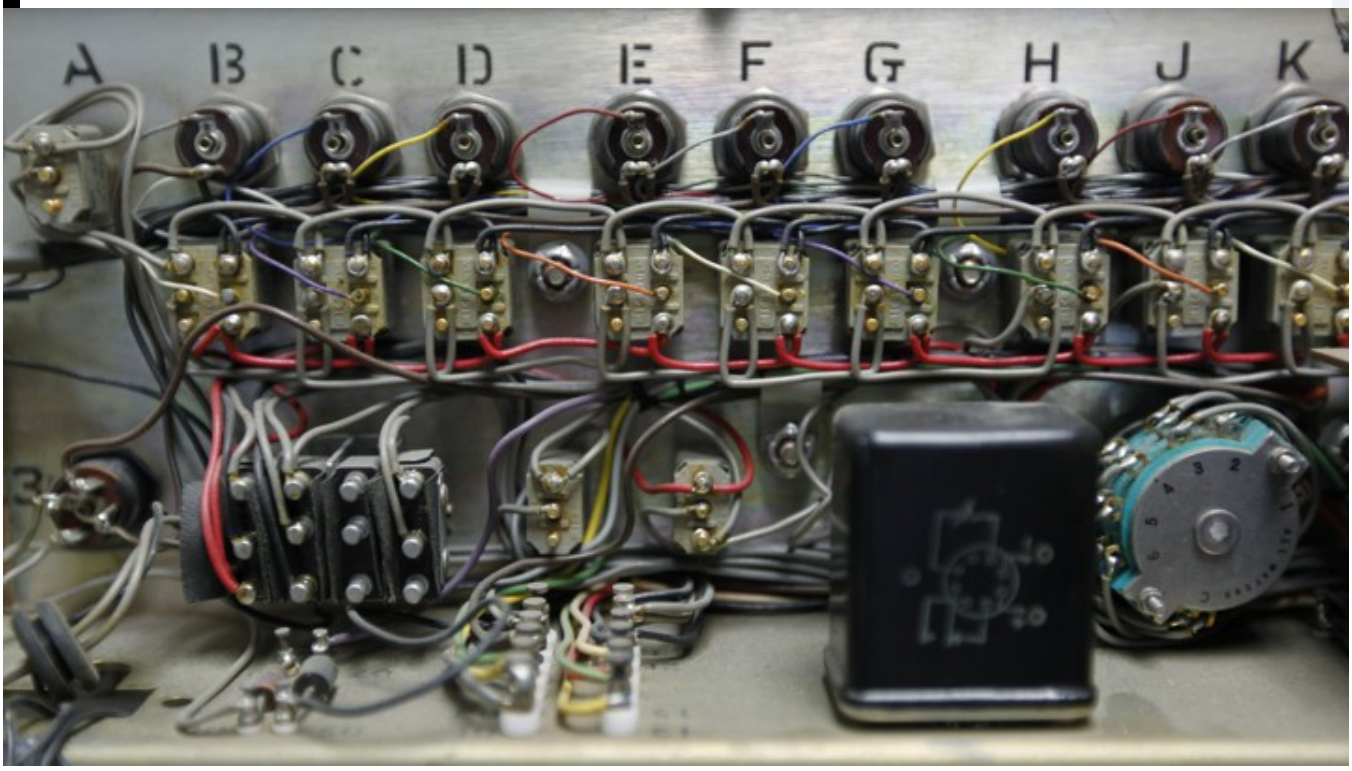
TECHNOLOGY

Email Is Dangerous

Electronic mail as we know it is drowning in spam, forged phishing mails, and other scams and hacks. It's going to get worse before it gets better.

QUINN NORTON MAY 21, 2018



A detailed view of UCLA's Interface Message Processor, which was used to interconnect participant networks to the ARPANET in 1969. (FRED PROUSER / REUTERS)

One week ago, a group of European security researchers warned that two obscure encryption schemes for email were deeply broken. Those schemes, called OpenPGP and S/MIME, are not the kinds of technologies you're using but don't know it. They are not part of the invisible and vital internet infrastructure we all rely on.

This isn't that kind of story.

The exploit, called Efail by the researchers who released it, showed that encrypted (and therefore private and secure) email is not only hard to do, but might be impossible in any practical way, because of what email is at its core. But contained in the story of why these standards failed is the story of why email itself is the main way we get hacked, robbed, and violated online. The story of email is also the story of how we lost so much of our privacy, and how we might regain it.

OpenPGP is hard to use, and S/MIME can only be maintained by IT departments. These tools were never popular, will never be popular, and don't even encrypt most of the metadata leaked by email. Most stories about these encryption schemes have focused on the special people who struggle through the process of using them or supporting them: human-rights defenders, journalists, security researchers, or the keepers of corporate secrets.

"We are bad at encrypted email for a lot of reasons," says Matthew Green, an assistant professor at the Johns Hopkins University Information Security Institute. "We don't know how to use it. We don't do key management right. But Efail is, surprisingly, not about all those problems. It's a bug that affects the people who actually put in the effort and do *everything* right."

What we think of as email got its start in the 1970s, with recognizable email addresses, mailboxes, folders, and sending and receiving as we know it now. The network was tiny then, mostly grad schools flirting with the American military-industrial complex. The trust model was around a small homogenous group of technical people, largely known to each other. Because of this, there was no authentication of emails, and there were no privacy measures. Forgery was not only easy, but common. Anyone could send mails saying they were from anyone, and the people running the servers could read everything that went by.

Email's privacy model was always based on courtesy: We wouldn't look at the messages crossing the network that weren't for us because that would be rude. It would be even more rude to change them, though system administrators did regularly insert strange messages or modify messages as pranks, or to get their users' attention. Emails from God or Santa Claus were not unheard of.

Email has changed since then, but not much. Most of what's changed in the last 45 years is email *clients*—the software we use to access email. They've clumsily bolted on new functionality onto the old email, without fixing any of the underlying protocols to support that functionality.

At the time email was being invented, so were new forms of cryptography. The important form for our purposes came from a paper published in 1976 called "New Directions in Cryptography," which introduced the ideas of public key cryptography to the world. This new method meant that for the first time people who didn't know each other could exchange information with mathematically

verifiable privacy, even on an open network where anyone could look at the data. Without having the cryptography keys of the people talking, all the system administrators would see was meaningless gobbledygook. But unlike casual email, crypto was always serious business, and from the start the U.S. government and many other governments wanted to control it.

By the 1990s, when email had opened up beyond universities and defense, and normal people could begin to get email addresses, cryptography was classified by the government as a munition: a material of war that couldn't be freely used or shared across national borders. To be clear, cryptography was then and is always a math technique, but this was a math technique that was being treated under the law like a bomb. In 1991, a man named Phil Zimmerman brought the math bomb and the anything-goes spirit of email together by creating Pretty Good Privacy, or PGP.

PGP allowed technologists to encrypt the contents of their emails, and to sign the emails in a way that allowed them to prove the mail's source was tied to a specific cryptographic key, held by a specific person. And with that, a power that was once the purview of nation-states became something determined hobbyists could do.

This kicked off what came to be known as the First Crypto War of the 1990s, a debate over who could use cryptography, and how, on the net. After an investigation of Zimmerman, with many lawyers and security researchers and corporations getting involved, the government backed down and stopped restricting encryption. New forms of cryptographic communication were born, allowing e-commerce and web services and online messaging to come into being. None of those things, and consequently very little of the net we know now, could have happened otherwise. This is because when everything is like email, you can't trust what arrives at your computer. Without that trust, you can't log in to anything, use your credit card, or even be sure that the information you have from trusted sources hasn't been altered on its way to you. The CDC website could tell you to smoke more, the front page of *The New York Times* could be wishing your co-worker a happy birthday, the White House website could be announcing that the intercontinental ballistic missiles were on the way. Cryptography allowed for the possibility of trust on an open network.

Around the same time that PGP was released, two computer scientists, Nathaniel Borenstein and Ned Freed, created a way to extend what email could do. They created a scheme called MIME (Multipurpose Internet Mail Extensions) to let

regular old email add new and strange features as people wanted them. By 1995, one of those features was S/MIME, which did the same essential things—encrypting and authenticating—as PGP did. MIME let you specify something you wanted email to do, and if people wrote support for it into the code of email clients, then it became something email could do. Other MIME extensions let you attach files to emails, and tell the computer what to do with them: open this one in Word, play that one as a movie. But none of these things changed email, they just piled more on top of it, extending and overloading the metaphors built into the basic technology people used for talking to each other online. Eventually when the web came along, and MIME standards and code let email turn into a full-fledged web browser, along with a calendaring app, a chat program, and the way you manage your identity online. All of this without ever doing much about the possibility that sneaky people were getting at your emails, or dealing with those mails from God or Santa Claus.

That brings us back to last week, and the release of Efail. The hack is simple and brilliant: It uses the fact that your email client thinks it's a web browser. An attacker sending mail can steal the content of secret messages you may have sent or received. It works like this: An email client running OpenPGP (the current standard of PGP) or S/MIME decrypts messages when it receives them, and since the clients are also web browsers, they fetch things from the web for displaying them to you in the email you open at the same time. So what if you happened to open an email, which decrypts whatever message it may have inside, even a hidden one, while the same email also tells your email client to fetch an image off the web whose name is now the entire contents of a message it just decrypted? It would just do it, invisibly, sending the now easily readable message anywhere on the net without you ever knowing it happened. Sure, an image named "Meet me at the park on Sunday at 3 a.m. and we'll make plans from there come alone.jpg" would never load on your screen, but you'll have invisibly asked for it, and that ask will now be recorded in whatever computer out there the person who sent the mail wanted it recorded on. And that mail could have just as easily said it was from your spouse or boss as God or Santa Claus.

Matt Blaze, an associate professor of computer and information science at the University of Pennsylvania, took to Twitter after the Efail announcement to say, "I've long thought HTML email is the work of the devil, and now we have proof I was right. But did you people listen? You never listen." (HTML is the language of web pages.) He went on to note: "An architecture that lets random people email

you code that executes whenever you read a message might interact with security features in unpleasantly surprising ways."

While this is bad news for the people who rely on this kind of cryptographic privacy, it's not surprising, and it's not just these security features. Experts like Green and Blaze have long been uncomfortable with everything that we pack into email. Email is so top-heavy and not fit for purpose that the system is drowning in spam, forged phishing mails, credential-stealing, and any number of other scams and hacks. It wasn't designed or engineered as a whole, it just grew and grew like kudzu over our networks and online lives. And still, people tend to believe that emails are what they say they are.

"Email is really dangerous," says Green, "People don't care about JavaScript in your browser or remote image access or even fancy encrypted email stuff because mostly the bad guys are phishing the CEO with straight HTML email from PayPal.EvilCompany.com and he goes for it."

Most hacks in the real world start this exact way: an email forged, telling you things that aren't true, while acting like a web browser. If they want to look like a bank, they can load their images from a bank's website. If the hacker wants you personally, they can send a mail from your coworker, spouse, or best friend, frantically asking you to check something on your calendar. When you try to log into that calendar, they have your login and password. Or in some cases, you don't have to do anything to be in trouble. Code execution, or the ability to run a program on someone else's computer, is often the most powerful attack, potentially letting someone else silently and invisibly take over your computer. Speaking of the Efail paper, Green says, "You could run JavaScript in ... five corporate email clients and it was a footnote in this paper." Such a thing, properly exploited by the sender of an email, could allow that sender access to the information stored on the receiver's computer.

Attachments on email, again an extension of what email is able to do created with MIME, are the other major source of computer insecurity. You click on something because you want to look at it, but that's the wrong metaphor. What you're telling your MIME-enabled email client to do is run this thing in the form that the creator of the email designated. If you don't know that it was designed to run code, and that code takes over your computer, you might not know that clicking on an attachment just gave control of your computer to someone else. You might never know.

The lesson of Efail is that you can build everything well, but if you've built on a bad foundation, there's no structure strong enough to stand. No one is responsible for email itself, and in the days since the Efail disclosure people have been pointing fingers at each other—email clients, vendors, OpenPGP standards, and S/MIME software vendors. It's no one's fault and it's everyone's fault. These kinds of disclosures, and the hacks built on the flaws of email, will keep coming for the foreseeable future.

"Email clients don't have to be this bad," says Green. Our interview was over the encrypted message application Signal. How does email get better? "If we're talking about real people, then we're using it … The path goes through Signal and WhatsApp and Wire and Wickr to someone doing a corporate product with email-like interface features." All of these tools use a much more modern encryption scheme, secure and authenticated from the start, and implemented with care. They aren't flawless, but they aren't crippled by design the way email was. They are meant for a network of strangers and are built to be suspicious of malignant forces.

This is also why they are making governments nervous again, kicking off a civil-liberties fight known as the Second Crypto War, bringing up questions about back doors in devices and messaging protocols, the FBI and cracking iPhones, and who gets to have privacy. But the First Crypto War, and the terrible state of email, taught us something: that vulnerability is never just about civilians facing off against governments. It's also about vulnerability to scammers, phishers, organized crime, abusive partners, and clever teenagers posing as God and Santa Claus.

We want to hear what you think about this article. Submit a letter to the editor or write to letters@theatlantic.com.