

Principles for evaluating the usability of computer security & Security through usability: Similarities and Differences

Ometere Louisa Ehinlaiye & Melissa Karolewski
Computer Science Department
Indiana University of Pennsylvania

Both “Usability of Security”¹ by Whitten & Tygar and “Security through Usability”² by Ross, contain many similarities and differences regarding two security principles. According to the not widely used Kerchoffs sixth law, “regarding the circumstances in which such system is applied; it must be easy to use and must neither require stress of mind nor the knowledge of a long series of rules.” In “Usability of Security” and “Security through Usability”, the highest principle *is that both the interface and the software or program being used must be easy to use and understand, and at the same time extremely successful in protecting and securing the data.* The major problem, clearly stated by Whitten & Tygar in their case study is that , “design techniques that create good user interfaces for other types of software are ill-fitted to creating user interfaces that enable effective security.”¹ There is always a compromise in either aspect. Security is compromised and the user interface is easy to use, or the user interface is complicated whilst the security is top notch. Yet, both models have great points regarding security and usability.

The first model is based on the article “Usability of Security: A Case Study”¹ by Alma Whitten and J.D. Tygar. In this article, Whitten and Tygar develop general principles for evaluating the security of computers. This model notes that many security systems are created that are extremely complicated for a normal user to understand and operate. When designing these types of programs or systems, people often ignore the security aspect, the HCI aspect, or both. Whitten and Tygar also note some important properties of the usability problem for security, such as the barn door property, the weakest link property, abstraction problem, and the lack of feedback.

The second model, based on security through usability in the article, “Security through Usability” by Seth Ross² believes that security should be based on the strength of keys rather than approaching cryptography with a secretive approach. “Security by obscurity” has many weaknesses, as outlined in “Kerchoffs’ Principle.” These principles state that the system must be substantially undecipherable, must not require secrecy, and should be easily stolen; something which a security professional would not agree with at first glance. The system must be easy to communicate and at the same time remember keys used, should be portable, and also be compatible with telegraph communication, and must require no more than one person. Lastly, the system must be easy to use, without stress and much knowledge of the system. Because numerous security professionals do not support Kerckhoff’s ideas, today’s systems do exactly the opposite of what Kerckhoff was trying to avoid, thus the term “security through complexity” .

Another similarity is (from Kerckhoff’s second law) that with both “Usability of Security” and “Security through Usability” the system must be substantially undecipherable. A problem however is that with usability of security, *the system is only as strong as its weakest link, and especially when it comes to encryption, if the data is not kept secure even for a second, you don’t know who would have already had access to that information.* It is simpler to find the difference between the principles, than to find the similarities due to the concern for the user. Either the user or security is enviably going to have top priority. It seems now that one must be compromised for the two to coexist. Another issue Whitten & Tygar call the “unmotivated user property” is important when analyzing “Usability of Security” and “Security through Usability.” This condition exists when the user is only concerned about going about doing their business whether its is browsing the web, emailing or downloading information. It is expected by these users that the actions they are

performing are secure, when in actuality they may not be secure. Even if using an encryption problem, without a well-developed user interface, security cannot be assumed.

Complexity of the encryption programs must also be noted. By reviewing the case study of PGP 5.0 by Whitten & Tygar many security and usability faults can be found. In Whitten and Tygar's testing, the participants thought they had carefully encrypted their information and sent it off securely meanwhile their information was not encrypted. There were also circumstances where although the GUI seemed really user friendly, most of the icons used did not represent what the actually task was. For example, the icon in the PGP 5.0 GUI used for *signatures* is a blue quill pen. When a novice may see this, they may automatically associate it with a regular signatures from everyday life, meanwhile it's the complete opposite. In encryption, signing is the method where they use their already generated private keys to generate signatures that could easily be recognized by the other party they are sending information to. Although here the GUI is very easy, it is also very ineffective and confusing because their designs are built on the premonition that most of the users would at least know what a signature is and how signatures are generated which is not necessarily always true.

The main purpose in Whitten & Tygar's, case study was to discover if an average user would be able to use PGP and if the user interface of PGP is not in violation of Kerckhoff's sixth law. This meant that the user would be able to understand that encryption is one of the safest ways to ensure that your data is safe and by using PGP understand the basic concepts of encryption and its use.

From Whitten & Tygar, its can be concluded that PGP is one of the safest methods of sending information and that the GUI interface was not user-friendly. Because of the lack of ease of use, users became frustrated and some even gave up when they could not get it to

work. This thought goes back to the initial conclusion that security is given top priority while GUI is given a lower priority because currently security is more important than user-friendliness. In the opinion of the user, software should be made to protect and serve, yet most software protects extremely well, but is not easily used.

Overall, both these models have very many good points as well as similarities and differences. Most importantly, two ideas that can be gathered from both models is that a security software must be reliable and easy to use, so that the user continues to use and buy the software. A user must also be able to easily use the software for detailed applications, if desired. By combining the two models together, a more concise model could be created and implemented.

References

- 1 Whitten, A, Tygar, J. Usability of Security; Carnegie Mellon University Computer Science Technical Report CMU-CS-98-155, Dec 1998.
- 2 Ross S, Security through Usability; www.securius.com.