n the last issue, Stephen Farrell’s “Practical Security” column was entitled “Why Don't We Encrypt Our Email?” (Jan./Feb. 2009, pp. 82–85). I thought back and realized that although I used Pretty Good Privacy (PGP) in the 1990s, when all my mail was on a Unix platform, I hadn't been able to encrypt or digitally sign my personal email for a very long time. The extent of my email security is in using Secure Sockets Layer (SSL) to be sure my Gmail access is secure, which is a feature configured as a user option rather than the default despite a known security flaw that lets attackers hijack Web sessions (see, for instance, http://blog.secure-my-wireless.com/2008/02/gmail-still-not-100-safe-even-over-ssl.html, which indicates that even SSL can be defeated but is an improvement).

As a former PGP user, I decided to try out the Gnu Privacy Guard (GPG, available at www.gnupg.org) along with FireGPG (getfiregpg.org) to use with Gmail. It was an interesting experience.

The Bootstrapping Problem
First, I had to install GPG. Its Web page gave all sorts of warnings about how the installer's digital signature should be compared against an advertised digital signature. It also said not to use the newly installed GPG to do this check because a compromised program could lie and report the required signature instead of its real one. So, without having GPG already (and with limited tools available on a simple Windows computer), how could I check the package's validity? I poked around some Linux systems until I found one with some version of GPG and ran the verification tool to confirm that the signatures matched. But what should an inexperienced user do, or someone with no access to anything but his or her own desktop?

Installing GPG was fairly painless, although I never quite got the companion “gpgee” Windows extension to work. This feature would have let me right-click on a file and encrypt it using GPG, either with a public or a symmetric key.

Next came FireGPG. Like GPG, it announced a signature for the Firefox plug-in. This time, I decided that because the signature and plug-in were coming from the same place, I could trust the signature about as much as the plug-in. We must truly take a “leap of faith” to use such software — after all, even if the signature and download match, all it tells me is that there was no man-in-the-middle attack to send me bogus code. It doesn't tell me that the plug-in I just installed isn't channeling a copy of every message to some offshore server. Worse, if the software were untrustworthy, it would be getting access to my most private messages, the ones that are encrypted so that even Gmail can't see them.

Anyway, once I installed FireGPG and restarted my browser, Gmail was augmented to look for signatures or encrypted mail that I receive and to allow me to sign and encrypt my outgoing mail. (See Stephen’s column for more discussion about how you can use tools.) The catch is that I have no keys for anyone yet, and I don't really know anyone who uses GPG — all dressed up for the party, but no place to go.

Public-key servers do let you search for people and see if they have a key. (You could look for me, and you'd find the PGP key I used a decade ago, for an address that no longer exists. If I were lucky, I could figure out whatever happened to my private key that goes with it. Then again, I understand users’ reluctance to supply keys at all, given that these repositories are excellent one-stop shopping sources for spammers to collect email addresses — that’s what is deterring me from uploading a key with my new address.) I tried a few people, didn’t locate keys for...
any of them, and gave up. Of course, even if I found someone's key, the next question is whether I could trust it. The “web of trust” model says that if the key is “signed” by someone I trust, I can trust the key. The same holds true for a path of trusted keys used to reach the intended party. But how does a new user establish trust with anyone?

Of course, an alternative to the “web of trust” model exists, namely, certificate authorities (CAs). You can use X.509 certificates (http://en.wikipedia.org/wiki/X.509) and CAs such as CACERT to get something that a real authority — rather than a handful of individuals — vouches for. You can even use notaries to affirm that people are who they say they are. As an extreme case, Barry Leiba tells me that at least one country now ties public-key certificates to drivers licenses! But CAs have issues as well, such as key expiration and revocation and the possibility of a compromised CA.

I’ll switch my focus from individuals to organizations, such as banks and hospitals. Stephen noted that for areas in which privacy is especially important, organizations could encrypt their mail, but they tend to take other approaches — for example, he mentioned that medical service providers use proprietary Web-based communication systems. This must be Anyplace But the USA, because I’ve never gotten a piece of email from any of my doctors, proprietary or otherwise. (OK, I admit that my dentist does sometimes communicate about scheduling appointments via open email.) I see a similar experience in another domain, though: many financial providers send email saying “come to our site” when they want to tell me something, without actually putting anything private in the email itself.

Someday, some large financial institution will bite the bullet and start signing its mail in addition to the other antiphishing tricks it currently uses (such as including the last four digits of my credit card in the mail). When it can both sign its mail and encrypt messages it sends to me, secure email will truly show its worth.

As an aside, Jim Miller’s column this month looks at CAs and user identities from another perspective: coping with collisions in the user identity namespace, especially over long periods of time. It adds a very interesting wrinkle to the discussion.

Issues of Trust

Let me return to some institutions’ practice of putting in safeguards that try to document their identity. When I go to a bank’s site and log in, and it presents me (via a secure connection) with an image and phrase that I previously gave it, I feel pretty good. I can be reasonably confident that I’m talking to the institution I intend. However, a man-in-the-middle attack is still possible — I could get a secure connection to someone else who uses my credentials to authenticate to the bank and present me with my image and phrase. This is why it’s good that some institutions present an extra screen when they see a user coming from a new address, asking for more information; however, a user’s reaction will probably be to provide the extra information rather than question why the security alert was triggered in the first place!

What about the bank that sends me email and “for security” includes part of my credit-card information? This seems to me to be more of a mixed bag.

In the last issue, I discussed a (literally, now) poor woman who lost a small fortune to online scammers. She erroneously trusted them because instead of telling her that she’d been picked randomly to help access the huge fortune of someone she’d never heard of, she was contacted because she’s the relative of someone who spent time in Africa. This targeted scam was way more convincing because it contained personal information.

These days, many forms such as credit-card statements display only a subset of the information you’d need to use the account. For instance, they print xxxx-xxxx-xxxx-1234 for an account ending in 1234. Receipts, which huge numbers of people in the sales industry can access, contain the same information, including the customer name. Imagine if someone could get the account number corresponding to an email address and then forge an email with the correct “security code.” Instead of being skeptical about a phishing attack, the recipient might be more inclined to believe the email to be legitimate and follow any provided link. Such targeted phishing has been referred to as “spear phishing” (see, for example, www.microsoft.com/protect/yourself/phishing/spear.mspx), though I believe existing targeted phishing focuses more on things like “Which bank does Fred use?” than “What is Fred’s account number?”

The same threat arises from social networking. I get huge amounts of spam that claims to be from people I’ve never heard of, and I get huge amounts of spam that claims to be from me. Both types are easily filtered, but what if a spammer were to
mine social networks and other public data to identify people a recipient knows? Not only did researchers demonstrate this approach some time ago,¹ documented cases now exist to solicit money from friends whom people whose Facebook accounts have been hacked and used to.

Although people should quickly learn to be skeptical when someone posts on Facebook claiming to have been robbed and needing money wired, they might be less suspicious when email from a friend asks for help. Some defenses, such as identifying which hosts can send mail on a given domain’s behalf, might provide some security here, as would signing mail from a domain (http://tools.ietf.org/html/draft-kucherawy-sender-auth-header). Using different email addresses in different contexts also provides help; if I give Facebook a different email than I give Citibank, I’m not going to believe the mail is from Citibank when it looks like it relates to Facebook. But in the end, we have to move to a more secure system with stronger safeguards in every message.

We have been warned.
(signed)

The One and Only Fred Douglis …

honest!

Acknowledgments

I thank Stephen Farrell and Barry Leiba for helpful comments on this column. The opinions expressed in this column are my personal opinions. I speak neither for my employer nor for IEEE Internet Computing in this regard, and any errors or omissions are my own.

Reference


---

Editor in Chief

Fred Douglis • f.douglis@computer.org

Associate Editors in Chief

Siobhán Clarke • siobhan.clarke@cs.tcd.ie
Doug Lea • dl@cs.oswego.edu
Michael Rabinovich • misha@eecs.cwru.edu

Editorial Board

Virgílio Almeida • virgilio@dcc.ufmg.br
Helen Ashman • helen.ashman@unisa.edu.au
Elisa Bertino • bertino@cerias.purdue.edu
M. Brian Blake • blakeb@cs.georgetown.edu
Vinton G. Cerf • vint@google.com
Stephen Farrell • stephen Farrell@cs.tcd.ie
Robert E. Filman* • filman@computer.org
Carole Goble • cag@cs.man.ac.uk
Michael N. Huhs • huhs@sc.edu
Barry Leiba • barryleiba@computer.org
Samuel Madden • madden@csail.mit.edu
Mark Manasse • manasse@microsoft.com
Cecilia Mascolo • c.mascolo@cs.ucl.ac.uk
Pankaj Mehra • pankaj.mehra@hp.com
Chris Metz • cmetz@cisco.com
Dejan Milojčić • dejan@hpl.hp.com
Jason Nieh • nieh@cs.columbia.edu
Charles J. Petrie* • petrie@stanford.edu
Krithi Ramamritham • krithi@cse.illinois.edu
Henning Schulzrinne • hgs@cs.columbia.edu
 Amit Sheh • amit.sheth@wright.edu
 Munindar P. Singh* • singh@ncsu.edu
 Oliver Spatscheck • oliver@spatscheck.com
 Torsten Suel • suel@poly.edu
 Craig W. Thompson • cwt@uark.edu
 Shengru Fu • shengru@cs.uno.edu
 Maarten van Steen • steen@cs.vu.nl
 Steve Vinoski • vinoski@ieee.org

* EIC emeritus

IEEE Communications Society Liaison

G.S. Kuo • gskuo@mail.com

CS Magazine Operations Committee

David A. Grier (chair), David Albonesi, Isabel Beichi, Arnold (Jay) Bragg, Carl Chang, Kwang-Ting (Tim) Cheng, Fred Douglis, Hakan Erdogmus, Carl E. Landwehr, Dejan Milojčić, Sethuraman (Panch) Panchanathan, Crystal R. Shif, Maureen Stone, Fei-Yue Wang, Roy Want, Jeff Yost

CS Publications Board

Sorel Reisman (chair), Alain April, Angela Burgess, Alan Clements, Frank E. Ferrante, David A. Grier, Audrey Kremer, Philip A. Laplante, Paolo Montuschi, Jon Rokne, R. Sampath, Steve Seidman, Linda I. Shafer, Roy Sterritt, Steven L. Tanimoto

Writers

Access www.computer.org/internet/author.htm. Articles are peer reviewed for technical merit and copy edited for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the authors’ or firms’ opinions; inclusion in this publication does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

Letters to the Editors: Email lead editor Rebecca Deuel-Gallegos at rdeuel-gallegos@computer.org.


Subscribe: Visit www.computer.org/subscriptions.

Subscription Change of Address: Send requests to address.change@computer.org.

Missing or Damaged Copies: Contact help@computer.org.

To Order Article Reprints: Email internet@computer.org or fax +1 714 821 4010.