Stopping Spam

What can be done to stanch the flood of junk e-mail messages?

By Joshua Goodman, David Heckerman and Robert Rounthwaite

In 1978 the first spam e-mail--a plug from a marketing representative at Digital Equipment Corporation for the new Dcsystem-20 computer--was dispatched to about 400 people on the Arpanet. Today junk correspondence in the form of unwanted commercial solicitations constitutes more than two thirds of all e-mail transmitted over the Internet, accounting for billions of messages every day. For a third of all e-mail users, about 80 percent of the messages received are spam. Recently spam has become more threatening with the proliferation of so-called phishing attacks--fake e-mails that look like they are from people or institutions you trust but that are actually sent by crooks to steal your credit-card numbers or other personal information. Phishing attacks cost approximately $1.2 billion a year, according to a 2004 Gartner Research study.

The phenomenon of spam afflicts more than just e-mail. Inside chat rooms lurk "robots" that pretend to be human and attempt to convince people to click on links that lead to pornographic Web sites. Instant messaging (IM) users suffer from so-called spIM--e-mail spam cognates. Blogs can be corrupted by "link spammers" who degrade Internet search engine operations by adding misleading links to sites that distort the utility ratings of Web sites and links.

The suffocating effect of spam sometimes seems likely to undermine, if not wreck, Internet communications as we have come to know them. The reality, however, is not so bleak. Several techniques for intercepting spam and discouraging spammers have been invented, and more are on the way. The methods we shall discuss focus on junk e-mail, but many of them could apply to other incarnations of spam as well. No one of these will be a magic cure, but combinations--if embraced by enough of us--could work wonders. It is not unrealistic to hope for a day when our e-mail boxes will once again be nearly spam-free.

Insidious E-mails

The proliferation of fraudulent e-mail results directly from favorable market forces: spam is exceedingly cheap to distribute. It is not altogether free, though. We estimate that a message costs about one hundredth of a cent to send. At these cut-rate prices a spammer can earn only $11 per sale and still make a profit, even if the response rate is as low as one in 100,000. Hence, although very few e-mail users ever buy anything advertised in spam, all of us suffer because of those who do.

One of the most infuriating aspects of spam is that it changes continually to adapt to new attempts to stop it. Each time software engineers attack spam in some way, spammers find a way around their methods. This spam arms race has led to a continuous coevolution of the two, which has resulted in ever increasing sophistication on both sides.

Another fundamental problem stems from the fact that engineers and legislators find it extremely difficult to define spam. Most laws define it as unsolicited commercial e-mail from someone without a preexisting business relationship. This characterization is too broad, however. We recently received an e-mailed proposal, for example, to turn a short story we had published on the Internet into a motion picture. This communication met the requirements of the law: unsolicited, commercial, from an unknown sender, but almost no one would call it spam. An alternative definition might include the fact that spam is typically mass-mailed. But we recently solicited papers for a technical conference to discuss e-mail systems and anti-spam methods by sending requests to 50 people we had never met who had published on this topic. None of them complained. Perhaps the best characterization of spam is that it is poorly targeted and unwanted. Formulating a precise definition of spam is exceedingly difficult, but, like pornography, we certainly know it when we see it flooding our mailboxes.

Morphing Messages

We have worked on the spam problem since 1997, when one of us (Heckerman) suggested that machine-learning methods might provide an effective line of attack. Since then, the three of us and our many colleagues in the software business have investigated and developed several approaches to stopping spam. They encompass combinations of technical and legal solutions as well as industrywide initiatives.

Some of the earliest schemes used to stop spam are so-called fingerprint-matching techniques. In these systems, engineers first find examples of spam and let computer programs "fingerprint" them. The fingerprint is a number derived from the content of the message, so that similar or identical messages get the same number. To give a simplified example, one could add the number of As in a message plus 10 times the number of Bs plus 100 times the number of Cs, and so forth. When a new message arrives,
anti-spam programs compute its fingerprint and then compare it with those of known spam. If the fingerprints match, the program deletes or archives the message.

Regrettably, these straightforward methods were easily defeated by spammers, who simply started adding random characters to their messages. Spam fighters responded with more sophisticated fingerprint techniques that try to exclude obvious sequences of random characters, but spammers overcame these efforts with more legitimate-looking random content, such as fake weather reports. Ultimately, making fingerprint systems sufficiently robust to see through spammer randomizations turns out to be quite hard.

**Smart Filters**

Rather than pursuing fingerprint methods, our group followed an avenue that exploited machine-learning capabilities. These specialized computer programs can learn to distinguish spam e-mails from valid messages, and they are not so easily confused by additions of a few random letters or words.

At first, we tried the simplest and most common machine-learning method. The Naive Bayes algorithm starts with the probabilities of each word in the message. "Click," "here" and "unsubscribe," for instance, might each have a probability of 0.9 of showing up in spam and a probability of 0.2 of showing up in legitimate e-mail messages (1.0 being certainty). By multiplying the probabilities of all the words in a message and using a statistical principle known as Bayes' rule, we get an estimate of how likely it is that a message is spam.

The Naive Bayes strategy works remarkably well at determining what genuine e-mail looks like, and like all such learning methods, it resists simple obfuscations. Yet we were well aware of its shortcomings. Its assumption that words in e-mail are independent and unrelated is in many cases false (for instance, "click" and "here" often appear together), which skews results.

Because of these difficulties, our research focuses on discriminative linear models, which optimize the models' later decisions when assigning weights to features. These features include words and properties of the messages, such as whether the message was sent to many recipients. These models can in some sense learn the relations between words—for instance, "knowing" not to place too much weight on words that tend to occur together, like "click," "here" and "unsubscribe." To explain further: let us say a Naive Bayes model saw these three words, which are often associated with spam. It might decide it has enough evidence to conclude that any message containing them is junk, leading it to sometimes delete valid e-mail. In contrast, a discriminatively trained model would know that the words tend to occur together and thus would assign lower, more reasonable, weights to them. Such a system could even learn that a word such as "here," which may occur more often in spam, should be given no weight at all because it does not really help tell good from bad. Discriminative methods can also discover that certain words cancel each other out. Although a word such as "wet" occurs more frequently in spam, when "wet" is found with "weather," chances are the message is legitimate.

An advantage of Naive Bayes systems is that they are easy to train. Determining the weights for discriminative methods is much harder: it requires programmers to try many sets of weight values for words and other features to find a combination that does the best overall job of distinguishing spam from non-spam. Fortunately, researchers have made significant progress here. Algorithms such as the Sequential Minimal Optimization algorithm, invented by John C. Platt of Microsoft, and the Sequential Conditional Generalized Iterative Scaling (SCGIS) algorithm, created by one of us (Goodman), are tens or hundreds of times faster than older techniques. When dealing with large amounts of spam training data, more than a million messages and hundreds of thousands of weights, quicker algorithms are critical.

**Hiding Spam**

We had always known that our machine-learning systems, which focus on the words in a message, would be vulnerable to spammers who obscure the wording of their output. Clever spammers, for example, learned to use words such as "M0NEY" (with a zero instead of the letter "O") or to use HTML (hypertext markup language) tricks, such as splitting a word into multiple parts (say, "cl" and "ick" instead of "click"). Because the telltale terms ("money," "click") are no longer in the message, the filter can be confused. The good news is that machine-learning systems can often learn about these tricks and adapt.

Unfortunately, we had assumed erroneously that few people would respond to a message that was obviously attempting to defeat a spam filter—for we thought, who would buy a product like that? Sadly, we were wrong; purchasers of illicit or illegal products do not expect the sellers to employ respectable advertising techniques. So we have had to alter our learning systems by employing what researchers call n-gram models. These techniques use subsequences of words to detect the key words often associated with spam. If an e-mail message contains the phrase "n@ked l@dies", for instance, the n-grams extracted from this phrase would include "n@k," "n@ke," "l@dies," and so on. Because these word fragments appear in confirmed spam messages, their presence provides valuable clues.

N-gram techniques have also helped us improve the utility of our filters when they are applied to foreign languages. Japanese and Chinese, for example, do not use spaces to separate words, so explicitly finding word breaks is very difficult. For these languages, n-gram-enabled systems simply screen every possible word and word fragment.

**Image-Based Spam**
Spammers sometimes hide their message in an image, where machine-learning systems cannot analyze the content (although they can still exploit other clues, such as the links in the message, sender reputation information, and so forth). One promising area of future research is the use of optical character-recognition (OCR) techniques for spam filtering. The same OCR techniques that are used for scanning a document could find all the text in the images and feed it to a machine-learning filter.

One of the more offensive aspects of spam is the appearance of pornographic images in one's mailbox. Fortunately, computer-vision researchers have made great progress in the automatic detection of pornographic images. Work in this field is surprisingly broad, for it has applications in preventing children's access to Web sites containing sexual material and in preventing pornographers from abusing free Web hosting systems. Such image recognition is, however, still time-consuming, and the reliability of identification needs to improve. Benign images, especially those showing large amounts of skin, can trigger false positives.

Our team is also investigating the analysis of universal resource locator (URL) information—the code that links to Web pages—to distinguish spam. Ninety-five percent of spam messages contain a URL. Most spammers' first goal is to get users to visit their Web site (although a small fraction prefer contact through telephone numbers), so URL information is an especially good target for filters.

Filters can exploit URL information in many ways. Some anti-spam software providers have already started blocking spam that contains links to known spam-related Web pages. Links to previously unknown domains can be considered suspicious: spammers generate new domains very quickly, whereas most legitimate domains are long-lived. On the other hand, URL information can also be an indicator of legitimate e-mail: a message that contains only pointers to known non-spam-related pages, or no URLs at all, is much less likely to be spam.

Prove It
Although filtering techniques work quite well, we recognize that spammers will always try to attack them. Rather than trying to win this endless competition, we believe the most effective approach in the long run would be to change the rules of the game. Hence, we are exploring proof systems—those whose goal is to require more from a spammer than he or she can afford.

That very first spam message was sent by manually typing in all 400 e-mail addresses. Today nearly all spam is sent automatically. If a sender can prove he or she is a human being, therefore, the sender is probably not a spammer. One of the earliest proof systems, suggested by Moni Naor of the Weizmann Institute of Science in Israel, made use of this notion. Naor proposed using what became known variously as HIPs (human interactive proofs), CAPTCHAs—an acronym for "completely automated public Turing test to tell computers and humans apart"—or reverse Turing tests [see "Baffling the Bots," by Lee Bruno; Scientific American, November 2003]. A HIP is a problem or puzzle designed to be easy for most humans but as difficult as possible for computers. People, for instance, are far superior to machines at recognizing sets of random alphabet letters that are partially obscured or distorted in an image.

A HIP forms part of a challenge-response system, which verifies that the sender is human. Before delivering a message, the system first checks a "safe list" of senders that the recipient considers trustworthy. If the sender is on the list, the message is delivered to the recipient's mailbox. If not, a challenge message goes to the original sender asking him or her to solve a HIP. After the sender solves the HIP, the response travels back to the recipient, whose e-mail software then transfers the message to the recipient's in-box.

This kind of interactive system can be annoying to users, however. Few people want to solve HIPs to send e-mail messages, and some even refuse to do so. An automated alternative proof mechanism, suggested by Naor and his colleague Cynthia Dwork, uses computational puzzles. To deliver a message successfully, the sender's e-mail system must first work out a computational puzzle posed by the recipient's system. The idea is to prove that the sender has expended more computer time on that individual message than a mass-marketing spammer could afford. Computational puzzles are like jigsaw puzzles—difficult to solve but easy to verify. On average, they could require many seconds or even minutes to find a solution but only milliseconds to validate. Solving these problems promptly would require spammers to buy many computers, making their costs prohibitive.

Yet another kind of proof system uses real money. Senders include with their message a kind of electronic check for a small amount, say a penny. Including the check allows their messages through spam filters. If the message is good, the recipient ignores the check, but if the message proves to be spam, a standardized complaint mechanism allows the recipient to cash it (or donate it to charity). Rate-limiting software meanwhile monitors senders' message volumes, ensuring they do not send more mail than their balance allows. For legitimate senders, this system is free, but for spammers, the cost per message might be one cent, 100 times our estimate of the current price—more than spammers can afford. For individuals, a small virtual deposit is also made by their Internet service provider or when they purchase e-mail software, so that for most users there is no cost at all.

Though straightforward in concept, monetary systems of this kind will be difficult to put into practice. Electronic systems require some overhead, so these transactions will not be free. Many questions about a micropayment banking infrastructure remain unanswered: Where will the money to pay for it come from? How will its operations be sustained, and who will profit? Who will get the payments, and how will the system prevent fraud? Although none of these problems are insoluble, setting up such a scheme will be tough.
All-Inclusive Attack

Our favorite strategy to halt spam combines e-mail filtering technology with a choice of proof tests: HIPs, computational puzzles and micropayments. In this approach, if the sender of a message is not on the recipient's safe list, the message is shunted to a machine-learning-based anti-spam filter that is designed to be especially aggressive; if the message is even a bit suspicious, the recipient is challenged. Most messages from one person to another, however, will not be contested, which reduces the number of proofs dramatically. The original sender is then given a choice: solve a HIP or a computational puzzle or make a refundable micropayment. If the sender's computer has newer software, it will work out the puzzle automatically, without the sender even being aware of the challenge. Otherwise, the sender will solve a HIP or make a micropayment.

Of course, individual companies or institutions, no matter how large, can make only so much progress against spam. A comprehensive solution requires cooperation of the entire computer and software industry, as well as national governments.

Approximately two thirds of all e-mail today uses "spoofed," or fake, sender addresses. The e-mail protocols in use today are based on trust: senders simply state who they are and the recipients believe them. This approach worked quite well in the early days of the Internet, before spam proliferated and before e-mail was used for business transactions.

Changing Internet standards is notoriously difficult, and it has been especially hard for e-mail protocols. A new industry standard, the Sender ID Framework, is finally addressing the spoofing problem, however. It works by adding supplementary information to the domain name server (DNS) to list the Internet protocol (IP) addresses from which mail sent from a specific domain (part of the network) can come. IP addresses are numeric addresses, like street addresses for individual computers, such as "1.2.3.4." The new DNS list of entries for a given domain--say, "example.com"--determines which IP addresses are allowed to send mail from that domain. If a spammer pretends to be example.com, his or her IP address will not match any IP address in example.com's Sender ID entries, and an e-mail program will know the spammer's mail is fake.

Although knowing the identity of the sender is a critical step in preventing fraud (such as phishing e-mails), it will not solve the spam problem. Nothing stops spammers from making up new identities every day or even every few minutes. That is why reputation services--by which senders can certify themselves as legitimate--will be so important.

In one case, IronPort's Bonded Sender program, sends deposit money as a bond surety. If complaint rates from the sender exceed a certain threshold, bond money is forfeited to a specified charity. Spam filters can check the Bonded Sender list and allow mail from a certified sender past the spam filter, even if it seems suspicious. Such programs can work even for those who send few messages. An Internet service provider (ISP) such as MSN or AOL, for example, might join a reputation service to gain access to its certification program; the ISP would then monitor each of their users' e-mail volume and complaint rates, thus ensuring that none of the provider's users are spammers.

If most legitimate senders adopted such a system (and there is little reason why they would not), spam filters could be made to be much more aggressive in dealing with the remaining mail, thus stopping the vast majority of junk messages. Reputation systems could be combined with challenge-response systems, so that those who cannot join have an alternative method for sending mail.

A complementary approach to stopping spam is governmental legislation. The CAN-SPAN act went into effect in the U.S. in January 2004. The act itself does not outlaw spamming; it only prohibits certain particularly egregious techniques, such as using fake "From:" information. Unfortunately, CAN-SPAN has had little measurable effect so far. The proportion of spam with a fraudulent "From:" address has actually increased from 41 to 67 percent since the act went into effect. European nations, in contrast, have passed much stricter opt-in laws, which prevent people from sending commercial e-mails without explicit permission from the recipient. According to anecdotal evidence, these laws have been somewhat effective, at least in stopping spamming by large legitimate companies.

Clearly, no law of a single country can hope to end spam. Only about half of all junk e-mail comes from the U.S.; the rest originates overseas. Only about one in three products sold via spam (such as insurance or mortgage refinancing) requires a domestic U.S. presence. Others, including pornography, "herbal enhancers" and confidence scams, are already abroad, can easily move offshore or are already illegal.

Spam-Free Future

Industry, the open-source community and the academic community all continue to study how to eliminate spam. We recently helped to establish the first formal conference on the topic—the Conference on Email and Anti-Spam, which attracted researchers from all over the world. Engineers at IBM showed how to use techniques from bioinformatics, originally designed for finding patterns in genes, to discern patterns in spam. AOL investigators demonstrated that multiple fingerprint systems with different vocabularies could better defend against spammer obfuscations. A team from the University of California at Davis described how the addition of a few common words could produce an effective attack against machine-learning spam filters and how, with training, the filters could be made more resistant to this attack.

We have little doubt that the combination of the current and next-generation techniques will eventually
stop most spam. There will always be a few spammers, of course, who are willing to pay the price to get through to our mailboxes, but the flood will turn into a trickle.