When Security Gets in the Way

Donald A. Norman
Nielsen Norman Group and Northwestern University | norman@nngroup.com

I recently attended two conferences on usability, security, and privacy. The first, SOUPS (Symposium on Usable Privacy and Security), was held on the Google campus in Mountain View, CA, the second at the National Academies building in Washington, DC. Google is a semi-restricted campus. People can freely wander about the campus, but most buildings are locked and accessible only with the proper badge. Security guards were visible, polite, and helpful, but always watching. Our meetings were held in a public auditorium that did not require authorization for entrance. But the room was in a secure building, and the toilets were within the secure space. How did the world's security experts handle the situation? The side door of the auditorium that led to the secure part of the building and the toilets was propped open with a brick. So much for key access, badges, and security guards.

Both conferences were attended by experts in usability, security, and privacy. Both conferences emphasized that if we ever are to have systems with adequate security and privacy that people are willing to use, then the three fields must work together as a team. Without usable systems, the security and privacy simply disappear as people defeat the processes in order to get their work done. My experience at Google illustrates the point.

The numerous incidents of defeating security measures prompts my cynical slogan: *The more secure you make something, the less secure it becomes*. Why? Because when security gets in the way, sensible, well-meaning, dedicated people develop hacks and workarounds to defeat it. Hence the prevalence of doors propped open by bricks and wastebaskets; passwords pasted on the fronts of monitors or hidden under the keyboard or in the drawer; house keys under the door mat, above the door frame, or under fake rocks that can be purchased for this purpose.

We are getting a mixed message. On the one hand, we are continually forced to use arbitrary security procedures. On the other hand, even the professionals ignore many of them. How is the ordinary person to know which ones matter and which don't? The confusion has unexpected negative side effects. I once discovered a computer system that was missing essential security patches. When I questioned the computer's user, I discovered that the continual warning against clicking on links or agreeing to requests from pop-up windows had been too effective. This user was so frightened of unwittingly agreeing to install all those nasty things from "out there" that he denied all requests, even the ones for essential security patches. On reflection, this is sensible behavior: It is very difficult to distinguish the legitimate from the illegitimate. Even experts slip up, as the confessions reported occasionally in various computer digests attest.

The situation with security is similar to the situation once faced by the human-centered design community. In the early days of software development, programmers and engineers devised the systems, sometimes giving in to feature lists from the marketing community. After they had finished, they would ask the usability and technical-writing communities to make it usable and understandable (and the designers to make it pretty). It is only when practices change to enable all of these groups to work as team members from the project's start that improvements occur. So too with security and privacy, except in this case, the security and privacy professionals are the outcasts. It is time to make them first-class citizens who work with the product team throughout the entire development cycle to produce cohesive systems that are understandable and usable, functional and safe, secure and private.

If this endeavor is to be successful, we need more understanding of the issues; better tool kits to deliver to developers; and a comprehensive set of tools, scripts, and templates for the administrative support staffs around the world so that the rules and polices they develop will be consistent both with one another and with the best prac-

tices of the security and privacy community. Today we lack a deep understanding of critical things—including people's conceptual models of security and privacy, what new models might be effective, and how to present the conceptual models so that they are both effective and unintrusive. The programming community needs better tools to defeat attacks. More research must be done, and the purpose of the meeting at the National Academies was to develop such a research plan.

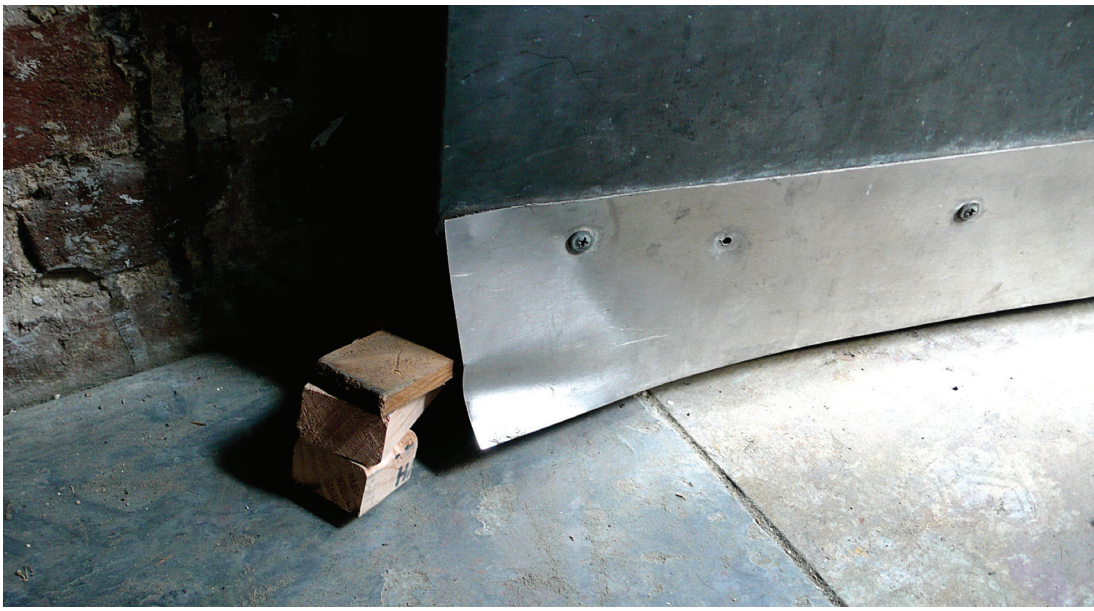
Usability or security: Do we really have to choose? At times the two seem immutably bound. Make it more secure, goes the belief, and as night follows day, things become harder to use. It is a never-ending challenge, with security experts pitting themselves against usability experts, and both fighting with the engineers and marketing representatives—all convinced that their view is the most important, each convinced that attention to the others defeats their goal.

Does added security make things more difficult to use? Will people always resent the extra steps? The answer to both questions is the same: not necessarily. Consider the physical world of doors and locks. Locks on houses, cars, and private records get in the way of easy access, but we tolerate them because they seem necessary and the amount of effort they demand usually seems reasonable. Note the two different components: the understanding of the necessity for protection and the reasonableness of the effort required. Both are design issues. And both require at their base a coherent, understandable conceptual model of both the need for security or privacy and the workings of the mechanisms that enforce them.

Note that the terms "security" and "privacy" oversimplify a complex set of issues. Thus, we often confuse the problem of identification (who is this person?) with that of authorization (is the person or system using the system authorized to do so in the way requested?). In many cases, authorization can be established without identifying people or systems, just as identifying the person or system need not establish authorization.

In the real world, identification and authorization are confounded, requiring needless identification checks that impair privacy yet overlook the important issue of authorization. Medical systems sometimes overenforce privacy concerns, in part because of overinterpretation of government-regulated privacy policies, impairing the very services they are designed to support.

Note that different groups are involved, each requiring a different form of design assistance. System developers provide the underlying mechanisms, but the IT administrators at the various sites determine just how those policies will be enforced. The IT staff is under considerable pressure from their own administration to reduce security and privacy concerns, but to do so they must be well versed in technology, in the law, in the needs of the user community, and in the psychology of both legitimate and illegitimate users. What the community needs is a set of standardized scripts, templates, and system tools that allows them to implement best practices in ways that are both effective and efficient, standardizing interactions across systems in order to simplify the lives of users, but still tailoring the requirements to any special needs of the organization. These tools do not exist.



Photograph by Conrad Bakker

Northwestern University's Password Requirements

IT MUST:

- be six to eight characters in length.
- contain a nonalphabetical character such as 1 2 3 ! $ & * , ? + =
- contain one or more nonalphabetical characters between alphabetical characters (example: "A3b," "j3;M").

IT CANNOT:

- be a password used in the previous 18 months.
- be similar to your previous password.
- be a palindrome (example: "abc.cba," "cb,,bc").
- include the characters # @ ' " ` \
- include character strings from any part of your name (examples for "James Smith": "James," "ames," "mith").
- include part of your name with uppercase characters (example: "AmeS").
- include part of your name with numbers substituted for letters (example: "j4m3s").
- include reserved strings (example: "wild," "cats").
- include part of reserved strings with uppercase characters (example: "WilD," "cAtS").
- include part of reserved strings with numbers substituted for letters (example: "wi1d," "E3e3").
- include repeated letters or numbers (example: "AaAa," "3E3e," "aaaa").
- include runs of adjacent keys (example: "hjkl").

In the absence of standard guidelines and adequate tools, different systems implement the same policies with very different philosophies and requirements, complicating life for users who must use multiple systems. Developers who lack an understanding of real human behavior tend to impose logical rules and requirements upon a bewildered, overwhelmed audience. The audience, either not understanding the rationale or simply disagreeing with the necessity for the procedures imposed upon them, sees these as impediments to accomplishing their jobs. Moreover, the systems people may lack an understanding of the clever ruses and social engineering skills of the illegitimate users, who break into systems the easy way: by lying, stealing, and deceiving. The strongest locks in the world do not deter the shrewd social engineer.

Only amateurs attack machines; professionals target people.

—Bruce Schneier

Does more security necessarily translate into less usability? No. A more secure lock is no more difficult to operate than a cheap, insecure one. With appropriate technology we can make some of these systems even easier to use while simultaneously enhancing security. Biometric identifiers and radio frequency identification (RFID) tags offer great potential on this front.

Want a classic example of a failure? Passwords. There are several myths in the world about security, but the most pervasive of these has to do with password security. Look at Northwestern University's password requirements: an overreaction to the problem of password discovery through brute-force attacks (see sidebar above). Breaches can occur, especially when people use simple, common passwords or the names of their family or pets. There are easy-to-find lists of common passwords, and personal information is readily obtained, sometimes supplied by the victims themselves on the social sites they belong to. Systems do need to be engineered to prevent rapid, prolonged attempts at finding passwords, but there is no need to go to onerous requirements—just long enough, nonword combinations that sufficiently slow up the guessers. Lacking sophisticated understanding of the nature of password theft, system administrators often take the extreme measures exemplified by Northwestern's IT department, even though security professionals continuously point out the harm such schemes produce. This is another example of where usability experts can collaborate with security professionals to develop secure yet humane solutions.

Although there is much emphasis on password security, most break-ins occur through other means. Thieves usually don't break into systems using brute force. They phish, luring unsuspecting but helpful people to tell them their login name and password. Or they install sniffers on keyboards and record everything that was typed. A password's strength is irrelevant if the thief already possesses it.

Passwords illustrate several of the difficulties with current methods of enforcing security. Passwords are the least expensive mechanism known for securing systems. But onerous password requirements reduce security and increase costs. They reduce security because many people either use trivial algorithms to generate memorable passwords, or write down their passwords and store them in easily discoverable places. They increase costs because help desks must be staffed to handle the large number of users who have forgotten their passwords and can no longer log in. These requirements keep out the good guys without deterring the bad guys.

Another problem is scale. Most of us have tens or even hundreds of passwords. If a single password exceeds the limits of nor-

mal memory skills, what about hundreds of them, each with different requirements?

What do security professionals do? I asked attendees at the security conferences. Many of the security experts said they do "what everybody does: have two passwords." They have a simple, easy one for everyday use (all those Internet sites that require logins for no apparent reason) and a complex one for places they care about. "Only two?" I asked, surprised. I can't get by with two; many places have obscure conflicting requirements about length, or how to start or end, and requirements for—or prohibitions against—weird characters. Or they make me change my password many times a year, never permitting the use of old ones. "Oh, yeah," they admitted, "we write those down." Is this a wise policy? No, but how else can one cope?

Bruce Schneier, security professional, has it right: Passwords are not where the action is among thieves and spies. It is like the front door of your home. Professional thieves don't even bother. They know the front door is solid, with expensive locks and bolts, but the rear and side doors are often left unlocked or equipped with cheap locks. While we carefully lock the doors, they get in through the windows, or by finding the key hidden above the threshold or under the potted plant. The easiest way to break in to a security system is to lie and spy, or to use their technical terms, "social engineering," or "key logging" or "sniffing" and "shoulder-surfing." Send out phishing attacks, install key loggers, peek over people's shoulders at public terminals (or install a video camera to do it for you).

Think about it: The big losses occur when someone accidentally emails important information, or loses an unencrypted hard drive or personal computer. One notable example of the "good news/bad news" category illustrates the problem. The bad news is that someone lost a hard drive with confidential information. The good news is that it was encrypted. The bad news is that because the password was so difficult to remember, it was taped to the drive in plain sight. Here is a case where usability should not have triumphed.

Both security and privacy are difficult problems. We need systems that are easy to use for their intended purposes or by the intended people, but difficult for unauthorized people or uses. For these purposes we need components not normally considered in simple product design: means of authenticating identities or authority, needs, and permissions. Some of this will require physical tokens, biometric identifiers, and private information. Some of this requires rules and policies, sometimes editable by the user of the system; sometimes editable only by authorized administrators; sometimes buried in the code and unchangeable without significant development costs.

Finally, we need more humane ways of interacting with the systems to establish eligibility, ways that the people who use them consider appropriate and reasonable. This means that the systems must be accompanied by a clear and understandable conceptual model. If people could understand why they were required to do these things, they would be more willing to pay a reasonable penalty. We all willingly adapt to the inconvenience of locks that seem reasonable for protection, but not those that get in the way—as the propped-open door at the security conference indicates.

We have a wonderful design challenge before us. It is time to make systems that are more secure, that enhance privacy, and that are still eminently usable. We need systems that are effective at performing their tasks, while providing high quality of user experience at reasonable cost. The solution is going to require sensible analyses, the development of appropriate technologies—probably including automation, enhanced interaction protocols, and interfaces with better feedback—and the development and continual communication to support the development of an appropriate conceptual models. The only way this will happen is if all parties work together as a team from the start. With notable exceptions, the security and privacy concerns have been addressed by the security and privacy experts, coupled with the arbitrary rules and policies of system administrators, where these concerns have been tacked on to existing systems as an afterthought.

Usable security and privacy: It's a matter of design.

ABOUT THE AUTHOR Don Norman wears many hats, including cofounder of the Nielsen Norman Group, professor at Northwestern University, and author. His latest book is *The Design of Future Things*. He lives at jnd.org.

DOI: 10.1145/1620693.1620708

Further Reading:

Adams, A., and Sasse, M. A. "Users Are Not the Enemy." *Communications of the ACM*, 42, 12 (1999): 40–46.

Anderson, R.J. *Security Engineering: A Guide to Building Dependable Distributed Systems*. New York: Wiley, 2007.

Florêncio, D., Herley, C., and Coskun, B. (2007). "Do Strong Web Passwords Accomplish Anything?" Paper presented at the Proceedings of the 2nd USENIX workshop on Hot topics in Security, Boston, MA. http://www.usenix.org/event/hotsec07/tech/full_papers/florencio/florencio.pdf and also http://research.microsoft.com/pubs/74162/hotsec07.pdf

Schneier, B. *Secrets and Lies: Digital Security in a Networked World.* New York: Wiley, 2000.