Encyclopedia of Information Science and Technology, Fourth Edition

Mehdi Khosrow-Pour
Information Resources Management Association, USA

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Khosrow-Pour, Mehdi, 1951- editor.
Title: Encyclopedia of information science and technology / Mehdi Khosrow-Pour, editor.
Description: Fourth edition. | Hershey, PA : Information Science Reference, [2018] | Includes bibliographical references and index.
Identifiers: LCCN 2017000834| ISBN 9781522522553 (set : hardcover) | ISBN 9781522522560 (ebook)
Subjects: LCSH: Information science--Encyclopedias. | Information technology--Encyclopedias.
Classification: LCC Z1006 .E566 2018 | DDC 020.3--dc23 LC record available at https://lccn.loc.gov/2017000834

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Usable Security

Andrea Atzeni
Politecnico di Torino, Italy

Shamal Faily
Bournemouth University, UK

Ruggero Galloni
Square Reply S.r.l., Italy

INTRODUCTION

Recent decades have been characterized by the growth of information technologies in the private and public sectors. The positive impact that ICT has on job performance, as well as the expansion and creation of business opportunities for companies, count as the main drivers for this growth. This growth led to the proliferation of distributed applications and physical devices, and the diffusion of technologies that facilitate social participation and social interaction. All these applications, devices and interactions may contain important information, or give access to sensitive data, putting them at risk.

The rapid diffusion of technology has led to the reduction of active security monitoring, as well as the lack of technically competent people in control of applications and devices. Moreover, the increment in social interaction increases the damage other people can directly or indirectly cause.

Traditionally, security is only considered as strong as its weakest link, and people were considered as the weak links (Schneier, 2003). This thinking triggers a vicious circle. (Adam & Sasse, 1999) stated that users are informed as little as possible on security mechanisms took by IT departments, precisely because they are seen as inherently untrustworthy. Their work has shown that users were not sufficiently aware of security issues and tend to build their own (often inaccurate) models of possible security threats. Users have a low perception of threats because they lack the necessary information to understand their importance. According to (Sasse & al., 2001) blaming users for a security breach is like blaming human error rather than bad design. Security has, therefore, a human dimension that must be neither ignored nor neglected. The increase in the number of breaches may be attributed to designers who fail to sufficiently consider the human factor in their design techniques. Thus, to undo the Gordian knot of security, we must provide a human dimension to security.

BACKGROUND

Human-Computer Interaction (HCI) is a field concerned with the interaction between people and technology, and how this supports humans in completing tasks to achieve one of more specific goals. Traditionally, it has been involved in analyzing and improving usability.

HCI has been an active area of research since the 1980s. It has focused on improving the design of user interfaces, and helping users transforming their goals into productive actions for the computers. Improving user interfaces and usability is important because poorly designed interfaces increase the potential for human error. In particular, human behavior is largely goal-driven, therefore the execution of activities which help the users to achieve their goals is the main key to create a

DOI: 10.4018/978-1-5225-2255-3.ch433

usable system. So, when a user "engages with a complex system of rules that change as the problem changes" (e.g. an interface does not present information clearly and coherently with a user mental model), it leads to "Cognitive Friction" (Cooper, 2004).

The "Cognitive Friction" is a by-product of the information age, and it is more evident in all the computing devices lacking a natural cause-effect relation between user input and device output, e.g. when similar inputs result in different outputs.

When a person is dealing with the cognitive friction, ancestral mechanisms of the human being come into play. As result, in this case, users cannot be modeled as purely rational beings. Thus, to understand users' behavior, and to appreciate how systems can be made usable, we need to consider the following factors:

- Users are driven by goals. People are naturally prone to pursuing goals. In achieving this, according to Krug "every question mark adds to our cognitive workload, distracting our attention from the task at hand" (Krug, 2005). This, according to Norman (Norman, 2002), creates usability issues, because it introduces the cognitive friction into play and leads users to make mistakes, which sometimes can also result into security flaws;
- Users do not read the instructions. Users proceed by trial and are not interested in reading manuals, instructions or documentation. For most of the users, it is not important to know how to do something, until the moment in which it is not necessary to use it (Krug, 2005);
- Users follow the path of least resistance. Several studies in the field of HCI have shown how users, in their task to accomplish a goal, tend to seek the path requiring them less effort (e.g. (Norman, 2002)). Once they find the first reasonable option allowing them to perform the desired action, it becomes irrelevant to them if it is not the most efficient and safe option. Furthermore, users have no incentive to improve. When users "find something that works - no matter how badly – they tend not to look for a better way" (Krug, 2005). Some operations can be inconvenient from the point of view of performance, others, in the long run, can cause damage to the system: users may be unaware of it until problems show up for the first time.

While many research studies in HCI has been focused in defining what usability is and, consequently, intervene in improving user interfaces, several studies have shown that the "ease of use" cannot be limited to those aspects alone (Whitten & Tygar, 1999) (Balfanz & al., 2004).

To increase the acceptance of the security mechanisms, conventional wisdom suggests it is sufficient to make them easier through a more usable user interface. In practice, however, it is not enough to provide a proper user interface, even in the case it is supported by specific configuration guidelines. This is what Whitten and Tygar argue, in their study "Why Johnny cannot encrypt" (Whitten & Tygar, 1999), which is a seminar paper in the usable security literature. This study focuses on analyzing data and email encryption of the security software Pretty Good Privacy 5.0 (PGP). They showed that user errors have not decreased, despite years of improvements to the graphical interface. This has led to additional studies looking beyond the interfaces.

This field of study, which deals with analyzing the usability issues related to security, is called HCI-Sec and was founded in 2000 by Whitten as a mailing list on Yahoo! Groups. It has been said that HCI-Sec "only rarely received significant attention as a primary subject for study" (Balfanz & al., 2004), this despite the fact that "usability remains one of the most pressing and challenging problems for computer security" (Whitten & Tygar, 1999).

Although HCI-Sec has only recently gained momentum, initial studies have their roots in 1975,

when (Saltzer & Schroeder, 1975) argued that the usability was an essential component of a secure system. In their seminar "The protection of Information in Computer System" they presented eight basic principles that serve as guidelines for the design of systems aimed at protecting information. The principle of the "Psychological Acceptability" is one of them, and states: "it is essential that the human interface is designed for ease of use so that users routinely and automatically apply the protection mechanisms correctly".

Since then, little work has been focused on HCI-Sec and, as a result, the security systems are sometimes poorly designed, leading to cases where users seek alternative interactions with the system or completely avoid the security mechanisms. Given the difficulty in making IT systems usable, it is unsurprising that the problem of "aligning usability and security" has been almost neglected until the beginning of the early 90s of the last century.

According to Fléchais, it is wrong to justify such a dearth of research as a tension between usability and security. Until that decade, the research community was more focused on technical trade-offs, such as for example the realization of robust encryption on low energy consumption microprocessors (Flechais & Sasse, 2005). This is reasonable, because, before the growth of the Internet, security was mainly a physical concern, and physical thinking was based on a military mindset.

The problem of usability in security, however, was not limited to this, and already existed during the 80s and 90s. For example, it was already possible to improve usability and possibly weaken security by automating common tasks. Brad Reid (Reid, 1987), argued that programmer convenience is the antithesis of security because it becomes intruder convenience if the programmer's account is compromised. Reid mentioned a "programmer" because at that time the main users of the computer systems were mostly researchers or computer science specialists who possessed some programming aptitude. These individuals possessed technical skills, received specific training and were, therefore, prone to ignore usability, and failing to identify the security implications this might have.

In the 1990s, with the diffusion of personal computers and the mass adoption of the Internet, the problem of usable security has remained virtually unexplored and did not leverage pre-existing HCI research. The initial solution to the problem addressed the symptoms rather than the root cause, by updating the anti-virus software, or the patching software in known problems. Therefore, research has been focused more on short-term practical gains, rather than long-term design changes that attend to both usability and security. A further problem is that few developers are trained in usability, or have significant software security experience.

The advent of HCI-Sec introduced the idea of security as an important consideration for usability, while usability is an important aspect of security (Cranor & Garfinkel, 2005). Therefore, if the purpose of HCI was to ensure that users would reach their goals by the use of better interfaces, HCI-Sec aims to ensure that users are able to achieve these goals also in the most secure way.

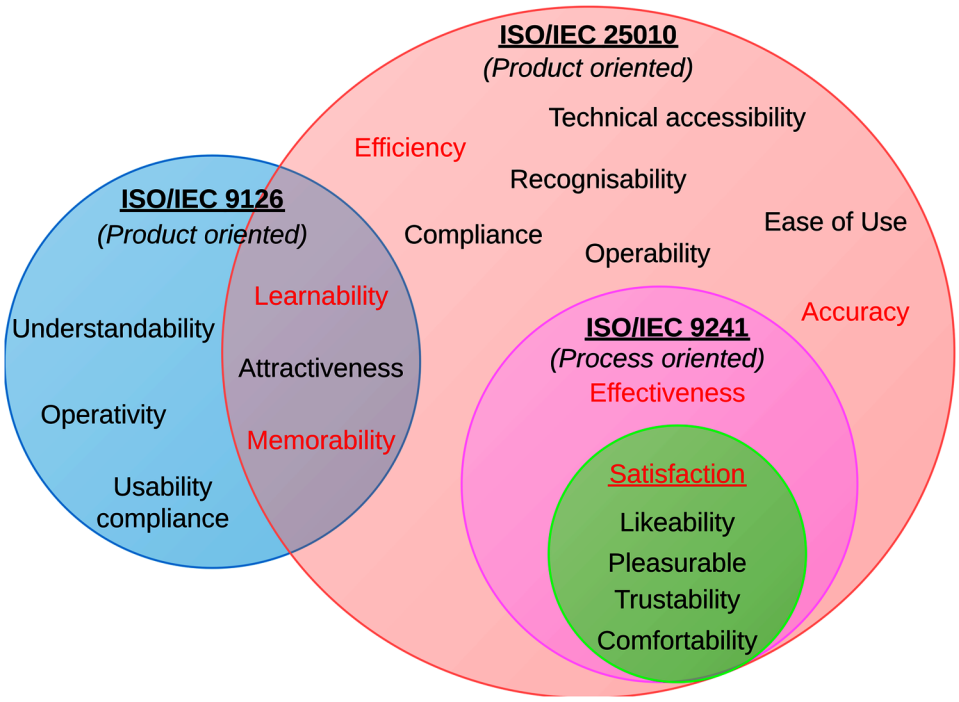
MAIN FOCUS OF THE CHAPTER

The aim of this chapter is to analyze the goals and state of the art of usability and security to determine where and how they can be effectively "aligned".

ISSUES, CONTROVERSIES, PROBLEMS

Usability has become a key factor in the quality of the software and has a determining role in productivity and acceptance (Cranor & Garfinkel, 2005). This term has more than one meaning, though; it refers multiple concepts that may or may not be taken together. Some are based on *execution time, performance, user satisfaction* and *ease of learn-*

Figure 1. Usability definitions according to ISO/IEC 9241-11, ISO/IEC 9126 and ISO/IEC 25010



ing (also known as *learnability*). Thus, it remains something that is not unequivocally defined, and is subject to interpretation based on the stake one has in usability (Hertzum, 2010).

Over the years, the International Organization for Standardization (ISO) itself produced various, and sometimes conflicting, definitions of usability (Figure 1). These definitions can be classified into two main categories: *product-oriented* and *process-oriented*. The former category provides definitions of the qualities that belong to the final product. This appears to be a reasonable approach, because software usability is essential for end-users, as crucial for achieving particular tasks quickly and effectively. The latter category focuses on the methodological aspects of obtaining usability: for a software developer, usability describes the internal attributes of a system, including concerns such as quality of design, documentation, and maintenance.

These various points of view and the different requirements have resulted in contrasting perspectives on usability, carried out by several groups of experts in a non-uniform and inconsistent way. For example, some terms have different meanings and labels. In document ISO/IEC 9241-11 (ISO, 1998) *learnability,* as a quality of a software, is designed as the "time of learning", while in ISO/IEC 9126 (ISO, 2012) is defined as "comprehensible input and output, instruction readiness, messages readiness".

Discrepancies among the standard can be even more significant: in the standard ISO/IEC 9126, usability is defined in a product-oriented way as a set of attributes that bear on the effort needed for use and on the individual assessment of such use, by a stated or implied set of users. The qualitative properties to be achieved are Understandability, Learnability, Operability, Attractiveness and Usability compliance (Abran & al, 2003). The standard that replaced it, ISO/IEC 25010, focuses on the user's goals and on how fast they are achieved, in addition to user satisfaction with the system. In this document, usability replaces learnability property in favour of o*perability* (Lew & Olsina, 2010), which is described as the degree

to which the product has attributes that enable it to be understood, to be learned, to be used, and to be attractive to the user, when used under specific conditions.

With this definition of operability, the properties to be achieved are *appropriateness, recognisability, ease of use, learnability, attractiveness, technical accessibility* and *compliance*. The same standard describes however also a different model, *Quality in Use*, in which the usability appears described as the extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency and satisfaction without adverse consequences in a specified context of use.

The process-oriented point of view was defined in the document ISO/IEC 9241, which is a suite of international standards on "Ergonomics of Human System Interaction". In Part 11, the definitions of usability from different perspectives are grouped together. The key components are effectiveness and satisfaction. The former describes the interactions from the point of view of process efficiency and puts the focus on the results and valuable assets. The latter requires carefulness on the user's needs. The standard attempts to explain how to identify the information that has to be taken into account when evaluating usability in terms of measures of user performance and user satisfaction.

The criterion of satisfaction is very difficult to be measured and, for this reason, additional usability factors have been proposed in Part 2 of ISO/IEC 25010. They are *likeability, pleasurable, comfort and trust.*

Given the subjectivity and the different contexts in which the term "usability" can be used, Kainda and Flechais (Kainda et al., 2010) proposed to consolidate it in six key factors which are defined as:

- **Effectiveness:** A system is only usable if its users can achieve intended goals, and effectiveness is measured by whether users are able to complete a particular task or not;
- **Satisfaction:** A system must be accepted by users, otherwise it is bound to fail, even if is usable;
- **Accuracy:** A system demands may have an impact on the user's tasks. For example, a system may require 100% accuracy in an providing information, such as a pin code or a password. However, this accuracy, is not always achievable by the user, making the system unusable;
- **Efficiency:** To guarantee usability, a system must ensure that each user's goals are achievable within an acceptable amount of time and effort;
- **Memorability:** A system may require users to memorize secrets, namely passwords. This may be problematic since the users are cognitively burdened with credentials, and other secrets;
- **Knowledge:** This corresponds to the Learnability property. However, the user may not attempt to learn or understand the system, as users tend to care only about the parts of the system of interest to them. Therefore, knowledge of the security mechanisms or policies is required by the user;

These characteristics can be measured in different ways. Effectiveness, satisfaction, efficiency and memorability can be measured directly, while accuracy and knowledge are measured indirectly, i.e. the first set of characteristics can be measured directly by quite simple empirical indicators, while the latter are typically derived by combining more indicators.

Another HCI-sec problem is that adequate usability is essential in specific security mechanism (e.g. authentication process), but the requirements for achieving it and a high level of security may collide (Braz & Robert, 2006). For example, in the case of password-based authentication, many usability principles (e.g. use shortcuts in case of frequent use, provide informative feedback)

contrast with best practices (e.g. password must not be showed during typing, and only success or failure must be reported, to mitigate social engineering and guessing).

SOLUTIONS AND RECOMMENDATIONS

From the perspective of HCI, there are several principles for building a system that is "quick to use and relatively error-free" (Johnson, 2007). One of the most important of these is ensuring the system "does what the user wants" without "complicating the user's task". Another important aspect is to evaluate the usability of a system. In this regard, the System Usability Scale (SUS) (Brooke, 1996) is a widely accepted base. The SUS is designed to give a quick impression of the overall usability of a product. It consists of ten questions (e.g. "I found this system unnecessarily complex", I found the various functions in this system were well integrated") rated on a Likert scale, resulting in an overall 0-100 value, where 100 represents excellent levels of usability. SUS has been adopted in many contexts, also in HCI-sec (De Witt, J. Kuljis, 2006) since quick to complete, thus avoiding user frustration and ensure answer accuracy.

From the perspective of methods and procedures used in HCI, many of them have been adapted to the HCI-Sec. The main difference is the focus on a balanced trade-off between usability and security.

The methodologies between HCI and HCI-Sec differ for at least five key aspects that are analyzed in the following paragraphs, detailing related recommendation as well.

THE SECONDARY GOAL

People do not generally sit at their computers wanting to manage their security; rather they want to send mail, browse web pages or download software.

Traditionally, security definitions have been defined around attackers. Unfortunately, doing so ignores the legitimate and non-malicious use, and also may adversely affect the system (Kainda & al, 2010). Users may not have the perception of damaging the system or, through making certain actions or inactions, bypassing security systems, putting their assets risk. Users must be constantly made aware of the operations involving security and the system must ensure that it is hard to make catastrophic errors. Furthermore, if such events occur, user actions should be reversible. To illustrate how this can be achieved, consider the implementation of dialog boxes requiring confirmation of a particular action. The implementation of the "Empty Trash" feature in desktop operating systems typically allows accidentally deleted files to be recovered, or the "Undo" button, now present in many desktop applications, allows for an action to be reverted. In a business setting, backup and redundancy of servers are amongst the systems used to avoid potential damages even from non-malicious users. Unfortunately, as Garfinkel has demonstrated, even "Empty Trash" functionality can behave in a manner inconsistent with a corresponding interface design (Garfinkel, 2005).

ABSTRACTION

Security policies are usually phrased as abstract rules that are easily understood by programmers but "alien" and unintuitive to many members of the wider user population.

(Johnson, 2007) proposes a focus on learnability and memorability, properties which, as we have discussed, belong to Usability. Facilitating the learning process is possible, by creating a consistent lexicon transmitted through the user interface. This is convenient since it was discovered that a particular trend also applies to IT users: they prefer not to invest time in training or reading manuals, but in learning the functionality of the system through the exploration of the user interface (Krug, 2005).

THE HIDDEN FAILURE

It is difficult to provide good feedback for security management and configuration because configurations are complex and not easy to summarize.

Making a secure system does not guarantee its security because the system must also be installed and used in a secure way. (Bishop, 2005) noted that the configuration is a key component of security because it is during the configuration of a system that it is defined who will interact with the system and how. Practitioners and security staff often make mistakes in applying default software configurations, ignoring the fact that different configurations lead to different security contexts. For example, a computer configured to be secure in a university research environment could be considered insecure in a military installation. In the former, information might be made accessible to the whole class or research group while, in the latter, they might be accessible only on a need to know basis.

One method used to counteract and minimize the adverse effects of an incorrect configuration is the "fail-safe default" principle (Saltzer & Schroeder, 1975). This states that the safest solution is a default configuration without any permission granted. During the configuration phase, a security responsible task selects the correct permissions for each system function and group of users. When configuring a firewall, this principle corresponds to the whitelist configuration: everything that is not explicitly allowed is forbidden by default. This policy contrasts with the blacklist, which grants any permission by default and chooses specifically the ones to forbid. The former one, despite being more difficult to handle, forces discussion on any permission to be enabled, thereby making the system more secure.

THE BARN DOOR

Once a secret has been left accidentally unprotected, even for a short time, there's no way to be sure it has not already been read by an attacker.

Once sensitive data or vital assets for the company are compromised and made public by mistake, it is possible that attackers will use it for their own advantage. There are several ways to approach this problem. You might try to avoid social engineering attacks, where even expert users fall victim to if channels of communication they trust and use regularly are compromised. This risk can be prevented using anti-fraud mechanisms, aimed at preventing phishing through e-mail or other channels. Should an attacker successfully obtain sensitive information such as passwords, private keys or credit card numbers, it should also be possible to erase and getting new information. In the case of commercially sensitive intellectual property, DRM can also be implemented, which can control access to resources, and revoke permissions in the event of a successful attack. DRM technology is, however, complex to maintain and not without its own usability issues (Favale & al., 2016)

THE WEAKEST LINK

The security of a networked computer is like a chain: it is only as strong as its weakest component.

It is generally recognized that the user is often the weak point of a computer system from a security perspective. However, as discussed, this creates a vicious circle in which users are kept unaware of what the security mechanisms are. Therefore, users are driven to the creation of their own security views, which fail to align with reality. To avoid this issue, security mechanisms should be complemented by specific guidelines that take into account the specific constraints of security mechanisms, minimizing discrepancies introduced by users with different backgrounds and skills.

FUTURE RESEARCH DIRECTIONS

Future research direction will address the problem of measuring usable security in a more systematic and practical way, or, as a first milestone toward the goal, understand if there are real advantages and tangible benefits resulting from formative and summative usability assessment processes. Such an assessment is not easy because of the previously discussed complex variables into play. Moreover, there are several aspects that are influenced by contextual conditions, such as economic resources, time, and other economic or innovation drivers.

To address these evaluation issues, we need to understand what it means to precisely evaluate usable security. Only then will it be possible to identify what methods can be effectively used in the analysis processes.

Also, all these facets activate different levels of sub-choices that depend on several variables and the context of use. One of these could, for example, be advancing the project in time: do you want to evaluate a system in its initial stage of development, or at a different iteration of the same application? It has already been noted that, while considering security at an early stage of a software product's design is virtuous, design techniques may be needed that specifically consider security at a later stage (Faily, 2015).

For any improvements, development should be scientifically measurable. As Lord Kevin said over 200 years ago "if you cannot measure it, you cannot improve it", meaning that without a scientifically sound evaluation methodology, would be difficult to draw any objective conclusions and take any proper improving actions (Atzeni & Lioy, 2005).

Measurement should not be an end in itself, but lead to something analogous to a benchmark, which is a result or a group of results that can become a point of reference and standard; this enables comparison and judgment on how good or bad things are.

CONCLUSION

Human behavior is goal-driven, therefore each aspect of a system interacting with users, security included, should be organized to help users to achieve their goals. In particular, security must be embedded paying attention to usability aspects, to avoid "cognitive friction".

Since usable security principles can be applied both to final product and to the production process, from one hand it is necessary to adopt methodologies to understand and measure the usability of the final artifact, from the other, all production line components should be considered in light of usability effectiveness, starting from the earliest steps in building software. This ensures that the quality of usable security is `built into' the final product, and diagnose the feedback that allows the project to be changed before its final release.

The ability to diagnose and correct an error in the usability of a software before entering the market is in itself a significant benefit, as are methods that can also be used to determine quality variations between two iterations of a given software. It is, therefore, necessary to describe and scientifically evaluate the properties of usability and security as two correlated factors, even if they are both difficult to quantify and even define. To facilitate quantification and definition, past literature split up usability and usable security in more atomic pieces (e.g. effectiveness, satisfaction, accuracy, efficiency, memorability) to make them more identifiable and comparable.

Finally, a scientifically sound usability assessment is a target of great interest. Further research is welcome because it is a complex problem (even when decomposed in sub-parts like satisfaction or memorability) and because the context of a product under evaluation can introduce influencing variables, enlarging the problem complexity.

REFERENCES

Abran, A., Khelifi, A., Suryn, W., & Seffah, A. (2003). Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*, *11*(4). DOI.10.1023/A:1025869312943

Adams, A. Sasse, M.A. (1999). Users are not the enemy. *Communications of the ACM*, *42*(12). DOI.10.1145/322796.322806

Atzeni, A., & Lioy, A. (2005). Why to adopt a security metrics? A little survey, *Proc. of QoP'05 - Quality of Protection, 1*(12).

Balfanz, D., Durfee, G., Grinter, R. E., & Smetters, D. K. (2004). In Search of Usable Security: Five Lessons from the Field. IEEE Security & Privacy, 2(5). DOI doi:10.1109/MSP.2004.71

Bishop, M. (2005). Psychological Acceptability Revisited. Security and Usability: Designing Secure Systems that People Can Use, 1(12).

Braz, C., & Robert, J. M. (2006). Security and usability: the case of the user authentication methods. *18th Conference on l'Interaction Homme-Machine*, 199-203. doi:10.1145/1132736.1132768

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability Evaluation in Industry*, *189*(194), 4-7.

Cooper, A. (2004). *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity* (2nd ed.). Sams Publishing.

Cranor, L. F., & Garfinkel, S. (2005). Security and Usability: Designing Secure Systems that People Can Use. O'Reilly.

De Witt, A. J., & Kuljis, J. (2006). Aligning Usability and Security-A Usability Study Of Polaris. *Proc. of the Symp. On Usable Privacy and Security.* doi:10.1145/1143120.1143122

Faily, S. (2015). Engaging Stakeholders during Late Stage Security Design with Assumption Personas. *Information and Computer Security, 435*(446).

Favale, M., McDonald, N., Faily, S., & Gatzidis, C. (2016), Human Aspects in Digital Rights Management: The Perspective of Content Developers. *Fourth International Workshop on Artificial Intelligence and Law.*

Flechais, I., & Sasse, M.A. (2005). Developing Secure and usable software. *OT2003*. DOI 10.1234/12345678

Garfinkel, S. (2005). Sanitization and Usability. *Security and Usability: Designing Secure Systems that People Can Use, 293*(317).

Hertzum, M. (2010). Images of Usability. *International Journal of Human-Computer Interaction*, *567*(600).

ISO/IEC. (1998). *ISO/IEC 9241-11 - Guidance on Usability management*. ISO.

ISO/IEC. (2012). *ISO/IEC 27002:2007 - Product quality*. ISO.

Johnson, J. (2007). *Common User Interface Design Don'ts and Dos*. Interactive Technologies.

Kainda, R., Flechais, I., & Roscoe, A. W. (2010). Security and Usability: Analysis and Evaluation. *ARES '10 International Conference on Availability, Reliability, and Security*, *275*(282). Doi:10.1109/ARES.2010.77

Krug, S. (2005). *Don't Make Me Think: A Common Sense Approach To The Web Usability*. New Riders Pub.

Lew, P., Li, Z., & Olsina, L. (2010). Usability and user experience as key drivers for evaluating GIS application quality. *International Conference on Geoinformatics, 1*(6). doi:10.1109/GEOINFORMATICS.2010.5567803

Norman, D. A. (2002). *The Design of Everyday Things*. Basic Books.

Reid, B. (1987). Reflections on some recent widespread computer break-ins. *Communications of the ACM*, *103*(105). doi:0.1145/12527.315716

Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE, 1278*(1308). doi:10.1109/PROC.1975.9939

Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the 'Weakest Link' - a Human/Computer Interaction Approach to Usable and Effective Security. *BT Technology Journal*, *122*(131). doi:10.1023/A:1011902718709

Schneier, B. (2003). *Beyond fear: Thinking sensibly about security in an uncertain world*. Copernicus Books. doi:10.1057/palgrave.sj.8340200

Whitten, A., & Tygar, J. D. (1999). Why Johnny can't encrypt: a usability evaluation of PGP 5.0. *USENIX Security Symposium*. Retrieved from http://dl.acm.org/citation.cfm?id=1251421.1251435

ADDITIONAL READING

Nielsen, J. (1993). *Usability Engineering*. Academic Press.

KEY TERMS AND DEFINITIONS

Cognitive Friction: The affinity friction between the user and the software that originates in the user mind when a product does not behave the way the user expects (e.g. a button on the screen that does not trigger any action when the user press it). (https://www.linkedin.com/pulse/20140801230851-205508682-what-the-heck-is-cognitive-friction).

Comfort (ISO/IEC 25010): The extent to which the user is satisfied with physical comfort.

Effectiveness: The properties which measures to what extent interactions achieve objective process efficiency indicators (i.e. concrete results of user actions while using the addressed product).

Likeability (ISO/IEC 25010): The extent to which the user perceives achievement of pragmatic goals, including successful subjective results of use and consequences of use.

Memorability: A factor which measures how much a product require users to memorize secrets (e.g. passwords or passphrases).

Operability: The degree to which the product has attributes that enable it to be understood, be learned, be used and be attractive to the user, when used under specific conditions.

Pleasurable (ISO/IEC 25010): The extent to which the user is satisfied with his perceived achievement of hedonistic goals of stimulation, identification and evocation and associated emotion responses.

Psychological Acceptability: A founding principle of usable security stating that **"***it is essential that the human interface is designed for ease of use so that users routinely and automatically apply the protection mechanisms correctly".*

Process-Oriented Usability: The categorization of usability aiming to achieve it addressing the characteristics of the process to obtain the final product (e.g. documentation and design effort).

Product-Oriented Usability: The categorization of usability aiming to achieve it addressing the final products characteristics (e.g. learning curve to use the product).

Satisfaction: The property which measures to what extent the user's needs are subjectively satisfied by the product.

Trust (ISO/IEC 25010): The extent to which the user is persuaded that the product will behave as intended.