# The history and future of SMTP

## SMTP's adaptations to a hostile internet

Kirk Strauser

SMTP is an abbreviation for "Simple Mail Transfer Protocol", and is the standard internet protocol for sending email from one system to another. Although the word "simple" belies the inherent complexity of the protocol, SMTP has proved to be a remarkably robust, useful, and successful standard. The design decisions that made it so useful, though, have given spammers and infectious code an easy way to spread their unwanted messages. Its recent evolution reflects the tug-of-war between those unsavory players and the administrators who want to protect their systems and their users.

### Early history

When Jonathan Postel wrote the SMTP definition RFC 821 in 1982, the internet was minuscule in comparison with today's pervasive mix of commercial, governmental, and private interests. At that time, it mostly comprised a small collection of military installations, universities, and corporate research laboratories. Connections were slow and unreliable, and the number of hosts was small enough that all of the participants could recognize each other. In this early setting, SMTP's emphasis on reliability instead of security was reasonable and contributed to its wide adoption. Most users helped each other by configuring their mail servers as "open relays". That meant that each cooperative host would accept mail meant for other systems and relay it toward its final destination. This way, email transfer on the fledgling internet stood a reasonable chance of eventual delivery. Most administrators were happy to help their peers – and receive their help in return.

Spam has existed since at least 1978, when an eager DEC sales representative sent an announcement of a product demonstration to a couple hundred recipients. The resulting outcry was sufficient to dissuade most users from repeating the experiment. This changed in the late 1990s: millions of individuals discovered the internet and signed up for inexpensive personal accounts and advertisers found a large and willing audience in this new medium.

### Spam becomes a problem

The helpful nature of open relays was among the first victims of the spam influx. In the young commercial internet, high-speed connections were prohibitively expensive for individuals and small businesses. Spammers quickly learned that it was easy to send a small number of messages – with recipient lists thousands of entries long – to helpful corporate servers, which would happily relay those messages to their targets. Administrators noticed sudden spikes in their metered service bills (and in the number of complaints) and realized that they could no longer help their peers without incurring significant monetary costs and bad will.

### First steps to secure the internet

Although the nature of the problem was clear, the solutions were not. The SMTP standard, which was designed with reliability as a key feature, had to be re-implemented to purposefully discard certain, recognized messages. This was a foreign idea and no one was sure how to proceed.

The first step was to close the open relays. Administrators argued loudly, and at great length, whether this was a necessary move, or even a good one at all. In the end though, it was universally agreed that the trusting nature of the old internet was dead, and in fact harmful in the current setting.

Some users took this idea a step farther and decided that they would not only close their own systems, but would no longer accept messages from other open relays. They eventually began to share their lists of those relays with peers by adding specially formatted entries to their domain name servers and allowing their neighbors to query their servers for this data. This was the beginning of the first "DNS blackhole lists", and they were highly controversial. For example, administrators debated whether it was acceptable to actively test remote servers to see if they were open relays, and discussed which procedures a system administrator should follow to remove his or her host from the list after correcting the problem.

The first victims of "collateral damage" were those whose mail servers were blocked through no fault of their own. This often happened when over-zealous blacklist operators added entire blocks of addresses to their lists, rather than just the offending addresses. As one group of operators argued that the lists should err on the side of caution to prevent these problems, others believed that this would put extra pressure on the open relay administrators. In one form or another, this debate continues.

## New threats

Huge numbers of people with very little computer-security experience came online, often with increasingly cheap, permanent high-speed connections. As a result, a new epidemic spread across the internet - most visibly as email worms. They infected poorly secured computers which then became the transmitters for new copies of those worms. Many of these propagate through the popular email clients on Microsoft Windows systems and move outward by emailing copies of themselves to people in the infected computer's address book. Many such infections are noticeable because they can overwhelm a machine and its internet connection to the point where both become useless to their end user, who then typically pay a business, or get a knowledgeable friend, to remove the worm.

There are more insidious infections which spread amongst computers rapidly. They then lie dormant to avoid drawing attention to themselves and wait for instructions from another system. A "botnet" is a collection of computers so compromised. Spammers often use botnets as a widely distributed means for sending large amounts of email.

## Fighting back

A recent and popular response to these problems is sender authentication. That is, many mail servers now look for proof that a computer attempting to send email to them is actually authorized to do so. For example, Sender Policy Framework (or SPF) is centered around another specialized DNS record that lists the servers authorized to transmit email from a given domain. The administrator of example.com may list "smtp.example.com" and "mail.example.com" as the outbound mail servers for that domain. When an SPF-aware server receives a message from a user with an example.com email address, it compares the name of the computer attempting to send that message with those names. If it isn't on the list, then the message can reasonably be assumed to be a forgery and may be discarded. Several proposals exist that are similar to SPF, such as Yahoo!'s DomainKeys, but all work in essentially the same way.

Another common measure is simply to enforce the SMTP definition and reject messages that do not adhere to it. This is highly effective because few, if any, worms or spam transmitters bother to comply with the standards. They often take shortcuts when generating the email address that a message claims to originate from, or lie about their own identity. Some seem to completely ignore the standard in hopes that the receiving system will blindly process their load anyway. The methods of enforcing the protocol must be implemented incrementally, though, as many old but legitimate mail servers may also fail to meet some of the more pedantic requirements. An old rule of networking is to "be liberal in what you accept". Sadly, spammers seem to be on the brink of making that impossible.

One of the positive side effects of sender authentication and standards enforcement is that email senders are being compelled to correctly identify themselves before they are allowed to transfer their messages. New DNS blackhole lists, able to narrowly identify specific senders, will be possible once a critical mass of servers have implemented such measures. This solution should neatly avoid the old problem of collateral damage, as well as greatly reducing the scope of the blackhole lists themselves.

Regardless, some spam and worms will always make it through the tightest of filters. To this end, a new class of intelligent, self-learning filtering software does a good job of identifying the remaining unwanted messages. Good, free antivirus programs also perform well at removing worm-infected messages before they can reach vulnerable email clients.

One of the newer and more exotic approaches is known as greylisting. The idea is simple: receiving mail servers make senders wait for a small amount of time before they are allowed to transmit email to a recipient they've never sent to before. This serves two purposes. First, very few worms, spam senders, or botnet machines actually have the patience to try again later (or the resources to remember which addresses should be retried). If their first attempt at delivering a message fails, they give up and move on to the next destination. Second, by increasing the effective length of time it takes for a spammer to send a message, a mail server also increases the chances that a DNS blackhole list will add that rogue server before it can deliver that message. Few methods can compete with the simple elegance of greylisting, and it offers to many frustrated administrators the hope that the war against unwanted email can be won.

Finally, some administrators have responded to the overwhelming loads which are sometimes sent by botnets by blocking certain operating systems. Almost no one runs a legitimate mail server on Window 98, for example. Therefore, configuring a firewall to block incoming SMTP connections from Windows 98 machines (assuming all of your desktop clients use newer version of Windows, or Mac or Unix desktops) can reduce the number of unwanted messages from hijacked computers.

## The future

SMTP has a long and illustrious past. It's one of the "killer applications" that led to the explosive growth of the internet. From love letters to stock transactions to family photos, countless users send an endless variety of messages to each other every day. Email in its current form is going to be around for a long time, but will likely undergo a series of incremental updates. For example, client authentication (which didn't exist when the SMTP RFC was written) has almost completely replaced open relaying, and some mail servers now use SSL certificates to verify another server's identity.

However, the future of SMTP depends largely upon those who abuse it. It currently provides a reliable, fault-tolerant system of email delivery. Any changes are likely to work against this reputation, as they would add to the complexity of the protocol. Several proposed alternatives have come and gone, and there are no widely accepted proposals that stand a reasonable chance of coming into common use. Only time will tell...

## Bibliography

[1] RFC 821 – Simple Mail Transfer Protocol (http://www.faqs.org/rfcs/rfc821.html)

[2] Reaction to the DEC spam of 1978 (http://www.templetons.com/brad/spamreact.html)

[3] Wikipedia entry on DNSBLs (http://en.wikipedia.org/wiki/DNSBL)

[4] Sender Policy Framework (http://spf.pobox.com/)

[5] DomainKeys: Proving and Protecting Email Sender Identity (http://antispam.yahoo.com/domainkeys)

## Copyright information

## About the author

Kirk Strauser has a BSc in Computer Science from Southwest Missouri State University. He works as a network application developer for The Day Companies, and runs a small consulting firm (`http://www.strausergroup.com/`) that specializes in network monitoring and email filtering for a wide array of clients. He has released several programs under free software licenses, and is active on several free software support mailing lists and community websites.