

# Fighting Spam



Every day, millions of people throughout the world are inundated with gigabytes of unsolicited advertisement email called spam. While the torrent of these unwanted emails has multiplied in size to intolerable levels over the past decade, more and more developers have joined the battlefield against spammers, who have caused billions of dollars in losses among corporations, and have taken hours of precious time away from Internet users due to the burdens of wading through flooded electronic mailboxes. Even with the current efforts used to block spam, such as government legislation, safe-sender lists, and spam filters, spam is becoming increasingly complex, and can now bypass the current multi-faceted defense scheme, requiring the need for new strategies.

## Current defense strategies

### Bayesian word distribution filters

A major technique used to eliminate spam involves training the computer to determine the difference between wanted email and unwanted email, using a computer program called a Bayesian word distribution filter (see

Figure 1), which analyzes messages as they are received. Initially, the user gives the spam filter a collection of categorized spam and legitimate emails. The program analyzes these messages and records different words, phrases, and other characteristics, called tokens, of each message in a database. After the initial training is complete, the program can calculate the probability that any given token in its database

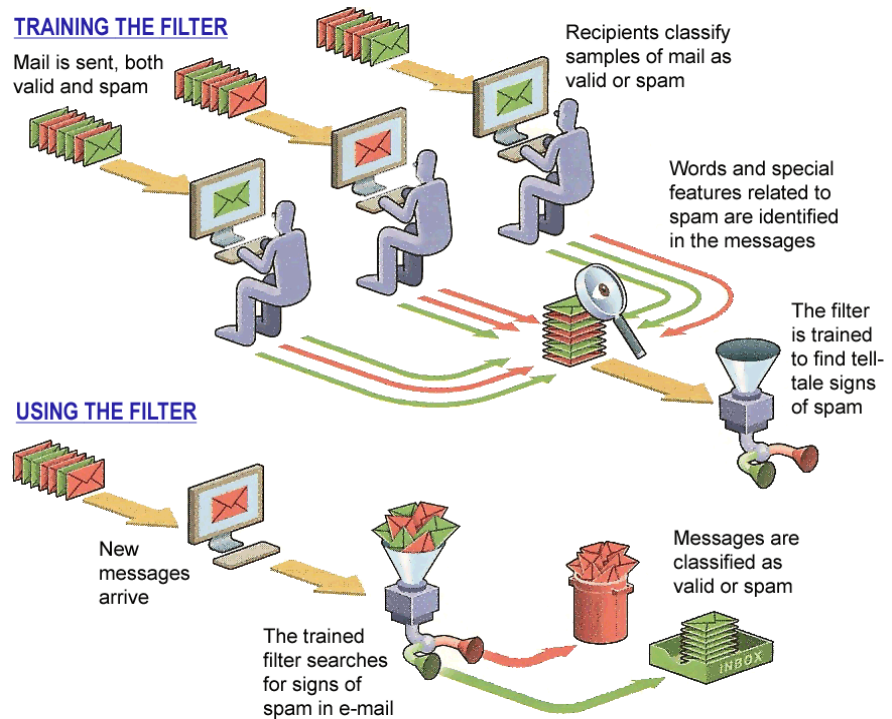


Figure 1: Training and using a typical Bayesian spam filter (46 Goodman).

<b>Combined Score: 100% (0.999741)</b>			
Internal ham score (*H*): 0.00025579			
Internal spam score (*S*): 0.999738			
# ham trained on: 1307			
# spam trained on: 2144			
<b>21 Significant Tokens</b>			
<b>token</b>	<b>spamprob</b>	<b>#ham</b>	<b>#spam</b>
'subject:Open'	0.116861	6	1
'commodities'	0.155172	1	0
'paper'	0.224715	11	5
'to:2**1'	0.621205	7	19
'header:Message-ID:1'	0.671762	380	1276
'header:MIME-Version:1'	0.720529	417	1764
'system'	0.75347	36	181
'to:no real name:2**1'	0.79741	5	33
'subject:-'	0.7978	26	169
'header:Received:1'	0.804992	69	468
'invest'	0.830941	1	9
'to:addr:[removed]'	0.863217	127	1316
'virus:src="cid:"'	0.866724	22	236
'to:addr:[removed]'	0.888812	12	159
'signals'	0.908163	0	2
'subject:Pre'	0.908163	0	2
'timing'	0.930965	1	25
'trading'	0.957622	1	42
'subject:Watch'	0.958716	0	5
'subject:Stock'	0.980349	0	11
'equity'	0.991803	0	27

**Figure 2:** A portion of a report generated by SpamBayes, an advanced Bayesian spam filter, during an analysis of an incoming spam message. Note: “ham,” the opposite of spam, denotes “good” email (self-created screen capture of SpamBayes).

is an indicator of spam by using the number of legitimate and spam emails associated with the token.

When a new email arrives, the program calculates the likelihood that the incoming message is spam by examining each word and phrase in the message and retrieving from the database the probability that it is associated with spam (see Figure 2). If the filter concludes that the message is spam, the email bypasses the inbox and is usually automatically deleted or moved to a separate “spam” folder. Emails that are incorrectly identified by the filter can be manually categorized by the user, and the program will integrate their characteristics into its database.

A key strength of a Bayesian spam filter is that it determines the probability that a message is spam based on previous emails it has processed. Since the characteristics of the legitimate emails and spam received by different people, such as the CEO of a corporation or a

high school student, can differ widely, the personalization aspect of this type of filter can yield very accurate results. However, one drawback is that the Bayesian spam filter is not a standard feature of most email clients, and must be manually installed by the user. Thus, most beginning Internet users do not take advantage of this technology.

Although Bayesian spam filters can correctly identify more than ninety percent of messages that pass through them, this number is simply too low. The cost of even a single important message mistakenly marked as spam by a filter can be detrimental because most users do not look at their “spam” folders, and the message may even have been automatically deleted.

### Other spam filters

Non-Bayesian spam filters used in many large email servers cannot be trained for each individual. Instead, they use a common database of probabilities that words appear in spam to determine whether an email is legitimate. This decreases the accuracy of the filter.

Both Bayesian and non-Bayesian filters can examine other aspects of incoming messages to find additional indicators of spam – for example, self-executing code and forged message headers. Ninety-five percent of spam messages contain a URL (universal resource locator), a link to another webpage; therefore, if a message does not contain a URL, the probability that it is spam decreases. Although spammers have employed new techniques over the years, such as substituting numbers and symbols for letters, as in “che@p pr3sc1pt10ns” (“cheap prescriptions”), or have stealthily broken words using HTML (hypertext markup language), as in “f<b></b>ree” (appearing as “free”), spam filters have quickly adapted to recognize these new “words” as characteristics of spam.

### **Safe-sender lists and blacklists**

Safe-sender lists, another technique used to fight spam, can help identify wanted emails by checking the “from” addresses against a list of senders known to the recipient. However, spammers can circumvent these lists by forging message headers to appear as legitimate senders, such as by changing the “from” address to “support@paypal.com” (appearing as valid at first glance to customers of PayPal).

In addition, there are two types of blacklisting techniques designed to reduce spam; however, both methods are somewhat weak. The first type involves keeping a list of email addresses to block, and rejecting messages from addresses that appear on the list. But, this method is no longer practical because spammers can generate a new address randomly for each message they send. The second type, in which certain servers are blocked, has been evaded by the spread of many harmful programs that turn computers into “zombies,” allowing for spam to be covertly sent through users’ email accounts.

## What’s next?

As spammers better understand the key weaknesses of the different types of spam filters and adopt more elaborate methods to penetrate current spam-filtering technology, a question arises, “What’s next?” Three new techniques — challenge-response systems, DomainKeys, and Sender ID — could eventually turn the tide against spammers.

### **Challenge-response systems**

Email servers could be set up to challenge unrecognized senders to verify that they are human. CAPTCHAs (completely automated public Turing tests to tell computers and humans apart), usually in the form of an image containing distorted text (*Figure 3*), are a way to weed out automated reply systems set up in spammers’ email servers, since these



**Figure 3:** A simple CAPTCHA (The CAPTCHA Project).

puzzles can only be solved by humans. After an email message is sent to a server, an email containing a CAPTCHA could be automatically sent to the “from” address, asking to reply with the text contained in the image. Since most email sent by spammers contain forged “from” addresses, most spammers will not receive the challenge. Those that do receive the challenge will not spend thousands of hours to manually

solve a CAPTCHA for each of the millions of emails they sent. However, most legitimate senders do not want to be bothered with CAPTCHAs either, and organizations sending newsletters or other subscribed email will be unable to solve them due to the magnitude of emails they distribute. Additionally, challenges may be sent to random people if their email addresses match the forged “from” addresses chosen by spammers.

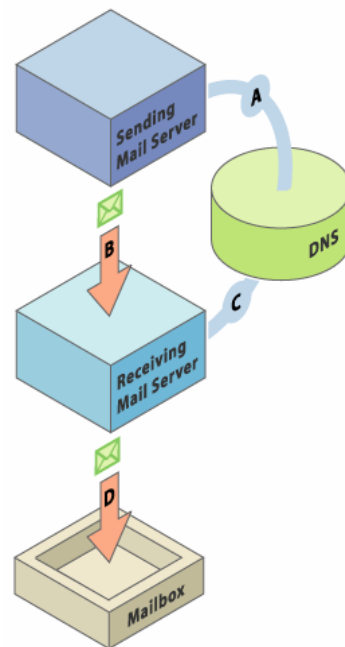
Another option is to require senders’ email servers to solve a computational puzzle that could take several seconds, making it impractical for spammers to send multitudes of emails. However, to be effective, this solution requires widespread adoption and the installation of special software on email servers to work the puzzles. An additional alternative to reduce the amount of spam requires senders to pay a small amount of money, called a micropayment, to have a message transmitted to the recipient. The money is refunded if the recipient approves of the message. Because of the sheer magnitude of messages that must be sent to make profits, the cost of spamming will no longer outweigh the benefits if spammers must pay even a cent to send each message, since the vast majority of recipients will not refund the money.

The most effective method is to use a layered system: if a message is not on a safe-sender list, it passes through a spam filter. If the spam filter is even slightly suspicious of the message, the sender automatically solves a computational puzzle if appropriate software is installed. Otherwise, the sender is given a choice between solving a CAPTCHA and paying a small, refundable amount of money.

### DomainKeys

Since most spammers alter the “from” address in the emails they send, the legitimacy of senders cannot easily be determined automatically. However, Yahoo! has developed a method for verifying the sender’s domain (indicated by the part of the email address after the “@”) to be legitimate, called DomainKeys. The domain owner creates a public key published in the DNS (domain name system) and a private key kept secret by the domain owner (Figure 4: A). Next, a signature, called a DomainKey, created with the private key, is automatically added to the message header when the email is sent (Figure 3: B). The recipient’s email server obtains the public key from the DNS record of the “from” domain (Figure 4: C), and uses that to confirm that the signature in the message header was created with the private key (Figure 4: D).

If the DomainKey is valid, then the message was sent from the server specified in the message header and the message has not been tampered with. If the DomainKey is invalid, one of these criteria was not met. However, if no DomainKey appears in the message header, then nothing can be learned from the particular message. Thus, this is not an



**Figure 4:** How DomainKeys works (Yahoo! Anti-Spam Resource Center).

anti-spam technique, but rather a technique to verify the sender of a message as legitimate.

A drawback of this technique is that because the message signature is created based on the content of the message, servers that alter the message in transit, such as mailing-list servers that add a small text advertisement to the end of messages that pass through them, will invalidate the DomainKey and produce a negative result when it is checked.

### **Sender ID**

Microsoft Corporation has developed a technology called Sender ID, similar to DomainKeys in that it verifies that emails are sent by the domain in the "from" address. This alternate method works by confirming that the server that sent the mail exists in a list of authorized servers listed in the DNS for the particular domain. Either Sender ID or DomainKeys should be used in combination with other spam-filtering techniques, such as challenge-response systems and Bayesian word distribution filters to ensure that a minimum amount of spam is received by users. Also, all of these technologies require widespread adoption to have a significant effect.

## The future of spam

The current spam-filtration and sender-verification technologies, enhanced by challenge-response systems along with DomainKeys or Sender ID, are projected to win significant victories in the war against spammers. However, says a retiring spammer, "so long as people click on spam and buy things advertised in their inboxes, spam will exist" (Spring).

## References

- Anonymous. "DomainKeys: proving and protecting email sender identity." Yahoo! Anti-Spam Resource Center. Yahoo! 30 Dec. 2005. <<http://antispam.yahoo.com/domainkeys>>.
- Anonymous. "Sender ID framework overview." Microsoft Safety. 17 Feb. 2005. Microsoft Corporation. 30 Dec. 2005. <<http://www.microsoft.com/mscorp/safety/technologies/senderid/overview.msp>>.
- Anonymous. "Telling humans and computers apart (automatically)." The CAPTCHA Project. 2005. Carnegie Mellon University. 29 Dec. 2005. <<http://www.captcha.net/>>.
- Avatar. "Evaluating spam costs and filtering techniques." Digital Silence. 25 Aug. 2004. 29 Dec. 2005. <<http://www.d-silence.com/feature.php?id=257&pn=0>>.
- Goodman, Joshua, David Heckerman, and Robert Rounthwaite. "Sopping spam." Scientific American Apr. 2003: 42-49.
- Mertz, David. "Spam filtering techniques." developerWorks. Sep. 1, 2002. IBM. 29 Dec. 2005. <<http://www-128.ibm.com/developerworks/linux/library/l-spamf.html>>.
- Spring, Tom. "Spam slayer: meet average-Joe Spammer." PC World.Com 21 Nov. 2005: 1.