UNIVERSITY OF CALIFORNIA

CHRIS JAY HOOFNAGLE
Adjunct Full Professor
School of Information
School of Law

Faculty Director
Berkeley Center for Law & Technology

University of California, Berkeley

Berkeley, CA
Tel: 5

https://hoofnagle.berkeley.edu

August 22, 2017

VIA THE WEB

Federal Trade Commission
Office of the Secretary
600 Pennsylvania Avenue NW.
Suite CC–5610 (Annex B)
Washington, DC 20580

Re: Comment of Chris Hoofnagle on Controlling the Assault of Non-Solicited Pornography and Marketing Act (CAN–SPAM Rule, 16 CFR part 316, Project No. R711010)

Dear Mr. Brown,

Thank you for soliciting public comment on the CAN–SPAM Rule. My comments below focus on the need for the CAN–SPAM Rule, the costs that spam imposes on consumers and the economy, the prospect that technical interventions on intermediaries can be effective, that spam senders strategically use transaction costs to deter recipients from opting out, that senders impose privacy penalties on those who opt out, for the FTC to consider third-party lookups for email addresses to be an aggravated violation of CAN–SPAM, to revisit that the idea of a Do-Not-Email Registry, and finally, to keep the computer science literature on spam in focus.

There is a Continuing Need for the CAN–SPAM Rule Because the Injuries Caused by Spam Are Economic and Social and Are on Par with Serious Crimes

In a 2001 speech, FTC Chairman Timothy Muris identified spam messages as injurious under the Commission's "harm-based" approach.[1] Today, the majority of e-mail is spam. Senders of marketing e-mails can leverage the technical and economic properties of the internet to send tens of billions of messages a day. Some miniscule number of recipients purchase items from these messages. Yet, that is enough to make spam profitable. For instance, a 2008 study from the University of California found that 350 million messages to promote online pharmaceutical companies resulted in twenty-eight sales, but it was still a profitable campaign.[2] The costs of the remaining 349 million messages are externalized to others through the need for antispam employees at internet service providers, in money spent on filtering technology, and in individuals' time. The economic cost of spam is high and is conservatively estimated at $20 billion annually.

[1] Timothy J. Muris, *Protecting Consumers' Privacy: 2002 and Beyond*, *Remarks at the Privacy 2001 Conference*, October 4, 2001.
[2] Chris Kanich et al., *Spamalytics: An Empirical Analysis of Spam Marketing Conversion*, CCS08 (2008).

Turning to illegal spammers, such senders are thought to collect only $200 million a year.[3] A paper written by two researchers, one from Google and the other from Microsoft Research, analyzed the "externality ratio" from illegal spam. This term refers to the difference between the private benefit from the activity and the social cost of the activity. The researchers concluded that the externality ratio from illegal spam was greater than that from automobile theft.[4]

The social cost of spam may be even higher. As Brian Krebs explained in SPAM NATION, spam email is seen as a mere nuisance, yet it "has become the primary impetus for the development of malicious software."[5] In other words, the infrastructure (malware installed on millions of computers) created to support spamming can be used for a variety of computer crimes. These range from sending phishing e-mails, to DDoS, to hosting copyrighted content and child pornography.

In the FTC's cost-benefit analyses, it should be mindful of the fact that spammers externalize the majority of the costs of their communication. Compliance burdens on individual senders can look extreme. But in context, these burdens are rather small and properly allocated to senders because of the externalized costs of spam. In weighing the costs of compliance, the FTC should also calculate the costs to recipient ISPs (salaries of anti-spam employees), costs to companies that have to filter by paying for expensive appliances and services (the most basic anti-spam appliance can cost thousands of dollars and it must be maintained), and the time that consumers waste managing unwanted mail (consider the costs of the actual time that consumers spend sorting through junk mailboxes, the costs to the consumer of the lost utility of email as a medium as a result of spam, the cost of legitimate messages being filtered and going unnoticed, and so on).

Technical Interventions: Focusing on Intermediaries

Technical interventions have great promise to reduce spam because the industry depends on sending billions of messages to make sales, and the infrastructure to make these sales appears to be highly concentrated. University of California researchers have found that servers used to effectuate sales are relatively few in number,[6] and that just a handful of companies process most sales for spammers.[7] Spam expert Professor Finn Brunton has written, "A few carefully directed and executed interventions could make an enormous dent in the production of email spam. Filtering and laws did not stop it, by any means, but they have painted it into a developmental corner with severe bottlenecks: an almost totally centralized, consolidated business dependent on colossal volumes of mail to

---

[3] Justin M. Rao & David H. Reiley, *The Economics of Spam*, 26(3) J. ECON. PERSPECTIVES 87 (2012).
[4] *Id.*
[5] BRIAN KREBS, SPAM NATION: THE INSIDE STORY OF ORGANIZED CYBERCRIME – FROM GLOBAL EPIDEMIC TO YOUR FRONT DOOR (2014).
[6] David S. Anderson et al., *Spamscatter: Characterizing Internet Scam Hosting Infrastructure*, 16th USENIX SECURITY SYMPOSIUM (2007). ("... although large numbers of hosts are used to advertise Internet scams using spam campaigns, individual scams themselves are typically hosted on only one machine.")
[7] Kirill Levchenko et al., *Click trajectories: End-to-end analysis of the spam value chain*, PROC. IEEE SYMP. SECURITY & PRIVACY (2011).

survive."[8] Brunton's analysis suggests that law can be combined with technical measures to address the spam problem.

Almost all spam prevention today is technical, accomplished through extensive filtering by e-mail and internet service providers.

My comment here suggests ways in which the Rule could be adjusted to enhance that technical filtering, and to better protect consumers when they exercise their CAN–SPAM rights.

Modify the CAN–SPAM Rule to Reduce Transaction Costs Imposed by Spam Senders

In the last year, I decided to opt out of as much commercial email as possible. I took notes on the opt out process. My observations and argument here is based on the assumption that senders and recipients can be in conflict, with senders having incentives to burden recipients in recipients' attempts to opt out. Senders act opportunistically to reduce opt outs by imposing non-economic costs on recipients. In aggregate, these costs are substantial.

Appendix 1 to this comment demonstrates the various burdens placed on recipients. One the most basic level, senders can make opting out hard by making the unsubscribe language difficult to see. My first example shows unsubscribe language that appears only 5 pixels tall, while the sender's preferred user action, to "pay now" is a button 38 pixels tall.

The second example in Appendix 1 shows a framing technique that slows down the recipient. When the recipient clicks on the unsubscribe option, he is brought to a page where the default action is to subscribe to more email lists. It would be more intuitive for the link to bring the user to a page to unsubscribe.

Example 3 comes from a non-CAN–SPAM regulated industry: politics. But it illustrates an interesting problem. I tried to opt out, but misspelled my email address in the process. The sender's web app recognized my error and reported, "The address you entered does not match the subscribed email address choofnagle@law.berkeley.edu." Thus, the application was aware of what I was trying to do (opt out) and aware of my correct email address, but it nevertheless demanded that I type it out again. What is the point of that except to cause transaction costs?

This same example showed some of the pathologies of a non-CAN–SPAM world (some evidence that there is a continuing need for the Rule): the DCC, to which I never subscribed, refuses to allow people to unsubscribe. It insists that its messages are so important that the only option is to reduce the number of emails received. If the CAN–SPAM Rule were repealed, commercial senders may mimic this practice.

Example 4 shows two examples that use the term "manage" preferences rather an "remove" or "unsubscribe." Presumably, if senders were required to use a standard term for opting out, it would be easier for email filtering to recognize spam and delete it.

[8] FINN BRUNTON: SPAM: A SHADOW HISTORY OF THE INTERNET (2013).

Example 5 shows an example of an opt out that requires that the user turn on Javascript. Such messages demand that the recipient to run code on their computer. This presents unwarranted security risks. Think of it this way: companies with no business relationship whatsoever can send spam under CAN–SPAM. These companies can then require recipients to run code from it or from third parties, without even having the opportunity to view the privacy policy. This code could be malicious. It could cause code injections (XSS), steal cookies, track user keyboard use, and install various web trackers, including device fingerprinting. Some of the most obnoxious web tracking behavior is enabled by Javascript.

Example 6 is similar to 1. It has an unsubscribe option with color contrast ratios that make it very difficult to read. The contrast ratio between the background and text colors in this message is 1.6, far below the AAA-rated standard of 7 under the W3C Web Content Accessibility Guidelines.

Example 7 combines several techniques to obfuscate opt out options.

Example 8 shows the common practice of embedding remotely-loading images in spam. This enables the sender to determine the recipient's IP address (when the images load, server logs report the user's IP address). This choice in design undermines the privacy of email recipients, enables cross-device tracking, and enables the aggregation of profiles based on email address and IP address.

Example 9 requires that the user complete a CAPTCHA in order to opt out. In a real sense, the sender here is requiring the recipient to engage in work—work that in other contexts is compensable[9]—in order to exercise a statutory opt out right.

Finally, I ran mitmproxy while opting out of various emails and found that the opt out process often set cookies with unique ids on my computer and included tracking scripts, sometimes from companies that specialize in building profiles of individuals.

In summary, these examples present several opportunities for the Commission to reduce costs to consumers by adjusting the Rule. Currently, senders undermine users' rights by making opting out more difficult, by placing recipients in a bad security posture, and by tracking individuals such that there is a "privacy cost" to opting out. The FTC could remedy these problems by:

- Clarifying a contrast ratio and text size to ensure that the opt out right is clear and conspicuous. Given the dominance of mobile devices, readability should conform to the WCAG AAA standard.
- Standardizing the term (e.g. "remove," "unsubscribe") used to decline to receive further commercial email messages from the sender.
- Requiring that opt out mechanisms not depend on Javascript, Flash or similar code. These technologies impose a security and privacy cost on recipients that is unjustified. They are also less likely to be universally compatible with devices.

[9] A number of services pay people to complete CAPTCHAs. See e.g. https://2captcha.com

- Requiring that once the recipient clicks on the option to decline to receive further messages, the recipient is taken to a page where the default option is to decline further messages (instead of for signing up for even more messages).
- Prohibiting the sender from using the opt out mechanism to set tracking cookies or other tracking mechanisms unrelated to effectuating the opt out choice.

Modify the Rule to Reduce Privacy Costs Imposed on Recipients from Third Party Data Lookups

Wolfie Christl recently observed:

> *Data companies often remove names from their extensive profiles and use hashing to convert email addresses and phone numbers into alphanumeric codes such as "e907c95ef289". This allows them to claim on their websites and in their privacy policies that they only collect, share, and use "anonymized" or "de-identified" consumer data.*
>
> *However, because most companies use the same deterministic processes to calculate these unique codes, they should be understood as pseudonyms that are, in fact, much more suitable for identifying consumers across the digital world than real names…*[10]

While these processes are sometime described as anonymous, Christl observes:

> *Data management platforms allow businesses in all industries to combine and link their own data on consumers, including real-time information about purchases, website visits, app usage, and email responses, with digital profiles provided by myriads third-party data providers.*

This merger of "anonymized" data through hashing exists in practice. Consider the practices described by Facebook's former product manager for advertising, Antonio García Martínez. In his book CHAOS MONKEYS, Martinez wrote:

> *…Facebook and companies like Acxiom and Datalogix have compared personal data (with none sharing actual data with the other, again via the miracle of hashing), and joined the universal FB user ID to the analogous IDs inside Acxiom, Datalogix, and Epsilon.*
>
> *[…]*
>
> *Facebook, Google, and others have achieved the holy grail of all marketers: a high-fidelity, persistent, and immutable pseudonym for every consumer online. Even better, they've joined that to your real-world persona…*

To make this more explicit, Martinez continues:

> *That personal information is stored in a database, along with the browser cookies that corresponds to it, forming a bridge from real-world you to the browser version of you. It's probably in hashed form, but that's just privacy theater; if everyone agrees on the same hash function, it doesn't matter how it's stored."*
>
> *That join, between a cookie and personal information, is then sold and resold a bazillion times a day to whoever is willing to pay for it…*

[10] Wolfie Christl, Corporate Surveillance in Everyday Life, Cracked Labs, June 2017, http://crackedlabs.org/en/corporate-surveillance/.

In light of these advances, and the fact that the actors that García Martínez identifies are involved in email marketing, the FTC should 1) be wary of definitional changes to "transactional" or "relationship" messages. Consumers do not expect that merely visiting a website enables the linkage of and discovery of information the user did not share, including their contact information and the enablement of spamming; 2) the FTC should investigate whether the use of such techniques is the functional equivalent of address harvesting, and should be considered to be an aggravated violation; and 3) the FTC should more broadly make inquiry into these techniques, with a focus on quality of notice and consent. Back in 2000, the Network Advertising Initiative promised to not merge profiles using online and offline sources without affirmative consent, yet this is exactly what is happening now on many websites.

The Do-Not-E-Mail Registry

The CAN–SPAM Act, at 15 USC § 7708, includes the idea of a Do-Not-E-Mail Registry. Congress ordered the FTC to make a report for establishing such a system and gave it authority to implement it. In a June 2004 report, the FTC recommended against an e-mail registry, however, because such a system would have perverse consequences. The FTC was rightly concerned that the registry would be used by marketers to send even more spam. The Agency recommended that no action be taken to establish a registry until a tamperproof system for e-mail sender authentication was implemented. Given that essentially all mail services now use sender authentication, the key assumption for rejecting the Registry is no longer valid. The FTC should reevaluate the feasibility of a Registry.

The Computer Science Literature on Spam

Finally, the FTC should keep relevant computer science literature in focus to guide its regulatory effort on spam. In particular, Anderson et al. (2007), Kanich et al. (2008), Levchenko et al. (2011), Krebs (2014). These appear in the footnotes to this comment. With consent of an author, I have attached several relevant papers to my submission.

Respectfully submitted,

Chris Jay Hoofnagle

Appendix 1: Hoofnagle Comment

Example 1: Difficult to read opt out language

PAY NOW

Domain Name:	Registration Period:	Price:	Term:
[redacted]		$67.00	1 Year

PAY NOW

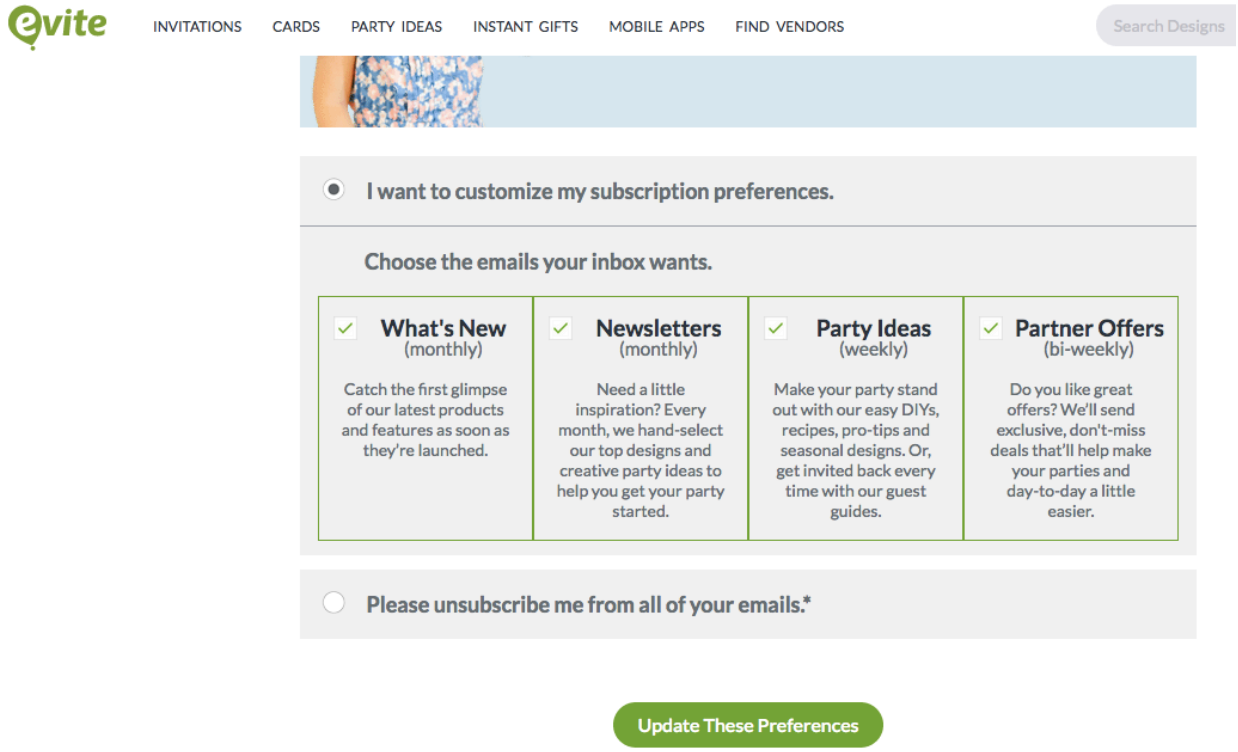
Instructions and Unsubscribe:

You have received this message because you elected to receive special notification offers. If you no longer wish to receive our notifications, please unsubscribe here or mail written request to SEO Optimization Services Inc., Miami, FL 33166. If you have multiple accounts with us, you must opt out for each one individually in order to stop receiving Seo notices. We are a search engine optimization company. We do not register or renew domain names. We sell traffic generator software. This message is Can-Spam compliant. This is not a bill or an invoice. This is a Seo purchase offer. Your are under no obligation to pay the amount stated unless you accept this purchase offer. This message contains promotional material strictly along the guidelines of the Can-Spam act of 2003. We have distinctly mentioned the source mail-id of this email and also disclosed our subject lines. They are in no way misleading. Please do not reply to this email, as we are not able to respond to messages sent to this address.

Figure 1: The "unsubscribe" language here is about 5 pixels tall, while the pay button is 38. The background color (444444) provides insufficient contrast for the foreground text (#adb3ad) unless the font is much bigger.

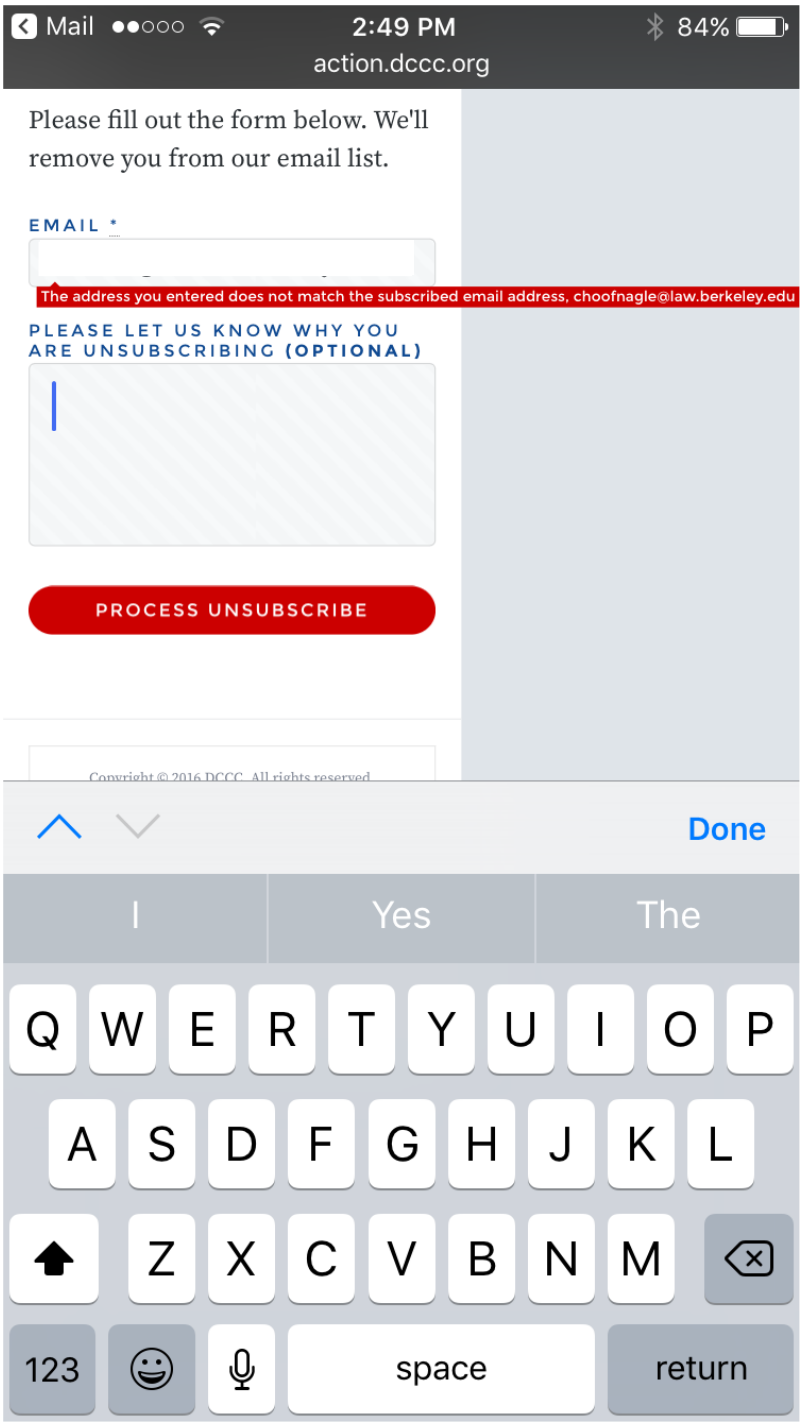
Appendix 1: Hoofnagle Comment

Example 2: Default option from clicking the opt out link is to opt the recipient into more messages.



Appendix 1: Hoofnagle Comment

Example 3: This example is from a non-profit. Notice that when I entered my email address incorrectly, it asked me to go through the exercise of typing it in correctly. Thus, the application knew my email address, and simply wanted me to type it in again to opt out.



Example 3 continued: In fact, the DCCC's (non-profit, not commercial) setup lend weight to the argument that CAN–SPAM is working. Here are two examples of where the DCCC basically says that you cannot opt out of its messages. You can only request fewer of them.

Paid for by the DCCC | 430 South Capitol Street SE, Washington, DC 20003 | (202) 863-1500 | www.dccc.org | Not authorized by any candidate or candidate's committee.

This message was sent to: | If you'd like to sign up for emails in a different account, click here.

We believe that emails are a crucial way for our campaign to stay in touch with supporters. However, if you'd just like to receive fewer emails, you can click here.

What We Stand For

Together, we're showing that millions of Democrats are standing strong behind President Obama's progressive agenda!

Your support couldn't be more critical right now. Ryan's Republicans are hoping they can catch us flat-footed and ram through their extreme Tea Party agenda. But thanks to supporters like you, we're fighting back.

However, if you're getting too many emails, sign up on this page and we'll only send you breaking news alerts at the most critical moments -- only a few times a month. If you still wish to unsubscribe, click here and enter your e-mail address.

Appendix 1: Hoofnagle Comment

Example 4: Some messages do not use the magic words "unsubscribe" or "remove"

Learn about new clouds we protect:
Enabling **SAP SuccessFactors** with Advanced Data Protection from CipherCloud
Ensure Privacy and Compliance in **Adobe Analytics**

Contact Us | Privacy Policy | Manage Email Preferences
Copyright © 2016 CipherCloud

The world's only aquarium, planetarium, rainforest and natural history museum all under one living roof

Membership | Lectures | Events

This email was sent to
Edit Profile | Manage Subscriptions

California Academy of Sciences
55 Music Concourse Dr. San Francisco, CA 94118 USA

Appendix 1: Hoofnagle Comment

Example 5: This opt out from KLM would not work unless the user turned on Javascript.

Click the link below if the page does not load in 5 seconds

Go to your link

Appendix 1: Hoofnagle Comment

Example 6: Some messages have unreadable color contrast

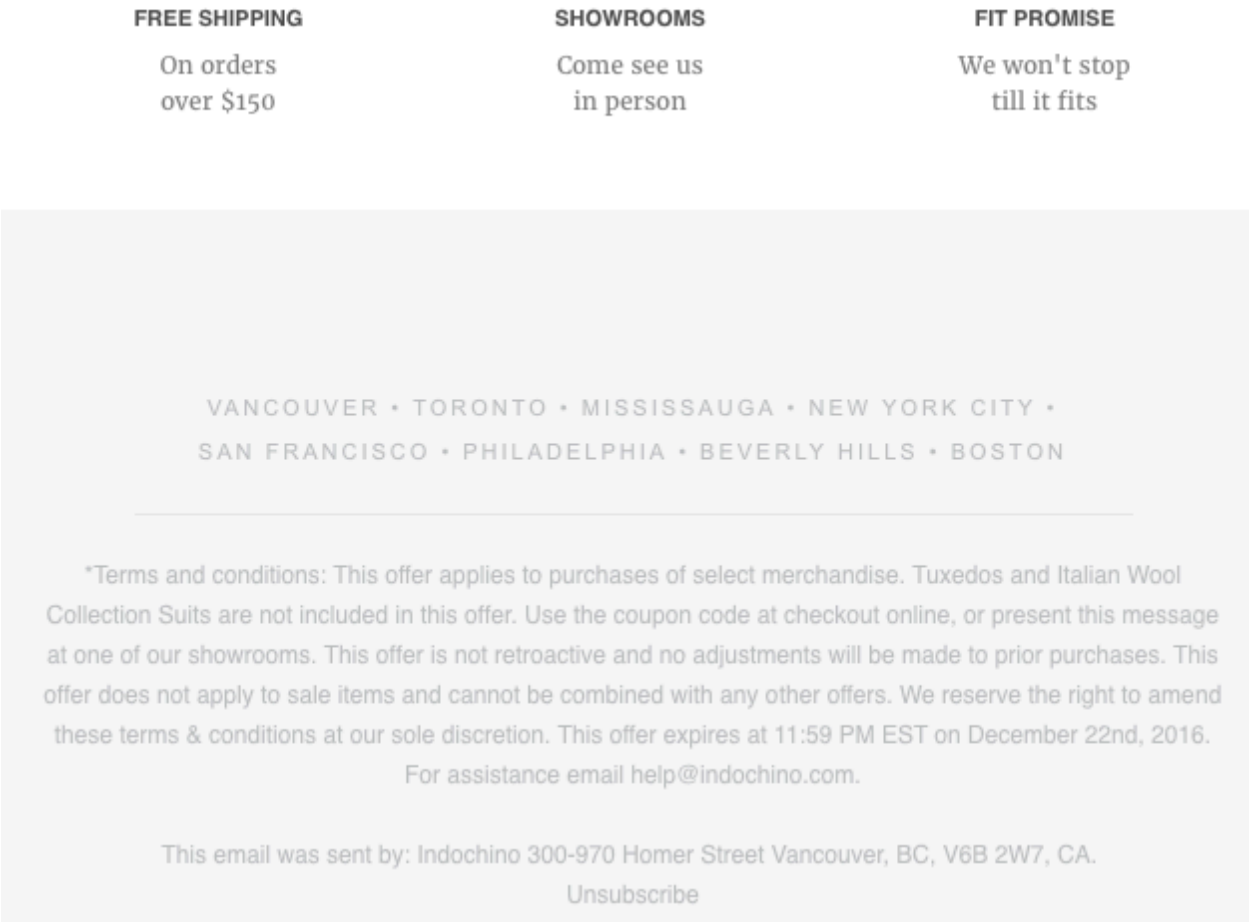
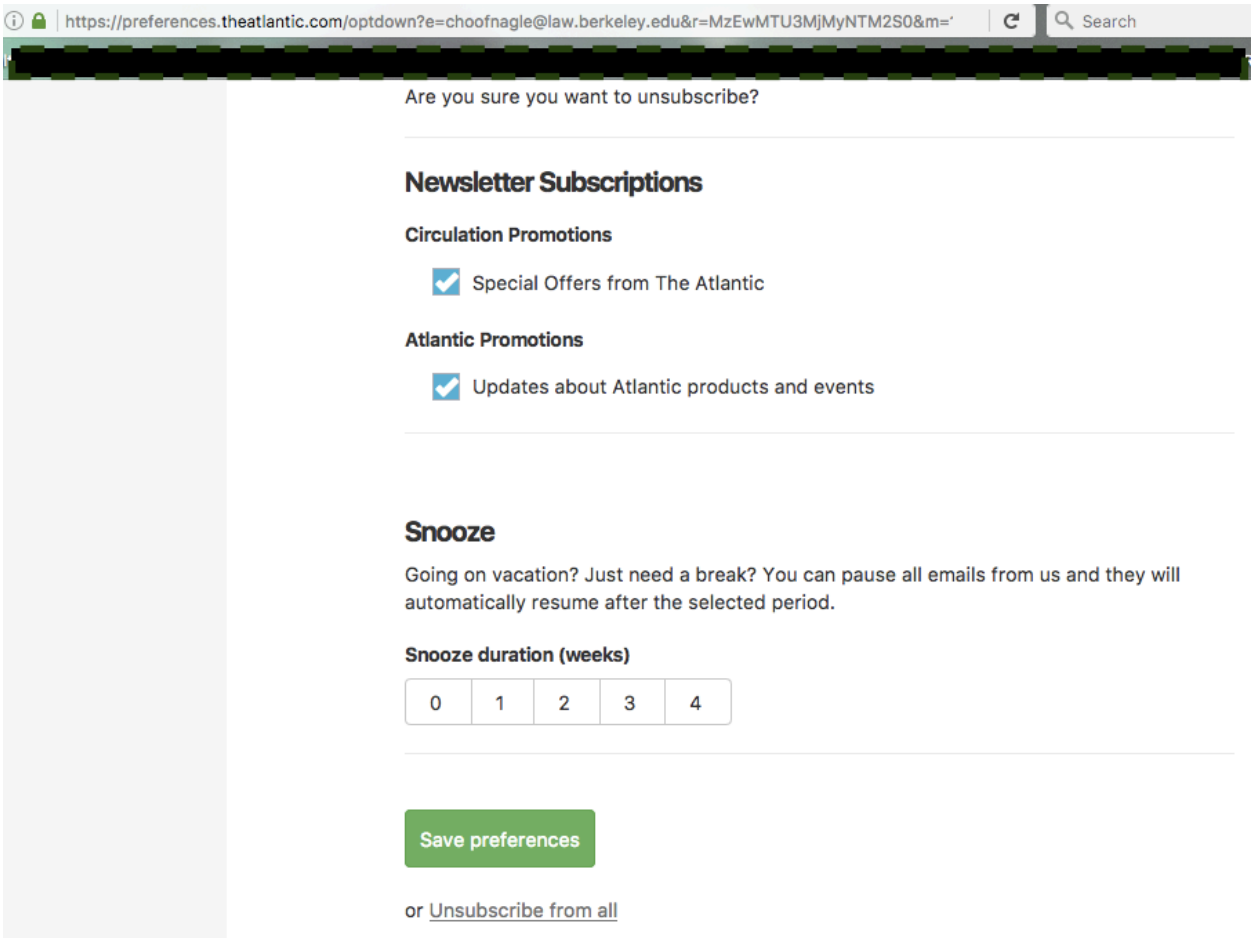


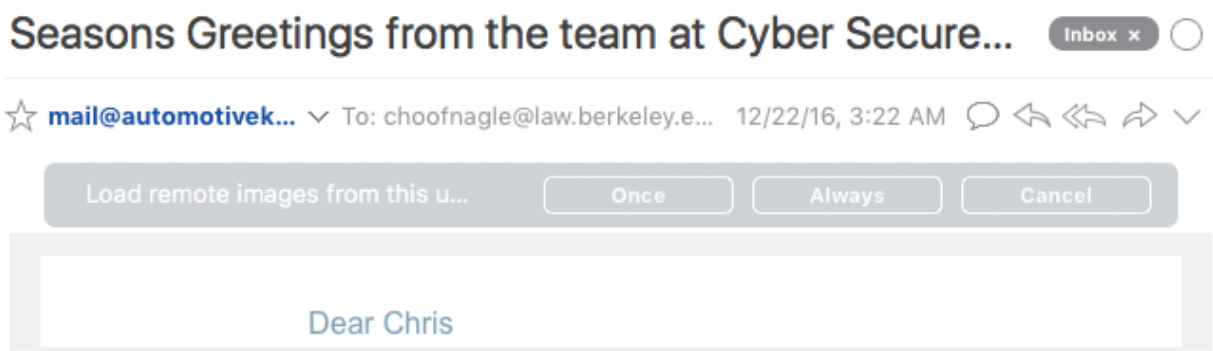
Figure 2: This email has a background color (#f5f5f5) and text color (#c0c2c4) that results in a 1.6 contrast ratio, failing all standards for adequate readability. For reference, white text on a white background would have a contrast ratio of 1.

Appendix 1: Hoofnagle Comment

Example 7: One clicks on the unsubscribe link, and is brought to a page where the default option is to remain subscribed (with a 38 pixel button) and the unsubscribe link is 1) at the very bottom of the page, 2) much smaller than the save button, and 3) in a lighter-color font, therefore reducing readability. The contrast ratio of the black text for remaining subscribed is 10.5, but the unsubscribe link is only 4.7, below the recommended 7 under the W3C Web Content Accessibility Guidelines 2.0.

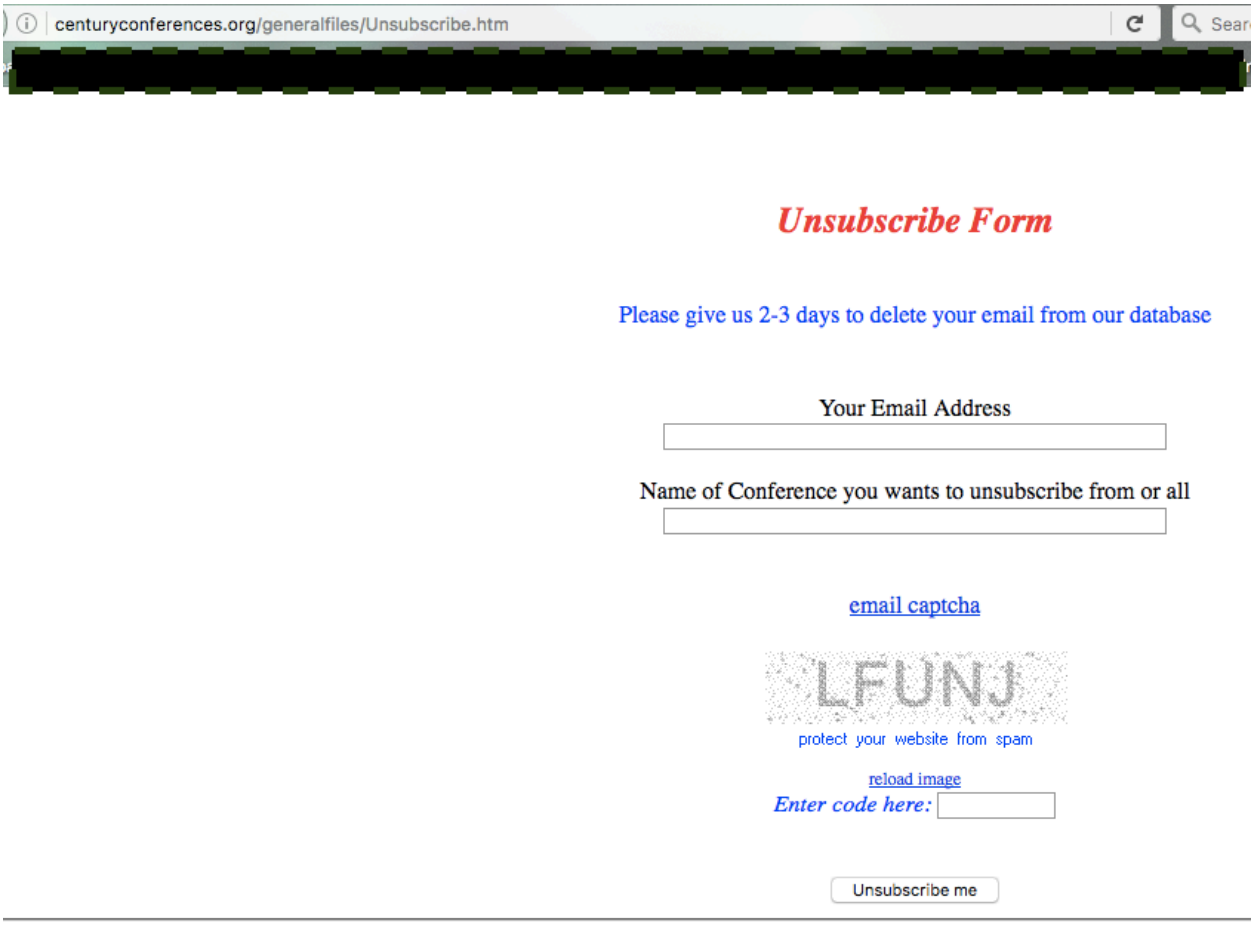


Example 8: Spam often has remotely-loading images. This enables the sender to determine the recipient's IP address (when the images load, server logs report the user's IP address). This choice in design undermines the privacy of email recipients, enables cross-device tracking, and enables the aggregation of profiles based on email address and IP address.



Example 9: This unsubscribe option requires the recipient to complete a CAPTCHA in order to opt out. That is, the sender is requiring the recipient to engage in a form of work in order to exercise the statutory opt out right.

In the cybercrime world, cybercriminals actually pay remote workers to complete CAPTCHAs. If they were smarter, these people would send out tons of spam and require CAPTCHA to opt out. It would be free labor ☺



DOI:10.1145/1562164.1562190

Spamalytics: An Empirical Analysis of Spam Marketing Conversion

By Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage

Abstract

The "conversion rate" of spam—the probability that an unsolicited email will ultimately elicit a "sale"—underlies the entire spam value proposition. However, our understanding of this critical behavior is quite limited, and the literature lacks any quantitative study concerning its true value. In this paper we present a methodology for measuring the conversion rate of spam. Using a parasitic infiltration of an existing botnet's infrastructure, we analyze two spam campaigns: one designed to propagate a malware Trojan, the other marketing online pharmaceuticals. For nearly a half billion spam emails we identify the number that are successfully delivered, the number that pass through popular antispam filters, the number that elicit user visits to the advertised sites, and the number of "sales" and "infections" produced.

1. INTRODUCTION

Spam-based marketing is a curious beast. We all receive the advertisements—"Excellent hardness is easy!"—but few of us have encountered a person who admits to following through on this offer and making a purchase. And yet, the relentlessness by which such spam continually clogs Internet inboxes, despite years of energetic deployment of antispam technology, provides undeniable testament that spammers find their campaigns profitable. Someone is clearly buying. But how many, how often, and how much?

Unraveling such questions is *essential* for understanding the economic support for spam and hence where any structural weaknesses may lie. Unfortunately, spammers do not file quarterly financial reports, and the underground nature of their activities makes third-party data gathering a challenge at best. Absent an empirical foundation, defenders are often left to speculate as to how successful spam campaigns are and to what degree they are profitable. For example, IBM's Joshua Corman was widely quoted as claiming that spam sent by the Storm worm alone was generating "millions and millions of dollars every day."[1] While this claim could in fact be true, we are unaware of any public data or methodology capable of confirming or refuting it.

The key problem is our limited visibility into the three basic parameters of the spam value proposition: the cost to send spam, offset by the "conversion rate" (probability that an email sent will ultimately yield a "sale"), and the marginal profit per sale. The first and last of these are self-contained and can at least be estimated based on the costs charged by third-party spam senders and through the pricing and gross margins offered by various Interne marketing "affiliate programs."[a] However, the conversion rate depends fundamentally on group actions—on what hundreds of millions of Internet users do when confronted with a new piece of spam—and is much harder to obtain. While a range of anecdotal numbers exist, we are unaware of any well-documented measurement of the spam conversion rate.[b]

In part, this problem is methodological. There are no apparent methods for indirectly measuring spam conversion. Thus, the only obvious way to extract this data is to build an e-commerce site, market it via spam, and then record the number of sales. Moreover, to capture the spammer's experience with full fidelity, such a study must also mimic their use of illicit botnets for distributing email and proxying user responses. In effect, the best way to measure spam is to be a spammer.

In this paper, we have effectively conducted this study, though *sidestepping* the obvious legal and ethical problems associated with sending spam.[c] Critically, our study makes use of an *existing* spamming botnet. By infiltrating the botnet parasitically, we convinced it to modify a subset of the spam it *already* sends, thereby directing any interested recipients to Web sites under our control, rather than those belonging to the spammer. In turn, our Web sites presented "defanged" versions of the spammer's own sites, with functionality removed that would compromise the victim's system or receive sensitive personal information such as name, address or credit card information.

Using this methodology, we have documented three spam campaigns comprising over 469 million emails. We identified how much of this spam is successfully delivered,

[a] Our cursory investigations suggest that commissions on pharmaceutical affiliate programs tend to hover around 40%–50%, while the *retail* cost for spam delivery has been estimated at under $80 per million.[14]

[b] The best known among these anecdotal figures comes from the *Wall Street Journal*'s 2003 investigation of Howard Carmack (a.k.a. the "Buffalo Spammer"), revealing that he obtained a 0.00036 conversion rate on 10 million messages marketing an herbal stimulant.[3]

[c] We conducted our study under the ethical criteria of ensuring *neutral actions* so that users should never be worse off due to our activities, while strictly *reducing harm* for those situations in which user property was at risk.

A previous version of this paper appeared in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, Oct. 2008.

how much is filtered by popular antispam solutions, and, most importantly, how many users "click-through" to the site being advertised (*response rate*) and how many of those progress to a "sale" or "infection" (*conversion rate*).

The remainder of this paper is structured as follows. Section 2 describes the economic basis for spam and reviews prior research in this area. Section 4 describes our experimental methodology for botnet infiltration. Section 5 describes our spam filtering and conversion results, Section 6 analyzes the effects of blacklisting on spam delivery, and Section 7 analyzes the possible influences on spam responses. We synthesize our findings in Section 8 and conclude.

2. BACKGROUND

Direct marketing has a rich history, dating back to the nineteenth century distribution of the first mail-order catalogs. What makes direct marketing so appealing is that one can directly measure its return on investment. For example, the Direct Mail Association reports that direct mail sales campaigns produce a response rate of 2.15% on average.[4] Meanwhile, rough estimates of direct mail *cost per mille*—the cost to address, produce and deliver materials to a thousand targets—range between $250 and $1000. Thus, following these estimates it might cost $250,000 to send out a million solicitations, which might then produce 21,500 responses. The cost of developing these prospects (roughly $12 each) can be directly computed and, assuming each prospect completes a sale of an average value, one can balance this revenue directly against the marketing costs to determine the profitability of the campaign. As long as the product of the conversion rate and the marginal profit per sale exceeds the marginal delivery cost, the campaign is profitable.

Given this underlying value proposition, it is not at all surprising that bulk direct email marketing emerged very quickly after email itself. The marginal cost to send an email is tiny and, thus, an email-based campaign can be profitable even when the conversion rate is negligible. Unfortunately, a perverse byproduct of this dynamic is that sending as much spam as possible is likely to maximize profit.[8]

While spam has long been understood to be an economic problem, it is only recently that there has been significant effort in modeling spam economics and understanding the value proposition from the spammer's point of view. Rarely do spammers talk about financial aspects of their activities themselves, though such accounts do exist.[10, 13] Judge et al. speculate that response rates as low as 0.000001 are sufficient to maintain profitability.[12]

However, the work that is most closely related to our own are the several papers concerning "Stock Spam."[5, 7, 9] Stock spam refers to the practice of sending positive "touts" for a low-volume security in order to manipulate its price and thereby profit on an existing position in the stock. What distinguishes stock spam is that it is monetized through price manipulation and not via a sale. Consequently, it is not necessary to measure the conversion rate to understand profitability. Instead, profitability can be inferred by correlating stock spam message volume with changes in the trading volume and price for the associated stocks.

3. THE STORM BOTNET

The measurements in this paper are carried out using the Storm botnet and its spamming agents. Storm is a peer-to-peer botnet that propagates via spam (usually by directing recipients to download an executable from a Web site).

Storm Hierarchy: There are three primary classes of machines that the Storm botnet uses when sending spam. Worker bots make requests for work and, upon receiving orders, send spam as requested. Proxy bots act as conduits between workers and master servers. Finally, the master servers provide commands to the workers and receive their status reports. In our experience there are a very small number of master servers (typically hosted at so-called "bulletproof" hosting centers) and these are likely managed by the botmaster directly.

However, the distinction between worker and proxy is one that is determined automatically. When Storm first infects a host it tests if it can be reached externally. If so, then it is eligible to become a proxy. If not, then it becomes a worker. All of the bots we ran as part of our experiment existed as proxy bots, being used by the botmaster to ferry commands between master servers and the worker bots responsible for the actual transmission of spam messages.

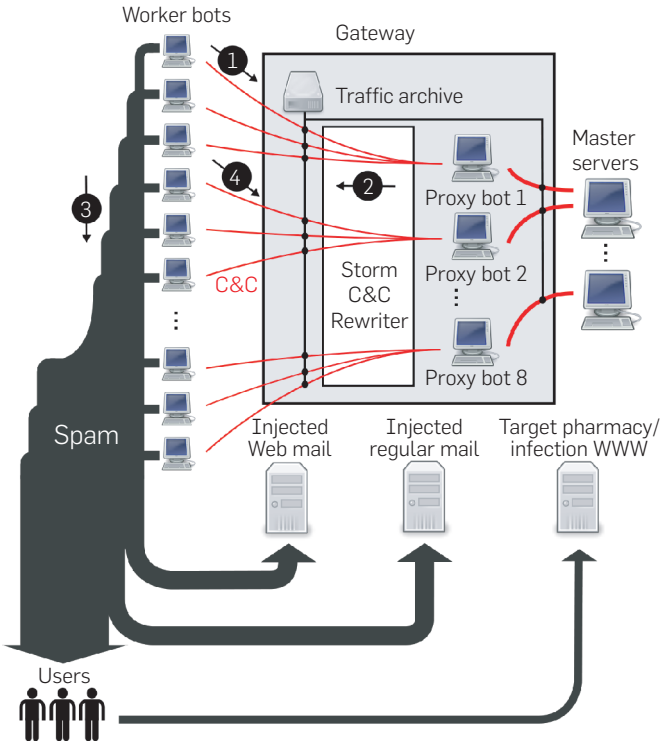
4. METHODOLOGY

Our measurement approach is based on *botnet infiltration*—that is, insinuating ourselves into a botnet's "command and control" (C&C) network, passively observing the spam-related commands and data it distributes and, where appropriate, actively changing individual elements of these messages in transit. Storm's architecture lends itself particularly well to infiltration since the proxy bots, *by design*, interpose on the communications between individual worker bots and the master servers who direct them. Moreover, since Storm compromises hosts indiscriminately (normally using malware distributed via social engineering Web sites) it is straightforward to create a proxy bot on demand by infecting a globally reachable host under our control with the Storm malware.

Figure 1 also illustrates our basic measurement infrastructure. At the core, we instantiate eight unmodified Storm proxy bots within a controlled virtual machine environment. The network traffic for these bots is then routed through a centralized gateway, providing a means for blocking unanticipated behaviors (e.g., participation in DDoS attacks) and an interposition point for parsing C&C messages and "rewriting" them as they pass from proxies to workers. Most critically, by carefully rewriting the spam template and dictionary entries sent by master servers, we arrange for worker bots to replace the intended site links in their spam with URLs of our choosing. From this basic capability we synthesize experiments to measure the click-through and conversion rates for several large spam campaigns.

C&C Protocol Rewriting: Our runtime C&C protocol rewriter consists of two components. A custom router redirects potential C&C traffic to a fixed IP address and port, where a user-space proxy server accepts incoming connections and impersonates the proxy bots. This server in turn forwards connections back into the router, which redirects the traffic

Figure 1. The Storm spam campaign dataflow and our measurement and rewriting infrastructure (Section 4). (1) Workers request spam tasks through proxies, (2) proxies forward spam workload responses from master servers, (3) workers send the spam, and (4) return delivery reports. Our infrastructure infiltrates the C&C channels between workers and proxies.



to the intended proxy bot. Rules for rewriting can be installed independently for templates, dictionaries, and email address target lists. The rewriter logs all C&C traffic between worker and our proxy bots, between the proxy bots and the master servers, and all rewriting actions on the traffic.

Measuring Spam Delivery: To evaluate the effect of spam filtering along the email delivery path to user inboxes, we established a collection of test email accounts and arranged to have Storm worker bots send spam to those accounts. These accounts were created at several different vantage points from which we could evaluate the effectiveness of different email filtering methods. When a worker bot reports success or failure back to the master servers, we remove any success reports for our email addresses to hide our modifications from the botmaster.

We periodically poll each email account (both inbox and "junk/spam" folders) for the messages that it received, and we log them with their timestamps, filtering out any messages not part of this experiment.

Measuring Click-Through and Conversion: To evaluate how often users who receive spam actually visit the sites advertised requires monitoring the advertised sites themselves. Since it is generally impractical to monitor sites not under our control, we have used our *botnet infiltration* method to arrange to have a fraction of Storm's spam advertise sites of our creation instead.

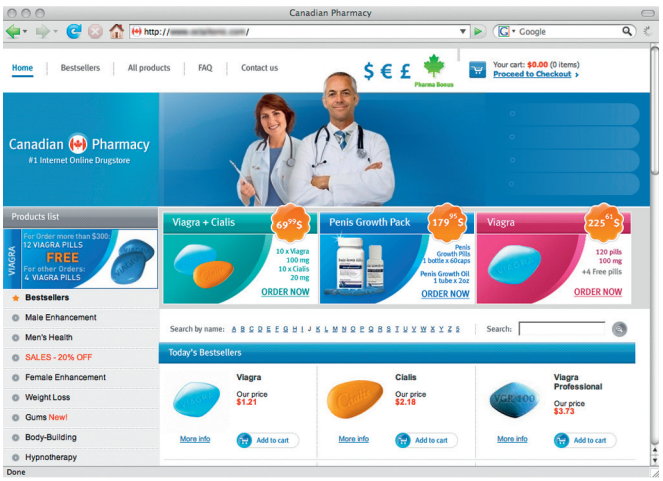
In particular, we have focused on two types of Storm spam campaigns, a self-propagation campaign designed to spread the Storm malware (typically under the guise of advertising an electronic postcard site) and the other advertising a pharmacy site. These are the two most popular Storm spam campaigns and represent over 40% of recent Storm activity.[11] We replaced Storm's links to its own sites with links to sites under our control, screenshots of which are shown in Figure 2.

These sites have been "defanged" in two important ways: the pharmaceutical site does not accept any personal or payment information, and the self-propagation site advertises a completely benign executable which only phones home to record an execution and exits.

4.1. Measurement ethics

We have been careful to design experiments that we believe are both consistent with current U.S. legal doctrine and are fundamentally ethical as well. While it is beyond the scope of this paper to fully describe the complex legal landscape in which active security measurements operate, we believe the ethical basis for our work is far easier to explain: *we strictly reduce harm*. First, our instrumented proxy bots do not create any new harm. That is, absent our involvement, the same set of users would receive the same set of spam emails sent by the same worker bots. Storm is a large self-organizing system and when a proxy fails its worker bots

Figure 2. Screenshots of the Web sites operated to measure user click-through and conversion.



(a) Pharmaceutical site

(b) Postcard-themedself-propagation site

automatically switch to other idle proxies (indeed, when our proxies fail we see workers quickly switch away). Second, our proxies are passive actors and do not engage themselves in any behavior that is intrinsically objectionable; they do not send spam email, they do not compromise hosts, nor do they even contact worker bots asynchronously. Indeed, their only function is to provide a conduit between worker bots making requests and master servers providing responses. Finally, where we do modify C&C messages in transit, these actions themselves strictly reduce harm. Users who click on spam altered by these changes will be directed to one of our innocuous doppelganger Web sites. Unlike the sites *normally* advertised by Storm, our sites do not infect users with malware and do not collect user credit card information. Thus, no user should receive more spam due to our involvement, but some users will receive spam that is less dangerous that it would otherwise be.

Needless to say, we encourage no one to recreate our experiments without the utmost preparation and care. Interacting with thousands of compromised machines that are sending millions of spam messages is a very delicate procedure, and while we encourage other researchers to build upon our work, we ask that these experiments only be attempted by qualified professionals with no less forethought, legal consultation, or safeguards than those outlined here.

5. EXPERIMENTAL RESULTS

We now present the overall results of our rewriting experiment. We first describe the spam workload observed by our C&C rewriting proxy. We then characterize the effects of filtering on the spam workload along the delivery path from worker bots to user inboxes, as well as the number of users who browse the advertised Web sites and act on the content there.

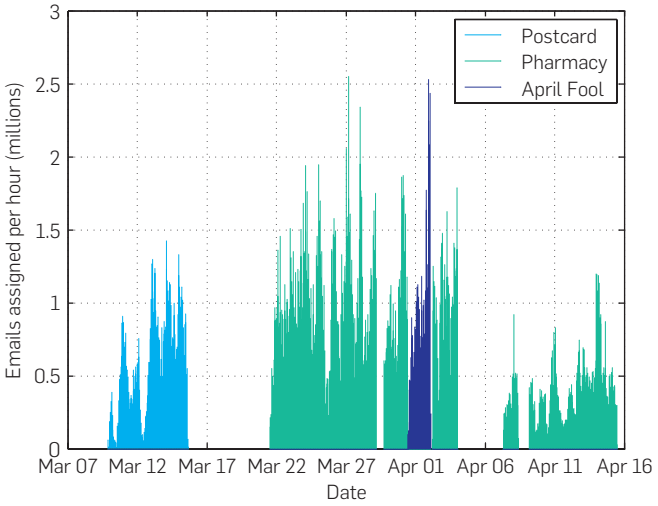
Campaign Datasets: Our study covers three spam campaigns summarized in Table 1. The "Pharmacy" campaign is a 26-day sample (19 active days) of an ongoing Storm campaign advertising an online pharmacy. The "Postcard" and "April Fool" campaigns are two distinct, serial instances of self-propagation campaigns, which attempt to install an executable on the user's machine under the guise of being postcard software. For each campaign, Figure 3 shows the number of messages per hour assigned to bots for mailing.

Storm's authors have shown great cunning in exploiting the cultural and social expectations of users—hence the April Fool campaign was rolled out for a limited run around April 1. Our Web site was designed to mimic the earlier

Table 1. Campaigns used in the experiment.

Campaign	Dates	Workers	Emails
Pharmacy	March 21–April 15	31,348	347,590,389
Postcard	March 9–March 15	17,639	83,665,479
April Fool	March 31–April 2	3,678	38,651,124
		Total	469,906,992

Figure 3. Number of email messages assigned per hour for each campaign.

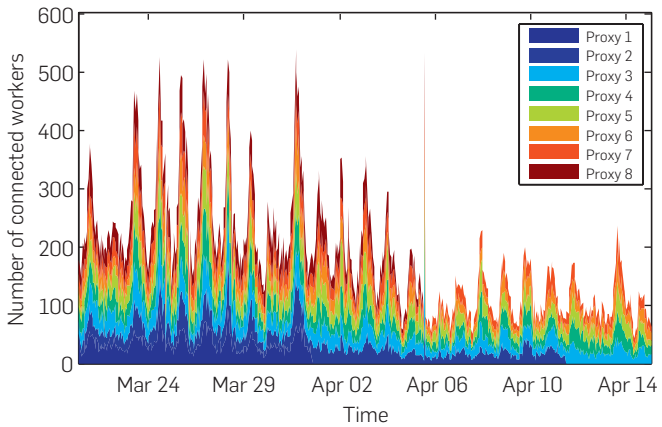


Postcard campaign and thus our data probably does not perfectly reflect user behavior for this campaign, but the two are similar enough in nature that we surmise that any impact is small.

We began the experiment with eight proxy bots, of which seven survived until the end. Figure 4 shows a timeline of the proxy bot workload. The number of workers connected to each proxy is roughly uniform across all proxies (23 worker bots on average), but shows strong spikes corresponding to new self-propagation campaigns. At peak, 539 worker bots were connected to our proxies at the same time.

Most workers only connected to our proxies once: 78% of the workers only connected to our proxies a single time, 92% at most twice, and 99% at most five times. The most prolific worker IP address, a host in an academic network in North Carolina, USA, contacted our proxies 269 times; further inspection identified this as a NAT egress point for 19 individual infections. Conversely, most workers do not connect to more than one proxy: 81% of the workers only connected to a single proxy, 12% to two, 3% to four, 4% connected to five

Figure 4. Timeline of proxy bot workload.



or more, and 90 worker bots connected to all of our proxies. On average, worker bots remained connected for 40 min, although over 40% workers connected for less than a minute. The longest connection lasted almost 81 h.

The workers were instructed to send postcard spam to 83,665,479 addresses, of which 74,901,820 (89.53%) are unique. The April Fool campaign targeted 38,651,124 addresses, of which 36,909,792 (95.49%) are unique. Pharmacy spam targeted 347,590,389 addresses, of which 213,761,147 (61.50%) are unique.

Spam Conversion Pipeline: Conceptually, we break down spam conversion into a pipeline with five "filtering" stages Figure 5 illustrates this pipeline and shows the type of filtering at each stage. The pipeline starts with delivery lists of target email addresses sent to worker bots (Stage A). For a wide range of reasons, workers will successfully deliver only a subset of their messages to an MTA (Stage B). At this point, spam filters at the site correctly identify many messages as spam, and drop them or place them aside in a spam folder. The remaining messages have survived the gauntlet and appear in a user's inbox as valid messages (Stage C). Users may delete or otherwise ignore them, but some users will act on the spam, click on the URL in the message, and visit the advertised site (Stage D). These users may browse the site, but only a fraction "convert" on the spam (Stage E) by attempting to purchase products (pharmacy) or by downloading and running an executable (self-propagation).

We show the spam flow in two parts, "crawler" and "converter," to differentiate between real and masquerading users. For example, the delivery lists given to workers contain honeypot email addresses. Workers deliver spam to these honeypots, which then use crawlers to access the sites referenced by the URL in the messages. Since we want to measure the spam conversion rate for actual users, we separate out the effects of automated processes like crawlers, including only clicks we believe to be user-generated in our results.

Table 2 shows the effects of filtering at each stage of the conversion pipeline for both the self-propagation and pharmaceutical campaigns. The number of targeted addresses (A) is simply the total number of addresses on the delivery lists received by the worker bots during the measurement period, excluding the test addresses we injected.

We obtain an estimate of the number of messages delivered to a mail server (B) by relying on delivery reports generated by the workers. The number of messages delivered to a user's inbox (C) is a much harder value to estimate. We do not know what spam filtering, if any, is used by each mail provider, and then by each user individually, and therefore cannot reasonably estimate this number in total. It is possible, however, to determine this number for individual mail providers or spam filters. The three mail providers and the spam filtering appliance we used in this experiment had a method for separating delivered mails into "junk" and inbox categories. Table 3 gives the number of messages delivered a user's inbox for the free email providers, which together accounted for about 16.5% of addresses targeted by Storm (Table 3), as well as our department's commercial spam filtering appliance. It is important to note that these are results from one spam campaign over a short period of time and should not be used as measures of the relative effectiveness for each service. That said, we observe that the popular Web mail providers all do a very a good job at filtering the campaigns we observed, although it is clear they use different methods (e.g., Hotmail rejects most Storm spam at the mail server level, while Gmail accepts a significant fraction only to filter it later as junk).

The number of visits (D) is the number of accesses to our emulated pharmacy and postcard sites, excluding any crawlers. We note that crawler requests came from a small fraction of hosts but accounted for the majority of all requests to our sites. For the pharmacy site, for instance, of the 11,720 unique IP addresses seen accessing the site with a valid unique identifier, only 10.2% were blacklisted as crawlers. In contrast, 55.3% of all unique identifiers used in requests originated from these crawlers. For all nonimage requests made, 87.43% were made by blacklisted IP addresses.

The number of conversions (E) is the number of visits to the purchase page of the pharmacy site, or the number of executions of the fake self-propagation program.

Figure 5. The spam conversion pipeline.

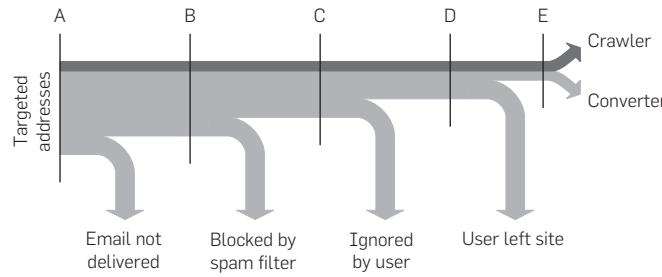


Table 2. Filtering at each stage of the spam conversion pipeline for the self-propagation and pharmacy campaigns. Percentages refer to the conversion rate relative to Stage A.

Stage	Pharmacy		Postcard		April Fool	
A—Spam Targets	347,590,389	100%	83,655,479	100%	40,135,487	100%
B—MTA delivery(est.)	82,700,000	23.8%	21,100,000	25.2%	10,100,000	25.2%
C—Inbox delivery	–	–	–	–	–	–
D—User site visits	10,522	0.00303%	3,827	0.00457%	2,721	0.00680%
E—User conversions	28	0.0000081%	316	0.000378%	225	0.000561%

Table 3. Number of messages delivered to a user's inbox as a fraction of those injected for test accounts at free email providers and a commercial spam filtering appliance. The test account for the Barracuda appliance was not included in the Postcard campaign.

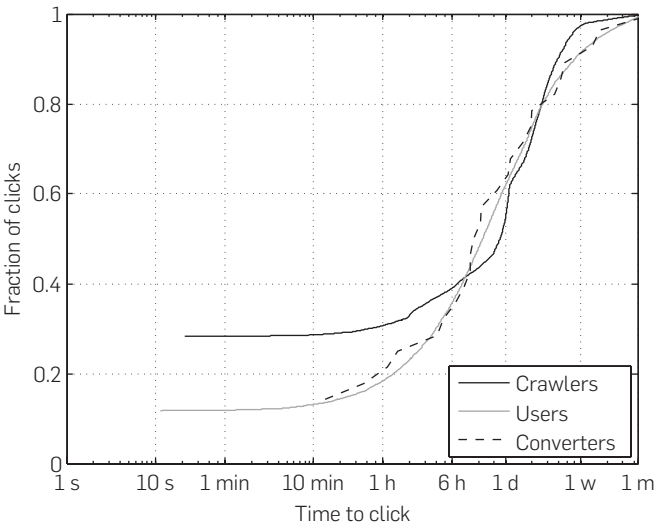
Spam Filter	Pharmacy	Postcard	April Fool
Gmail	0.00683%	0.00176%	0.00226%
Yahoo	0.00173%	0.000542%	None
Hotmail	None	None	None
Barracuda	0.131%	N/A	0.00826%

Our results for Storm spam campaigns show that the spam conversion rate is quite low. For example, out of 350 million pharmacy campaign emails only 28 conversions resulted (and no crawler ever completed a purchase so errors in crawler filtering plays no role). However, a very low conversion rate does not necessary imply low revenue or profitability. We discuss the implications of the conversion rate on the spam conversion proposition further in Section 8.

Time-to-Click: The conversion pipeline shows what fraction of spam ultimately resulted in visits to the advertised sites. However, it does not reflect the latency between when the spam was sent and when a user clicked on it. The longer it takes users to act, the longer the scam hosting infrastructure will need to remain available to extract revenue from the spam.[2] Put another way, how long does a spam-advertised site need to be online to collect potential revenue?

Figure 6 shows the cumulative distribution of the "time-to-click" for accesses to the pharmacy site. The time-to-click is the time from when spam is sent (when a proxy forwards a spam workload to a worker bot) to when a user "clicks" on the URL in the spam (when a host first accesses the Web site). The graph shows three distributions for the accesses by all users, the users who visited the purchase page ("converters"), and the automated crawlers (14,716 such accesses).

Figure 6. Time-to-click distributions for accesses to the pharmacy site.



The user and crawler distributions show distinctly different behavior. Almost 30% of the crawler accesses are within 20 s of worker bots sending spam. This behavior suggests that these crawlers are configured to scan sites advertised in spam immediately upon delivery. Another 10% of crawler accesses have a time-to-click of 1 day, suggesting crawlers configured to access spam-advertised sites periodically in batches. In contrast, only 10% of the user population accesses spam URLs immediately, and the remaining distribution is smooth without any distinct modes. The distributions for all users and users who "convert" are roughly similar, suggesting little correlation between time-to-click and whether a user visiting a site will convert. While most user visits occur within the first 24 h, 10% of times-to-click are a week to a month, indicating that advertised sites need to be available for long durations to capture full revenue potential.

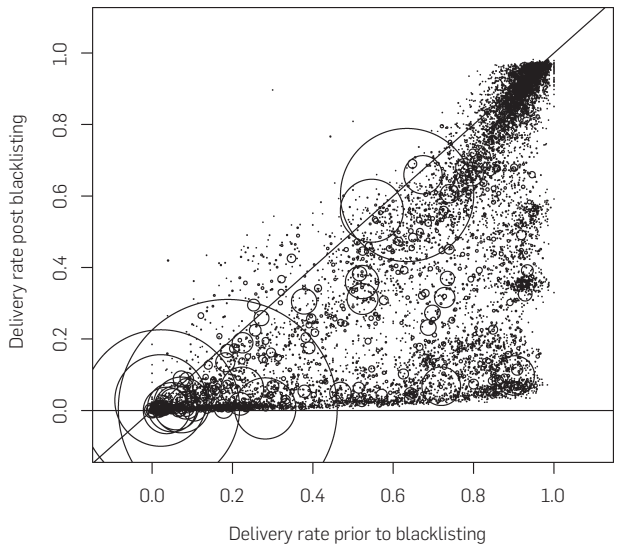
6. EFFECTS OF BLACKLISTING

A major effect on the efficacy of spam delivery is the employment by numerous ISPs of address-based blacklisting to reject email from hosts previously reported as sourcing spam. To assess the impact of blacklisting, during the course of our experiments we monitored the *Composite Blocking List* (CBL),[6] a blacklist source used by the operators of some of our institutions. At any given time the CBL lists on the order of 4–6 million IP addresses that have sent email to various spamtraps. We were able to monitor the CBL from March 21–April 2, 2008, from the start of the pharmacy campaign until the end of the April Fool campaign.

We downloaded the current CBL blacklist every half hour, enabling us to determine which worker bots in our measurements were present on the list and how their arrival on the list related to their botnet activity. Of 40,864 workers that sent delivery reports, fully 81% appeared on the CBL. Of those appearing at some point on the list, 77% were on the list prior to our observing their receipt of spamming directives, appearing first on the list 4.4 days (median) earlier. Of those not initially listed but then listed subsequently, the median interval until listing was 1.5 h, strongly suggesting that the spamming activity we observed them being instructed to conduct quickly led to their detection and blacklisting. Of hosts never appearing on the list, more than 75% never reported successful delivery of spam, indicating that the reason for their lack of listing was simply their inability to effectively annoy anyone.

We would expect that the impact of blacklisting on spam delivery strongly depends on the domain targeted in a given email, since some domains incorporate blacklist feeds such as the CBL into their mailer operations and others do not. To explore this effect, Figure 7 plots the per-domain delivery rate: the number of spam emails that workers reported as successfully delivered to the domain divided by number attempted to that domain. The x-axis shows the delivery rate for spams sent by a worker prior to its appearance in the CBL, and the y-axis shows the rate after its appearance in the CBL. We limit the plot to the 10,879 domains to which workers attempted to deliver at least 1,000 spams. We plot

Figure 7. Change in per-domain delivery rates as seen prior to a worker bot appearing in the blacklist (*x*-axis) vs. after appearing (*y*-axis). Each circle represents a domain targeted by at least 1,000 analyzable deliveries, with the radius scaled in proportion to the number of delivery attempts.



delivery rates for the two different campaigns as separate circles, though the overall nature of the plot does not change between them. The radius of each plotted circle scales in proportion to the number of delivery attempts, the largest corresponding to domains such as hotmail.com, yahoo.com, and gmail.com.

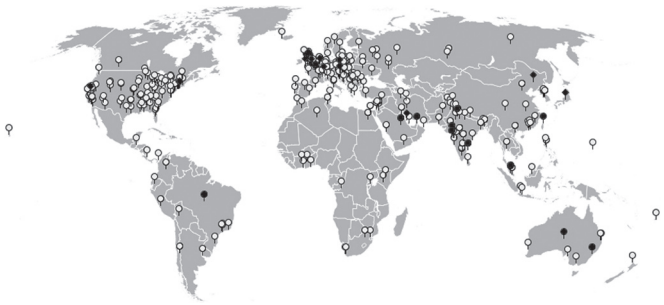
From the plot we clearly see a range of blacklisting behavior by different domains. Some employ other effective antispam filtering, indicated by their appearance near the origin—spam did not get through even prior to appearing on the CBL blacklist. Some make heavy use of either the CBL or a similar list (*y*-axis near zero, but *x*-axis greater than zero), while others appear insensitive to blacklisting (those lying on the diagonal). Since points lie predominantly below the diagonal, we see that either blacklisting or some other effect related to sustained spamming activity (e.g., learning content signatures) diminishes the delivery rate seen at most domains. Delisting followed by relisting may account for some of the spread of points seen here; those few points above the diagonal may simply be due to statistical fluctuations. Finally, the cloud of points to the upper right indicates a large number of domains that are not targeted much individually, but collectively comprise a significant population that appears to employ no effective antispam measures.

7. CONVERSION ANALYSIS

We now turn to a preliminary look at possible factors influencing response to spam. For the present, we confine our analysis to coarse-grained effects.

We start by mapping the geographic distribution of the hosts that "convert" on the spam campaigns we monitored. Figure 8 maps the locations of the 541 hosts that execute the emulated self-propagation program, and the

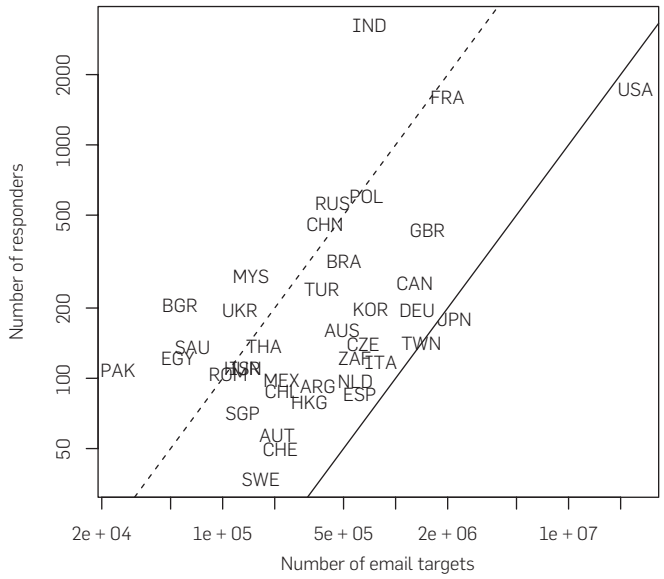
Figure 8. Geographic locations of the hosts that "convert" on spam: the 541 hosts that execute the emulated self-propagation program (light gray), and the 28 hosts that visit the purchase page of the emulated pharmacy site (black).



28 hosts that visit the purchase page of the emulated pharmacy site. The map shows that users around the world respond to spam.

Figure 9 looks at differences in response rates among nations as determined by prevalent country-code email domain TLDs. To allow the inclusion of generic TLDs such as `.com`, for each email address we consider it a member of the country hosting its mail server; we remove domains that resolve to multiple countries, categorizing them as "international" domains. The *x*-axis shows the volume of email (log-scaled) targeting a given country, while the *y*-axis gives the number of responses recorded at our Web servers (also log-scaled), corresponding to Stages A and D in the pipeline (Figure 5), respectively. The solid line reflects a response rate of 10^{-4} and the dashed line a rate of 10^{-3}. Not surprisingly, we see that the spam campaigns target email addresses in the United States substantially more than any other

Figure 9. Volume of email targeting (*x*-axis) vs. responses (*y*-axis) for the most prominent country-code TLDs. The *x* and *y* axes correspond to Stages A and D in the pipeline (Figure 5), respectively.



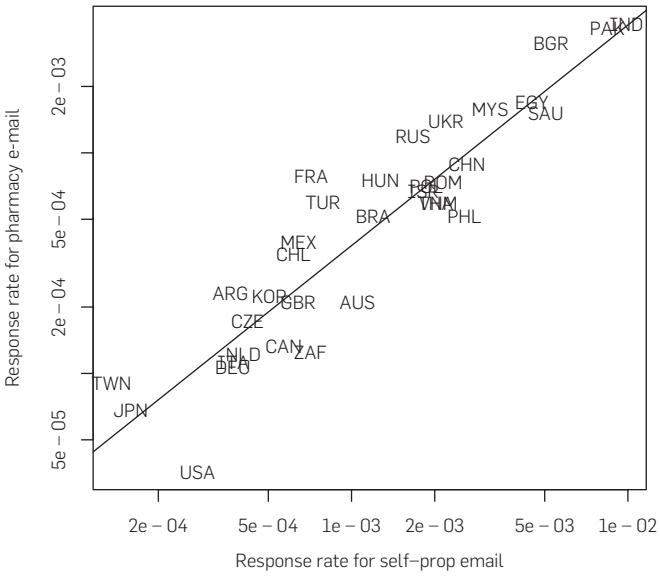
country. Further, India, France, and the United States dominate responses. In terms of response *rates*, however, India, Pakistan, and Bulgaria have the highest response rates than any other countries (furthest away from the diagonal). The United States, although a dominant target and responder, has the lowest resulting response rate of any country, followed by Japan and Taiwan.

However, the countries with predominant response rates do not appear to reflect a heightened interest in users from those countries in the specific spam offerings. Figure 10 plots the rates for the most prominent countries responding to self-propagation vs. pharmacy spams. The median ratio between these two rates is 0.38 (diagonal line). We see that India and Pakistan in fact exhibit almost exactly this ratio (upper-right corner), and Bulgaria is not far from it. Indeed, only a few TLDs exhibit significantly different ratios, including the United States and France, the two countries other than India with a high number of responders; users in the United States respond to the self-propagation spam substantially more than pharmaceutical spam and vice versa with users in France. These results suggest that, for the most part, per-country differences in response rate are due to *structural* causes (quality of spam filtering, user education) rather than differing degrees of cultural or national interest in the particular promises or products conveyed by the spam.

8. CONCLUSION

This paper describes what we believe is the first large-scale quantitative study of spam conversion. We developed a methodology that uses botnet infiltration to indirectly instrument spam emails such that user clicks on these messages are taken to replica Web sites under our control. Using this methodology we instrumented almost 500 million spam messages, comprising three major campaigns, and quantitatively characterized both the delivery process and the conversion rate.

Figure 10. Response rates (stage D in the pipeline) by TLD for executable download (*x*-axis) vs. pharmacy visits (*y*-axis).



We would be the first to admit that these results represent a single data point and are not necessarily representative of spam as a whole. Different campaigns, using different tactics and marketing different products will undoubtedly produce different outcomes. Indeed, we caution *strongly* against researchers using the conversion rates we have measured for these Storm-based campaigns to justify assumptions in any other context. At the same time, it is tempting to speculate on what the numbers we have measured might *mean*. We succumb to this temptation below, with the understanding that few of our speculations can be empirically validated at this time.

After 26 days, and almost 350 million email messages, only 28 sales resulted—a conversion rate of well under 0.00001%. Of these, all but one was for male-enhancement products and the average purchase price was close to $100. Taken together, these conversions would have resulted in revenues of $2,731.88—a bit over $100 a day for the measurement period or $140 per day for periods when the campaign was active. However, our study interposed on only a small fraction of the overall Storm network—we estimate roughly 1.5% based on the fraction of worker bots we proxy. Thus, the total daily revenue attributable to Storm's pharmacy campaign is likely closer to $7000 (or $9500 during periods of campaign activity). By the same logic, we estimate that Storm self-propagation campaigns can produce between 3500 and 8500 new bots per day.

Under the assumption that our measurements are representative over time (an admittedly dangerous assumption when dealing with such small samples), we can extrapolate that, were it sent continuously at the same rate, Storm-generated pharmaceutical spam would produce roughly 3.5 million dollars of revenue in a year. This number could be even higher if spam-advertised pharmacies experience repeat business, a bit less than "millions of dollars every day," but certainly a healthy enterprise.

The next obvious question is, "How much of this revenue is profit?" Here things are even murkier. First, we must consider how much of the gross revenue is actually recovered on a sale. Assuming the pharmacy campaign drives traffic to an affiliate program (and there are very strong anecdotal reasons to believe this is so) then the gross revenue is likely split between the affiliate and the program (an annual net revenue of $1.75 million using our previous estimate). Next, we must subtract business costs. These include a number of incidental expenses (domain registration, bullet-proof hosting fees, etc.) that are basically fixed sunk costs, and the cost to distribute the spam itself.

Anecdotal reports place the *retail* price of spam delivery at a bit under $80 per million.[14] In an examination we conducted of some spam-for-hire service advertisements, we found prices ranging from $70 to over $100 per million for delivery to US addresses, with substantial discounts available for large volumes. This cost is an order of magnitude less than what legitimate commercial mailers charge, but is still a significant overhead; sending 350M emails would cost more than $25,000. Indeed, given the net revenues we

estimate, retail spam delivery would only make sense if it were 20 times cheaper still.

And yet, Storm continues to distribute pharmacy spam—suggesting that it is in fact profitable. One explanation is that Storm's masters are vertically integrated and the purveyors of Storm's pharmacy spam are none other than the operators of Storm itself (i.e., that Storm does not deliver these spams for a third-part in exchange for a fee). There is some evidence for this, since the distribution of target email domain names between the self-propagation and pharmacy campaigns is virtually identical. Since the self-propagation campaigns fundamentally must be run by the botnet's owners, this suggests the purveyor of the pharmacy spam is one and the same. A similar observation can be made in the harvesting of email addresses from the local hard drives of Storm hosts. These email addresses subsequently appear in the target address lists of the pharmacy campaign and self-propagation campaigns alike. Moreover, neither of these behaviors is found in any of the other (smaller) campaigns distributed by Storm (suggesting that these may in fact be fee-for-service distribution arrangements). If true, then the cost of distribution is largely that of the labor used in the development and maintenance of the botnet software itself. While we are unable to provide any meaningful estimates of this cost (since we do not know which labor market Storm is developed in), we surmize that it is roughly the cost of two or three good programmers.

If true, this hypothesis is heartening since it suggests that the third-party *retail* market for spam distribution has not grown large or efficient enough to produce competitive pricing and thus, that profitable spam campaigns require organizations that can assemble complete "soup-to-nuts" teams. Put another way, the profit margin for spam (at least for this one pharmacy campaign) may be meager enough that spammers must be sensitive to the details of how their campaigns are run and are economically susceptible to new defenses.

Acknowledgments

This was one of the most complex measurement studies our group has ever conducted and would have been impossible without the contributions of a large and supportive cast. Here we offer our thanks for their insightful feedback and individual contributions to our effort.

Jordan Hayes provided decidedly nontrivial help with site domain registration. Peter Blair, Paul Karkas, Jamie Knight, and Garrick Lau at Tucows supported this activity (once we convinced them we were not spammers) and allowed us to use reputable registrars. Randy Bush provided overall guidance and help concerning Internet operations and policy issues while Erin Kenneally advised us on legal issues. Brian Kantor set up and managed our DNS, Web, and SMTP servers, while Scott Campbell and Stephen Chan performed massive DNS lookups for us. Jef Poskanzer provided data access for debugging our experiment, Stephen Chenette provided technical assistance and Fallon Chen was our in-house graphic designer. Bill Young and Gregory Ruiz-Ade set up target email accounts in UCSD's CSE department. Special thanks to Gabriel Lawrence and Jim Madden of UCSD's ACT for supporting this activity on UCSD's systems and networks. Finally, our thanks to the anonymous reviewers for their time and commentary.

This work was made possible by the National Science Foundation grants NSF-0433702 and NSF-0433668 and by generous research, operational and in-kind support from Cisco, Microsoft, HP, Intel, VMWare, ESnet, the Lawrence Berkeley National Laboratory, and UCSD's Center for Networked Systems. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of these organizations.

References

1. Akass, C. Storm worm 'making millions a day.' http://www.pcw.co.uk/personal-computer-world/news/2209293/strom-worm-making-millions-day, February 2008.
2. Anderson, D.S., Fleizach, C., Savage, S., Voelker, G.M. Spamscatter: Characterizing internet scam hosting infrastructure. In *Proceedings of the USENIX Security Symposium* (Boston, MA, August 2007).
3. Angwin, J. Elusive Spammer Sends EarthLink on Long Chase. http://online.wsj.com/article_email/SB105225593382372600.html, May 2003.
4. D. M. Association. DMA Releases 5th Annual 'Response Rate Trends Report.' http://www.the-dma.org/cgi/disppressrelease?article=1008, October 2007.
5. Boehme, R., Ho, T. The effect of stock spam on financial markets. In *Proceedings of the 5th Workshop on the Economics of Information Security (WEIS)* (June 2006).
6. Composite Blocking List (CBL). http://cbl.abuseat.org/, March 2008.
7. Frieder, L., Zittrain, J. Spam works: evidence from stock touts and corresponding market activity. *Berkman Center Research Publication*, 2006.
8. Goodman, J., Rounthwaite, R. Stopping outgoing spam. *Proceedings of the 5th ACM Conference on Electronic Commerce* (2004), 30–39.
9. Hanke, M., Hauser, F. On the effects of stock spam emails. *J. Financ. Mark. 11*, 1 (2008), 57–83.
10. Kirk, J. Former Spammer: 'I Know I'm Going to Hell.' http://www.macworld.com/article/58997/2007/07/spammer.html, July 2007.
11. Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G.M., Paxson, V., Savage, S. On the Spam Campaign Trail. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*, April 2008.
12. Judge, W.Y.P., Alperovitch, D. Understanding and Reversing the Profit Model of Spam. In *Workshop on Economics of Information Security 2005 (WEIS 2005)* (Boston, MA, USA, June 2005).
13. Watson, D. All Spammers Go to Hell (posting to funsec list). http://www.mail-archive.com/funsec%40linuxbox.org/msg03346.html, July 2007.
14. Wilson, T. Competition May Be Driving Surge in Botnets, Spam. http://www.darkreading.com/document.asp?doc_id=142690, 2008.

Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, and Stefan Savage

Department of Computer Science and Engineering University of California, San Diego.

Christian Kreibich and Vern Paxson

International Computer Science Institute Berkeley.

© 2009 ACM 0001-0782/09/0800 $10.00

Click Trajectories: End-to-End Analysis of the Spam Value Chain

Kirill Levchenko* Andreas Pitsillidis* Neha Chachra* Brandon Enright* Márk Félegyházi‡ Chris Grier†
Tristan Halvorson* Chris Kanich* Christian Kreibich†◇ He Liu* Damon McCoy*
Nicholas Weaver†◇ Vern Paxson†◇ Geoffrey M. Voelker* Stefan Savage*

*Department of Computer Science and Engineering
University of California, San Diego

†Computer Science Division
University of California, Berkeley

◇International Computer Science Institute
Berkeley, CA

‡Laboratory of Cryptography and System Security (CrySyS)
Budapest University of Technology and Economics

***Abstract*—Spam-based advertising is a business. While it has engendered both widespread antipathy and a multi-billion dollar anti-spam industry, it continues to exist because it fuels a profitable enterprise. We lack, however, a solid understanding of this enterprise's full structure, and thus most anti-spam interventions focus on only one facet of the overall spam value chain (e.g., spam filtering, URL blacklisting, site takedown). In this paper we present a holistic analysis that quantifies the full set of resources employed to monetize spam email—including naming, hosting, payment and fulfillment—using extensive measurements of three months of diverse spam data, broad crawling of naming and hosting infrastructures, and over 100 purchases from spam-advertised sites. We relate these resources to the organizations who administer them and then use this data to characterize the relative prospects for defensive interventions at each link in the spam value chain. In particular, we provide the first strong evidence of payment bottlenecks in the spam value chain; 95% of spam-advertised pharmaceutical, replica and software products are monetized using merchant services from just a handful of banks.**

I. Introduction

We may think of email spam as a scourge—jamming our collective inboxes with tens of billions of unwanted messages each day—but to its perpetrators it is a potent marketing channel that taps latent demand for a variety of products and services. While most attention focuses on the problem of spam *delivery*, the email vector itself comprises only the *visible* portion of a large, multi-faceted business enterprise. Each click on a spam-advertised link is in fact just the start of a long and complex trajectory, spanning a range of both technical and business components that together provide the necessary infrastructure needed to monetize a customer's visit. Botnet services must be secured, domains registered, name servers provisioned, and hosting or proxy services acquired. All of these, in addition to payment processing, merchant bank accounts, customer service, and fulfillment, reflect necessary elements in the spam value chain.

While elements of this chain have received study in isolation (e.g., dynamics of botnets [20], DNS fast-flux networks [17], [42], Web site hosting [1], [22]), the relationship between them is far less well understood. Yet it is these very relationships that capture the structural dependencies—and hence the potential *weaknesses*—within the spam ecosystem's business processes. Indeed, each distinct *path* through this chain—registrar, name server, hosting, affiliate program, payment processing, fulfillment—directly reflects an "entrepreneurial activity" by which the perpetrators muster capital investments and business relationships to create value. Today we lack insight into even the most basic characteristics of this activity. How many organizations are complicit in the spam ecosystem? Which points in their value chains do they share and which operate independently? How "wide" is the bottleneck at each stage of the value chain—do miscreants find alternatives plentiful and cheap, or scarce, requiring careful husbanding?

The desire to address these kinds of questions empirically—and thus guide decisions about the most effective mechanisms for addressing the spam problem—forms the core motivation of our work. In this paper we develop a methodology for characterizing the end-to-end resource dependencies ("trajectories") behind individual spam campaigns and then analyze the relationships among them. We use three months of real-time source data, including captive botnets, raw spam feeds, and feeds of spam-advertised URLs to drive active probing of spam infrastructure elements (name servers, redirectors, hosting proxies). From these, we in turn identify those sites advertising three popular classes of goods—pharmaceuticals, replica luxury goods and counterfeit software—as well as their membership in specific affiliate programs around which the overall business is structured. Finally, for a subset of these sites we perform on-line purchases, providing additional data about merchant bank affiliation, customer service, and fulfillment. Using this data we characterize the resource footprint at each step in the spam value chain, the extent of sharing between spam organizations and, most importantly, the relative prospects for interrupting spam monetization at different stages of the process.

The remainder of this paper is organized as follows. Section II provides a qualitative overview of the spam ecosystem coupled with a review of related research.

Section III describes the data sources, measurement techniques and post-processing methodology used in our study. Section IV describes our analysis of spam activities between August and October of 2010, and the implications of these findings on the likely efficacy of different anti-spam interventions, followed by our conclusions in Section V.

II. Background and Related Work

As an advertising medium, spam ultimately shares the underlying business model of all advertising. So long as the revenue driven by spam campaigns exceeds their cost, spam remains a profitable enterprise. This glib description belies the complexity of the modern spam business. While a decade ago spammers might have handled virtually all aspects of the business including email distribution, site design, hosting, payment processing, fulfillment, and customer service [33], today's spam business involves a range of players and service providers. In this section, we review the broad elements in the spam value chain, the ways in which these components have adapted to adversarial pressure from the anti-spam community, and the prior research on applied e-crime economics that informs our study.

A. How Modern Spam Works

While the user experience of spam revolves principally around the email received, these constitute just one part of a larger value chain that we classify into three distinct stages: *advertising*, *click support*, and *realization*. Our discussion here reflects the modern understanding of the degree to which specialization and affiliate programs dominate the use of spam to sell products. To this end, we draw upon and expand the narrative of the "Behind Online Pharma" project [4], which documents the experience of a group of investigative journalists in exploring the market structure for online illegal pharmaceuticals; and Samosseiko's recent overview [46] of affiliate programs, including many that we discuss in this paper.

Advertising. Advertising constitutes all activities focused on reaching potential customers and enticing them into clicking on a particular URL. In this paper we focus on the email spam vector, but the same business model occurs for a range of advertising vectors, including blog spam [39], Twitter spam [12], search engine optimization [53], and sponsored advertising [26], [27]. The delivery of email spam has evolved considerably over the years, largely in response to increasingly complex defensive countermeasures. In particular, large-scale efforts to shut down open SMTP proxies and the introduction of well-distributed IP blacklisting of spam senders have pushed spammers to using more sophisticated delivery vehicles. These include botnets [13], [20], [56], Webmail spam [9], and IP prefix hijacking [45]. Moreover, the market for spam services has stratified over time; for example, today it is common for botnet operators to rent their services to spammers on a contract basis [40].

The advertising side of the spam ecosystem has by far seen the most study, no doubt because it reflects the part of spam that users directly experience. Thus, a broad and ongoing literature examines filtering spam email based on a variety of content features (e.g., [2], [19], [43], [57]). Similarly, the network characteristics of spam senders have seen extensive study for characterizing botnet membership [58], identifying prefix hijacking [45], classifying domains and URLs [14], [32], [44], [55], [56], and evaluating blacklists [47], [48]. Finally, we note that most commercial anti-spam offerings focus exclusively on the delivery aspect of spam. In spite of this attention, spam continues to be delivered and thus our paper focuses strictly on the remaining two stages of the spam monetization pipeline.

Click support. Having delivered their advertisement, a spammer depends on some fraction of the recipients to respond, usually by clicking on an embedded URL and thus directing their browser to a Web site of interest. While this process seems simple, in practice a spammer must orchestrate a great many moving parts and maintain them against pressure from defenders.

Redirection sites. Some spammers directly advertise a URL such that, once the recipient's browser resolves the domain and fetches the content from it, these steps constitute the fullness of the promoted Web site. However, a variety of defensive measures—including URL and domain blacklisting, as well as site takedowns by ISPs and domain takedowns by registrars—have spurred more elaborate steps. Thus, many spammers advertise URLs that, when visited, redirect to additional URLs [1], [22]. Redirection strategies primarily fall into two categories: those for which a legitimate third party inadvertently controls the DNS name resource for the redirection site (e.g., free hosting, URL shorteners, or compromised Web sites), and those for which the spammers themselves, or perhaps parties working on their behalf, manage the DNS name resources (e.g., a "throwaway" domain such as `minesweet.ru` redirecting to a more persistent domain such as `greatjoywatches.com`).

Domains. At some point, a click trajectory will usually require domain name resources managed by the spammer or their accomplices. These names necessarily come via the services of a domain registrar, who arranges for the root-level registry of the associated top-level domain (TLD) to hold NS records for the associated registered domain. A spammer may purchase domains directly from a registrar, but will frequently purchase instead from a domain reseller, from a "domaineer" who purchases domains in bulk via multiple sources and sells to the underground trade, or directly from a spam "affiliate program" that makes domains available to their affiliates as part of their "startup package."

Interventions at this layer of the spam value chain depend significantly on the responsiveness of individual registrars and the pressure brought to bear [29]. For example, a recent industry study by LegitScript and KnujOn documents heavy

concentration of spam-advertised pharmacies with domains registered through a particular set of registrars who appear indifferent to complaints [28].

Name servers. Any registered domain must in turn have supporting name server infrastructure. Thus spammers must provision this infrastructure either by hosting DNS name servers themselves, or by contracting with a third party. Since such resources are vulnerable to takedown requests, a thriving market has arisen in so-called "bulletproof" hosting services that resist such requests in exchange for a payment premium [23].

Web servers. The address records provided by the spammer's name servers must in turn specify servers that host (or more commonly proxy) Web site content. As with name servers, spam-advertised Web servers can make use of bulletproof hosting to resist takedown pressure [3], [51]. Some recent interventions have focused on effectively shutting down such sites by pressuring their upstream Internet service providers to deny them transit connectivity [6].

To further complicate such takedowns and to stymie blacklisting approaches, many spammers further obfuscate the hosting relationship (both for name servers and Web servers) using fast-flux DNS [17], [41], [42]. In this approach, domain records have short-lived associations with IP addresses, and the mapping infrastructure can spread the domain's presence over a large number of machines (frequently many thousands of compromised hosts that in turn proxy requests back to the actual content server [5]). Furthermore, recently innovators have begun packaging this capability to offer it to third parties on a contract basis as a highly resilient content-hosting service [7].

Stores and Affiliate Programs. Today, spammers operate primarily as advertisers, rarely handling the back end of the value chain. Such spammers often work as affiliates of an online store, earning a commission (typically 30–50%) on the sales they bring in [46]. The affiliate program typically provides the storefront templates, shopping cart management, analytics support, and even advertising materials. In addition, the program provides a centralized Web service interface for affiliates to track visitor conversions and to register for payouts (via online financial instruments such as WebMoney). Finally, affiliate programs take responsibility for contracting for payment and fulfillment services with outside parties. Affiliate programs have proven difficult to combat directly—although, when armed with sufficient legal jurisdiction, law enforcement has successfully shut down some programs [8].

Realization. Finally, having brought the customer to an advertised site and convinced them to purchase some product, the seller realizes the latent value by acquiring the customer's payment through conventional payment networks, and in turn fulfilling their product request.

Payment services. To extract value from the broadest possible customer base, stores try to support standard credit card payments. A credit card transaction involves several parties in addition to the customer and merchant: money is transferred from the *issuing bank* (the customer's bank) to the *acquiring bank* (the bank of the merchant) via a *card association network* (i.e., Visa or MasterCard). In addition to the acquiring bank, issuing bank, and card association, the merchant frequently employs the services of a *payment processor* to facilitate this process and act as the technical interface between the merchant and the payment system.

Card associations impose contractual restrictions on their member banks and processors, including the threat of fines and de-association; but to our knowledge little public documentation exists about the extent to which the associations apply this pressure in practice nor the extent to which it plays an important role in moderating the spam business. Evidence from this study suggests that any such pressure is currently insufficient to stop this activity.

Fulfillment. Finally, a store arranges to fulfill an order[1] in return for the customer's payment. For physical goods such as pharmaceuticals and replica products, this involves acquiring the items and shipping them to the customer. Global business-to-business Web sites such as Alibaba, ECPlaza, and ECTrade offer connections with a broad variety of vendors selling a range of such goods, including prepackaged drugs—both brand (e.g., Viagra) and off-brand (e.g., sildenafil citrate capsules)—and replica luxury goods (e.g., Rolex watches or Gucci handbags). Generally, suppliers will offer direct shipping service ("drop shipping"), so affiliate programs can structure themselves around "just in time" fulfillment and avoid the overhead and risk of warehousing and shipping the product themselves.[2] Fulfillment for virtual goods such as software, music, and videos can proceed directly via Internet download.

B. Pharmacy Express: An Example

Figure 1 illustrates the spam value chain via a concrete example from the empirical data used in this study.

On October 27th, the Grum botnet delivered an email titled *VIAGRA® Official Site* (❶). The body of the message includes an image of male enhancement pharmaceutical tablets and their associated prices (shown). The image provides a URL tag and thus when clicked (❷) directs the user's browser to resolve the associated domain name, `medicshopnerx.ru`. This domain was registered by REGRU-REG-RIPN (a.k.a. `reg.ru`) on October 18th (❸)—it is still active as of this writing. The machine providing name service resides in China, while hosting resolves to a

[1] In principle, a store could fail to fulfill a customer's order upon receiving their payment, but this would both curtail any repeat orders and would lead to chargebacks through the payment card network, jeopardizing their relationship with payment service providers.

[2] Individual suppliers can differ in product availability, product quality, the ability to manage the customs process, and deliver goods on a timely basis. Consequently, affiliate programs may use different suppliers for different products and destinations.

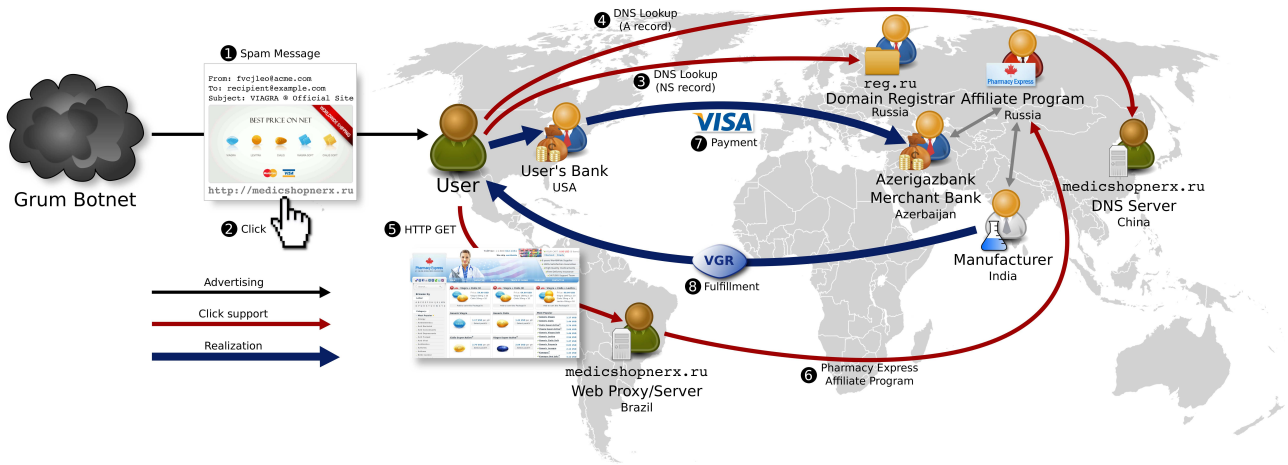


Figure 1: Infrastructure involved in a single URL's value chain, including advertisement, click support and realization steps.

machine in Brazil (❹). The user's browser initiates an HTTP request to the machine (❺), and receives content that renders the storefront for "Pharmacy Express," a brand associated with the Mailien pharmaceutical affiliate program based in Russia (❻).

After selecting an item to purchase and clicking on "Checkout", the storefront redirects the user to a payment portal served from `payquickonline.com` (this time serving content via an IP address in Turkey), which accepts the user's shipping, email contact, and payment information, and provides an order confirmation number. Subsequent email confirms the order, provides an *EMS* tracking number, and includes a contact email for customer questions. The bank that issued the user's credit card transfers money to the acquiring bank, in this case the Azerigazbank Joint-Stock Investment Bank in Baku, Azerbaijan (BIN 404610, ❼). Ten days later the product arrives, blister-packaged, in a cushioned white envelope with postal markings indicating a supplier named PPW based in Chennai, India as its originator (❽).

C. Cybercrime economics

Alongside the myriad studies of the various components employed in spam (e.g., botnets, fast flux, etc.), a literature has recently emerged that focuses on using economic tools for understanding cybercrime (including spam) in a more systematic fashion, with an aim towards enabling better reasoning about effective interventions. Here we highlight elements of this work that have influenced our study.

Some of the earliest such work has aimed to understand the scope of underground markets based on the value of found goods (typically stolen financial credentials), either as seen on IRC chatrooms [10], forums [59], malware "drop-zones" [16], or directly by intercepting communications to botnet C&C servers [50]. Herley and Florêncio critique this line of work as not distinguishing between claimed and true losses, and speculate that such environments inherently reflect "lemon markets" in which few participants are likely to acquire significant profits (particularly spammers) [15]. While this hypothesis remains untested, its outcome is orthogonal to our focus of understanding the structure of the value chain itself.

Our own previous work on spam conversion also used empirical means to infer parts of the return-on-investment picture in the spam business model [21]. By contrast, this study aims to be considerably more comprehensive in breadth (covering what we believe reflect most large spam campaigns) and depth (covering the fullness of the value chain), but offering less precision regarding specific costs.

Finally, another line of work has examined interventions from an economic basis, considering the efficacy of site and domain takedown in creating an economic impediment for cybercrime enterprises (notably phishing) [6], [35], [36]. Molnar *et al.* further develop this approach via comparisons with research on the illicit drug ecosystem [34]. Our work builds on this, but focuses deeply on the spam problem in particular.

III. Data Collection Methodology

In this section we describe our datasets and the methodology by which we collected, processed, and validated them. Figure 2 concisely summarizes our data sources and methods. We start with a variety of full-message spam feeds, URL feeds, and our own botnet-harvested spam (❶). Feed parsers extract embedded URLs from the raw feed data for further processing (❷). A DNS crawler enumerates various resource record sets of the URL's domain, while a farm of Web crawlers visits the URLs and records HTTP-level interactions and landing pages (❸). A clustering tool clusters pages by content similarity (❹). A content tagger labels the content clusters according to the category of goods sold, and the associated affiliate programs (❺). We then make targeted purchases from each affiliate program (❻), and store the feed data and distilled and derived metadata in a database

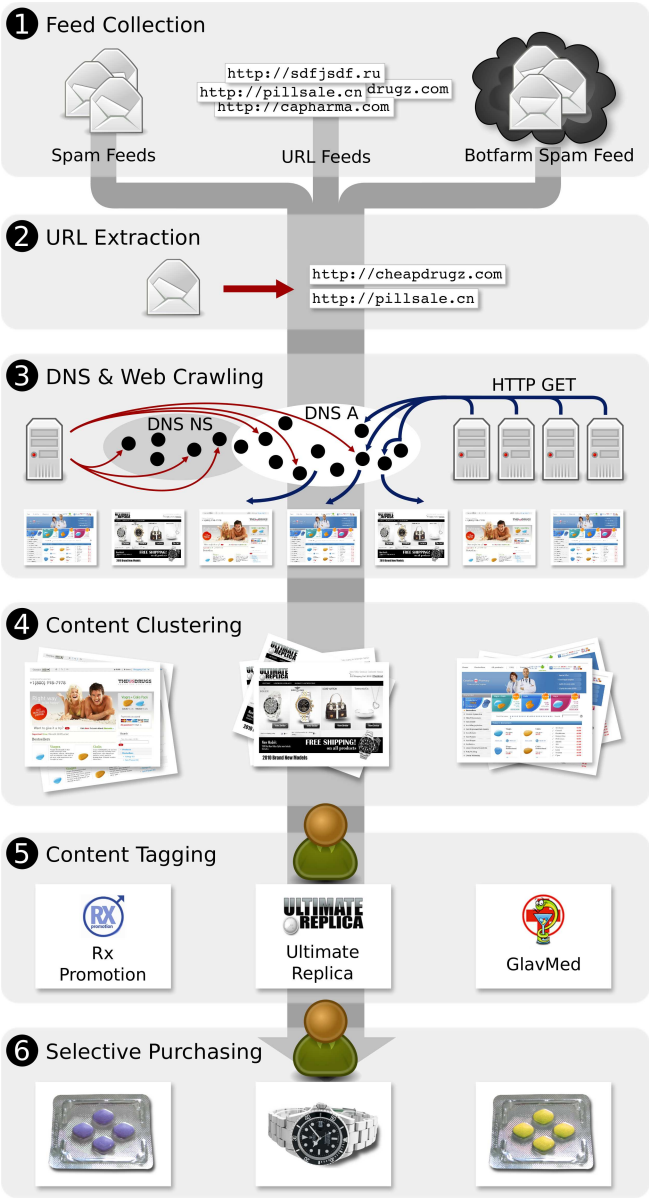


Figure 2: Our data collection and processing workflow.

for subsequent analysis in Section IV. (Steps ❺ and ❻ are partially manual operations, the others are fully automated.)

The rest of this section describes these steps in detail.

A. Collecting Spam-Advertised URLs

Our study is driven by a broad range of data sources of varying types, some of which are provided by third parties, while others we collect ourselves. Since the goal of this study is to decompose the spam ecosystem, it is natural that our seed data arises from spam email itself. More specifically, we focus on the URLs embedded within such email, since these are the vectors used to drive recipient traffic to particular Web sites. To support this goal, we

Feed Name	*Feed Description*	*Received URLs*	*Distinct Domains*
Feed A	MX honeypot	32,548,304	100,631
Feed B	Seeded honey accounts	73,614,895	35,506
Feed C	MX honeypot	451,603,575	1,315,292
Feed D	Seeded honey accounts	30,991,248	79,040
Feed X	MX honeypot	198,871,030	2,127,164
Feed Y	Human identified	10,733,231	1,051,211
Feed Z	MX honeypot	12,517,244	67,856
Cutwail	Bot	3,267,575	65
Grum	Bot	11,920,449	348
MegaD	Bot	1,221,253	4
Rustock	Bot	141,621,731	13,612,815
Other bots	Bot	7,768	4
Total		968,918,303	17,813,952

Table I: Feeds of spam-advertised URLs used in this study. We collected feed data from August 1, 2010 through October 31, 2010.

obtained seven distinct URL feeds from third-party partners (including multiple commercial anti-spam providers), and harvested URLs from our own botfarm environment.

For this study, we used the data from these feeds from August 1, 2010 through October 31, 2010, which together comprised nearly 1 billion URLs. Table I summarizes our feed sources along with the "type" of each feed, the number of URLs received in the feed during this time period, and the number of distinct registered domains in those URLs. Note that the "bot" feeds tend to be focused spam sources, while the other feeds are spam sinks comprised of a blend of spam from a variety of sources. Further, individual feeds, particularly those gathered directly from botnets, can be heavily skewed in their makeup. For example, we received over 11M URLs from the Grum bot, but these only contained 348 distinct registered domains. Conversely, the 13M distinct domains produced by the Rustock bot are artifacts of a "blacklist-poisoning" campaign undertaken by the bot operators that comprised millions of "garbage" domains [54]. Thus, one must be mindful of these issues when analyzing such feed data in aggregate.

From these feeds we extract and normalize embedded URLs and insert them into a large multi-terabyte Postgres database. The resulting "feed tables" drive virtually all subsequent data gathering.

B. Crawler data

The URL feed data subsequently drives active crawling measurements that collect information about both the DNS infrastructure used to name the site being advertised and the Web hosting infrastructure that serves site content to visitors. We use distinct crawlers for each set of measurements.

DNS Crawler: We developed a DNS crawler to identify the name server infrastructure used to support spam-advertised domains, and the address records they specify for hosting those names. Under normal use of DNS this process would be straightforward, but in practice it is significantly

complicated by *fast flux* techniques employed to minimize central points of weakness. Similar to the work of [18], we query servers repeatedly to enumerate the set of domains collectively used for click support (Section II-A).

From each URL, we extract both the fully qualified domain name and the registered domain suffix (for example, if we see a domain `foo.bar.co.uk` we will extract both `foo.bar.co.uk` as well as `bar.co.uk`). We ignore URLs with IPv4 addresses (just 0.36% of URLs) or invalidly formatted domain names, as well as duplicate domains already queried within the last day.

The crawler then performs recursive queries on these domains. It identifies the domains that resolve successfully and their authoritative domains, and filters out unregistered domains and domains with unreachable name servers. To prevent fruitless domain enumeration, it also detects wildcard domains (`abc.example.com`, `def.example.com`, etc.) where all child domains resolve to the same IP address. In each case, the crawler exhaustively enumerates all A, NS, SOA, CNAME, MX, and TXT records linked to a particular domain.

The crawler periodically queries new records until it converges on a set of distinct results. It heuristically determines convergence using standard maximum likelihood methods to estimate when the probability of observing a new unique record has become small. For added assurance, after convergence the crawler continues to query domains daily looking for new records (ultimately timing out after a week if it discovers none).

Web Crawler: The Web crawler replicates the experience of a user clicking on the URLs derived from the spam feeds. It captures any application-level redirects (HTML, JavaScript, Flash), the DNS names and HTTP headers of any intermediate servers and the final server, and the page that is ultimately displayed—represented both by its DOM tree and as a screenshot from a browser. Although straightforward in theory, crawling spam URLs presents a number of practical challenges in terms of scale, robustness, and adversarial conditions.

For this study we crawled nearly 15 million URLs, of which we successfully visited and downloaded correct Web content for over 6 million (unreachable domains, blacklisting, etc., prevent successful crawling of many pages).[3] To manage this load, we replicate the crawler across a cluster of machines. Each crawler replica consists of a controller managing over 100 instances of Firefox 3.6.10 running in parallel. The controller connects to a custom Firefox extension to manage each browser instance, which incorporates the Screengrab! extension [38] to capture screen shots (used for manual investigations). The controller retrieves batches of URLs from the database, and assigns URLs to Firefox instances in a round-robin fashion across a diverse set of IP address ranges.[4]

[3]By comparison, the spam hosting studies of Anderson *et al.* and Konte *et al.* analyzed 150,000 messages per day and 115,000 messages per month respectively [1], [22].

Stage	*Count*	
Received URLs	968,918,303	
Distinct URLs	93,185,779	(9.6%)
Distinct domains	17,813,952	
Distinct domains crawled	3,495,627	
URLs covered	950,716,776	(98.1%)

Table II: Summary results of URL crawling. We crawl the registered domains used by over 98% of the URLs received.

Table II summarizes our crawling efforts. Since there is substantial redundancy in the feeds (e.g., fewer than 10% of the URLs are even unique), crawling every URL is unnecessary and resource inefficient. Instead, we focus on crawling URLs that cover the set of registered domains used by all URLs in the feed. Except in rare instances, all URLs to a registered domain are for the same affiliate program. Thus, the crawler prioritizes URLs with previously unseen registered domains, ignores any URLs crawled previously, and rate limits crawling URLs containing the same registered domain—both to deal with feed skew as well as to prevent the crawler from being blacklisted. For timeliness, the crawler visits URLs within 30 minutes of appearing in the feeds.

We achieve nearly complete coverage: Over 98% of the URLs received in the raw feeds use registered domains that we crawl. Note that we obtain this coverage even though we crawled URLs that account for only 20% of the nearly 18 million distinct registered domains in the feeds. This outcome reflects the inherent skew in the feed makeup. The vast majority of the remaining 80% of domains we did not crawl, and the corresponding 2% URLs that use those domains, are from the domain-poisoning spam sent by the Rustock bot and do not reflect real sites (Section III-A).

C. Content Clustering and Tagging

The crawlers provide low-level information about URLs and domains. In the next stage of our methodology, we process the crawler output to associate this information with higher-level spam business activities.

Note that in this study we exclusively focus on businesses selling three categories of spam-advertised products: pharmaceuticals, replicas, and software. We chose these categories because they are reportedly among the most popular goods advertised in spam [31]—an observation borne out in our data as well.[5]

[4]Among the complexities, scammers are aware that security companies crawl them and blacklist IP addresses they suspect are crawlers. We mitigate this effect by tunneling requests through proxies running in multiple disparate IP address ranges.

[5]We did not consider two other popular categories (pornography and gambling) for institutional and procedural reasons.

Stage	*Pharmacy*	*Software*	*Replicas*	*Total*
URLs	346,993,046	3,071,828	15,330,404	365,395,278
Domains	54,220	7,252	7,530	69,002
Web clusters	968	51	20	1,039
Programs	30	5	10	45

Table III: Breakdown of clustering and tagging results.

To classify each Web site, we use *content clustering* to match sites with lexically similar content structure, *category tagging* to label clustered sites with the category of goods they sell, and *program tagging* to label clusters with their specific affiliate program and/or storefront brand. We use a combination of automated and manual analysis techniques to make clustering and tagging feasible for our large datasets, while still being able to manageably validate our results.

Table III summarizes the results of this process. It lists the number of received URLs with registered domains used by the affiliate programs we study, the number of registered domains in those URLs, the number of clusters formed based on the contents of storefront Web pages, and the number of affiliate programs that we identify from the clusters. As expected, pharmaceutical affiliate programs dominate the data set, followed by replicas and then software. We identify a total of 45 affiliate programs for the three categories combined, that are advertised via 69,002 distinct registered domains (contained within 38% of all URLs received in our feeds). We next describe the clustering and tagging process in more detail.

Content clustering: The first step in our process uses a clustering tool to group together Web pages that have very similar content. The tool uses the HTML text of the crawled Web pages as the basis for clustering. For each crawled Web page, it uses a q-gram similarity approach to generate a fingerprint consisting of a set of multiple independent hash values over all 4-byte tokens of the HTML text. After the crawler visits a page, the clustering tool computes the fingerprint of the page and compares it with the fingerprints representing existing clusters. If the page fingerprint exceeds a similarity threshold with a cluster fingerprint (equivalent to a Jaccard index of 0.75), it places the page in the cluster with the greatest similarity. Otherwise, it instantiates a new cluster with the page as its representative.

Category tagging: The clusters group together URLs and domains that map to the same page content. The next step of category tagging broadly separates these clusters into those selling goods that we are interested in, and those clusters that do not (e.g., domain parking, gambling, etc). We are intentionally conservative in this step, potentially including clusters that turn out to be false positives to ensure that we include all clusters that fall into one of our categories (thereby avoiding false negatives).

We identify interesting clusters using generic keywords found in the page content, and we label those clusters with *category tags*—"pharma", "replica", "software"—that correspond to the goods they are selling. The keywords consist of large sets of major brand names (Viagra, Rolex, Microsoft, etc.) as well as domain-specific terms (herbal, pharmacy, watches, software, etc.) that appear in the storefront page. These terms are tied to the content being sold by the storefront site, and are also used for search engine optimization (SEO). Any page containing a threshold of these terms is tagged with the corresponding keyword. The remaining URLs do not advertise products that we study and they are left untagged.

Even with our conservative approach, a concern is that our keyword matching heuristics might have missed a site of interest. Thus, for the remaining untagged clusters, we manually checked for such false negatives, i.e., whether there were clusters of storefront pages selling one of the three goods that should have a category tag, but did not. We examined the pages in the largest 675 untagged clusters (in terms of number of pages) as well as 1,000 randomly selected untagged clusters, which together correspond to 39% of the URLs we crawled. We did not find any clusters with storefronts that we missed.[6]

Program tagging: At this point, we focus entirely on clusters tagged with one of our three categories, and identify sets of distinct clusters that belong to the same affiliate program. In particular, we label clusters with specific *program tags* to associate them either with a certain affiliate program (e.g., EvaPharmacy—which in turn has many distinct storefront brands) or, when we cannot mechanically categorize the underlying program structure, with an individual storefront "brand" (e.g., Prestige Replicas). From insight gained by browsing underground forum discussions, examining the raw HTML for common implementation artifacts, and making product purchases, we found that some sets of the these brands are actually operated by the same affiliate program.

In total, we assigned program tags to 30 pharmaceutical, 5 software, and 10 replica programs that dominated the URLs in our feeds. Table IV enumerates these affiliate programs and brands, showing the number of distinct registered domains used by those programs, and the number of URLs that use those domains. We also show two aggregate programs, Mailien and ZedCash, whose storefront brands we associated manually based on evidence gathered on underground Web forums (later validated via the purchasing process).[7] The "feed volume" shows the distribution of the affiliate programs as observed in each of the spam "sink" feeds (the feeds *not* from bots), roughly approximating the

[6]The lack of false negatives is not too surprising. Missing storefronts would have no textual terms in their page content that relate to what they are selling (incidentally also preventing the use of SEO); this situation could occur if the storefront page were composed entirely of images, but such sites are rare.

[7]Note, ZedCash is unique among programs as it has storefront brands for each of the herbal, pharmaceutical and replica product categories.

Affiliate Program		*Distinct Domains*	*Received URLs*	*Feed Volume*
RxPrm	RX–Promotion	10,585	160,521,810	24.92%
Mailn	Mailien	14,444	69,961,207	23.49%
PhEx	Pharmacy Express	14,381	69,959,629	23.48%
EDEx	ED Express	63	1,578	0.01%
ZCashPh	ZedCash (Pharma)	6,976	42,282,943	14.54%
DrMax	Dr. Maxman	5,641	32,184,860	10.95%
Grow	Viagrow	382	5,210,668	1.68%
USHC	US HealthCare	167	3,196,538	1.31%
MaxGm	MaxGentleman	672	1,144,703	0.41%
VgREX	VigREX	39	426,873	0.14%
Stud	Stud Extreme	42	68,907	0.03%
ManXt	ManXtenz	33	50,394	0.02%
GlvMd	GlavMed	2,933	28,313,136	10.32%
OLPh	Online Pharmacy	2,894	17,226,271	5.16%
Eva	EvaPharmacy	11,281	12,795,646	8.7%
WldPh	World Pharmacy	691	10,412,850	3.55%
PHOL	PH Online	101	2,971,368	0.96%
Aptke	Swiss Apotheke	117	1,586,456	0.55%
HrbGr	HerbalGrowth	17	265,131	0.09%
RxPnr	RX Partners	449	229,257	0.21%
Stmul	Stimul-cash	50	157,537	0.07%
Maxx	MAXX Extend	23	104,201	0.04%
DrgRev	DrugRevenue	122	51,637	0.04%
UltPh	Ultimate Pharmacy	12	44,126	0.02%
Green	Greenline	1,766	25,021	0.36%
Vrlty	Virility	9	23,528	0.01%
RxRev	RX Rev Share	299	9,696	0.04%
Medi	MediTrust	24	6,156	0.01%
ClFr	Club-first	1,270	3,310	0.07%
CanPh	Canadian Pharmacy	133	1,392	0.03%
RxCsh	RXCash	22	287	<0.01%
Staln	Stallion	2	80	<0.01%
	Total	54,220	346,993,046	93.18%
Royal	Royal Software	572	2,291,571	0.79%
EuSft	EuroSoft	1,161	694,810	0.48%
ASR	Auth. Soft. Resellers	4,117	65,918	0.61%
OEM	OEM Soft Store	1,367	19,436	0.24%
SftSl	Soft Sales	35	93	<0.01%
	Total	7,252	3,071,828	2.12%
ZCashR	ZedCash (Replica)	6,984	13,243,513	4.56%
UltRp	Ultimate Replica	5,017	10,451,198	3.55%
Dstn	Distinction Replica	127	1,249,886	0.37%
Exqst	Exquisite Replicas	128	620,642	0.22%
DmdRp	Diamond Replicas	1,307	506,486	0.27%
Prge	Prestige Replicas	101	382,964	0.1%
OneRp	One Replica	77	20,313	0.02%
Luxry	Luxury Replica	25	8,279	0.01%
AffAc	Aff. Accessories	187	3,669	0.02%
SwsRp	Swiss Rep. & Co.	15	76	<0.01%
WchSh	WatchShop	546	2,086,891	0.17%
	Total	7,530	15,330,404	4.73%
	Grand Total	69,002	365,395,278	100%

Table IV: Breakdown of the pharmaceutical, software, and replica affiliate programs advertising in our URL feeds.

distribution that might be observed by users receiving spam.[8]

To assign these affiliate program tags to clusters, we manually crafted sets of regular expressions that match the page contents of program storefronts. For some programs, we defined expressions that capture the structural nature of the software engine used by all storefronts for a program (e.g., almost all EvaPharmacy sites contained unique hosting conventions). For other programs, we defined expressions that capture the operational modes used by programs that used multiple storefront templates (e.g., GlavMed).[9] For others, we created expressions for individual storefront brands (e.g., one for Diamond Replicas, another for Prestige Replicas, etc.), focusing on the top remaining clusters in terms of number of pages. Altogether, we assigned program tags to clusters comprising 86% of the pages that had category tags.

We manually validated the results of assigning these specific program tags as well. For every cluster with a program tag, we inspected the ten most and least common page DOMs contained in that cluster, and validated that our expressions had assigned them their correct program tags. Although not exhaustive, examining the most and least common pages validates the pages comprising both the "mass" and "tail" of the page distribution in the cluster.

Not all clusters with a category tag ("pharma") had a specific program tag ("EvaPharmacy"). Some clusters with category tags were false positives (they happened to have category keywords in the page, but were not storefronts selling category goods), or they were small clusters corresponding to storefronts with tiny spam footprints. We inspected the largest 675 of these clusters and verified that none of them contained pages that should have been tagged as a particular program in our study.

D. Purchasing

Finally, for a subset of the sites with program tags, we also purchased goods being offered for sale. We attempted to place multiple purchases from each major affiliate program or store "brand" in our study and, where possible, we ordered the same "types" of product from different sites to identify differences or similarities in suppliers based on contents (e.g., lot numbers) and packaging (nominal sender, packaging type, etc.). We attempted 120 purchases, of which 76 authorized and 56 settled.[10]

Of those that settled, all but seven products were delivered. We confirmed via tracking information that two undelivered packages were sent several weeks after our mailbox lease had ended, two additional transactions received no follow-up email, another two sent a follow-up email stating that the order was re-sent after the mailbox lease had ended,

[8]We remove botnet feeds from such volume calculations because their skewed domain mix would bias the results unfairly towards the programs they advertise.

[9]We obtained the full source code for all GlavMed and RX–Promotion sites, which aided creating and validating expressions to match their templates.

[10]Almost 50% of these failed orders were from ZedCash, where we suspect that our large order volume raised fraud concerns. In general, any such biases in the order completion rate do not impact upon our analysis, since our goal in purchasing is simply to establish the binding between individual programs and realization infrastructure; we obtained data from multiple transactions for each major program under study.

and one sent a follow-up email stating that our money had been refunded (this refund, however, had not been processed three months after the fact).

Operational protocol: We placed our purchases via VPN connections to IP addresses located in the geographic vicinity to the mailing addresses used. This constraint is necessary to avoid failing common fraud checks that evaluate consistency between IP-based geolocation, mailing address and the Address Verification Service (AVS) information provided through the payment card association. During each purchase, we logged the full contents of any checkout pages as well as their domain names and IP addresses (frequently different from the sites themselves). We provided contact email addresses hosted on domain names purchased expressly for this project, as several merchants did not allow popular Web-based email accounts during the purchase process. We recorded all email sent to these accounts, as well as the domain names and IP addresses of any customer service sites provided. We also periodically logged into such sites to record the current status of our purchases. For physical goods, we always selected the quickest form of delivery, while software was provided via the Internet (here too we recorded the full information about the sites used for software fulfillment).

All of our purchases were conducted using prepaid Visa payment cards contracted through a specialty issuer. As part of our relationship with the issuer, we maintained the ability to create new cards on demand and to obtain the authorization and settlement records for each transaction. We used a unique card for each transaction.

We had goods shipped to a combination of individual residences and a suite address provided by a local commercial mailbox provider. We regularly picked up, tagged, and photographed shipments and then stored them in a centralized secure facility on our premises. We stored software purchases on a secure hard drive, checked for viruses using Microsoft Security Essentials and Kaspersky Free Trial, and compared against other copies of the same software (including a reference version that we owned).

Legal and ethical concerns: This purchasing portion of our study involved the most careful consideration of legal and ethical concerns, particularly because this level of active involvement has not been common in the academic community to date. We worked with both our own project legal advisors and with general counsel to design a protocol for purchasing, handling, analyzing and disposing of these products within a legal framework that minimizes any risk of harm to others. While the full accounting of the legal considerations are outside the scope of this paper, most of our effort revolved around item selection and controls. For example, we restricted our pharmaceutical purchasing to non-prescription goods such as herbal and over-the-counter products, and we restricted our software purchases to items for which we already possessed a site license (also communicating our intent with the publisher). We did not use any received products (physical or electronic) and, aside from a few demonstration lots, they are scheduled to be destroyed upon the completion of our analyses.

Finally, while these controls are designed to prevent any explicit harm from resulting through the study, a remaining issue concerns the ethics of any implicit harm caused by supporting merchants (through our purchasing) who are themselves potentially criminal or unethical. Since our study does not deal with human subjects our institutional review board did not deem it appropriate for their review. Thus, our decision to move forward is based on our own subjective evaluation (along with the implicit oversight we received from university counsel and administration). In this, we believe that, since any such implicit support of these merchants is small (no individual affiliate program received more than $277 dollars from us), the potential value from better understanding their ecosystem vastly outweighs the potential harm.[11]

IV. Analysis

A major goal of our work is to identify any "bottlenecks" in the spam value chain: opportunities for disrupting monetization at a stage where the fewest alternatives are available to spammers (and ideally for which switching cost is high as well). Thus, in this section we focus directly on analyzing the degree to which affiliate programs *share* infrastructure, considering both the *click support* (i.e., domain registration, name service and Web hosting service) and *realization* (i.e., payment and fulfillment) phases of the spam value chain. We explore each of these in turn and then return to consider the potential effectiveness of interventions at each stage.

A. Click Support

As described in Section III we crawl a broad range of domains—covering the domains found in over 98% of our spam feed URLs—and use clustering and tagging to associate the resulting Web sites with particular affiliate programs. This data, in combination with our DNS crawler and domain WHOIS data, allows us to associate each such domain with an affiliate program and its various click support resources (registrar, set of name server IP addresses and set of Web hosting IP addresses). However, before we proceed with our analysis, we first highlight the subtleties that result from the use of Web site *redirection*.

Redirection: As we mentioned, some Web sites will redirect the visitor from the initial domain found in a spam message to one or more additional sites, ultimately resolving the final Web page (we call the domain for this page the "final domain"). Thus, for such cases one could choose to measure the infrastructure around the "initial domains" or the "final domains".

[11]This is similar to the analysis made in our previous study of the CAPTCHA-solving ecosystem [37].

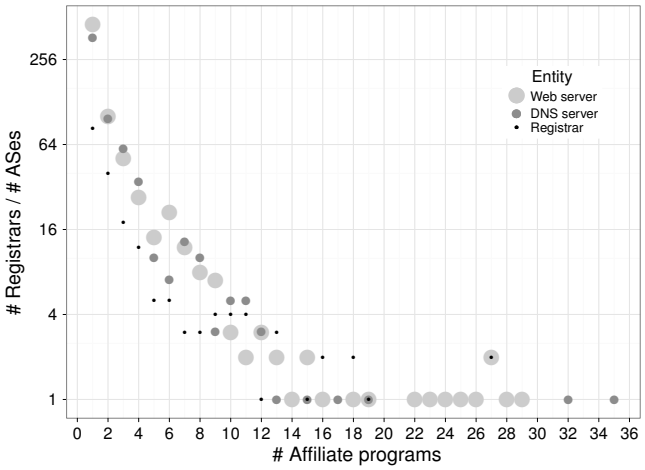


Figure 3: Sharing of network infrastructure among affiliate programs. Only a small number of registrars host domains for many affiliate programs, and similarly only a small number of ASes host name and Web servers for many programs. (Note y-axis is log scale.)

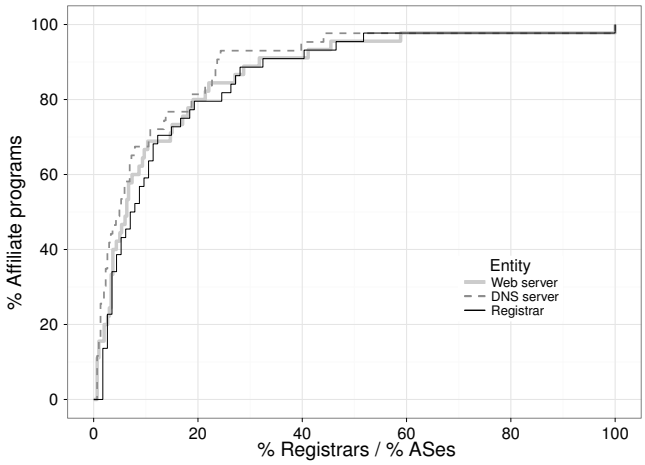


Figure 4: Distribution of infrastructure among affiliate programs. Only a small percentage of programs distribute their registered domain, name server, and Web server infrastructure among many registrars and ASes, respectively.

To explain further, 32% of crawled URLs in our data redirected at least once and of such URLs, roughly 6% did so through public URL shorteners (e.g., `bit.ly`), 9% through well-known "free hosting" services (e.g., `angelfire.com`), and 40% were to a URL ending in `.html` (typically indicating a redirect page installed on a compromised Web server).[12] Of the remainder, the other common pattern is the use of low-quality "throw away" domains, the idea being to advertise a new set of domains, typically registered using random letters or combinations of words, whenever the previous set's traffic-drawing potential is reduced due to blacklisting [24].

Given this, we choose to focus entirely on the final domains precisely because these represent the more valuable infrastructure most clearly operated by an affiliate.

Returning to our key question, we next examine the set of resources used by sites for each affiliate program. In particular, we consider this data in terms of the service organization who is responsible for the resource and how many affiliate programs make use of their service.

Network infrastructure sharing: A spam-advertised site typically has a domain name that must be resolved to access the site.[13] This name must in turn be allocated via a registrar, who has the authority to shutdown or even take back a domain in the event of abuse [30]. In addition, to resolve and access each site, spammers must also provision servers to provide DNS and Web services. These servers receive network access from individual ISPs who have the authority to disconnect clients who violate terms of service policies or in response to complaints.

[12]In our data, we identified over 130 shortener services in use, over 160 free hosting services and over 8,000 likely-compromised Web servers.

[13]Fewer than half a percent use raw IP addresses in our study.

Figure 3 shows that network infrastructure sharing among affiliate programs—when it occurs—is concentrated in a small number of registrars and Autonomous Systems (ASes).[14] Many registrars and ASes host infrastructure for just one or two affiliate programs, only a small number host infrastructure for many affiliate programs, and no single registrar or AS hosts infrastructure for a substantial fraction of the programs overall. (As we will see in Section IV-C however, this situation can change drastically when we weight by the volume of spam advertising each domain.) Specifically, Figure 3 shows the number of registrars (y-axis) that serve registered domains for a given number of affiliate programs (x-axis). Over 80 registrars, for instance, serve domains for a single affiliate program, while just two registrars (NauNet and China Springboard) serve domains for over 20 programs. For name servers and Web servers, it shows the number of ASes hosting servers for a given number of affiliate programs. Over 350 and 450 ASes host DNS and Web servers, respectively, for a single affiliate program; yet, just two and nine ASes host DNS and Web servers, respectively, for over 20 programs (including Hanaro Telecom, China Communication, and ChinaNet).

Although most registrars and ASes host infrastructure for just one affiliate program, each program could still engage many such registrars to serve their domains and many such ASes to host their DNS and Web servers. Figure 4 shows, though, that programs do not in general distribute their infrastructure across a large set of registrars or ASes: for most programs, each of them uses only a small fraction of registrars and ASes found in our data set. Specifically, Figure 4 shows the cumulative distribution of the fraction of registrars and ASes in our data set used by affiliate

[14]We use the AS number as a proxy for ISP.

Bank Name	*BIN*	*Country*	*Affiliate Programs*
Azerigazbank	404610	Azerbaijan	GlvMd, RxPrm, PhEx, Stmul, RxPnr, WldPh
B&N	425175	Russia	ASR
B&S Card Service	490763	Germany	MaxGm
Borgun Hf	423262	Iceland	Trust
Canadian Imperial Bank of Commerce	452551	Canada	WldPh
Cartu Bank	478765	Georgia	DrgRev
DnB Nord (Pirma)	492175	Latvia	Eva, OLPh, USHC
Latvia Savings	490849	Latvia	EuSft, OEM, WchSh, Royal, SftSl
Latvijas Pasta Banka	489431	Latvia	SftSl
St. Kitts & Nevis Anguilla National Bank	427852	St. Kitts & Nevis	DmdRp, VgREX, Dstn, Luxry, SwsRp, OneRp
State Bank of Mauritius	474140	Mauritius	DrgRev
Visa Iceland	450744	Iceland	Staln
Wells Fargo	449215	USA	Green
Wirecard AG	424500	Germany	ClFr

Table V: Merchant banks authorizing or settling transactions for spam-advertised purchases, their Visa-assigned Bank Identification Number (BIN), their location, and the abbreviation used in Table IV for affiliate program and/or store brand.

programs. For 50% of the affiliate programs, their domains, name servers, and Web servers are distributed over just 8% or fewer of the registrars and ASes, respectively; and 80% of the affiliate programs have their infrastructure distributed over 20% or fewer of the registrars and ASes. Only a handful of programs, such as EvaPharmacy, Pharmacy Express, and RX Partners, have infrastructure distributed over a large percentage (50% or more) of registrars and ASes.

To summarize, there are a broad range of registrars and ISPs who are used to support spam-advertised sites, but there is only limited amounts of organized sharing and different programs appear to use different subsets of available resource providers.[15]

B. Realization

Next, we consider several aspects of the realization pipeline, including post-order communication, authorization and settlement of credit card transactions, and order fulfillment.

We first examined the hypothesis that realization infrastructure is the province of *affiliate programs* and not individual affiliates. Thus, we expect to see consistency in payment processing and fulfillment between different instances of the same affiliate program or store brand. Indeed, we found only two exceptions to this pattern and purchases from different sites appearing to represent the same affiliate program indeed make use of the same merchant bank and same pharmaceutical drop shipper.[16] Moreover, key customer support features including the email templates and order number formats are consistent across brands belonging to the same program. This allowed us to further confirm our understanding that a range of otherwise distinct brands all belong to the same underlying affiliate program, including most of the replica brands: Ultimate Replica, Diamond Replicas, Distinction Replica, Luxury Replica, One Replica, Exquisite Replicas, Prestige Replicas, Aff. Accessories; most of the herbal brands: MaxGentleman, ManXtenz, Viagrow, Dr. Maxman, Stud Extreme, VigREX; and the pharmacy: US HealthCare.[17]

Having found strong evidence supporting the dominance of affiliate programs over free actors, we now turn to the question how much realization infrastructure is being shared *across* programs.

Payment: The sharing of payment infrastructure is substantial. Table V documents that, of the 76 purchases for which we received transaction information, there were only 13 distinct banks acting as Visa acquirers. Moreover, there is a significant concentration even among this small set of banks. In particular, most herbal and replica purchases cleared through the same bank in St. Kitts (a by-product of ZedCash's dominance of this market, as per the previous discussion), while most pharmaceutical affiliate programs used two banks (in Azerbaijan and Latvia), and software was handled entirely by two banks (in Latvia and Russia).

Each payment transaction also includes a standardized "Merchant Category Code" (MCC) indicating the type of goods or services being offered [52]. Interestingly, most affiliate program transactions appear to be coded correctly.

[15] We did find *some* evidence of clear inter-program sharing in the form of several large groups of DNS servers willing to authoritatively resolve collections of EvaPharmacy, Mailien and OEM Soft Store domains for which they were outside the DNS hierarchy (i.e., the name servers were never referred by the TLD). This overlap could reflect a particular affiliate advertising for multiple distinct programs and sharing resources internally or it could represent a shared service provider used by distinct affiliates.

[16] In each of the exceptions, at least one order cleared through a different bank—perhaps because the affiliate program is interleaving payments across different banks, or (less likely) because the store "brand" has been stolen, although we are aware of such instances.

[17] This program, currently called ZedCash, is only open by invitation and we had little visibility into its internal workings for this paper.

Supplier	*Item*	*Origin*	*Affiliate Programs*
Aracoma Drug	Orange bottle of tablets (pharma)	WV, USA	ClFr
Combitic Global Caplet Pvt. Ltd.	Blister-packed tablets (pharma)	Delhi, India	GlvMd
M.K. Choudhary	Blister-packed tablets (pharma)	Thane, India	OLPh
PPW	Blister-packed tablets (pharma)	Chennai, India	PhEx, Stmul, Trust, ClFr
K. Sekar	Blister-packed tablets (pharma)	Villupuram, India	WldPh
Rhine Inc.	Blister-packed tablets (pharma)	Thane, India	RxPrm, DrgRev
Supreme Suppliers	Blister-packed tablets (pharma)	Mumbai, India	Eva
Chen Hua	Small white plastic bottles (herbal)	Jiangmen, China	Stud
Etech Media Ltd	Novelty-sized supplement (herbal)	Christchurch, NZ	Staln
Herbal Health Fulfillment Warehouse	White plastic bottle (herbal)	MA, USA	Eva
MK Sales	White plastic bottle (herbal)	WA, USA	GlvMd
Riverton, Utah shipper	White plastic bottle (herbal)	UT, USA	DrMax, Grow
Guo Zhonglei	Foam-wrapped replica watch	Baoding, China	Dstn, UltRp

Table VI: List of product suppliers and associated affiliate programs and/or store brands.

For example, all of our software purchases (across all programs) were coded as 5734 (*Computer Software Stores*) and 85% of all pharmacy purchases (again across programs) were coded as 5912 (*Drug Stores and Pharmacies*). ZedCash transactions (replica and herbal) are an exception, being somewhat deceptive, and each was coded as 5969 (*Direct Marketing—Other*). The few other exceptions are either minor transpositions (e.g., 5921 instead of 5912), singleton instances in which a minor program uses a generic code (e.g., 5999, 8999) with a bank that we only observed in one transaction, and finally Greenline which is the sole pharmaceutical affiliate program that cleared transactions through a US Bank during our study (completely miscoded as 5732, *Electronic Sales*, across multiple purchases). The latter two cases suggest that some minor programs with less reliable payment relationships do try to hide the nature of their transactions, but generally speaking, category coding is correct. A key reason for this may be the substantial fines imposed by Visa on acquirers when miscoded merchant accounts are discovered "laundering" high-risk goods.

Finally, for two of the largest pharmacy programs, GlavMed and RX–Promotion, we also purchased from "canonical" instances of their sites advertised on their online support forums. We verified that they use the same bank, order number format, and email template as the spam-advertised instances. This evidence undermines the claim, made by some programs, that spammers have stolen their templates and they do not allow spam-based advertising.

Fulfillment: Fulfillment for physical goods was sourced from 13 different suppliers (as determined by declared shipper and packaging), of which eight were again seen more than once (see Table VI). All pharmaceutical tablets shipped from India, except for one shipped from within the United States (from a minor program), while replicas shipped universally from China. While we received herbal supplement products from China and New Zealand, most (by volume) shipped from within the United States. This result is consistent with our expectation since, unlike the other goods, herbal products have weaker regulatory oversight and are less likely to counterfeit existing brands and trademarks. For pharmaceuticals, the style of blister packs, pill shapes, and lot numbers were all exclusive to an individual nominal sender and all lot numbers from each nominal sender were identical. Overall, we find that only modest levels of supplier sharing between pharmaceutical programs (e.g., Pharmacy Express, Stimul-cash, and Club-first all sourced a particular product from PPW in Chennai, while RX–Promotion and DrugRevenue both sourced the same drug from Rhine Inc. in Thane). This analysis is limited since we only ordered a small number of distinct products and we know (anecdotally) that pharmaceutical programs use a network of suppliers to cover different portions of their formulary.

We did not receive enough replicas to make a convincing analysis, but all ZedCash-originated replicas were low-quality and appear to be of identical origin. Finally, purchased software instances were bit-for-bit identical between sites of the same store brand and distinct across different affiliate programs (we found no malware in any of these images). In general, we did not identify any particularly clear bottleneck in fulfillment and we surmise that suppliers are likely to be plentiful.

C. Intervention analysis

Finally, we now reconsider these different resources in the spam monetization pipeline, but this time explicitly from the standpoint of the defender. In particular, for any given registered domain used in spam, the defender may choose to intervene by either blocking its advertising (e.g., filtering spam), disrupting its click support (e.g., takedowns for name servers of hosting sites), or interfering with the realization step (e.g., shutting down merchant accounts).[18] But which of these interventions will have the most *impact*?

[18]In each case, it is typically possible to employ either a "takedown" approach (removing the resource comprehensively) or cheaper "blacklisting" approach at more limited scope (disallowing access to the resource for a subset of users), but for simplicity we model the interventions in the takedown style.

Ideally, we believe that such anti-spam interventions need to be evaluated in terms of two factors: their overhead to implement and their business impact on the spam value chain. In turn, this business impact is the sum of both the replacement cost (to acquire new resources equivalent to the ones disrupted) and the opportunity cost (revenue forgone while the resource is being replaced). While, at this point in time, we are unable to precisely quantify all of these values, we believe our data illustrates gross differences in scale that are likely to dominate any remaining factors.

To reason about the effects of these interventions, we consider the registered domains for the affiliate programs and storefront brands in our study and calculate their relative volume in our spam feeds (we particularly subtract the botnet feeds when doing this calculation as their inherent bias would skew the calculation in favor of certain programs). We then calculate the fraction of these domain trajectories that could be completely blocked (if only temporarily) through a given level of intervention at several resource tiers:

Registrar. Here we examine the effect if individual registrars were to suspend their domains which are known to be used in advertising or hosting the sites in our study.

Hosting. We use the same analysis, but instead look at the number of distinct ASs that would need to be contacted (who would then need to agree to shut down all associated hosts in their address space) in order to interrupt a given volume of spam domain trajectories. We consider both name server and Web hosting, but in each case there may be multiple IP addresses recorded providing service for the domain. We adopt a "worst case" model that *all* such resources must be eliminated (i.e., every IP seen hosting a particular domain) for that domain's trajectory to be disrupted.

Payments. Here we use the same approach but focused on the role played by the acquiring banks for each program. We have not placed purchases via each domain, so we make the simplifying assumption that bank use will be consistent across domains belonging to the same brand or affiliate program. Indeed this is strongly borne out in our measurements. For the two small exceptions identified earlier, we assign banks proportionally to our measurements.

Figure 5 plots this data as CDFs of the spam volume in our feeds that would be disrupted using these approaches. For both registrars and hosters there are significant concentrations among the top few providers and thus takedowns would seem to be an effective strategy. For example, almost 40% of spam-advertised domains in our feeds were registered by NauNet, while a single Romanian provider, Evolva Telecom, hosts almost 9% of name servers for spam-advertised domains and over 10% of the Web servers hosting their content; in turn, over 60% of these had payments handled via a single acquirer, Azerigazbank.

However, these numbers do not tell the entire story. Another key issue is the availability of alternatives and their switching cost.

For example, while only a small number of individual IP addresses were *used* to support spam-advertised sites, the supply of hosting resources is vast, with thousands of hosting providers and millions of compromised hosts.[19] The switching cost is also low and new hosts can be provisioned on demand and for low cost.[20]

By contrast, the situation with registrars appears more promising. The supply of registrars is fewer (roughly 900 gTLD registrars are accredited by ICANN as of this writing) and there is evidence that not all registrars are equally permissive of spam-based advertising [28]. Moreover, there have also been individual successful efforts to address malicious use of domain names, both by registries (e.g., CNNIC) and when working with individual registrars (e.g., eNom [25]). Unfortunately, these efforts have been slow, ongoing, and fraught with politics since they require global cooperation to be effective (only individual registrars or registries can take these actions). Indeed, in recent work we have empirically evaluated the efficacy of *past* registrar-level interventions and found that spammers show great agility in working around such actions [29]. Ultimately, the low cost of a domain name (many can be had for under $1 in bulk) and ease of switching registrars makes such interventions difficult.

Finally, it is the banking component of the spam value chain that is both the least studied and, we believe, the most critical. Without an effective mechanism to transfer consumer payments, it would be difficult to finance the rest of the spam ecosystem. Moreover, there are only two networks—Visa and Mastercard—that have the consumer footprint in Western countries to reach spam's principal customers. While there are thousands of banks, the number who are willing to knowingly process what the industry calls "high-risk" transactions is far smaller. This situation is dramatically reflected in Figure 5, which shows that just three banks provide the payment servicing for over 95% of the spam-advertised goods in our study.

More importantly, the replacement cost for new banks is high, both in setup fees and more importantly in time and overhead. Acquiring a legitimate merchant account directly with a bank requires coordination with the bank, with the card association, with a payment processor and typically involves a great deal of due diligence and delay (several days

[19]Note, spam hosting statistics can be heavily impacted by the differences in spam volume produced by different affiliates/spammers. For example, while we find that over 80% of all spam received in this study leads to sites hosted by just 100 distinct IP addresses, there are another 2336 addresses used to host the remaining 20% of spam-advertised sites, many belonging to the same affiliate programs but advertising with lower volumes of spam email.

[20]The cost of compromised proxies is driven by the market price for compromised hosts via Pay-Per-Install enterprises, which today are roughly $200/1000 for Western hosts and $5–10/1000 for Asian hosts [49]. Dedicated bulletproof hosting is more expensive, but we have seen prices as low as $30/month for virtual hosting (up to several hundred dollars for dedicated hosting).

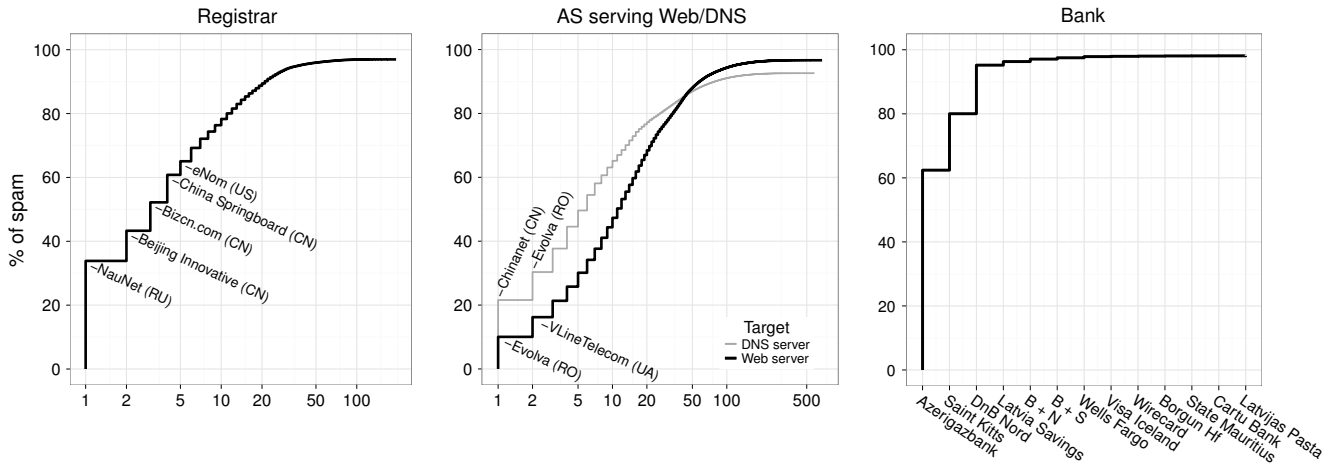


Figure 5: Takedown effectiveness when considering domain registrars (left), DNS and Web hosters (center) and acquiring banks (right).

or weeks). Even for so-called third-party accounts (whereby a payment processor acts as middleman and "fronts" for the merchant with both the bank and Visa/Mastercard) we have been unable to locate providers willing to provide operating accounts in less than five days, and such providers have significant account "holdbacks" that they reclaim when there are problems.[21] Thus, unlike the other resources in the spam value chain, we believe payment infrastructure has far fewer alternatives and far higher switching cost.

Indeed, our subsequent measurements bear this out. For four months after our study we continued to place orders through the major affiliate programs. Many continued to use the same banks four months later (e.g., all replica and herbal products sold through ZedCash, all pharmaceuticals from Online Pharmacy and all software from Auth. Soft. Resellers). Moreover, while many programs did change (typically in January or February 2011), they still stayed within same set of banks we identified earlier. For example, transactions with EvaPharmacy, Greenline, and OEM Soft Store have started clearing through B&N Bank in Russia, while Royal Software, EuroSoft and Soft Sales, have rotated through two different Latvian Banks and B & S Card Service of Germany. Indeed, the *only* new bank appearing in our follow-on purchases is Bank Standard (a private commercial bank in Azerbaijan, BIN 412939); RX–Promotion, GlavMed, and Mailien (a.k.a. Pharmacy Express) all appear to have moved to this bank (from Azerigazbank) on or around January 25th. Finally, one order placed with DrugRevenue failed due to insufficient funds, and was promptly retried through two different banks (but again, from the same set). This suggests that while cooperating third-party payment processors may be able to route transactions through merchant accounts at difference banks, the set of banks currently available for such activities is quite modest.

[21]To get a sense of the kinds of institutions we examined, consider this advertisement of one typical provider: "We have ready-made shell companies already incorporated, immediately available."

D. Policy options

There are two potential approaches for intervening at the payment tier of the value chain. One is to directly engage the merchant banks and pressure them to stop doing business with such merchants (similar to Legitscript's role with registrars [25], [28]). However, this approach is likely to be slow—very likely slower than the time to acquire new banking facilities. Moreover, due to incongruities in intellectual property protection, it is not even clear that the sale of such goods is illegal in the countries in which such banks are located. Indeed, a sentiment often expressed in the spammer community, which resonates in many such countries, is that the goods they advertise address a real need in the West, and efforts to criminalize their actions are motivated primarily by Western market protectionism.

However, since spam is ultimately supported by Western money, it is perhaps more feasible to address this problem in the West as well. To wit, if U.S. issuing banks (i.e., banks that provide credit cards to U.S. consumers) were to refuse to *settle* certain transactions (e.g., card-not-present transactions for a subset of Merchant Category Codes) with the banks identified as supporting spam-advertised goods, then the underlying enterprise would be dramatically demonetized. Furthermore, it appears plausible that such a "financial blacklist" could be updated very quickly (driven by modest numbers of undercover buys, as in our study) and far more rapidly than the turn-around time to acquire new banking resources—a rare asymmetry favoring the anti-spam community. Furthermore, for a subset of spam-advertised goods (regulated pharmaceuticals, brand replica products, and pirated software) there is a legal basis for enforcing such a policy.[22] While we suspect that the political challenges for

[22]Herbal products, being largely unregulated, are a more complex issue.

such an intervention would be significant—and indeed merit thoughtful consideration—we note that a quite similar action has already occurred in restricting U.S. issuers from settling certain kinds of online gambling transactions [11].

V. Conclusion

In this paper we have described a large-scale empirical study to measure the spam value chain in an end-to-end fashion. We have described a framework for conceptualizing resource requirements for spam monetization and, using this model, we have characterized the use of key infrastructure—registrars, hosting and payment—for a wide array of spam-advertised business interests. Finally, we have used this data to provide a normative analysis of spam intervention approaches and to offer evidence that the payment tier is by far the most concentrated and valuable asset in the spam ecosystem, and one for which there may be a truly effective intervention through public policy action in Western countries.

Acknowledgments

This is, again, the most ambitious measurement effort our team has attempted and even with 15 authors it would have been impossible without help from many other individuals and organizations. First and foremost, we are indebted to our spam data providers: Jose Nazario, Chris Morrow, Barracuda Networks, Abusix and a range of other partners who wish to remain anonymous. Similarly, we received operational help, wisdom and guidance from Joe Stewart, Kevin Fall, Steve Wernikoff, Doug McKenney, Jeff Williams, Eliot Gillum, Hersh Dangayach, Jef Pozkanzer, Gabe Lawrence, Neils Provos, Kevin Fu and Ben Ransford among a long list of others. On the technical side of the study, we thank Jon Whiteaker for an early implementation of the DNS crawler and Brian Kantor for supporting our ever expanding needs for cycles, storage and bandwidth. On the purchasing side of the study, we are deeply indebted to the strong support of our card issuer and their staff. On the oversight side, we are grateful to Erin Kenneally and Aaron Burstein for their legal guidance and ethical oversight, to our Chief Counsel at UCSD, Daniel Park, and UC's Systemwide Research Compliance Director, Patrick Schlesinger, for their open-mindedness and creativity, and finally to Marianne Generales and Art Ellis representing UCSD's Office of Research Affairs for helping to connect all the dots.

This work was supported in part by National Science Foundation grants NSF-0433668, NSF-0433702, NSF-0831138 and CNS-0905631, by the Office of Naval Research MURI grant N000140911081, and by generous research, operational and/or in-kind support from Google, Microsoft, Yahoo, Cisco, HP and the UCSD Center for Networked Systems (CNS). Félegyházi contributed while working as a researcher at ICSI. McCoy was supported by a CCC-CRA-NSF Computing Innovation Fellowship.

References

[1] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: Characterizing Internet Scam Hosting Infrastructure. In *Proc. of 16th USENIX Security*, 2007.

[2] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An Evaluation of Naive Bayesian Anti-Spam Filtering. In *Proc. of 1st MLNIA*, 2000.

[3] J. Armin, J. McQuaid, and M. Jonkman. Atrivo — Cyber Crime USA. http://fserror.com/pdf/Atrivo.pdf, 2008.

[4] Behind Online Pharma. From Mumbai to Riga to New York: Our Investigative Class Follows the Trail of Illegal Pharma. http://behindonlinepharma.com, 2009.

[5] C. Castelluccia, M. A. Kaafar, P. Manils, and D. Perito. Geolocalization of Proxied Services and its Application to Fast-Flux Hidden Servers. In *Proc. of 9th IMC*, 2009.

[6] R. Clayton. How much did shutting down McColo help? In *Proc. of 6th CEAS*, 2009.

[7] Dancho Danchev's Blog — Mind Streams of Information Security Knowledge. The Avalanche Botnet and the TROYAK-AS Connection. http://ddanchev.blogspot.com/2010/05/avalanche-botnet-and-troyak-as.html, 2010.

[8] Federal Trade Commission. FTC Shuts Down, Freezes Assets of Vast International Spam E-Mail Network. http://ftc.gov/opa/2008/10/herbalkings.shtm, 2008.

[9] W. Feng and E. Kaiser. kaPoW Webmail: Effective Disincentives Against Spam. In *Proc. of 7th CEAS*, 2010.

[10] J. Franklin, V. Paxson, A. Perrig, and S. Savage. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *Proc. of 14th ACM CCS*, 2007.

[11] Gamblingplanet.org. Visa blocks gaming transactions for US players. http://www.gamblingplanet.org/news/Visa-blocks-gaming-transactions-for-US-players/022310, 2010.

[12] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: The Underground on 140 Characters or Less. In *Proc. of 17th ACM CCS*, 2010.

[13] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. In *Proc. of 15th NDSS*, 2008.

[14] S. Hao, N. Feamster, A. Gray, N. Syed, and S. Krasser. Detecting Spammers with SNARE: Spatio-Temporal Network-Level Automated Reputation Engine. In *Proc. of 18th USENIX Security*, 2009.

[15] C. Herley and D. Florencio. Nobody Sells Gold for the Price of Silver: Dishonesty, Uncertainty and the Underground Economy. In *Proc. of 8th WEIS*, 2009.

[16] T. Holz, M. Engelberth, and F. Freiling. Learning More About the Underground Economy: A Case-Study of Keyloggers and Dropzones. In *Proc. of 15th ESORICS*, 2009.

[17] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and Detecting Fast-Flux Service Networks. In *Proc. of 15th NDSS*, 2008.

[18] X. Hu, M. Knysz, and K. G. Shin. RB-Seeker: Auto-detection of Redirection Botnets. In *Proc. of 16th NDSS*, 2009.

[19] D. Irani, S. Webb, J. Giffin, and C. Pu. Evolutionary Study of Phishing. In *eCrime Researchers Summit*, pages 1–10, 2008.

[20] J. P. John, A. Moshchuk, S. D. Gribble, and A. Krishnamurthy. Studying Spamming Botnets Using Botlab. In *Proc. of 6th NSDI*, 2009.

[21] 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 *Proc. of 15th ACM CCS*, 2008.

[22] M. Konte, N. Feamster, and J. Jung. Dynamics of Online Scam Hosting Infrastructure. In *Proc. of 10th PAM*, 2009.

[23] Krebs on Security. Body Armor for Bad Web Sites. http://krebsonsecurity.com/2010/11/body-armor-for-bad-web-sites/, 2010.

[24] 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 *Proc. of 2nd USENIX LEET*, 2009.

[25] LegitScript and eNom. LegitScript Welcomes Agreement with eNom (DemandMedia). http://www.legitscript.com/blog/142, 2010.

[26] LegitScript and KnujOn. No Prescription Required: Bing.com Prescription Drug Ads. http://www.legitscript.com/download/BingRxReport.pdf, 2009.

[27] LegitScript and KnujOn. Yahoo! Internet Pharmacy Advertisements. http://www.legitscript.com/download/YahooRxAnalysis.pdf, 2009.

[28] LegitScript and KnujOn. Rogues and Registrars: Are some Domain Name Registrars safe havens for Internet drug rings? http://www.legitscript.com/download/Rogues-and-Registrars-Report.pdf, 2010.

[29] H. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage. On the Effects of Registrar-level Intervention. In *Proc. of 4th USENIX LEET*, 2011.

[30] B. Livingston. Web registrars may take back your domain name. http://news.cnet.com/2010-1071-281311.html, 2000.

[31] M86 Security Labs. Top Spam Affiliate Programs. http://www.m86security.com/labs/traceitem.asp?article=1070, 2009.

[32] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying Suspicious URLs: An Application of Large-Scale Online Learning. In *Proc. of 26th ICML*, 2009.

[33] B. S. McWilliams. *Spam Kings: The Real Story Behind the High-Rolling Hucksters Pushing Porn, Pills and @*#?% Enlargements*. O'Reilly Media, Sept. 2004.

[34] D. Molnar, S. Egelman, and N. Christin. This Is Your Data on Drugs: Lessons Computer Security Can Learn From The Drug War. In *Proc. of 13th NSPW*, 2010.

[35] T. Moore and R. Clayton. The Impact of Incentives on Notice and Take-down. In *Proc. of 7th WEIS*, 2008.

[36] T. Moore, R. Clayton, and H. Stern. Temporal Correlations between Spam and Phishing Websites. In *Proc. of 2nd USENIX LEET*, 2009.

[37] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy, G. M. Voelker, and S. Savage. Re: CAPTCHAs — Understanding CAPTCHA Solving from an Economic Context. In *Proc. of 19th USENIX Security*, 2010.

[38] A. Mutton. Screengrab! http://www.screengrab.org/, 2010.

[39] Y. Niu, Y.-M. Wang, H. Chen, M. Ma, and F. Hsu. A Quantitative Study of Forum Spamming Using Context-based Analysis. In *Proc. of 14th NDSS*, 2007.

[40] C. Nunnery, G. Sinclair, and B. B. Kang. Tumbling Down the Rabbit Hole: Exploring the Idiosyncrasies of Botmaster Systems in a Multi-Tier Botnet Infrastructure. In *Proc. of 3rd USENIX LEET*, 2010.

[41] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi. FluXOR: Detecting and Monitoring Fast-Flux Service Networks. In *Proc. of 5th DIMVA*, 2008.

[42] R. Perdisci, I. Corona, D. Dagon, and W. Lee. Detecting Malicious Flux Service Networks through Passive Analysis of Recursive DNS Traces. In *Proc. of 25th ACSAC*, 2009.

[43] A. Pitsillidis, K. Levchenko, C. Kreibich, C. Kanich, G. Voelkera, V. Paxson, N. Weaver, and S. Savage. Botnet Judo: Fighting Spam with Itself. In *Proc. of 17th NDSS*, 2010.

[44] Z. Qian, Z. M. Mao, Y. Xie, and F. Yu. On Network-level Clusters for Spam Detection. In *Proc. of 17th NDSS*, 2010.

[45] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proc. of ACM SIGCOMM*, 2006.

[46] D. Samosseiko. The Partnerka — What is it, and why should you care? In *Proc. of Virus Bulletin Conference*, 2009.

[47] S. Sinha, M. Bailey, and F. Jahanian. Shades of Grey: On the effectiveness of reputation-based "blacklists". In *Proc. of 3rd MALWARE*, 2008.

[48] S. Sinha, M. Bailey, and F. Jahanian. Improving SPAM Blacklisting through Dynamic Thresholding and Speculative Aggregation. In *Proc. of 17th NDSS*, 2010.

[49] K. Stevens. The Underground Economy of the Pay-Per-Install (PPI) Business. http://www.secureworks.com/research/threats/ppi, 2009.

[50] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your Botnet Is My Botnet: Analysis of a Botnet Takeover. In *Proc. of 16th ACM CCS*, 2009.

[51] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda. FIRE: FInding Rogue nEtworks. In *Proc. of 25th ACSAC*, 2009.

[52] Visa Commercial Solutions. Merchant Category Codes for IRS Form 1099-MISC Reporting. http://usa.visa.com/download/corporate/resources/mcc_booklet.pdf.

[53] Y.-M. Wang, M. Ma, Y. Niu, and H. Chen. Spam Double-Funnel: Connecting Web Spammers with Advertisers. In *Proc. of 16th WWW*, 2007.

[54] G. Warner. Random Pseudo-URLs Try to Confuse Anti-Spam Solutions. http://garwarner.blogspot.com/2010/09/random-pseudo-urls-try-to-confuse-anti.html, Sept. 2010.

[55] C. Whittaker, B. Ryner, and M. Nazif. Large-Scale Automatic Classification of Phishing Pages. In *Proc. of 17th NDSS*, 2010.

[56] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming Botnets: Signatures and Characteristics. In *Proc. of ACM SIGCOMM*, 2008.

[57] L. Zhang, J. Zhu, and T. Yao. An Evaluation of Statistical Spam Filtering Techniques. *ACM Trans. on ALIP*, 3(4), 2004.

[58] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, and E. Gillum. BotGraph: Large-Scale Spamming Botnet Detection. In *Proc. of 6th NSDI*, 2009.

[59] J. Zhuge, T. Holz, C. Song, J. Guo, X. Han, and W. Zou. Studying Malicious Websites and the Underground Economy on the Chinese Web. In *Proc. of 7th WEIS*, 2008.

Spamcraft: An Inside Look At Spam Campaign Orchestration

Christian Kreibich[†] Chris Kanich[*] Kirill Levchenko[*] Brandon Enright[*]
Geoffrey M. Voelker[*] Vern Paxson[†‡] Stefan Savage[*]

[†]International Computer Science Institute
[‡]University of California, Berkeley, USA

[*]Dept. of Computer Science and Engineering
University of California, San Diego, USA

1 Introduction

Over the last decade, unsolicited bulk email—spam—has evolved dramatically in its volume, its delivery infrastructure and its content. Multiple reports indicate that more than 90% of all email traversing the Internet today is considered spam. This growth is partially driven by a multi-billion dollar anti-spam industry whose dedication to filtering spam in turn requires spammers to recruit botnets to send ever greater volumes to maintain profits. While we all bear witness to this evolution via the contents of our inboxes, far less is understood about the spammer's viewpoint. In particular, for each spam *campaign*, spammers must gather and target a particular set of recipients, construct enticing message content, ensure sufficient IP address diversity to evade blacklists, and maintain sufficient content diversity to evade spam filters.

In this paper we present an inside look at how such *campaign orchestration* takes place. Over a period of ten months, we have infiltrated the spamming campaigns hosted on a large-scale spamming platform: the Storm botnet. Our analysis is two-pronged. First, instead of focusing on particular corpora of spam, we analyze the raw material used to *produce* spam, including textual *templates* employed for generating highly diverse spam instances. We identify over 90 different campaign types hosted on the Storm platform during the timeframe of our investigation, targeting over 630 million different email addresses and harnessing well over 90,000 different spamming zombies. We classify individual campaigns by topic and time, and study the evasive maneuvers employed by the spammers to stay ahead of filtering infrastructure. Second, we study the spammer's campaign targeting strategies, including usage patterns of "spamvertized" domains, harvested email addresses, target group selection, and target list maintenance.

Our findings indicate a wide range in campaign duration, evasive sophistication, and user targeting – even within a single botnet.

2 Background

The study we perform in this paper continues the line of efforts infiltrating real-world botnets [8, 10, 11, 14, 16] and directly follows from previous work we have performed on the mechanisms the Storm botnet uses to support spam campaigns [9] and to measure spam conversion rates [7]. Whereas the previous work introduced our infiltration methodology we use in this study, it focused on documenting Storm's mechanisms for spam delivery and, when interposing on the command and control (C&C) channel, modifying the commands sent *downward* in the hierarchy. We extend that work by modifying C&C flow *upward*: we inject target addresses into email address harvests gathered from infected machines, and present a more comprehensive analysis of the spam campaigns themselves over a longer period of time.

Spammers need to collect email addresses for targeting spam. Many people are aware of the fact that spammers harvest target email addresses from Web pages, forums, wikis, etc [12]. These lists are valuable, as evidenced by their popularity on the Internet underground market [4].

Spam corpora have been used for a variety of studies, including the spam relay infrastructure used to deliver spam [15], scam hosting infrastructure used to host sites advertised in spam [3], characterization of botnets by the spam they send [17], effectiveness and dominance of content-based spam filtering tests over time [13], and the impact and susceptibility of stock scam campaigns on financial markets [5, 6].

3 The Storm Botnet

Our measurements are driven by a combination of probing and infiltration of the Storm botnet. This network appeared in 2006 and by 2007 had grown to be one of the dominant spamming platforms. By mid-2008 its size had dwindled, and on 21 September 2008 it fell silent when its hosted infrastructure was taken off-line.

We next review Storm's technical operation for required context, and refer the reader to the related work

for additional details. The Storm botnet propagates via spam, relying on gullible users to download and execute its binaries. Once running, Storm employs an encrypted version of the UDP-based Overnet protocol to locate *proxy bots*, to which it then issues work *requests* via its custom TCP-based command and control (C&C) protocol. Proxy bots are themselves infected PCs but, in contrast to *worker bots*, they are world-reachable. Proxy bots serve as a conduit to the third tier, *HTTP proxies*, which are hosted under control of the botnet operators at hosting services. The result is a four-tiered architecture of worker bots, proxy bots, HTTP proxies, and the botmaster.

Worker bots acquire new spamming instructions in a pull-based fashion. They issue requests for spam material, which are answered by *update messages*. These messages consist of three independent parts, each of which may be empty: (i) template material defining the raw format from which to construct spam messages; (ii) sets of *dictionaries* containing text material to substitute into templates, thereby instantiating spam messages; and (iii) lists of target email addresses. These lists typically provide roughly 1,000 addresses per update message. Templates and target lists are associated via a numbering scheme, which we call *slots*.

A single spam instance, pseudo-randomly composed from the dictionary material and additional template language macros, is sent to each address given in an update message. Storm's templating language consists of over 25 different formatting macros for text insertion and formatting, pseudo-random number generation, computation of MTA message identifiers, dates, and reuse of previous macro invocations and text blocks. Macros are delineated by a start marker "`%^`" and a corresponding end marker "`^%`". A single letter after the initial marker identifies the macro's functionality. It is followed by zero or more macro input arguments, which may consist of the output of nested macros. We refer the reader to our earlier work for a comprehensive list of macros [9].

Once spamming completes, workers send *delivery reports*, listing addresses to which spam was delivered successfully, and error codes for failed deliveries.

Worker bots also search for email addresses on the hard drive of the compromised computer and send these up to the botmaster, an activity called *harvesting*. In fact, bots harvest anything syntactically resembling an email address, that is, any string matching the pattern "*@*.*". We study this operation of the botnet in Section 5.3.

4 Methodology

We operated two separate platforms to conduct the measurements presented in this paper: a *C&C crawler* which tapped into Storm's network to collect update messages

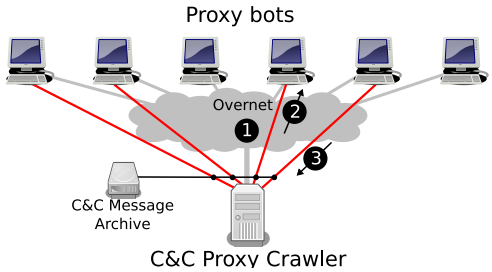


Figure 1: C&C crawler setup used for long-term collection of spam update messages. ❶ Overnet crawler taps into Overnet to find proxy bots. ❷ C&C crawler queries active proxies for update messages. ❸ Proxy bots respond with update messages.

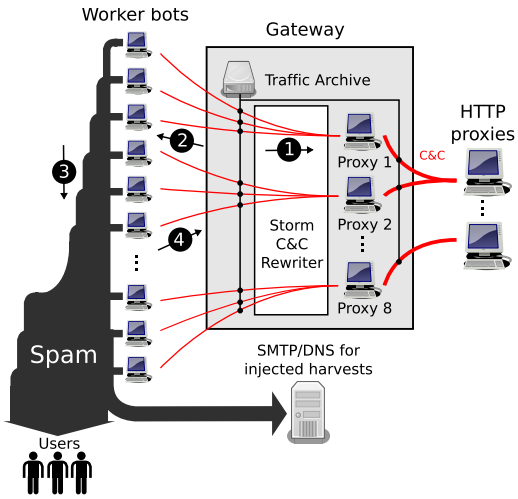


Figure 2: Measurement and rewriting infrastructure for proxy-based C&C traffic. ❶ Workers report harvests, which are optionally rewritten by our interposition setup. ❷ Workers obtain update messages. ❸ Workers start spamming. ❹ Workers summarize spam run in delivery reports.

containing spamming information, and a *C&C rewriting engine* using proxy bots in a controlled environment. We next describe both platforms and summarize the collected datasets in Section 4.3.

4.1 C&C crawler

To collect data on the long-term operation of the Storm botnet, we developed a C&C crawler which requests a spamming workload from an active Storm proxy every 15 seconds. Such a workload consists of spam templates and email target lists for active campaigns, which the C&C crawler stores for later analysis. Figure 1 illustrates the platform.

4.2 C&C rewriter

To observe the activity of real worker bots, we infiltrated the botnet at the proxy bot level. We ran between 8 and 10 proxy bots in a controlled environment of virtual machines hosted on VMware ESX servers. All C&C traffic was recorded to disk for later analysis. Figure 2 shows

our C&C infiltration architecture.

To investigate the use of harvested email addresses by the botmaster, we interposed a C&C rewriting engine into the worker bots' traffic. This interposition enabled us to inject "marker" email addresses of our choosing into the harvests reported by the workers. When performing harvest injections, we injected 3 unique email addresses into each non-empty harvest. We injected them in a format that allowed us to track their subsequent use: [harvest].[worker]`@`[random].[domain].

Here, "harvest" is an identifier for a particular harvest message, "worker" encodes the IP address of the worker bot reporting the harvest, "random" is a random character sequence unique to each injected address, and "domain" is one of a set of second-level domains we acquired for the purpose of the experiment. We operate DNS servers and SMTP sinks for each of these domains. We monitored the target lists seen by the crawler and our proxy for occurrences of the marker addresses.

4.3 Collected datasets

Table 1 summarizes the three data sets we collected for this study. We began operating the crawler on 20 November 2007 and stopped it almost one year later, on 11 November 2008. The crawler was in operation 228 days. We refer to the resulting data as the crawl-based (CB) dataset, which we used to analyze spam campaigns. The proxy platform has been in continuous operation since 9 March 2008 until 6 May 2008. Until 2 April 2008 we passively collected C&C traffic, producing the proxy-based (PB) dataset. From 26 April until 6 May we actively injected email addresses into the harvests reported by the worker bots, producing the harvest injection (HI) dataset. The PB and HI datasets were used to study address harvesting.

4.4 Terminology

The term "spam campaign" is commonly used with varying degrees of generality to mean anything from all spam of a certain type (e.g., pharmaceutical), to spam continuously generated from a single template. In this paper, we talk about campaigns at three levels of abstraction:

- CLASSES of campaigns correspond to the broad intended purpose of spam emails, such as phishing, pharmaceutical offers, or stock scams.
- TYPES of campaigns are sets of spam messages, all of which share a characterizing content element. This element can be verbatim text, or the text resulting from identical templating language constructs. For example, in our dataset all templates containing the string `linksh` define a type of self-propagation campaigns. Each campaign type belongs to a campaign class.

CRAWL-BASED DATASET (CB)		
Timeframe	20 Nov 07 – 11 Nov 08	
Proxies contacted	492,491	(2,794 distinct)
Spam templates	536,607	(23.1% unique)
Targeted email addresses	350,291,617	(59.1% unique)

PROXY-BASED DATASET (PB)		
Timeframe	09 Mar 08 – 02 Apr 08	
Worker bots	94,335	
Update messages	679,976	
Spam templates	813,655	(51.9% unique)
Delivery reports	266,633	
Harvest reports	843,982	(6.6% non-empty)
Targeted email addresses	580,312,064	(43.5% unique)
Harvested email addresses	1,211,971	(44.8% unique)

HARVEST-INJECTION DATASET (HI)		
Timeframe	26 Apr 08 – 06 May 08	
Worker bots	36,037	
Update messages	296,794	
Spam templates	388,310	(12.9% unique)
Delivery reports	101,884	
Harvest reports	1,029,566	(6.3% non-empty)
Harvested email addresses	1,820,360	(50.4% unique)
Targeted email addresses	280,304,900	(60.9% unique)
Markers injected	87,846	
Targeted markers	1,957	(97.8% unique)
Spams delivered to markers	1,017	

Table 1: Summary of datasets used in the study. In the HI dataset, "markers" are email addresses we injected into email harvests, and "targeted markers" are those markers we observed being used as spam delivery addresses in later campaigns.

- INSTANCES of campaigns correspond to campaign types conducted continuously during a period of time. Campaign inactivity for at least 24 hours creates multiple campaign instances (see Section 5.1). For example, one instance of the `linksh` self-propagation campaign type ran from 19:17 on 19 January 2008 to 20:38 on 22 January 2008. Each campaign instance belongs to a particular campaign type.

5 Campaign Analysis

We now present the results of our campaign infiltration. We first summarize elementary properties of the campaigns we observed, then study the evasive tactics employed by the campaign authors to evade filtering, and finally study harvesting and address targeting strategies.

5.1 Conducted campaigns

The nearly year-long span of the CB dataset gives us a full view of the types of campaigns undertaken by the Storm botnet. To identify the individual campaigns we iteratively classified the templates by manually identifying strings characteristic to individual campaign types. For example, templates containing `linksh` or `wormsubj` uniquely identify two self-propagation cam-

CLASS	DESCRIPTION
Image spam	Image-based spam
Job ads	Mule scams, “employee” forwards money/goods
Other ads	Other kinds of advertising
Personal ad	Fake dating/matchmaking advance money scams
Pharma	Pointers to web sites selling Viagra, Cialis, etc
Phishing	Entices victims to enter sensitive information
Political	Political campaigning
Self-prop.	Tricks victims into executing Storm binaries
Stock scam	Tricks victims into buying a particular penny stock
(Other)	Broken/empty templates, noise-only templates, etc

Table 2: Campaign classes encountered in the study.

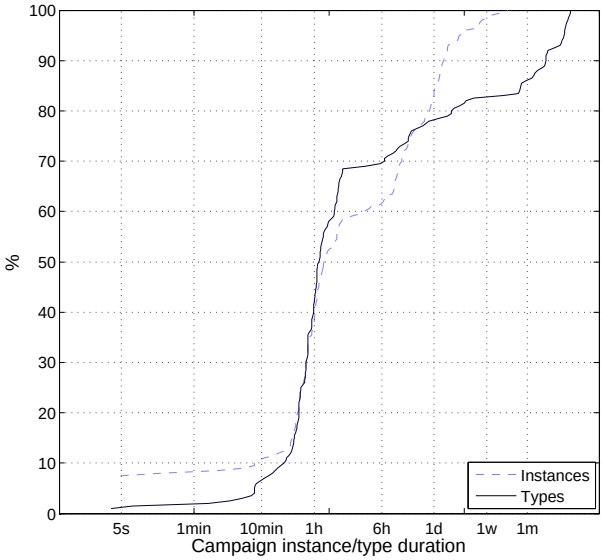


Figure 3: Distribution of campaign instance durations in comparison to campaign type durations.

paign types, while those containing `its-budget` identify a mule campaign. Our examination revealed a rich set of campaigns—94 in all—which we have grouped into ten classes described in Table 2.

We next focus on campaign instance duration. Since individual campaign types may occur repeatedly after long periods of absence, we used a cutoff of 24h to delineate individual instances of the same type. It is hard to find the absolutely correct interval here. If too small, the risk of incorrectly splitting an ongoing campaign instance increases; if too large, we begin to miss individual instances. Based on manual inspection of the resulting campaign instances, we concluded that 24h seems a good compromise, yielding 188 campaign instances. Where monitoring outages occurred for a period of more than 24h, we count campaign types active right before and after the outage as separate instances.

Figure 3 compares durations of campaign types and instances. Instances are often short: 65% of them last less than 2 hours. The longest-running instances are the pharmaceutical ones, running months at a time, and the crucial self-propagation instances which we observed

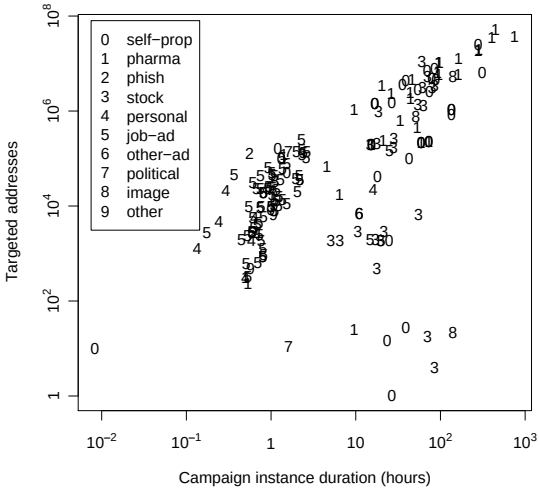


Figure 4: Duration of campaign instances vs. number of email address targets the crawler obtained per instance. The average address retrieval rate (gray line) is 570 addresses per minute. Square markers indicate test campaigns.

running up to 12 days without interruption. Indeed, campaign types are likewise typically short and many campaign types coincide with campaign instances. For some campaign classes, the briefness is inherent, as in stock touting scams. For others (particularly the job ads we observed), we believe the infrastructure behind the campaigns to be substantially less sophisticated than for the long-running pharmaceutical one, as evidenced by templates with fixed domain names which are more easily filtered.

Figure 4 shows for each campaign instance the number of addresses the crawler obtained. The average address retrieval rate is 570 addresses per minute. Nine instances target at least one order of magnitude fewer addresses than the remaining ones; we believe those to be test campaigns, conducted briefly to check filter penetration. The fact that those campaigns use the same slot and that this slot is not otherwise used strengthens the hypothesis (one German stock scam instance, using a seemingly untargeted address list of 463 addresses, looks like a misfire). The ability to identify test campaigns can provide crucial information to law enforcement, since it points out email addresses directly connected to the spammers. Figure 5 summarizes campaign instances, types, and classes observed over time. The short lifetimes of instances in most campaign classes are clearly visible, as is the dominance of job advertisements in the overall set of instances. Stock scams took a four-month break in February 2008, returning in June.

5.2 Evasive maneuvers

Next we characterize the approaches spammers use to create diverse spam messages to evade spam filters.

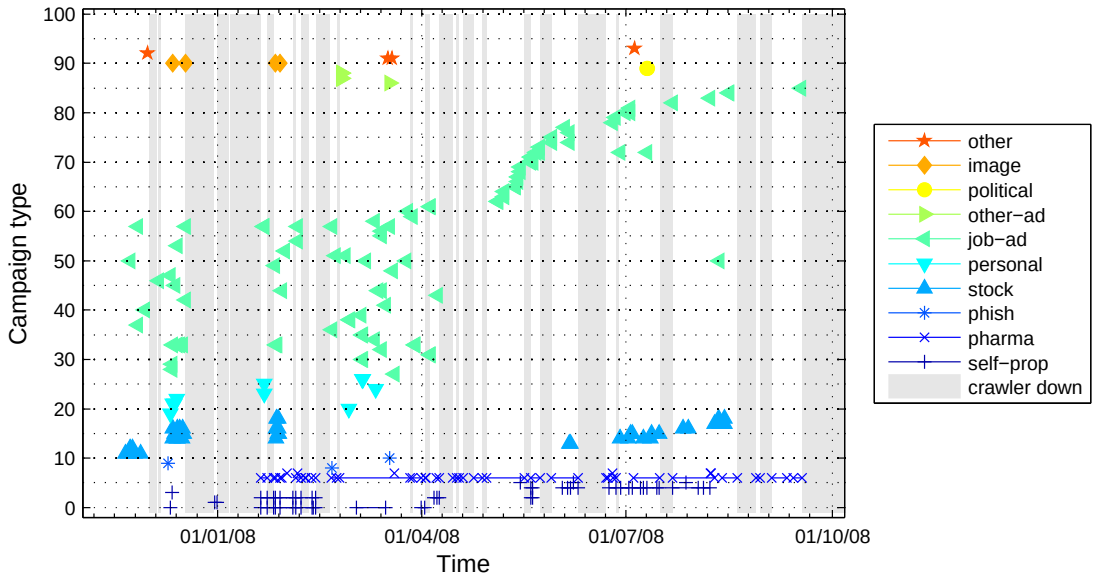


Figure 5: Classes, types, and instances of spamming campaigns identified over time. Shaded areas indicate periods when the crawler was off-line for maintenance or development.

Name	Unique	Total	Redundancy (%)	Average Size	Sample
linksh	72,225	145,201	49.74	97.97	68.37.82.21
names	22,494	374,260	93.99	883.09	a-m.guillerm.acces
domains	21,329	374,116	94.30	1019.60	123glitter.com
mps	20,493	21,094	2.85	39.58	Make the move now
mpb21	20,493	21,093	2.84	16.62	Move fast "buy mpix
words	6,475	6,500	0.38	998.62	obliging
stormlink2	6,130	280,749	97.82	67.82	yourfireworks.com
words_cent	5,835	7,218	19.16	241.95	A part of The New York Times Company.
pharma	107	361,203	99.97	821.95	10 Mistakes All Men Make!
wormsubj	9	86,514	99.99	63.78	Love You

Table 3: Summary of size and uniqueness properties of the five most diverse dictionaries (top) and five select others (bottom).

Dictionaries. The spammers' primary technique for introducing textual diversity into the resulting spam is the use of dictionaries. The template language's F-macro randomly picks an entry from a specific dictionary and inserts it at the location of the macro. We only encountered a single template (a PayPal phish) which did not use any dictionaries. We identified 173 different kinds of dictionaries in the CB dataset. Table 3 summarizes the most diverse ones. 80% of the templates use 10 or less, while the most dictionary-driven template, an image-based stock scam instance, employed 50 (which mostly generated noise via 2 "words" and 40 "words_cent" applications).

Template diversity. Just as dictionaries provide diversity to the spam built from a particular template, so can sets of templates belonging to the same campaign type potentially provide *higher-order* diversity to all spam messages belonging to the campaign. Such diversity certainly seems to hold promise; for example, different kinds of dictionary material could be introduced in rapid succession, or elements of the templates could be adjusted dynamically and coordinated across campaigns.

We investigated this diversity starting from the observation that different parts of templates are of different importance to a campaign. While the body of the resulting messages necessarily needs to convey a particular meaning to human readers of the message, humans will generally not care as much about the details of the email headers. To understand the template diversity at the overall, header, and body levels we counted the number of unique overall templates, headers, and bodies for each campaign type. We excluded the `Subject` header, which frequently relates to the semantic meaning of the body, from the header uniqueness calculation and instead included it in the body's.

Figure 6 compares the distribution of the overall templates, unique templates, their unique headers, and unique bodies in campaign types. Interestingly, while longer-running campaigns do employ more templates, only a fraction of those templates differ. Nearly half of the campaigns employ only a single template. Those that

employ multiple focus the modification on the headers (observe the nearly coinciding lines for unique templates and unique headers), while the body sections change even less frequently.

Table 4 documents length and diversity of the campaign classes. From it, we make the following observations.

First, the image spam campaign is an obvious outlier. Storm neither employed dictionaries to provide the image data, nor did it provide template macros that mutate the image data upon message construction. As a result, recipients received identical images until the template itself was updated to contain a new image. The images were all GIFs without image annotation. All contained stock touting texts.

Second, the three most diverse classes, Pharma, Self-prop, and Stock scams, have a strikingly large number of unique headers. It turned out that the majority of those diversifications merely consist of a large variety of partially hard-coded SMTP `Message-ID` strings designed to look compatible to that of the Sendmail MTA. These identifiers consist of strings such as

```
SMTP id %^Y%^C5%^R20-300^%^%^%002009;
```

which contain a randomized invocation of the Y-macro, used to generate parts of Sendmail-compatible `Message-ID` strings. The only difference among the headers is the numerical suffix of the line. With subsequent templates, the suffix number increases continuously, simulating the effect of a timestamp. This construct accounts for over 99% of unique headers in Pharma, 94% in Self-prop, and 95% in Stock scams.

Third, the long-running Pharma and Self-prop classes used comparatively few different bodies (10 vs. 20, respectively). The differences in those templates reflect changes of dictionaries — for example, to change the malware lure from variants of "Greeting cards for you" to ones for "Happy April Fool's Day" — and thus consist of few changes of campaign types. By contrast, the short Stock tout and Job ad classes use a much larger number of campaign types. Campaign instances here come close to campaign types.

Header diversity. To better understand how the template headers are diversified, we further subdivided the header part into (*i*) the simulated user-agent, (*ii*) the MTA responsible for the `MessageID` header, and (*iii*) the (possibly empty) sequence of `Received-By` headers.

We encountered 11 different header part combinations, largely independent of campaign types. The combination of all-Microsoft MUAs/MTAs was particularly popular, occurring in 51 different campaign types. Two popular MUAs are simulated: Thunderbird and Outlook. The MTA population consists of combinations of Microsoft's SMTPSVC, Sendmail, Exim, and Qmail.

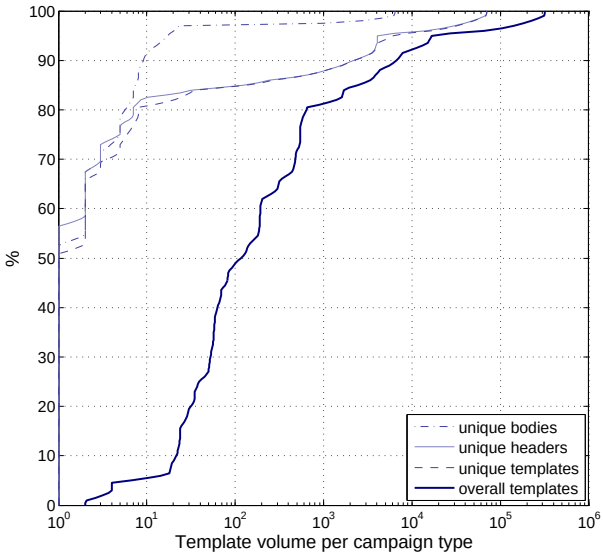


Figure 6: Distribution of overall templates, unique templates, unique headers, and unique bodies, across campaign types.

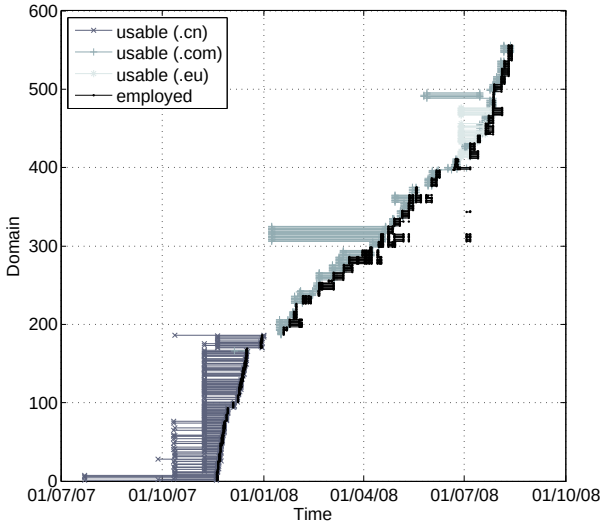


Figure 7: Spamvertized domain usage in the Pharma campaign, from the CB dataset. Black lines indicate timeframes during which the domain was actively used, while lighter colors illustrate the period from a domain's registration until its appearance on the Spam Domain Blacklist. Different colors represent different TLDs.

Domain diversity. A crucial component of many scams is an HTTP link luring customers to the scammer's site. While such links can be provided using IP addresses as well, real domains are commonly employed since they seem more familiar to most users. The presence of such domains is an important vector for blacklisting services (such as jwSpamSpy [1] or SURBL [2]), and requires spammers to change domains frequently to avoid detection. To study the usage patterns of such "spamvertized" domains, we focused on

Type	Time	Types	Insts	Total templates	Unique templates (#/%)		Unique headers (%)	Unique Msg-IDs (%)	Unique bodies (#/%)	
pharma	241 d	2	31	4,139,577	69,902	1.69	99.34	99.30	10	0.01
self-prop	240 d	5	32	2,042,755	35,489	1.74	94.45	94.38	20	0.06
stock	267 d	8	25	595,517	11,041	1.85	96.21	95.92	47	0.43
image	47 d	1	2	82,680	6,323	7.65	62.90	25.32	6,323	100.00
job-ad	299 d	60	79	75,114	72	0.10	65.28	1.39	71	98.61
personal	91 d	8	8	1,352	7	0.52	85.71	0.00	7	100.00
political	1h 6m	1	1	3,952	3	0.08	33.33	0.00	3	100.00
other-ad	21 d	3	3	650	3	0.46	100.00	0.00	3	100.00
phish	72 d	1	2	1,794	2	0.11	100.00	0.00	2	100.00
other	1 d	3	4	195	2	1.03	50.00	0.00	2	100.00

Table 4: Duration, number of campaign types & instances, and template uniqueness properties of the ten campaign classes, sorted by template uniqueness. The Unique templates column lists absolute numbers as well as percentages relative to the total number of templates, while the following columns list percentages relative to the number of unique templates. The top four campaign classes exhibit inflated header template uniqueness due to suboptimal macro-less variation of Message-ID headers.

the long-running Pharma campaign as it employed domains throughout. We downloaded daily blacklisting information from the jwSpamSpy blacklist, as it has the added benefit of also providing the day each blocked domain was registered, and used the CB dataset to contrast with the times at which we observed the domains in use.

The Pharma campaign used 557 different second-level domains (often in combination with a random third-level prefix). On average, a domain was used for 5.6 days. The shortest occurrences are just a single dictionary (all in the `.cn` ccTLD), the longest 86 days (all in the `.com` gTLD). In any given hour, an average of 12.9 domains were in active use, 14.7 on any given day. Domain introduction was largely, though not absolutely, abrupt: when new domains were introduced, in 8% of the instances all current domains were replaced, and at least half replaced in 46% of the instances. The average time from a domain's registration to its use is 21 days, while the average time from use of a domain until it appeared on the jwSpamSpy blacklist is just 18 minutes (although, as we have observed in prior work, blacklist usage varies considerably across e-mail domains [7]). This delay varied considerably: half of the domains appeared on the blacklist before the crawler even observed their use; a clear indication of the strong pressure on the spammers to burn through domains quickly.

Figure 7 shows a timeline for all domains, comparing the time from domain registration to its appearance on the jwSpamSpy blacklist and the time when the domain was actively spamvertized by Storm. Several observations can be drawn. First, domains were generally abandoned relatively soon after being blacklisted. Second, large numbers of domains were registered in batches (particularly in 2007), and domains from different registration batches were deployed simultaneously. Third, there is a clear change in the spammers' *modus operandi* at the beginning of 2008: they abandoned domains from the `.cn` ccTLD, they shortened the time from registration to domain use, and they used domains for longer periods of time.

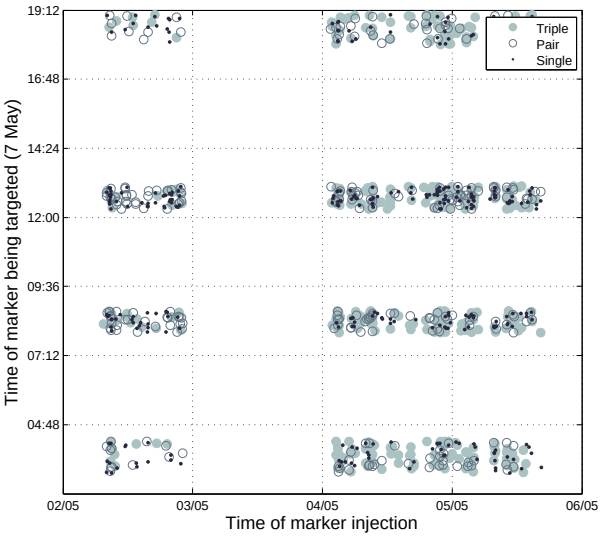


Figure 8: Relative order of marker injection into harvest (x-axis) and appearance as targeted address (y-axis) in a continuous Pharma campaign. "Single" indicates markers being targeted alone, "Pair" with one, and "Triple" with both of the markers they were injected with.

Summary. Template authors are relying nearly exclusively on the use of dictionaries to confuse spam filters. The importance of this approach is evident in spamvertized domains, which appear on blacklists within minutes after being used. Template diversity is employed more as a workaround for lacking template language functionality rather than as a real source of diversity.

5.3 Address harvesting

Our harvest injection experiment confirmed our hypothesis that email addresses harvested from compromised machines were added to the spammer's distribution list. Five days after we injected marker addresses into harvests, both the crawler and the proxies observed the addresses in target lists of the the Pharma campaign, active throughout the experiment. Figure 8 shows the relative

order of injection and appearance of marker addresses.

We make the following observations. First, enumeration shows that the addresses are not used repeatedly, which suggests availability of enough addresses to satisfy worker demand. Second, since the campaign the markers appeared in was operating continuously, the batched appearance in four roughly hour-long bands suggests that addresses are picked in a round-robin fashion from available harvests. Third, since occurrences of markers in a small timeframe cover addresses across the entire span of our injection experiment, it appears some randomization is present in the selection process. Fourth, this randomization partially conserves the grouping of addresses harvested together: 40.2% of the marker addresses were found together with the other two marker addresses injected in the same harvest; 26.3% of marker addresses with together with another marker address and 33.4% of marker addresses with neither. This suggests automated processing of the harvests, but with an algorithm whose strategy is not obviously inferred.

Invalid addresses in target lists. As mentioned in Section 3, bots harvest any string of the form "*@*.*" including strings that are not valid email addresses: about 0.6% of addresses in the CB dataset did not end in a valid TLD. Of these, about 12% are `.com` followed by an additional character, e.g., `.comc`. Another 8% are common file extensions, e.g., `.jpg` or `.dll`. The presence of these addresses indicates that there is very little checking performed on the harvested addresses before use.

Bot- vs. Web-based harvesting. The advent of botnets has provided spammers with an additional feed of addresses, namely those found locally on the infected systems. Storm's address harvests are an example of this approach. While we lack the means to compile a comprehensive hitlist of email addresses found on the Web (and refrained from purchasing any such list) for comparison against the targeted addresses in our datasets, we can do the opposite and measure the "Web presence" of the latter to get an indication of how much this visibility into the end host benefits the spammers.

We constrained ourselves to randomly sampling 10,000 unique addresses (with valid TLDs) from the harvests and target lists of the PB dataset, and issued queries for the precise address strings to the Google search engine. For both lists, the fraction of addresses not found on the Web is substantial: over 76% of the harvested addresses are only available on infected machines, as are over 87% of the targeted addresses. Interestingly, the fraction of Web-visible addresses is actually larger among the harvests than in the target lists, which suggests it is unlikely that the target lists contain significant feeds of Web-based addresses. A third, unknown source of addresses may also account for the difference.

Campaign	Class/ # Types	Total Addrs	Top TLD	TLD %
Main pharma	pharma / 1	233,904,960	com	59.81
Main self-prop	self-prop / 5	78,446,044	com	62.25
TBCO stock	stock / 2	14,047,724	com	64.83
MPIX stock	stock / 1	8790,387	com	66.62
Image spam	image / 1	5984,753	com	64.14
Hyphenated A	job-ad / 18	1,006,992	ca	80.83
Italian	job-ad / 3	458,615	it	96.72
German stock	stock / 1	167,779	de	51.56
William	job-ad / 1	147,035	ca	56.15
Polit. party	political / 1	142,229	ua	82.00
Global union	job-ad / 1	131,453	au	87.75
Canada	job-ad / 4	130,883	ca	79.21
Worldlines	job-ad / 1	77,712	it	60.32
Spanish	job-ad / 2	62,357	es	81.10
Hyphenated B	job-ad / 2	48,857	au	99.44

Table 5: Campaign types, classes, sizes, and TLD targeting for the 5 largest campaigns (of 25) where the top TLD is `.com` (top) and 10 largest (of 30) where it is a ccTLD (bottom).

5.4 Spam targeting

We observed large differences in size, domain distribution, and email address overlap between the target lists of the campaigns. Table 5 shows the largest untargeted and country-targeted campaigns. Here, we aggregated campaign types where we suspect a common campaign initiator. This aggregation mainly affected a series of job ads where the domains in the contact addresses followed a two-part, hyphen-separated naming scheme.

The text of several job advertisements and stock scams limited the intended respondents to specific countries, particularly Canada, the United States, and Australia. Two job offer campaigns explicitly soliciting US citizens advertised exclusively to `.us` domains, implying that the spammer was intentionally limiting distribution to United States residents, even though usage of gTLDs (generic TLDs, e.g., `.com`) for American email addresses is much more common. A third US-targeted campaign included a very small minority of non-`.us` domains, mostly large email providers.

Although a large majority of the addresses in the associated distribution lists for the Canadian and Australian campaigns end in `.ca` and `.au`, each list also includes non-ccTLD addresses from the countries' major ISPs as well as other domains not specifically associated with the corresponding country. This artifact suggests that the strategy for compiling these lists differs from that used for the US targeted campaigns detailed above.

We observed multiple instances of target list overlap between self-propagation and pharmaceutical campaigns. This overlap strongly suggests that both campaigns use the same address list. Comparing the domain distribution and email address overlap for address lists, we inferred that a majority of the campaigns using different template bodies were likely drawing from the same collection of email addresses. Furthermore, it seems that

domain meta-information is leveraged for targeting in order to select geographically relevant gTLD domains in addition to ccTLDs.

5.5 Noteworthy encounters

It is commonly assumed that spam is mostly driven by insidious motives. One campaign class we encountered suggests otherwise: political campaigning. Th campaign in question — lasting less than two hours on 10 July 2008 and targeting over 142,000 addresses of which 82% have the Ukrainian TLD `.ua` and 4% `.ru` — is a call to establish a new Ukrainian political party. (A translation of the (static) template body is available at `http://www.icir.org/christian/storm/ukraine-campaign/`.)

On 21 days between 20 November 2007 and 11 February 2008, we observed 670 instances of `pharma_links` dictionaries containing a web server error message rather than a list of domains. These messages included a SpamIt.com copyright notice, suggesting was using SpamIt.com, which we believe to be a pharmacy affiliate program.

6 Conclusion

In this paper we have presented a detailed study of spam campaign orchestration as observed in the wild. Our investigation was enabled by a long-term infiltration of the Storm botnet, comprising both passive probing and active manipulation of the botnet's C&C traffic.

Our study includes over 800,000 spam templates, more than 3 million harvested email addresses, and target lists comprising more than 630 million email addresses from 94 different campaign types hosted over a period of 10 months. Our analysis confirms that today's spamming business operates at a frightening scale without requiring truly sophisticated mechanisms to conquer the hurdles put in place by the anti-spam industry. Thus, to the detriment of productivity worldwide, the filtering arms race continues.

7 Acknowledgements

Our thanks to Joe Stewart and Pat Peterson for offering insight into Storm, Erin Kenneally for legal advice, and to Gabriel Lawrence and Jim Madden for support at UCSD. Our data collection was made possible by generous support and gifts from ESnet, the Lawrence Berkeley National Laboratory, Cisco, Microsoft Research, VMware, HP, Intel and UCSD's Center for Networked Systems, for which we are very grateful.

This work was made possible by the National Science Foundation grants NSF-0433702 and NSF-0433668. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the National Science Foundation.

References

[1] jwSpamSpy Spam Domain Blacklist. `http://www.joewein.net/spam/spam-bl.htm`.

[2] Spam URI Realtime Blackhole Lists. `http://www.surbl.org`.

[3] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: Characterizing Internet Scam Hosting Infrastructure. In *Proceedings of the USENIX Security Symposium*, Boston, MA, Aug. 2007.

[4] J. Franklin, V. Paxson, A. Perrig, and S. Savage. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *ACM CCS*, Washington, D.C., Nov. 2007.

[5] L. Frieder and J. Zittrain. Spam Works: Evidence from Stock Touts and Corresponding Market Activity. Technical Report 2006-11, Berkman Center, March 2007.

[6] M. Hanke and F. Hauser. On the effects of stock spam e-mails. *Journal of Financial Markets*, 2007.

[7] 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 *ACM CCS*, pages 3–14, Alexandria, Virginia, USA, October 2008.

[8] C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, and S. Savage. The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, April 2008.

[9] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. On the Spam Campaign Trail. In *Proceedings of the First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, San Francisco, CA, April 2008.

[10] Paul Bäher, Thorsten Holz, Markus Kötter and Georg Wicherski. Know your Enemy: Tracking Botnets. In *The Honeynet Project & Research Alliance*, March 2005.

[11] P. Porras, H. Saïdi, and V. Yegneswaran. A Multi-perspective Analysis of the Storm (Peacomm) Worm. Technical report, Computer Science Laboratory, SRI International, October 2007.

[12] M. Prince, B. Dahl, L. Holloway, A. Keller, and E. Langheinrich. Understanding How Spammers Steal Your E-Mail Address: An Analysis of the First Six Months of Data from Project Honey Pot. In *Second Conference on Email and Anti-Spam (CEAS)*, 2005.

[13] C. Pu and S. Webb. Observed Trends in Spam Construction Techniques: A Case Study of Spam Evolution. In *Third Conference on Email and Anti-Spam (CEAS)*, 2006.

[14] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Proceedings of the ACM Internet Measurement Conference*, Rio de Janeiro, Brazil, Oct. 2006.

[15] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proceedings of the ACM SIGCOMM Conference*, Pisa, Italy, Sept. 2006.

[16] S. Sarat and A. Terzis. Measuring the Storm Worm Network. Technical Report 01-10-2007, Johns Hopkins University, 2007.

[17] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, I. Osipkov, G. Hulten, and J. Tygar. Characterizing Botnets from Email Spam Records. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, April 2008.