Preface

The security of software systems in recent years has been transformed from a mono-dimensional technical challenge to a multi-dimensional technico-social challenge, due to the wide usage of software systems in almost every area of the human life. This situation requires a different and more holistic approach to the development of secure software systems. Recent research argues that it is essential for security to be considered from the early stages and throughout the software development life-cycle; thus, sound software engineering methodologies and practices need to be developed that support the simultaneous analysis of both security and software requirements, their transformation to an appropriate design and the implementation of that design. Towards this direction, a number of relevant challenges have been identified[1]ranging from the development of appropriate security requirements techniques to security modelling languages to secure code analysis. On the other hand, a number of research-led and industrial-led projects have been presented in the literature aiming to provide some answers to these challenges and successfully integrate security considerations during the development of software systems starting from the early stages of the development process. This area of researh and development, widely termed in the literature as secure software engineering[2], is currently very active and fast moving.

This book aims to capture the essential elements of this area and provide a forum for presenting the most recent and innovative lines of research and industrial practice related to secure software engineering. The book provides coverage of recent advances in the area of secure software engineering that address the various stages of the development process from requirements to design to testing to implementation. The contributions of this book are three-fold: it provides a comprehensive understanding of the curent state of the art in the area of secure software engineering; it inspires and motivates further research and development; and it bridges the gap between academic research and industrial practice.

BOOK AUDIENCE

The book is addressed to a wide range of industrial and professional audiences from project managers to security engineers and software developers; and anyone else in an industrial context who is involved with any aspects of secure software systems. The book is also addressed to researchers who are involved in creating the future theories, methods, methodologies and tools for secure software engineering. Finally, the book is addressed to university lecturers and professors developing programmes of studies in secure software systems engineering and their students (especially at master level).

ORGANISATION OF THE BOOK

The book is organised into twelve (12) chapters. The first chapter provides an overview of the state-of-practice in the area of secure software and it presents a synthesis of expert views on some important actions needed to improve the state of practice in secure software. The authors base their study on experiences as panel moderators, rapporteurs and report writers involved in drafting the views of experts. The rest of the chapters are organised into four (4) sections.

Section 1 is on *Security Patterns* and it includes three (3) chapters. Together, these three chapters form a comprehensive introduction to security patterns for the novice reader but also give novel insights about recent research to the expert reader. The concept of security patterns has achieved prominence as an important vehicle for sharing and reusing security knowledge among developers, accessible even to those with limited security expertise assisting them in the construction of secure software systems. Chapter 2 (*Using Security Patterns to Develop Secure Systems*) by Fernandez et. al. provides an overview of how security patterns can be used in various development stages from analysis to design to testing. Chapter 3 (*A Pattern-Based Method to Develop Secure Software*) by Schmidt et. al., presents a security engineering process based on security problem frames and concretized security problem frames. The presented process is supported by formal models that are used to prove that the solution approaches are correct solutions to specified security problems. Chapter 4 (*Security Patterns: Comparing Modeling Approaches*) by Nhlabatsi et. al. presents a comparative analysis and evaluation of a number of secure software systems approaches, by examining the extent to which their constructs can support the use of security patterns as part of the analysis of security problems.

Section 2 is on *Methodologies and Frameworks* and it includes 3 chapters. Chapter 5 (*Security over the Information Systems Development Cycle*) by Blanco et. al., discusses the problem of integrating security into the software development process, paying more attention to the requirements engineering discipline and the software design stage. They present their efforts to integrate security considerations into the software systems development process in various domains such as software product lines, business processes, web services, and databases and data warehouses. Chapter 6 (*Balancing Security and Performance Properties During System Architectural Design*) by Houmb et. al., presents the Aspect-Oriented Risk Driven Development (AORDD) methodology, which integrates the analysis of two quality properties, namely security and performance, into the development process of critical systems. The approach is illustrated using a transactional web e-commerce benchmark (TPC-W) originally developed by the Transaction Processing Performance Council. Chapter 7 (*State Model Diagrams – a universal, model driven method for network system configuration and management*) by Maj presents an approach concerned with the configuration and management of network devices. In particular, the paper demonstrates how the State Model Diagram method is useful for the configuration and management of complex security protocols and devices.

Section 3 is on *Privacy and Trust*. Two topics very related to security itself. Privacy sometimes is considered as a sub-factor of security, while other times the two might also be seen as opposites, since security mandates the recording of information (e.g., users' details) whereas privacy might motivate anonymity. On the other hand, trust and security are also closely related. This is the case for a number of reasons. For example, security always assumes some degree of trust in its mechanisms. Consider, for instance, a software system that is based on passwords to provide access to an account. The software engineers may have assumed that each user is trustworthy and will not make their passwords freely available to potential attackers of the system. Further, the system and its administrator are assumed to

be trusted not to divulge, alter or remove passwords. However, it is only when such trustworthiness is demonstrated, that the security can properly assumed. Chapter 8 (*Designing Privacy Aware Information Systems*) by Kalloniatis et. al. identifies a number of privacy requirements that should be considered during system analysis and design. The authors also present and analyse 10 methods from the area of secure requirements engineering. They then compare these methods based on their initial set of privacy requirements. Chapter 9 (*Privacy aware systems - from models to patterns*) by Coen-Porisini et. al., presents work on the development of a conceptual model to support the definition of privacy policies. The presented model introduces a set of concepts concerning privacy and defines the existent relationships among those concepts along with the interfaces for the definition of privacy related mechanisms. An assessment of the model is presented with the aid of an example from the health care domain. Chapter 10 (*Incorporating social trust into design practices for secure systems*) by Cofta et. al., discusses how the "designing for trust" paradigm leverages trust governance into the design practices of ICT systems developers by complementing security-based methodologies. In particular, they argue for the need to consider trust as part of the software systems development process; they present three different (but complimentary) views of the notion of trust and they discuss how trust governance and security management can benefit from integration.

Section 4 is on *Secure Code Analysis*. There is a large collection of well established analysis techniques and recent research developments, and the two chapters in this section complement the existing literature. Chapter 11 (*Static program analysis of multi-applet JavaCard applications*) by Loizidis et. al., investigates recent advances in theory and tool support for static program analysis of security critical applications. Based on their investigation the authors present an approach for automatic verification of critical application based on the domain of smart cards. Chapter 12 (*Automatic Timed Automata Extraction from Ladder Programs for Model-Based Analysis of Control Systems*) by Vasconcelos Oliveira et. al., presents a method to increase the confidence in the behaviour of critical control systems. The presented method automatically generates the timed automata models from the specification ISA 5.2 Binary Logic Diagrams, and the implementation Ladder programs, for model-based analysis. The method is based on the use of the Uppaal tool and the Uppaal-TRON testing tool.

Haralambos Mouratidis
University of East London, UK

ENDNOTES

1 H. Mouratidis and P. Giorgini (2006), Integrating Security and Software Engineering: Advances and Future Visions, Idea Group Publishing, pages 290.

2 The term secure software engineering is defined by Mouratidis and Giogini (see footnote 1 above for full reference details) as a branch of research investigating the integration of security concerns into software engineering practices, which draws from expertise from the security and softtware engineering community. It is thought of as an umbrella term under which the areas of security requirements engineering, security modelling and secure sofware development lie. It is worth noting that there are also alternative names used in the literature, which refer to the same area of research and development, such as software security engineering, software engineering for security, software engineering for secure systems.