No Plan Survives Contact: Experience with Cybercrime Measurement

Chris Kanich[*] Neha Chachra[*] Damon McCoy[*] Chris Grier[†] David Y. Wang[*]
Marti Motoyama[*] Kirill Levchenko[*] Stefan Savage[*] Geoffrey M. Voelker[*]

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

[†]*Computer Science Division*
University of California, Berkeley

Abstract

An important mode of empirical security research involves analyzing the behavior, capabilities, and motives of adversaries. By definition, such measurements cannot be conducted in controlled settings and require "engagement" directly with adversaries, their infrastructure or their ecosystem. However, the operational complexities required to successfully carry out such measurements are significant and rarely documented; blacklisting, payment instruments, fraud controls and contact management all represent real challenges in such studies. In this paper, we document our experiences conducting such measurements over five years (covering a range of distinct studies) and distill effective operational practices for others who might conduct similar experiments in the future.

1 Introduction

Experimental research, by definition, seeks to examine unknowns that cannot be evaluated entirely in the closed world of analytic methods. In many cases, repeated experimentation can address such questions, teasing apart independent and dependent variables by carefully controlling the environment. However, in other situations the environment is too large and complex to control and it is infeasible to test repeatedly (e.g., astronomy, economics, geology, etc.). Thus, researchers rely on "natural" or "observational" experiments instead—studies focused on gathering the data needed to drive inference.

Experimental computer security research has a similar dichotomy. While there is an important place for controlled experiments (e.g., how well a particular technology can defend against a known attack), so too do partially controlled experiments and observational studies play an important role in understanding the nature of current threats by measuring the behavior, *in situ*, of attackers and victims alike. However, while the methodological requirements of controlled security experiments are widely documented (indeed, CSET has been a primary venue for such discussions) there is far less published work addressing the needs of those working "in the field". This situation is particularly true for experiments that directly and actively engage attackers, their infrastructure or their ecosystem.

In this paper, we focus squarely on these issues and describe our experience over several years of studies measuring cybercrime activities. In Section 2, we describe our efforts performing large-scale crawling and monitoring of spam-advertised sites and cloaked Web search results. In Section 3, we discuss the methods we have developed for purchasing goods and services from such sites, as well as purchasing directly from scammers themselves. In particular, we focus on the operational challenges to achieving verisimilitude—obtaining measurements that capture "real" behaviors—many of which arise from the adversarial nature of the measurement process. We describe the difficulties we encountered, and how our measurement protocols evolved over time to address these issues. Finally, in Section 4 we provide a roadmap for researchers seeking to perform the same types of studies (without repeating our mistakes).

2 Crawling

Crawling URLs advertised in spam is the most natural way to engage the infrastructure that directly underlies the spam business model, and has long been a standard technique among security groups in both academia and industry. We have used crawling for a variety of measurement projects, as have other groups investigating a wide variety of problems including the network characteristics of spam relays [18], Web hosting [9], phishing sites [13], blacklist effectiveness [19], spamming botnets [5, 21], and fast-flux networks [4], just to name a few.

In this section we describe the evolution of our crawling methodology across a series of projects spanning five years (Table 1). The changes that we have made over time, while reflecting a progression of our research goals, have nearly all been motivated in response to the escalation of defenses on scam sites (which have ranged from indifference to outright aggression to active deterrence). Broadly speaking, over time crawling spam-advertised Web sites has required ever more fidelity in mimicking the behavior and activity of real users. Whereas a command-line tool (e.g., wget) running on a single machine was sufficient to crawl the URLs in a moderate spam feed five years ago, today we use a cluster of machines each running up to 100 instances of a complete

Source	Challenges	Tool	Years
Spam	HTTP, HTML Meta, simple JavaScript redirects	Command-line tool	2006–2007
Spam	Popups, image overlays, IP blacklisting, malware	Browser w/ plugins	2008–2011
Search	Cloaking based on Referer, User-Agent fields	Parameterize HTTP fields per URL	2010–2011

Table 1: Evolution of crawling methodology.

modern browser that incorporates add-ons to mimic user activity (e.g., clicking on popups) and specific browsing situations (e.g., setting `Referer` and `User-Agent` HTTP fields to mimic search results). In turn, we tunnel traffic through a diverse range of independent IP endpoints and monitor behavior to automatically accommodate a wide range of failure conditions (from IP blacklisting to browser malware).

2.1 Redirection

There were few difficulties in our first experiences with crawling spam URLs in 2006; compared to later trends, spam-advertised sites did little to deter crawling. Our initial goal was to understand the host and network characteristics of the infrastructure hosting spam-advertised sites [1]. At the time, we could simply use a command-line tool for visiting a URL and downloading the contents of a Web page. The primary obstacle was handling redirection chains between the spam-advertised URL and the final landing page of the site being advertised. To handle this situation, the tool included logic to follow HTTP redirects, HTML META redirects, and JavaScript redirects (which were straightforward to identify and parse automatically at the time). If the URL to the final landing page was one we had not encountered before, we then used a browser to download, render, and record a screenshot of the page.

This crawling approach was efficient and scalable. The command-line tool required few resources, and we could use a single machine to crawl all URLs in our spam feed at the time (45K URLs/day). Using a browser to render and capture a screenshot requires substantially more resources, but only unique URLs to final landing pages required a screenshot (about 10% of the feed).

Soon thereafter came a radical change: some spam-advertised sites went from being indifferent to highly aggressive in reacting to crawling. Crawling sites advertised via the Storm botnet in 2007 triggered a high-bandwidth distributed denial-of-service attack against the crawling machine [17]. As with many groups crawling spam sites at the time, we quickly became DDoS targets since our spam feeds included spam sent by Storm bots. Network filters rendered the DDoS attack harmless, though, underscoring a benefit of having a good working relationship with the network operations group of one's enterprise.

Storm eventually stopped this behavior. We suspect that those operating Storm concluded that this reaction was *too* aggressive: in the long term it backfired, drawing substantial attention to Storm and the sites it advertised.[1] As a result, we do not expect DDoS to be a primary risk for crawling efforts of a purely research nature (although we are aware of significant DDoS attacks directed at sites whose results are used operationally, e.g., Spamhaus, `abuse.ch`, etc.).

2.2 Deterrence

Instead, spam sites began to use more practical defenses to deter crawling. Over time we gained access to a variety of large-scale spam feeds, which we used in a subsequent project to more deeply measure and characterize the spam value chain [12]. Crawling again was central to this goal.

Over the past two years spam-advertised sites have increasingly used more sophisticated redirection techniques designed to trick users, but also make crawling more difficult. In particular, sites use JavaScript to present popups to users that require a mouse click event to proceed to the final landing page, and use image overlays on the page to the same effect. Furthermore, such sites blacklist IP addresses suspected of crawling.[2]

As a result, today crawling spam-advertised sites requires greater efforts in appearing indistinguishable from real users, greater resources dedicated to crawling, and more sophisticated failure handling. Simply put, crawling today requires using a popular browser to ensure fidelity; gone are the days of command-line tools. As extensible platforms, add-ons can make browsers act more like users; to deal with the more sophisticated "redirects", we added an extension to detect popups and images and issue mouse clicks. Unfortunately, crawling URLs using a full browser greatly increases the CPU and memory resources required; connection timeouts from blacklisting and unreachable domains further

[1] Including our group—after we became a target of Storm, we began reverse-engineering the botnet to crawl and infiltrate the system.

[2] In our related activities monitoring underground forums, and through collaborations with similarly focused researchers, we have found a range of "blacklist" firewall configurations designed to specifically block traffic from various security groups, including our own. This blacklisting includes both individual IP addresses as well as entire address ranges, /24 and larger, associated with particular security organizations.

tie up resources, preventing quick recycling (although some of this can be addressed through per-machine parallelism and configuring kernels to provision more network resources). Expect to dedicate a cluster to crawl any reasonably-sized feed of URLs; during busy days for [12], using a cluster of 30 servers we crawled over 600K URLs/day with brief bursts corresponding to peak rates of 2M URLs/day.

To counter blacklisting, we use a combination of prevention and detection. To avoid being blacklisted, we tunnel HTTP requests through proxies running in multiple disparate IP address ranges, using various cloud hosting and IP address reselling services, as well as address blocks loaned to us from individuals and via experimental allocations from the Regional Internet Registries. We then randomize HTTP requests across the address ranges to minimize the footprint of any single IP address for any given site. Blacklisting manifests either as DNS errors (the name server is also commonly an element of scam infrastructure), 5xx HTTP error codes, or connection timeouts. We detect that our crawling system is being blacklisted by monitoring the rates of such errors and reacting when short-term rates well exceed long-term rates. In response, we retry requests using a different IP address range.

For long-term steady-state operation, we have had to make the current incarnation of our crawling system robust to a variety of failure conditions. To tolerate intermittent network or server issues, the crawler makes multiple attempts to visit a URL before deciding it is not valid. To prevent stalled connections from idling a browser instance indefinitely, it times out long page loads after multiple minutes. It also detects browser failures (e.g., a hung process) with heartbeat requests from a controller every 15 seconds; if a browser does not respond, the controller restarts it. To ameliorate the effects of any malware infections, memory leaks, or other resource leaks, we reboot the crawler and its browsers on every machine in the crawling cluster every 24 hours.

One final challenge is *implicit* DDoS on crawlers via spam poisoning. In particular, the Rustock bot started emitting large amounts of spam e-mail containing URLs with random `.com` domains (literally millions of both real and unregistered domains, none of which was truly being advertised [2]). The purpose of this campaign appears to be both poisoning blacklisting services with large numbers of false positives and overwhelming crawlers such as ours with timeouts and diverse useless page loads. When this behavior started in September of 2010, we were able to manually identify some lexical patterns used across most of these URLs and tried to filter them out using regular expressions. This approach was ultimately unsuccessful as the operators of Rustock changed their poisoning code to become ever more random. To address this issue we were forced to add state to our crawler and, instead of blindly crawling all URLs, use a method that tracks the appearance of individual registered domains over time. Thus, the system now schedules crawls based on how frequently a registered domain has been seen. This approach prioritizes new domains, minimizing the overhead and blacklisting risk of re-crawling the same domain many times, but not crawl millions of domains that have only been seen once.

2.3 Search URLs

Finally, expanding into different crawling domains inevitably introduces new challenges. We recently started crawling URLs in search results to explore Web site cloaking [20] and black-hat search-engine optimization (SEO) activity [3], which requires yet more sophisticated mimicry to emulate real users. In particular, crawling a cloaked page returns different results depending on the HTTP `Referer` and `User-Agent` fields. Sites decide whether a request comes from the result of a search based upon the contents of the `Referer`, cloaking the contents otherwise. Sites further return different content depending on the operating system specified in the `User-Agent` string (e.g., a scam site will sell fake anti-virus software to Windows-based visitors and offer an iPod scam to Mac-based visitors). A crawling system for such URLs therefore requires the further ability to parametrize specific HTTP fields for each URL crawled (and, when possible, to proxy such requests through address space that would be appropriate for a search engine crawler).

3 Financial transactions

While some studies can be completed purely using network-level measurements (either active or passive), in many cases this vantage point can only take one so far. In particular, when studying the nature of goods and services on offer via the criminal ecosystem (e.g., including those advertised to the general public, such as spam-advertised pharmaceuticals, and those rendered to “the trade” such as underground VPNs, exploit kits, compromised accounts and so on) it is difficult to do so without placing direct financial transactions via purchasing.

Placing such orders can be operationally difficult, however, and ensuring “realism” creates particular challenges. In this section we explain these challenges and how our protocol for handling financial transactions has changed over the last two years of active involvement.

First, we should make clear that independent of the challenges to verisimilitude, active measurements such as purchasing from criminals create their own ethical, legal and operational sensitivities. These issues are not the focus of this paper, but we wish to emphasize that they have consumed significant attention. All of the work we describe has been with the knowledge and oversight of multiple lawyers—both specialists in cyberlaw and general counsel for our institution—and has either been reviewed by our IRB (when they deemed human subjects to be involved) or consistent with a pre-established set of ethical guidelines that our group has followed consistently. Indeed, we invested significant time in consultation with, and education of, our administrators, overseers and advisers, to arrive at these decisions. Finally, managing the *funding* of such activities through a university administration took several years of internal trust building—"We need to be reimbursed for large numbers of cash equivalent payments for goods that may never be delivered, that will probably have no receipt and will, at times, involve our being defrauded. Is that okay?"—and the development of appropriate industry funding sources.

The remainder of this section focuses on the *operational* requirements of such purchasing activity. We separate these financial interactions into two categories: those in which we pose as fellow scammers and those in which we pose as customers from the general public. The distinction is driven both by the unique characteristics of each domain as well as the requirements needed to maintain "cover" in taking measurements.

3.1 Purchasing in the underground

For several past studies (as well as studies ongoing) we needed to directly conduct commerce with a miscreant individual. In our experience there are similar issues across a broad range of underground goods and services, including participating as a customer of CAPTCHA-solving services (spending $3,400 over five months) [14], obtaining underground software packages ($640) [14], hiring freelance workers for Web service abuse-related tasks ($2,100) [15], purchasing VPN service, packers, etc.

The first issue is language and culture. While some goods and services are broadly sold and therefore can be negotiated for in English, there are important subcommunities for which speaking in a particular native language (e.g., Russian) is a de facto requirement to establish baseline credibility. In such cases, it is necessary to have a *native* speaker (with appropriate IM client and keyboard) and sufficient underground context to be able to conduct business appropriately (i.e., Google Translate is insufficient). For example, much Russian underground slang is transliterated from English; e.g., the term конверт, meaning *conversion rate*, is taken directly from the English word "convert", although absent this slang context the word would be translated in English as "envelope". Similarly, ICQ is by far the most popular instance messaging service used in Russia and anyone from that environment would be familiar with its use.

The next most important issue is the means of payment. Unlike sales to the broader market, underground sales are never conducted using traditional payment instruments such as Visa or MasterCard (except perhaps via trade). Instead most actors prefer to use online payment systems such as WebMoney, Liberty Reserve, Ukash, and so on, which offer the benefits of anonymity (payments are made entirely using identifiers) and assurance (these are not credit transactions and have no "charge back" facility; all payments are final). However, obtaining access to these currencies is itself non-trivial, as is transferring money into them (e.g., WebMoney does not accept payment via Visa, PayPal or Western Union). Indeed, in conducting experiments using WebMoney we found that the easiest approach was to obtain and fund credentials via a trusted colleague in Russia.

The prevalence of instant messaging and the widespread use of VPNs minimizes the need to establish appropriate IP address origination. Although we have experimented with VPN providers who can offer exit points in cities across the globe for modest fees, we have not found that this level of "cover" is typically necessary (in part because ICQ interactions do not typically leak IP address information unless file transfers take place).

3.2 Purchasing as a customer

Purchasing goods or services offered for sale via traditional channels (e.g., using credit cards via Web sites) appears far easier on the surface, but introduces its own unique challenges in execution at scale. We purchased from a large selection of pharmaceutical, software, and counterfeit luxury goods affiliate programs to identify critical elements of the payment infrastructure [12]; for this effort, we attempted 120 purchases totaling $10,400 over the course of one month in 2010. We subsequently purchased from the subset which revealed volume information in successive purchases to estimate total affiliate program volume and revenue [8]; for this effort we attempted 156 purchases over three weeks totaling $6,600.

Payment cards

The first challenge is in finding appropriate payment instruments. While it is easy to use one's own personal credit card to place a order for spam-advertised herbal supplements or for SEO-advertised fake anti-virus software, this approach has many drawbacks. First it exposes researchers to potential fraud by providing their credit card information. Second, it is difficult to differentiate between orders on a credit card statement since the merchant's identification string is frequently unrelated to the name of the Web site at which an order was placed. Finally, placing many orders from the same card creates a suspicious profile and will quickly trigger standard velocity checks in the merchant's (or issuer's) payment fraud system (or, more likely, a fraud check system operated by their payment processor or gateway service provider).

One alternative that seems to address this issue is the prepaid gift card (issued by banks through both the Visa and MasterCard associations). Such cards are practically anonymous (since the card holder's identity is bound late, only after the card is purchased, with with zero due diligence) and are cheap enough that different cards can be used for different transactions (albeit with some loss due to the residual balance not being easily transferable).[3] Moreover, some issuers will provision "virtual" cards online, allowing new cards to be created on demand and with variable amounts. However, after using such cards for almost eighteen months, we can report that they are less attractive than they first appear.

First, most gift cards only provide telephone support. The assumption is that most users care primarily about their gift card balance (which can be provided via an automated telephone menu) and all other requests (e.g., finding out if a particular transaction settled, the date it settled on, the claimed merchant ID, etc.) require speaking with the customer support desk. Our experience is that even in widely-used gift card brands the customer support desk is staffed by only a few individuals and calling tens of times to request further information creates suspicion of fraud. Moreover, other key information (e.g., information about transactions that authorized but did not settle and the Acquirer's Reference Number, or ARN, identifying the acquiring bank used by the merchant) does not appear to be available to most support desk operators.

A subset of gift cards provides a Web interface that allows holders to obtain most of this information online, thus avoiding the customer support problem.[4] However, after researching the Visa gift card market extensively, we found extremely few U.S. issuers who provide an online Web interface that includes ARN information (critical for experiments that seek to cluster payment processing infrastructure used by different scammers). Through consulting with another researcher in the field, we were directed to one particular brand of card—sold over the counter in West Coast supermarkets—having these properties. We purchased several thousand dollars of these cards only to be undone by Federal legislation. As part of the Credit Card Reform Act of 2009, the FinCEN division of the Treasury department was mandated to revise regulations on the use of prepaid credit cards to address their use as money laundering vehicles. Among these new rules (first proposed in mid-2010) are strict reporting requirements on international transfers. In response, most U.S. Visa gift card issuers elected to simply restrict their cards to domestic transactions. This change was problematic for cybercrime research since virtually all interesting transactions are settled through foreign banks.

In the end, after two years of experimentation, we do not believe there is an easy solution to this problem that is broadly available. Our ultimate solution was to contract directly with a specialty card issuer whose products do not constitute a "prepaid program" and who agreed explicitly to support our research, issue new card numbers on demand (our protocol is to use a single credit card number only one time), and manually export fine-grained information on each transaction (authorization, settlement, acquiring Bank Identification Number (BIN), Card Acceptor ID (CAID), etc.) for a nominal fee. This method required significant negotiation effort on our part as well as an initial investment of $16,000. While it has been tremendously valuable for us, it is unfortunately non-trivial to replicate.

Online fraud checks

Armed with a large supply of Visa payment cards we quickly found that many of our purchases were declined by the sites we purchased from. Through a combination of trial and error (as well as research into commercial payment fraud services) we discovered that many merchants employ a range of anti-fraud measures that we had not anticipated.

The first is the standard Address Verification System (AVS), which is provided through the card association and validates the numeric portion of the customer's street address as well as their ZIP code. In each purchase we had inadvertently forgotten to correctly "program" each card with the corresponding shipping address we planned to use and our transactions were nearly always declined as a result.

Having fixed the address issue, we still found a variety of sites that rejected our orders, typically stating that the "fraud score" was above a threshold. Based on our experiments and examining third-party "fraud check" services, we believe these checks include IP geolocation (matching the location of the purchaser with the shipping address and AVS), validating that shipping addresses correspond to "residential" locations, and flagging "free" email accounts (which, being free, are considered riskier).

For such sites, we modified our operational protocol to address all three issues. We obtained IP endpoints located close to the associated shipping addresses, we

[3] In our experience, placing a single purchase per card number is highly preferable, both for associating individual transactions to individual cards as well as for analyzing future fraud actions. In initial experiments we used $500 gift cards and made multiple purchases on each card, only to find that we could not determine the source of subsequent fraudulent transactions because we had used the card at a variety of sites, making the culprit ambiguous. In our current efforts, we use each credit card number only a single time.

[4] Note that there is no guarantee of such service and one such provider changed their policy *after* we had purchased their cards, returning us to telephone support for key information.

switched to using residential shipping addresses instead of the commercial mailbox provider we had used initially, and we created a range of new domain names to source email addresses (using Google Apps to host the underlying email service). While this significantly reduced our decline rate, we still found sites that tracked past IP purchase history and we needed to allocate a unique IP address for each purchase at such sites.[5] Finally, as with our use of payment cards, we learned that there was significant value in using a unique email address for each purchase—doing so allows disambiguation among customer service messages, and tracking differences in post-order advertising.

Voice contact

Most scam sites require the purchaser to provide a range of contact information, not only shipping and email addresses, but a voice phone number as well—all to support their own fraud concerns. In our experience, these voice numbers get used frequently to confirm orders (as yet another fraud check). This in turn requires a range of phone numbers and some way to relate caller to orders (to recall which "identity" is being called by the merchant). Over time we have adopted a system using multiple prepaid cellular telephones, each associated with multiple Google Voice accounts. This arrangement allows for easy centralized access to voicemail for each account, but also permits the use of geographically accurate phone numbers. We maintained an online spreadsheet that identified outstanding orders, as well as the associated names, credit cards and phone numbers, allowing our purchasers to quickly determine what an incoming call might be related to.

Managing the operations of this channel was among the most problematic of our efforts. First, the time overhead of playing phone tag with different merchants can add up over hundreds of orders. Moreover, even using our shared spreadsheet it was not always clear which order was being called about and our group members were forced to bluff their way through conversations until they could determine their appropriate identity. A further challenge, similar to our experience with gift card support, is that over time phone operators would come to recognize our voices, requiring different "actors" to handle such calls. Finally, if our purchasing behavior exceeded a certain level (at one point we inadvertently placed a large number of orders for an identical product to the same address) it was via phone that we would be challenged, further placing pressure on our buyers to "think fast" for an explanation. When we failed at such ruses, we found that all related orders would be blacklisted (presumably using some combination of source IP address and shipping address).

Shipping

Finally, while virtual goods (e.g., fake anti-virus software, counterfeit software, malware, etc.) can be delivered online, physical goods must be shipped to a postal address. Purchasing these physical goods creates a number of challenges: since we use a range of distinct names for placing orders, there must be associated postal addresses that will accept mail for those individuals.

A natural concern when making these purchases is whether any goods would be received at all. In fact, similar to other online businesses, our experience is that customer service is prioritized and all orders were fulfilled (with a few exceptions due to our own errors).

Our first approach was to deliver all packages to a "virtual suite" at a rented commercial mailbox in a postal annex. Under this arrangement, we simply provided the postal annex with a list of names and then one of us was permitted to pickup shipments for any of those names. Over time this approach created multiple points of stress. First, while products shipped within the U.S. (i.e., via the postal service) did not require a signature on delivery, international shipments typically did. In the beginning the annex employees would allow a single individual to sign for these packages, but as volume increased they demanded signatures from each recipient, creating a bottleneck. Moreover, it became challenging to associate each individual product with the associated order (since it was common for the packaging to not identify the seller or the particular order number provided via the site's payment page). We also believe that our use of a non-residential address increased our fraud score for some merchant's payment systems, increasing the probability of a decline. Finally, at least one seller began to notice the range of purchases to different names being shipped to the same address and this made them suspicious of fraud (eventually declining a range of such orders).

Ultimately, we addressed these problems by using a range of individual residential addresses (volunteered by researchers in our group). This approach significantly increased overhead for our group, however, and required that a large number of our members make regular trips to the post office to sign for international packages.

Finally, we note that receiving each individual shipment, inventorying its contents, shipping information and custom slip, and then mapping it back to an associated purchase transaction was a time consuming task and one fraught with ambiguity. The mapping challenge was greatly eased when we used EMS international shipping (a more expensive shipping option in which a tracking number is associated with an order and appears on the packaging) at the cost of roughly an additional $20 per

[5] Note that a range of sites will block Tor exit nodes and thus Tor is not an effective solution to this problem.

order. Generally, items were packaged as advertised (including a proper customs declaration) but occasionally we received items where the goods had been secreted inside other items (e.g., handicrafts).

4 Lessons Learned

Our experiences in measuring aspects of the cybercrime ecosystem over the past five years have been fraught with missteps and inefficiencies. That being said, both Web crawling and purchasing have provided invaluable insight into underground activities on the Internet. Distilled into a list, our specific actionable recommendations to those starting out in this space include:

- Use full-featured Web browsers to automate user actions required to reach final landing pages. Shortcuts will inevitably be unable to capture the full complexity of how scammers use the Web.

- IP diversity is necessary to prevent blacklisting when crawling a scammer's infrastructure repeatedly. Cloud hosting and IP address resellers are expedient and inexpensive solutions.

- Interactions on the underground can require native language skills and should always be preceded by an extended period of research into cultural norms and behaviors (e.g., via reading underground forums), including the acquisition of appropriate financial and communication services.

- Instrumenting purchases at the financial transaction level is difficult without maintaining a relationship with a payment card issuer. While the start-up costs of creating such a relationship are non-trivial, the continuing cost can be low relative to the value of the information and the reduction in operational effort.

- Purchasing at scale requires significant preparation and identity management. A comprehensive strategy for managing purchaser names, delivery addresses, email addresses, and contact phone numbers is necessary for ensuring successful order processing and minimizing the possibility of being identified as "unusual".

- It is important to build trust over time with administrative oversight organizations to whom such work will appear unusual and potentially risky. It is difficult to go "all in" and get approval for everything at once, and instead one should negotiate around individual efforts and capabilities, making sure to provide both careful documentation and positive feedback at each stage. Finally, one should be persistent and not assume that "no" means "never".

We end with a couple of high-level observations based upon our experiences. First, the adversarial conditions of engaging with attackers, in this case the ecosystem surrounding spam-advertised Web sites, requires repeated updates to experimental methodology over time. Extensible infrastructure, although often a more time-intensive investment up front, more easily accommodates unexpected yet ultimately necessary changes.

Second, actively engaging with attackers and their infrastructure, such as via crawling and purchasing, often results in accidents or serendipitous insights that lead to unexpected discoveries. As just a few examples from our history, crawling efforts triggered Storm denial-of-service, which motivated us to reverse-engineer and infiltrate the Storm botnet [7, 11, 10]; infiltrating Storm inspired botnet-tailored spam filtering defenses [16], as well as deeper insights into the business practice of spamming botnets [6]. Similarly, purchasing from spam-advertised sites at scale [12] revealed patterns tied to the business processes of spam-advertised sites [8]. Little of this knowledge could have been acquired via a passive approach.

Acknowledgments

We would like to thank the anonymous reviewers and Steve Schwab, our shepherd, for their helpful feedback and comments. In the various projects supported and enabled by the efforts described in this paper, many individuals and organizations aided our work over the years and we wish to gratefully acknowledge their contributions, support, and feedback. In particular, we would like Chris Fleizach and David Anderson who, early on, were willing to chip away at understanding the spam business model. This work was supported in part by National Science Foundation grants NSF-0433668, 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). 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 *Proceedings of the 16th USENIX Security Symposium*, Boston, MA, Aug. 2007.

[2] K. Chiang and L. Lloyd. A Case Study of the Rustock Rootkit and Spam Bot. In *Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, Apr. 2007.

[3] Z. Gyöngyi and H. Garcia-Molina. Web Spam Taxonomy. In *Proceedings of the 1st ACM International Workshop on Adversarial Information Retrieval on the Web*, May 2005.

[4] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and Detecting Fast-Flux Service Networks. In *Proceedings of the 15th Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2008.

[5] J. P. John, A. Moshchuk, S. D. Gribble, and A. Krishnamurthy. Studying Spamming Botnets Using Botlab. In *Proceedings of the 6th ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, Apr. 2009.

[6] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, V. Paxson, G. M. Voelker, and S. Savage. Spamalytics: An Empirical Analysis of Spam Marketing Conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, Alexandria, VA, Oct. 2008.

[7] C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, and S. Savage. The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff. In *Proceedings of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, San Francisco, CA, Apr. 2008.

[8] C. Kanich, N. Weaver, D. McCoy, T. Halvorson, C. Kreibich, K. Levchenko, V. Paxson, G. M. Voelker, and S. Savage. Show Me the Money: Characterizing Spam-advertised Revenue. In *Proceedings of the 20th USENIX Security Symposium*, San Francisco, CA, Aug. 2011.

[9] M. Konte, N. Feamster, and J. Jung. Dynamics of Online Scam Hosting Infrastructure. In *Proceedings of the 10th Passive and Active Measurement Conference (PAM)*, Seoul, South Korea, Apr. 2009.

[10] 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 1st USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, San Francisco, CA, Apr. 2008.

[11] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamcraft: An Inside Look at Spam Campaign Orchestration. In *Proceedings of the 2nd USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Boston, MA, Apr. 2009.

[12] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, N. Weaver, V. Paxson, G. M. Voelker, and S. Savage. Click Trajectories: End-to-End Analysis of the Spam Value Chain. In *Proceedings of the 32nd IEEE Symposium and Security and Privacy*, Oakland, CA, May 2011.

[13] T. Moore, R. Clayton, and H. Stern. Temporal Correlations between Spam and Phishing Websites. In *Proceedings of the 2nd Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Boston, MA, Apr. 2009.

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

[15] M. Motoyama, D. McCoy, K. Levchenko, G. M. Voelker, and S. Savage. Dirty Jobs: The Role of Freelance Labor in Web Service Abuse. In *Proceedings of the 20th USENIX Security Symposium*, San Francisco, CA, Aug. 2011.

[16] A. Pitsillidis, K. Levchenko, C. Kreibich, C. Kanich, G. M. Voelker, V. Paxson, N. Weaver, and S. Savage. Botnet Judo: Fighting Spam with Itself. In *Proceedings of the 17th Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2010.

[17] P. Porras, H. Sadi, and V. Yegneswaran. A Multiperspective Analysis of the Storm (Peacomm) Worm. SRI International Technical Report, October, 2007.

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

[19] S. Sinha, M. Bailey, and F. Jahanian. Improving Spam Blacklisting through Dynamic Thresholding and Speculative Aggregation. In *Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2010.

[20] B. Wu and B. D. Davison. Detecting Semantic Cloaking on the Web. In *Proceedings of the 15th ACM International Conference on World Wide Web*, Edinburgh, Scotland, May 2006.

[21] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming Botnets: Signatures and Characteristics. In *Proceedings of the ACM SIGCOMM Conference*, Seattle, WA, Aug. 2008.