Re-thinking electronic mail

Lars Wirzenius

2020-04-12 09:19

Abstract

There are many problems with the existing Internet email system, such as spam, scam, surveillance, insecurity, centralisation, and complexity. The problems are starting to outweigh the benefits of the system. Fixing the problems by evolving the current system seems overwhelmingly difficult. This essay examines some solutions to the problems on the assumption that a completely new, parallel email system can be built.

This is not a proposal for a new system, but an exploration of the solution space, meant to provoke constructive discussion.

Contents

1 Introduction

I am tired of the existing Internet email system, both as a sender of email, as a recipient, and as an operator of an email server.

There's spam, scam, and the email system is getting centralised in all sorts of bad ways. This essay is about sketching what a good email system would look like, if it were re-designed from scratch, using everything we've learnt over the decades, and not necessarily using any part of the existing system.

As an anecdote, I am currently not on any active email discussion lists, or groups, or subscribed to newsletters. I have a separate email address that I give to online shops as contact information. My main address has been used for free and open source software contribution for many years.

I get less than two valid emails a day, usually from friends. Also, a small number of notification emails from my own automated systems. I get on the order of 400 spam and scam emails a day. They vary greatly in how targeted they are. They are all unsolicited and unwelcome. Unfortunately, despite honing my email filters for decades now, sometimes a valid email from a new sender ends up being filtered as spam, and I'm at risk of missing it.

Sometimes those emails are important, such as questions about some of my contributions. If I didn't skim my spam folder manually I would've missed the email about some of my software being used in Africa to provide local people with useful SMS services that would've been financially impossible with proprietary software.

There are good aspects the existing Internet email has that are still valuable enough that I continue to use it. I am, however, getting closer to the point that I'd like to make things radically better.

This essay collects my thoughts about email and what a replacement system might look like. I am not in a position to build the new system, but I can at least try to inspire more people to think about this and maybe the discussion will end up with something good.

1.1 Good aspects to keep

I find the following aspects of the current email system good and valuable and would like a new system to retain them.

- Ubiquitous. Approximately everyone on the Internet has email, or can get it.
- Anyone can email anyone. This lowers barriers for communication, globally. It's especially important for free and open source software projects, but also to allow people all over the world to easily self-organise to build a better world in general.
- Distributed: sender and recipient don't need to use the same server. Anyone can set up their own server, assuming time, know-how, and a little money.
- Standardised: there are many implementations and they're mostly interoperable.
- Supports off-line use. Not everyone can, or wants to, be online all the time.

1.2 Problems with the existing email system

- Spam, or unsolicited bulk messages. Worse, anti-spam measures drive centralisation, and are still ineffective, especially for those not using one of the huge centralised providers. End result: you either sacrifice privacy or you get tons of spam.
  - Spam is a result of the desirable feature that anyone can email anyone, combined with the fact that sending an email costs approximately nothing, even if you send millions of emails, and aggravated by the fact that spamming has de facto no real financial or legal repercussions.
- Scam, or trying to convince people to do something that they shouldn't or that's harmful to them or others. There is no widely used method to digitally sign email, and thus criminals can easily fake messages to look like they come from, say, Amazon, Netflix, Paypal, or other companies the recipient is likely to use. Everyone needs to be constantly on their toes to avoid mistakes.
  - There are standards for digitally signed email, but they're not great, and the big providers of email software or service tend not to support them, or support them badly.
- Becoming centralised to a few huge providers. This is bad for privacy and can be catastrophic for security. If one of the big providers gets breached, up to hundreds of millions of people's communications are at risk.
  - We're moving towards a future where hosting email yourself makes you suspicious. It's already the case that it can be difficult for a self-hosted email server to reach people on Gmail.
  - Worse, all the big email service providers have track records of closing people's accounts with no warning. Sometimes this happens by accident, sometimes due to nefarious policies.
- No real privacy, even if you self-host. Email is by default in clear text, and while there are standards for encryption, they are not widely used, and tend to also leave metadata (headers) unencrypted. It's difficult to hide who is sending email to whom.
- HTML email is not well standardised, and is a security and privacy risk. Different email clients implement HTML in different ways, and the web standards are not followed very closely.
  - Worse, there's an assumption that HTML emails can embed images from the Internet, which results in more security problems (images are complicated data provided by a potential attacker, and just viewing them is a security risk) and privacy issues (the image hoster will be notified when you view the email, see tracking pixels).

    There are moves to restrict this, but the problems have been known since HTML email was introduced, and the problems continue to exist. Some problems (such as embedding remote images) have gotten at least partial solutions, but the problems shouldn't exist at all.

- Attachments fill disks. Email is commonly used to share files, because it's easy and ubiquitous, even if it's not very good at it. There are services that make this better, but they are mostly proprietary, and require extra effort, are not ubiquitous, and people mostly don't use them routinely.
- There is no good support for group discussions. Massive dumps of forwarded discussions are commonplace in most large organisations. Mailing list managers exist, but they tend to be clunky, and tend to not be great for having discussions among large groups of people. They're better at sending out announcements and newsletters.
  - Email threads work, technically, but tend to result in surprisingly little communication happening, in the general case. People mix topics in threads, split the same topic into new threads, and generally don't use threads as intended. This is not the fault of people, but the technology.
- The technologies and standards for email are getting ridiculously complicated. Email was originally designed for relatively short messages in English only. To support non-English languages in a backwards compatible manner, email has gained whole extended families of ways to encode text and data: uuencode, base64, quoted-printable, and header encoding, to start with.

  The email tech stack is getting so hard to understand that it's difficult to even use, never mind implement correctly. Never mind that the complexity results in more effort to operate and keep the servers secure.

2 The spam and scam problem

There are a large number of problems. Rather than attacking all of them at once, let's consider them one at a time, and let's start with the most obvious problem: spam. As a side effect, the solution proposed below should also solve the scam problem.

2.1 Problem statement

The spam problem can be stated as follows:

> Anyone can send email to anyone else. There is practically no cost to sending many emails. It's difficult for the recipients to filter unwanted mail away automatically, because it would require the computer to understand human communication as well as humans.

The scam problem can be stated as follows:

Anyone can send email that looks like it comes from someone else, at least sufficiently well that an unobservant recipient is fooled. This can be used to con the recipient to click a link in the email that leads to a fake web shop, for example, or a site that attacks the recipient with malware.

2.2 Overview of solution

- Every email user has one or more identities, represented by cryptographic keys.
- All email is digitally signed using the cryptographic keys.
- No email is delivered unless it carries a digital stamp issued by the recipient, or someone authorised to issue one on behalf of the recipient.

The idea for stamps comes to me from (Wroclawski 2019), who seems to have gotten it from Christopher Lemmer Webber and Taler, and who knows where it originated from.

2.3 Digital identities

In this approach, each email user can have as many email identities as they want, and each identity is represented by a key pair for public key cryptography. The identities are not necessarily linked, just like personal and work email addresses are not linked.

The key pair, consisting of a public and a private key, is used to identify the email account and messages from the account. Every message sent using an identity is signed with the key for that identity.

This means misrepresenting the sender becomes much harder, reducing the possibility for scam.

Each identity (key pair) can have metadata associated with it, such as a name. There can be digital signatures for the metadata for certifying it, to avoid miscreants faking identities by creating new keys and associating someone else's name on them. With the metadata signatures, the recipient's email software can at least attempt to verify correctness of the metadata.

Alternatively, names are handled only on the recipient's side. If I get a message from you, and I'm sure it's from you, I can tell my email address book that the key you used to sign the message should have your name. If a miscreant creates a new key, my email software won't say it's from you, and the miscreant has to convince me that it's you. (This needs further thought.)

2.4 Digital signatures

For the purposes of this discussion, assume a way to digitally sign messages that covers the whole message, including its metadata. The details of how that is achieved do not matter: digital signatures have well-known, good solutions and since we are talking about a new system, we don't have to be compatible with the problems of the existing email system.

For this discussion, assume each message can be securely verified as having been sent by its sender identity. If a message claims to be from an identity, but its signature can't be verified, the message is rejected by the recipient's email software.

2.5 On encryption

To solve the problem of surveillance, email encryption is going to be needed. However, it doesn't seem to be necessary for solving the spam and scam problem, so it's not discussed, for now. A future version of this essay may address that.

2.6 Digital stamps

A digital stamp is a digital token issued by a recipient which gives a sender the capability to send one or more messages to the recipient.

A digital stamp is more powerful than a physical, paper stamp. Paper stamps can be transferred (sold, given) without limit. A digital stamp, however, allows more features:

- only the recipient can decide if it's still valid: the recipient can invalidate otherwise valid stamps
- digital stamps can have a complicated validity time: perhaps they're only valid for three months? or only on Mondays? or only during office hours?
- digital stamps may be indefinitely usable, or single-use: you might give someone new a stamp they can use only once, and if you don't give them another, longer-lived stamp, you won't get further email from them
  - for example, I might order a mug from an online shop and give them two single-use stamps: one for sending me the order confirmation and another for sending me a notification of shipment
- digital stamps may be valid only for a specific sender: I might issue a stamp to a shop and if they sell my contact information to a spammer, the spammer can't use the stamp to send me email; further, I will know the shop gave the stamp to the spammer

As an extra twist, digital stamps may also be an authorisation to someone else to issue stamps on your behalf. Rather than the stamp allowing them to send you an email, it lets them create a stamp that lets a third party send you an email. Your email software can put any and all the constraints it puts on stamps you issue directly on the delegation.

For example, if you and Alfred have a mutual friend, Bruce, you can give Bruce a stamp that authorises Bruce to issue single-use stamps to other identities. If Bruce thinks you and Alfred should know each other, Bruce can issue Alfred a stamp that lets Alfred send you a single email. If you like Alfred, you can issue further stamps to Alfred.

An employer runs their own email server, and that server determines which stamps it accepts. This lets an employer issue stamps on behalf of each of their employees.

Email servers could also, if so configured, issue stamps to senders with no previous connection to the recipient. This might be done by the sender having to produce some proof of work, which can be made arbitrarily costly in terms of computing resources. For example, the proof of work might require using five seconds of CPU time. This is costly enough that it makes large-scale spamming infeasible. (See (Back 2002) for an early suggestion.)

This makes the stamp system vulnerable to attackers who have enormous amounts of computing power, perhaps by using a botnet. It would be good to replace proof-of-work with something that's not vulnerable to a botnet.

Alternatively, the email server could require the person sending the email to solve a CAPTCHA-like puzzle, which can be made sufficiently varied to make it difficult to solve automatically. The actual puzzle does not need be standardized, only the mechanism by which the user is pointed at it, and how the result is communicated back to the mail server. There could, and should, be a very large number of different puzzles.

Email servers could also sell stamps for real money. Even at trivial costs, such as one US/EURO cent, this would be too costly for spammers.

I emphasise that the recipient decides what stamps are valid. Their mail server does not have to issue stamps to anyone who asks, if the recipient doesn't want email from strangers.

3 What next?

Do you think the solution proposed in this essay for spam and scam will help? If not, why not? Can you see a way for a miscreant to circumvent the proposed solution to get their unwanted message delivered to the recipient?

Let me know, preferably via the legacy email system, as a response to this fediverse thread, or using the GitLab issue system. If you want to propose improvements to the essay, feel free to file a merge request or send patches.

References

Back, Adam. 2002. "Hashcash - a Denial of Service Counter-Measure."

Wroclawski, Serge. 2019. “Preventing Spam on the Fediverse.” https://github.com/emacsen/rwot9-prague/blob/ap-unwanted-messages/topics-and-advance-readings/ap-unwanted-messages.md#postage.