Humans in the Loop

Human–Computer Interaction and Security

S.W. Smith
Dartmouth College

The security field suffers from an endemic problem: despite our best efforts, the current infrastructure is continually full of security vulnerabilities. The systems that comprise this infrastructure also are full of boundaries and interfaces where humans and systems must interact: most secure systems exist to serve human users and carry out human-oriented processes, and are designed and built by humans.

From the perspective of the *human-computer interaction* (HCI) community, many of these interfaces do not reflect good thinking on how to make them easy to use in a manner that results in security. From the perspective of the security community, many widespread security problems arguably might stem from bad interaction between humans and systems. I recently attended a workshop (ACM/CHI 2003 Workshop on Human-Computer Interaction and Security Systems; www.andrewpatrick.ca/CHI2003/HCISEC) that tried to bring together these communities to trigger further inquiry into this area. In this article, I want to discuss the workshop and how the thinking there applies to the secure systems topic this department addresses.

Leaving out the most important piece

If I stood up on a soapbox in a roomful of security colleagues and said, "It's been 30 years and we're still fielding insecure systems, so we must be doing something fundamentally wrong," nothing would be thrown at me—although debate might rage about exactly what we're doing wrong and how to fix it. But I'm going to go a bit further (because if you're standing on a soapbox and nothing gets thrown at you, then you're not trying hard enough). Support is growing for the thesis that a root cause of the pervasive (in)security problem is the interaction between humans and computers—we're trying to secure a system that embodies human processes and includes human users, but we restrict our analysis and designs to the computers themselves.

Perhaps this thesis received its fullest expression at the ACM/CHI 2003 workshop, which Andrew Patrick and Scott Flinn of the National Research Council of Canada and Chris Long of Carnegie Mellon University organized. However, the thesis has a longer history. Informal anecdotes from veterans of the Orange Book (the essentially defunct US Department of Defense initiatives to specify and validate secure computing systems) lament how, in multilevel security (MLS) systems, everything ended up at maximum security level because otherwise it was too difficult to get any work done. Bob Blakely of Tivoli Systems jokes about the password rules: they should be too difficult for you to remember, but you should never write them down.

In their textbook, Charlie Kaufman, Radia Perlman, and Mike Speciner discussed the difficulty of designing security when humans are in the loop:

> "Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptably slow speed and accuracy when performing cryptographic operations. (They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed. But they are sufficiently pervasive that we must design our protocols around their limitations.)"[1]

The issue has also received some formal attention from HCI researchers. In their seminal paper "Why Johnny Can't Encrypt," Alma Whitten and Doug Tygar, now at the University of California, Berkeley, pointed out that perhaps the strong cryptography in a best-of-breed secure email package was rendered moot because even computer-literate, well-intentioned users found it difficult to use it correctly.[2] In the aptly titled "Users Are Not the Enemy," Anne Adams and Angela Sasse, both from University College, London, lamented that, "systems security is one of the last areas in IT in which user-centered design and user training are not regarded as essential."[3] They also quipped, "hackers

pay more attention to the human link in the security chain than security designers do." More recently, Ka-Ping Yee from UC Berkeley, published a set of guidelines for usable secure systems;[4] derived from Jerome Saltzer's and Michael Schroeder's, "Old Testament" principles,[5] the guidelines provide an interesting metric for evaluating whether the easy ways to use a system are in fact the most secure ways.

What's the problem?

My own path to this research area workshop came from looking at using public key infrastructure (PKI) to build systems that were not only trustworthy, but perhaps trustable as well: users would be able to make a reasonable decision about the system's trustworthiness (with sufficiently many caveats around what "reasonable" meant, of course). Work in defending systems had taught me that user–computer trust is dynamic and usually decreasing (for example, each new BugTraq mailing list announcement tells a systems administrator that a previously trusted system has a hole). Work with an industrial customer base taught me that different users have different opinions about what to trust. Public-key cryptography is a potentially unique technology to enable different types of relying parties to make meaningful trust judgments across boundaries. Unfortunately, when I started taking a critical look at combining standard PKI tools with standard information tools, I repeatedly found that things just didn't work.

You might think that the Web-server PKI with SSL would let users judge whether they had a secure connection to the intended machine; however, a malicious server can provide content that simulates the SSL and certificate signals in many browser configurations.[6] You also might think that using PKI to sign electronic documents and email ensures that a valid signature implies that the signer approved the contents and that the contents haven't changed. However, common off-the-shelf tools on common document formats (even without Word macros) can permit documents whose apparent contents change in usefully malicious ways without invalidating the signature.[7] You might believe that emerging client-side PKI with SSL would let service providers easily authenticate the client, but devious Web content can trick browsers—in common configurations—to use the personal private key to authenticate without the person being aware of or approving the action.[8]

Repeatedly, I ended up with problems because what computers are doing with cryptography doesn't match the mental model that humans have—end users as well as system programmers. Perhaps we should reexamine interfaces; perhaps we should be more careful with server programming; perhaps we need to rethink underlying technology. But it shouldn't be this hard. It should be easy for users and system builders to do the right thing; it also should be easy for each party to verify that the other party is doing the right thing.

The HCI workshop

Consequently, I spent a good deal of time with HCI researchers—as well as a few security and privacy folks and a few who do both—discussing issues of usability, security, privacy, and what the next steps are. The workshop's Web site (www.andrewpatrick.ca/CHI2003/HCISEC) has attendee contact information, links to their position papers, and a pointer to a discussion forum that currently has five times more subscribers than workshop attendees. In the small invitational workshop, each attendee had a chance to present experiences and ideas in this area. I'll summarize what each participant contributed (except for me, which I've already covered).

- *Angela Sasse* from University College, London—who earlier characterized security as the last area of computing to think about users—discussed many aspects concerning security technology's lack of usability in, and that both the HCI and the security community generally ignore this pending disaster. Because businesses tend to notice security/usability problems when "they show up on the balance sheet" (that is, cost money), examining ways to reduce expensive help-desk services can be fruitful. In particular, she critically examined the policy (standard in many installations that use password authentication) of locking users out after three failed login attempts (and requiring the help desk to reset the password before reactivating the account). This research establishes that, in environments

New department name; same mission

Sharp-eyed readers might notice that we've changed this department's name to Secure Systems, which captures its focus more accurately. Our goal is to bring security and privacy to real systems in the real world—where "system" must be broad enough to encompass everything that's necessary to make it work. In this department, we examine these systems-oriented issues "vertically"—the issues that arise when we build systems for specific application domains—as well as "horizontally"—specific aspects that don't show up until you actually try to field a real system. In this issue, I consider a horizontal component: the interaction between our computing technology and the humans that use them.

that enforce strong passwords, changing the number of tries from "3" to "10" substantially reduces costs and increases usability and user confidence (and perhaps compliance?) without impacting security. However, she stressed that her implicit message is that too many aspects of deployed security measures are dreamed up in the absence of any user context or empirical evidence—and that we suffer from a dearth of such evidence.

- *Alma Whitten* from UC Berkeley, followed up on her "Why Johnny Can't Encrypt" work by presenting her recent research on security user interface (UI) design techniques, which have resulted in effective security. Drawing on staged UI literature and on the American National Standards Institute standard for consumer product warning labels, her *safe staging* approach attempts to clearly delineate the dangers a user might face, but lets users gradually progress to advanced security techniques as they gain understanding and confidence. She's currently prototyping this in a PGP email client.
- *Ka-Ping Yee* from UC Berkeley, presented his usable security guidelines and noted problems in the prevailing viewpoint that software is secure if it meets its design specifications. As evidence, he observes that the Melissa and Love Letter viruses did not exploit software errors. Instead, they exploited the difference between the so-called functionally correct behavior of system software, and what its users expected it to do. Personally, I'd like to frame this point and place it next to the decade-old observation by Peter Neumann (SRI) that however much some in our community wanted to pillory Robert Morris Jr. for unauthorized access to systems, many of the access points he used required no authorization. We—the community—keep missing the point.
- *Paul Dourish* from the University of California, Irvine, also examined users' expectations. He hypothesized that security is a practical problem: users wonder, "is the thing in front of me right now secure enough for what I want it to do?" Consequently, it's critical that systems communicate the right information for users to make this decision. He quipped that "transparency" (as some systems people use the term) may only make things worse because "it's hard to see things that are transparent." He and his colleagues interviewed users to determine the mental models they had of system security. This work revealed several surprising things. For example, when queried about security, users nearly always answered in terms of things like "spam" and "threats to my physical person offline," suggesting that a secure system that does not protect against such things risks being perceived as a failure. Dourish's studies also found an age gap: younger users (who had grown up with computers) perceived security as an obstacle they had to work around. (This contrasted with Sasse's observation that in many work environments, the "ability to flaunt security regulations is a badge of seniority.")
- *Lorrie Cranor* from AT&T Labs discussed issues she encountered translating the complexity of the Platform for Privacy Preferences (P3P)—which might only make sense to lawyers and computers—into a UI that typical users might understand. She noted that those who drafted the specification never considered how these elements would be presented to users and also noted that even unsophisticated users possessed fairly sophisticated and nuanced privacy preferences, even though they could not always express them. To complement her talk, Cranor also demonstrated the resulting Privacy Bird UI, a browser plug-in downloadable from AT&T.
- *Carl Turner* from State Farm Insurance presented a fresh look at how users judge whether a site is trustworthy. Turner's approach demonstrated what we miss by just looking at the security technology; after an extensive study of user perceptions of security, his statistical analysis showed that the best predictor of "judged trustworthy" was the quality of the site's visual presentation.
- *Beki Grinter* from PARC commented on the difficulty of finding a common vocabulary for meaningful discussions with HCI and security people. She discussed her research into *implicit security*—embedding security directly into applications, rather than considering security design, configuration, and interaction in a "vacuum divorced from application issues." She stressed the importance of keeping users central in threat model security design—in aligning and inferring security state changes from a user's actions, and in communicating this state back to the user.
- *Dave Wilson* from IBM discussed experiences with security and usability in Lotus Notes and Domino from the perspective of

Younger users (who had grown up with computers) perceived security as an obstacle they had to work around.

end users, administrators, and developers. In progressive versions, his team has tried to recognize and mitigate HCI issues with security—for example, by consolidating several different security-relevant end-user UIs into one central UI. Wilson also noted real users' surprising behavior, such as overloading an access-control mechanism as a tool for tracking revision history (which caused usability problems when the access-control mechanism changed).

- *James Barlow* from the US National Center for Supercomputing Applications presented research concerning creating GUI tools to enable more effective network administration. "Know thy network" is a central network defense tenet, but perceiving threats and attacks in mountains of data is a daunting task. Security officers complain that advanced monitoring techniques are rendered useless by the last two feet between the telephone-book-sized weekly report and the eyes of the officer. Driven by this need, and by the concerns of the security operators they consulted, Barlow and his team prototyped interactive animated tools to more easily achieve the grail of "overall situational awareness" of the network.
- *Mike Just* from the Treasury Board of Canada discussed a taxonomy and usability analysis of credential recovery schemes as part of the Canadian government's Government OnLine initiative. To be effective, an authentication system for a large population of ordinary users needs to accommodate the fact that users will forget passwords, lose credentials, and so on. Angela Sasse pointed out that the "pure theory of knowledge-based authentication does not take into account how people think;" Just's work attempts to bring structure to that issue.
- *Lynne Coventry* from NCR discussed her research into usability and user acceptance of biometrics. This prompted a lively discussion regarding experiences with this technology, from the perspective not only of end users, but also from business managers and system designers. One participant recalled a business that deployed and then recalled a fingerprint system because dirty readers caused unacceptable errors in authentication—and the business had not allowed for the cost of someone cleaning them "every 20 minutes." Another recalled a system design with a major security flaw, which the designers had apparently overlooked due to the "sci fi" seductiveness of the biometrics. This seductiveness leads to other surprising behavior: Coventry pointed out that her users invariably found the idea of some of the systems unacceptably intrusive, after which they queued up to try it out.
- *Nate Paul* of the University of Virginia discussed work that used visual cryptography for mutual authentication in electronic voting, and noted the advantage of the "gee whiz" factor when users place their transparency over the cathode ray tube and watch the encoded message emerge. This work provides a nice counterpoint to standard so-called secure and private voting schemes in which it's hard even for a typical computer scientist to accept that the system works—let alone for a typical user.

What's next

The purpose of the workshop was not just to present our individual perspectives, but also to look at deeper issues. What are the basic problems? What can the HCI and security communities, working together, do to fix them?

A repeated theme was mental models and user expectations: the gap between what a system does and what users (and, perhaps, system programmers) think it does continually causes problems. A related point, which participants repeated, was the importance that a system effectively communicates the relevant data that lets a user decide whether to trust it for a particular purpose.

A truism in the security community is that a fundamental contradiction exists between making a system easy to use and making it secure. For example, anecdotes about the commercial pressure for vendors to ship systems with insecure default configuration usually cite this contradiction. Chris Long cited John Ousterhout, who claimed that security was "anti-CS" (computer science) because it got in the way of people getting things done. But is

Clarifications and corrections

In the premier of this department (Threats Perspectives, January/February 2003, p. 89), I used a "fairy dust" metaphor for cryptography (we hope that it magically makes the protocol work), and cited a talk by an anonymous member of our field's old guard. The old-guard member I had in mind was Roger Schell, whose invited talk "Information Security: Science, Pseudoscience, and Flying Pigs" (at the 2001 ACSA/ACM Annual Computer Security Applications Conference; www.acsac.org/2001/frames.html) skewered many aspects of modern security work. I had not felt an explicit citation was appropriate because it was an off-the-cuff remark, and the metaphor did not appear in his accompanying paper. In the interest of completeness, I must point out that the earliest published record of this metaphor is Bruce Schneier's discussion of "magic security dust" on page xii of *Secrets and Lies* (John Wiley, 2000). It's a good metaphor; my kudos to Bruce for thinking it up.

this fundamental contradiction real? Ka-Ping Yee insisted not; we can build systems that align security and usability. Others wondered whether the multiplicity of relevant parties might complicate things: an employee user, a corporation, and society as a whole (whose collective will is, in theory, expressed in regulations such as HIPAA) might all have different goals; scenarios certainly exist where furthering the goals of one party appears to restrict the actions of another.

Another matter of debate was whether better UIs would suffice to resolve security and HCI problems, or whether the underlying technology should be rethought. Alma Whitten was concerned that because policy is often based on a perception of what's possible with a UI, we need better UIs; other folks had experiences that suggested that at least sometimes, the technology itself doesn't work. Whitten also envisioned a model in which the security techniques with which users interact become an understandable tool common to a wide range of applications (much the same way the "print button" or file browsers or the trash can icon are now). However, for this to work, the security magic necessary for applications really needs to be a common abstraction—if we push too far, we might end up breaking the "implicit security" ideal that Grinter advocates.

In the group, tension existed between those who felt that formalism and perhaps even formal methods can help address the problem of characterizing and designing systems that are usably secure, and those who felt such approaches are inherently futile. What's really needed is for the security community to include some key HCI thinking about people-centered design.

Overall, a central question was why security seems to present some uniquely difficult HCI challenges. Perhaps it is because security considers the effects of active human adversaries; perhaps because security technology comes from military and mathematical settings that don't accommodate the full spectrum of human behavior; perhaps because (as Miron Livny at the University of Wisconsin once observed) security deals with the worst case, while the rest of computer science deals with the average case. With tongue in cheek, the HCI researchers at the workshop admitted that their field's answer to any question was, "if you considered human factors, all problems would be solved." However, having worked in many aspects of secure systems, I believe we could be on to something here.

Systems have security failures because configuration is too difficult; even conscientious users cannot understand security-relevant user interfaces; cryptographic implementations fail because the reality does not match the mental model the designers had; vulnerabilities exist in system software because complexity (and programming language flaws) get the better of conscientious implementers. A basic design tenet (the workshop organizers recommend Donald Norman's *The Design of Everyday Things*[9] as a good starting point) is that more careful design can reduce such human error. What would happen if we applied these principles to secure systems? Grinter suggests that we might have the "potential to do something unbelievably different." I think this could be fun. □

References

1. C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed., Prentice-Hall, 2002, p. 237.
2. A. Whitten and J.D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," *Proc. 8th Usenix Sec. Symp.*, Usenix Assoc., 1999; www-2.cs.cmu.edu/~alma/johnny.pdf.
3. A. Adams and M.A. Sasse, "Users are Not the Enemy: Why Users Compromise Security Mechanisms and How to Take Remedial Measures," *Comm. ACM*, vol. 42, no. 12, 1999, pp. 41–46.
4. K.-P. Yee, "User Interaction Design for Secure Systems," *Proc. 4th Int'l Conf. Info. and Comm. Sec.* (ICICS 02), Inst. for Infocomm Research/Chinese Academy of Sciences, 2002.
5. J.H. Saltzer and M.D. Schroeder, "The Protection of Information in Computer Systems," *Proc. IEEE*, vol. 63, 1975, pp. 1278–1308.
6. E. Ye and S.W. Smith, "Trusted Paths for Browsers," *Proc. 11th Usenix Sec. Symp.*, Usenix Assoc., Berkeley, Calif., 2002, pp. 263–279.
7. K. Kain, S.W. Smith, and R. Asokan, "Digital Signatures and Electronic Documents: A Cautionary Tale," *Advanced Comm. and Multimedia Security*, Kluwer Academic Publishers, pp. 293–307; www.cs.dartmouth.edu/~sws/papers/cmso2.pdf.
8. J. Marchesini, S.W. Smith, and M. Zhao, "Keyjacking: Risks of the Current Client-Side Infrastructure," *Proc. 2nd PKI Research Workshop*, Nat'l Inst. Standards and Technology, 2003; http://middleware.internet2.edu/pki03.
9. D. Norman, *The Design of Everyday Things*, Basic Books, 2002.

S.W. Smith *is currently an assistant professor of computer science at Dartmouth College. Previously, he was a research staff member at IBM Watson, working on secure coprocessor design and validation, and a staff member at Los Alamos National Laboratory, doing security reviews and designs oratory for public-sector clients. He received a BA in mathematics from Princeton University and an MSc and PhD in computer science from Carnegie Mellon University. He is a member of ACM, Usenix, the IEEE Computer Society, Phi Beta Kappa, and Sigma Xi. Contact him at sws@cs.dartmouth.edu or through his homepage, www.cs.dartmouth.edu/~sws/.*