Spam and the Ongoing Battle for the Inbox

Even as spammers and phishers try ever-more sophisticated techniques to get past filters and into users' mailboxes, anti-spam researchers have managed to stay several steps ahead, so far.

Since August 1998, when *Communications* published the article "Spam!" by Lorrie Faith Cranor and Brian A. LaMacchia describing the then rapidly growing onslaught of unwanted email, the amount of all email sent has grown exponentially, but the volume of spam has grown even more. Spam has increased from approximately 10% of overall mail volume in 1998, constituting an annoyance, to as much as 80% today [8], creating an onerous burden on both tens of thousands of email service providers (ESPs) and tens of millions of end users worldwide. Large email ser-

By Joshua Goodman, Gordon V. Cormack, *and* David Heckerman

Illustration by Robert Neubecker

vices (such as Microsoft's Hotmail) may be sent more than a billion spam messages per day. Fortunately, however, since 1998, considerable progress has also been made in stopping spam. Only a small fraction of that 80% actually reaches end users. Today, essentially all ESPs and most email programs include spam filters. From the point of view of the end user, the problem is stabilizing, and spam is for most users today an annoyance rather than a threat to their use of email.

Meanwhile, an ongoing escalation of technology is taking place behind the scenes, with both spammers and spam filter providers using increasingly sophisticated solutions. Whenever researchers and developers improve spam-filtering software, spammers devise more sophisticated techniques to defeat the filters. Without constant innovation from spam researchers, the deluge of spam being filtered would overwhelm users' inboxes.

Spam research is a fascinating topic not only in its own right but also in how it relates to other fields. For instance, most spam filtering programs use at least one machine learning component. Spam research has exposed shortcomings in current machine learning technology and driven new areas of research. Similarly, spam and phishing have made the need to verify senders' identities on the Internet more important than ever and led to new more practical verification methods. Spam filtering is an example of adversarial information processing; related methods may apply not only to email spam but to many situations in which an active opponent attempts to thwart any new defensive approach.

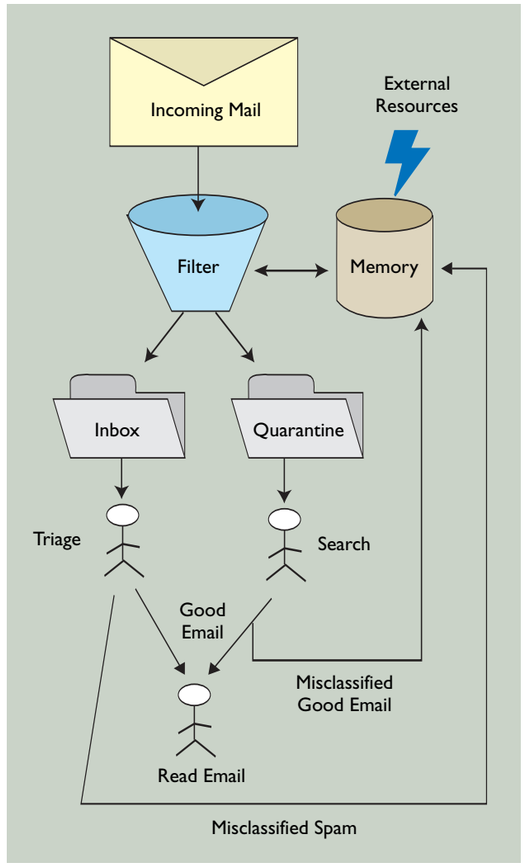


Figure 1. Spam filters separate email into two folders: the inbox, which is read regularly, and quarantine, which is searched occasionally. Mistakes—spam in the inbox or good email in quarantine—may be reported to the filter if noticed.

Around the time spam was becoming a major problem in 1997, one of us (Heckerman), along with other colleagues at Microsoft Research, began work on machine learning approaches to spam filtering [11]. In them, computer programs are provided examples of both spam and good (non-spam) email (see Figure 1). A learning algorithm is then used to find the characteristics of the spam mail versus those of the good mail. Future messages can be automatically categorized as highly likely to be spam, highly likely to be good, or somewhere in between. The earliest learning approaches were fairly simple, using, say, the Naive Bayes algorithm to count how often each word or other feature occurs in spam messages and in good messages.

To be effective, Naive Bayes and other methods need training data—known spam and known good mail—to train the system. When we first shipped spam filters, spam was relatively static. We had 20 users manually collect and hand-label their email. We then used this collection to train a filter that was not updated for many months. Words like "sex," "free," and "money" were all good indicators of spam that worked for an extended period. As spam filtering became more widely deployed, spammers adapted, quickly learning the most obvious words to avoid and the most innocuous words to add to trick the filter. It became necessary to gather ever-larger amounts of email (as spammers began using a wider variety of terms), as well as to update the filters frequently to keep up with spammers. Today, Hotmail uses a feedback loop system in which more than 100,000 volunteers each day are asked to label a message that was sent to them as either "spam" or "good" email. This regularly provides us new messages to train our filters, allowing us to react quickly to new spammer attacks and ploys.

Besides getting more data, faster, we also now use much more sophisticated learning algorithms. For instance, algorithms based on logistic regression and that support vector machines can reduce by half the amount of spam that evades filtering, compared to Naive Bayes. These algorithms "learn" a weight for each word in a message. The weights are carefully adjusted so results derived from the training examples of both spam and good email are as accurate as possible. The learning process may require repeatedly adjusting tens of thousands or even hundreds of thousands of weights, a potentially time-consuming process. Fortunately, progress in machine learning over the past few years has made such computation possible. Sophisticated algorithms (such as Sequential Conditional Generalized Iterative Scaling) allow us to learn a new filter from scratch in about an hour, even when training on more than a million messages.

Spam filtering is not the only beneficiary of advances in machine learning; it also drives exciting new research. For instance, machine learning algorithms are typically trained to maximize accuracy (how often their predictions are correct). But in practice, for spam filtering, fraud detection, and many other problems, these systems are configured very conservatively. Only if an algorithm is nearly certain that a message is spam is the message filtered. This issue has driven recent research in how to learn specifically for these cases. One clever technique that can reduce spam by 20% or more—developed by Scott Yih of Microsoft Research—involves training two filters. The first identifies the difficult cases; the second is trained on only those cases. Focusing on them improves overall results [12].

Meanwhile, spammers have not been idle as machine learning has progressed. Traditional machine learning for spam filtering has many weaknesses. One of the most glaring is that the first step in almost any system is to break apart a message into its individual words, then perform an analysis at the word level. Initially, spammers sought to overcome these filters by making sure that words with large (spammy) weights, like "free," did not appear verbatim in their messages. For instance, they might break the word into multiple pieces using an HTML comment (`fr<!--><-->ee`) or encode it with HTML ASCII codes (`fr&#101xe`). When displayed to a user, both these examples look like "free," but for spam-filtering software, especially on servers, any sort of complex HTML processing is too computationally expensive, so the systems do not detect the word "free."

Spammers do not use these techniques randomly, carefully monitoring what works and what doesn't. For instance, in 2003, we saw that the HTML character encoding trick was being used in 5% of spam sent to Hotmail. The trick is easy to detect, however, since it is rare for a normal letter like "e" to be encoded in ASCII in legitimate email. Shortly after we and others began detecting it, spammers stopped using it; by 2004, it was down to 0% [6]. On the other hand, the token-breaking trick using comments or other tags can be done in many different ways, some difficult to detect; from 2003 to 2004, this exploit went from 7% to 15% of all spam at Hotmail. When we attack the spammers, they actively adapt, dropping the techniques that don't work and increasing their use of the ones that do.

Scientific evaluation is an essential component of research; researchers must be able to compare methods using standard data and measurements. This kind of evaluation is particularly difficult for spam filtering. Due to the sensitivity of email (few of us would allow our own to be publicly distributed, and those that would are hardly typical), building a standard benchmark for use by researchers is difficult. Within the context of the larger Text REtrieval Conference (TREC) effort begun in 1991, a U.S.-government-supported program (trec.nist.gov) that facilitates evaluations, one of us (Cormack) founded and coordinates a special spam track to evaluate participants' filters on real email streams; more important, it defines standard measures and corpora for future tests. It relies on two types of email corpora:

- Synthetic, consisting of a rare public corpus of good email, combined with a carefully modified set of recent spam. It can be freely shared, and researchers run their filters on it; and
- Private, whereby researchers submit their code to testers, who run it on the private corpora and return summary results only, ensuring privacy.

Results in terms of distinguishing spam from good email on the two types are similar, suggesting that, for the first time, relatively realistic comparisons of different spam-filtering techniques may be carried out by different groups. Future TREC tracks aim to develop even more realistic evaluation strategies. Meanwhile, the European Conference on Machine Learning (www.ecmlpkdd2006.org) addressed the issue of spam-filter evaluation through its 2006 Discovery Challenge, testing the efficacy of spam filtering without user feedback.

One surprising result is that a compression-based

technique is more effective for spam filtering than traditional machine learning systems [1]. Compression-based systems build a model of spam and a model of good email. A new message is compressed using both the spam model and the good-email model. If the message compresses better with the spam model, the message is likely spam; if it compresses better with the good-email model, the message is more likely legitimate. While compression-based filtering techniques have (in theory) been well understood for years, this is the first instance we know of in which they beat traditional machine-learning systems. The best compression-oriented results have used Dynamic Markov Coding; however, better known techniques (such as Prediction by Partial Matching, or PPM) work nearly as well.

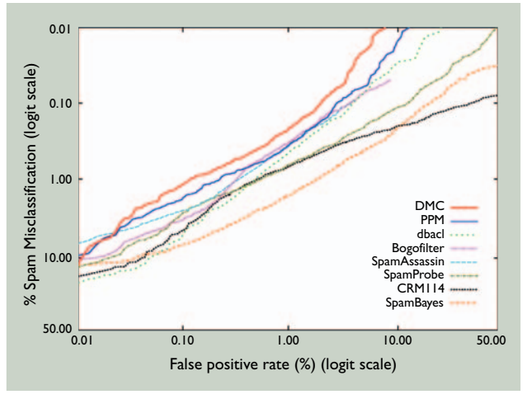


Figure 2. Comparison of compression-based techniques DMC and PPM in common filters using the TREC methodology [1].

These compression-oriented results open a variety of avenues for ongoing research. However, we have yet to understand fully why they work so well for spam filtering. Can they be adapted to work well for other text-classification problems, or is there something unique about spam, and if so, what? One clear advantage of these techniques is they work even against sophisticated obfuscations. Because they apply to a stream of bits they are inherently insensitive to character encoding and to the makeup of words; hence they are robust to many spammer tricks, like the HTML obfuscations outlined in Figure 2.

Compression models also pose deployment challenges. First, they can be large, especially when trained on large amounts of data. Second, they may implicitly contain pieces of real email, causing privacy issues. For some kinds of filters (such as personal ones users build for themselves on their own data), they are extremely promising. For a filter built by, say, a large company using many users' data widely deployed to other users, these issues remain to be solved, while more conventional learning techniques are still state of the art (see Figure 3).

One notable challenge for spam-filter research is that the techniques spammers use are different for different recipients; their ability to adapt also differs. For instance, Hotmail and other large ESPs are subjected to targeted spammer attacks. Spammers easily obtain accounts on these systems, and before sending in bulk, they might keep sending test messages over and over until they understand how to beat a particular filter. Our filters adapt over time. Some spammers, as they send large quantities of spam, monitor whether or not these messages are being received on their test accounts and may dynamically adapt their techniques. On the other hand, large ESPs also give some advantage to spam fighters. Hotmail and other large ESPs are able to quickly aggregate information about hundreds of millions of users, rapidly updating their filters as new attacks are detected.

It is difficult to measure how successful spammers' adaptations are in defeating spam filters. In the course of the TREC spam track, we have evaluated new and old machine-learning filters on new and old email, finding no material difference in filtering performance related to the age of the email. Yet we have also seen that filters trained on recent spam perform much better than those trained on email even a few weeks old. We have also found through certain techniques deployed at Hotmail that over a broad range of historical data, the techniques worked well, but within a week of deployment, spammers have already adapted. Overall, it is clear that spam changes quickly, and spammers react to changes in filtering techniques. Less clear is whether spam is getting more difficult over time or whether spammers are simply rotating from one technique to another, without making absolute progress.

Spammers themselves rarely take over machines. Instead, SPECIALISTS INFECT MACHINES, then rent them out to spammers.

```
From: "Iris Gist" <irisgist@chpw.org>
To: gvcormac@plg2.math.uwaterloo.ca
Subject: postnata 4344
Date: Fri, 26 May 2006 11:21:26 -0700

Hi,
=20
M e R / D / A
V / a G R A
P R O z & C
A m o x / c i I l / n
C i A L / S
V A L / u M
T r & m a d o I
A m B / E N
X & n a x
L e V / T R A
S O m &
=20
http://www.prosebutis.com <http://www.prosebutis.com>=20

have no more argument. I have chosen Mr. Baggins and that ought to !6te=20
enough for all of you. If I say he is a Burglar, a Burglar he is, or=20
will be when the time comes. There is a lot more in him than you guess,=20
and a deal more than he has any idea of himself. You may (possibly) all=20
live to thank me yet. Now Bilbo, my boy, fetch the lamp, and lets have=20
little light on this!=20
```

Figure 3. DMC colors fragments of the message: red if they are likely to be found in spam, green if they are likely to be found in good email.

IP-Address-Based Techniques

It may be that techniques based on the content of the message are defeated too easily; there may simply be too many ways to obfuscate content. Many spam-filter researchers have thus focused on aspects of spam that cannot be hidden. The sender of a message—its IP address—is the most important of them.

The most common method for IP-address filtering is to simply blacklist certain IP addresses. When an address is known to send spam, it can just be barred from sending any email for a period of time. This approach can be effective, and several groups produce and share lists of bad addresses. This approach also involves limitations, however. For instance, spammers have become adept at switching IP addresses. Most blacklists are updated hourly, prompting some spammers to acquire huge amounts of bandwidth to allow them to send tens of millions of messages per IP address in the hour or so before the email is blocked; they then switch to another one. Blacklists can also result in false positives (lost good mail) when a good sender inherits a blacklisted IP address or a single IP address is used to send both spam and good email. Blacklists are a powerful tool but no panacea.

Some spammers are extremely clever at trying to circumvent IP-blocking systems. One common technique is to enlist so-called zombie machines or botnets, or computers, typically owned by consumers, that have been infected with viruses or Trojans that give spammers full control of the machine. The spammers then use them to send spam. Zombies provide interesting insight into the spam ecosystem. Spammers themselves rarely take over machines. Instead, specialists infect machines, then rent them out to spammers. Estimates of the price charged by the specialists for these machines vary, but at least one botnet operator rented them for $3/computer/month.

Some methods spammers use to obtain IP addresses are amazingly sophisticated. For instance, in one complex attack, where an ISP blocked outbound traffic on port 25 (the email port) but not inbound traffic, spammers were able to perform low-level TCP/IP protocol hacking to route outbound traffic through unblocked machines and inbound packets to the blocked machines; the result was that the email appeared to have been sent by the blocked machines.

Secure Identity

Numerous attempts have sought to introduce cryptographically secure identities to email, including such standards as PGP and S/MIME, but none has been widely adopted. Identity is particularly important in spam filtering. Almost all spam filters have some form of safe list, allowing users and administrators to identify senders whose email they trust. But without a working identity solution, spammers might abuse these safe lists by, for instance, sending email from someone (such as receipts@amazon.com) who is commonly safelisted. In addition, in phishing spam—a particularly insidious type of spam—spammers impersonate a legitimate business in order to steal passwords, credit card numbers, Social Security numbers, and other sensitive personal information. A working identity solution can have a substantial effect on these spammers.

Traditional cryptographic approaches to identity security have been robust to most attacks but too difficult to deploy for practical reasons. They typically focus on the identity of a person rather than the identity of an email address, thus requiring a certifying agency of some sort. Some proposals would require all Internet users to go to their local Post Office and pay a fee to get a certificate. In addition, these proposals

Although SenderID was released **in 2004**, ALMOST 40% OF NON-SPAM EMAIL **TODAY IS** SENDERID-COMPLIANT, thus reducing the opportunities for email spoofing.

usually require some form of attachment or inclusion in the email message itself, confusing some users.

In contrast, identity solutions driven by spam have been more pragmatic. In particular, Domain Keys Identified Mail and SenderID have both focused on identity at the domain level; they make it possible for email servers to determine whether this email really came from this domain. In addition, both DKIM and SenderID have used the existing Domain Name System infrastructure to distribute key information. While the DNS infrastructure is far less secure than are commonly proposed cryptographic solutions, it has only rarely been compromised in practice. This pragmatic approach to identity has allowed surprisingly quick adoption of these new techniques. Although SenderID was released in 2004, almost 40% of non-spam email today is SenderID-compliant, thus reducing the opportunities for email spoofing.

Spammers adapted their attack techniques to this technology as well. When first released, SenderID was used more by spammers than by legitimate senders; spammers would create a new domain name, then create the relevant records, proving that the email was not spoofed. It is important to understand that these identity solutions are not aimed at stopping spam directly. Rather, they are a key part of a more complex strategy, aiming to prevent safe-list abuse and phishing while allowing the spam filtering component to learn "good" reputations for legitimate senders. They've shown early success in moving toward all three goals.

Other Filtering Technology

One of the most widely deployed spam filtering techniques is similarity-matching solutions. They attempt to find examples of known spam; for example, email that has gone to a special trap account that should receive no legitimate email that users have complained about. They then try to match new examples to this known spam. Spammers actively randomize their email in an attempt to defeat these matching systems. In some cases (such as spam where the primary content is an image meant to defeat both matching-based and machine-learning-based text-oriented filters), spammers even randomize the image to defeat image-matching technologies. Promising recent research has focused on text-oriented matching systems; for instance, work at AOL [7] has used multiple different hashes to make the matching systems more robust to randomization; and work at IBM [10] inspired by bioinformatics has sought to find characteristic subsequent uses of words that occur in spam but not in good email.

Similarity-matching systems are generally a good complement to machine-learning-based systems. They help prevent spammers from finding a single message that can beat a learning-based filter, then send it to hundreds of millions of users. In order to defeat a combined system, spammers must find email that beats a machine-learning system, randomizing the message in such a way that it simultaneously defeats a matching-based system.

Image-based spam is one way to attack both machine-learning systems and matching systems. In this form of spam, the text of a message is random (defeating matching systems) and innocuous (defeating machine-learning systems). Prominently displayed, perhaps before the text, is an image consisting entirely of an image of text. Optical character recognition software is too slow to run on email servers and probably not accurate enough in an adversarial situation. These images were initially stored mostly on the Web, using image-source links, rather than embedded in the message. Because most messages are never opened, using these links reduces bandwidth cost to spammers, allowing them to spam even more.

Many email providers have responded by blocking most Web-based images while still allowing embedded images stored in the message itself. Spammers have responded by embedding their images in messages. These images were initially identical across millions of messages. Many spam-filter providers responded by using image-matching algorithms. Spammers countered by randomizing the content of

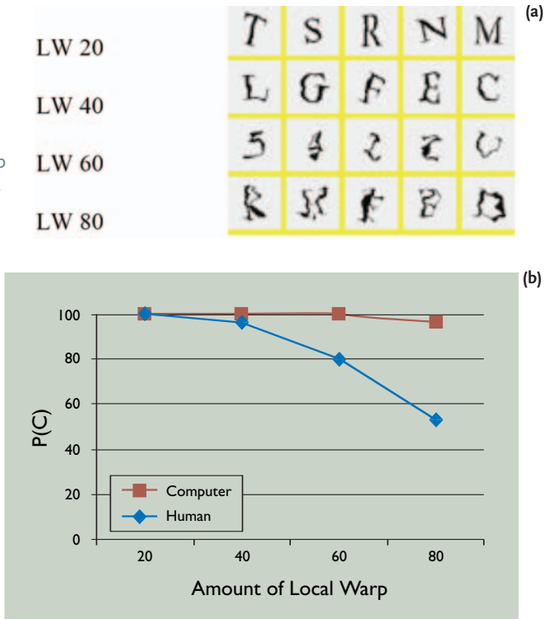
HUMAN INTERACTION PROOFS

HIPs (also known as "completely automated public Turing Tests to tell computers and humans apart," or CAPTCHAs, or just plain Turing Tests) are a key component in preventing abuse. The most common type of HIP is an image of a sequence of letters and digits that has been automatically distorted. One of the many ways they are used is before signing up for most free email accounts, users are required to solve one—correctly entering the sequence of letters and numbers in the image. Without HIPs, spammers would use these services to produce a torrent of spam (see the other sidebar). They are also used to prevent automated password attacks. Several products (such as MailBlocks and Matador) have used HIP challenges for suspected spam as a kind of economic approach. HIPs also prevent, for instance, the automated harvesting of Web site data and automated attempts to steal passwords.

As HIPs have been used more widely and become more critical in preventing abuse, it has become more important to understand just how robust and effective they are. Kumar Chellapilla, a scientist at Microsoft Live Labs, and his colleagues there and at Microsoft Research have studied HIPs in detail, finding them to be surprisingly vulnerable to attack. In [2], they reported that nearly all commercially deployed HIPs could be broken with high accuracy. Because the goal of a HIP is to prevent automation, adversaries can accept relatively low solution rates; if they fail, they just try again over and over. The most effective HIP tested in [2] could be broken only 5% of the time, while the worst could be solved 67% of the time.

Chellapilla and his colleagues then set out to design better HIPs. Since the goal of a HIP is to be too difficult for a computer (while being readily solvable by a human), they conducted both computer studies and human studies. For each of seven different distortion techniques, they tried various levels of distortion, measuring the rate at which the problem became more difficult for humans to solve compared to the rate at which it became more difficult for computers to solve.

An example of their experiments is shown in part (a) and part (b) of the figure here for the local warp distortion type. For this distortion, at levels of 60 and 80, humans found the task—decoding/transcribing the HIP—extremely difficult, while computer performance was barely affected. In all seven of their experiments, computers did as well as humans or better at decoding/transcribing the HIP as the distortion level increased.



(a) Local warp distortion at four arbitrary parameter settings. (b) Human vs. computer accuracy at four settings of local warp distortion. Human accuracy falls more quickly than computer accuracy as the distortion increases. (c) Three samples of a segmentation-based HIP.

This is a very disturbing outcome. If HIPs are automatically solvable by computers, the key barrier against mass automation of many abuses will be gone. There is hope, however. In other research, Chellapilla and his colleagues focused on building segmentation-based HIPs in which the key distortion focused on making it difficult to find word boundaries (see part c of the figure). This kind of segmentation distortion appears to be the only problem where computers are still inferior to humans. The human brain does this task effortlessly but has become an algorithmic and computational challenge for computer vision and handwriting recognition. Hopefully, this finger-in-the-dike approach is sufficient for stopping a flood of abuse. C

OUTBOUND SPAM

Even though most research on spam focuses on stopping inbound spam, outbound spam is an important issue as well. Spammers love to send email from free email providers; no one wants to block all the mail from a major service (such as Hotmail or Yahoo! Mail). In many ways, outbound spam is a more interesting research topic than its inbound counterpart. Imposing economic costs on senders to prevent inbound spam has proved difficult in practice. For outbound spam, however, the ESP controls the environment and tools and can more readily impose economic costs (such as computational resources, money, and HIPs). For instance, an ESP might provide the tools to solve computational puzzles to all its users. ESPs that charge a fee have access to credit-card information, making it possible for them to impose monetary costs. ESPs typically control the user interface, making it easier for them to impose HIP challenges.

Our 2004 theoretical analysis of economic approaches to stopping outbound spam produced some interesting, somewhat surprising, results [5]. The most interesting was that a technique imposing costs initially (but then stops charging for additional messages) can be as effective as one that charges for every single message. This means that users may at first be annoyed with HIPs, computation, or monetary costs, but after a while, these costs stop. Asymptotically, legitimate users of the system pay zero cost per message, but the costs to spammers can be kept high, ideally higher than the benefit they get from spamming in the first place. While most inbound spam research is empirical in nature, these are realistic, provable bounds on spammer costs.

One disappointing result was that rate limiting is surprisingly ineffective. When ESPs are notified that they are a major source of spam, a natural reaction is for them to impose rate limits on senders. We found that it is important for ESPs to include some sort of rate limiting; with no limits, spamming is very cheap. But past a certain monetary point, rate limiting has almost no effect. Intuitively, if the rate limit is cut in half, it takes about twice as long to receive enough complaints to terminate the account; the same total spam is sent, and the spammer's cost per message is unchanged. If a spammer wishes to maintain his sending rate, he can purchase twice as many accounts. His up-front costs double, but the asymptotic costs per message stay the same.

Fortunately, we also found good ways to increase spammer costs, hopefully above the point at which spamming is cost-effective. Spammer costs are roughly inversely proportional to the number of messages a spammer can send before a complaint is received. A consequence is that any method that allows ESPs to more quickly learn about abusive accounts and terminate them will substantially raise the cost to spammers. One such system is the Windows Live Mail Smart Network Data Services (https://postmaster.live.com/snds/index.aspx), a public service provided by Microsoft that allows ISPs to quickly identify IP addresses they own that are major sources of spam, helping them quickly take action. C

the images; some even broke the images into multiple pieces to be reassembled only when HTML-based email is rendered. These randomized image-based messages with innocuous-looking text are especially difficult to identify through automated means.

Many payment-based systems have also been proposed over the years for spam filtering. Examples include: those that require a time-consuming computation, first suggested in [4]; those that require solving a human interaction proof (see the sidebar "Human Interaction Proofs"), first suggested in [9]; and those that require making some form of cash micropayment, possibly refundable. Unfortunately, these economic approaches are difficult to deploy in practice. For computational puzzles and cash micropayments to succeed, these systems must be widely deployed, and in order to be widely deployed, there must be some expectation of success—a catch-22. In one exciting development, Microsoft Outlook 2007 includes computational puzzles—the first wide-scale deployment of a computational system to help stop spam. Observing the effectiveness of this approach in practice will be interesting. For the related problem of outbound spam—stopping people spamming from a public service like Hotmail—economic approaches have been surprisingly successful (see the sidebar "Outbound Spam").

Also worth mentioning are legislative attempts to stop spam (such as the 2003 CAN-SPAM Act, also known as the Controlling the Assault of Non-Solicited Pornography and Marketing Act). Unfortunately, these legislative approaches have had only a limited effect (see Grimes's article on page 56). Many forms of spam can be sent internationally, and clever spammers are good at making themselves difficult to trace. Many forms of spam are fraudulent or illegal (such as phishing scams and pump-and-dump stock schemes), so additional laws are likely to offer only incremental disincentives. Technology will continue to be the most important mechanism for stopping spam.

CONCLUSION

From the end-user point of view, spam appears to be roughly under control—an annoyance that has stabilized at a tolerable level. From the point of view of spam researchers and developers, it is an ongoing battle, with both spammers and spam fighters becoming ever more sophisticated.

As spam filtering has evolved, so has the community of spam fighters. One active part of that community is the Conference on Email and Anti-Spam (www.ceas.cc) begun in 2004. Many of the methods and results described here were first presented at CEAS conferences, focusing not just on spam research but on positive ways to improve email as well. Studies from the Pew Foundation show email to be the number-one application on the Internet and clearly deserving of its own research community. CEAS brings together academic researchers, industrial researchers, developers of email and spam products, and operators of email systems. The practical nature of spam fighting and email development encourages and requires a degree of collaboration and interaction across disciplines that is relatively rare in other areas of computer science.

Email spam is by no means the only type of abuse on the Internet. Almost any communication method involves a corresponding form of spam. For example, instant messaging systems are subject to IM Spam (SPIM), and chat rooms are subject to chat spam (SPAT). A key problem for Internet search engines is Web spam perpetrated by people who try to artificially boost the scores of Web pages to generate traffic. Other forms of abuse include click fraud, or people clicking on advertisements to steal money or hurt competitors. It turns out that the same technique can be used across these different types of spam. For instance, IP-address-based analyses can be very helpful for filtering spam; spammers respond by attempting to acquire a variety of IP addresses cheaply (such as using zombies and open proxies); countermeasures for detecting zombies and open proxies can then be used to identify and stop the spam. Machine learning can also be applied to these forms of abuse, and work (such as learning optimized for low-false positive rates, originally developed for email spam) may be applied to these other areas as well.

We hope to solve the problem of email spam, removing the need for endless escalations and tit-for-tat countermeasures. When anti-spoofing technology is widely deployed, we'll be able to learn a positive reputation for all good senders. Economic approaches may be applied to the smallest senders, even to those who are unknown. Even more sophisticated machine-learning systems may be able to respond to spammers more quickly and robustly than they can adapt to. Even when that day comes, plenty of interesting and important problems will still have to be solved. Spammers won't go away but will move to other applications, keeping us busy for a long time to come. C

REFERENCES

1. Bratko, A., Cormack, G., Filipic, B., Lynam, T., and Zupan, B. Spam filtering using statistical data compression models. *Journal of Machine Learning Research 7* (Dec. 2006).
2. Chellapilla, K. and Simard, P. Using machine learning to break visual human interaction proofs. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS) Conference* (Vancouver, Canada). MIT Press, 2005, 265–272.
3. Chellapilla, K., Simard, P., and Czerwinski, M. Computers beat humans at single character recognition in reading-based human interaction proofs (HIPs). In *Proceedings of the Second Conference on Email and Anti-Spam (CEAS)* (Palo Alto, CA, July 21–22, 2005).
4. Dwork, C. and Naor, M. Pricing via processing or combatting junk mail. In *Proceedings of the 12th Annual International Cryptology Conference (Lecture Notes in Computer Science)* (Santa Barbara, CA, Aug. 16–20). Springer, 1992, 137–147.
5. Goodman, J. and Rounthwaite, R. Stopping outgoing spam. In *Proceedings of the ACM Conference on Electronic Commerce (EC'04)* (New York, May 17–20). ACM Press, New York, 2004, 30–39.
6. Hulten, G., Penta, A., Seshadrinathan, G., and Mishra, M. Trends in spam products and methods. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)* (Mountain View, CA, July 30–31, 2004).
7. Kolcz, A., Chowdhury, A., and Alspector, J. The impact of feature selection on signature-driven spam detection. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)* (Mountain View, CA, July 30–31, 2004).
8. Messaging Anti-Abuse Working Group. *MAAWG Email Metrics Program, First Quarter 2006 Report.* June 2006; www.maawg.org/about/FINAL_1Q2006_Metrics_Report.pdf.
9. Naor, M. *Verification of a Human in the Loop or Identification via the Turing Test*; www.wisdom.weizmann.ac.il/~naor/.
10. Rigoutsos, I. and Huynh, T. Chung-Kwei: A pattern-discovery-based system for the automatic identification of unsolicited e-mail messages. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)* (Mountain View, CA, July 30–31, 2004).
11. Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization—Papers from the AAAI Workshop.* AAAI Technical Report WS-98-05 (Madison, WI, 1998).
12. Yih, W., Goodman, J., and Hulten, G. Learning at low false positive rates. In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS)* (Mountain View, CA, July 27–28, 2006).

JOSHUA GOODMAN (joshuago@microsoft.com) is a senior researcher at Microsoft Research, Redmond, WA.
GORDON V. CORMACK (gvcormac@waterloo.ca) is a professor in the David R. Cheriton School of Computer Science at the University of Waterloo, Waterloo, ON, Canada.
DAVID HECKERMAN (heckerma@microsoft.com) is a senior researcher at Microsoft Research, Redmond, WA.