Common Spam Filtering Techniques

PROCESS™ SOFTWARE

Every year, the amount of unsolicited email received by the average email user increases dramatically. According to IDC, Spam has accounted for 38 percent of the 31 billion emails sent each day in North America in 2004, up from 24 percent in 2002. Keeping pace with the quantity of spam is the quantity of filtering solutions available to help eliminate it. This document describes in detail how several of the most common spam filtering technologies work, how effective they are at stopping spam, their strengths and weaknesses, and techniques used by spammers to circumvent them.

Signature Matching

One of the distinguishing characteristics of spam is that there's a flood of it (most definitions of spam deliberately include the word "bulk"). Spammers send a copy of their spam message to every valid email account they can find. Signature matching takes advantage of this by automatically discarding every copy of a spam message as soon as it recognizes it as spam.

Vendors of signature matching anti-spam software maintain a large number of test accounts at ISPs and free email services such as Hotmail and Yahoo. They monitor these accounts closely, waiting for a spam message to arrive. When a spam message does arrive, the vendor quickly generates a signature for that message. Usually the signature is a string of 32 to 128 alphanumeric digits that is calculated based on the content of the message. This signature is added to a database of all of the spam signatures that the vendor has calculated.

Sites using the signature matching software are provided with a copy of this database by the anti-spam software vendor. This database is installed on their mail server, and is updated on a very frequent basis. When the site receives a message, it generates a signature for it using exactly the same method that their anti-spam vendor uses. To determine if the message is spam, the anti-spam software simply checks to see if the signature for the incoming message matches any of the signatures in the spam signature database. If it does, then the message is treated as spam.

Signature matching has an extremely low false positive rate[1], since the signature generation methods are deliberately designed so it's mathematically impossible for a "good" message to have the same signature as a spam message. It also has low system resource requirements, since both the signature generation routines and the database search are lightweight operations.

Unfortunately, signature matching also has very low spam detection accuracy[2] (the rate at which spam is correctly identified). Simple signature matching solutions are trivial for spammers to work around, and even more complex systems are easily fooled. In addition, signature matching has serious potential issues. The signature database is generated and updated remotely, with no input from a site's users or administrators. If the anti-spam vendor's master database system is compromised by a spammer, they can fill the signature database with the signatures of non-spam

[1] The false positive rate is the industry-standard metric used to measure the rate at which good messages are incorrectly identified as spam.

[2] Spam detection accuracy is the industry-standard metric used to measure how accurate an anti-spam filter is at correctly identifying spam. Generally, higher spam detection accuracy is obtained at the cost of a higher false positive rate. A good anti-spam filter will have an acceptable trade-off between the two metrics.

messages while removing signatures for spam messages. Because each site's copy of the signature database needs to be updated on a very frequent basis, preventing access to the vendor's systems with a denial of service attack for even a few hours will erode accuracy levels to almost zero.

The most obvious way for a spammer to sneak messages through a signature matching solution is to subtly change each message. Most of the software used by spammers to create and send messages can automatically insert random text into each message. Vendors of signature matching solutions have responded by developing more sophisticated signature generation routines that recognize and ignore random text and strings of words. In turn, spammers are now writing several versions of each paragraph of their messages. The newest generation of spam software randomly combines the different versions to create messages that are so substantially different that each requires a different signature.

Since signature matching solutions depend on generating a signature for a spam message before it becomes widespread, spammers can avoid having their messages filtered if they keep them away from sites where the anti-spam vendor keeps test accounts. These test accounts are usually at large ISPs and free mail services, so spammers can virtually guarantee that their messages will reach their intended destination as long as they avoid those sites. Even if spammers don't avoid ISP and free mail service addresses, they still have a window of opportunity until the anti-spam vendor sees one of the spam messages, generates a signature for it, and distributes that signature to all of the vendor's customers.

To keep the size of the signature database from growing so large it becomes unusable, signatures are removed as soon as the anti-spam vendor thinks that a particular spam message is no longer being sent. By sending messages in bursts with several hours between bursts, spammers can make sure the signature for their messages has been removed from the database, forcing the anti-spam vendor to repeat the signature generation and distribution process. During the time that takes, the spammers can freely send their messages to email servers running the vendor's software.

In the early days of spam, signature matching was a highly effective method for filtering spam. As spammers have increased their level of sophistication, the efficacy of signature matching anti-spam software has proportionately decreased.

Heuristics

Large numbers of spam messages tend to share the same set of characteristics. For example, most spam messages advertising mortgage refinancing contain phrases like "lowest interest rate" and try to disguise the word "mortgage" by spelling it "M*o*r*t*g*a*g*e" (or any of a hundred other possibilities). Heuristic filtering applies a set of rules to each incoming message to detect these spam-like features.

Each of the rules in a heuristic system has a value associated with it. To determine if a message is spam or not, the values for all the rules the message matches are added together. If the total value is greater than a threshold set by the user or system administrator, the message will be filtered as spam.

Simple heuristic filters use a small number of simple rules to look for obvious "bad" words and phrases, while filters that are more evolved use hundreds of rules and look for very complex features.

One of the most accurate spam filtering methods (with a consistent accuracy around 95%), heuristic filtering is also relatively fast. It's easy to install and configure, and is effective right out of the box without relying on a training period or constant updates over the Internet.

Heuristic filters can have a high false positive rate if the rules are not carefully constructed and tested before being applied to the system. It's very easy to construct a rule that is triggered by a large group of spam messages, but is also triggered by legitimate messages. Because the rules are static, they have to be updated frequently to counter new tricks developed by spammers.

The primary way spammers avoid having their messages caught by a heuristic filter is to word the messages in such a way that they aren't likely to trigger any of the rules used by the filter. Unfortunately for the spammer (and fortunately for the rest of us), it's very difficult to do this and still present a cohesive marketing message that will induce people to purchase a product or service.

If spammers can obtain a copy of the rules used by a heuristic system, such as they can for freeware solutions, they can deliberately craft a message that will bypass the rules. Spammers can even pay for a service that runs their messages through several of the most popular filtering products, and shows them how to alter the message to bypass the filters. Keeping the rules used by a heuristic system a bit of a secret, as well as updating them frequently, can significantly reduce the spammer's ability to engage in this sort of nefarious behavior.

Heuristic filtering is one of the best anti-spam filtering technologies currently available, when applied properly. It's easy to set up, has consistently high accuracy, and is difficult for spammers to circumvent if the rules are updated on a frequent basis.

Bayesian Filtering

Although they have been used for years to perform text classification, Bayesian filters are one of the newest technologies used for filtering spam. The filters "learn" the difference between spam and non-spam messages, and they continuously update their knowledge to stay current with new spam messages.

A Bayesian filter is taught the difference between spam and non-spam mail by looking at two large collections of email messages. One collection contains spam messages received by a site, and the other collection contains non-spam messages received by the same site. In essence, the filter picks each message apart into individual words. Based on a comparison of how often a given word appears in spam messages as opposed to non-spam messages, the filter calculates the probability that a message containing that given word is spam.

When a new message is received by the filter, it's pulled apart into individual words. The Bayesian filter chooses the words from the message that it thinks are the most interesting. (In this case, "interesting" means the words that are most likely to predict if a message is spam or not.) The

probabilities of each of those words appearing in a spam message are combined using Bayes' Formula, and the result is used to determine if the message is spam.

Bayesian filters have a very low false positive rate, since they carefully weigh both the spam and non-spam characteristics of every email message. A good email message that contains one spammy word, such as "Viagra", but also many non-spam words will not be accidentally classified as spam. The filters also "learn" about new tricks that spammers develop almost as fast as the spammers can come up with them.

Unlike most other filtering solutions, Bayesian filters require a training period to learn the difference between spam and non-spam email for a given site. During this time, there's likely to be a large number of false positives and false negatives. This can be avoided by pre-training the filter on large collections of spam and non-spam messages, but this can require several days of a system administrator's time for solutions that aren't capable of automatically training themselves.

Because of their need to perform a significant amount of string parsing, database access, and arithmetic computations, Bayesian filters have one of the highest system resource usage levels of any spam filtering solution. If a site's mail system is already heavily loaded, the installation of a Bayesian filter will overload the system and cause noticeable mail delays.

So far, spammers haven't managed to develop a method to consistently sneak their messages past a Bayesian filter. The most commonly attempted circumvention is to include random dictionary words in messages, hoping that there will be enough "good" words to get the message by the filter. That method rarely (if ever) works, since the random words are usually discarded by the Bayesian filter as unknowns. The only sure way to get a message past a Bayesian filter is to avoid using "spammy" words or phrases in the message. However, it's very difficult to sell Viagra without actually using some variation of the word "Viagra" in the message.

Bayesian filtering is an extremely accurate filtering technology for email accounts where good email has significantly different content than spam. The large memory, disk, and CPU requirements may make it unsuitable for some sites, but it greatly complements other filtering technologies that have high levels of accuracy.

DNS Blacklisting

One of the oldest forms of spam prevention, DNS blacklisting uses a centralized database to block all email from a host being used to send spam. The provider of the blacklisting service maintains the database, adding entries for hosts that are being used by spammers. Access to several of these databases is free, while others require a yearly fee for usage.

During an SMTP transaction, an email server configured to use a DNS blacklist will perform a DNS query on the host that is sending the message. Rather than performing the query against its own DNS server, the email server queries a DNS server provided by the DNS blacklisting service. Based on the information returned from the query, the email server will either accept or reject the incoming message.

The two primary benefits to this approach are its low system resource requirements and its ease of maintenance. The email server only needs to make an additional DNS query to use this filtering method – large amounts of CPU time and memory aren't required to scan the complete headers and content of an incoming message. Since the message is rejected during the actual SMTP transaction, the amount of system resources consumed by spam is reduced. A nice side-effect of this is that several software packages used by spammers will automatically remove addresses that are rejected by an email server, cutting down on the amount of spam received by the site in the future.

This technology is used by many sites because of its simplicity – enabling it requires only a few configuration changes inside most email server software. There's no additional software to install, no updates to download, and no regular maintenance required.

Despite its small footprint and ease of use, DNS blacklisting has several serious flaws that prevent most sites from being able to use it. By far the largest is the lack of granularity – either all of the mail from a given host is accepted, or all of it is rejected. Most blacklist service providers have a pre-defined set of rules a site must violate for it to be blacklisted. Spammers often hide behind the anonymity of large ISPs such as AOL or free email providers such as Hotmail, causing these services to be blacklisted. E-commerce sites, ISPs, and companies that deal directly with large numbers of email users can ill afford to perform a wholesale rejection of mail from ISPs and free email providers. In addition, legitimate sites are occasionally blacklisted either by accident or because a spammer forged messages that appear to come from the site. Once blacklisted, it's usually difficult to be removed from the blacklist database. Other technologies that identify spam on a per-message basis are much more acceptable to most sites for these reasons.

Because DNS blacklisting depends on being able to access a remote DNS server over the network, if a network link drops or the remote DNS server crashes the email server will have no choice but to accept all mail without checking to see if it's spam or not. Even if the remote DNS server is accessible, incoming mail messages will be delayed during periods of high network latency or when the remote DNS server is slow.

In the past, blacklisting domains was a partially political process. Blacklist service providers would blacklist any site that offended them (including competing service providers and sites that criticized them). Several high-profile lawsuits were filed by blacklisted sites, but none were successful. While the situation has stabilized recently, the potential for this sort of behavior still exists. Since a site that uses a DNS blacklist has no control over the sites that are blacklisted, they can quickly find themselves rejecting legitimate mail for no discernible reason.

Spammers use several basic techniques to circumvent DNS blacklists. The most common is to send spam from multiple "throw-away" host addresses. Usually, several people must complain to a blacklist service provider before a host is placed in the blacklist database. Several hours or even days can pass before a host that has been complained about is placed in the blacklist database. Meanwhile, the spammer can send millions of messages from that host. As soon as the host is blacklisted, the spammer purchases another host address for a nominal fee and the blacklist process must begin again.

A second technique is for the spammer to masquerade as a legitimate site, hoping that either they will escape being blacklisted or they will cause a legitimate site to be blacklisted. By causing legitimate sites to be blacklisted on a regular basis, spammers can reduce the accuracy of DNS blacklisting and force some sites to stop using it rather than lose important messages.

At best, DNS blacklisting can be used to identify and discard around 40% of the spam a site receives. As long as a site is willing to live with the possibility of legitimate mail being rejected by factors outside of the administrator's control, DNS blacklisting is a useful technology as long as it's used in conjunction with other spam filtering techniques.

Challenge/Response

Virtually every spam message is generated and sent by an automated software utility (spammers don't sit in front of a computer in their basement clicking the "Send" button as fast as they can). Challenge/response systems take advantage of this by forcing email senders to prove that they're human through some sort of test (the "challenge").

When an email message is sent to an account protected by a challenge/response system, it is placed in a holding area and a message containing a challenge is sent back to the sender. Usually this challenge message contains a brief explanation of why it was sent, and includes a link to a web page where the actual challenge will be presented. If the message sender passes the challenge, the original message is released from the holding area and sent to the intended recipient. If the message sender doesn't pass the challenge, then the original message is deleted after a specified period of time.

For a challenge to be effective it has to be something that humans can do easily but computers cannot. The most common type of challenge consists of an image of distorted text. To pass the challenge, a human must type the text correctly.

In theory, challenge/response is an ideal spam filtering solution. There are no false positives, and no spam messages manage to sneak through the system (if a spammer has to manually pass a challenge for each message sent, the outgoing spam rate will be cut from millions of messages an hour to a couple dozen). There are very low system resource requirements, since no CPU-intensive pattern matching is required. And best of all, spammers can try to disguise their message and it will still be identified as spam.

Unfortunately, challenge/response causes more problems than it solves. For inexperienced computer users or those with visual handicaps, the challenges are completely unsolvable. Even those who are physically able to solve the challenges will often choose not to do so because they view it as an unacceptable irritation. Likewise, automated email that a user would want to receive (travel confirmations, online purchase receipts, etc.) will be trapped by the challenge/response software and never delivered.

Challenge/response systems also create mail delays that are unacceptable, especially for corporate users who deal in time-sensitive information. Between the time the original message is sent and received, a challenge message has to be generated and delivered to the sender, the sender has to read

the challenge message and take whatever steps are required to solve the challenge, and the original message has to be released from the holding area and delivered to the intended recipient. Even under optimal conditions, this usually takes between 15 to 30 minutes. In sub-optimal conditions (also known as "lunch hour"), this process can require several hours.

A system that supports whitelisting can alleviate these issues to some degree, but such a system is easy for spammers to circumvent. If a spammer can guess a whitelisted address (which wouldn't be too hard to do if the user associated with the whitelist conducts any sort of online transaction), he can forge that address in his messages so they sail right by the anti-spam software. And best of all, the challenge/response system provides spammers with instant feedback in the form of a challenge message if they try an address that isn't whitelisted.

It even turns out that the distorted text images aren't that much of a challenge anymore. Researchers at UC-Berkeley have developed a software system that can accurately read even the most distorted characters from an image file. Vendors of challenge/response systems have responded by adding background distortion to their challenge images, but this often makes them so challenging that even real humans can't solve them.

Even if spammers don't want to go to the trouble of putting together a high-end character recognition system to defeat challenge/response, they can pay real humans to do it for them. In developing countries, a human can be hired for as little as 40 cents a day. A trained human can consistently solve challenges in ten seconds, making it cost a fraction of a penny per message to guarantee it lands in a user's Inbox.

Some anti-spam researchers have even suggested that porn fiends can be used to solve challenges at no cost to the spammer. After every two or three images are displayed, a challenge is presented that must be solved before more images will be displayed. The challenge is actually one that was presented to the spammer by anti-spam software, which has been cross-linked to the "free" porn site that the spammer runs.

In an unusual twist, spammers are starting to send large numbers of messages that purport to be from a challenge/response system. When the recipient visits the URL, they are presented with a marketing message rather than a challenge.

Challenge/response is an attractive solution in theory, but in practice it disrupts email more than spam does. In an anti-spam solution that uses another filtering method for the bulk of messages, challenge/response could possibly play a small role in the case of messages that the primary filtering method isn't sure about.

Conclusion

A large number of anti-spam technologies are commonly available today, and several more are under development. No single filtering method is a panacea for the spam problem, since each has

weaknesses that spammers can exploit. The best solution is to use different, overlapping methods in parallel with one another. While a spammer may be able to craft messages that can sneak by one type of filter, it's virtually impossible to write a message that can evade multiple filtering methods.

At the same time, it's important not to use too many filtering methods at the same time. Each one has a noticeable effect on email server performance. After messages have passed through two or three filtering methods, the additional accuracy imparted by additional methods is going to be minimal.

Method	Pros	Cons
Signature Matching	• Low false positive rate • Minimal system resource requirements	• Low spam catch rate • Easy for spammers to evade • Requires constant access to anti-spam vendor's systems • System reacts to spammers, instead of proactively discarding spam messages
Heuristics	• Very high spam detection accuracy • Difficult for spammers to circumvent unless they acquire a copy of the rules • Moderate system resource requirements	• Can have a high false-positive rate if rules are poorly authored
Bayesian	• High spam detection accuracy and low false positives when trained properly • "Learns" spammer tricks and uses them against the spammers	• Extremely high system resource requirements • Requires training period to learn the difference between spam and non-spam messages
DNS Blacklisting	• Very low system resource requirements • Complements other antispam filtering methods • Potentially high false positive rate	• Relatively low spam catch rate
Challenge/Response	• Low system resource requirements	• Trivial for spammers to circumvent Induces message delivery delays that most sites will find unacceptable • Can't deal with legitimate automated messages (e-commerce invoices, mailing lists, etc.)

- Effectively discriminates against visually impaired users

About PreciseMail Anti-Spam Gateway

PreciseMail Anti-Spam Gateway is an enterprise software solution that eliminates spam, phishing and virus threats at the Internet gateway or mail server. It has a proven 98% spam detection accuracy rate out-of-the-box without filtering legitimate messages. PreciseMail Anti-Spam Gateway has a highly sophisticated filtering engine is based on a combination of proven heuristic, DNS blacklisting, and Bayesian artificial intelligence technologies, which automatically learn how to separate spam messages from legitimate email. As a result, PreciseMail Anti-Spam Gateway can determine whether email is spam instead of passively reacting to known spammers by creating rules that block them after a spam attack occurs.

About Process Software

Process Software has been a premier supplier of communications software solutions to mission critical environments for twenty years. We were early innovators of email software and anti-spam technology. Process Software has a proven track record of success with thousands of customers, including many Global 2000 and Fortune 1000 companies.

PROCESS™ SOFTWARE

U.S.A.: (800) 722-7770 • International: (508 879-6994 • Fax: (508) 879-0042
E-mail: info@process.com • Web: http://www.process.com/