Designing an Evaluation Method for Security User Interfaces: Lessons from Studying Secure Wireless Network Configuration

Cynthia Kuo | Carnegie Mellon University | cykuo@cmu.edu
Adrian Perrig | Carnegie Mellon University | adrian@ece.cmu.edu
Jesse Walker | Intel Corporation | jesse.walker@intel.com

SPECIAL SECTION HCI & SECURITY

Ten or 20 years ago, evaluating security products was not as much of a problem as it is today. Systems were managed by people able—and willing—to master the complexities. However, with the proliferation of personal computing devices and network connectivity in the home, systems are now regularly managed by nonexperts. Each system needs to be secured by each user in each home. Therefore designing effective, unbiased evaluation methods for consumer products is one of the first steps in improving both users' experiences and their security practices.

Evaluating the usability of security is a challenge. A common question evaluators face is: "How do I test whether users will configure and use a product securely?"

In this article, we outline problems we encountered in evaluating secure wireless network configuration and examine the assumptions many user study methods make, but which may not hold for security.

EMPIRICAL TESTING FOR SECURITY

When users are not domain experts, designing empirical evaluation methods for security can be tricky. For example, imagine that you want to test users' ability to enable encryption in a wireless network. Your first instinct might be to suggest a usability study that requires that you provide a task list your test participants should attempt. However, instructing participants to enable encryption may be problematic: Do participants know what encryption is? Should you provide a general description, such as suggesting that "no one can eavesdrop on your data?" This might be even more confusing: Participants might know what encryption is but misconceive its purpose. Whatever the wording, the task list itself may give users information they did not know before the study. Furthermore, a task list may guide users to perform tasks that they might not attempt outside of the lab; users who are unaware that a feature exists almost certainly will not configure it on their own.

Alternatively, the study could educate users about encryption so that you can ask them to enable it. However, educating users in a study has its problems. It is unlikely users in the "real world" would have the same educational experience. Who would teach them? In addition, the study may become a test for the quality of the educational material, rather than a test of the application. Moreover, educational materials reveal the purpose of the study, and users often tailor their responses or behavior to please the researchers.

The issues raised in this section touch on the problems we encountered when we evaluated 802.11 network

Method	Strengths	Drawbacks	Application in Our Study
Mental models interviews	Understand people's conceptualization of a topic; may discover unanticipated viewpoints	Time-consuming; hard to analyze free-form responses; social acceptability biases	Analyze users' mental models of wireless technology. Responses were matched with responses to more specific survey questions.
Surveys	Quickly gather large amounts of quantitative data on people's attitudes	Social acceptability biases; may miss important insights if questions are too narrow	Measure within-subject attitude changes. We asked participants to complete a questionnaire twice: before and after working with an access point. Because attitude ratings are subjective, only within-subject changes were used.
Contextual inquiries	Observe what people do in their normal environment	Time-consuming; unstructured; best for primary goals	Evaluate which tasks participants would attempt on their own. The experimenter initially allowed participants to configure the access point without guidance. Later, we intervened to direct participants to the security tasks they failed to attempt independently.
Usability studies	Gather structured, quantitative results in a limited amount of time	Assumes familiarity with underlying concepts and tasks	Evaluate participants' ability to complete the set of five tasks.

Table 1: Strengths and Drawbacks of Various Evaluation Methods

configuration. To fully understand the complications of evaluating security, we need to examine the assumptions existing evaluation methods often make.

ASSUMPTIONS THAT MAY NOT HOLD FOR SECURITY

Traditional HCI methods were developed with assumptions that may be inapplicable for security. After designing and conducting several user studies on secure 802.11 network configuration, we identified five common assumptions.

Assumption 1: *There are clear-cut criteria for success.* For many applications, it is easy to evaluate whether user-study participants have achieved a goal. Were participants able to find the correct form? Could they complete a transaction? These are usually clear-cut, black-and-white judgments.

Evaluating security is often more nuanced, like separating shades of gray. Computer security is a risk-management process. Each user may be exposed to different risks, and, as a result, may require a different configuration. Questions such as "Is it secure?" elicit noncommittal answers from security experts, such as the common (but infuriating) "It depends." For example, a countryside resident whose closest neighbors are herds of deer may not need to secure her wireless network. Someone living near self-described computer hackers has much higher security needs. Security admits no one-size-fits-all solution, and evaluators must consider this ambiguity when developing evaluation criteria. Users' responses will vary based on their assumptions about the situation.

Assumption 2: *Applications should tolerate variation in user behavior and user error.* Many applications enable multiple paths through the user interface to the same end state. For example, Windows users can right click on a file and choose "Delete" to send a file to the Recycle Bin, or they could also drag the file to the Recycle Bin. Both methods reach the same end state. From an evaluator's perspective, it may not matter which method users implement, as long as the correct file is deleted. Moreover, there is room for error; accidentally deleted files can be recovered from the Recycle Bin.

Security tends to be more fragile. For example, suppose a novice user installs a wireless network. He lives in a dense urban environment and knows that security is important. However, he is not proficient with networking, and proper setup takes several hours. He first establishes Internet connectivity and later configures the security features. In the meantime, someone nearby connects to his still unprotected network, notices some shared files, and downloads confidential material. The unsuspecting user may never know this information was released. Additionally, there may be no way to "undo" the ensuing damage (e.g., sensitive photos were posted on the Web). Whitten and Tygar [2] call this the "barn door" property of security. Since every user action taken might introduce some security vulnerability, evaluators must consider more than the end result only.

Assumption 3: *Users are familiar with the underlying concepts.* Many methods assume users understand the underlying concepts behind the study. For example, consider a basic, textbook usability study that examines whether users can buy a book online. Participants are given a book name, the bookstore URL, and perhaps the buyer's information (e.g., a name, address, and

credit card number). Researchers can safely assume the subjects understand what books are, how to purchase items, and how to use a Web browser.

Evaluating security is more difficult, however. Many users do not understand—or may not be aware of—the threats associated with a technology. For example, 802.11 wireless networks broadcast messages, meaning any device within radio range of the network can capture the transimissions. If the network is insecure, unauthorized parties could read emails sent over the network, observe what Web sites are visited, and so on. Securing a network requires users to decide which devices may access the network and whether to encrypt data transferred over the network. This requires an understanding of the underlying technology—which many users may not have. In a user study, asking participants to enable encryption or to ensure no one can "eavesdrop" on their data may give participants information they did not previously have: that a wireless network makes their data public. Researchers must be careful about assuming what knowledge users possess. *The very act of providing a set of evaluation tasks may introduce a bias into a user study.*

Assumption 4: *Users' tasks are their primary goals.* Most HCI methods were developed to evaluate primary goals, i.e., the main objectives users seek to accomplish. For example, consider a study of how librarians use library management software to locate books. A textbook contextual inquiry would be ideal for learning how the librarians actually use the software and what features they find valuable. This works when the goal is to study users' primary tasks (e.g., locating books), but often fails for studying security.

As Whitten and Tygar noted, security is usually a secondary goal [2]. The primary goal of most software users is to get their work done, not to fiddle with security settings. The primary goal for a consumer deploying an access point is to establish (and use) wireless network connectivity—not to enable encryption or populate lists of authorized devices. In addition, technology marketing may set unrealistic expectations that everything "just works," thereby making users unhappy when they need to expend effort for security. Researchers evaluating security must consider that users interact with security features as little as possible—or not at all. As a result, study designs need to account for the secondary nature of security-related goals.

Assumption 5: *Users will respond without bias.* Study participants sometimes try to please the experimenter by saying what they think the experimenter wants to hear. Experimenters must consider whether users have an incentive to misreport responses. This is true for any type of study, security or otherwise.

In security studies, social acceptability can bias the experiment in opposite ways. Users may know that they *should* do something about security, but they lack the time, understanding, or concern to properly address the problems. In our preliminary studies, we observed that telling participants we were studying security may have encouraged them to exaggerate their level of concern. For example, one participant discussed the importance of using encryption to maintain data privacy, but he had not enabled encryption on his home network. In the opposite direction, Weirich and Sasse document how secure behavior may be

viewed as paranoid, nerdy, or anti-social [1]. These impressions may bias users' responses in the negative direction.

Social-acceptability biases commonly affect attitudinal studies, such as surveys, focus groups, and interviews. Often, studies must be designed with some type of subterfuge to avoid social-acceptability biases. This becomes even more challenging for security, because experiments must design an experiment, with subterfuge, to test a secondary goal.

Because of these assumptions, many standard methods in an evaluator's tool kit resist application to security software. This is a serious problem: It is difficult to improve the quality of security software without good evaluation methods.

DESIGNING AN EVALUATION METHOD

After trying several user-study methods, we designed our own study by adapting several different techniques: mental models interviews, surveys, contextual inquiries, and usability studies. The study was designed to exploit the strengths of each technique while minimizing the drawbacks. This is briefly summarized in Table 1. For evaluating other applications, these factors should be considered when designing an appropriate study.

CONCLUSION

For most users, thinking about security is a demanding task. It entails imagining how other people may use or take advantage of a technology; making value judgments; performing risk-benefit analyses; and configuring a user interface. Each task may appear tractable by itself, but the combination poses an intimidating challenge.

For application designers and evaluators, a user interface is merely the surface surrounding a complex problem. To truly improve the user experience of security, we must delve into the interaction between a technology and users' value systems. Standard user testing methods may be inappropriate for testing security. These methods often make assumptions that do not hold for security, and it is necessary to modify existing methods for testing.

ACKNOWLEDGEMENTS *This research was supported by a gift from Intel. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of Carnegie Mellon University or Intel Corporation.*

REFERENCES **1.** Dirk Weirich and Martina Angela Sasse. Pretty Good Persuasion: A First Step towards Effective Password Security in the Real World. In *NSPW '01: Proceedings of the 2001 Workshop on New Security Paradigms*, Cloudcroft, New Mexico, USA, 2001. **2.** Alma Whitten and J. D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, Washington, D.C., USA, August 1999.

ABOUT THE AUTHORS

Cynthia Kuo is a PhD student in the Engineering & Public Policy department at Carnegie Mellon University. Her research is currently focused on finding human-friendly ways to promote Internet security.

Adrian Perrig is an assistant professor at Carnegie Mellon University. He has appointments in the departments of Electrical and Computer Engineering, Engineering and Public Policy, and Computer Science. His research focuses on networking and systems security, security for mobile computing, and sensor networks. Other research interests include human interfaces for security, networking, operating systems, and cryptography.

Jesse Walker is a principal engineer in Intel Corporation's Communications Technology Lab. His primary interest concerns network security protocols. Dr. Walker served as editor for IEEE 802.11i and has contributed to many other IEEE 802.11 amendments. He also has contributed to numerous IETF standards. Prior to joining Intel, he worked at Shiva, Raptor Systems, Digital Equipment Corporation, Rockwell International, Datapoint, and Iowa State. He holds a PhD in mathematics from University of Texas.