Two Requirements for Usable and Secure Software Engineering

Shamal Faily
Computing Laboratory,University of Oxford
Wolfson Building,Parks Road,Oxford UK OX1 3QD
shamal.faily@comlab.ox.ac.uk

ABSTRACT
Despite the acknowledged need for systems to be both usable and secure, we lack guidance on how developers might build such systems. Based on recent research, we believe evidence exists that blending techniques from Security, Usability, and Software Engineering can lead to effective design processes for building such systems. In this position paper, we discuss two requirements that need to be satisfied before blending can occur: treat Security and Usability as complementary, and provide convincing tools for developers.

Author Keywords
User-Centered Design, Secure Software Engineering, Requirements Engineering,

ACM Classification Keywords
D.2.2 Software Engineering: Design Tools and Techniques

INTRODUCTION

There is no simple answer for why we cannot consistently build usable and secure systems. A knee-jerk answer would be to simply blame the developer. However, a developer might reasonably ask how such a system should be specified? Even the question of where to start has no agreed answer. Security and Usability Engineers argue that their respective design techniques need to lead the way; Software Engineers believe that understanding the characteristics of the system being built is a more natural first step. Invariably, the decision of what comes first is delegated to whatever methodology a developer selects. In applying these, developers have many techniques to choose from, some of which include treatment for Security or Usability concerns, yet design processes do not treat both. Best practice continues to involve treating Security and Usability as generic qualities which contend with functionality. Yet, there is growing evidence that the design of usable and secure systems deserves more attention. The US Department of Homeland Security as ranked Usable Security as one of the top cyber-security research topics for government and the private sector [12], and HCI-Security is a growing research topic. Nevertheless, there is still a dearth of work prescribing how usable and secure systems should be built.

Based on the evidence from our research, we believe that Security, Usability, and Software Engineering lifecycles *can* be blended into an effective process for developing software. However, several requirements need to be satisfied in order for this to be achieved. We shall discuss two of these in the following sections.

TREAT SECURITY AND USABILITY AS COMPLEMENTARY

Security and Usability are considered by many developers to be conflicting qualities. However, if we consider the perspectives both communities take on what constitutes design then we discover that these are complementary. The Security Engineering community views design as a means of understanding how a system can be securely developed, or made more secure; its techniques aim to understand what system risks are, and what design decisions are necessary to adequately respond to them. The Usability Engineering community views design as a means of understanding how a system can be situated for its users; their techniques are used to build artifacts embodying users, their goals, and their activities.

Both communities treat design not as a process, but as a hermeneutic circle. Nuseibeh alludes to this in his twin-peaks model [14], which talks about the dialogue between requirements and architectural activities. We assert that, when building usable and secure systems, Usability design insights can inform Security, and vice-versa. In one Critical Infrastructure Protection study we carried out, we found that the analysis of empirical data contributing to Personas and Scenarios led to the discovery of several important vulnerabilities and threats [8]. In another study [to appear], we found that modelling how one user unintentionally exploited a system led to modified security controls, stopping this exploitation, and simultaneously improving the effectiveness and efficiency of his tasks.

PROVIDE CONVINCING TOOLS

Integrating design techniques from different areas accounts for little if the resulting data cannot be managed, analysed, and applied. Unfortunately, the quality of tool-support available for Security, Usability, and Software Engineering activities is variable; tool-support is particularly sparse and

poorly-integrated during the early stages of design. For example, the Requirements Engineering tools of choice continue to be spreadsheets, word processors, or wikis. In Usability Engineering, software tools are predominantly used for sketching prototypes rather than supporting conceptual design. Security design tools often rely on the addition of modelling extensions to existing CASE tools, e.g. attack tree support for Eclipse [13], and Misuse Case support for DOORS [2]. We believe developers will remain ambivalent about the merits of Security and Usability techniques until convincing tool-support is available.

Our experiences developing and applying CAIRIS (Computer Aided Integration of Requirements and Information Security) [1] suggest that tool-support has enormous potential for fostering the use of Security, Usability, and Software Engineering techniques in a single design process. In our studies, we found that not only can tools encourage the use of such techniques, but the associations between Security, Usability, and Software Engineering concepts can glean hitherto unnoticed insights from the data they manage. For example, we have illustrated how simple quantitative and qualitative data analysis can form the basis for automatically visualising the impact to Usability of Security design decisions in different contexts of use, and vice versa [5]. Tools can also lead to process efficiencies in the time taken to build User-Centered Design artifacts. We recently demonstrated how the sense-making activities associated with qualitative data analysis can, with tool-support, be leveraged to semi-systematically generate Persona characteristics [11].

ABOUT THE AUTHOR

Shamal's research interests are in understanding how existing techniques and tools from Security, Usability, and Software Engineering can be integrated to support the practical design of usable and secure systems. This research has informed the design of a meta-model for Usable and Secure Requirements Engineering [4], and tool-support embodying the characteristics needed to design secure and usable systems [10]. This research has also provided new insights into how existing work from User-Centered Design [6, 7] and innovation theory [9] can be used to engineer secure systems.

Shamal's doctoral research has led to the development of the IRIS (Integrating Requirements and Information Security) framework; processes instantiated by this framework act as exemplars for what technique and tool integration from these different areas might look like. In particular, Shamal's thesis [3] is that this framework is an exemplar for integrating existing techniques and tools towards the design of usable and secure systems.

REFERENCES

1. CAIRIS web site. `http://www.comlab.ox.ac.uk/cairis`, January 2011.
2. I. Alexander. Misuse cases: use cases with hostile intent. *Software, IEEE*, 20(1):58 – 66, 2003.
3. S. Faily. *A framework for usable and secure system design*. PhD thesis, University of Oxford, 2011. To Appear.
4. S. Faily and I. Fléchais. A Meta-Model for Usable Secure Requirements Engineering. In *Software Engineering for Secure Systems, 2010. SESS '10. ICSE Workshop on*, pages 126–135. IEEE Computer Society Press, May 2010.
5. S. Faily and I. Fléchais. Analysing and Visualising Security and Usability in IRIS. In *Availability, Reliability and Security, 2010. ARES 10. Fifth International Conference on*, pages 543–548, 2010.
6. S. Faily and I. Fléchais. Barry is not the weakest link: Eliciting Secure System Requirements with Personas. In *Proceedings of the 24th British HCI Group Annual Conference on People and Computers: Play is a Serious Business*, BCS-HCI '10, pages 113–120. British Computer Society, 2010.
7. S. Faily and I. Fléchais. The secret lives of assumptions: Developing and refining assumption personas for secure system design. In *HCSE'2010: Proceedings of the 3rd Conference on Human-Centered Software Engineering*, pages 111–118. Springer, 2010.
8. S. Faily and I. Fléchais. Security through usability: a user-centered approach for balanced security policy requirements. In *Poster at: Computer Security Applications Conference, 2010. ACSAC '10. Annual*, Dec. 2010.
9. S. Faily and I. Fléchais. To boldly go where invention isn't secure: applying Security Entrepreneurship to secure systems design. In *Proceedings of the 2010 workshop on New security paradigms*, NSPW '10, pages 73–84, New York, NY, USA, 2010. ACM.
10. S. Faily and I. Fléchais. Towards tool-support for Usable Secure Requirements Engineering with CAIRIS. *International Journal of Secure Software Engineering*, 1(3):56–70, July-September 2010.
11. S. Faily and I. Fléchais. Persona cases: A technique for grounding personas. In *CHI '11: Proceedings of the 29th international conference on Human factors in computing systems*, Vancouver, BC, Canada, 2011. ACM. To Appear.
12. D. Maughan. The need for a national cybersecurity research and development agenda. *Communications of the ACM*, 53(2):29–31, 2010.
13. P. Meland, D. Spampinato, E. Hagen, E. T. Baadshaug, K.-M. Krister, and K. S. Velle. SeaMonster: Providing tool support for security modeling. *NISK 2008. Norsk informasjonssikkerhetskonferanse, Universitetet i Agder, Kampus Gimlemoen*, November 2008.
14. B. Nuseibeh. Weaving together requirements and architectures. *Computer*, 34(3):115–117, 2001.