

THE GAME GOES ON: AN ANALYSIS OF MODERN SPAM TECHNIQUES

Ross Thomas, Dmitry Samosseiko
SophosLabs Canada, Sophos, 580 Granville
Street, Vancouver, BC V6C 1W6, Canada

Tel + 1 604 484 6479 • Email {Ross.Thomas,
Dmitry.Samosseiko}@sophos.com

ABSTRACT

Spam is perhaps one of the most rapidly changing forms of communication we see today. The spammers' methods of evading detection evolve constantly, differing significantly now from what was employed even in the recent past.

Content-based filtering – still a necessary part of any broad and proactive anti-spam solution – is by no means immune from their efforts. Whether based on signatures, URL blocking or heuristic rules, these filters are still sometimes thwarted by sophisticated HTML- and CSS-based obfuscation methods, or by placing the entire content of the message in randomized attached images.

Spammers also tirelessly seek loopholes in domain name registration systems that allow them to avoid pre-emptive detection, and in the security measures of free web-hosting providers so they can mass-register thousands of new home pages every day.

The paper will provide an analysis of many modern anti-anti-spam techniques, accompanied by statistical reports and real-life examples. It will also outline some possible approaches to combat these often highly effective and thus increasingly 'popular' spam techniques.

Although Internet spamming has been with us since as early as 1978 it first became more than a minor annoyance around September 1993, when *America Online* released *AOL for Windows* and the exponential expansion of the Internet began. At first, and for years subsequently, Usenet- and then email-based spam was very simple, consisting of unvarying ASCII messages sent from a limited number of IP addresses. Such simple 'plain text' spam required correspondingly unsophisticated approaches to blocking it. Content-based techniques such as keyword scanning and straightforward hashes (or 'signatures') over the message body were very effective, and at the connection level IP blocklist pioneers such as *Spamhaus* and *MAPS* helped turn spammers away before they could even ring the doorbell.

Since then spammers have developed a variety of methods to bypass the filters, having a counter-countermeasure for every anti-spam technique devised, targeting both connection- and content-level filtering. In the former case, huge networks of compromised home PCs, known as 'botnets', are the most well-known. But another trick employed recently (predominantly by so-called 'Nigerian' scammers) is effective not only against IP blocklists but also such emerging technologies as DomainKeys and SPF/Sender-ID. In this particular example the trick exploits a *Yahoo! Mail* service

allowing new customers to inform their contacts of their new *Yahoo!* email address. The scammer pastes a big list of target email addresses, writes his plea for assistance in the 'personal message' area of the form (see Figure 1), passes the CAPTCHA ('Completely Automated Public Turing test to tell Computers and Humans Apart') test, and his email is dispatched from *Yahoo!*'s mail servers complete with valid SPF and DomainKeys information. Note that, while *Yahoo! Mail* claims to restrict the personal message to 100 characters, emails arriving on *SophosLabs*' spam traps at the time of writing indicate that the scammers have discovered a way to greatly exceed this limit. This technique is, of course, of limited utility and (presumably) longevity, but though the proportion of spam sent this way is negligible, this and similar exploits could be a thorn in the side of anti-spam filters relying solely on the connection-level approach.

Figure 1: *Yahoo! Mail*'s 'Announce Address' feature has been exploited by scammers.

On the content front, obfuscation still lies at the heart of anti-anti-spam methodology. It's a well-known fact that given all the tricks spammers use to veil their words, it becomes possible according to one estimation to misspell 'Viagra' in more than 10^{21} different ways. (That is, rather appropriately, over one *sextillion* combinations.) But modern spam has evolved many more sophisticated ways of mentioning the unmentionable.

Take, for example, one of the numerous 'float tricks'. Cascading Style Sheets (CSS) allow block-level elements to be 'floated' alongside each other, this functionality being most often employed to implement column-style layouts in web pages. But it also allows spammers to break words into bits, and insert spurious characters into common trigger words, in an attempt to fool filters. Once rendered by the HTML engine, though, those bits are reassembled in the correct order, the spurious additions shunted off to the right-hand margin (see Figures 2 and 3). Another, albeit rarer, technique is to use the

```
<DIV><FONT face=3Darial size=3D2><STRONG>Uni<span style=3D"
float
: right
"> t </span>ersity Deg<span style=3D"
float
: right
"> h </span>ree</STRONG><BR>
&nbsp; &nbsp; <BR>
Have you e<span style=3D"
float
: right
"> S </span>ver thou<span style=3D"
float
: right
">C</span>ght that the on<span style=3D"
float
```

Figure 2: The 'float-right' trick: Jumbled HTML with random spurious characters inserted...

<p>University Degree</p> <p>Have you ever thought that the only thing stopping you from A great job and better pay was a few letters behind your name?</p> <p>Well NOW you may get em'!</p> <p>BA - MBA - Ph.D - High School Diploma</p> <p>-No Study Required! -100% Verifiable!</p> <p>These are real, genuine degrees They are fully verifiable and certified transcripts are also available.</p> <p>Simply Call : 1-707-274-4201 24 hours a day, 7 days a week including Sundays & Holidays.</p>	<p>ht</p> <p>6fxCS A4R2s5</p> <p>Scf</p> <p>5Pm18 4 7Qs biP</p> <p>cMQI wiKmsYY</p> <p>YT6GWH YeIJ</p>
--	--

Figure 3: ...becomes entirely legible when rendered by an HTML engine.

'right-to-left override' feature of Unicode to reverse the order of letters bracketed by special codes. This hides the offending word or phrase from the filter (which sees, for example, 'argaiV'), but the user sees the letters in the correct order thanks to the Unicode-compliant HTML engine.

But whatever methods spammers use to disguise their true message, most types of spam have an Achilles' heel: the URL-based call to action. In the majority of English spam a profit can only be made if there's a link to click. Attacking the URLs contained in spam emails is a very effective technique (*Sophos's* products block over 80% of spam using URL blocklists alone), and thus spammers have developed ways to attempt to circumvent this approach too.

One method that saw a huge resurgence in the last year (see Figure 4) is the use of 'freeweb' hosting providers to host pages that redirect (often via 'encrypted' Javascript) to the spammers' main sites. Again, this is not a new trick, but now it is common to see one spam campaign use thousands of randomized 'freeweb' URLs, rendering URL blocklists far less useful. Naturally, these providers have taken precautions against such abuse of their systems, for the most part by requiring the passing of a CAPTCHA test while signing up for an account, which involves presenting an image containing heavily obfuscated letters and numbers and asking the user to enter the characters into a form.

A recent study, though, shows that when targeted specifically by an attacker most CAPTCHA systems can be solved more often and more accurately by a computer than by a human! Due to the sheer volume of unique freeweb URLs seen in modern spam it seems likely that spammers have cracked the CAPTCHAs used by large freeweb providers such as *Yahoo!* *Geocities*, and have automated systems to register such sites in mass quantities. Of course, they and other freeweb

providers are constantly striving to eliminate this kind of abuse of their services, and with the significant amount of research currently being conducted into improving CAPTCHA technologies it is hoped this particular 'marketing tool' will one day be denied to spammers permanently.

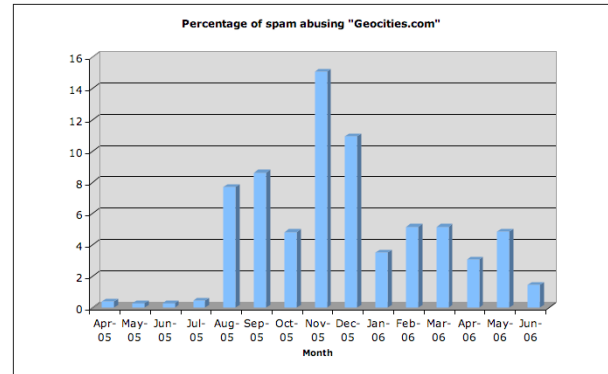


Figure 4: Spam abusing geocities.com peaked in volume in November 2005.

While some spammers turned to freeweb providers to circumvent URL blocklists, the majority continued to register their own domains, knowing that during the delay between spam containing a given domain name first appearing on spam traps and that domain being added to blocklists they could (with their botnets) dispatch hundreds of thousands, if not millions, of emails.

But anti-spammers responded with ingenuity once again, this time by monitoring WHOIS and related information for new domain registrations and comparing the name servers and other information against their databases of known spammers. In this way it became possible in many cases to add domains to blocklists before any spam had even been sent. Naturally, it wasn't long before some spammers found ways around this as well, and when registering their domains some will specify well-known (and trusted) freeweb name servers in the registration form, only to switch to using their own name servers just minutes before beginning their spam run.

Another similar technique is to use a new name server for each domain registered, preventing analysis of WHOIS information linking the domain with known-bad spammy name servers until it's too late. Yet another common trick is to unleash their spam run moments after registering a new domain, thereby reducing the risk of the domain being blocked proactively. The spammy domain 'fyefga.org', for example, was created at 01:28 UTC on February 16 2006. The first email making use of this domain appeared on *SophosLabs'* spam traps a mere two minutes later.

As far as content filtering goes, the most notable development in recent months has been the huge increase in the volume of image spam – rasterized text in an image, usually in GIF format, attached to the spam email – which has increased on *SophosLabs'* spam traps more than twofold over the first half of the year (see Figure 5). This approach is by no means new (Figure 6 shows an early Russian image spam, from 2004), but in the last year or so became for the first time economically viable for spammers. The surge in availability and popularity of consumer-level broadband connections means that using botnets to send larger amounts of data has become feasible, and even for those spammers who actually

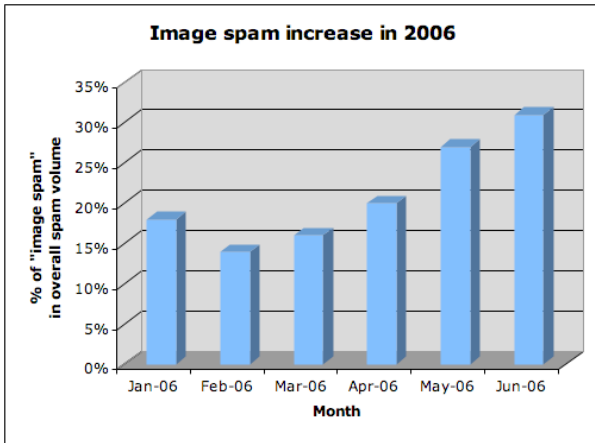


Figure 5: Image spam volume has increased more than twofold since the beginning of 2006.

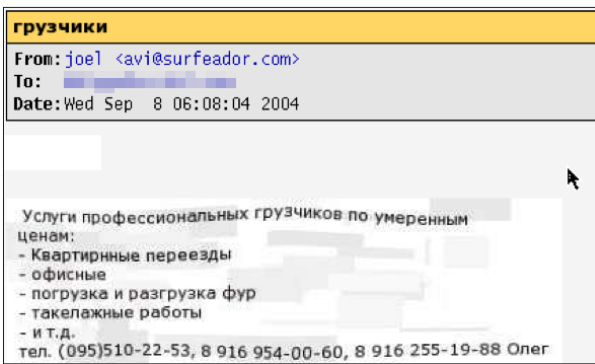


Figure 6: An early example of Russian image spam, from September 2004.

pay for their own bandwidth, the associated costs have reduced so dramatically that sending out millions upon millions of images is no longer prohibitively expensive.

Image spam can be used in most, if not all, of the areas traditionally occupying spammers' efforts, but seems particularly well-suited to campaigns requiring no call to action. Image spam is commonly employed in so-called 'pump 'n' dump' schemes, in which the stock of a company is hyped in fake investment newsletters in an attempt to fool the unwary into buying shares, thus pumping up the price. The spammers or their employers then sell ('dump') at the higher value all the shares they hold, theoretically making a profit. The prevalence of this type of spam has exploded in the last nine months, on some days comprising up to 40% of the spam seen on *SophosLabs'* spam traps. Image spam is also often used where the call to action is a phone number for the victim to call, such as in spam touting online degrees, and in a large proportion of non-English spam.

Image spam is arguably the ultimate in text obfuscation: spammers can say whatever they want without fear of triggering even the most sophisticated ASCII-based text filters. And straightforward hashes over the attachment body are prevented by (thus far) simple randomization of the image content, such as changing the compression level, adding faint dots in random locations within the image (Figure 7), rotating the image slightly in either direction, offsetting the actual content of the image within the frame around it, randomly changing font styles, sizes and colours, randomly chopping

up the image and reassembling with HTML (Figure 8), and so on. There are far more ways to obfuscate image spam than text spam, and given the range of image effects available even in consumer-level image processing tools it is clear that the possible combinations are as good as infinite, with only little impact on the readability of the text.

There are numerous other challenges that must be surmounted in order to recognize and thus block image spam. First, the email often looks, at a source-code level, identical to a legitimate email containing only an attached image. In fact, a large portion of the image spam we analyse seems to have been created by first composing the email with a dummy image attached in *Outlook Express* or other popular mail user agent, then simply replacing the attachment with a randomly altered image and providing a random subject line, each time the message is mailed. This means the headers, the MIME structure and the enclosed HTML are entirely consistent with legitimate emails, and so there are no spam signs upon which to base detection other than the image itself and the IP address from which it originated.

A seemingly promising approach to the problem is, of course, employing optical character recognition to turn the rasterized text back into ASCII so it could then be scanned with existing text-based technologies. While theoretically appealing, this is unlikely to be a sustainable approach in practice. Though OCR technology has advanced a great deal in recent years the main focus of development has been on improving

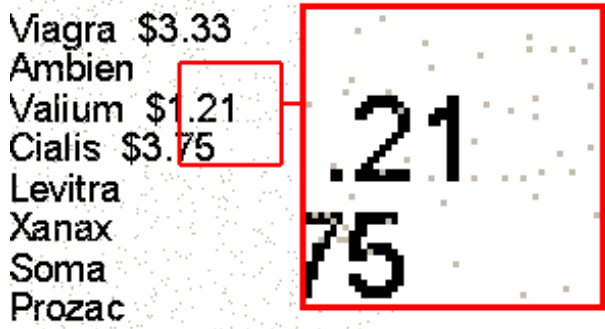


Figure 7: Random 'hashbuster' speckles added to the background of this pills spam.

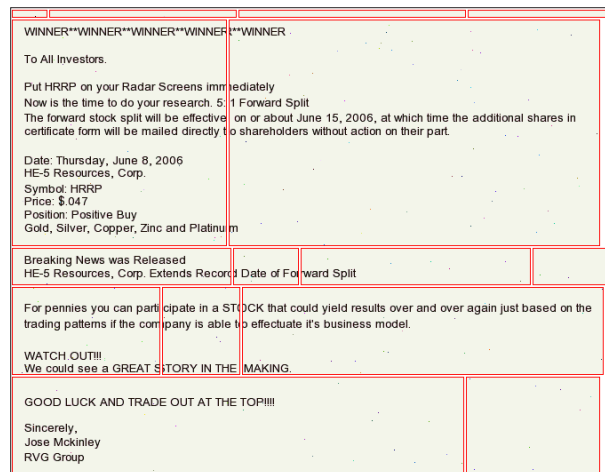


Figure 8: Random segmentation is another means of image 'hashbusting'.

recognition of stable and reasonable inputs, such as printed material and handwriting. These inputs are designed to be readable (by humans, at least) and more or less consistent, and typeface designers have significant incentive to make their creations more accessible to OCR software.

For spammers, on the other hand, the incentive is precisely the opposite. The moment anti-spam filters begin employing OCR to pre-process image spam (and a *SpamAssassin* plug-in already exists to do just that, though it's still in a fairly early stage of development at the time of writing), spammers will begin to manipulate their images in such a way as to make this harder to do, i.e. by further obfuscating the content. Given the myriad ways in which this is possible, and given the sensitivity of current OCR technology to unexpected input, it is difficult to envisage this approach being sufficiently reliable to justify the research and development investments required.

Even if a full OCR-based analysis of an image proves impractical, there are a variety of other, less fragile approaches that should be considered. A great deal of information can be easily and quickly extracted from image headers, for example, that can provide valuable clues as to the 'spamminess' of the image in question.

Perhaps the most valuable of these is the compression level of the image, which can be expressed as the number of bytes required to represent all the pixels present. Generally speaking, the more complex the image in terms of texture, the less compressible it is, whereas images with large areas of very similar colours tend to compress well. Since the great majority of the spam images currently consist of text on a plain background, they exhibit a significantly higher compression level than 'normal' images sent through email (Figure 9), which more often than not are texturally complex photographs or drawings. This can be a very good indicator of the spamminess of an image.

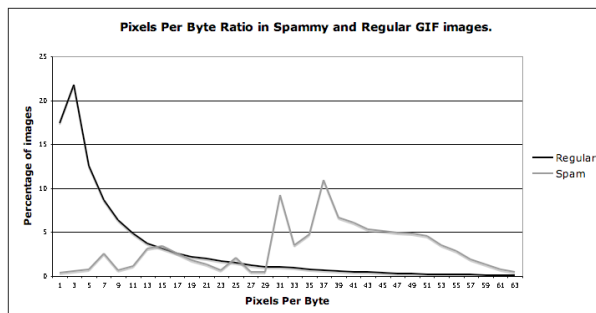


Figure 9: Compression ratios can provide a good indicator of a GIF's 'spamminess'.

If it is judged feasible to decompress the entire image (rather than just extracting the metadata) for further analysis, then another promising technique is to produce a histogram of the unique colours used within the image. Again, normal images tend to exhibit a large number of unique colours, and their frequency distribution is relatively smooth. Spam images consisting of text on a flat background, by contrast, contain few colours, one of which is seen far more frequently than any other, and thus their histograms are often dramatically different from normal images.

Once the image is decompressed it becomes possible to perform some of the classic image processing manipulations, such as converting it from the spatial domain to the frequency

domain with a Fourier transform. With such processing it may well be possible to differentiate between normal – especially photographic – images (with relatively little very-high-frequency information) and rasterized text images (with a predominance of very-high-frequency information due to the rapid contrast changes where text is present) with a reasonably high degree of accuracy. Converting to the frequency domain before analysis also makes the algorithm less sensitive to such obfuscations as random rotations and faint random speckles added to the background of the image.

These and many more image processing techniques may prove valuable in anti-spammers' efforts to remain standing in the latest round of this decades-old competition.