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
christian@icir.org,vern@cs.berkeley.edu

[*]Dept. of Computer Science and Engineering
University of California, San Diego, USA
{ckanich,klevchen,voelker,savage}@cs.ucsd.edu
bmenrigh@ucsd.edu

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 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.

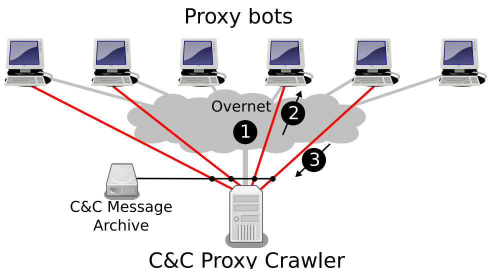

Figure 1: C&C crawler setup used for long-term collection of spam update messages. ❶ Overnet crawler taps into Overnet to find proxy bots. ❷ C&C crawler queries active proxies for update messages. ❸ Proxy bots respond with update messages.

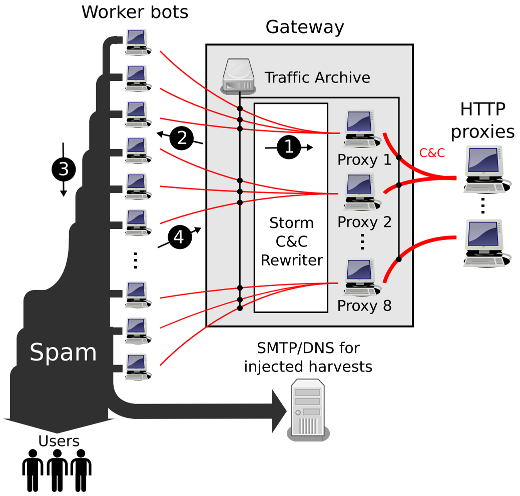

Figure 2: Measurement and rewriting infrastructure for proxy-based C&C traffic. ❶ Workers report harvests, which are optionally rewritten by our interposition setup. ❷ Workers obtain update messages. ❸ Workers start spamming. ❹ Workers summarize spam run in delivery reports.

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.

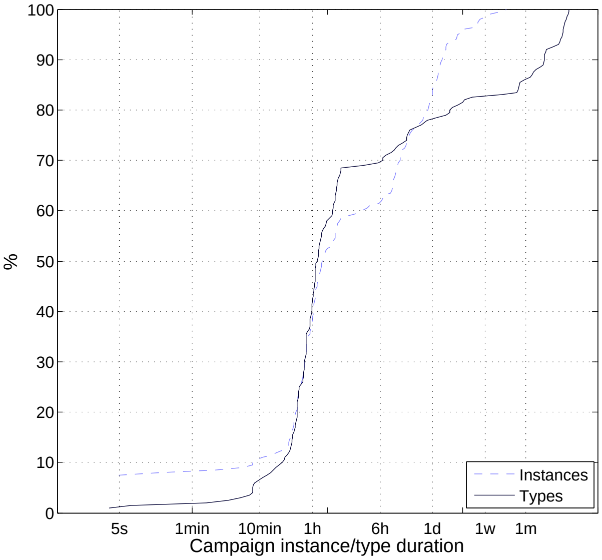

Figure 3: Distribution of campaign instance durations in comparison to campaign type durations.

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 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.

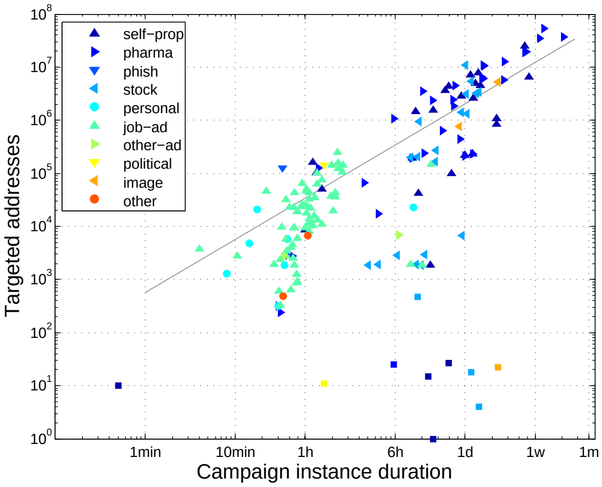

Figure 4: Duration of campaign instances vs. number of email address targets the crawler obtained per instance. The average address retrieval rate 570 addresses per minute. Square markers indicate test campaigns.

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.

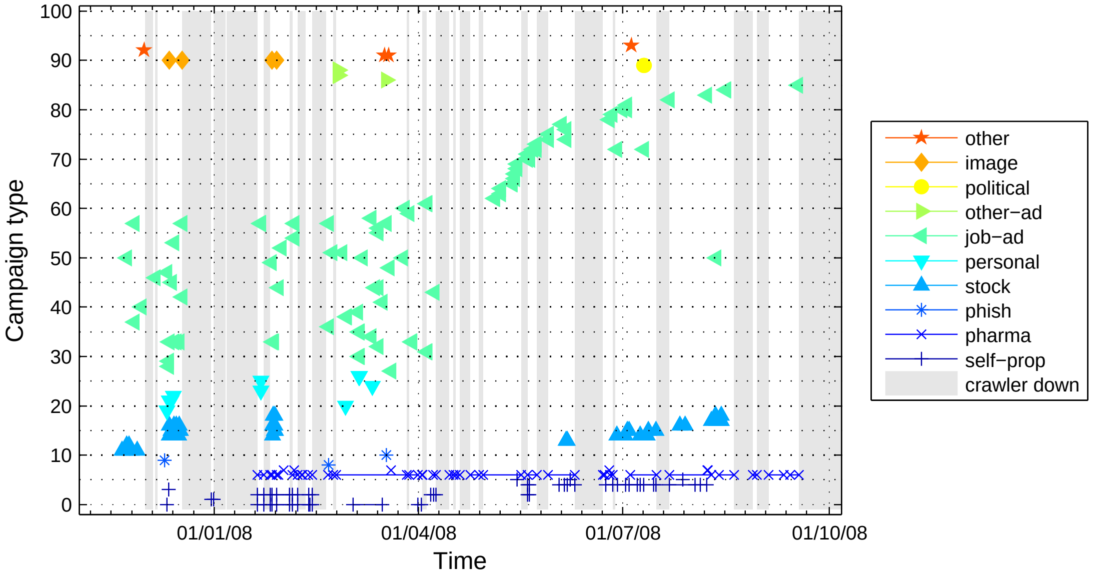

Figure 5: Classes, types, and instances of spamming campaigns identified over time. Shaded areas indicate periods when the crawler was off-line for maintenance or development.

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.

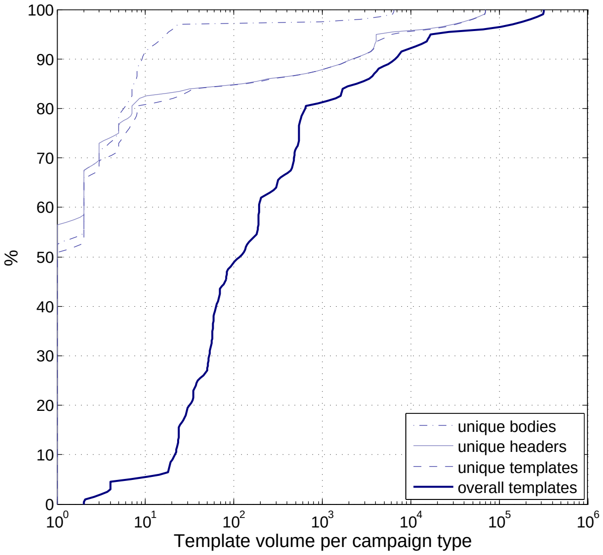

Figure 6: Distribution of overall templates, unique templates, unique headers, and unique bodies, across campaign types.

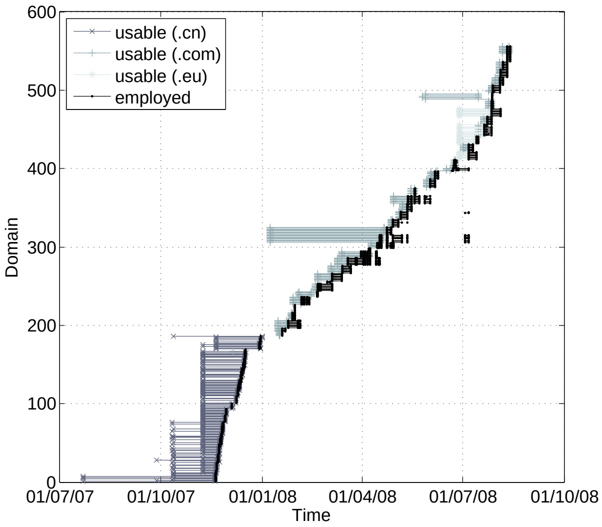

Figure 7: Spamvertized domain usage in the Pharma campaign, from the CB dataset. Black lines indicate timeframes during which the domain was actively used, while lighter colors illustrate the period from a domain's registration until its appearance on the Spam Domain Blacklist. Different colors represent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

				Total	Unique		Unique	Unique	Unique	
Type	Time	Types	Insts	templates	templates (#/%)		headers (%)	Msg-IDs (%)	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.

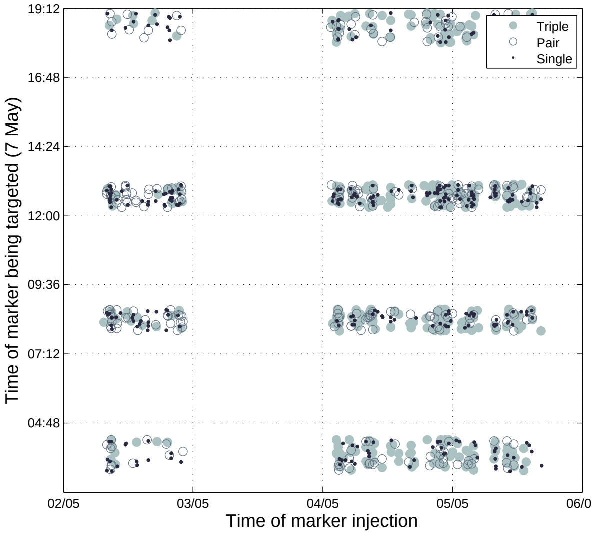

Figure 8: Relative order of marker injection into harvest (x-axis) and appearance as targeted address (y-axis) in a continuous Pharma campaign. "Single" indicates markers being targeted alone, "Pair" with one, and "Triple" with both of the markers they were injected with.

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.

	CLASS/	TOTAL	TOP	TLD
CAMPAIGN	# TYPES	ADDRS	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. The 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.