Independent Research: Intersections of Human-Computer Interaction and Computer Security

Peter Story

May 13, 2015

This semester I read many papers at the intersection of Human-Computer Interaction (HCI) and Computer Security. These papers emphasized the importance of considering users when designing secure software. Without understanding users' needs and the ways they think, it is quite likely that software which is designed to be secure won't be used securely. As a very simple example, if password strength requirements are too demanding, users will write their passwords down, negating the security of complicated passwords.

Reading these papers has informed my thinking on HCI and Computer Security, and given me ideas for further research. For example, several papers emphasized the importance of users having an accurate understanding of what their computers are doing. Consequently, I've started noticing when my computer does things that I wouldn't expect, and considering the security implications. One example is that Safari starts pre-loading pages as you type a URL, even before you press "Enter." In a country where your browsing history could get you in trouble (like in China), this could be a dangerous feature that most users don't know about.

References

[1] Communications of the ACM. 57(9):1–108, September 2014.

Contains two very interesting security articles: "Accountability in Future Internet Architectures" and "Security, Cybercrime, and Scale." "Accountability" describes the desirability

of accountability in a future internet architecture. It could hold ISPs accountable for throttling and make it easier to assign responsibility for network hijacking attacks. The author cites a system where accountability doesn't necessarily come at the cost of privacy. The author says increased accountability wouldn't unilaterally increase the power of governments like China, since it would make it harder for them to hide their censhorship activities. However, I don't know how strong an argument that is; I'm pretty sure Chinese citizens know China is censoring them, and in fact they participate in self-censorship to avoid being arrested. "Security, Cybercrime, and Scale" describes the economics of cybercrime. "[an] attacker must then do three things: decide who and what to attack, successfully attack, or get access to a resource, and monetize that access (page 67)." Understanding the economics of cybercrime can help defenders decide where to allocate their limited resources. Parameters which affect the economic viability of attacks include scalablility, the ability to distinguish betwen viable and non-viable targets, the density of viable targets, and the challenge of monetizing the attack.

[2] Anne Adams and Martina Angela Sasse. Users Are Not The Enemy. *Commun. ACM ()*, 42(12):40–46, 1999.

Users are often trying to be secure, but have an uninformed notion of security. Policies that frustrate users can lead them to disregard the importance of security. It is important for IT to be in communication with users. It is useful to provide feedback on the security of passwords during password creation.

[3] Lalit Agarwal, Nisheeth Shrivastava, Sharad Jaiswal, and Saurabh Panjwani. Do not embarrass: re-examining user concerns for online tracking and advertising. *SOUPS*, page 8, 2013.

People are more concerned about being shown embarrassing ads than they are about being tracked by advertisers. People don't hate ads; they would prefer to be shown relevant ads than to have ads removed from all websites. Instead of a do-not-track header, maybe a header that gives user preferences? There needs to be a way to do this anonymously.

[4] Farzaneh Asgharpour, Debin Liu, and L Jean Camp. Mental Models of Computer Security Risks. *WEIS 2007*, 2007.

> Focus is on improving the security practices of average users. Explaining concepts using the most natural mental models for average users increases learning and retention. "For example, strong-experts mark passwords as corresponding to a criminal model, while both weak and strong non-experts conceive of passwords as belonging to the physical realm. Therefore, non-experts perceive password risk as closer to the risk of a lost key; while experts perceive passwords as more closely corresponding top subverted credit card numbers."

[5] Dirk Balfanz, Glenn Durfee, Rebecca E Grinter, and Diana K Smetters. In Search of Usable Security: Five Lessons from the Field. *IEEE Security & Privacy*, 2(5):19–24, 2004.

> Takeaways from the example of making connection to WiFi networks easier. The steps were so challenging to follow that it took more than 40 steps. Built a configuration station that set up laptops on the WiFi network. Lessons: 1. You can't retrofit usable security. Think about usability throughout software development. 2. Tools aren't solutions. 3. Mind the upper layers. Users should be able to stay focused on their primary goal. 4. Keep your customers satisfied. Security systems should be tested by users. 5. Think locally, act locally. Users aren't good at thinking about abstract concepts, like certificates.

[6] D Besnard and B Arief. Computer security impaired by legitimate users. *Computers & Security*, 23(3):253–264, 2004.

> Studies the ways users contribute to security compromises. Considers security flaws as a gap between attacker and user effort. People make cost-benefit decisions everything, including security. When people don't fully understand the consequences of their actions in regards to security, they will make poor decisions. Helpful to consider security compromises as

a result of both ”weak protections and malicious intentions” instead of just the fault of the hacker. Perhaps making things simpler, like high pressure boilers were simplified, would help evolve more secure practices and software. Software should be designed for users, not just against attackers. It may also be helpful to investigate why some attacks fail.

[7] J Bonneau, C Herley, P C van Oorschot, and F Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567, 2012.

Describes the pros and cons of many different technologies intended to replace passwords. Concludes that the reasons passwords are still widely used is that proposed replacements are inadequate in a variety of ways. Idea: considering different authentication requirements for different activities. EX, less convenient but more secure authentication required before wiping a device. URRSA would be interesting, used in conjunction with something like an RSAkey (which could be replaced by a smartphone or watch). My conclusion: if you’re logging into a compromised computer, the only security you can have is if you have read-only access to your account. How to protect against someone stealing your cookie, if you log in on a compromised computer (even if you used a disposable password)? Combining two useable and deployable systems can increase security. Combining hardware tokens with federated schemes.

[8] Cristian Bravo-Lillo, Saranga Komanduri, Lorrie Faith Cranor, Robert W Reeder, Manya Sleeper, Julie S Downs, and Stuart E Schechter. Your attention please: designing security-decision UIs to make genuine risks harder to ignore. *SOUPS*, page 6, 2013.

Describes different ways to design dialogue boxes in order to help users make informed security decisions. For example, whether or not to run a new piece of software. Proposes “inhibitive attractors”: dialogue boxes that draw user’s attention to important information, or slow them down so they will pay more attention to that information. Effective in preventing malicious software installs and permission grantings.

[9] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive Password-Strength Meters from Markov Models. *NDSS 2012*, 2012.

> Works by determining the probability of a password based on the n-grams (sequences of n characters) composing it. A potential problem with this technique: people may use patterns, like palindromes, which wouldn't be identified by n-grams. Markov model seems to be like a Finite Automata, unable to handle palindromes. What about a model that can find palindromes, and similar patterns? Could use SFE to secure n-gram database. Are there ways to make SFE less secure, but greatly increase performance? The paper suggests the importane of seeding the ngram database. I could generate an n-gram dictionary from the Bible and other famous books. What if password distribution changes over time? Ex, FB user age distribution changing; the average age of users have increased over time. Distributed n-gram database? Possibly worth creating a high-performance implementation. Could be relatively straightforward to create an open-source implementation. Being able to record the strength of passwords makes it possible for a production website to do A/B testing on the effects different password prompts have on password strength! Could do this at CBD. With a system like this, it becomes possible to describe the percentile of a user's password strength. I doubt studies have been done on the effect that receiving such feedback would have on the strength of passwords designed by users: possibly an area for me to publish research.

[10] Lorrie Faith Cranor and Norbou Buchler. Better Together: Usability and Security Go Hand in Hand. *IEEE Security & Privacy*, 12(6):89–93, 2014.

> Avoid unnecessary warnings (ex, phishing warnings) by checking security in multiple ways; only issue a warning if a number of problems are identified. Also, better explaining risks and further action are important. It is important to consider what

decisions your software requires users to make. A really helpful system would make security and privacy decisions based on users' intentions. ”They also found that password meters can impact user behavior, but most password meters found in the wild have minimal impact because they praise users too soon.” **Valuable info**, But I can't figure out exactly which paper says this. **See ”Password Strength” tag**

[11] Lorrie Faith Cranor and Simson L Garfinkel. Guest Editors' Introduction: Secure or Usable? *IEEE Security & Privacy*, 2(5):16–18, 2004.

Emphasizes that security and usabiity are not opposites; they are both necessary and can both be achieved simultaneously. One good HCI principle that the paper mentions is that users should be able to see and undo actions they take. Several approaches to HCI-Sec are: 1. Make systems that are secure by default. A drawback is that without understanding the internal functioning of a system, users might undermine its security. 2. Give users metaphors to help them understand security. 3. Effectively teach users how to use security software. This requires considering the time-constraints users face.

[12] Iulia Ion, Niharika Sachdeva, Ponnurangam Kumaraguru, and Srdjan Capkun. Home is safer than the cloud!: privacy concerns for consumer cloud storage. *SOUPS*, page 13, 2011.

Users don't think it's safe to store things in the cloud. But are they aware that in some ways things are safer in the cloud? Ex, automatic backups and prevention of bit rot.

[13] Julie A Jacko and Andrew Sears. *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications.* Danbury: Lawrence Erlbaum Associates, 2002.

HCI emerged when computers began being used by the general populace; programmers could no longer rely on their own experience to design software. Today, most R&D departments have anthropologists and social scientists on staff. Seems like a very helpful book to read before designing a study. Discusses techniques for learning from users.

[14] K Kain, Sean W Smith, and R Asokan. Digital signatures and electronic documents: a cautionary tale. *Communications and Multimedia Security*, (Chapter 22):293–308, 2002.

> Many common document formats are dynamic; signing a document doesn't mean that it will appear the same way when it is viewed later. For example, PDFs can include javascript. An attacker could change the information displayed according to the date on which it is viewed, for example.

[15] B Laurie and I Goldberg. Replacing passwords on the Internet AKA post-Snowden Opportunistic Encryption. 2014.

> We should emphasize the use of self-signed certificates. No more scary warnings. Then, moving toward PAKE (Strong Password-Only Authenticated Key Exchange) connection or Diffie-Hellman to authenticate the encrypted connection. Basically, a scheme that relies on password managers to encrypt all traffic online.

[16] Mainack Mondal, Peter Druschel, Krishna P Gummadi, and Alan Mislove. Beyond Access Control: Managing Online Privacy via Exposure. In *Workshop on Usable Security*, pages 1–6, Reston, VA, February 2015. Internet Society.

> Describes a new system for privacy protection for social media sites. Current privacy protection schemes use access-control, but this is insufficient for users' needs. Access-control is challenging to configure, doesn't account for the difficulty of finding information, and can't control sharing of information. As an alternative to access-control, this paper introduces the concept of "exposure as an alternate model for information privacy; exposure captures the set of people expected to learn an item of information eventually." Part of the system relies on analytics available to Facebook and other big companies to predict the expected number of views items will recieve, and to alert users if items become unexpectedly popular. One limitation of the system described is that apps like Instagram

encourage sharing via re-posting screenshots, so privacy protection on the original photo could be negated, unless further steps were taken.

[17] Frank Nielsen. Logging safely in public spaces using color PINs. *arXiv.org*, April 2013.

Describes a graphical authentication system based on positioning overlapping grids. The system protects against over-the-shoulder snooping and compromised peripherals. It only requires a digitally signed piece of software. Of course, if the keyboard and mouse are compromised, what is the chance you'll be able to trust the software running on the computer, which verifies the digital signature of the software? Can you even necessarily trust the monitor? Not sure how practical this is. Video of a similar system: https://www.youtube.com/watch?v=IDgaH-ilUCw

[18] Emilee J Rader, Rick Wash, and Brandon Brooks. Stories as informal lessons about security. *SOUPS*, page 6, 2012.

People tell and retell stories about computer security. People behave like people who are similar to them. People pass on stories that have a lesson. Action points: helping users tell stories. Creating/shaping stories that help users make the right decisions.

[19] Scott Ruoti, Nathan Kim, Ben Burgon, Timothy W van der Horst, and Kent E Seamons. Confused Johnny: when automatic encryption leads to confusion and mistakes. *SOUPS*, page 5, 2013.

If encryption is too automatic and magical, users have trouble understanding it and remembering to use it. A system where users encrypted passwords using an external program and pasted the cipher text into emails was actually well-received by users.

[20] M A Sasse, S Brostoff, and D Weirich. Transforming the 'Weakest Link' — a Human/Computer Interaction Approach to Usable and Effective Security. *BT technology journal*, 19(3):122–131, 2001.

Making user responsible for the security of their employers by securing their own pay details using their company single-sign-on password. You need to account for the cost of rejecting the legitimate user; if someone is trying to make a sale and can't access their account, they are going to borrow a coworker's credentials. It is important to motivate users. Several ways: 1. Emphasizing importance of security to the business. 2. Have a policy of punishing bad behavior in the case of a breach. 3. Security breaches should be made known to employees.

[21] M Angela Sasse. Computer Security:Anatomy of a Usability Disaster, and a Plan for Recovery. pages 1–4, February 2003.

> If CBD's security team hasn't read this, they should. It's important to consider users' motivations. Users are more likely to follow security protocols if they understand the threat. Being security conscious isn't considered cool. If user1 forgets their password and asks to use user2's, if user2 refuses, he looks like a jerk. p.2: The study by Whitten & Tygar (1999) showed how designing a user interface to a security tool (PGP) with no consideration of users' tasks can render it completely ineffective. – Highlighted jan 26, 2015

[22] Roger R Schell. Information Security: Science, Pseudoscience, and Flying Pigs. *ACSAC*, pages 205–216, 2001.

> Criticizing attempts at security that aren't verifiable. Today, we have stronger cryptography. But we don't have assurance of the systems on which the cryptographic algorithms run. See the last page for a list summarizing security concepts.

[23] R Shay, P G Kelley, S Komanduri, and M L Mazurek. Correct horse battery staple: Exploring the usability of system-assigned passphrases. *Symposium on Usable Privacy and Security (SOUPS)*, 2012.

> It might be worth providing a password suggestion button for the CBD website; elderly customers might benefit from it. The study didn't show passphrases to be particularly helpful. However, it might be worth replicating their work, and testing if that is actually the case. Would be worth measuring what

percent of CBD's users use password managers. Lots of good references talking about Mechanical Turk.

[24] Sean W Smith. Humans in the Loop: Human-Computer Interaction and Security. *IEEE Security & Privacy*, 1(3):75–79, 2003.

Describes the importance of HCI to computer security. Many problems come from users and programmers having inaccurate mental models of software. It is important to consider users (end users, programmers, etc.) using HCI principles during the process of design. Security policies should not be designed in a vacuum. Emperical evidence should be used to determine parameters as simple as the number of password attempts allowed before locking users out. This paper describes the work of other researchers who are working at the intersection of HCI and Computer Security.

[25] Christopher Thompson, Maritza Johnson, Serge Egelman, David Wagner, and Jennifer King. When It's Better to Ask Forgiveness than Get Permission: Attribution Mechanisms for Smartphone Resources. pages 1–16, June 2013.

Smartphones shouldn't ask users to grant permissions for actions that can be undone or are only annoyances. Instead, the OS should show which app was responsible for these kinds of behaviors, so users have the option to uninstall the app. Avoids habituation, where users don't pay attention to security messages. Also, user can rarely remember which apps they have granted privileges to. Users will write negative reviews of misbehaving apps, which will discourage app developers from being irresponsible.

[26] Wenley Tong, Sebastian Gold, Samuel Gichohi, Mihai Roman, and Jonathan Frankle. Why King George III Can Encrypt, June 2014.

A blog post summarizing an undergraduate paper by the same title. The metaphor of King George III sending orders to his troops with locked and sealed strongboxes is used to explain public-key encryption and signatures. The metaphor allowed the concepts to be conveyed more succinctly and with the same effectiveness as existing documentation for PGP.

[27] A Whitten and J D Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. *Usenix Security*, 1999.

p.2: In this paper, we offer a specific definition of usability for security, and identify several significant properties of security as a problem domain for user interface design. – Highlighted jan 26, 2015 Uses a ”cognitive walkthrough analysis” and a ”laboratory test user.” Most users couldn't use the software, even given 90 minutes of time to figure it out. “Definition: Security software is usable if the people who are expected to use it:1. are reliably made aware of the security tasks they need to perform;2. are able to figure out how to successfully perform those tasks;3. don't make dangerous errors; and4. are sufficiently comfortable with theinterface to continue using it.” You need to give the user a cognitive model of the security. The key aspect is learnability.

[28] Ka-Ping Yee. User Interaction Design for Secure Systems. *ICICS*, 2513(Chapter 24):278–290, 2002.

Describes design principles that aid the development of secure systems. “We believe that usability and security goals rarely need to be at odds with each other. In fact, often it is rather the opposite: a system that's more secure is more predictable, more reliable, and hence more usable.” The user's actor-state should be up-to-date (users should have an accurate understanding of the states and activities of software actors). Users might not realizing what programs are running in the background on their computers is an example of inaccurate actor-state. Secure desktop environment for 10/10 principles: http://www.cs.berkeley.edu/˜pingster/ For 8/10 principles: Cap Desk http://www.combex.com/tech/edesk.html

[29] William Yurcik. NVisionIP and VisFlowConnect-IP: Two Tools for Visualizing NetFlows for Security. pages 1–24, February 2006.

Describes a system for visualizing network activity. Could be useful for detecting malware activity. It's hard to gauge how useful this would be. It seems like attackers could just adapt to hide from this. Then the software would just be giving a false sense of security. There needs to be an evaluation of how useful this is in practice (ex, false positives vs false negatives). This might be an example of the "pseudoscience" critiqued in "Science, Pseudoscience, and Flying Pigs" by Schell.