User-Centric Security

Optimization of the Security-Usability Trade-Off

Denis Feth
Fraunhofer Institute for Experimental Software Engineering IESE
Kaiserslautern, Germany
denis.feth@iese.fraunhofer.de

ABSTRACT

Security and usability are highly important and interdependent quality attributes of modern IT systems. However, it is often hard to fully meet both in practice. Security measures are complex by nature and often complicate work flows. Vice versa, insecure systems are typically not usable in practice. To tackle this, we aim at finding the best *balance* between usability and security in software engineering and administration. Our methodology is based on active involvement of large user groups and analyzes user feedback in order to optimize security mechanisms with respect to their user experience, with a focus on security awareness. It is applied during requirements elicitation and prototyping, and to dynamically adapt unsuited security policies at runtime.

Keywords

Security, Usability, Security-Awareness, Measurement

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: [Security and Protection]; H.1.2 [**Information Systems**]: User/Machine Systems—*Human factors*

1. INTRODUCTION

Security and usability are key success factors of modern software systems. However, security and usability requirements are frequently conflicting, depend on the context, are driven by different stakeholders, and realization strongly depends on the project budget. There exist plenty of examples where software is highly user-friendly, but neglect the security of data or digital assets. Vice versa, security systems frequently restrict the user or put additional work on him, and thus have a negative impact on satisfaction, efficiency, perceived security, or other usability aspects. This usually results in a low acceptance of the software or system. In the worst case, inappropriate security measures decrease security of the system, as users are tempted to circumvent them in order to fulfill their tasks.

ESEC/FSE'15, August 30 – September 4, 2015, Bergamo, Italy
© 2015 ACM. 978-1-4503-3675-8/15/08...$15.00
http://dx.doi.org/10.1145/2786805.2803195

We approach this challenge from two sides: *Security Tailoring* and *Security Awareness*. Security tailoring deals with the problem that user requirements are complex, context-dependent and evolve over time. Cranor and Garfinkel state that "insufficient communication with users produces a lack of a user-centered design in security mechanisms" [3]. However, there is no approach to systematically involve users in order to assess the influences and dependencies between security and usability and adapt security measures at runtime. Balancing security and usability is typically an ad-hoc task performed by system engineers (development time) or administrators (runtime). At development time, user-testing helps to understand the users' needs—however results are typically project-specific and neither stored nor generalized for reuse. Furthermore, user-testing can hardly cover runtime problems. Thus, the first part of our solution is to collect and analyze user feedback from the crowd in order to, potentially dynamically, adapt security measures and find an optimal security-usability balance. The second problem is, that users are typically neither aware of the current threat situation nor of the required behavior in this situation. As the goal is to inherently integrate the user into the security chain, the user must be aware about these facts at any time. Thus, the second part of our solution is to integrate security awareness into business processes and applications. Again, we use feedback from the crowd to elicit and understand the user's needs, but also his typical behavior and view on security.

Contribution and Research Questions. We propose a methodology that actively collects and interprets feedback and passively monitors (mis-)uses and policy violations from potentially large user groups. Based on this, we create an experience base that is used to optimize the security-usability trade-off at development time (e.g., during requirements engineering) and to adapt unusable security measures, business processes and applications at runtime. In order to achieve this goal, we have to solve different research questions:

- How can usability and security dependencies be measured using a joint quality model?
- What are suitable (e.g., effective, non-disturbing, privacy-preserving) techniques to collect and enrich user feedback?
- How can root causes be identified based on the collected data, considering semantics and interdependencies?

- How can ratings and similarity measures be used to generate recommendations based on data from the experience base?

- How can the methodology be integrated into the software development life cycle, including administration?

2. METHODOLOGY

Idea. Technical security measures are typically sophisticated and, in principle, very effective. However, in the end, a human administrates, uses or is effected by the technical measure. In order to achieve *user-centric* security measures, the user has to be considered an essential part of the security chain. To this end, we target three goals: Security *Acceptance, Understanding* and *Appreciation*

Acceptance of security is the minimum level. Users accept that security measures are necessary and use them in practice. On the other hand, they are not assumed to have an understanding of underlying threats, problems or technology and do not have an intrinsic motivation to apply security measures. However, if users are "forced" to apply security measures without knowledge or intrinsic motivation, this can only be successful if security measures have minimal a negative effect on user experience. We approach this problem with *Security Tailoring*, i.e., the optimization of security functions for the end user with respect to usability. Security understanding and appreciation build upon this basis. The challenge is to motivate users to engage in security and security optimization. When users understand threats and objectives of security measures, we assume them to have a higher tolerance with respect to security (e.g., execution of additional steps). Finally, when users have an (ideally intrinsic) motivation to actively use and apply security measures, we can consider them as real part of the security chain. We approach these two goals with *Security Awareness*, i.e., the comprehensive support and provision of information with respect to threats, security functions and expected behavior in case of security incidents.

Example. In the following, we illustrate our methodology with examples from the area of mobile computing and Mobile Device Management (MDM). Enterprises increasingly support the use of (private and business) mobile devices, such as smart phones and tablets. These devices are integrated into the company's infrastructure and business processes, and are controlled and protected via MDM solutions. The problem is that MDM policies are typically defined statically (i.e., company-wide and context-independent) by administrators or security officers without involvement of the users.

Methodology. In this section, we present our envisaged methodology that uses user feedback and monitoring to dynamically analyze and optimize the security-usability trade-off. Feedback evaluation is used both for tailoring of security measures (i.e., we use feedback to rate and optimize unusable security measures) and for optimizing awareness (i.e., we identify information demands of the user and adapt business processes or applications to close the information gap). The proposed methodology is depicted in Figure 1 and has four steps: *Reporting*, *Root Cause Analysis*, *Rating*, and *Recommendation*.

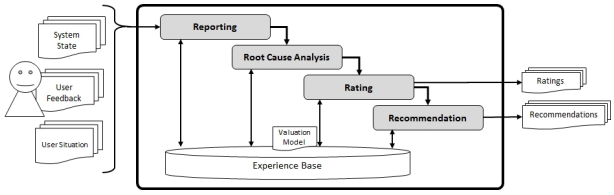

Figure 1: Methodology

2.1 Reporting

Reporting is the starting point of the process. Users actively provide (probably mainly negative) feedback when they use security measures or experience problems. Types of feedback include:

- *Security measures are too restrictive.* For example, the user cannot take pictures to document meeting results, because an MDM policy prohibits camera usage at companies premises (data loss prevention policy).
- *Security measures put additional load on the user.* For example, the display timeout is set very low, which causes frequent password authentication and results in low comfort for the user.
- *Security measures are too complex.* For example, user has no knowledge about cryptography and is unable to setup or use PGP on the device.
- *Threat situation and expected behavior is unclear.* For example, the user is not aware about the risk of espionage and reads sensitive document in a train without further protection.

It is important that each collected report creates a complete picture about the user's situation and problem. Thus, user-provided feedback is automatically enriched with information about the system state and the user situation. The user situation typically contains information that describe the context the user is currently in (e.g., traveling, office, meeting) [11, 10]. However, this information is also critical with respect to privacy, which needs to be considered. The system state contains information about the system and security measures (e.g., running apps, configuration, and active security policies). Which data is required in a specific setting strongly depends on the used quality model (cf. Section 2.3).

2.2 Root Cause Analysis

User-provided feedback is typically vague and informal. Furthermore, users are seldom aware about security functions and system internals and just report "unexpected behavior". For example, a user simply reports "camera application not working". Like in this example, user feedback does not necessarily have a direct relation to a specific security measure. Thus, a *Root Cause Analysis* has to be performed that maps a report to one or more security measures, if applicable. In our case, the camera app is not working because an MDM policy prohibits camera usage at the company's premises.

This phase is especially challenging, as usability issues are typically the result of a combination of different aspects.

This includes task specifics, the user context and interdependencies between security measures or other system components. As the following steps rely on the correct identification of the underlying security function(s), it is important to achieve a high precision and recall in this phase.

2.3 Rating

The *Rating* of security functions based on a joint quality model for security and usability is the main part of the methodology. Figure 2 shows typical security (left) and usability (right) aspects that need to be considered in the model.

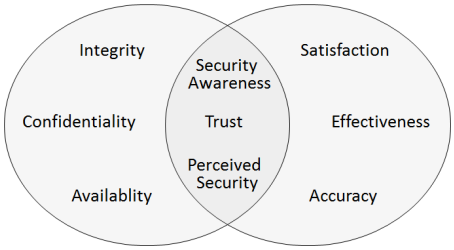

Figure 2: Aspects of Security and Usability

Joint aspects, such as trust, perceived security or security awareness are of special interest for us. In this area, there are two things to consider: First, *usability requires security.* On the first glance, a system without any security measure should have the highest usability. However, the user might not have a good feeling using such a system, i.e. trustworthiness is an issue. Second, *security requires usability.* All technical security functions are useless, if the user cannot apply them. He is typically the weakest link in the security chain—a fact that is massively exploited in social engineering attacks.

In our example, the MDM policy increases confidentiality. However the complete prohibition of camera usage has negative impact on satisfaction, effectiveness and trust. In order to perform a rating that is used in to compare and optimize security measures, we have to build a joint model that reflects these dependencies. In turn, this requires all of the aspects to be quantifiable, i.e., there need to be metrics for both security and usability, which is a challenging task [5] and currently part of our future work.

2.4 Recommendation

Finally, ratings and similarities of different security measures or security-related processes and applications are used to derive recommendations from data in the experience base (ratings, solutions, categorizations of security functions). This can be done by experts, or automatically via models and similarity metrics. Depending on the context, recommendations target the following:

- *Improvement of security measures.* Optimization mainly targets the optimization of restrictive, complex or inconvenient security measures. An exemplary recommendation is to change MDM policy to allow camera usage (while warning the user), but prohibit the redistribution of the picture (if technically feasible).
- *Improvement of business processes.* Optimization mainly targets extensive security awareness. An example is adding guidance during security-relevant process steps.
- *Improvement of business applications.* Optimization targets both, the optimization of unusable security applications and the creation of security awareness on an application level. An example is the use of consistent and approved design patterns to highlight security-critical assets.

We are currently researching how recommendations can be derived for each of the afore-mentioned types.

2.5 Application and Integration

There exist three potential scenarios, where our methodology can be applied and stakeholders benefit from our knowledge base and/or recommendations:

1. *Software engineers* use recommendations and data from the experience base in early phases of the system development. In particular, the methodology can be used to evaluate different alternatives during prototyping phases, or during requirements engineering in terms of A/B testing[4] or Crowd-RE. In the maintenance phase, software can be optimized based on experience gained during the productive operation of the software.
2. *Administrators* establish a continuous feedback loop about the usability of security measures and receive recommendations about how to adapt unusable security functions or policies. If dynamic runtime policies are enforced (like in IND^{2}UCE [6]), administrators can be supported during policy specification [13].
3. *Users* provide feedback and benefit from usable security measures that are tailored to their needs. Furthermore, they can be supported with explanations or solutions for current problems.

3. EVALUATION

In terms of evaluation, applicability (effectiveness) and expectable benefits (efficiency) of the proposed methodology are of major importance. In addition, the quality model needs to be evaluated with respect to correctness and completeness, and it has to be analyzed which guarantees can be given with respect to ratings and recommendations. We plan to perform at least two case studies and one controlled experiment to evaluate our approach. In the case studies we aim to find problems and trade-offs, especially with respect to security awareness, for different application scenarios. In the controlled experiment, we try to show that effects on usability can be predicted when changing characteristics of security measures.

4. RELATED WORK

This work is cross-sectional and is closely related to security and usability/user experience (UX) measurement. There exist multiple theoretical work, both in the areas of usability and UX [9, 17, 15] and security [2, 16, 14] measurement. However, the focus is often limited to one aspect without considering the respective other.

Work on the security-usability trade-off has been performed, both theoretically [7, 3, 1] and in form of case studies [19, 18, 8]. However, case studies are specific to one system or application domain and lack a generic, systematic methodology and theoretical work frequently lacks a practical applicability. In contrast to the presented work, we systematically involve large user groups for analysis and optimization of security measures and integrate into the software life cycle.

5. DISCUSSION AND CONCLUSION

We presented our idea to rate and optimize security measures in order to improve usability and security awareness. The core of our methodology is a quality model (as part of the rating phase) for security and usability and the involvement of potentially large user groups to compare the security and usability of different security functions at run time and at development time.

Our main application scenario is the assessment and optimization of usage control [12] policies. Usage control policies provide powerful means to control the usage of data and services, but typically restrict the user. Previous work focused on the usability of the policy specification [13], but neglected the usability of the enforced policy on the user. However, it is also important to know how (categories of) policies have to be implemented in order to achieve a good user experience while still fulfilling the security requirements.

The presented idea is in a very early phase, and there exists little evidence about the applicability of the methodology or specific parts of the presented process. Many issues, topics and research questions are still open and need to be further developed. Besides others, these are:

- Elicitation of requirements for an integrated quality model for security and usability that builds the foundation for the rating phase
- Based on the quality model, identification of required information for rating and recommendation and how it can be collected
- Identification and implementation of non-disturbing and privacy compliant data collection techniques
- Integration of the proposed methodology into the software life cycle (development time) and administration (runtime)

Despite these open issues, we assume that a systematic involvement of large user groups, combined with the implementation of an experience base will lead to an increased level of security *and* usability. Both aspects are highly interdependent and can benefit from each other.

6. ACKNOWLEDGEMENTS

This work is supervised by Prof. Dr. Dr. h.c. Dieter Rombach (TU Kaiserslautern) and realized at Fraunhofer IESE.

7. REFERENCES

[1] Al-Saleh, M.: Fine-grained reasoning about the security and usability trade-off in modern security tools. Dissertation, The University of New Mexico (2011)

[2] Brotby, W., Hinson, G.: PRAGMATIC Security Metrics. Auerbach Publications (Jan 2013)

[3] Cranor, L., Garfinkel, S.: Security and Usability. O'Reilly Media, Inc. (Aug 2005)

[4] Dixon, E., Enos, E., Brodmerkle, S.: A/B testing (Nov 2013)

[5] Dörr, J.: Elicitation of a complete set of non-functional requirements. Fraunhofer-Verlag (2010)

[6] Feth, D., Pretschner, A.: Flexible Data-Driven Security for Android. In: 2012 IEEE Sixth International Conference on Software Security and Reliability. pp. 41–50. IEEE (Jun 2012)

[7] Garfinkel, S.L.: Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable. Gene 31, 234–239 (2005)

[8] Good, N., Krekelberg, A.: Usability and privacy: a study of Kazaa P2P file-sharing. Proceedings of the SIGCHI conference on ... (5), 137–144 (2003)

[9] Jordan, P.W., Thomas, B., McClelland, I.L., Weerdmeester, B.: Usability Evaluation In Industry. CRC Press (1996)

[10] Jung, C., Feth, D., Elrakaiby, Y.: Automatic Derivation of Context Descriptions. In: 2015 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support. IEEE (2015)

[11] Jung, C., Feth, D., Seise, C.: Context-Aware Policy Enforcement for Android. In: 2013 IEEE 7th International Conference on Software Security and Reliability. pp. 40–49. IEEE (2013)

[12] Pretschner, A., Hilty, M., Basin, D.: Distributed usage control. Communications of the ACM 49(9), 39 (Sep 2006)

[13] Rudolph, M.: User-friendly and Tailored Policy Administration Points. In: 1st International Conference on Information Systems Security and Privacy (to appear) (2015)

[14] Rudolph, M., Schwarz, R.: A Critical Survey of Security Indicator Approaches. In: 2012 Seventh International Conference on Availability, Reliability and Security. pp. 291–300. IEEE (Aug 2012)

[15] Sarodnick, F., Brau, H.: Methoden der Usability Evaluation. Verlag Hans Huber (2011)

[16] Scandariato, R., Paci, F., Tran, L.M.S., Labunets, K., Yskout, K., Massacci, F., Joosen, W.: Empirical Assessment of Security Requirements and Architecture: Lessons Learned. In: Engineering Secure Future Internet Services and Systems, Lecture Notes in Computer Science, vol. 8431, pp. 35–64. Springer (2014)

[17] Tullis, T., Albert, B.: Measuring the User Experience. Elsevier (2008)

[18] Whitten, A., Tygar, J.: Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In: Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8. p. 14. USENIX Association (Aug 1999)

[19] Whitten, A., Tygar, J.D.: Usability of security: A case study. Computer Science pp. 1–41 (1998)