Interview with Steve Bellovin

RIK FARROW

Steven M. Bellovin is a professor of computer science at Columbia University, where he does research on networks, security, and, especially, why the two don't get along, as well as related public policy issues. In his spare professional time, he does some work on the history of cryptography. He joined the faculty in 2005 after many years at Bell Labs and AT&T Labs Research, where he was an AT&T Fellow. He received a BA degree from Columbia University, and an MS and PhD in computer science from the University of North Carolina at Chapel Hill. As a graduate student, he helped create Netnews; for this, he and the other perpetrators were given the 1995 USENIX Lifetime Achievement Award (The Flame). Bellovin has served as chief technologist of the Federal Trade Commission. He is a member of the National Academy of Engineering and is serving on the Computer Science and Telecommunications Board of the National Academies, the Department of Homeland Security's Science and Technology Advisory Committee, and the Technical Guidelines Development Committee of the Election Assistance Commission. He has also received the 2007 NIST/NSA National Computer Systems Security Award.
bellovin@acm.org

Rik is the editor of *;login:*.
rik@usenix.org

I first met Steve at an early USENIX Security symposium, and over the years, that's where we would often meet. I mostly knew Steve through the research he had published, as well as the book he co-authored with Bill Cheswick [1]. But I really didn't know much more than that Steve and Bill had met while they were both working at Bell Labs in Murray Hill, New Jersey. When we met at conferences, we'd mostly talk about what was happening at the time, not the past.

I decided that it was time to ask Steve a few questions about his past, as well as get a better understanding of the path that has lead him from being a student to being a professor, with many interesting adventures along the way.

Rik: What I am curious about is how you became involved with Internet security. I know you worked on EKE and helped write Netnews while a graduate student, but that doesn't really tell me how you got to writing a popular and important book on firewalls plus multiple security-related RFCs over the next 10 years.

Steve: My background, ultimately, is as a sysadmin. I learned programming in my sophomore year of high school, when that was very rare. The students who knew how to program—about half a dozen of us—ran the machine, an IBM 1130, at Stuyvesant High School in NYC, without interference from the teachers. We knew more about it than they did, and they didn't resent us for it. I was curious how a "kernel" (to use today's terminology) worked, so I wrote a disassembler to study the OS.

Rik: How is it that your high school had a computer, back when computers were truly rare?

Steve: Stuyvesant is an examination-entrance NYC public high school that's for students interested in math and science. Someone there talked someone into believing Stuyvesant needed a computer...after all, the "competition," Bronx High School of Science, already had one.

When I got to college, my part-time jobs were all systems programming. My last two years, I worked at the City College of New York Computer Center at the time when it was the central site for all of the City University of New York; we had a smallish IBM mainframe that was used for academic and administrative computing. My college years are also when I learned networking—IBM Bisync in those days. At CCNY, I caught my first two hackers; they were poking around at the administrative side of things. After studying their card decks(!), I hired one and referred the other to the dean.

Rik: I remember card decks all too well. What did you focus on at grad school?

Steve: I was mostly interested in programming languages. For breadth, I did a theory dissertation on proving the output of compilers correct; although I wasn't doing security then, the dissertation actually turned out to be security-relevant. I learned UNIX and kept up my systems programming and kernel-hacking skills (writing device drivers, studying how things worked, etc.).

Netnews was a separate story, but the original impetus was the need for a convenient mechanism for administrative announcements. It's also when I first got involved with USENIX—I

was at the meeting where, for trademark reasons, they had to change the name from the "Unix User's Group." I also started studying TCP/IP around 1980 or 1981, when I got a copy of the "Internet Protocol Transition Handbook."

Rik: And when you started working at Bell Labs, you also worked on networking?

Steve: When I got to Bell Labs, I became responsible for 1.5 of the first 3 lengths of thick Ethernet cable in the entire company: my lab, another lab, and the "backbone" cable connecting us. That got me more heavily involved with TCP/IP, which I helped bring to the rest of Bell Labs.

There were occasional network outages due to misconfigurations. In those days, there were no dedicated routers; you simply stuck another Ethernet board into your VAX and used it as a router. Multihomed 4.2bsd hosts would forward packets by default—a misfeature, but few people realized that at the time. Many of the outages were routing-related, and I realized that what could happen by accident could happen intentionally. One cannot be a good sysadmin without worrying about security. I started worrying about routing security and address-based authentication. Adding to that, Robert T. Morris interned at the Labs and invented sequence number attacks, so I had more to worry about.

In addition, there were ongoing attacks from the outside against Bell Labs [1]. I was one of several people who detected the attacks—I'd added a cron job that scanned the UUCP log files for attempts to snarf /etc/passwd. I had nothing to do with the subsequent investigation.

Rik: Perhaps you could expand on the problems with multihomed 4.2bsd, since these also occurred with SunOS. I believe this is one of the reasons the Morris Worm was so successful: People used Suns and VAXen as routers, and they shared directory structure and commands like Sendmail.

Steve: No, I don't think it was a factor in the worm's spread; that was more a case of monocultures and no filtering. The issue was more subtle: It was a confusion of the difference between a multihomed host and a router, which meant that topologies were richer than intended. The folks at Berkeley who wrote 4.2bsd were very good UNIX and kernel hackers, but arguably didn't have a good grasp of some of the more subtle points of TCP/IP. It took RFCs 1122 and 1123 to sort that out.

Rik: I recall just how unpleasantly mysterious TCP/IP was. I had been a UUCP expert, but when it came to assigning IP addresses on a private network, no guidance existed in the late '80s. I, like many others, wound up using an address that appeared in Sun documentation, so we could use the thin Ethernet cables and connectors that came with Sun 3 workstations.

Steve: Right, that and a related issue were some of the things that caused trouble. Sun implemented something that was what today we'd call "zeroconf"—if you didn't set an IP address, the software would pick one via an algorithm and protocol known only to other Suns. When this happened on the backbone, it meant that someone would suddenly grab .1 on that net, and .1 was really the router to other locations in the Labs. Then I asked myself, "What would happen if someone did that maliciously?" and my career took a sharp turn.

Worrying about the TCP/IP issues led to my first major paper, on TCP/IP protocol-level security [2]. The authentication and routing issues led me to think more about crypto; in addition, sometime in the 1980s my wife gave me a copy of the hardcover of Kahn's *The Codebreakers*. Looking at Kerberos and thinking about password guessing led me to worry about guessing attacks on the initial sequence used to get a ticket-granting ticket. Mike Merritt and I talked about it, and I worried about it for several months. Finally, my mind wandered while I was sitting in a really boring talk and I had an inspiration; the result was EKE. Today, I use this story today to motivate students to come to class, no matter how boring it is; they might invent something while I'm droning on.

By this point, I was doing network security full time. Ches and I dealt with Berferd in 1991. A chance meeting with him on a train ride to Baltimore for the USENIX Security conference led to us agreeing to write a book. John Wait, the eventual editor of the book, happened to pay his routine annual visit to me shortly afterwards. He always wanted to know whether I was interested in writing a book; I always declined, because I didn't have anything to say. This time, I did have something to say.

Firewalls were a pretty obvious path for network security then, given both the Presotto/Cheswick design of the AT&T gateway to the Internet and my statement about topological defenses in the TCP/IP protocol insecurity paper. It was easy around then to be one of the top network security people because there were so few, but that meant I was noticed. I was invited to be on the IPng directorate; that got me involved with the IETF, so I wrote RFCs, etc.

Rik: With your early interest in IPv4 routing, and your participation in the IETF, did that lead to any advances in improving the security of routing protocols? Or have any influence on IPng?

Steve: I was one of the people responsible for IPv6 requiring IPSec in all implementations; this is precisely because of all the risks from address-based authentication. Routing security is still an open issue, although I was one of the people who did the work leading to the IETF's BGPSEC working group.

Rik: You once mentioned to me that one reason for the lack of routing security is the convenience of the current state of affairs

for nation states who might wish to route traffic past points where they can intercept it. I am of course paraphrasing, as you said this quickly in passing. But BGP is still just tables of information exchanged between routers with no signatures to verify routing assertions.

Steve: I don't think I said that that's one reason, but it's certainly something that many countries like and exploit now. There have been incidents, such as when Pakistan decided to block YouTube internally and affected global routing [3].

Rik: I would appreciate it if you would say more about the lack of robust security for the routing infrastructure. I've often assumed this is what the L0pht members were referring to when they claimed they could "bring down the Internet in 30 minutes," given just how much trouble we've experienced from accidents (routing of a lot of the Internet to a single small ISP in Florida as an example [4]).

Steve: Routing and the DNS [5]. A 1999 National Academies study committee that I was a member of called routing and DNS the two main trouble spots on the Internet.

There have been two main reasons why BGPSEC hasn't happened yet. First, it's expensive: Lots of routers will have to be replaced by ones with a lot more RAM and a lot more CPU power to do signing and signature verification. Second—and this is the interesting one—it creates new failure modes, and some of these failure modes have political components.

For BGPSEC to work, you MUST have a PKI for IP addresses. A failure at any node in the path from you to the root means that you won't have a good certificate, which in turn means that you'll be off the air. Worse yet, this PKI is inherently a tree structure, i.e., every node is a monopoly, and monopolies don't have particularly much market pressure to make them behave efficiently or to provide good customer service. Also, any node is susceptible to pressure or compulsion by its government: "Revoke this address space certificate under penalty of law." Today, ISPs work by trusting each other on such issues; BGPSEC will require correct technical functioning at all levels of the PKI.

Rik: In your 2003 statement before the DHS subcommittee [6], you wrote that today's operating systems are far more reliable than those used a generation ago. They are also far more complex. What do you think about research toward building partitioned kernels, such as the seL4 microkernel, or the work being done by Robert Watson and others to build a system with efficient hardware segment registers for enforced separation both within applications and at the OS layer?

Steve: Well, strong walls are something we're pretty good at. The problem is that the components have to talk to each other, which implies gates, and these gates have policies attached. That's what we're lousy at: specifying and implementing the gates and their policies. More walls can lead to higher assurance, which is good, but it's not really the solution. My overarching research goal is to understand how to divide a system into walls-separated components in the proper way [7].

Rik: While reading an article you co-authored [8], I learned that call detail records (CDRs), which are described as call-metadata, also cover information that's included in email headers and can be collected without requiring a search warrant. It seems that CDRs for email provide a lot more information than just the caller and the callee's number, date, and length of call.

Steve: CDRs for mobile devices give approximate location to a pretty fine granularity. CDRs for wireline devices give the phone number, which for ordinary PSTN is pretty closely tied to an address. All CDRs have call length; most have caller and callee. Email headers have more or less that; in particular, the first "Received:" line generally gives the sender's IP address, which is a decent clue to location for non-mobile devices. There are two exceptions: if you use cryptographic tunnels (including, but not limited to, VPNs) or if you use Gmail. Gmail strips all that off—Google knows, but the recipient doesn't.

Rik: As members of research or IT communities, what should we be doing to encourage greater privacy?

Steve: First and foremost, don't collect data you don't need. If you do need it for immediate operational purposes (e.g., mail logs), discard it when you don't need it, or perhaps hash some of the fields.

Second, consider what privacy-preserving options might exist in the systems we design. Take the "Message-ID:" header as defined in RFC 5322:

> The message identifier (msg-id) itself MUST be a globally unique identifier for a message. The generator of the message identifier MUST guarantee that the msg-id is unique. There are several algorithms that can be used to accomplish this. Since the msg-id has a similar syntax to addr-spec (identical except that quoted strings, comments, and folding white space are not allowed), a good method is to put the domain name (or a domain literal IP address) of the host on which the message identifier was created on the right-hand side of the "@" (since domain names and IP addresses are normally unique), and put a combination of the current absolute date and time along with some other currently unique (perhaps sequential) identifier available on the system (for example, a process id number) on the left-hand side. Though other algorithms will work, it is RECOMMENDED that the right-hand side contain some domain identifier (either of the host itself or otherwise) such that the generator of the message identifier can guarantee the uniqueness of the left-hand side within the scope of that domain.

How about making the part to the right of the @ the SHA-256 hash of the domain name? It's just as unique and doesn't leak information. Yes, you can brute force it—*if* you know the name of all original sending hosts. Is that example too contrived? The Canadian Privacy Commissioner's report on the TJX hack slapped them down for storing driver's license numbers instead of a hash thereof.

As a researcher, the problem is easier to state: Where is privacy lost, and what technically can be done?

Rik: In your 2010 undergraduate "Computers and Society" course, you point out that voluntary surrender of data can lead to secondary use of that data. Many people are happy to share information about their everyday lives, as well as their likes (and dislikes), with social media, but even using a public email account like Gmail results in the sharing of personal information.

Our choices there do not appear to be good: Participate or don't participate. Encrypting email doesn't work without a simple way of securely sharing keys. What do you suggest?

Steve: There are three things. One, of course, is education. Second is research on things like key distribution and easier-to-use encryption. I think this can be done decently well; in fact, I have some projects going on now on that topic. Third is law or regulation, concepts that I, at least, am not allergic to. I do think that we're better off with use restrictions rather than collection restrictions, but in the privacy community I think I'm in the minority on that.

Why do I prefer use restrictions? Some data has to be collected for operational reasons—I've been Postmaster; I know how important mail logs are—and some data, such as health records, can be used for exceedingly important purposes that don't violate anyone's privacy. The risk is that today's rules will be ignored or will be changed for a compelling-enough—or currently compelling-enough—reason (e.g., the misuse of the 1940 census records to aid in interning the Japanese).

Rik: In your presentation about your year as the chief technologist at the FTC [9], you explain that the FTC is reactive in how they can act. For example, if a company promises to keep data secure, but has inadequate technical controls, the FTC has seen this as "deceptive and/or unfair" practices, and can take the perpetrator to court. Most companies sign consent orders, and companies that fail to improve can then be fined. But one company, Wyndham, which had lost data three times in two years, has decided to fight back. What's happening with Wyndham? You said that Wyndham wants a ruling that would limit the FTC's ability to regulate in this area.

Steve: So that's a very interesting question. Wyndham's basic position is that since the FTC has never issued any rules, they don't know what standard they should meet, so the enforcement is unfair. About a month or so ago, the judge finally handed down her ruling, completely rejecting Wyndham's arguments. Naturally, they're going to appeal to the Second Circuit.

Another company, LabMD, filed similar objections. They didn't survive the process; they went bankrupt. A week or two ago, an administrative law judge—part of the FTC, but organizationally independent—was very, very critical of the FTC, but just this week ruled against LabMD despite that. I haven't had a chance to look at the opinion, so I don't know the grounds; it might have been hyper-technical rather than substantive.

Rik: How did you get involved in the legal realm?

Steve: The short answer is that I've always been interested in law and policy. I did (minor) campaign work as a teenager; in college, I took a constitutional law class because it was interesting. I was one of very few people in that class who wasn't intending to go to law school.

About 20 years ago, I started working with legal issues professionally. These were the days of the Clipper Chip, the Crypto Wars, and the bills that would become the DMCA and CALEA. Matt Blaze and I were able to work with the AT&T policy people and persuade them that the things we wanted for privacy reasons were in the company's interest—and, of course, any company wants to avoid government regulation, so that wasn't that hard.

In the Berferd incident, the lawyers made us kick him off the machine, so I started wondering about liability. When we did the *Firewalls* book, I threw in a chapter on the legal aspects. Unlike most of the book, which was joint work, that chapter was all mine. I got an attorney (who later went on to become second in charge of DoJ's computer crime section) to teach me the basics; I was also assisted by one of the AT&T patent attorneys. Things grew from there.

Around 1995 or so, a Fordham law professor spent his sabbatical in my department at Murray Hill. I still work with him on tech and law. Basically, the need was there and I was interested. There was never any danger, way back when, of me going to law school; I liked computers too much. But I was always interested, and over the years, I've done more and more of it professionally.

Rik: After working for many years at the Labs, you moved to Columbia. Can you tell us why you decided to become a professor?

Steve: I left AT&T Labs Research for a number of reasons. One was simply that it was time to do something different. I'd been there for more than 20 years, I had a great time, and I had management that supported me. But I'd always wanted to teach, and I decided that it was time. I really enjoy teaching, and a lot of

what I've done—talks, writing papers, the *Firewalls* book—is just another form of teaching. The other factor, of course, was that I was not sanguine about the future of research there, and when a good opportunity arose I decided to take it. I'd received other offers from other universities in the past, but it wasn't time to leave.

Sadly, I was right about AT&T Labs. It's not the place it was; from what I hear there's not nearly as much freedom to do research and to publish, and many of the very best people have left or been laid off.

If I wanted to teach, why didn't I do that straight out of grad school? I did—and do—dislike everything to do with getting grants. In fact, it's been worse than I had expected. On the other hand, some of the benefits—the ability to work and speak freely on public policy issues, the freedom to do things like write law review and history of cryptography articles, and, above all, access to a wonderful research library—have been greater than I had anticipated. AT&T was a wonderful place, but I don't regret moving on—it was time.

Rik: Finally, why did you start writing about technical history?

Steve: During a 1993 conference, Matt Blaze and I heard an ex-NSA cryptologist say that the needs of Permissive Action Links (PALs)—the cryptographic combination locks on nuclear weapons—led the NSA to invent public key cryptography in the 1960s. Now, PALs are supposedly impossible to bypass, and a security mechanism that can't be broken is of course of great interest to security people. Matt and I wondered about both parts of this: How do PALs work, and is that historical statement accurate? I did a lot of digging, including a Freedom of Information Act request, and eventually generated a lengthy Web page and a talk, which I gave at USENIX Security in 2004. I also honed my historian skills doing a non-computer research project.

Coincidence then took a hand. I have an odd hobby: I collect old telegraph codebooks. (I gave a talk on them at USENIX Security in 2009.) A few years ago, I had a free day in Washington—what should I do? In the morning, I went to the Supreme Court to hear oral arguments in a case—for all of my interest in legal matters, I'd never done that before. In the afternoon, I decided to go to the Library of Congress to look at some of their codebooks. They have hundreds, though; which should I examine? I spotted one from 1882 whose title spoke of "privacy" and "secrecy"—it sounded better than most for a security guy, even though I had low hopes. However, when I read its preface, I realized that it described the one-time pad 35 years before the textbooks say it was invented. I dropped a note to David Kahn asking if he knew anything about it. He didn't and suggested that I write a paper—which I needed no prompting to do; I'm an academic and academics write papers. Before I went to sleep that night, I'd tentatively identified the author. By the time I was done, I had a 20-page paper with 78 references, ranging from the society pages of the *San Francisco Chronicle* from 1907 to a biography of the founder of theosophism to an 1829 history of Freemasonry.

Well, that paper led to another (which has led to two accidental spin-offs), and I have several more planned. None of these will change the way we do things, but it's always good to learn where we've come from. Fun fact: that 1882 codebook shows the use of someone's mother's maiden name to authenticate certain financial transactions. That's one mistake we haven't corrected yet!

Resources

[1] *Firewalls and Internet Security: Repelling the Wily Hacker* (Addison-Wesley Professional, 2003): https://encrypted.google.com/books/about/Firewalls_and_Internet_Security.html?id=_ZqIh0IbcrgC.

[2] S. Bellovin, "A Look Back at 'Security Problems in the TCP/IP Protocol Suite'": https://www.cs.columbia.edu/~smb/papers/ipext.pdf.

[3] Ryan Singel, "Pakistan's Accidental YouTube Re-Routing Exposes Trust Flaw in Net," *Wired*, Feb. 25, 2008: http://www.wired.com/2008/02/pakistans-accid/.

[4] AS 7007 Incident: http://en.wikipedia.org/wiki/AS_7007_incident.

[5] S. Bellovin, "Using the Domain Name System for System Break-ins": https://www.cs.columbia.edu/~smb/papers/dnshack.pdf.

[6] Statement before the House Select Committee on Homeland Security Subcommittee on Cybersecurity, Science and Research & Development: https://www.cs.columbia.edu/~smb/papers/Statement.pdf.

[7] K. Dent, S. Bellovin, "Newspeak: A Secure Approach for Designing Web Applications": https://mice.cs.columbia.edu/getTechreport.php?techreportID=506.

[8] S. Bellovin, M. Blaze, W. Diffie, S. Landau, P. Neumann, and J. Rexford, "Risking Communications Security: Potential Hazards of the Protect America Act": https://www.cs.columbia.edu/~smb/papers/j1lanFIN.pdf.

[9] S. Bellovin, "Life Amidst the Lawyers: A Technologist's Year at the FTC": https://www.cs.columbia.edu/~smb/talks/Life_FTC.pdf.