Blog Spam: A Review

Adam Thomason
Six Apart Ltd.
San Francisco, CA 94107
athomason@sixapart.com

Abstract

Blogs are becoming an increasingly popular target for spammers. The existence of multiple vectors for spam injection, the potential of reaching many eyeballs with a single spam, and limited deployment of anti-spam technologies has led to a sustained increase in the volume and sophistication of attacks. This paper reviews the current state of spam in the blogosphere at large and in particular as seen at TypePad, a major hosted blog service. Furthermore the effectiveness of two popular open-source email antispam programs at classifying blog comment spam is evaluated.

1 Vectors

Blogging tools provide a variety of ways to incorporate text on a blog and all of these are exploited by spammers. The first vector is the *blog post*, which may only be created by the blog owner or authorized delegates. These posts generally structure the narrative of the blog. Second is the *comment*, which is attached to a particular post and is generally only visible from the page dedicated to the post. Third is the *trackback*[1], which is a server-to-server notification that a post on one blog references one on another. The linked-to blog may then elect to include a reciprocal link to the tracking post.

1.1 Splogs

Since ownership is required to create a top-level blog post, spam posts are necessarily found on spammer-created blogs, deemed *splogs*. Splogs are found primarily on hosted blog platforms for several reasons: to entice users familiar with the service to visit the splog; to exploit search-engine reputation of the hosted service; and to attract traffic from "neighboring" blogs. Additionally, free hosting services are the primary target for splogs due to the minimal cost of establishing one. A recent study (Weinberg 2007) estimates that 75% of blogs on Google's free BlogSpot service are spam. Conversely, paid hosting services such as Six Apart's TypePad have negligible splog content (Ishenko 2007). Free social networking sites (such as LiveJournal, Vox, MySpace, and Friendster) which incorporate blogging tools are also targeted by spammers, although more involved registration processes and faster abuse reporting mitigate the problem to some extent. Across the blogosphere at large, a study in February 2007 found that 56% of blogs which sent update notifications to the weblogs.com[2] ping server were splogs (Kolari 2007).

1.2 Comment Spam

Unlike splogs, comment spam has been targeted to all types of blogs which allow commenting. The popular Akismet blog spam classification service presently handles approximately 10,000,000 comments per day, double the rate from 75 days prior (Akismet.com 2007). Akismet classifies 95% of submitted messages as spam.

1.3 Trackback Spam

While the HTML internals of comment submission forms may be changed to confuse spam robots without affecting legitimate users, trackbacks are transmitted by an HTTP-based protocol with a fixed API. The trackback specification makes no mention of verification, allowing spammers to inject arbitrary URLs into a trackback ping message along with camouflaging text of the spammer's choosing. This has led to an abundance of trackback spam targeted at supporting blog software.

2 Blog Spam Considerations

While blog spam and email spam are similar in many ways, there exist a number of differences in both the attack and defense profiles.

Blog spammers have numerous advantages over email spammers. First, a single spam can potentially reach as

[1] Including similar, more recent protocols such as *pingback*, *linkback, and refback*.

[2] A centralized server which a blog may notify as soon as new posts are made; other services then query the ping server rather than each individual blog to check for new posts.

many viewers as there are readers of the blog post to which the spam is attached. Second, spam can be disguised by incorporating text related to the accompanied post, either by naïve copy-and-paste or more advanced natural language processing. Similarly, as the spammer has complete access to the target's "inbox", spam can also be targeted to blogs related to the product being offered. Anecdotal review of TypePad spam suggests that trackback spammers are presently using this technique more than comment spammers, for instance posting spam about car insurance to automotive blogs. Lastly, spammers can immediately detect when they have evaded a filter since the spam appears on a public webpage.

Bloggers and hosting services also have tools not available to email administrators. In the realm of comment spam, there is no fixed interface for posting a comment as there is SMTP for email transmission. Each time a legitimate commenter wishes to post, her browser must retrieve a page containing the web form which effectively encodes the correct way to post. Modification of this interface presents difficulties for simplistic robots which either give up or continue to submit malformed requests until a human can review the new interface. The content that a spam comment or trackback can include is also controlled by the blog host. Image spamming, an increasing problem in email, is not possible on the majority of blogs where HTML `<img>` tags are filtered from comments and trackbacks[3].

On the flipside, certain technologies useful for email anti-spam are not applicable to blog spam. SPF and similar systems which attempt to authenticate senders based on SMTP server addresses fail to translate as any network host may be a legitimate source of comments for services where commenter authentication is not performed.

As an additional challenge, blog hosts are particularly sensitive to classification latency, as comment submitters expect immediate feedback regarding the status of their contribution.

3 TypePad Spam

All incoming comments and trackback pings to the TypePad service were collected over a period of three months (1/02-4/02)[4]. A variety of filters produced a spam score for each message; this was reduced to a binary spam/ham decision for reporting.

3.1 Comment Spam

During the collection period, 21.7% of comments were accepted as ham, and 78.3% were either rejected or challenged. Both ham and spam have a highly periodic nature, with traffic dipping during weekends.

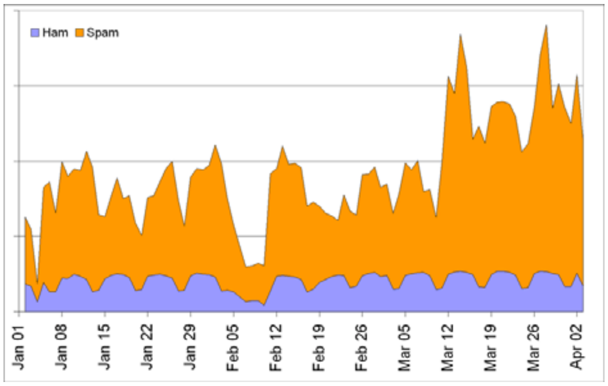

Figure 1: TypePad Comment Volume

3.2 Corpus Development

While some efforts have been made to classify spam on the web (Webb 2006), no up-to-date public corpus exists for blog spam. To establish a reliable set of baseline data, employees of Six Apart manually rated a set of 6159 comments as spam or ham. Of these, 3119 were classified as spam and 3040 as ham. Since human accuracy is limited to no less than 1/1000 errors (Yerazunis 2003), it is expected that some classification errors will exist in the corpus (as observed below).

80% of the corpus was randomly selected as the *training* set, while the remaining 20% constitutes the *validation* set.

3.2.1 Spam Categories

To determine a snapshot of what is being advertised in comment spam, a set of diagnostic tokens was created to categorize each spam comment in the training corpus. Distinguishing tokens were iteratively extracted from unclassified comments until each matched at least one token. Comments were then assigned to the categories in Table 1 based on manual review of the tokens. Each comment belonged to an average of 1.87 categories.

[3] Though not an issue presently, splogs may eventually incorporate image spam as blog owners have much more leeway with allowed HTML.

[4] Measurement error over a period of days accounts for the drop in volume observed near Feb 7.

Table 1: Comment Spam Categories

Pornography	974	Cellular phones/service	52
Pharmaceuticals	694	Lottery	43
Non-English language spam	576	Event tickets	39
Travel	575	Books	32
Credit offers	231	Games	31
Movies	180	Hardware	28
Product knockoffs	165	Restaurants	20
Health	164	Real estate	18
Gambling	133	Home improvement	17
Automobiles	109	Alcohol	16
Insurance	104	Academic degrees	16
Cell phone ringtones	82	Mail Order Brides	15
No description included	73	Dating	11
Search	65	Tattoo	6
Music	63	Hacking	2
Software	62	Academic cheating	1

4 Statistical Filtering

To test the effectiveness of statistical filtering algorithms on the blog spam corpus, two open-source email filtering packages, DSPAM (Zdziarski 2007) and CRM114 (Yerazunis 2007), were selected. An RFC2822 email message was constructed for each comment using all available information to fill in headers. The filters were deliberately left in near-as-possible default configurations. As trackbacks are not required to include descriptive text, this test was restricted to comments only.

4.1 CRM114

CRM114 version 20070301-BlameBaltar was used with the mailreaver filter in a near-default configuration. The classifier flag set was `osb unique microgroom`, with a single-sided thick-threshold of 10.0. The .css data files were established by passing the training fraction of the corpus through mailtrainer.crm (TUNE, 10 cycles). The validation portion was then classified. Though CRM114 informs the user that classifications with a pR score (Figure 2) of magnitude less than 10 are not definitive, a binary decision was nevertheless required for the purposes of this test.

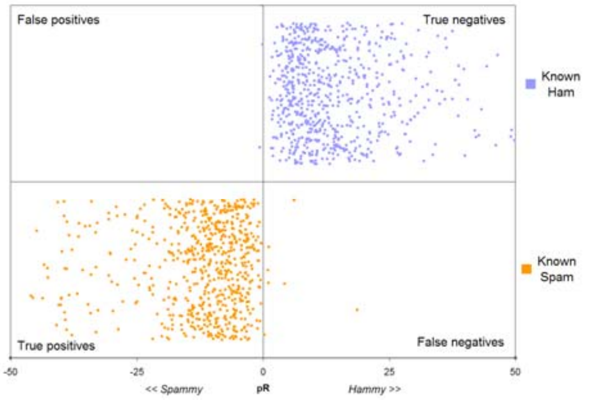

Figure 2: CRM114 Performance

Of 660 spam messages, 7 were misclassified as ham. Of 685 ham messages, 2 were misclassified as spam. Three of the false negatives and one false positive were manually determined to be spurious (corpus classification errors). Of the true errors, all false negatives had pR values ≤6.17, and the single false positive had a pR value of just -0.66.

4.2 DSPAM

DSPAM version 3.8.0 was used in train-on-error mode using the Fisher-Robinson chi-square algorithm and chain (bi-gram) tokenizer with whitelisting disabled. The dspam-train.pl script was run 6 times on the training set, at which point no more classification errors were made. DSPAM reported a chi-square confidence value with each classification (Figure 3).

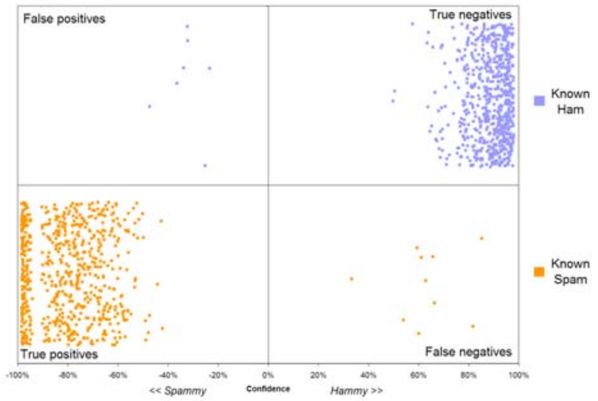

Figure 3: DSPAM Performance

To simplify comparison with CRM114 results, a pR score was computed as $\log_{10}(P_{ham})$-$\log_{10}(P_{spam})$, where P_x is the probability of being either spam or ham as computed by the chi-square algorithm[5] (Figure 4).

[5] Done for visual comparison only; this does not affect classification.

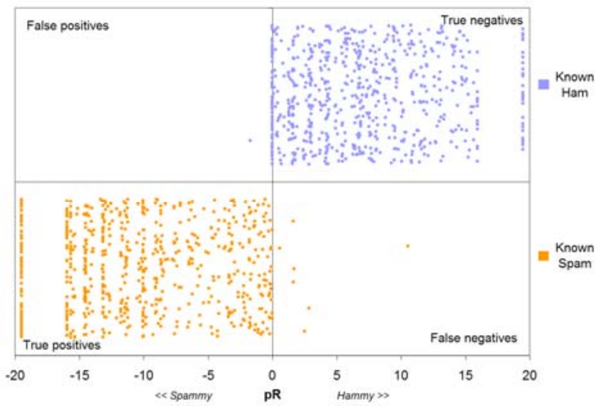

Figure 4: DSPAM Performance (pR)

Of 660 spam messages, 10 were misclassified as ham (including three corpus errors). Of 685 ham messages, 7 were misclassified as spam (including one corpus error). Three of the false negatives and one false positive were spurious (corpus classification errors). Of the true errors, all false negatives had confidence values ≤66%, and all false positives had confidence values ≤36%.

5 Conclusions

Blog spam is a significant and increasing problem. However, the above results show that even without tuning parameters, statistical anti-spam solutions developed for email are effective in detecting blog comment spam.

References

Akismet.com (2007). *Akismet Spam Zeitgeist*. http://akismet.com/stats/.

O. Ishenko (2007). *How Much Blog Spam? A Study of a Ping Dataset.* http://www.seoresearcher.com/how-much-blog-spam-a-study-of-a-ping-dataset.htm.

P. Kolari (2007). *Pings, Spings, Splogs and the Splogosphere: 2007 Updates.* http://ebiquity.umbc.edu/blogger/2007/02/01/pings-spings-splogs-and-the-splogosphere-2007-updates/.

S. Webb, J. Caverlee, C. Pu (2006). *Introducing the Webb Spam Corpus: Using Email Spam to Identify Web Spam Automatically*. http://www.ceas.cc/2006/6.pdf.

T. Weinberg (2007). *75% of Google's Blogspot Blogs are Spam*. http://www.seroundtable.com/archives/012778.html.

W. Yerazunis (2003). *Sparse Binary Polynomial Hashing and the CRM114 Discriminator*. http://crm114. sourceforge.net/CRM114_paper.pdf

W. Yerazunis (2007). *CRM114 - the Controllable Regex Mutilator*. http://crm114.sourceforge.net/.

J. Zdziarski (2007). *The DSPAM Project*. http://dspam.nuclearelephant.com/