Beyond usability: Security Interactions as Risk Perceptions

Professor L Jean Camp
School of Informatics and Computing
Indiana University, Bloomington, IN

Abstract

Translucent security argues for the integration of the human element in the design of secure systems and secure components in a systematic manner. The following work details the theoretical arguments for translucent security and enumerates the principles behind the approach. After briefly listing the principles, the browser experience is critiqued for lack of translucence. Straight-forward changes in design are suggested, that range from the trivially easy to engaging challenges. [1]

Position Paper

To understand what translucence is, it is useful to understand what translucence is not. Translucent security is explicitly not usable security; as it applies to the entire system design not the interaction. It has been well argued by others that usable security is not usability for three reasons. First, individuals rarely want to perform security. Security is not the desired goal of the individual. In fact, security is usually orthogonal and often in opposition to the actual goal. Second, security information is about risk and threats. Such communication is most often unwelcome. Increasing unwelcome interaction is not a goal of usable design. Third, since individuals must trust their machines to implement their desired tasks, risk communication itself may undermine the value of the networked interaction. For the individual discrete technical problems are all understood under the rubric of online security (e.g., privacy, malware).

By coordinating the user communication and security settings of the system interaction, translucent security both enables responsive secure computer-mediated interaction and also mitigates the risks of casual connections, allowing individuals to distinguish the two and protect themselves appropriately.

Yet security is not only risk communication, as the computer can actively mitigate risk. Certainly airbags, child proof medical caps and other physical design changes mitigate risk. Risk-mitigating interactions on the computer can be conceptualized as control panels as well as airbags; as providing feedback as well as minimizing risks. Because much of security is inherently subject to user action (e.g., information sharing via document attachment or password disclosure) security can never be entirely mitigated by default. Yet because security risks are

[1] Please reference as L. Jean Camp, "Translucent Security", Worksho on Risk Perception in IT Security and Privacy (RP-IT) Workshop, Newcastle upon Tyne, 26 July 2013.

often at identifiable choice points (e.g., download or click) it is possible to mitigate only at moments of risk.

Neither truly opaque nor transparent design is effective for the creation of mechanisms to enable non-technical users to secure their digital assets. Usable security may argue for visibility as a principle of design. However, in the case of security, the user-action-system-consequence may be overwhelming, create anxiety, or frankly incomprehensible. Further, as actions that fail to implement security create risk, and risk is probabilistic by definition, there may in fact be no consequence at all.

Alternatively, security as a default is opaque. Default security limits user autonomy and must presume context. As a result it often appears arbitrary to the computer operator, who then seeks to subvert the security mechanism. Usability experts have been critiqued for demanding that security conform to the technically impossible. Simultaneously even security engineers recognize that the human requirements for secure systems are not humanly possible (e.g., make up a random password you will neither guess nor recall, don't write it down, and don't forget it).

Translucent security is differentiated from usability in two fundamental dimensions. First, translucent security is focused on the threat assessment. Threat and risk assessments are not consider components of the usability domain, as the expertise of the interaction designer is not on the threat. Second, translucent security assumes that the individual does not want to use security technologies at all. Components of the risk communication approach are closer to persuasive design [4] or safety engineering [1] than usable design, more [2] than design.

Five Guidelines

To summarize, translucent security has the following design goals. **First**, implement high security defaults and then automatically decrease them as feasible.

Second, make it possible to override these in a simple automated manner, with a single click. However, require the individual to experience either highly personalized or demographically specified (based on information available to the system) risk communication. Thus individuals can take risks but they do so knowingly.

Third, personalize security for the context. Some situations require temporary disarmament by the individual. One example is connecting to a commercial wireless service in an airport. Scripts must be enabled, and active advertisements accepted. Third party cookies are required. In this case the individual is required to accept high levels of risk in the interests of the party controlling the connectivity. Recall these contexts. Isolate as much of the machine as possible, and clean the machine on context changes. In contrast, some situations require the highest levels of security. Some of those can be recognized by the client machine. Examples include entering authentication credentials previously used in a financial context into a non-banking site, installing downloaded software, or entering critical information in a recently created site, i.e. Social Security Numbers or bank account numbers. When these contexts are in conflict, for example, banking online using airport wireless, isolated the functionality that was required to establish connectivity and implement a more secure sandbox for the sensitive transaction.

Fourth, personalize security for the individual. A translucent design utilizes history and automated intelligence to provide contextual security setting and minimize individual risk. Intelligent secure interactions observe individual security behaviors by enabling individuals to disable even when security is recommended, but also implements the most stringent security settings the individual will tolerate in a given context.

Fifth, use social context not only in the interest of advertisers but also in the interest of the individual. This implies observing histories of interactions (locally) as well as histories of simple clicks.

Translucent security models users as individuals making complex risk decisions with a limited cognitive budget and following the set of heuristics documented in other risk domains. Instead of a plethora of add-ins, add-ons, and an ever expanding vocabulary of attack and defense, each individual is provided with a single narrative with a consistent metaphor about a given digital context, and a path to risk mitigation when the user chooses the risky path. The path to risk mitigation should be, as with all layers of translucent security, automated when and to the extent possible.

Thus translucence translates into default security, automated system isolation, intelligent interactions, clear communication of risk, clear path to mitigation, while enabling the choice to knowingly make risk-seeking actions. Without the last option, the security becomes undesirable and will be disabled. interfaces are in the spirit of direct manipulation [3]. However, security and risk interaction are often not only not subject to physical manipulation versus command lines, but in fact give the communication recipient with no options at all.

Closing

This paper is a position paper arguing for a more systematic approach to designing secure systems, one that includes the expectations of human in the design loop. First, all that can be automated should be automated. Therefore the individual is engaged as rarely as possible. Second, when there is the inevitable uncertainty of context or risk, risk communication is provided that is appropriate based on the behavior and the history of the individual's interaction with the device being secured. Third, interruptions for risk communication are appropriately timed for the workflow or task at hand. Fourth, with the risk communication there exists clear paths to mitigation if the individual chooses to take the risk after having been warned. Notice that the first principle also applies here, and that mitigation appropriate to the risk at hand should be automated to the extent possible.

References

[1] Levenson. Software engineering: A look back and a path to the future (html), February 1997.

[2] M. G. Morgan, B. Fischhoff, and A. Bostrom. *Risk Communication : A Mental Models Approach.* Cambridge University Press, 1997.

[3] B. Shneiderman. Direct manipulation: A step beyond programming languages. In *Human-computer interaction*, pages 461–467. Morgan Kaufmann Publishers Inc., 1987.

[4] D. Weirich and M. A. Sasse. Pretty good persuasion: a first step towards effective password security in the real world. *Proc. of the 2001 workshop on New security paradigms*, pages 137–143, 2001.