The systems we worry about securing include the people who use them. In everyday offline life, an average person’s “security policy” consists of a few simple, intuitive rules. Most people tend to acquire them rather quickly, even in unfamiliar and complex environments (such as a large, unfamiliar city), and they typically prove sufficient. Offline, we’re quite capable of not subjecting ourselves or our property to unnecessary risk—we all know not to give keys to strangers, leave our doors unlocked, or walk in dangerous urban locations after dark. When we violate our basic safety rules, we’re typically fully aware of doing so and of the possible consequences.

Why, then, do we face a continuing epidemic of poor security-related decisions online—by the same people who apparently take proper care in their numerous offline transactions? Why do offline security-related intuitions that tend to form quickly and save the day seem to continually fail to emerge for online behavior? We can only assume that some significant factors are interfering with people’s abilities to formulate simple security rules-of-thumb and develop proper intuition, despite repeated negative experiences.

We believe that the majority of users continuously employ risk analysis heuristics to plan both their online and offline actions; the overwhelming problem of online security is that this analysis, in the online case, is based primarily on entirely wrong assumptions, intuitively derived from incorrect interpretation of GUI elements and processes. We propose a core user interface design principle for the designers to follow when considering and building trust-decision-related user interface features.

**The user interface problem**

Today’s typical computer user actively engages in casual browsing, carrying out financial transactions via Web and email, exchanging documents over email, sending instant messages, and similar activities. Such a user’s security situation, however, is nothing short of deplorable. At the heart of the matter are bad trustworthiness decisions. Users’ incorrect assumptions about processes involved in computer transactions lead to false or incomplete models of these transactions, which then result in decisions that more informed users would classify as obviously bad. However, we assert that an effective general approach to mitigating this problem hasn’t emerged—the problem remains unsolved. Moreover, we keep seeing new interfaces that exacerbate rather than mitigate this issue.

We argue that user interfaces themselves promote such incorrect assumptions. Designed for convenience and simplicity (for a particular task, from the designer’s viewpoint), the interfaces fail to consider security. Moreover, interface features purporting to achieve “simplicity” often eliminate details that could let users distinguish between good and bad choices in a given security situation. The consequence of this engineering process is a failure to let users form more discriminating security models and develop the corresponding intuitions.

We accept as fact that most computer users possess real-world intuitions that help them make correct trustworthiness decisions and protect their assets and privacy in most situations unrelated to computers. Although they might be flawed, these intuitions function well enough for most people. Rather than building on these real-world intuitions, however, problematic interfaces tend to train users away from applying them or fail to account for their likely misapplications. Consequently, users don’t realize the real implications of their actions.

We consider several scenarios in which this happens and propose a new principle for designing user interfaces that involve trust and privacy decisions.

**Training users away from correct intuitions**

Several fundamental examples show how computer interfaces fail to give users critical information relevant to making correct trust decisions.
How can reading a piece of paper harm me? The “piece of paper” icon for documents belies the fact that they might contain possible active content (code). By “opening” or “reading” them, users might in fact be surrendering control of their computers or user accounts to a third party. In fact, the very choice of the term “open” to describe an entire range of actions, from launching a well-known trusted program to executing arbitrary code of unknown origin, is unfortunate and confusing to users. It makes the task of explaining security risks of such actions and pitfalls very hard.

How can sending a piece of paper harm me? Complex formats such as Word and PDF have become the standard way to package documents electronically. Application interfaces make these documents look like pieces of paper, so users make trust decisions about the electronic files based on the contents of the virtual paper they see. However, it’s been well documented in many amusing (for us) cases that the electronic version can contain far more data than intended.

How can reading an email harm me? General intuition suggests that merely opening a letter—paper or electronic—shouldn’t be in any way hazardous. However, HTML-based mail-reading programs do more than merely display a message’s text. Attempts by such mail-readers to access external content or handle embedded non-text content frequently lead to security problems.

How can a phone call harm me? A flaw similar to those just mentioned exists in the interface design of Bluetooth-enabled phones. A phone asks the owner whether he or she wants to “accept a message.” The message in question can be either a text via SMS or a program that immediately runs on the phone’s operating system; however, the question asked in both cases is the same. Thus, users have no basis for distinguishing between these two very different levels of risk when making a trust decision.

Was that a “safe” or a “bad” double click? Numerous researchers have pointed out that the consistent use of double-clicking to both launch programs (a high-risk action in case of an unknown program) and open a text document (which users either imagine is a low-risk, everyday action or know is one they must perform routinely, ignoring any misgivings in order to do their daily work) has led to confusion and has been the source of a whole class of bad trust decisions. In the current desktop paradigm used by Microsoft and Apple, a double-click is the most basic user action for performing a range of operations with vastly different security implications. From untrained user’s viewpoint, double-clicking a file’s icon means “do what needs to be done for this file, automatically.” Thus, concepts such as “executing a program” (unsafe for programs of unknown origin) versus “using a trusted program to interpret a document file” (safe when the user believes the program to be trustworthy) are merged in the user’s mind. The first author recalls the shock of discovering that even advanced Windows and Mac OS users had a hard time comprehending what Unix “execute” permission bits were for—no separate concept of “program execution” existed in their cozy desktop world.

Launching a program typically amounts to giving away (to the program’s authors or owners) any privileges that users have under their current account, whereas running a known program with a document without active content as input has no such implications. It might still result in control transfer if the program contains a vulnerability and the input document is crafted to exploit it; however, this condition isn’t usual.)

Who was that masked man? In the physical world, users decide to trust otherwise unknown strangers based on attributes that can (hopefully) be authenticated. We assume that the uniformed person in the police car are police; we assume that the white-coated person about to administer a hypodermic injection in the hospital is, in fact, a qualified medical person. In neither case do we care what their full names or social security numbers are. However, the online world has focused on such things as user IDs and “distinguished names”—not attributes.

In the physical world, users make decisions based on an interaction’s context and evaluate information based on its source. People might trust the third author (Sean Smith) for opinions about bicycle technology but not about cars; people might be suspicious of the second author (Chris Masone) claiming to be an ordained minister, if the ordination came from a mail-order mill. The online approach to authentication bungles this as well; just because Alice’s browser successfully carries out SSL authentication with foo.com doesn’t mean that Alice can trust that site to actually be competent at its business. Furthermore, Alice might have to work quite hard to find out who said it was foo.com (it’s a fact that Chris has used to create validly signed email from Dartmouth’s president).

Our proposed principle

These grumbles have led us to formulate a principle:

“When a user must make a trust decision, the appropriate interface action should evoke the matching real-world intuition connected with surrendering a degree of control over user’s assets or privacy. Different secu-
In the current psychological climate, the concept of phishing is less abstract by forcefully invoking relevant real-life intuitions in the very GUI through which users implement their decisions.

Related work in human-computer interaction and security

Various sources have identified inaccurate mental system models as a source of user error. Don Norman\(^1\) discusses this idea as it relates to systems that range from door handles to computer software. The long-known “Principle of Least Surprise”\(^2\) tells software engineers that their programs should act as the user expects. Apple has included a discussion of the importance of discovering—and designing for—users’ mental models in the Apple Human Interface Guidelines for years.\(^3\) These sources discuss this concept as a general software or user interface design principle, but not as a specific challenge for securing software.

Alma Whitten and Doug Tygar\(^4,5\) as well as Simson Garfinkel\(^6\) explore the security-specific challenges of matching system design to users’ mental models and derive a set of design principles from their studies. Whitten, in particular, describes security’s lack of feedback property, which tells us that it’s extremely difficult to provide feedback about a system’s security state that’s both meaningful and comprehensible to the average user. The hypothesis we present in the main text can be viewed in terms of this property: years of security applications providing poor or oversimplified feedback have engendered within users a continually worsening set of mental models. She also describes metaphor tailoring, a design methodology that provides guidelines to “construct, piece by piece, new visual symbols that effectively convey a carefully designed set of cues to the user.”\(^4\) We propose applying this methodology to make the representations of private, work, and public zones in our experiment—and aligning the metaphor with users’ real-world security intuitions.

In his recent Communications of the ACM article,\(^5\) Ryan West gives a broad overview of psychological factors affecting users’ security decisions—from underestimating the degree of risk applied to them personally and preferring a risky gamble to guaranteed loss, to disparity in perception of losses and gains—and makes several general recommendations for improving the current situation. Notably, he argues that security remains an abstract concept to users, which affects the analysis of trade-offs between risk, gains, and losses that drive their trust decisions. We agree, and propose a way of making it less abstract by forcefully invoking relevant real-life intuitions in the very GUI through which users implement their decisions.

References

browsers, from obvious design flaws, such as Internet Explorer’s URL location bar being too short to show a full address, to features such as a JavaScript interception of the mouse-over event generated by the browser when users placed their cursors over the link, replacing the actual target URL with a fake, legitimate-looking one. It turns out that users’ trust in the oracle was misplaced to begin with because they didn’t understand the oracle’s actual behavior.

Thus, according to our analysis, the interface’s principal problem was in leading users to develop a false concept of the oracle in which they place their trust, failing to disclose that, in so many words, the record that was supposed to arrive from the trusted phone directory arrived from the same people as the original “call.”

In our analysis, the interface should have clearly defined the trusted party, so that the user understands who or what is the trusted oracle that answers the question, “Where is this link going to lead?”

**Outsourcing Web-based services**

Even assuming that the oracle in the example just given is fixed and always correctly identifies the destination URL, a tendency to outsource sensitive Web services threatens the very foundation on which users can make trust decisions: the target’s domain name. Namely, at some critical point during a transaction, users are redirected to a third-party domain apparently unrelated to the organization they were doing business with. Again, instead of building on a useful real-world intuition, implementational convenience is paving the way for larger problems likely to result in substantial costs.

We can reasonably assume that users have some idea of a typical company’s organizational structure and project that concept on the domain name hierarchy. Thus, when giving away personal information or control to, say, billing.somebank.com, users retain the idea that they’re dealing with the billing department of Somebank.com—that is, a unit of Somebank. When, however, Somebank routinely forwards her to, for example, paybills.billing.com, she loses this useful intuition (and every chance of eventually deriving it from examples set by trusted sites) and will likely stop making connections between URLs and entities entirely.

Worse, due to a site designer’s decision to remove obvious distinctions between safer and potentially riskier choices, users are put in the position of making trust decisions blindly. In our analysis, this design mistake isn’t much different from the double-clicking debacle already discussed.

We propose that, as in the previous cases, user awareness should be directed to the fact that they’re giving away privacy or control to an external party; they’ll then be more likely to question that party’s identity and its relationship with the entities they trust.

We believe that once offline security intuitions and related entities are mapped closely enough to online ones, we’ll see a substantial improvement in the security of users’ online behaviors.

**References**


**Sergey Bratus** is a senior research associate at the Institute for Security Technology Studies at Dartmouth College. He has a PhD in mathematics from Northeastern University. Contact him at sergey@cs.dartmouth.edu; www.cs.dartmouth.edu/~sergey/.

**Chris Masone** is a PhD candidate at Dartmouth College. His research interests include usable security and hybrid PKIs. He expects to receive a PhD in computer science from Dartmouth this summer. Contact him at cmasone@cs.dartmouth.edu.

**Sean W. Smith** is an associate 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. He has a BA in mathematics from Princeton University and an MSc and a PhD in computer science from Carnegie Mellon University. Contact him at ssw@cs.dartmouth.edu; www.cs.dartmouth.edu/~sws/.