Collateral Damage

Consequences of Spam and Virus Filtering for the E-Mail System

Peter Eisentraut
credativ GmbH
peter.eisentraut@credativ.de

Abstract

This paper takes a critical look at the impact that contemporary spam and virus filter techniques have on the stability, performance, and usability of the e-mail system.

1 Introduction

This is the year twelve of the Spam era.[1] By most counts, more than half of the current e-mail traffic on the Internet is due to spam or e-mail-borne viruses [1]. The computing community has constructed an impressive toolkit of anti-spam and anti-virus measures which most regular e-mail users apply or have applied on their behalf lest they be bombarded with e-mail junk.

This is also the year twenty-four of the SMTP e-mail era.[2] The "simple mail transfer protocol" nowadays governs virtually all electronic mail communication on IP networks, having obsoleted several alternative protocols over the years. Two important properties of SMTP were simplicity—it was easy for heterogeneous systems to participate in the message exchange—and reliability—it was ensured that messages would be delivered or the delivery failure be reported.

But a more thorough consideration will reveal that a strict SMTP implementation alone no longer stands a chance to participate successfully in the e-mail network of today. There are additional protocols and conventions stacked on top of it that are the result of the behavior of spam filters, virus scanners, and other defense mechanisms. The behavior of these systems is not codified anywhere, it varies between sites and over time, and the existence is usually not even announced. Users of this network of hosts of ill-defined protocol conventions, mutual distrust, and hostility are faced with a new set of challenges to get their e-mail through. This paper analyzes these problems.

2 Filtering Techniques

Most sites that want to stand a chance to participate reasonably in the e-mail exchange equip their mail servers with a number of filter routines to weed out junk e-mail. While this is an unfortunate fact, it is clear that running a mail server without spam and virus filters is no longer feasible today. Not all filtering

[1] Legend has it that the first spam was sent in 1994 [1].

[2] RFC 821 appeared in 1982.

techniques, however, have equal merit. Some have lost their impact over the years, some are frequently misconfigured, and some simply cause more harm than good. This section will show a number of e-mail filtering techniques that have shown to be troublesome.

2.1 DNS Blackhole Lists

DNS blackhole lists (DNSBL) were the first technique invented specifically to fight junk e-mail [1]. Lists of hosts, first IP addresses, later also host or domain names, that have appeared in connection with junk e-mail are published via the DNS system. Mail servers all over the Internet can query these lists to reject connections from hosts that are known to send out (or relay) spam.

The first public DNSBL, the MAPS Realtime Blackhole List (RBL), had a rather strict policy for adding entries. Nominations were inspected manually, nominated sites were contacted and given a chance to react to the problem before a listing would be added. If you thought that MAPS was a trustworthy organization, you could sensibly block e-mail from all hosts listed at MAPS.

Two things have happened in the meantime. First, the MAPS RBL no longer exists as a free service. At first, access was restricted to paying customers, and later the entire operation was bought by Kelkea, Inc., which in turn has been bought by Trend Micro in the meantime [2]. There are now dozens of alternative DNSBL providers available. Second, both the Internet and the spam problem have grown significantly. It is no longer manageable to inspect all listing nominations manually. Therefore, most of the current DNSBLs rely on some kind of automatic listing (and delisting) process. This leads to some problems:

- Temporary misconfigurations are not treated discriminatorily. They can cause immediate listings which are slow to be removed. This way, the entire customer bases of large ISPs are occasionally blocked.

- Most spam nowadays is sent over dial-up accounts. The IP address of a dial-up account changes every day or so. A blacklist maintained over DNS, which typically has a propagation delay of approximately one day, is therefore useless for tracking rogue dial-up accounts.

The RBL was as much an education program as it was a spammer blacklist. At that time, Sendmail was the dominating MTA program on the Internet and it was unfortunately configured as an open relay by default. Nowadays, all common MTA products are secure against relaying by default, so if there is a configuration problem it is more or less intentional. The problem is that most of the junk e-mail is no longer sent via common MTAs but via zombies behind dial-up accounts.

These points do not mean that DNSBLs are useless. An open relay is an open relay after all. And scanning the e-mail body for mentions of listed host names (URLBLs) has shown itself to be useful. It means, however, that the correlation between a DNSBL listing and an actual junk mail problem is no longer strong. If the purpose of DNSBLs is to fight junk mail, rather than to push through agendas about proper system configuration, DNSBLs can no longer by universally trusted. Therefore blocking e-mails simply because of a DNSBL listing is not appropriate anymore. Countless reputable sites on the Internet continue to do that nevertheless.

If one considers a DNSBL listing to indicate an increased *likelihood* that the sending host may be connected to a junk e-mail problem, then factoring in this likelihood with other spam indicators, as is done in weighted scoring systems such as in SpamAssassin, continues to be useful.

An additional point is that DNS as a protocol is not secure. It is fairly easy to influence the system in such a way that a user is presented with wrong information. This could be used by criminally minded parties to launch denial-of-service attacks by presenting faked DNSBL listing data to an e-mail receiver, or to bypass DNSBLs by hiding such listing data. While this has not been an actual problem to any noticeable degree so far, building reputation services on DNS still presents a potential trouble spot.

2.2 Bounce Messages

In the old days, it worked like this: Host A wants to transmit an e-mail message to host B. Host B checks whether the message is acceptable, then acknowledges the receipt or sends a rejection message. As spam and virus filtering became more important, the acceptability checking became more and more involved, to the point that host A started to time out before host B could finish the checks. So in the new days, it works like this: Host A wants to transmit an e-mail message to host B. Host B immediately accepts that message (after a minimum of checking). Later, host B checks whether the message is acceptable. If it's not, then host B sends an e-mail to inform the original sender that the message was rejected. This is in principle already a protocol violation, but would rarely have any practical impact for the end users.

The problem is that once host B has (apparently) accepted the message, host B no longer has any reliable information about the original sender of the e-mail message. As long as the connection to host A is still open, sending the error to host A is a good bet for reaching the sender. (Granted, in complex relay schemes, host A might be just as much at a loss about the original sender as host B, but ordinarily, host A is the mail server at the ISP of the sender and therefore has a pretty good idea about where the message came from.) But the sender address in the envelope or the content could be the actual sender, or it could be misconfigured, or it could be deliberately faked, as would be the case for spam, so that after the connection is closed, the sender cannot be reached anymore. In other words, the new days approach *does not work.*

So what to do? One popular course of action is to ignore the problem and send the rejection messages anyway. Even popular open-source e-mail filter packages like Amavisd-new [3] continue to ship with a default configuration to that effect as of this writing[3]. Assuming that most junk e-mail uses fake sender addresses, and assuming further that the spam and virus filters are reasonably accurate and have few false positives, then almost all of these rejection messages go to the wrong person. Given that at least 50% of all e-mails are junk e-mails, and assuming that filters detect at least 80% thereof, if only one in ten sites would enable these rejection messages, e-mail traffic on the Internet would rise by about 5%. Add to that the impact that these misguided rejection messages have on their actual recipients, this mechanism is not only useless, both for the senders and the recipients, but wastes "common" Internet resources,

[3] version 2.3.3

and is hostile to innocent users. This has in turn led to a wave of anti-anti-spam measures that stop the bogus bounces, and some users now reject bounces altogether, further reducing the reliability of e-mail communication.

People who really trust their filters then simply discard messages classified as junk e-mail without any notice. But this approach is rarely appropriate. Putting suspected junk messages in a quarantine area for manual inspection works reasonably well on balance, but is hardly manageable in larger organizations [1]. There are also significant privacy concerns with this approach.

What is rarely considered is that the two problematic factors in the old days approach should simply be fixed. First, SMTP clients that time out too fast: Fixing the actual e-mail client programs would be hard to achieve because that end of the bargain is influenced by unresponsive manufacturers and inexperienced users. Most of the time, however, the client in these transactions is another MTA program, which can easily be reconfigured to support longer timeouts. Second, checking the message on the receiving host takes too long: The internals of many of the e-mail filtering applications are quite frankly a mess. Assembled during a period where new filtering techniques appeared by the week, they lack proper design, are overloaded with features, and consequently do not perform well. More robust and streamlined implementations could easily outperform the toolkits of today to make e-mail filtering at the point of delivery possible again.

2.3 Greylisting

The amazing fact about greylisting is that it still works at all. Greylisting relies on the fact that spamming software and in particular mailing software installed on zombies, does not retry sending after receiving a temporary failure reply from the recipient. Greylisting is extremely effective; in my experience it can block between 80% and 90% of all junk e-mail. The reason why so few spamming software makers have reacted and added a sending queue to their software can only be assumed to be that so few sites use greylisting.

The problem with greylisting is not so much that is hinders the e-mail traffic—the delay is usually about 15 minutes and the additional traffic is minimal—but that it's easy to get the configuration wrong:

- The concept sounding so simple, many of the early greylisting implementations are hack jobs that break easily and are full of security holes.
- Distributing the mail server load on more than one machine causes various kinds of problems:
  - If it is done on the sender side, each new delivery attempt will appear to come from a different IP address which will again be greylisted. This can usually be circumvented by greylisting not the IP address but, say, the entire class C network.
  - If the load spreading is done on the server side, it is important that the greylisting database is shared between all nodes, otherwise the client is greylisted again if the next delivery attempt is serviced by a different node.

- A number of sites do not have well-functioning SMTP server implementations that schedule a new delivery attempt if the first one failed with a temporary error [4]. A list of these sites needs to be collected and whitelisted.

- Some mailing list software sends out each e-mail with a unique sender address [4]. Such sites can only be reasonably greylisted if the sender address is not part of the lookup key.

- If users are expecting time-critical e-mails (say, from Internet auctions), then they need to be whitelisted. If this is unmanageable, greylisting cannot be used.

- If more than one hop in the delivery chain is configured to use greylisting, the delivery time grows superlinearly. This should be avoided.

Even though the concept sounds simple, a properly functioning greylisting implementation for a mail server that is to service many different kinds of users is very difficult to get right. It is better to stay away from it if the user population is too diverse.

2.4 SPF

The Sender Policy Framework (SPF) [5] and its cousin Sender ID [6] are two more recent developments in the fight against spam. The owner of a domain announces through a DNS record over which hosts e-mail from that domain is allowed to be sent. If a recipient gets e-mail from that domain over a different host, the e-mail should, under the SPF theory, be rejected. The snail mail analogue of SPF would be the post office saying that letters with return addresses in Berlin may only be dropped in mailboxes in Berlin. (Note that this does not offer any satisfactory explanation for what return address to write if you send a postcard while on vacation in Hamburg.)

SPF does not, in fact, hinder much of any spam. SPF could only work if all or at least most sites on the Internet used it. Otherwise, spammers who wish to equip their spam with fake sender addresses can simply pick a domain which does not publish SPF records. This can obviously be automated with ease, so spammers are not bothered by SPF at all. Even if all reputable sites on the Internet used SPF, new domains are a dime a dozen. The actual junk mail fighting task would then be to quickly identify those domains and publish a list of them, but DNS Blackhole Lists already do that.

SPF is also supposed to prevent phishing. Note, however, that SPF only checks the envelope sender address, which most end users of e-mail never see at all, so the usefulness of SPF against phishing is nearly zero. The related approach Sender ID works similar to SPF but checks the Purported Responsible Sender (PRS) address instead, which is taken from the headers of the e-mail contents. This is the address that e-mail users do see, so Sender ID does seem useful against phishing. But Sender ID has not reached widespread acceptance because it is patent encumbered and has a restrictive license [1].

What SPF does prevent to some degree is that a certain domain is abused as a fake sender address in junk e-mail. Note that the SPF web site [5] is titled, “A Sender Policy Framework to Prevent Email *Forgery*” (my emphasis). It doesn’t prevent it, of course, but it does reduce it. One might suspect that this is the actual reason why certain ISPs apply this technique.

On the flip side, SPF breaks the e-mail protocols. Forwarding no longer works, because the forwarding host might not be registered as a valid sending host for the domain. The Sender Rewriting Scheme (SRS) is supposed to fix that but has not been widely implemented. Users are locked into the mail servers of their e-mail providers. If the mail server is misconfigured (see section 2.1 for an example) or unavailable (see section 2.5 for an example), the user cannot send e-mails anymore. People who have their own e-mail domains are faced with a new set of issues: Does the domain hoster publish SPF records? Is the domain owner able to publish their own SPF records? Is the domain owner forced to set up his own mail server, or alternatively, is that option available? Moreover, SPF relying on DNS as the distribution protocol, it is susceptible to the same security problems as DNSBLs, explained in 2.1.

To summarize, SPF is a means for large ISPs to control how users route their e-mail, it creates a number of problems for ordinary e-mail users, but does not solve any actual problems for them.

2.5 Blocking Port 25

Initially, spam was send through ordinary ISP mail servers. When spam began to be frowned upon, spammers sought out mail servers with insecure configurations. As those disappear, spam is nowadays mostly sent through so-called zombies, ordinary PCs that have been taken over by a virus. Spammers, in cooperation with virus authors, command networks of thousands of cracked PCs to relay their junk e-mail.[4] Blacklists have trouble keeping up with this development because most of these vulnerable PCs are behind dial-up accounts which change their IP address every day. And even if the IP addresses could be tracked, spammers could simply switch to the next set of a thousand zombies.

Over the last few years, ISPs have begun blocking the TCP port 25 for their dial-up customers. This means that dial-up users can no longer connect to port 25 on arbitrary hosts but have to route all e-mail through the designated mail server of their ISP. Compromised PCs would no longer be usable as zombies for junk mailing.

For the “average” e-mail users, this doesn’t make a difference, and the fact that an ISP blocks port 25 might even slightly increase their security indirectly, as their hosts are no longer an attractive target for installing “zombieware”. Many e-mail users, however, wish to command e-mail accounts other than the one offered by their ISP from their machines. People use alternative e-mail providers, use office e-mail accounts while working at home, or host entire company networks behind DSL lines. Simply blocking port 25 is therefore not an acceptable measure. To offset this problem, ISPs might then offer that all dial-up customers can relay outgoing mail through the ISP’s mail server, no matter what domain. (If the ISP’s mail server properly checks the connecting IP address or requires authentication, this is not an open relay and therefore acceptable.) This, however, will still not allow customers to bypass the ISP’s mail server for other reasons, such as the ISP’s mail server being misconfigured or the desire

[4] Impressive examples were related by Hauptkommissar Frank Eißmann, Dezernat für Computerkriminalität, Landeskriminalamt Baden-Württemberg at [7].

to use other mailing software. It's also fundamentally incompatible with the SPF system. These two measures combined are a particularly powerful way to control and restrict how users send e-mail.

The only acceptable measure if an ISP blocks port 25 is to give every customer the option to unblock the port without any questions.

It can be expected that even more ISPs will begin to block port 25 by default in the future. Government groups such as the "London Action Plan" might even endorse the measure.[5] Consumers should be wary that ISPs do not take this as an excuse to restrict communications instead.

2.6 Challenge/Response Systems

A challenge/response (CR) system is a different kind of greylisting, if you will. The mail server of the recipient of an e-mail intercepts the message and sends a challenge e-mail to the sender. The challenge might be to simply reply to the challenge e-mail or perhaps to solve a small puzzle first. In any case, some evidence of human intervention should be shown, under the assumption that spam software would not respond to such challenges. Only after the challenge is completed, the original e-mail will be delivered.

CR systems are a major annoyance of e-mail users, have a high risk of losing e-mail, and are quite useless against spam, for the following reasons [8]:

- Sender addresses of most spam are faked, so innocent parties are hit by challenge messages. (This is related to the bounce message problem explained in section 2.2.)

- For that reason, using a CR system will likely land *you* on blacklists for spamming.

- Spam using fake sender addresses can conceivably bypass CR systems by guessing sender addresses that have likely been authenticated already.

- Two parties using CR systems could never begin to talk to each other.

- To make sure that challenge messages get through, loopholes would need to be created in spam filter and other CR software. These loopholes could be exploited by spam [9].

Installing a CR system on the mail server of an ISP or large organization would also allow the provider the track the e-mail transactions of the users in fine detail, which is a clear privacy violation.

If CR systems became widespread and users became accustomed to responding to these challenges, spammers could easily send out fake challenges for e-mail address harvesting.

CR is therefore counterproductive in the fight against spam and should not be used under any circumstances.

[5] Related by Jean-Jacques Sahel, Head International Communications Policy, Departement of Trade and Industry, U.K. at [7].

2.7 Being Smarter Than Everyone Else

There are a number of other things that people have tried when filtering junk mail that are better not repeated. Some examples:

- Manually maintaining a DNSBL because you don't trust the external ones. It is impossible to keep such a system up to date by oneself.
- Running a local Pyzor server and feeding it only with your locally detected spam e-mail. This will not help you detect more or less spam.
- Securing your system in all kinds of ways but forgetting the MX backup. The concept of the classical MX backup hosted by someone else is pretty much obsolete.
- Randomly checking for RFC or other standard purity. This has nothing to do with spam.
- Changing your e-mail address regularly and sending everyone (including the spammers) an e-mail about that.

E-mail communication is governed by public protocols. Adding filters on top of that already alters the protocols in ways that are sometimes hard to comprehend, but if the filters are at least widely known, a general understanding can be developed about how e-mail should be delivered. Adding private filtering solutions that are not well thought out, have little correlation with junk e-mail occurrence, or annoy other users, do not benefit the reliability of the e-mail system.

3 Legal Issues

Besides the technical challenges, spam filtering also raises legal questions. Foremost, there are privacy issues. Of course, mail servers already keep extensive logs of all e-mail activity. But consider for instance the following additional points:

- Bayesian filters build a database of all words found in e-mails.
- A greylisting database keeps a record of all senders, recipients, and connecting IP addresses. This information is already in the mail server logs, but is the greylisting database adequately secured?
- Distributed systems like DCC inform the world about how many times an e-mail was sent, something which you perhaps did not intend the world to know.
- Challenge/response systems keep very detailed records about an e-mail transaction in order to verify the challenge [8].
- Techniques like SPF and blocking port 25 give ISPs increasing control about the paths that e-mails may take.

These kinds of checks tend to take place on the mail server of the ISP, so these databases are not under the control of the user.

Any kind of spam fighting effort begins with building a database of spam. All major providers of spam and virus filtering solutions have massive databases of e-mails. Various industry associations and government agencies are collecting e-mails as well, for example: The association of German Internet enterprises (*eco – Verband der deutschen Internetwirtschaft e. V.*) is collecting spam at `hotline@eco.de`. The *Zentrale zur Bekämpfung unlauteren Wettbewerbs e. V.* (a German association against unfair trading) is collecting spam at `beschwerdestelle@spam.vzbv.de`. The Federal Trade Commission (FTC) of the USA is collecting spam at `spam@uce.gov`. More databases are in preparation. As part of the "European Spambox Project", many of these types of databases will be shared at the European level.[6] Now the average user might not care who collects spam, but it is not hard to imagine other uses for these databases, such as tracking user accounts and users themselves.

Fortunately, consumers often have a recourse against these kinds of actions. Under German law, analyzing e-mails for signs of spam is not allowed without the consent of the recipient [1]. (The situation elsewhere in the EU should be similar since the relevant laws follow from EU regulations.) This follows both from privacy laws (*Bundesdatenschutzgesetz*) and the secrecy of telecommunications (*Fernmeldegeheimnis*). (Different rules apply to virus filtering.) Severe criminal penalties may apply in cases where e-mails are blocked, withheld, delayed, or analyzed by telecommunications providers without consent of the communicating parties. This would include all spam filtering methods including statistical analysis, distributed filtering, greylisting, and quarantines.

ISPs therefore usually include rules about e-mail filtering in their use policies or require users to explicitly activate the junk mail filter on their account. Users are still invited to verify what exactly happens to their e-mails before activating the "Please filter my spam" checkbox, and should request that information from the ISPs if necessary. As usual, of course, few people will actually bother about this, and the privacy of e-mail in general will deteriorate further.

4 Conclusion

"I can't find your e-mail. I think my spam filter ate it." This utterance is becoming commonplace, and worse, it seems to become an acceptable explanation for e-mail communication failures. Compare this to "I didn't get your postcard. I think the post office destroyed it because it contained too many exclamation marks." or "I didn't get your package. I think the dog killed the carrier because UPS in on our building's blacklist." No one would accept these as valid explanations for failure to deliver postal items.

The attempt to establish e-mail as a reliable and trustworthy communications medium is already lost. Spam and other forms of e-mail abuse certainly share most of the blame for that. But poorly thought-out countermeasures, the general failure of the computing industry to improve and amend the e-mail protocols, and the failure of governments to react to these developments in a timely manner have certainly contributed.

[6] Related by Jean-Christophe LeToquin, Attorney, Microsoft EMEA HQ, at [7].

I have shown that a number of e-mail filtering techniques have a negative impact on the stability, performance, and usability of the e-mail system. Users are invited to evaluate each filtering technique critically and thoroughly before putting it to use. I have also shown how the increased spam fighting efforts raise a number of privacy and other legal issues. Users are therefore also invited to critically examine the configuration and policies of their ISP's e-mail service.

It is unclear how the fight against junk e-mail will continue. New defense mechanisms are slow to arrive and will likely impose additional burdens on users. Increased efforts by governments are commendable but have yet to show large-scale results. More likely, the combat of spam will continue to be an uphill battle for all legitimate users of e-mail.

References

[1] Eisentraut, P., and Wirt, A., *Mit Open Source-Tools Spam & Viren bekämpfen*, Köln: O'Reilly, 2005.

[2] Trend Micro Incorporated, "MAPS – Stopping Spam at its Source", `http://www.mail-abuse.com/`.

[3] Martinec, M., `http://www.ijs.si/software/amavisd/`.

[4] Lundgren, B., `http://www.greylisting.org/`.

[5] "SPF: A Sender Policy Framework to Prevent Email Forgery", `http://www.openspf.org/`.

[6] Microsoft Corporation, "Sender ID Home Page", `http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx`.

[7] eco – Verband der deutschen Internetwirtschaft e.V., "3. Deutscher Anti Spam Kongress", Sept. 2005, `http://www.eco.de/servlet/PB/menu/1639239/index.html`.

[8] Self, K. M., "Challenge-Response Anti-Spam Systems Considered Harmful", Apr. 2004, `http://kmself.home.netcom.com/Rants/challenge-response.html`.

[9] Felten, E., "A Challenging Response to Challenge-Response", May 2003, `http://www.freedom-to-tinker.com/index.php?p=389`.