An Assessment Framework for Usable-Security Based on Decision Science

Yasser M. Hausawi and William H. Allen

Department of Computer Sciences,
Florida Institute of Technology,
Melbourne, FL 32901, USA
{yhausawi@my.,wallen@}fit.edu

Abstract. The balance between security and usability must be addressed as early as possible in the Software Development Life Cycle (SDLC) to ensure the inclusion of usable-security in software products. Unfortunately, there has been little research on assessing and integrating security, usability, and usable-security during the requirements engineering phase of the SDLC. To address that deficiency, this paper proposes an Assessment Framework for Usable-Security (AFUS) based on two well-known techniques from the decision science field.

Keywords: Security, Usability, Human Computer Interaction, HCI, HCI-SEC, Usable-Security, Quality Attributes Assessment, Decision Science.

1 Introduction

Security and usability are two important software quality attributes that should be incorporated into software projects during the requirements phase [10,15]. However, implementing both in a particular product is problematic because the goals of security and usability are often in conflict [1,8,24,25]. Much research has been done by HCI and security specialists to bring security and usability into a synergetic integration [7,18] and a more recent approach to resolving these potential conflicts is to employ a hybrid attribute, namely: usable-security [14,16,19]. However, most of the research on usable-security has focused on the design phase of the SDLC, resulting in a usable-security assessment gap in the requirements phase [5]. A recent literature survey found no current usable-security assessment methodology that addresses the requirements phase.

The field of Decision Science provides tools and techniques for resolving conflicts between differing objectives [6]. In this paper, we propose an Assessment Framework for Usable-Security (AFUS) that explores the benefits of using two well-known techniques from Decision Science, namely Utility Functions and Decision Trees, for assessing the balance between security, usability and usable-security represented in the set of requirements for a particular software product.

The goal of this work is not to produce an objective measure for comparing two products, but rather to generate a metric that developers can use to gauge the balance between the attributes. We assume that the developers of a product

T. Tryfonas and I. Askoxylakis (Eds.): HAS 2014, LNCS 8533, pp. 33–44, 2014.
© Springer International Publishing Switzerland 2014

are aware of the balance between security and usability that is appropriate for their product, thus the proposed technique is intended to assist in reaching that desired balance. As changes to the requirements are made, reassessment using AFUS can indicate if the product has shifted to a greater emphasis on one attribute at the expense of the others, or if all attributes have moved towards the developer's preferred equilibrium.

Next section (Section 2 presents background information about security, usability, and usable-security; and presents the related assessment research work. Section 3 introduces our usable-security assessment framework (AFUS). Section 5 discusses the results, and Section 6 concludes this article.

2 Background

2.1 Security

There are various definitions of the term "security". Garfinkel and Spafford define a computer as secure, "if you can depend on it and its software to behave as you expect it to" [14]. Pfleeger and Pfleeger define computer security as, "preventing the weaknesses from being exploited and understanding preventive measures that make the most sense" [21]. Essentially, system security is a set of methods and techniques that work together to generate what is called *security mechanisms.* The security mechanisms are used to prevent weaknesses of computer systems from being exploited by applying three main security properties: 1) confidentiality, 2) integrity, and 3) availability [21].

2.2 Usability

Usability is defined by the International Standard Organization (ISO) as the limit that a product can be used by legitimate users to satisfactorily perform specific tasks in an effective, efficient, and specified way [17]. Usability specialists have developed various techniques to achieve three main usability properties: 1) effectiveness, 2) efficiency, and 3) user satisfaction.

2.3 Usable-Security

In 2003, a multi-discipline group of researchers formed a working group called *Human-Computer Interaction and Security (HCI-SEC)* [12]. This group was formed to bridge the gap between usability and security under the main goal of "Usable-Security". In other words their goal was to come up with usable-security mechanisms to secure computer systems. Usable-security is defined by Whitten and Tygar [26] as a software product that makes its users: 1) reliably aware of the needed security tasks, 2) able to figure out how to successfully perform such tasks, 3) able to avoid dangerous errors when performing their tasks, and 4) sufficiently comfortable to use and be happy with the software interface.

Unfortunately, much of the recent research on the assessment of quality attributes does not consider assessing the results of aligning two or more attributes.

As a result, each researcher focused on assessing one attribute. However, the Security Usability Symmetry "SUS" [4] is a novel subjective metrics-based usability inspection design model proposed to design, inspect, and evaluate the usability of security systems through identifying and then subjectively rating security-usability related problems according to the three-level severity rating (low, medium, and high). One disadvantage of the SUS is that, like many other usability and security evaluation techniques, it adopts the subjective (qualitative) evaluation methodology rather than the objective (quantitative) one.

Therefore, in this paper, we propose a framework that uses a mathematical modeling assessment [3] through application of utility functions and decision trees. Moreover, our framework reduces the subjective-based assessment to produce a more objective-based assessment.

3 Assessment Framework for Usable-Security (AFUS)

The proposed framework (see Figure 1) has three main components: 1) requirements filtering and merging using the OWASP Risk Rating Methodology [20] for security and the SALUTA Attribute Preference Table [13] for usability as guides, 2) utility functions, and 3) a decision tree.

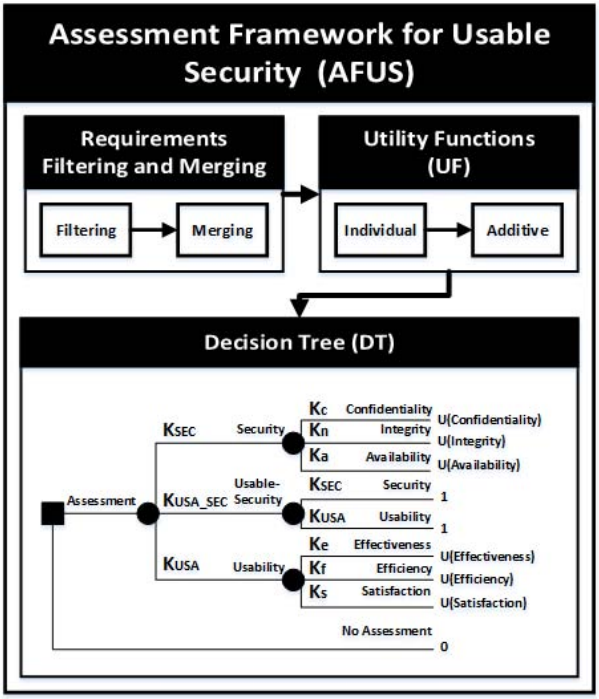

Fig. 1. Assessment Framework for Usable-Security (AFUS)

The framework was evaluated by using case studies based on three real-world scenarios. Scenarios are often used in software engineering to gather and validate non-functional requirements [5] and in HCI to improve communication between stakeholders and developers [2]. The first scenario was based directly on the non-functional usability, humanity, and security requirements from the Volere Requirements Specification Template [22]. Two additional scenarios were produced with different specifications for their non-functional requirements. We should note that the usable-security requirements were derived from the other non-functional requirements, as the Volere template does not have a usable-security section.

3.1 Requirements Filtering and Merging

Requirements engineering is the first phase of the SDLC [23] . In this phase, the stakeholders meet together to set the project requirements and analyze the generated requirements [23]. Among those requirements are security, usability, and other requirements. The scope of the AFUS is limited to the requirements that are related to security, usability, and usable-security. Therefore, the intended system requirements are filtered to select and gather the requirements that are within the scope of the AFUS, and then they are grouped into three main groups, namely: security requirements, usability requirements, and usable-security requirements. The requirements of each of the three groups are rated according to their importance as the following processes:

Security Requirements Group (SR). The OWASP Risk Rating Methodology (OWASP RRM) [20] has been adapted for use as the foundation for assessing security requirements. The importance of each security requirement is rated on each security property [21] (confidentiality (CI), integrity (NI), and availability (AI)) based on Table 1. We observed that OWASP does not use standardized rating values, as each security property has its own rating. Also, there are some gaps in rating some properties. For example, OWASP uses the values (1, 5, 7, and 9) to estimate the impact of availability loss on the system if vulnerability is exploited. Such estimation values may experience lack of accuracy, because an estimated impact between 1 and 5, let us say 3, cannot be accurately given. This may force the estimator to choose between 1 or 5. Therefore, we adapted the OWASP rating methodology to fill in those rating gaps to better align it with the usability requirement ratings, as will be explained later in the next section. The security rate of each requirement (SecR) is calculated through averaging the rates of the three security properties. The calculation formula is shown as the following:

$$SecR_i = \frac{CI_i + NI_i + AI_i}{3} \tag{1}$$

The overall security rate (SEC) for the requirements set is derived by:

$$SEC = \sum_{i}^{SR} SecR_i \tag{2}$$

Moreover, to assess the shares (SH) of each security property (confidentiality (SH_c), integrity (SH_n), and availability (SH_a)), summing of each property importance is calculated from all of the security requirements and divided by overall security rate (SEC) as depicted on the following three formulae respectively:

$$SH_c = \frac{\sum_i^{SR} CI_i}{SEC}, SH_n = \frac{\sum_i^{SR} NI_i}{SEC}, SH_a = \frac{\sum_i^{SR} AI_i}{SEC} \tag{3}$$

Table 1. Security and Usability Properties Importance Rating Guidance

Requirement Importance on Properties	[9] Critical [7] Very Important [6] Important [3] Some Important [1] Not Important

Usability Requirements Group (UR). SALUTA [13] is a usability assessment technique used to rate usability based on assigning quantitative values for usability preferences. An adapted rating methodology based on SAULTA Attribute Preference Table [13] is used to rate usability requirements. The same rating values that were used to rate the security requirements group are used to rate the importance of each usability property [17] (effectiveness (EI), efficiency (FI), and satisfaction (SI)) for the usability requirements. It is worthwhile to note that having unified rating values for both security and usability requirements provides a consistent qualification strategy for measuring the requirements. Table 1 is used as a guide for the rating process. A five-value rating is used (1, 3, 6, 7, and 9). This rating can be justified as the most appropriate for usability requirements rating process, because it efficiently helps rating any usability requirement on the three usability properties where the evaluator is not forced to give an inappropriate rate. Although SAULTA uses a four-value rating (1, 2, 3 and 4), as it ranks scenarios based on four usability properties (each property gets one ranking value), our framework rates requirements, but does not rank them, and the rating guidance should work with all of the requirements. The usability rate of each requirement (UsaR) is calculated through averaging the rates of the three usability properties. The calculation formula is:

$$UsaR_j = \frac{EI_j + FI_j + SI_j}{3} \tag{4}$$

The overall usability rate (USA) for the requirements set is derived by:

$$USA = \sum_j^{UR} UsaR_j \tag{5}$$

To assess the shares (SH) of each usability property (effectiveness (SH_e), efficiency (SH_f), and satisfaction (SH_s)), summing of each property importance is calculated from all of the usability requirements and divided by overall usability rate, USA, as depicted on the following three formulae respectively:

$$SH_e = \frac{\sum_j^{UR} EI_j}{USA}, SH_f = \frac{\sum_j^{UR} FI_j}{USA}, SH_s = \frac{\sum_j^{UR} SI_j}{USA} \tag{6}$$

Hence, after rating the requirements of both security and usability (based on the modified OWASP and the SALUTA methods) to produce overall static ratings for both attributes (SEC and USA), a static assessment is calculated for the two attributes (SEC_{static} and USA_{static}) by applying the following formulae:

$$SEC_{static} = \frac{SEC}{SEC + USA}, USA_{static} = \frac{USA}{SEC + USA} \tag{7}$$

Usable-Security Requirements Group (USR). This requirements group has two sub-groups, namely: 1) initial usable-security requirements sub-group ($IUSR$), and 2) merged usable-security requirements sub-group ($MUSR$). The overall usable-security rate (USA_SEC) is calculated by summing the two sub-groups. The following sections describe each sub-group.

Initial Usable-Security Requirements Sub-Group (IUSR). The requirements of this sub-group are rated by a different rating methodology, as usable-security does not have standard properties like those associated with security and usability and a usable-security requirement may mix security and usability properties. Moreover, the requirements that are based on the Human-Computer Interaction and Security (HCI-SEC) are considered as IUSR [9,11]. Therefore, each initial usable-security requirement is rated based on Table 2, then multiplied by 2, and then divided by 3 as illustrated on this formula:

Table 2. Initial $Usable - Security$ Importance Rating Guidance

Importance (I)	[9] Critical
	[7] Very Important
	[6] Important
	[3] Some Important
	[1] Not Important

$$IUsa\_SecR_k = \frac{I_k * 2}{3} \tag{8}$$

The overall initial usable-security rate ($IUSA\_SEC$) is calculated from the following formula:

$$IUSA\_SEC = \sum_k^{IUSR} IUsa\_SecR_k \tag{9}$$

Merged Usable-Security Requirements Sub-Group (MUSR). To assess usable-security in the most appropriate manner, both security and usability requirements must be analyzed with merging and alignment in mind [14,16]. If the requirements are merged successfully, the security-usability alignment can be balanced to achieve usable-security. Therefore, the requirements of the two groups, security requirements and usability requirements, are visited again and analyzed to prepare them for merging. Once new usable-security requirements are derived from the existing security and usability requirements, they are rated ($MUsa\_SecR$) through averaging the security and usability rates (SecR ,UsaR) of all contributing requirements (CSR and CUR) multiplied by 2 as the following:

$$MUsa\_SecR_l = \frac{\left(\left(\frac{\sum_i^{CSR} SecR_i + \sum_j^{CUR} UsaR_j}{CSR+CUR}\right) * 2\right)}{3} \qquad (10)$$

The overall merged usable-security rate ($MUSA\_SEC$) is calculated from the following formula:

$$MUSA\_SEC = \sum_l^{MUSR} MUsa\_SecR_l \qquad (11)$$

The overall prediction of usable-security rate (USA_SEC) for the entire system is calculated by the following formula:

$$USA\_SEC = IUSA\_SEC + MUSA\_SEC. \qquad (12)$$

3.2 Utility Functions

Utility Functions (UF) are a relatively straightforward methodology for dealing with conflicting objectives and can capture stakeholders' attitudes about predictive assessment and the evaluation of trade-offs [6]. Utility functions are often used in systems engineering and management for decision and risk analysis purposes. There are various models of utility function. One is the Additive Utility Function (AUF) that is used to estimate total utility of conflicting objectives. Another utility function model is the Individual Utility Function (IUF). The IUF is used to predictively estimate utilities for subjectively measurable/non-measurable objectives. More details about the above utility function models are available in [6]. Usable-security is a subjectively measurable hybrid software quality attribute that is based on two conflicting quality attributes, namely: security and usability, along with consideration of HCI-SEC principles [11,16]. Therefore, the utility function models can be adapted for usable-security assessment during the requirements engineering phase. Assessing usable-security during the requirements phase can provide clear prediction about the balance between security and usability early in software development process. Based on the requirements filtering and merging component's process, both the IUF and AUF models can be used to assess usable-security.

First, the ratio-based IUF is used to calculate weights for the software quality attributes: security, usability, and usable-security. The ratios of security (R_{SEC}), usability (R_{USA}), and usable-security ($R_{USA-SEC}$) are derived by the following calculation, where α represents an attribute (security, usability, or usable-security) and β represents another attribute: α is $\frac{\alpha}{\beta}$ times as important as β. For instance, the ratio of security over usability is calculated based on the above calculation as follows: Security is $\frac{SEC}{USA}$ times as important as usability. The attribute's accumulative ratio is calculated through summing its ratios over all the attributes. For instance, the security accumulative ratio is calculated by summing the security ratios over all the attributes as in the following:

$$R_{SEC} = \frac{SEC}{SEC} + \frac{SEC}{USA} + \frac{SEC}{USA\_SEC} \tag{13}$$

After the accumulative ratios of the attributes are derived, each attribute is weighted on the following formulae, where security, usability, and usable-security weights are $K_{SEC}, K_{USA}, andK_{USA\_SEC}$ respectively [6], i represents a quality attribute, and QA represents the number of all the quality attributes :

$$K_{SEC} = \frac{1}{\sum_i^{QA} R_i} * R_{SEC}, K_{USA} = \frac{1}{\sum_i^{QA} R_i} * R_{USA}, K_{USA\_SEC} = \frac{1}{\sum_i^{QA} R_i} * R_{USA\_SEC} \tag{14}$$

Second, the IUF is used to calculate weights of each of security and usability properties based on their ratios (R) and pointing (P), where the starting pointing value is five (5). The following formulae are used to calculate the weights of security properties: confidentiality (K_c), integrity (K_n), and availability (K_a) [6], i represents a property, and SP represents the number of all properties. The weights calculation is applied as follows:

$$K_c = \frac{1}{\sum_i^{SP} P_i} * P_c, K_n = \frac{1}{\sum_i^{SP} P_i} * P_n, K_a = \frac{1}{\sum_i^{SP} P_i} * P_a \tag{15}$$

The following formulae are used to calculate the weights of usability properties: effectiveness (K_e), efficiency (K_f), and satisfaction (K_s) [6], j represents a property, and UP represents the number of all properties. The weights calculation is applied as follows:

$$K_e = \frac{1}{\sum_j^{UP} P_j} * P_e, K_f = \frac{1}{\sum_j^{UP} P_j} * P_f, K_s = \frac{1}{\sum_j^{UP} P_j} * P_s \tag{16}$$

Third, the IUF is used to calculate utilities of each of security and usability properties based on the ratios (R), pointing (P) where the starting pointing value is five (5), and the following equations are used to find values of constants a and b for each of security and usability individually [6]:

$$b = \frac{1}{(-1) * minP_{properties} + maxP_{properties}}, a = ((-1) * minP_{properties}) * b \tag{17}$$

Based on the values of the constants a and b on security, the following formulae are used to calculate the utilities (U) of security properties: confidentiality (U_c), integrity (U_n), and availability (U_a):

$$U_c = a + (b * P_c)\,, U_n = a + (b * P_n)\,, U_a = a + (b * P_a) \tag{18}$$

Similarly, based on the values of the constants a and b on usability, the following formulae are used to calculate the utilities (U) of usability properties: effectiveness (U_e), efficiency (U_f), and satisfaction (U_s) [6]:

$$U_e = a + (b * P_e)\,, U_f = a + (b * P_f)\,, U_s = a + (b * P_s) \tag{19}$$

Fourth, the AUF is used to calculate the overall utility of the quality attributes based on their properties' weights and utility values. The following formulae represent the AUF for the security and usability quality attributes:

$$U_{SEC}\left(c, n, a\right) = K_c * U_c + K_n * U_n + K_a * U_a, U_{USA}\left(e, f, s\right) = K_e * U_e + K_f * U_f + K_s * U_s \tag{20}$$

Usable-security utility (U_{USA_SEC}) differs from the utility of the above two attributes because usable-security does not have properties. However, it is a result of merging the two quality attributes, namely: security and usability. Therefore, the following formula is used to calculate the utility of usable-security:

$$U_{USA\_SEC}\left(SEC, USA\right) = K_{SEC} * U_{SEC} + K_{USA} * U_{USA} \tag{21}$$

3.3 Decision Trees

The Decision Tree (DT) is a tool used during the process of modeling decisions [6]. It is a method of structuring different objectives' elements in order to make decisions for using the objectives based on displaying all of the minute details. Quality attributes in general, and security, usability, and usable-security in particular, are objectives of software development within the scope of our framework. More information about decision trees is available in [6].

To get the overall utility value of the Decision Tree for the three quality attributes, the weights and utilities of each attribute are calculated by the following formulae. It is important to mention that to get the overall utility for usable-security, we subtracted the gap between security and usability utilities as one important factor that plays a role in assessing the usability-security interaction (usable-security):

$$DTU_{SEC} = K_{SEC} * U_{SEC} \tag{22}$$

$$DTU_{USA} = K_{USA} * U_{USA} \tag{23}$$

$$DTU_{USA\_SEC} = \left(K_{USA\_SEC} * U_{USA\_SEC}\right) - \left|DTU_{SEC} - DTU_{USA}\right| \tag{24}$$

Finally, to get the final assessment value for the three quality attributes, the resulted Decision Tree utility value of each attribute is divided by the total summing of the three Decision Tree utilities of all the three attributes as in the following formulae, where the sum of the results must equal 1:

$$ASS_{SEC} = \frac{DTU_{SEC}}{(DTU_{SEC} + DTU_{USA} + DTU_{USA\_SEC})} \quad (25)$$

$$ASS_{USA} = \frac{DTU_{USA}}{(DTU_{SEC} + DTU_{USA} + DTU_{USA\_SEC})} \quad (26)$$

$$ASS_{USA\_SEC} = \frac{DTU_{USA\_SEC}}{(DTU_{SEC} + DTU_{USA} + DTU_{USA\_SEC})} \quad (27)$$

4 Results and Discussion

For each of the three scenarios (see Figure 2), we first created a baseline by applying the static OWASP RRM [20] and SALUTA APT [13] assessments for security and usability requirements respectively, using predetermined values for rating each of the two attributes' requirements. Then, we applied the AFUS approach to reassess the balance between the three attributes.

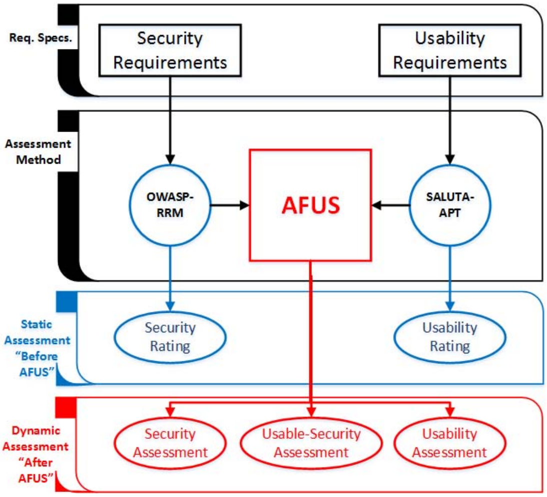

Fig. 2. The "Before" and "After" Assessment Results

The table below shows the outcomes from the static assessments ("before") and after applying the AFUS. As seen in Table 3, the results of all three scenarios show moderate differences in assessing security and usability attributes before and after applying AFUS. Moreover, usable-security weight was only incorporated after applying AFUS. The AFUS assessment of the first two scenarios show a moderate to small range of assessment gap between the security and usability attributes, 16.96% and 10.44% respectively. The third scenario showed a very little assessment gap between the security and usability attributes, 0.80%. Therefore, it provided a higher weight for usable-security.

Table 3. Assessment for Scenarios 1, 2, and 3

Scenario	Assessment	Security	Usable-Security	Usability
Scenario 1	Before applying AFUS	0.364	No Assessment	0.636
	After applying AFUS	0.365	0.197	0.439
Scenario 2	Before applying AFUS	0.432	No Assessment	0.568
	After applying AFUS	0.348	0.338	0.315
Scenario 3	Before applying AFUS	0.504	No Assessment	0.496
	After applying AFUS	0.251	0.500	0.249

5 Conclusion

We proposed an Assessment Framework for Usable-Security (AFUS) that employs two well-known techniques from Decision Science to assess the balance between security, usability and usable-security represented in the set of requirements for a particular software product. We demonstrated that this approach can extend the work of the currently available techniques in order to produce objective results, but more work is needed to determine how responsive this approach is to changes in requirements and how accurately it measures the balance between the three attributes. Unfortunately, the lack of prior work on assessing usable-security requirements complicates this task.

References

1. Adams, A., Sasse, M.A.: Users are not the enemy. Communications of the ACM 42(12), 40–46 (1999)
2. Anton, A.I., Carter, R.A., Dagnino, A., Dempster, J.H., Siege, D.F.: Deriving goals from a use-case based requirements specification. Requirements Engineering 6(1), 63–73 (2001)
3. Bosch, J.: Design and use of software architectures: adopting and evolving a product-line approach. Pearson Education (2000)
4. Braz, C., Seffah, A., M'Raihi, D.: Designing a trade-off between usability and security: A metrics based-model. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4663, pp. 114–126. Springer, Heidelberg (2007)
5. Chung, L., do Prado Leite, J.C.S.: On non-functional requirements in software engineering. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 363–379. Springer, Heidelberg (2009)
6. Clemens, R.T., Reilly, T.: Making hard decisions with decision tools® (2001)
7. Cranor, L.F., Garfinkel, S.: Guest editors' introduction: Secure or usable? IEEE Security & Privacy 2(5), 16–18 (2004)
8. DeWitt, A.J., Kuljis, J.: Is usable security an oxymoron? Interactions 13(3), 41–44 (2006)
9. Dhamija, R., Dusseault, L.: The seven flaws of identity management: Usability and security challenges. IEEE Security & Privacy 6(2), 24–29 (2008)

10. Ferre, X.: Integration of usability techniques into the software development process. In: International Conference on Software Engineering (Bridging the gaps between software engineering and human-computer interaction), pp. 28–35 (2003)
11. Ferreira, A., Rusu, C., Roncagliolo, S.: Usability and security patterns. In: Second International Conferences on Advances in Computer-Human Interactions, ACHI 2009, pp. 301–305. IEEE (2009)
12. Flechais, I., Mascolo, C., Sasse, A.: Integrating security and usability into the requirements and design process. International Journal of Electronic Security and Digital Forensics 1(1), 12–26 (2007)
13. Folmer, E., van Gurp, J., Bosch, J.: Scenario-based assessment of software architecture usability. In: ICSE Workshop on SE-HCI, Citeseer, pp. 61–68 (2003)
14. Garfinkel, S.: Design Principles and Patterns for Computer Systems that are Simultaneously Secure and Usable. PhD thesis, Massachusetts Institute of Technology (2005)
15. Gorton, I.: Software quality attributes. In: Essential Software Architecture, pp. 23–38 (2011)
16. Hausawi, Y.M., Mayron, L.M.: Towards usable and secure natural language processing systems. In: Stephanidis, C. (ed.) HCII 2013, Part I. CCIS, vol. 373, pp. 109–113. Springer, Heidelberg (2013)
17. WD ISO. 9241-11. ergonomic requirements for office work with visual display terminals (VDTs). In: The International Organization for Standardization (1998)
18. Lampson, B.: Privacy and security usable security: How to get it. Communications of the ACM 52(11), 25–27 (2009)
19. Mayron, L.M., Hausawi, Y., Bahr, G.S.: Secure, usable biometric authentication systems. In: Stephanidis, C., Antona, M. (eds.) UAHCI 2013, Part I. LNCS, vol. 8009, pp. 195–204. Springer, Heidelberg (2013)
20. OWASP. Risk rating methodology (2013)
21. Pfleeger, C.P., Pfleeger, S.L.: Security in Computing. Prentice Hall PTR (2006)
22. Robertson, J., Robertson, S.: Volere requirements specification template: Edition January 14 (2009)
23. Sommerville, I., Sawyer, P.: Requirements engineering: a good practice guide. John Wiley & Sons, Inc. (1997)
24. Weir, C.S., Douglas, G., Carruthers, M., Jack, M.: User perceptions of security, convenience and usability for e-banking authentication tokens. Computers & Security 28(1), 47–62 (2009)
25. Whitten, A.: Making Security Usable. PhD thesis, Princeton University (2004)
26. Whitten, A., Tygar, D.: Why johnny can't encrypt: A usability evaluation of pgp 5.0. In: Proceedings of the 8th USENIX Security Symposium, vol. 99, McGraw-Hill (1999)