Jan Jürjens

Secure Systems Development with UML

Jan Jürjens

Secure Systems Development with UML

With 79 Figures

Springer

Jan Jürjens
Dep. of Informatics
Software and Systems Engineering
Technische Universität München
Boltzmannstr. 3
85748 München/Garching
e-mail: juerjens@in.tum.de

Library of Congress Control Number: 2004112217

ACM Computing Classification (1998): D.2.2, D.2.4

ISBN 3-540-00701-6 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Cover design: KünkelLopka, Heidelberg
Production: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Typesetting: by the Authors
Printed on acid-free paper 45/3142/YL - 5 4 3 2 1 0

to Li

Foreword

Those who spend their professional life developing and deploying – or observing – new information technology systems may believe that the security issues raised in this process are a direct consequence of developments in technology. This is not quite so. New technologies create opportunities for new applications, sometimes not even foreseen when the technology was first fielded. Pertinent examples are e-mail, the first major application of the Internet and its precursors (rather than the "serious" scientific collaborations originally intended), or the success of SMS (Short Messaging Service), which was initially perceived as a minor addition to the services offered by second-generation mobile telecommunications systems.

At the same time, new technologies and the applications they are facilitating also open up new opportunities for creating mischief of various hues, which in turn trigger a demand for "security technologies" that should prevent – or at least reduce – unwelcome use of those new applications. To stay with the example of e-mail, spam has today become a major nuisance, to the extent that some see success in the battle against spam as essential for e-mail to survive as a useful service.

For software systems, the release of the Internet for commercial use in the early 1990s was an incisive event, whose implications have still not been fully digested. It first led to the development of distributed applications in closed environments that use the Internet as an open communications network. In this domain, security requirements are mainly, but not exclusively, related to communications security. Virtual Private Networks may serve as an example. However, today we are also dealing with open environments without central points of control or authority, which require novel ways of approaching security. Indeed, the fact that in different applications fundamentally different security requirements have to be met is one of the reasons why the design of security protocols is difficult and error prone.

All of which brings us to security. Security professionals like to state that security must not be treated as an add-on feature, and that systems cannot be made secure by adding some so-called security features in the later design

stages. To the extent that security requirements depend on the application, it is then the task of the application designer to include those requirements in the specification early on, and the task of the design process to make sure that adequate protection mechanisms are implemented. There is thus an obvious demand for design methodologies that help in specifying security requirements, and in making sure that suitable security mechanisms are implemented.

To add a second general statement on security, there are hardly ever correct answers to security challenges, only answers that are better or worse than others. When proposing design methodologies for security, we are walking a tightrope if security-unaware application writers are asked to decide on matters of security. In application areas where security requirements are well understood and met by a fairly standardized set of security mechanisms, we may justifiably hope that such methodologies can be put to good use. However, particularly in novel kinds of applications, we will not always know the security requirements in advance, and prudent engineering practices may change over time. As an example, robustness against denial-of-service attacks and identity protection (plausible deniability) have become new aspects in the design of key establishment protocols in recent years, as witnessed in the discussions about a successor to the Internet Key Exchange protocol (IKE).

This book makes valuable contributions towards the development of well-founded design methodologies for security engineering. By building on a widely adopted specification language like UML, consideration of security aspects fits into the design process in a natural way. The proposed methodology has solid theoretical foundations so that it is possible to verify in a precise setting whether a design has its desired security properties. The definition of these foundations would in itself constitute a substantial piece of work, but the book goes further. For any design methodology striving to have practical impact, the proverbial saying that "the proof of the pudding is in the eating" applies. The book does not fall short on this count either, covering several case studies the methodology has been applied to, and presenting the tools that have been developed to support this approach.

To say that this book is a first step in a promising direction would thus seriously underrate what has already been achieved. The reader may treat this book as an exemplary demonstration of how formal methods for the design of secure systems could be made accessible to application software designers in general, and wait with interest for further developments as the methodology matures.

Hamburg, *Dieter Gollmann*
May 2004

Preface

Attacks against computer networks, which modern society and modern economies rely on for communication, finance, energy distribution, and transportation, can threaten the economical and physical well-being of people and organizations. Due to the increasing interconnection of systems, such attacks can be waged anonymously and from a safe distance. Thus networked computers need to be secure.

The high-quality development of security-critical systems is difficult. Many systems are developed, deployed, and used that do not satisfy their criticality requirements, sometimes giving rise to spectacular attacks.

Part of the difficulty of secure systems development is that the goal of correctness is often in conflict with that of low development cost. Where thorough methods of system design pose high cost through personnel training and use, they are all too often avoided.

The Unified Modeling Language (UML) offers an unprecedented opportunity for high-quality and cost- and time-efficient secure systems development:

- As the de facto standard in industrial modeling, a large number of developers are trained in UML.
- Compared to previous notations with a user community of comparable size, UML is relatively precisely defined.
- A variety of tools exist that provide the basic functionality required to use UML (such as the drawing of UML diagrams).

To exploit this opportunity, however, some challenges remain: One needs to adapt the UML to the application domain of security-critical systems and advance its correct use in this application domain. One has to develope advanced tool support for secure systems development with UML, such as automatic analysis of UML specifications with respect to security requirements. This requires dealing with conflicts between flexibility and unambiguity in the meaning of UML models. This book aims to contribute to overcoming these challenges.

We present the UML extension UMLsec for secure systems development, using the standard UML extension mechanisms. The possibility of a high degree of abstraction, and diagrams offering different views of a system, allow the modeling of security-critical components in the system context. One can thus automatically evaluate UML specifications for vulnerabilities using the UMLsec tool support based on a formal semantics of a simplified core of UML 1.5 which we also provide.[1] One may also encapsulate established rules of prudent security engineering and make them available to developers. Our method thus aims to be useful both to security experts and to developers who may not be experts in security. We demonstrate the adequacy of UMLsec by using it in several case studies. For example, we develop a secure channel specification and uncover flaws in a published variant of the Internet protocol TLS and in the Common Electronic Purse Specifications, propose corrections, and verify them. We use UMLsec in the context of banking applications and of Java security. We present the concepts and technologies needed for constructing tool support for analyzing UML models for sophisticated requirements, such as the constraints included in UMLsec specifications. The tool support is based on an XML dialect called XMI which allows interchange of UML models.

This book is based on a PhD thesis, several invited talks and summer school lectures, a series of about thirty tutorials at international conferences and feedback from many of their participants, and about thirty articles in international journals and at conferences by the author, as well as on feedback from projects with industrial partners (including a major German bank, car manufacturer, and telecommunications company), and on discussions at international workshops organized on the topic of model-based security engineering with UML, as well as the supervision of about thirty Master's and Bachelor theses and advanced study projects on related topics and seven courses given at the University of Oxford and TU München which included part of the topics covered in this book. Additional material is given on the website [Jür04] associated with this book which is continuously being updated. It includes the following material:

- Slides and audio recordings from the tutorials and courses mentioned above.
- Other learning and teaching materials, including exercises and answers.
- A web interface for a tool which analyzes UMLsec models written using an industrial UML modeling tool (which one can upload over the Internet) for security requirements.[2]

Note that although the UML extension proposed in this book aims to also offer assistance to developers who are not security experts (for example, by enabling them to use security mechanisms in a secure way), parts of the book are

[1] In the appendix, we explain how to adjust our approach to the upcoming version UML 2.0.

[2] The tool is currently being made available as open-source.

concerned with advanced applications (such as the analysis of cryptographic protocols) for which background knowledge in security would be helpful.

I would like to express my sincerest gratitude to all of the people involved in some way or another with the above undertakings, and with the compilation of this book in particular. These include my advisor for the PhD thesis on which this book is based, Samson Abramsky (for his insights and advice, encouragement, and patience), as well as Manfred Broy, head of my subsequent affiliation being the Software & Systems Engineering group at TU Munich (for interesting discussions, for sharing his profound experience in formal methods and software engineering, and for providing a very stimulating working environment), various people who provided encouragement to pursue the idea to write a book based on the thesis, my coauthors and colleagues (for fruitful collaborations, and inspiring discussions on security or UML), my students (for helpful collaborations on tool support and for questions on dubious parts of the material), several people reading various portions of the draft and offering useful comments and advice, as well as the many reviewers of the papers on which this book is based, altogether several hundred participants of my tutorials, as well as the audiences of my other talks related to security or UML, many of whom contributed comments and questions. I would also like to thank the members of different organizations in which I am involved (including the working group for Formal Methods and Software Engineering for Safety and Security (FoMSESS) within the German Society for Informatics (GI), the Division of Safety and Security within the GI, the Bavarian Competence Center for Safety and Security, the working group on e-Security of the Bavarian regional government, and the IFIP Working Group 1.7 "Theoretical Foundations of Security Analysis and Design") for interesting discussions about security, the technical support at TU Munich (including the system administrators and the student assistants, in particular Britta Liebscher). Last but not at all least I would like to thank my editor at Springer-Verlag, Ralf Gerstner, for his interest in the book project and his enduring and understanding patience. Apologizing to those who currently manage to escape my mind, I would like to name in particular the following: Martín Abadi, Lionel Van Aertryck, Ewgeny Alter, Axelle Apvrille, David Basin, Peter Braun, Matthias Braun, Ruth Breu, Alexander Chatzigeorgiou, Dominique Chauveau, Pierpaolo Degano, Martin Deubler, Rik Eshuis, Andreas Fedrizzi, Eduardo B. Fernandez, Robert France, Onno Garms, Geri Georg, Andreas Gilg, Dieter Gollmann, Roberto Gorrieri, Susanne Graf, Johannes Grünbauer, Joshua Guttman, Sebastian Höhn, Helia Hollmann, Siv Hilde Houmb, Anna Ioshpe, Gergely Kokavecz, Dimitri Kopjev, Thomas Kuhn, Simon Kulla, Markus Lehrhuber, Britta Liebscher, Wolfgang Linsmeier, Volkmar Lotz, Gavin Lowe, Frank Marschall, Shasha Meng, Carlo Montangero, Haris Mouratidis, Gerhard Popp, Max Raith, Jan Romberg, Bernhard Rumpe, Robert Sandner, Robert Schmidt, Marillyn Schwaiger, Stephan Schwarzmüller, Bran Selic, Pasha Shabalin, Shunwei Shen, Oscar Slotosch, Perdita Stevens, Martin Strecker, Guido Wimmel, and Bo Zhang.

Finally, I particularly thank my parents and my brother for their continued moral support.

This work was financially supported in part by the Studienstiftung des deutschen Volkes, Laboratory for Foundations of Computer Science (University of Edinburgh), Bell Laboratories (Lucent Technologies), Computing Laboratory (University of Oxford), Software & Systems Engineering (TU Munich), the FairPay project (German Ministry of Economy), the HypoVereinsbank (Munich), and the Verisoft project (German Ministry of Education and Research). The support is greatly appreciated.

Comments or questions regarding the content of this book are always welcome, and can be made through the book's website [Jür04].

München, *Jan Jürjens*
August 2004

Two roads diverged in a wood, and I,
I took the one less traveled by,
And that has made all the difference.

Robert Frost, The Road Not Taken

Contents

Part II Developing Secure Systems

Part III Tool Support

List of Figures

Part I

Prologue