotibablo and PixProfit. Kolotibablo is a Russian-run job site that supplies solutions for the retail service Antigate (which, along with CaptchaBot, is the current price leader).

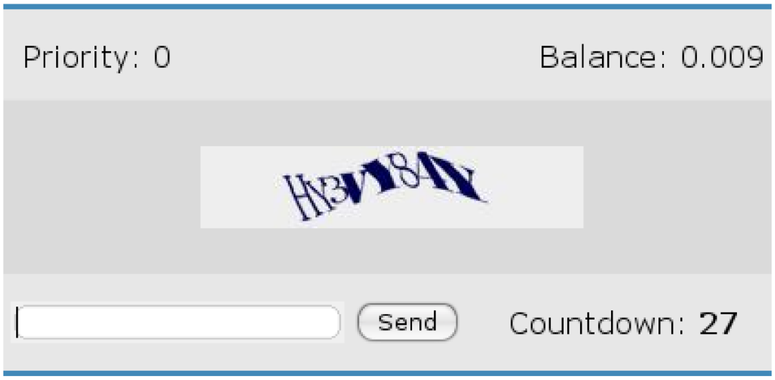
4.4.1 Account Creation

For each job site, account creation is similar to the retail side, but due diligence remains minimal. As a form of quality control, some job sites will evaluate new workers using a corpus of "test" CAPTCHAS (whose solutions are known *a priori*) before they allow them to solve externally provided CAPTCHAS. For this reason, we discard the first 30 CAPTCHAS provided by PixProfit, which we learned by experience correspond to test CAPTCHAS.

4.4.2 Worker Interface

Services provide workers with a Web based interface that, after logging in, displays CAPTCHAS to be solved and provides a text box for entering the solution. Figure 4.9 shows an example of the interface for PixProfit, while Figure 4.10 shows the worker interface for Kolotibablo.

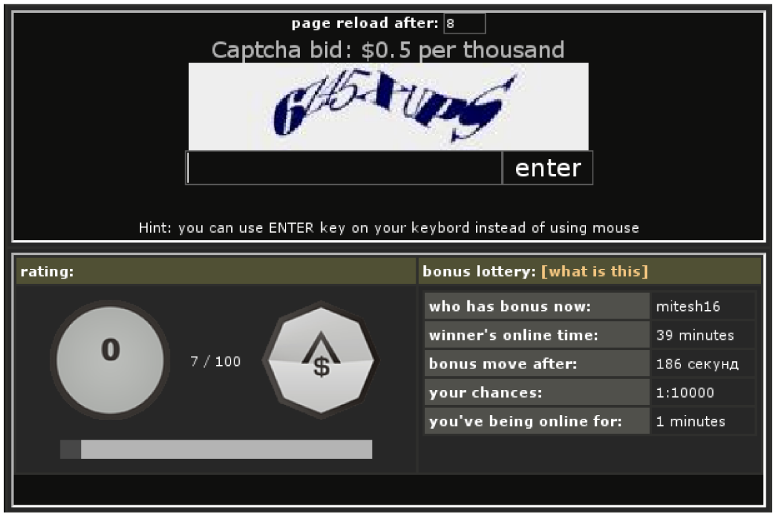
Each site also tracks the number of CAPTCHAS solved, the number that were reported as correct (by customers of the retail service), and the amount of money earned. PixProfit also assigns each worker a "priority" based on solution accuracy. Better accuracy results in more CAPTCHAS to solve during times of lower load. If a solver's



Priority: 0 Balance: 0.009

Countdown: 27

Figure 4.9: Part of a PixProfit worker interface showing a Microsoft CAPTCHA.



page reload after: 8

Captcha bid: $0.5 per thousand

Hint: you can use ENTER key on your keybord instead of using mouse

rating: 0 7 / 100

bonus lottery: [what is this]

who has bonus now:	mitesh16
winner's online time:	39 minutes
bonus move after:	186 секунд
your chances:	1:10000
you've being online for:	1 minutes

Figure 4.10: Portion of a Kolotibablo worker interface displaying a Microsoft CAPTCHA.

accuracy decreases too much, services ban the account. In our experiments, our worker agents always used fresh accounts with the highest level of priority.

4.4.3 Worker Wages

Kolotibablo pays workers at a variable rate depending on how many CAPTCHAs they have solved. This rate varies from $0.50/1,000 up to over $0.75/1,000 CAPTCHAs. PixProfit is the equivalent supplier for DeCaptcher and offers a somewhat higher rate of $1/1,000. Typically, workers must earn a minimum amount of money before payout ($3.00 at PixProfit and $1.00 at Kolotibablo), and services commonly provide payment

Table 4.2: Percentage of responses from the services with correct answers for the language CAPTCHAs.

Language	Example	AG	BC	BY	CB	DC	IT	All
English	one two three	51.1	37.6	4.76	40.6	39.0	62.0	39.2
Chinese (Simp.)	一 二 三	48.4	31.0	0.00	68.9	26.9	35.8	35.2
Chinese (Trad.)	一 二 三	52.9	24.4	0.00	63.8	30.2	33.0	34.1
Spanish	uno dos tres	1.81	13.8	0.00	2.90	7.78	56.8	13.9
Italian	uno due tre	3.65	8.45	0.00	4.65	5.44	57.1	13.2
Tagalog	isá dalawá tatló	0.00	5.79	0.00	0.00	7.84	57.2	11.8
Portuguese	um dois três	3.15	10.1	0.00	1.48	3.98	48.9	11.3
Russian	один два три	24.1	0.00	0.00	11.4	0.55	16.5	8.76
Tamil	ஒன்று இரண்டு மூன்று	2.26	21.1	3.26	0.74	12.1	5.36	7.47
Dutch	een twee drie	4.09	1.36	0.00	0.00	1.22	31.1	6.30
Hindi	एक दो तीन	10.5	5.38	2.47	1.52	6.30	9.49	5.94
German	eins zwei drei	3.62	0.72	0.00	1.46	0.58	29.1	5.91
Malay	satu dua tiga	0.00	1.42	0.00	0.00	0.55	29.4	5.23
Vietnamese	một hai ba	0.46	2.07	0.00	0.00	1.74	18.1	3.72
Korean	일 이 삼	0.00	0.00	0.00	0.00	0.00	20.2	3.37
Greek	ένα δύο τρία	0.45	0.00	0.00	0.00	0.00	15.5	2.65
Arabic	واحد اثنين ثلاثة	0.00	0.00	0.00	0.00	0.00	15.3	2.56
Bengali	এক দুই তিন	0.45	0.00	9.89	0.00	0.00	0.00	1.72
Kannada	ಒಂದು ಎರಡು ಮೂರು	0.91	0.00	0.00	0.00	0.55	6.14	1.26
Klingon		0.00	0.00	0.00	0.00	0.00	1.12	0.19
Farsi	یک دو سه	0.45	0.00	0.00	0.00	0.00	0.00	0.08

via an online e-currency system such as WebMoney.

While we cannot directly measure the gross wages paid by either service, Kolotibablo provides a public list to its workers detailing the monthly earnings for the top 100 solvers each day (presumably as a worker incentive). We monitored these earnings for two months beginning on Dec. 1st, 2009. On this date, the average monthly payout among the top 100 workers was $106.31. However, during December, Kolotibablo revised its bonus payout system, which reduced the payout range by approximately 50% (again reflecting downward price pressure on CAPTCHA-solving labor). As a result, one month later on Jan. 1st, 2010, the average monthly payout to the top 100 earners decreased to $47.32. In general, these earnings are roughly consistent with wages paid to low-income textile workers in Asia [29], suggesting that CAPTCHA-solving is being outsourced to similar labor pools; we investigate this question next.

4.4.4 Geolocating Workers

We crafted CAPTCHAs whose solutions would reveal information about the geographic demographics of the CAPTCHA solvers. We created CAPTCHAs using words corresponding to digits in the native script of various languages (“uno”, “dos”, “tres”, etc., for the CAPTCHA challenge in Spanish), where the correct solution is the sequence of Roman numerals corresponding to those words (“1”, “2”, “3”, etc.) for any CAPTCHA in any language. Ideally, such CAPTCHAs should be easy to grasp and fast to solve by the language’s speakers, yet substantially less likely to be solved by non-speakers or random chance. We expect a measurably high accuracy for services employing workers familiar with those languages.

Table 4.2 lists the languages we used in this experiment along with an example three-digit CAPTCHA in the language corresponding to the solution “123”. For broad global coverage, we selected 21 languages based on a combination of factors including global exposure (English), prevalence of world-wide native speakers (Chinese, Spanish, English, Hindi, Arabic), regions of expected low-cost labor markets with inexpensive Internet access (India, China, Southeast Asia, Latin America), and developed regions unlikely to be sources of affordable CAPTCHA labor (e.g., Western Europe) and lastly one synthetic language as a control (Klingon [35]).

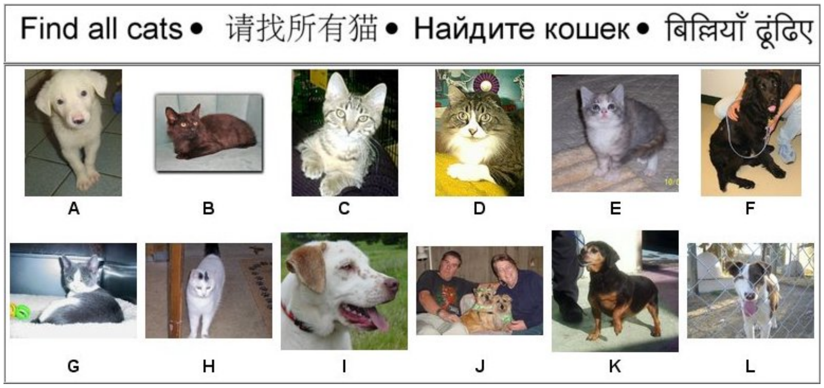
The CAPTCHA we submitted had instructions in the language for how to solve the CAPTCHA (e.g., “Por favor escriba los números abajo” for Spanish), as well as an initial word and Roman numeral as a concrete example (“uno”, “1”). In our experiments, we randomly generated 222 unique CAPTCHAs in each language and submitted them to the six services still operating in January 2010. We rotated through languages such that we submitted a CAPTCHA in this format once every 20–25 minutes. The CAPTCHAs did not repeat digits to reduce the correlated effect of a random guess. As a result, the actual probability for guessing a CAPTCHA is 1/504 ($9 \times 8 \times 7$, reduced by 1 due to the example), although workers unaware of the construction would still be making guesses out of 1,000 possibilities.

Table 4.2 also shows the accuracy of the services when presented with these CAPTCHAs. The accuracy corresponds to a response with all three digits correct (since we created them we have their ground truth). For a convenient ordering, we sort the languages by the average accuracy across all services.

The results paint a revealing picture. First, although Roman alphanumerics in typical CAPTCHAs are globally comprehensible —and therefore easily outsourced— English words for numerals represent a noticeable semantic gap for presumably non-English speakers. Very high accuracies on normal CAPTCHAs drop to 38–62% for the challenge presented in English.

Second, workers at a number of the services exhibit strong affinities to particular languages. Five of the services have accuracies for Chinese (Traditional and Simplified) either substantially higher or nearly as high as English. The services evidently include a sizeable workforce fluent in Chinese, likely mainland China with available low-cost labor. In addition, Antigate has appreciable accuracies for Russian and Hindi, presumably drawing on workforces in Russia and India. Similarly for CaptchaBypass and Russian; BeatCaptcha and Tamil, Portuguese, and Spanish; and DeCaptcher and Tamil. Other non-trivial accuracies in Bengali and Tagalog suggest further recruitment in India and southeast Asia. Services with non-trivial accuracies in Portuguese, Spanish, and Italian could be explained by a workforce familiar with one language who can readily deduce similar words in the other Romance languages. Consistent with these observations, MR. E reported in our interview that they draw from labor markets in China, India, Bangladesh, and Vietnam.

Finally, the results for ImageToText are impressive. Relative to the other services, ImageToText has appreciable accuracy across a remarkable range of languages, including languages where none of the other services had few if any correct solutions (Dutch, Korean, Vietnamese, Greek, Arabic) and even two correct solutions of CAPTCHAs in Klingon. Either ImageToText recruits a truly international workforce, or the workers were able to identify the CAPTCHA construction and learn the correct answers. ImageToText is the most expensive service by a wide margin, but clearly has a dynamic and adaptive labor pool.



Find all cats • 请找所有猫 • Найдите кошек • बिल्लियाँ ढूंढिए

Figure 4.11: Custom Asirra CAPTCHA: workers must type the letters corresponding to pictures of cats.

Time Zone. As another approach for using CAPTCHAs to reveal demographic information about workers—in this case, their time zone—we translated the following instruction into 14 of the languages as CAPTCHA images: "Enter the current time". We sent these CAPTCHAs to each of the six services at the same rate as the other language CAPTCHAs with numbers. We received 15,775 responses, with the most common response being a retype of the instruction in the native language. Of the remaining responses, we received 1,583 (10.0%) with an answer in a recognizable time format. Of those, 77.9% of them came from UTC+8, further reinforcing the estimation of a large labor pool from China; the two other top time zones were the Indian UTC+5.5 with 5.7% and Eastern Europe UTC+2 with 3.0%.

4.4.5 Adaptability

As a final assessment, we wanted to examine how both CAPTCHA services and solvers adapt to changes in state-of-the-art CAPTCHA generation. We focused on the recently proposed Asirra [22] and rotational [25] CAPTCHAs. Asirra is based on identifying pictures of cats and dogs among a set of 12 images, while rotational CAPTCHAinvolves orienting circular pictures of objects until they appear upright. We begin first by describing our approach to testing worker adaptability to the Asirra CAPTCHA.

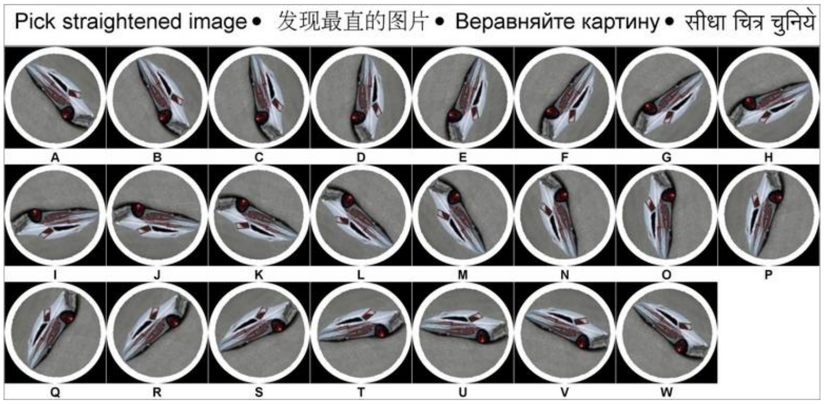

Pick straightened image • 发现最直的图片 • Веравняйте картину • सीधा चित्र चुनिये

Figure 4.12: Custom Rotational CAPTCHA: workers must type the letter corresponding to the image that appears "upright" (letter U)

Table 4.3: The top 5 targeted CAPTCHA types on Kolotibablo and PixProfit, based on CAPTCHAs observed posing as workers.

	Kolotibablo (Antigate)				**PixProfit (DeCaptcher)**		
Service	**# CAPTCHAs**	**% Total**	**% Cum.**	**Service**	**# CAPTCHAs**	**% Total**	**% Cum.**
Microsoft	6,552	25.5%	25.5%	Microsoft	12,135	43.1%	43.1%
Vkontakte.ru	5,908	23.0%	48.5%	reCaptcha	10,788	38.3%	81.4%
Mail.ru	3,607	14.0%	62.5%	Google	1,202	4.3%	85.7%
Captcha.ru	2,476	9.6%	72.2%	Yahoo	1,307	3.7%	89.3%
reCaptcha	921	3.6%	75.8%	AOL	415	1.5%	90.8%
Other (18 sites)	3680	14.3%	90.1%	Other (18)	1086	3.9%	94.7%
Unknown	2551	9.9%	100%	Unknown	1505	5.3%	100%
Total	25,695			Total	28,166		

Asirra

Using the corpus of images provided by the Asirra authors, we hand crafted our own version of the CAPTCHA suitable for use with standard solver image APIs. Figure 4.11 shows an example. We wrote the instructions "Find all cats" in English, Chinese (Simpl.), Russian and Hindi across the top, as the majority of the workers speak one of these languages. We submitted this image once every three minutes to all services over 12 days. ImageToText displayed a remarkable adaptability to this new CAPTCHA type, successfully solving the CAPTCHA on average 39.9% of the time. Figure 4.13 shows the declining error rate for ImageToText; as time progresses, the workers become increasingly adept at solving the CAPTCHA. The next closest service was BeatCaptchas,

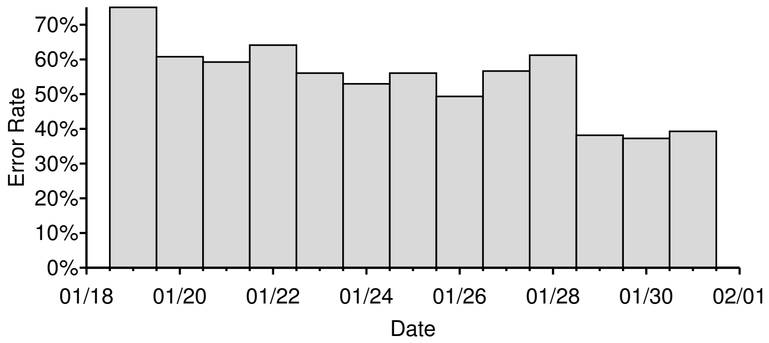


Figure 4.13: ImageToText error rate for the custom Asirra CAPTCHA over time.

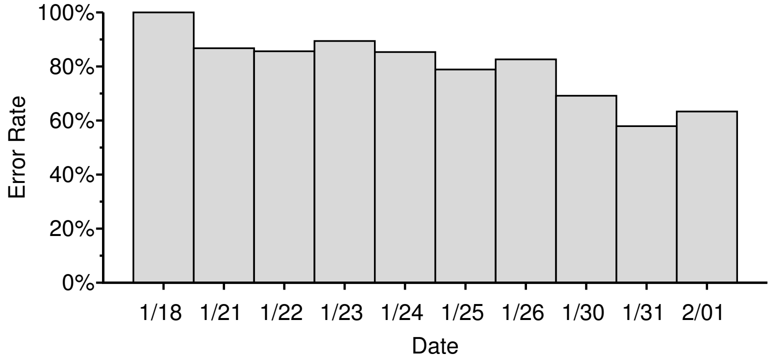


Figure 4.14: ImageToText error rate for the custom Rotational CAPTCHA over time.

which succeeded 20.4% of the time. The remaining services, excluding DeCaptcher, had success rates below 7%.

Coincidentally, as we were evaluating our own version of the Asirra CAPTCHA, on January 17th, 2010 DeCaptcher began offering an API method that supported it directly—albeit at $4 per 1,000 Asirra solves (double its base price). Microsoft had deployed the Asirra CAPTCHA on December 8th, 2009 on Club Bing. Demand for solving this CAPTCHA was apparently sufficiently strong enough that DeCaptcher took only five weeks to incorporate it into their service. We then performed the same experiment described above using the new DeCaptcher API method and received 1,494 responses. DeCaptcher successfully solved 696 (46.5%) requests with a median response time of 39 seconds, about 2.3 times its median of 17 seconds for regular CAPTCHAs. DeCaptcher appears to have factored in the longer solve times for the Asirra CAPTCHAs into the charged price. From what we can tell, though, DeCaptcher does not pay PixProfit workers double the amount for solving them, consequently increasing its profit margin on these new CAPTCHAs.

Rotational CAPTCHA

The premise of the rotational CAPTCHA is that humans are better than computers at recognizing the proper upright orientation of images. Thus, Gossweiler et al. propose displaying an image offset at a random angle to the user, then having he or she rotate the image until it is back in its "natural" upright position. The user is allowed an error window of six to 16 degrees around the correct upright position. Since this CAPTCHAis rather complex, we permit a 16 degree error window, leaving 360/16=23 possible ways to find the natural position. We create our own customized version of the rotational CAPTCHAby starting with a randomly offset image, then rotating the image by 16 degrees increments and concatenating all the variants into a single image. Figure 4.14 illustrates our modified version of this CAPTCHA. Again, we wrote the instructions "Pick straightened image" in the four most common worker languages.

The complexity of this custom CAPTCHA resulted in almost a 100% error rate across all solving sites. Even ImageToText, when initially exposed to the CAPTCHA, failed to solve any of the them correctly; however, Figure 4.14 shows that ImageToText once again improves at solving the CAPTCHA over time. Their error rate declines over the course of the week, eventually reaching below 65%. This further supports the notion that ImageToText workers are adept at solving many CAPTCHA alternatives.

4.4.6 Targeted Sites

Customers of CAPTCHA-solving services target a number of different Web sites. Using our worker accounts on Kolotibablo and PixProfit, the public worker sites of Antigate and DeCaptcher, respectively, we can identify which Web sites are targeted by the customers of these services. Over the course of 82 days we recorded over 25,000 CAPTCHAs from Kolotibablo and 28,000 CAPTCHAs from PixProfit.

To identify the Web sites from which these CAPTCHAs originated, we first grouped the CAPTCHAs by image dimensions. Most groups consisted of a single CAPTCHA type, which we confirmed visually. We then attempted to identify the Web sites from which these CAPTCHAs were taken. In this manner we identified 90% of Kolotibablo

CAPTCHAs and 94% of PixProfit CAPTCHAs.

Table 4.3 shows the top five CAPTCHA types we observed on Kolotibablo and PixProfit, with the remaining identified CAPTCHA types (18 CAPTCHA in both cases) representing 14% and 4% of the CAPTCHA volume on Kolotibablo and PixProfit respectively. Both distributions of CAPTCHA types are highly skewed: on PixProfit, the top two CAPTCHAs types represent 81% of the volume, with the top five accounting for 91%. Kolotibablo is not quite as concentrated, but the top five still account for 76% of its volume.

Clearly the markets for the services are different. Although Microsoft is by far the most common target for both, PixProfit tailors to CAPTCHAs from large global services (Google, Yahoo, AOL, and MySpace) whereas Russian sites otherwise dominate Kolotibablo (VKontakte.ru, Mail.ru, CAPTCHA.ru, Mamba.ru, and Yandex) — a demographic that correlates well with the observed worker fluency in Russian for Antigate (Table 4.2).

4.5 Summary

By design, CAPTCHAs are simple and easy to solve by humans. Their "low-impact" quality makes them attractive to site operators who are wary of any defense that could turn away visitors. However, this same quality has made them easy to outsource to the global unskilled labor market. In this study, we have shed light on the business of solving CAPTCHAs, showing it to be a well-developed, highly-competitive industry with the capacity to solve on the order of a million CAPTCHAs per day. Wholesale and retail prices continue to decline, suggesting that this is a demand-limited market; an assertion further supported by our informal survey of several freelancer forums where workers in search of CAPTCHA-solving work greatly outnumber CAPTCHA-solving service recruitments. One may well ask: *Do* CAPTCHAs *actually work?* The answer depends on what it is that we expect CAPTCHAs to do.

Telling computers and humans apart. The original purpose of CAPTCHAs is to distinguish humans from machines. To this day, no completely general means of solving CAPTCHAs has emerged, nor is the cat-and-mouse game of creating automated solvers viable as a business model. In this regard, then, CAPTCHAs have succeeded.

Preventing automated site access. Today, the retail price for solving one million CAPTCHAs is as low as $1,000. Indeed, for well-motivated adversaries, CAPTCHAs are an acceptable cost of doing business when measured against the value of gaining access to the protected resource. E-mail spammers, for example, solve CAPTCHAs to gain access to Web mail accounts from which to send their advertisements, while blog spammers seek to acquire organic "clicks" and influence result placement on major search engines. Thus, in an absolute sense, CAPTCHAs do not prevent large-scale automated site access.

Limiting automated site access. However, it is short-sighted to evaluate CAPTCHAs as a defense in isolation. Rather, they exert friction on the underlying economic model and should be evaluated in terms of how efficiently they can undermine the attacker's profitability.

Put simply, a CAPTCHA reduces an attacker's expected profit by the cost of solving the CAPTCHA. If the attacker's revenue cannot cover this cost, CAPTCHAs as a defense mechanism have succeeded. Indeed, for many sites (e.g., low PageRank blogs), CAPTCHAs alone may be sufficient to dissuade abuse. For higher-value sites, CAPTCHAs place a utilization constraint on otherwise "free" resources, below which it makes no sense to target them. Taking e-mail spam as an example, let us suppose that each newly registered Web mail account can send some number of spam messages before being shut down. The marginal revenue per message is given by the average revenue per sale divided by the expected number of messages needed to generate a single sale. For pharmaceutical spam, Kanich *et al.* [32] estimate the marginal revenue per message to be roughly $0.00001; at $1 per 1,000 CAPTCHAs, a new Web mail account starts to break even only after about 100 messages sent.[8]

Thus, CAPTCHAs naturally limit site access to those attackers whose business models are efficient enough to be profitable in spite of these costs and act as a drag on profit for all actors. Indeed, MR. E reported that while his service had thousands of

[8] These numbers should be taken with a grain of salt, both because the cited study is but a single data point, and because they studied SMTP-based spam, which generally has lower deliverability than Webmail-based spam. Anecdotally, the retail cost of Webmail-based delivery can be over 100 times more than via SMTP from raw bots.

customers, 75% of traffic was generated by a small subset of them (5–10).

The role of CAPTCHAs today. Continuing our reasoning, the profitability of any particular scam is a function of three factors: the cost of CAPTCHA-solving, the effectiveness of any secondary defenses (e.g., SMS validation, account shutdowns, additional CAPTCHA screens, etc.) and the efficiency of the attacker's business model. As the cost of CAPTCHA solving decreases, a site operator must employ secondary defenses more aggressively to maintain a given level of fraud.

Unfortunately, secondary defenses are invariably more expensive both in infrastructure and customer impact when compared to CAPTCHAs. However, a key observation is that secondary defenses need only be deployed quickly enough to undermine profitability (e.g., within a certain number of messages sent, accounts registered per IP, etc.). Indeed, the optimal point for this transition is precisely the point at which the attacker "breaks even." Before this point it is preferable to use CAPTCHAs to minimize the cost burden to the site owner and the potential impact on legitimate users. While we do not believe that such economic models have been carefully developed by site owners, we see evidence that precisely this kind of tradeoff is being made. For example, a number of popular sites such as Google are now making aggressive use of secondary mechanisms to screen account sign-ups (e.g., SMS challenges), but *only* after a CAPTCHA is passed and some usage threshold is triggered (e.g., multiple sign-ups from the same IP address).[9]

In summary, we have argued that CAPTCHAs, while traditionally viewed as a *technological* impediment to an attacker, should more properly be regarded as an *economic* one, as witnessed by a robust and mature CAPTCHA-solving industry which bypasses the underlying technological issue completely. Viewed in this light, CAPTCHAs are a low-impact mechanism that adds friction to the attacker's business model and thus minimizes the cost and legitimate user impact of heavier-weight secondary defenses. CAPTCHAs continue to serve this function, but as with most such defensive mechanisms, they simply work less efficiently over time.

[9] Anecdotally, this strategy appears effective for now and Gmail accounts on the underground market have gone from a typical asking price of $8/1,000, to being hard to come by at any price. We will not be surprised, however, if this mechanism leads to the monetization of smartphone botnets, or mobots [23], in response.

Chapter 4, in part, is a reprint of the material as it appears in Proceedings of USENIX Security 2010. Motoyama, Marti; Levchenko, Kirill; Kanich, Chris; McCoy, Damon; Savage, Stefan; Voelker, Geoffrey M. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Conclusion

Outsourcing is not a new business strategy. Numerous legitimate businesses have employed outsourced labor to cut down on costs and simplify their operations. These very same reasons make outsourcing an attractive option for abusers. Abusers target Web services for a variety of purposes; for example, to exploit their large user bases or to create unsanctioned advertising channels. Many individuals who commission abuse tasks are running real businesses, and their schemes often reduce to driving consumers to their products while avoiding legitimate advertising fees.

Because the schemes are nontrivial to execute, abusers will often hire workers from various technical and economic backgrounds to actualize their plans. Thus, a labor market has emerged to supply workers for these tasks, namely, online freelancing sites, where abusers can commission their tasks, and workers will bid to complete the jobs. Using cheap, human labor reduces cost, while providing abusers with an alternate attack vector against standard security mechanisms. Web services typically develop defenses with automated tools in mind. Thus, abusers can use the vast size of the freelance labor pool to sidestep many of the countermeasures deployed by Web services. This ability to scale using outsourced labor is one reason why the human-based CAPTCHA solving industry emerged, becoming the dominant method for bypassing CAPTCHAs. Furthermore, the workers are flexible, and willing to adapt their skill sets to meet rising demand for tasks (e.g., switching from spamming users on MySpace to Facebook).

The cost of abusively obtained goods and services varies; understanding the pricing and demand trends for various abuse jobs is important to evaluate the effectiveness of security mechanisms. How can one determine how much “security” a CAPTCHA or phone verification system provides? Observing demand and pricing for goods in response to the implementation of security countermeasures provides another objective feedback mechanism to service providers. Since the products are virtual, the price for abusively obtained goods is a strong indicator of the labor cost involved in procuring the item. Without understanding the impact of their defensive strategies, the service providers run the risk of alienating legitimate users while doing nothing to prevent attacks. Studying freelance marketplaces also allows Web services to glean insight into the attack methodologies employed by abusers.

The thesis of this dissertation is that we can effectively determine the role of outsourced labor in Web abuse by analyzing online labor markets that connect employers with cheap laborers and by engaging with the workers responsible for executing these abusive schemes. We can then assess the efficacy of the defensive measures protecting Internet goods by looking at their prices in retail and freelance marketplaces.

As the Internet becomes yet more pervasive, the labor pool will grow, and the cost of acquiring abusive goods will decrease. Web services must be willing to make an effort to prevent abuse on a massive scale. One approach is to disrupt the manner in which buyers connect with sellers; another is to continue improving defensive mechanisms. We suggest some possible future directions to accomplish these goals in the subsequent section.

5.1 Future Directions

Much of our work focused on methods for measuring and characterizing outsourced abuse labor. We see three possible continuations of this work.

5.1.1 Purchasing Via Retail Sites Versus Outsourcing

The human-based CAPTCHA solving industry was initially fed by workers recruited on freelancing sites. Demand for such services grew, causing entrepreneurial scammers to commoditize human-based CAPTCHA solving. Many abuse tasks identified in Chapter 3 have corresponding retail Web sites; for example, one can purchase accounts directly through a Web site rather than by hiring workers. Researchers have yet to characterize the prices found on retail Web sites, how the site operators process payments, and who fulfills the purchased orders. More research can be done to evaluate the role of retail Web sites in the abuse ecosystem. Interactions with several retail site operators have revealed that they hire human labor, but how retailers of abusive services find workers remains unknown. Are freelancing sites the only venue where abusers can recruit labor? Furthermore, investigating the pricing differences between outsourced and retail purchases provides some indication into how much abusers value convenience. Using retail Web sites eliminates the need to interact with the workers directly, but abusers are forced to pay higher costs.

5.1.2 Identification of Abuse Jobs at Scale

To create the list of job categories for Freelancer.com, we initially randomly sampled jobs. Chapter 3 describes our time-intensive method for building classifiers to identify abusive job types over the entire posting corpus. Future work should pursue a scalable means for recognizing clusters of abuse jobs, since manual efforts are too time consuming and prone to missing classes of jobs. We have conducted preliminary work into using latent Dirichlet allocation (LDA) to identify clusters of abuse jobs [33]. Our work shows that LDA produces meaningful clusters, with the top-weighted keywords in each cluster yielding insight into abuse methodologies and Web service targets. LDA also minimizes the amount of necessary human intervention; users no longer need to label hundreds to thousands of examples to train classifiers, but instead only need to identify which clusters of jobs are abusive. We observe relatively good performance when comparing this completely automated, unsupervised approach to our technique for identifying abuse jobs. Using LDA has many advantages over manual methods, and additional work in this area can lead to an operational system for effectively identifying abuse jobs.

5.1.3 Combating Human-Based CAPTCHA Solving

We have two ideas to combat the use of human-based CAPTCHA solving. In the first, we make the process of identifying and extracting the CAPTCHA more difficult. One way to accomplish this goal is to modify the placement of the CAPTCHA on every Web page refresh, to prevent easily and automatically extracting CAPTCHA images. The authors of Xrumer, for example, focus much of their development efforts on improving their ability to extract CAPTCHAs. This solution can be circumvented with screenshot software, but rendering a Web page can be costly for attackers. Another way to combat this problem is to embed the CAPTCHA in a video or some type of Flash object, making relaying the CAPTCHA more difficult than simply downloading and uploading the image. These approaches make the relay attack more difficult, imposing additional costs on the abusers by forcing them to develop more sophisticated software.

In the second approach, we make the CAPTCHA more difficult to complete by "tying" the CAPTCHA to the challenged user. This breaks the assumption that a given CAPTCHA can be solved by anyone, which is the property exploited by solving services. For example, one idea is to make CAPTCHAs dependent on the user's location (determined using IP address geolocation). The service provider can rotate among various question types, so the attacker cannot apriori know the answers. Suppose a CAPTCHA is presented to someone from a particular US state. The question tied to the CAPTCHA might ask who is the governor of that state, then the user might select among a set of human faces. Another idea is to directly tie a CAPTCHA to the content being protected, using what Kai *et al.* [44] term a Content Based Access Test. For example, if the CAPTCHA is protecting a forum, then the test might ask the user to select images related to the forum topics.

5.2 Final Thoughts

In the future, we expect that outsourcing will continue to play a large role in the abuse ecosystem, especially given the increasing size of the labor pool. The attacks carried out by outsourced labor are often subtle, and may appear indistinguishable from legitimate user behavior. However, there are markers that reveal the outsourced origins of the user; for example, creating profiles on OSNs with an inordinate number of social links, or taking longer to solve CAPTCHAs depending on the hour of the day. Researchers and Web services must cooperate to engage with the outsourced workforce, to better understand how the workers implement their attacks. One cannot hope to combat outsourced attacks without any insight into how they are executed; worker engagement will allow defenders to quickly acquire this knowledge. Frequently, the defenders make assumptions about how attacks are carried out, and these assumptions are often what abusers target.

Web services can also exert more pressure on online labor marketplaces to filter abusive job posts. For example, Craigslist PVA jobs are aggressively removed from Freelancer.com, suggesting that Craigslist has pressured Freelancer into policing their posts. Researchers can assist in developing strategies to identify abuse jobs at scale. Also, investigating abusive jobs on the open market yields great insights for Web service providers. The commissioned jobs often reveal a tremendous amount of information on how the schemes are executed. Thus, the service providers can obtain knowledge into how abusers are gaming their algorithms and defenses. Monitoring the pricing of goods gives objective feedback to providers regarding their defensive countermeasures. Lastly, by understanding the economic incentives behind abusive schemes, researchers can potentially find unexplored weak points in the abuse chain.

Appendix A

Interesting Job Posts and Bids from Freelancer.com

A.1 Interesting Jobs

This appendix includes representative real jobs posted to Freelancer from all the job groups. These examples provide context and help to clarify the various legitimate and dirty job categories.

A.1.1 Legitimate

Private. project has already be awarded to <...>. thanks

Legitimate Miscellaneous. I have a simple document for translation from Dutch to English. Those who are available for immediate start and freelancers only apply.

A.1.2 Accounts

Human CAPTCHA Solving. PixProfit.com is the portal for data-typist. We're looking for individuals or team of data-entering workers. We'll pay from $1 for 1000 correctly typed images.

Phone Verified Accounts. We are looking for a reliable provider of new CL Phone Verified Accounts(PVA).Will be buying up to 1000-2000/month. Willing to pay no more than \$2.00/PVA Or best offer.

A.1.3 SEO

SEO Content Generation.

I need 20 articles written about penis enlargement and 40 articles written about male enhancement. The total is 60 articles with the following requirements. Your writing must be your own original work (no article spinning). Length 500-600 words per article. Written in excellent english with perfect grammar. Keyword density of 2%.

SEO Spinning Article. I am looking for native content providers to provide me articles with spinner syntax. Something like this : {Deciding||Determining} in what {type||kind||sort} of credit card to {apply||go for||lend oneself||put on||employ} for {depends||counts||reckons} on your {past||previous||recent||former} credit {history ||account||report||theme}. Providers without prior spinning knowledge, Please don't bid. I will pay 1.5 USD per spun article to start with only through Paypal.

Link Building/Grey Hat. I am looking to outsource large numbers of blog commenting. Quality blog commenter needed. Can provide 1000 comments per week upwards. This will be for a trial of 100-200 comments per week.

Link Building/White Hat. 100 Gambling Links from related PR 4 or higher pages. All on different sites and servers Requirements: No link farms, link-exchange programs, No black hat links or Tricks.

SEO Miscellaneous. keyword : trader joes
website : will mention via message
SE : google.com
i wan't my website rank 1 in google.com. If interested pls send detail what is your skill to get this website top on google.com

A.1.4 Spamming

Human Oriented Postings. I need per day 2K Classified Ad Posting for my site I willings to pay for it \$100. Per ad \$0.05

A.1.5 OSN Linking

Create Social Networking Links. I am lonely I want to give my facebook account details to someone and have them populate it with 5000 English speaking friends help me please.

A.1.6 Miscellaneous

Abuse Tools. The first tool necessary is Micro Niche Finder. You will need this to do keyword research, and select keywords based on our requirements. The tool will also allow you to see which keywords have .com, .org, or .net domains available. Once the available domains have been determined, we will review your picks, and purchase them after approval. Once purchased, you will need to create articles for each page, and install the necessary wordpress theme and plugins. Once this is complete, you will need to run SE Nuke or Evo II for each site, at least 4 times per month.

Academic Fraud. For this project, you will put together several techniques and concepts learned in CS <deleted > and some new techniques to make an application that searches a large database of people which we will call a Personal Information Manager (PIM), even though it only contains a few fields, and even fewer advanced functions. This project creates a simple program that allows people to enter names or email addresses and check whether they are found in the PIM.

Account Creation Tools. Hey all! I'm in need of US telephone numbers with call forwarding for CL PVA creation. Please quote your rate. Bids lower or equal to $1 will be given higher priority.

Other Malicious. Hello, I have a small sized EXE file of 40KB and I need someone who can build a script who will DOWNLOAD AND EXECUTE the EXE file AUTOMATICALLY. What I mean by automatically? By entering a single URL in the browser.

Here is a PERFECT example: http://www.<deleted>.com. In the example above the EXE file is EXECUTED even when you click on CANCEL in the javascript prompt screen.

A.2 Interesting Bids

This appendix includes representative real bids received from Freelancer workers from some abuse job groups. These bids shed light into the various tools and techniques used by workers to circumvent Web security mechanisms. Also, the bids provide some insight into worker demographics.

A.2.1 Accounts

Account Creation. 1 Account create on 1 ip, Cookies/Cache is cleared after every account automatically. All accounts are created using real human names. We have the ability to provide accounts as per your required format. ——— We created those account with this requirement as below:

1) All Gmail accounts created with unique US IP Addresses 2) All Gmail accounts created separate/unique passwords 3) All accounts created a prefix with names &/or words. Preferably no numbers 4) All accounts to have random First and Last names assigned. 5) All passwords have minimum of 8 characters and preferably alpha-numeric

A.2.2 SEO

SEO Content Generation. Hi!I am <deleted>.I am currently a stay at home mom with 9 month old daughter so I currently have free time throughout the day. I can write quality articles/blogs, academic research papers and LSI/SEO written content of any nature.These articles are put through Copyscape premium dupe test before submission. Also find attached a sample News article I did for a local News paper.I assure you that your articles will be written in the most professional manner possible. I charge $1 per 100 word.I look forward to working with you.Take care

A.2.3 Spamming

Create Social Networking Links. Techniques:(100% white hat) 1. Following people manually: Twitter let us follow 500 people in a day and maximum 2000 follow using one account. So i found a nice technique by which i am able to make 1000 follower. That is

#First follow huge people manually up to 500 using an account similar to your account and after following 500 i will receive a massage, “You have cross the hourly limit . You cant follow now”. Then i will use another account to follow targeted follower up to 500...

A.2.4 OSN Linking

Human Oriented Postings. I am experienced with the CL posting .Now i am working use for daily posting (RDSL With AT@T Line ,CLAD Soft, Ip rental,Proxy,AOL,US hide IP, line with Logmein soft Or,Team Viewer & go to my PC), We have so much experience a team for all adds posting site such as craigslist, backpage, kijiji, gumtree, olx, oddle and all classified site) also have all requirements which need your project done. My company poster allover honest and provide daily work delivery time to time Spreed sheet and provide Any section Sticks ads then active two or more days stick ad !!

A.2.5 Misc

OCR CAPTCHA Solving. 2000usd for already completed recaptcha ocr bot include source code 30%+ accurate

Bibliography

[1] BBC news PC stripper helps spam to spread. http://news.bbc.co.uk/2/hi/technology/7067962.stm.

[2] Cleaning Blogspot Spam: Is Google Responding to Public Pressure? http://www.blogherald.com/2008/04/01/cleaning-blogspot-spam-is-google-responding-to-public-pressure/.

[3] Crowdflower. http://crowdflower.com/.

[4] Data entry assistant. http://www.dataentryassistant.com/.

[5] ESP Game. http://www.espgame.org/gwap/.

[6] Facebook User Statistics. https://www.facebook.com/press/info.php?statistics.

[7] Freelance Market Review Q3 2010. http://whichlance.com/review/freelance-market-review-q3-2010.pdf.

[8] Happy 15th Birthday, Hotmail; Have a Mini-Calendar. http://www.pcmag.com/article2/0,2817,2388108,00.asp.

[9] Inside Craigslist's Increasingly Complicated Battle Against Spammers. http://www.techdirt.com/articles/20080523/0327151211.shtml.

[10] Mechanical Turk: Now with 40.92% spam. http://behind-the-enemy-lines.blogspot.com/2010/12/mechanical-turk-now-with-4092-spam.html.

[11] Spam From Hijacked Webmail Accounts. http://voices.washingtonpost.com/securityfix/2009/04/spam_sent_through_hijacked_web.html.

[12] Ticketmaster, LLC v. RMG Technologies, Inc., et al. 507 F.Supp.2d 1096 (C.D. Ca., October 16, 2007).

[13] Twitter has 105,779,710 Registered Users, Adding 300K A Day. http://techcrunch.com/2010/04/14/twitter-has-105779710-registered-users-adding-300k-a-day.

[14] J. Buckmaster. Phone verification in erotic services. http://blog.craigslist.org/2008/03/phone-verification-in-erotic-services, March 2008.

[15] E. Bursztein, S. Bethard, J. C. Mitchell, D. Jurafsky, and C. Fabry. How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation. In *IEEE S&P '10*, 2010.

[16] M. Chew and D. Tygar. Image recognition CAPTCHAs. In *Information Security, 7th International Conference, ISC 2004*, pages 268–279. Springer, 2004.

[17] Copyscape. http://www.copyscape.com/.

[18] D. Danchev. Inside India's CAPTCHA solving economy. http://blogs.zdnet.com/security/?p=1835, 2008.

[19] D. Danchev. Report: Google's reCAPTCHA flawed. http://blogs.zdnet.com/security/?p=5123, 2009.

[20] R. Datta, J. Li, and J. Z. Wang. Exploiting the Human-Machine Gap in Image Recognition for Designing CAPTCHAs. *IEEE Transactions on Information Forensics and Security*, 4(3):504–518, 2009.

[21] M. Egele, L. Bilge, E. Kirda, and C. Kruegel. CAPTCHA Smuggling: Hijacking Web Browsing Sessions to Create CAPTCHA Farms. In *The 25th Symposium On Applied Computing (SAC)*, pages 1865–1870. ACM, March 2010.

[22] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In *CCS '07*, pages 366–374, New York, NY, USA, 2007. ACM.

[23] C. Fleizach, M. Liljenstam, P. Johansson, G. M. Voelker, and A. Méhes. Can You Infect Me Now? Malware Propagation in Mobile Phone Networks. In *Proceedings of the ACM Workshop on Recurring Malcode (WORM)*, Washington D.C., Nov. 2007.

[24] Freelancer.com. http://www.freelancer.com/info/about.php.

[25] R. Gossweiler, M. Kamvar, and S. Baluja. What's up CAPTCHA?: a CAPTCHA based on image orientation. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 841–850, New York, NY, USA, 2009. ACM.

[26] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The Underground on 140 Characters or Less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[27] A. Hindle, M. W. Godfrey, and R. C. Holt. Reverse Engineering CAPTCHAs. In *Proc. of the 15th Working Conference on Reverse Engineering*, pages 59–68, 2008.

[28] P. G. Ipeirotis. Analyzing the Amazon Mechanical Turk Marketplace. *XRDS: Crossroads*, 17:16–21, Dec. 2010.

[29] L. Jassin-O'Rourke Group. Global Apparel Manufacturing Labor Cost Analysis 2008. `http://www.tammonline.com/files/GlobalApparelLaborCostSummary2008.pdf`, 2008.

[30] T. Joachims. *Making large-scale support vector machine learning practical*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[31] R. F. Jonell Baltazar, Joey Costoya. The Heart of KOOBFACE: C&C and Social Network Propagation. `http://us.trendmicro.com/us/trendwatch/research-and-analysis/white-papers-and-articles/`, October 2009.

[32] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamalytics: an empirical analysis of spam marketing conversion. In *CCS '08*, pages 3–14, New York, NY, USA, 2008. ACM.

[33] D. Kim, M. Motoyama, L. K. Saul, and G. M. Voelker. Topic Modeling of Freelance Job Postings to Monitor Web Service Abuse. In *Proceedings of ACM Workshop on Security and Artificial Intelligence*, October 2011.

[34] J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy enlisted personnel. Naval Technical Training Command Research Branch Report 8–75, February 1975.

[35] The Klingon language institute. `http://www.kli.org`, Accessed February 2010.

[36] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamcraft: An Inside Look at Spam Campaign Orchestration. In *Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Boston, MA, Apr. 2009.

[37] G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *CVPR*, volume 1, pages 134–141, 2003.

[38] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage. Re: CAPTCHAs — Understanding CAPTCHA-Solving from an Economic Context. In *Proceedings of the USENIX Security Symposium*, Washington, D.C., Aug. 2010.

[39] G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion estimation techniques in solving visual CAPTCHAs. pages II: 23–28, 2004.

[40] PWNTcha. Pretend We're Not a Turing computer but a human antagonist. http://caca.zoy.org/wiki/PWNtcha.

[41] G. Sauer, H. Hochheiser, J. Feng, and J. Lazar. Towards a universally usable CAPTCHA. In *SOUPS '08*, 2008.

[42] Symantec. A captcha-solving service. http://www.symantec.com/connect/blogs/captcha-solving-service.

[43] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. In *Advances in Cryptology - EUROCRYPT*, 2003.

[44] K. Wang and C. Kanich. Content Based Access Test (CBAT). http://vision.ucsd.edu/collaborate/index.php/Content_Based_Access_Test_(CBAT).

[45] S.-Y. Wang, H. S. Baird, and J. L. Bentley. CAPTCHA challenge tradeoffs: Familiarity of strings versus degradation of images. In *ICPR '06*, 2006.

[46] J. Wilkins. Strong captcha guidelines v1.2. http://bitland.net/captcha.pdf.

[47] Xrumer. http://www.botmasternet.com/.

[48] YahooSiteExplorerAPI. http://developer.yahoo.com/search/boss/boss_guide/site_explorer.html.

[49] J. Yan and A. S. El Ahmad. A low-cost attack on a Microsoft CAPTCHA. In *CCS '08*, pages 543–554, New York, NY, USA, 2008. ACM.

[50] J. Yan and A. S. El Ahmad. Usability of CAPTCHAs or usability issues in CAPTCHA design. In *SOUPS '08*, pages 44–52, New York, NY, USA, 2008. ACM.

[51] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17, 2008.

[52] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: Defending Against Sybil Attacks via Social Networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '06, pages 267–278, New York, NY, USA, 2006. ACM.

[53] Q. Zhang, T. Ristenpart, S. Savage, and G. M. Voelker. Got Traffic? An Evaluation of Click Traffic Providers. In *Proceedings of the WICOM/AIRWeb Workshop on Web Quality*, 2011.