Proliferation and Detection of Blog Spam

A study of more than one million blog comments during the last two weeks of June 2009 showed more than 75 percent of them to be spam.

SAEED ABU-NIMEH
Websense Security Labs

THOMAS M. CHEN
Swansea University

Web 2.0 sites generally revolve around collaborative authoring such as blogs, wikis, discussion boards, and forums. Unfortunately, the popularity and ease of posting comments to blogs have made them attractive vectors for luring visitors to spammers' websites. Blog spam works differently from conventional email spam in that its purpose isn't necessarily to get visitors to click a hyperlink in the spam. Instead, blog links are intended to increase the spammer's website ranking among search engines, attracting more visitors to the site by listing it higher in search results.[1] Blog spam (or "link spam"[2]) on authentic blogs is distinguished from spam blogs or "splogs," which are artificial blogs with fake content created solely to host ads or increase the search-engine rankings of spammer websites.[3]

Blogs have been spam targets for a few years, and blogging software has incorporated various means to discourage it. Many websites use CAPTCHA, a display of alphanumeric text embedded in an image. Visitors must copy the text before the site will accept comments in a form. CAPTCHA distorts the text or otherwise adds noise to challenge a spam tool's character-recognition features. A drawback of this method is that it can annoy and discourage visitors who want to post legitimate comments.

Another means of discouraging blog spam is the `rel="nofollow"` attribute for hyperlinks. Most blog software automatically defaults to give this attribute to any links that get posted. The popular search engines don't count hyperlinks with this attribute toward the link target's rank. However, the attribute doesn't prevent a victim from clicking the hyperlink and going to the spammer's site.

LinkSleeve, Akismet, and Defensio are methods for preventing spam that depend on collaborative techniques. They send comments first to a central server that performs tests to determine whether the comments are spam. Centralized servers have the advantage of seeing URLs that appear on multiple sites—a key characteristic of spam. This collaborative-detection method can effectively prevent link spam, but comment spam that's designed to blend into an ongoing blog discussion is more challenging to detect.

Despite preventive measures, one recent study estimated that 85 percent of blog comments are inserted by automated bots.[4] Traditional spam filters aren't very effective against blog spam. For instance, the rich features that blogs typically allow make it easy for spammers to launch cross-site scripting or drive-by download attacks. Coping with such features requires more than a spam filter.

To clarify blog spam's characteristics, we analyzed two weeks of it using a classifier based on support vector machines (SVMs) enhanced with heuristic rules. We present our experimental setup and study results here. For related work on classification of blog spam and splogs, see the sidebar.

Characteristics of Blog Spam

Web spam appears in different forms:

- *Comment spam*—unsolicited posts in editable Web pages such as blogs, wikis, and guestbooks for the purpose of corrupting the authentic meaning of community-provided feedback.[2]

- *Term spam*—extraneous words inserted in spam pages to make them seem more relevant to some search queries or popular keywords.
- *Link spam*—unsolicited posts containing URLs to increase the number of links pointing toward a spammer's site, thereby increasing the page's rank in search engines.
- *Spam pages*—entirely fake webpages created solely to mislead a search engine.[1] Each fake page receives a minimum guaranteed PageRank value, and the ac-

Related Work in Classifying Blog Spam

Previous studies have proposed classifying blog spam by examining page contents, hyperlink structures, or both. Marco Ramilli and Marco Prandini proposed content analysis to detect comment spam.[1] Their method consists of a self-learning filter that remembers every posted sentence and associates a score to each message according to the number of already-seen sentences in it. If the score is above a given threshold, the message is classified as comment spam.

Na Dai and his colleagues investigated a content-analysis method that measured changes in a website's content over time.[2] Their work presumes that spam drastically increases the fraction of popular words on a site. If a page has a sudden increase in popular words over a short time interval, they classify it as spam.

Along a similar idea, Yu-ru Lin and his colleagues proposed detecting splogs by temporal characteristics as well as content.[3] To make splogs appear relevant to blog search engines, their content is updated frequently using automated frameworks. Lin's method captures blog temporal characteristics in self-similarity matrices and detects splogs through regularity and joint features across different attributes.

Dennis Fetterly and his colleagues found distinct statistical properties to use in detecting spam pages.[4] These properties involve the URL's host name; in-degrees and out-degrees in the graphs formed by webpages and the hyperlinks between them; webpage rates of change on a site; and the number of similar pages. Several heuristics detect spam on the basis of these statistical properties. Alexandros Ntoulas and his colleagues explored additional content-analysis heuristics.[5]

Several researchers have used support vector machines (SVMs) for text classification, including spam filtering, because of their effectiveness and relative efficiency.[6] In comparison studies for other classification problems, SVMs have performed well in terms of classification error and mean-square error.[7] For these reasons, several studies have evaluated SVMs for blog-spam classification.

Pranam Kolari has used SVMs with his colleagues for splog detection.[8,9] In addition to the usual bag-of-words features, they introduced new features: bag of anchors (the anchor text of all URLs), bag of URLs (all tokens created by splitting URLs at "/", ".", "?", and "="), and bag of *n*-grams (*n*-character–long text tokens). They selected significant features on the basis of mutual information.

Taichi Katayama and his colleagues also used SVMs for splog detection, which they tested on Japanese blogs.[10] Their chosen features included whitelisted or blacklisted URLs, noun phrases, noun phrases in anchor texts, link out-degrees, maximum number of outlinks from a blog homepage to any single URL, and number of mutual links to any other blogs.

D. Sculley and Gabriel Wachman evaluated SVM effectiveness for blog comment spam.[11] Although they found it to be accurate, they pointed to the training time as a prohibitive cost for large-scale spam detection. They proposed a computationally less-complex SVM called relaxed-online SVM, which they demonstrated experimentally to perform equally well for blog spam.

References

1. M. Ramilli and M. Prandini, "Comment Spam Injection Made Easy," *Proc. 6th IEEE Consumer Comm. and Networking Conf.*, IEEE Press, 2009, pp. 1–5.
2. N. Dai, B. Davison, and X. Qi, "Looking into the Past to Better Classify Web Spam," *Proc. 5th ACM Int'l Workshop Adversarial Information Retrieval on the Web* (AIRWeb 09), ACM Press, 2009, pp. 1–8.
3. Y-R. Lin et al., "Detecting Splogs via Temporal Dynamics Using Self-Similarity Analysis," *ACM Trans. Web*, vol. 2, no. 1, 2008, pp. 1–35.
4. D. Fetterly, M. Manasse, and M. Najork, "Spam, Damn Spam, and Statistics," *Proc. 7th ACM Int'l Workshop Web and Databases*, ACM Press, 2004, pp. 1–6.
5. A. Ntoulas et al., "Detecting Spam Web Pages through Content Analysis," *Proc. 15th ACM Int'l Conf. World Wide Web*, ACM Press, 2006, pp. 83–92.
6. L. Zhang, J. Zhu, and T. Yao, "An Evaluation of Statistical Spam Filtering Techniques," *ACM Trans. Asian Language Information Processing*, vol. 3, no. 4, 2004, pp. 243–269.
7. D. Meyer, F. Leish, and K. Hornik, "The Support Vector Machine under Test," *Neurocomputing*, vol. 55, 2003, pp. 169–186.
8. P. Kolari, A. Java, and T. Finin, "Characterizing the Splogosphere," *Proc. 3rd Ann. Workshop Weblogging Ecosystem: Aggregation, Analysis and Dynamics* (WWW 06), Univ. Maryland, 2006; http://ebiquity.umbc.edu/paper/html/id/299/Characterizing-the-Splogosphere.
9. P. Kolari, T. Finin, and A. Joshi, "SVMs for the Blogosphere: Blog Identification and Splog Detection," *Proc. AAAI Spring Symp. Computational Approaches to Analyzing Weblogs*, Am. Assoc. Artificial Intelligence, 2006, pp. 92–99.
10. T. Katayama et al., "An Empirical Study on Selective Sampling in Active Learning for Splog Detection," *Proc. 5th ACM Int'l Workshop Adversarial Information Retrieval on the Web* (AIRWeb 09), ACM Press, 2009, pp. 29–36.
11. D. Sculley and G. Wachman, "Relaxed Online SVMs for Spam Filtering," *Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval* (SIGIR 07), ACM Press, 2007, pp. 415–422.

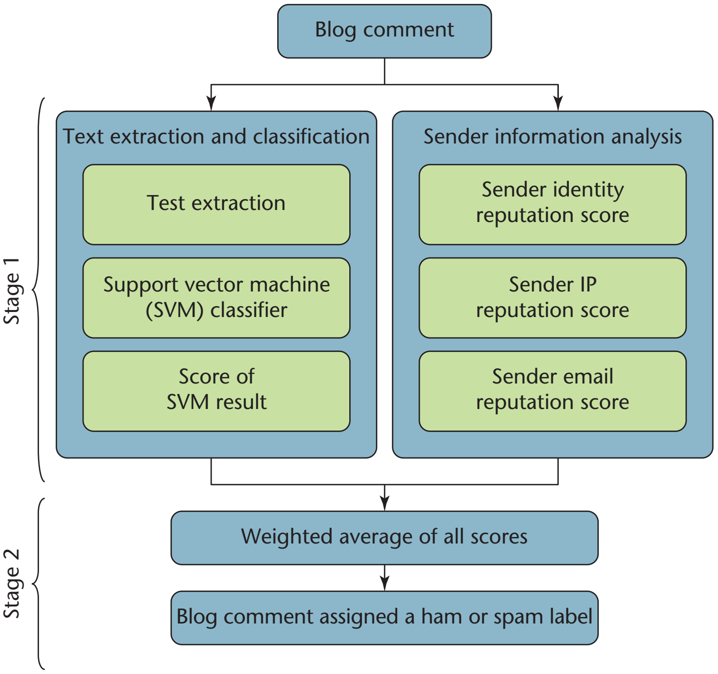
Blog comment
Stage 1
Text extraction and classification
Test extraction
Support vector machine (SVM) classifier
Score of SVM result
Sender information analysis
Sender identity reputation score
Sender IP reputation score
Sender email reputation score
Stage 2
Weighted average of all scores
Blog comment assigned a ham or spam label

Figure 1. High-level overview of approach to detecting blog spam. The first stage analyzes and scores the text and sender information of each blog comment in parallel. The second stage assigns a weighted average to those scores.

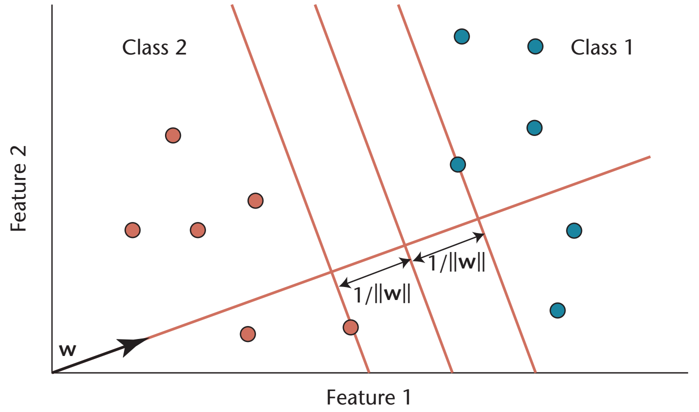

Figure 2. Support vector machine (SVM). In this example, the SVM has two classes and two features.

cumulation of endorsements from many spam pages can substantially raise the PageRank for a target page.

- *Splogs or spam blogs*—entirely fake blogs created solely to attract search engines and influence them to promote a spammer's site.[3]

Comment spam is intended to alter the perceived value of a product under discussion, alter the reputation of community members, or divert the audience's interest to other products.[2] It's more challenging to detect than link spam, which more closely resembles traditional email spam, but it exhibits some identifying traits: comments often unrelated to the blog topic, a repetition of the same words in similar patterns, a large number of anchor texts, and a high concentration of noun phrases.[5]

Spam pages have revealed some unusual statistical properties.[1] For instance, their host names tend to be longer than 45 characters and to have more than six dots, five dashes, or 10 digits. Their out-degrees and in-degrees are unusually high, and the pages exhibit a high average rate of change—almost all pages change completely every week. In addition, many spam pages appear to be very similar to each other.

Spam-page content has also exhibited some distinctive statistics, such as a large number of extraneous words, keywords stuffed in page titles, frequent use of composite words (that is, multiple concatenated words), a high percentage of text used for anchor text, a high percentage of visible content, and content replicated multiple times.[6]

Spammers increase splog visibility using tricks such as keyword stuffing or content duplication so that search engines will index them with a high rank in a particular topic.[3] Splogs are characterized by machine-generated content, useless or duplicated content (copied from other blogs), commercial intent, and highly dynamic content.

Our Detection Approach

Figure 1 depicts a high-level overview of the two-stage approach we used in our experiments to detect blog spam.

In the first stage, we extracted the comment text and ran it against an SVM classifier that assigned it a score, designating its class as spam or ham (that is, legitimate comments). In parallel, we used heuristic rules to examine the sender information and assign a reputation score to the sender's identity, IP address, and email.

In the second stage, we calculated a final overall score as a weighted average of the first-stage scores.

The SVM Classification

SVMs belong to the general class of supervised linear-discrimination methods.[7] In classifying blog comment text, we treated it as a "bag of words," where features were common spam words. Suppose we had n features of interest. We first trained the SVM with N data points, each of which is already classified as blog spam (class 1) or nonspam (class 2).

We can visualize the data points as data vectors in n-dimensional space, as shown for two classes in Figure 2. The SVM classifier divides the space into regions and locates a new data point according to its regional classification. SVMs assume that the decision boundary has the form of an $(n - 1)$-dimensional hyperplane.

In the example, the decision boundary is a line.

The margin is the distance from the hyperplane to the data points in either class that are closest to it. The SVM classifier seeks the hyperplane that optimally maximizes the margin. As Figure 2 shows, the problem comes down to finding a vector **w** that's normal to the hyperplane minimizing $\|\mathbf{w}\|^2/2$. In other words, it's the solution to a standard quadratic optimization problem.[7]

The Heuristic Rules

To improve the SVM classifier's accuracy, we combined it with several heuristics and decided whether to classify a blog comment as spam by weighing the classifer and heuristics results in a final score.

We based our heuristics on the reputation of the comment author's IP address, email address, and identity. We assigned a weight w_1 to the classifier's decision. Similarly, we assigned weights w_2, w_3, and w_4 to the author's IP reputation, email reputation, and identity reputation, respectively. To find the optimal weights, we conducted several experiments and found the weights that maximized accuracy and minimized false positives. We calculated the total score for the blog comment as the summation of the weighted classifier decision and the weighted reputation scores. If the total score exceeded a chosen threshold (t), the blog comment was classified as spam.

We calculated the reputation score for these heuristics by finding the spam-to-ham ratio in each case. For example, to calculate the reputation score for a specific IP address, we calculated the total number of ham comments posted from that IP address and divided it by the total number of comments, whether ham or spam, posted from that IP address. This assigns each IP address a reputation score between 0 and 100 percent, with 0 percent being the worst score. We followed the same approach to calculate the email and identity reputations.

Experimental Setup

In our experiments, we extracted the comment text from the blog comments and then built a bag-of-words dictionary for the terms frequently found in spam.

We used the term-frequency/inverse-document-frequency (TF/IDF) method to find the terms used most frequently in the comments. TF/IDF assigns a higher weight to terms that appear often in a single comment but don't appear in many comments. We didn't perform stemming on terms, but we did remove all stopwords. We didn't analyze URLs. We used the SVM classifier only to classify spam terms and the heuristic rules only to improve overall detection accuracy.

We trained the classifier offline and then classified new-arriving blog comments in real time. We trained the classifier on 884 blog comments, using a linear kernel and cost $c = 16$. The cost parameter c is the cost-of-constraints violation, which is the constant of the regularization term in the Lagrange formulation. The total number of support vectors in this run was 1,091. We optimized all the input parameters to achieve the minimum false-positive rate.

Table 1. Performance of the support vector machine classifier during the training phase.

Measure	Value
Total number of comments	884
Number of hits	837
Number of misses	47
Accuracy	0.947
False positives	24
False negatives	23
Average runtime (seconds)	0.073

Sixty percent of the training comments were legitimate, and 40 percent were spam. Table 1 summarizes the accuracy, false positives, false negatives, and average runtime on the 884 comments.

For the validation set, we constructed a completely different dataset. We collected 8,724,994 blog posts between 1 April and 1 June 2009 and tested the performance of the trained model on them. Because we used an entirely different dataset from the one we used in training, we didn't perform cross-validation during the testing phase.

We obtained the training and validation sets from proprietary Defensio (www.defensio.com) logs. Defensio is a service that detects malware in blogs and Web 2.0 applications. It had collected the comments in these sets from various comments posted across several blogs. Defensio software had determined the comment labels—that is, ham or spam. Assuming the initial Defensio classification had caused false positives or false negatives, the service users would have flagged the comments in question for reclassification. Therefore, we assumed the current labels in the training and validation sets to be accurate.

Using merely the SVM classifier, we achieved an accuracy of roughly 95 percent (see Table 1). However, when we combined the classifier with the heuristic rules, the accuracy increased to almost 99 percent. Figure 3 depicts the accuracy of the SVM classifier combined with the heuristics between April and June 2009.

Experimental Results: Mostly Spam

With the SVM classifier enhanced with heuristic rules, we carried out an experiment to measure the prevalence of blog spam in a corpus of 1,048,567 blog posts collected between 18 June and 30 June 2009. Our experiments showed that more than 75 percent

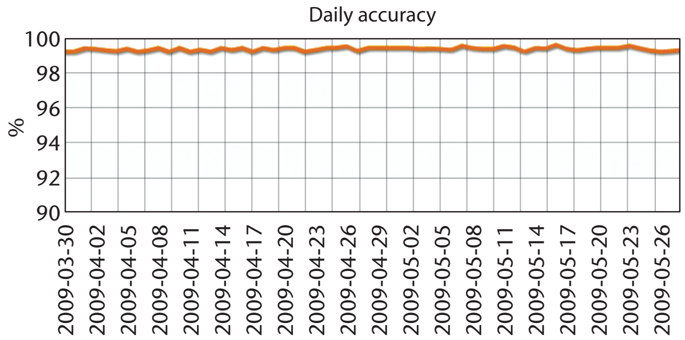

Figure 3. SVM and heuristics combined classification results between April and June 2009. The combined method identified blog spam with almost 99 percent accuracy over the three-month test period.

Table 2. Top 10 offender IP addresses.

IP address	Total posts	Spam (%)
66.232.97.145	110,186	99
96.31.68.140	103,386	99
69.46.23.47	92,961	99
69.46.16.14	60,360	99
94.102.49.76	44,118	94
78.159.112.178	28,662	99
194.8.75.163	15,291	100
194.8.75.149	15,253	100
194.8.74.220	13,828	100
67.215.237.98	12,396	98

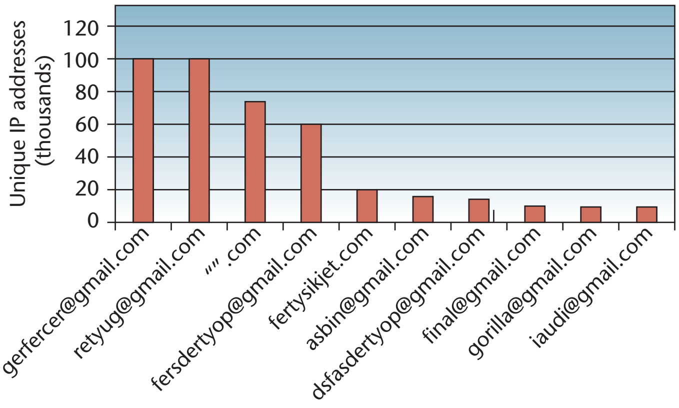

Figure 4. Unique IP addresses used per email address over the test period from 18 to 30 June 2009. The empty quotation marks in the figure correspond to comment authors with no email address.

of the comments were spam.

Traditional spammers employed evasion techniques to circumvent detection. One basic method was to change their IP addresses, using proxies to avoid getting their IP addresses blacklisted by ISPs or webmasters. Blog spammers are using this same technique. Our experiments show that spammers used more than 100,000 different IP addresses to post spam comments from one email address during two weeks. Table 2 summarizes the top 10 offender IP addresses, the total number of posted comments, and the percentage of spam comments.

Figure 4 graphs the number of unique IP addresses used by top the 10 email addresses for comment authors.

Ham and spam comments were posted from more than 6,000 unique autonomous systems numbers (ASNs). An ASN is a group of IP addresses that have the same routing policy. ISPs and large institutions can own their own ASN. Figure 5 graphs the number of spam posts for the top five offender ASNs. ASN 29802 was responsible for more than 50 percent of blog spam with more than 30,000 spam posts during the two-week interval.

More than 30,000 different IP blocks were used in posting blog comments. Among these, six IP blocks were responsible for more than 50 percent of blog spam. An IP block is a range of IP hosts in a network. Usually, it's represented by an IP address/number. For example, 192.168.1.1/24 covers all IP addresses in the range between 192.168.1.0 and 192.168.1.255. Figure 6 depicts the top six offender IP blocks, with IP block 69.46.0.0/19 yielding more than 150,000 spam posts.

From more than one million blog posts collected in the two weeks of our experiments, more than 75 percent of the comments were spam. Our results showed that thousands of IP addresses are associated with a few email addresses. For example, the email gerfercer@gmail.com was associated with more than 100,000 unique IP addresses. This could be due to the use of conventional evasion techniques to avoid IP blacklisting, such as proxies and IP spoofing.

Furthermore, the experiments found that five ASNs and six IP blocks are responsible for more than 50 percent of blog spam. This indicates that spammers are still leveraging conventional techniques in launching their operations. Traditional spammers used to rent colocation facilities—such as servers in a commercial datacenter—to launch their operations.[8] Now they rely more on botnets and compromised machines in launching attacks. However, the few numbers of ASNs and IP blocks from our experiments suggest that blog spammers are still using colocation facilities to launch their attacks. □

References

1. D. Fetterly, M. Manasse, and M. Najork, "Spam, Damn Spam, and Statistics," *Proc. 7th ACM Int'l Workshop Web and Databases*, ACM Press, 2004, pp. 1–6.

2. M. Ramilli and M. Prandini, “Comment Spam Injection Made Easy,” *Proc. 6th IEEE Consumer Comm. and Networking Conf.*, IEEE Press, 2009, pp. 1–5.
3. Y-R. Lin et al., “Detecting Splogs via Temporal Dynamics Using Self-Similarity Analysis,” *ACM Trans. Web*, vol. 2, no. 1, 2008, pp. 1–35.
4. “Security Threat Report: 2009,” white paper, Sophos, Jan. 2009; www.sophos.com/sophos/docs/eng/marketing_material/sophos-security-threat-report-jan-2009-na.pdf.
5. A. Bhattarai, V. Rus, and D. Dasgupta, “Characterizing Comment Spam in the Blogosphere through Content Analysis,” *Proc. IEEE Symp. Computational Intelligence in Cyber Security*, IEEE Press, 2009, pp. 37–44.
6. A. Ntoulas et al., “Detecting Spam Web Pages through Content Analysis,” *Proc. 15th ACM Int’l Conf. World Wide Web*, ACM Press, 2006, pp. 83–92.
7. E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2004.
8. “Reputation-Based Mail Flow Control,” white paper, IronPort, 2002; www.ironport.com/pdf/ironport_reputation_based_control_whitepaper.pdf.

***Saeed Abu-Nimeh** is a security researcher at Websense Security Labs, San Diego. His research interests include Web security, phishing and spam detection, and machine learning. Abu-Nimeh has a PhD in computer science from Southern Methodist University. Contact him at sabu-nimeh@websense.com.*

***Thomas M. Chen** is a professor of networking in the School of Engineering at Swansea University, Wales. His research interests include network security, traffic control, and network protocols. Chen has a PhD in electrical engineering from the University of California, Berkeley. He’s a member of IEEE and the ACM. Contact him at t.m.chen@swansea.ac.uk.*

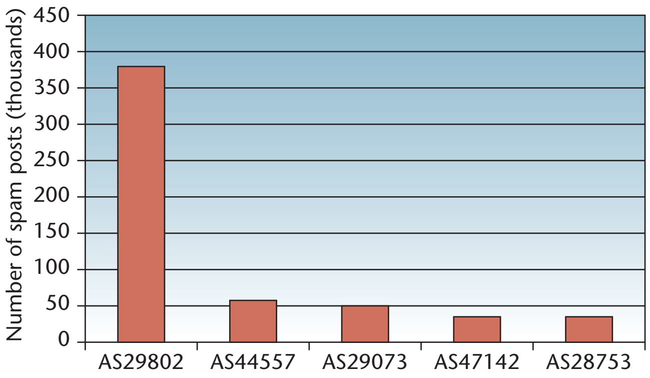

Figure 5. The top five offender autonomous systems numbers (ASNs). We measured the number of spam posts for these ASNs for a test period between 18 and 30 June 2009.

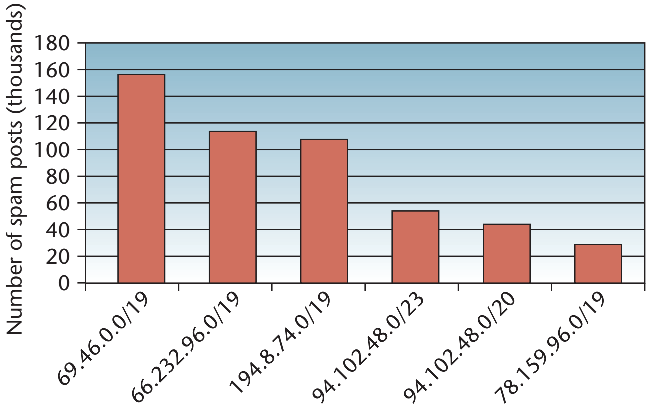

Figure 6. The top six offender IP blocks. These six blocks were responsible for more that 50 percent of blog spam, measured over a test period from 18 to 30 June 2009.