Designing Usable and Secure Software with IRIS and CAIRIS

Shamal Faily

Designing Usable and Secure Software with IRIS and CAIRIS

Springer

Shamal Faily
Department of Computing & Informatics
Bournemouth University
Poole, Dorset
UK

ISBN 978-3-319-75492-5 ISBN 978-3-319-75493-2 (eBook)
https://doi.org/10.1007/978-3-319-75493-2

Library of Congress Control Number: 2018938649

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Claudia

Foreword

It is seldom the case in software engineering that we encounter a book that is relevant to practitioners, researchers and educators. Most books will either focus on the practitioner audience or the researcher/educator audience. This book is a welcome addition to the small set of books in our field that are relevant to all of these audiences. Regardless of which lifecycle model you use, if you are at all concerned with security and usability (and you should be!), this book is relevant to you.

I especially liked the mapping of the material in the book to its use by practitioners, researchers and educators. Since I have had all of those roles over the course of my career, I know how important it is to be able to focus on the topics of interest to me, and to be able to skim the others. Naturally, those of us who have written books would like our readers to read the entire book, but when we put ourselves in the shoes of our readers, we realise that we seldom have the time to do this.

The requirements engineering research community is vibrant but relatively small —a few hundred researchers, almost all of whom know one another. When we focus specifically on security and usability, as it relates to specification and design, the field becomes even more specific. This book is a welcome addition to the relatively small number of books that focus on security and usability in design.

The vast majority of the literature in the area consists of journal and conference research papers. The educational material tends to consist of tutorials, a few lectures, and maybe a course or two here and there. One of the significant contributions of this book is to provide an excellent source of information about security and usability in requirements and design, with copious references. It also provides a combination of overview material and more detailed explanations, depending on the subject at hand.

The first few chapters provide a nice overview of the security and usability concepts needed to understand the remainder of the book. In Chap. 3, the IRIS meta-model is introduced. I particularly liked the table with the core concepts of IRIS and the description of each. So often terminology is introduced without any explanation whatsoever, but that is not the case here. In Chap. 4, the IRIS process framework is introduced. The associated perspectives on usability, requirements

and security are all discussed, and then in Chap. 5 the tool support provided by CAIRIS is introduced. The discussion of personas and scenarios in Chap. 6 sets the stage for the case studies that follow.

The case studies help to make the methodologies more useful to practitioners, and from a practitioner perspective, tool support is essential. Paper exercises are OK for the classroom, but when large complex systems are under development, tool support is needed. Remaining chapters delve into architectural risk analysis, design issues and future research challenges.

All in all this book is an excellent choice for those who need to be concerned with security and usability, and that is just about everyone! This book is especially useful during the early lifecycle activities of requirements engineering, architecture and high-level design. In fact, given the current focus on security and usability, I think this book is a good choice for all systems and software engineers. It will expand your horizons.

Pittsburgh, PA, USA
March 2018

Nancy R. Mead, Ph.D.
SEI Fellow, Carnegie Mellon University
IEEE Fellow
ACM Distinguished Educator

Preface

Why do We Need This Book?

Everyone agrees that security needs to be considered as early as possible when designing any form of technology. Unfortunately, the security industry has been so effective at promoting stories about potential 'digital Pearl Harbours' and astronomical costs of cybercrime to national economies that many people are being paralysed into inaction, or pushed into irrationality when thinking about security. Given the demands to deliver more, sooner, with fewer resources, it can be appealing to suspend disbelief and trust that a piece of technology which promises the moon and costs the earth really will mitigate the latest threats, fit seamlessly into our enterprise's infrastructure, or otherwise 'add value'. Alternatively, one might treat scare stories as noise, and—as Cormac Herley once wrote[1]—conclude that the rational thing to do is ignore any advice that has been given. Unfortunately, relying on such shortcuts means that a product or service may fail to protect things of most value to us when we need them most.

When technology fails or is at risk of failing, it can also be tempting to blame the person interacting with it, i.e. blaming the user for committing a human error. However, if security is to be built-in to technology then it needs to respect the goals and demands of those that rely on it. This means rather than treating users as the 'weakest link' in security, a system design should consider the characteristics of its direct or indirect users. Unfortunately, as hard as building security it can be, it appears building both usability *and* security are even harder. Platitudes like 'you should think of security and usability up-front' are prescribed frequently, but useful best practice or case study examples seem to be in short supply. Identifying the

[1]Herley C. So Long, and No Thanks for the Externalities: The Rational Rejection of Security Advice by Users. In: NSPW '09: Proceedings of the 2009 New Security Paradigms Workshop. ACM; 2009. pp. 133–144.

form that design techniques and tools might take to demonstrably incorporate security and usability into the earliest stages of software design was the main motivation for devising IRIS and CAIRIS.

What are IRIS and CAIRIS?

IRIS (Integrating Requirements and Information Security) is a process framework that can be used to devise processes for designing usable and secure software. CAIRIS (Computer-Aided Integration of Requirements and Information Security) is a software platform for eliciting, specifying and validating secure and usable systems. It was built from the ground up to support IRIS, and includes the elements necessary for usability, requirements and risk analysis.

The book contains everything you need to understand what IRIS and CAIRIS are, and how you can use the two to design usable and secure software at an early stage. However, despite the book's title, this book is more than just a playbook for IRIS, or a manual for CAIRIS. The book was written to show how design activities can attend to both security and usability without sacrificing one for the other, or sacrificing one or both for innovation of functionality. As such, IRIS and CAIRIS are exemplars for design frameworks and software tools for secure and usable design, not the final word.

Assumed Background

This book assumes you have some background or general interest in Security Engineering, Usability or Requirements Engineering, and are motivated to learn more about the other two. You are not expected to be an expert in any of these areas.

Those wanting a general overview of the field of Security Engineering will find *Security Engineering: A Guide to Building Dependable Distributed Systems* by Ross Anderson (2nd Edition, Wiley, 2008) useful. Because Threat Modelling techniques are useful when designing for security, readers may also be interested in *Threat Modeling: Designing for Security* by Adam Shostack (Wiley, 2014), as well as *Securing Systems: Applied Security Architecture and Threat Models* by Brook Schoenfield (CRC Press, 2015).

Many options exist for those wanting an overview of the field of Usability and related design topics; *Interaction Design: Beyond Human-Computer Interaction* by Jenny Preece, Helen Sharp and Yvonne Rogers (4th Edition, Wiley, 2015) provides both good coverage of the field, together with practical advice on the techniques covered in the book.

There are fewer options available when it comes to Requirements Engineering textbooks, but—given the attention IRIS and CAIRIS pays to KAOS and modelling

in general—then *Requirements Engineering: From System Goals to UML Models to Software Specifications* by Axel van Lamsweerde (Wiley, 2009) provides a useful overview of what Requirements Engineering can offer to design. Those desiring more hands-on advice on the art of eliciting, specifying and validating requirements would benefit greatly from *Mastering the Requirements Process: Getting Requirements Right* by Suzanne and James Robertson (3rd Edition, Addison-Wesley, 2012); this book is also the authority on the Volere framework, which informed how IRIS and CAIRIS deal with requirements.

When devising an IRIS process, it is also assumed you have some knowledge or experience on the design techniques that form the basis of the process. To illustrate the adaptation of usability design techniques, this book covers the persona technique in some depth; other techniques are given a lighter treatment. References are, however, provided at the end of each chapter for readers interested in using the framework techniques discussed in Sect. 4.3.

How to Use This Book

This book was written to be read from cover-to-cover, but you can still skip to certain chapters based on your background and interests.

Practitioners

Chapter 1 will help you put this book in context, and provides a roadmap for where to read next. If your focus is usability and incorporating interaction design for security, then Chap. 4 provides guidance on getting up and running with IRIS processes, and Chap. 6—using personas and scenarios as examples—provides ideas about how design techniques can be adapted to provide assurance necessary to support security decision-making.

If your background is in security, and you want ideas for incorporating Human Factors into your work, looking at how CAIRIS provides tool support for risk and task analysis in Chap. 5 will provide some inspiration, together with Chap. 9, which shows how security and usability can be incorporated into later stages of software design. You will also find that Chap. 11 provides practical advice on tackling security problems by treating solutions as innovations.

Irrespective of your background, you will find the case study chapters (Chaps. 7, 8 and 10) useful for seeing how security and usability processes and tools work in practice. If, while reading this book, you need more background material on why certain concepts are related in IRIS and CAIRIS, then Chap. 3 will provide a useful aide-memoire.

Researchers

For those wishing to leverage or extend IRIS and CAIRIS, Part I will put whatever knowledge you have in context with complementary areas of design. The case study chapters will also put IRIS and CAIRIS in context; they illustrate the use of IRIS and CAIRIS, and present lessons learned as directions for future research.

The final chapters of this book describe numerous opportunities for future work. For example, Chap. 11 will benefit researchers in Security Economics by illustrating how economics and entrepreneurship can benefit each other. For researchers interested in the future potential of tool support for usable and secure design, then Chap. 12 provides examples of how the CAIRIS platform facilitates research and design in areas such as system-of-system engineering, practice integration with DevOps and the provision of rich exemplars for evaluating research as well as design.

Educators and Students

This book was also written to support advanced undergraduate or postgraduate teaching in HCI-Security. The material in this book has already been used at Bournemouth University in a one-semester (10 weeks) course on *Security by Design* taken by final year students on the Forensic Computing and Security undergraduate degree programme, and postgraduate students on the Cyber Security and Human Factors postgraduate degree programme. The Indicative Learning Outcomes (ILOs) for that course are

1. Select and critically evaluate the usability criteria that security mechanisms have to meet to be usable in their contexts of use.
2. Apply techniques from interaction design and security engineering to design and evaluate secure systems.
3. Devise and evaluate supplemental measures that an organisation needs to ensure long-term, productive security.

The material used to satisfy ILOs 1 and 3 is heavily inspired by a course on *People and Security* designed by Angela Sasse that I once had the opportunity to deliver to students at University College London and the University of Oxford; this material is drawn from the academic literature in HCI-Security, Information Security and Economics of Information Security.

This book supports the satisfaction of ILO 2; Table 1 summarises the contribution this book makes to cogent lectures associated with this learning outcome. The seminars for these lectures use an on-going case study example based on *ACME Water*: a 'human-centred' specification exemplar that represents the operating environment of a hypothetical UK water company, and was designed on the basis of experiences gleaned carrying out the case study of Chap. 8.

Table 1 Material from this book used to support a course on *Security by Design*

Lecture	Material used
Introduction	Challenges designing usable and secure software in Chap. 1 are used to motivate the course
Design techniques and tools	Material in Chap. 6 on personas and misusability cases provide examples of adapting interaction design techniques for security
Requirements	Examples of Security Requirements Engineering techniques and frameworks from Chap. 2, together with the IRIS framework in Chap. 4. The IRIS meta-model in Chap. 3 illustrates the relationship between Security, Usability and Requirements Engineering concepts
User Interfaces	Processes and tools for risk and task analysis with CAIRIS in Chap. 5
Architecture	Patterns and processes associated with Architectural Risk Analysis in Chap. 9, and the end-to-end webinos case study in Chap. 10
Security economics and entrepreneurship	Chapter 11 is the reading for the Security Entrepreneurship component of this lecture

Using and Contributing to CAIRIS

This book illustrates the capabilities of CAIRIS, but does not go into the installation, setup and general use of the tool in any depth. You can, however, find full instructions on how to setup and use the platform on the CAIRIS website: https://cairis.org.

CAIRIS is an open-source software product. The source code has been released under an Apache License, and is freely available on GitHub. Raising an issue on the CAIRIS GitHub site is strongly encouraged if you find any problems with the software or its documentation. As an open-source project, CAIRIS is sustained by its community of developers and users. There are many ways interested readers can make a contribution to CAIRIS, from proposing new enhancements, raising bug reports or even contributing pull requests for any typographic errors or inconsistencies found in the available documentation. No contribution to CAIRIS is too small.

Acknowledgements

The doctoral research that led to initial versions of both IRIS and CAIRIS was funded by EPSRC CASE Studentship R07437/CN001. I am very grateful to EPSRC and QinetiQ Ltd. for the sponsorship of this work. I am also grateful to the Department of Computer Science at the University of Oxford and Wolfson College for additional sponsorship of fieldwork and conference-related activities while

presenting this early work. Thanks go to Annamaria Carusi, Marina Jirotka, Angela Sasse and Andrew Simpson for their detailed and insightful comments on this work during its evolution from doctoral studentship proposal to submitted and corrected DPhil thesis. I would like to give special thanks to Dr. Ivan Fléchais, for his advice and encouragement, not only while carrying out the initial work leading to IRIS and CAIRIS, but follow-on work that took place at both Oxford and Bournemouth.

I am grateful to everyone on the DSS project, as well the water company behind the ACME name for allowing themselves to be used as case studies for evaluating IRIS and CAIRIS, finding the time to support my fieldwork, and provide feedback on the deliverables resulting from the respective IRIS processes applied. I am also grateful to everyone on the EU FP7 webinos project (FP7-ICT-2009-5 257103) for their support. The webinos project provided an important case study in the evolution of IRIS and CAIRIS; it provided a useful testbed for CAIRIS, and led to numerous improvements to the platform. The use of IRIS and CAIRIS to support the end-to-end design and development of webinos over its 3-year life also provided one of the first published long-term case studies for security and usability design.

The work described on Security Entrepreneurship in Chap. 11 was carried out while participating in the EPSRC sponsored *Science Innovation Programme* at the University of Oxford. The aim of the programme was to get research ideas out of the lab and into the next generation of technology innovation. However, after noticing the analogies between how technology entrepreneurs and security practitioners solve problems, I felt it necessary to get ideas on technology innovation out of practice and into the next generation of security research instead! I am grateful to Marc Ventresca and Victor Seidel at the Entrepreneurship Centre in the Säid Business School for their comments and insights on this work.

The on-going evolution of CAIRIS has benefitted from the generous assistance of the UK Defence Science and Technology Laboratory (Dstl). They provided invaluable feedback and comments on research resulting from the EPSRC-funded *Evaluating the Usability, Security, and Trustworthiness of Ad-hoc Collaborative Environments (EUSTACE)* project, which led to the work on trust modelling and trust expectations described in Sect. 12.1. More recently, they have supported on-going work on the use of CAIRIS to support risk assessment for systems of systems, which is briefly discussed in Sect. 12.1.

This book would not have been possible without the support of Bournemouth University (BU). BU has been generous in providing the time and resources needed to make this book a reality. Their support has ranged from pump-priming grants to kick start the work on CAIRIS shortly after I joined BU, matched funding for Ph.D. projects, support for research software engineers to support the development and maintenance of CAIRIS and general sympathy for the demands of writing a book while simultaneously managing multiple research and teaching commitments. My students at BU have also played an important role in this book by using IRIS and CAIRIS in exercises, assignments and, in several cases, undergraduate dissertations. Where things did not always go according to plan, they provided useful feedback and raised GitHub issues and, in several cases, pushed fixes to improve the stability and performance of the platform.

I would like to thank everyone who reviewed early drafts of this book for their suggestions for improving the presentation of this book, and smoothing away many rough edges: Andrea Atzeni, Benjamin Aziz, Huseyin Dogan, Michael Eaton, Jane Henriksen-Bulmer, Duncan Ki-Aries, Nancy Mead, Simon Parkin, Adam Shostack, Jacqui Taylor and Eylem Thron. I am particularly grateful to Andrew Simpson, whose comments and attention to detail went well beyond the call of duty.

On a personal note, I would like to thank my parents for their patience, support and love. Last, but by no means least, I would like to thank Claudia for her love, and for always being there for me.

Poole, UK Shamal Faily
February 2018

Contents

List of Figures

List of Tables