Ensuring Security in Human Computer Interface

M A Aziz Jahan
Electrical and Computer Engineering
Michigan Technological University
jahan@mtu.edu

Abstract - The fast spread of the internet and technologies has resulted in a slew of security challenges in the workplace and in the community. Although the increasing use rate indicates that there is a demand and need for developing applications, a secure online infrastructure is also necessary. It has been demonstrated that users do not take privacy issues seriously. HCI studies and theories aid developers in providing a secure environment for users, as HCI is concerned with user trust with their confidentiality as well as improving system quality. This paper aims to identify users' major concerns HCI-Sec in apps, as well as the usability of security systems in networks, in order to protect their personal information.

Key Words: Human Computer Interface, Security, Data Breach, Privacy, Information Security.

1. *Introduction*

Users interact with computers and technology through a variety of user interfaces, including mobile phone menus, computer buttons, icons, and windows, car dials and knobs, and Internet return buttons and hyperlinks.

These user interfaces are intended to improve people' comprehension of technology and productivity when using it. A well-designed interface, for example, can help a user become adept in the functioning of a software application in a shorter amount of time. This allows the user to improve his or her efficiency when finishing a task. The user feels in command of the technology and satisfied with it. A poorly designed interface, on the other hand, might annoy users and make it difficult to complete tasks, leading to aversion and skepticism toward utilizing the technology.

This research focused on features of human computer interfaces (HCIs) that are important in the context of information security. A software product's interface, such as an encryption program or a firewall, is an example of this. Almost all of the functions performed by these applications are related to security. Parts of other interfaces, such as the login screen of an Internet banking website, are similarly interlaced with security mechanisms.

As the globe gets more linked and more commerce is performed electronically, computer and information security is becoming increasingly important. Anti-virus software (used by 99 percent of organizations asked) and firewalls are the most common security technology utilized by businesses, as per the Computer Crime and Security Survey [1] (98 percent of companies). Anti-virus software and firewalls have now transitioned into the realm of the common user who is not a security specialist as a result of the development of workplace and home computers. This means that with technologies like anti-virus software and firewalls that

communicate and make appropriate recommendations through security features, interfaces play a critical role. The interface provides security functions to the user. The user is informed about the security protocols that are offered and how to use them through the interface. A user may be unaware of a security feature or may be misusing it. A personal firewall, for example, can only protect a user's computer if it is turned on, and it will only be turned on if the person understands how to do it. The user interface must assure that the user is directed in order to reduce the risk of the user being the 'weakest' link.

There are a number of well-established parameters that can be used to improve the efficiency of employing different technologies when developing an interface. Consistency and standards, as described by Jakob Nielsen and Rolf Molich in 1990 [2], are an example of such a criterion. Consistency and standards imply that the words and actions used in an interface must be consistent and have the same interpretation throughout. Consider some of the firewall products now available on the market. Many of these products utilize the phrases 'firewall' and 'gateway' interchangeably, causing misunderstanding among end users.

The goal of this work is to demonstrate how existing and well-established HCI criteria may be used to analyze and improve an interface's security features. A number of recommendations are made for modifying existing available interfaces with the ultimate goal of increasing the use of these products' security features.

2. ***What is HCI?***

Human-computer interaction (HCI) [3] is a term that refers to how people interact with computers. HCI is concerned with the interaction in between or more individuals and one or more computers from the standpoint of computer science. The image of a person utilizing a user interface application, such as Microsoft Windows on a workstation [4], comes to mind.

HCI can be described as "the part of a computer program responsible for creating the common ground with a specific (i.e. well-known) user," according to Sjoerd Michels [3]. His mission is fulfilled by expanding and preserving this common ground throughout the application's engagement process. Direct manipulation of familiar items should be the primary interaction principle wherever possible."

The goal of HCI is to improve a system's 'userfriendliness.' This is sometimes misinterpreted as being incompatible with the goals of a safe system [5]. In a secure system, for example, information secrecy is needed and is achieved to some extent through the use of credentials. According to conventional wisdom, the more passwords and the more complicated the passwords, the better a system's security. Users, on the other hand, are unlikely to remember a long, complex password, which indicates they will write that down, thereby jeopardizing the system's security. The fewer and simpler the passwords are, the better when it comes to usability standards. This seems to point to a tension between usability and security. When creating a secure, useable password, a balance must be established.

Perception — gaining knowledge through the senses, is one of the most important human resources used in HCI. Cognition is the process of understanding how information and physiology are processed. Font classification and size, color contrast, and other factors influence perception. The user's reasoning resource and communications should be clear, with powerful reaction options. [6]

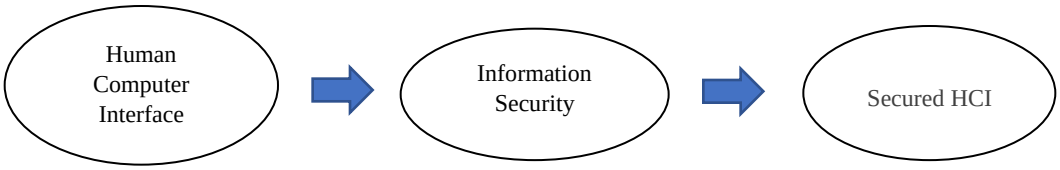

Figure 01: Security in HCI.

3. ***Criteria for a Robust HCI***

Different scientists and researchers established different criterias for a robust HCI system but the their main focus are same. An American computer scientist named Ben Shneiderman compiled certain unspoken realities about design and came up with the following eight broad guidelines [7]:

4. Strive for Consistency.
5. Cater to Universal Usability.
6. Offer Informative feedback.
7. Design Dialogs to yield closure.
8. Prevent Errors.
9. Permit easy reversal of actions.
10. Support internal locus of control.
11. Reduce short term memory load.

Both regular designers and interface designers will benefit from these rules. It is feasible to distinguish a great interface design from a lousy one using these eight rules. These are helpful in the experimental evaluation of better GUIs.

Donald Norman developed seven criteria in 1988 to evaluate the interaction between humans and computers. He created a seven-stage process for transforming challenging jobs. Norman's seven principles are listed below:

- Use both knowledge in world & knowledge in the head.
- Simplify task structures.
- Make things visible.
- Get the mapping right (User mental model = Conceptual model = Designed model).
- Convert constrains into advantages (Physical constraints, Cultural constraints, Technological constraints).
- Design for Error.
- When all else fails − Standardize.

Evaluating these criteria’s, it can be summarized and described as follows:

- **Visibility of System**:
  The system must be able to keep users up to date on its status by providing suitable feedback in a timely manner like the recent changes of data, connectivity of the internet connection etc.

- **User Control and Freedom**:
  Users should have the ability to manage the system processes or operations, including the ability to reverse and retake them if desired. A confirmation message should always be there before committing the task.

- **Sync between system and the real world**:
  There should be a synchronous relationship between the system and the app by making the user interface generalized for every type of users while navigating. Using too much technical terms can make the system boring of ineffective to some users.

- **Error Prevention**:
  Application should be interactive to prevent error happening at first place by providing hints or generalized information to users about the task they are attempting to.
- **Consistency and Standards**:
  The system should be transparent. It should not employ the same or close terminology for separate operations, as this could lead to user misunderstanding.

- **Recognition rather than recall:**
  For subsequent searches, the location or part of the inputs may be the same. Previous inquiries can be retained so that users can choose from a drop-down menu rather than having to memorize details.

- **Flexibility and Efficiency of use:**
  The system should be flexible enough to allow users of varying levels of skill to utilize it effectively; experts should be capable of speeding things up by eliminating excess features, while novices should be able to traverse the system easily.

- **Aesthetic and minimalist design:**
  As the App is being used for a critical inspection, every important section should be highlighted in a simplistic manner. Color choosing is very important while selecting for each section. Irrelevant and unnecessary information should be avoided.

- **Help and Documentation:**
  The app should include a help section that explains how to utilize it. The statement should be simple to read and free of complexity and technical terminology. If any technical phrase is used, it should be properly explained.

4. Security – Usability Analysis to make a Secure HCI System

It can be employed the concept of usage scenarios (or just scenarios) and threat (negative) scenarios to analyze the security and usability of a system based on the threat model [8]. In this context, we define use scenarios as behaviors that stakeholders of a secure system would like to have happen, and threat scenarios as actions that the system should not allow to happen. On the one hand, HCISec is focused with making usage situations accessible to users with minimal mental and physical effort. Composing an email, discovering a contact, sending email, or adding a new contact are examples of usage scenarios in an email program.

HCISec, on the other hand, is dealing with threat scenarios, or unwanted activities, that could cause non-malicious users to compromise a system's security. Non-malicious users who may breach the system due to issues outlined in the security-usability threat model are the focus. Rather of being connected with malevolent attackers, researchers associate danger situations with genuine users who have a non-malicious aim. In a secure email program, for example, we're concerned not only about whether individuals can encrypt emails, but also about whether they might mistakenly encrypt a message with a key belonging to an unwanted recipient.

While usage and threat scenarios are commonly utilized in requirements gathering and design [9], it can be used in security-usability assessments throughout the system development life cycle and after product release. The stages in the security-usability analysis process are summarized in Figure 2. [10]

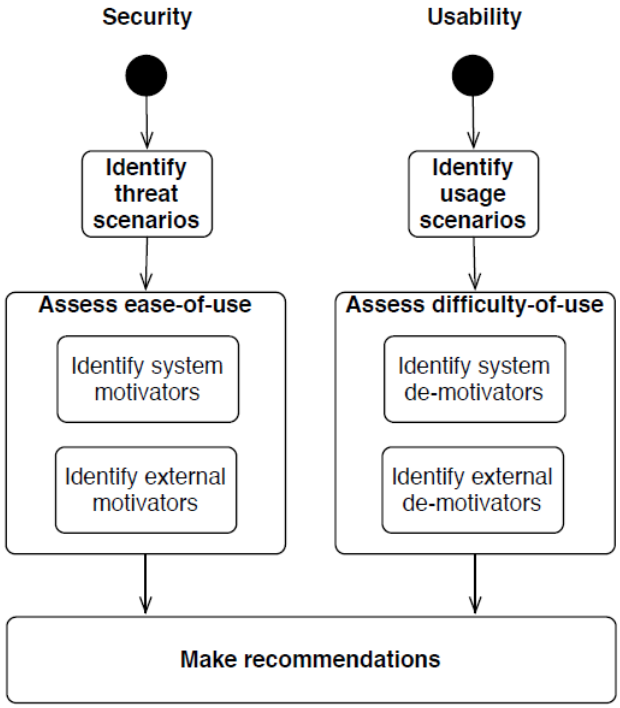

Figure 02: Security Usability Analysis Process for HCI.

- ***Identify Usage Scenarios***

Before a usability evaluation, usage scenarios are needs to be identified in HCI. The scenarios represent specific tasks that a conventional system user would attempt to complete. Participants are shown usage scenarios (for example, in usability testing) or experts evaluate them, and performance metrics are recorded. An assessment of scenarios offers performance statistics on all the threat model's usability elements. If evaluated situations meet pre-determined criteria, they are considered usable.

- ***Identify Threat Scenarios***

As previously stated, HCISec is indeed concerned with legitimate users making errors that compromise a system's security. It can be proposed [10] that occurrences (threat scenarios) that could lead to such behavior be studied and evaluated. The idea is to see how easy it is for genuine people to unintentionally damage a system. A threat scenario in device coupling, for example, may be users not paying close attention when comparing strings, resulting in the strings being reported as matching when they are not.

- ***Make Recommendations***

Making proposals based on the previous processes is the final stage. Regions that need to be improved to make usage scenarios more accessible to legitimate users, as well as areas that need to be hardened for threat scenarios, will be the focus of recommendations.

The analysis method considers external issues in addition to users and the system. In theory, a system might become usable or secure, but in practice, external influences may trump internal ones. In difficult conditions such as being late for work and needing to deliver an urgent report, an individual who is conscious of strong passwords and the need to avoid revealing passwords may be obliged to share it with a colleague.

It is impossible to expect all safe systems to attain optimal usability and security. There will be a trade-off between usability and security in most systems. The goal is to reduce the likelihood of threat scenarios as much as possible while increasing the accessibility of application scenarios. Allowing users to write down passwords, for example, may be appropriate if the threat of dictionary-based cryptanalysis tools is extremely serious.

It's likewise impossible to assume a total removal of all local and global motivators, or demotivators, under threat or usage contexts. In any case, we want to keep these components to a bare minimum. An appropriate level depends on the facts to situation and must be determined according to the system and its surroundings. Both de-motivators and motivators can be addressed using a risk-level matrix [11] to decide which usage and threat scenarios should be addressed first. Each of them can be ranked according to its impact and likelihood on the ecosystem using the matrix.

5. Challenges in HCI

For starters, many existing security systems are difficult to map onto a task-centered strategy. As a result, we've been working on new security "building blocks" that can be utilized to create more secure applications (for example). Second, and more importantly in this context, we've recognized a few HCI design issues that this technique presents.

The creation of an acceptable threat model is a crucial stage in the development of a safe system. Before designing a standard threat model, however, we believe that system design should start with a user model – an understanding of what operations the user is seeking to accomplish and what concepts the user must work with. Even if the system overall becomes less protected as a result, the new system must be capable of completing the user's desired duties.

It's difficult for users to recognize and explain these expectations as security threats rather than system expectations, thus identifying them might be tough. To know how to take user-centered discoveries and integrate them into the threat model, a conversation between HCI and security experts is required. [12]

Making implicit security actual — inferring what necessary changes in security state are suggested by the user's activities — is an even bigger problem in infusing effective HCI design into security-related components of applications. Even more difficult is to design the user environment so that it is simple to deduce from user behaviors what tasks, and thus what security activities, they are intending to complete at any given time.

6. Conclusion

The interface of a system is crucial and should not be overlooked, especially in a security setting. By using the HCI-S criteria, a compromise between the seemingly disparate objectives of HCI and security can be established. This will result in a system that is both easier to use and more secure.

Security interface usability is merely one aspect of a larger picture. Even the most user-friendly interface may be disregarded by users unless policies requiring the use of security tools are in place. A corporation should, for example, establish a policy of always encrypting critical correspondence.

For conducting security-usability evaluations, a security-usability threat model has been suggested. While recommending, comprehend utilization scenarios and hazard scenarios were taken into consideration. Also, it needs to identify both internal and external dangers to a system's usability, security, or both. Threat scenarios are used to discover regions that may assist non-malicious users compromise the system's security, whereas user circumstances are used to identify locations that may hamper the system's usability. Users are more likely to conduct the former when the danger scenarios are more practical than the usage scenarios. External factors may also force users to do activities they would not typically take.

This is the initial effort in building a security-threat model for HCISec security-usability analysis. Future work will involve adding detailed metrics that can be used to calculate the likelihood of users performing a threat scenario over a usage scenario. Further work is also necessary to extend the threat model to malicious users.

References:

[1] Richardson, R., 2003. 2003 CSI/FBI Computer Crime and Security Survey, Computer Security Institute, www.gocsi.com

[2] Nielsen, Jakob, and Rolf Molich. "Heuristic evaluation of user interfaces." In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 249-256. 1990.

[3] Michels, Sjoerd. "Co-writing, look and feel." *Master's thesis, Tilburg University* (1995).

[4] Hewett, Thomas T., Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.

[5] Botha, Reinhardt A., and Tshepo G. Gaadingwe. "Reflecting on 20 SEC conferences." *Computers & Security* 25, no. 4 (2006): 247-256.

[6] Patrick, Andrew S., A. Chris Long, and Scott Flinn. "HCI and security systems." In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pp. 1056-1057. 2003.

[7] https://www.tutorialspoint.com/human_computer_interface/guidelines_in_hci.htm

[8] Rosson, Mary Beth, John M. Carroll, Con Rodi, Ian Alexander, and Neil Maiden. "Teaching computer scientists to make use." *Putting scenarios into practice: The state of the art in scenarios and use cases* (2004): 445-463.

[9] Kazman, Rick, Gregory Abowd, Len Bass, and Paul Clements. "Scenario-based analysis of software architecture." *IEEE software* 13, no. 6 (1996): 47-55.

[10] Kainda, Ronald, Ivan Flechais, and A. W. Roscoe. "Security and usability: Analysis and evaluation." In *2010 international conference on availability, reliability and security*, pp. 275-282. IEEE, 2010.

[11] G. Stoneburner, A. Goguen, A. Feringa, N. I. of Standards, and T. (U.S.), *Risk Management Guide for Information Technology Systems [Electronic Resource] : Recommendations of the National Institute of Standards and Technology / Gary Stoneburner, Alice Goguen, and Alexis Feringa* . U.S. Dept. of Commerce, National Institute of Standards and Technology, Gaithersburg, Md. :, 2002.

[12] Grinter, Rebecca E., and D. K. Smetters. "Three challenges for embedding security into applications." In *Proceedings of CHI 2003 Workshop on HCI and Security Systems*, pp. 129-136. 2003.