Software security is about protecting information and ensuring that systems continue to function correctly even when under malicious attack. The traditional approach of securing a system has been to create defensive walls such as intrusion detection systems and firewalls around it, but there are always cracks in these walls, and thus such measures are no longer sufficient by themselves. We need to be able to build better, more robust and inherently more secure systems, and we should strive to achieve these qualities in all software systems, not just in the ones that “obviously” need special protection.

The first Workshop on Secure Software Engineering (SecSE) was held in Vienna in 2007, focusing on techniques, experiences and lessons learned for engineering secure and dependable software. Since then we have organized three additional successful workshops in Barcelona, Fukuoka and Krakow, each serving as a forum for fruitful discussions and exchange of ideas between academia and industry. From 2008, these workshops have been organized to support the main goal of the SHIELDs project (funded by the European Community Seventh Framework Programme, grant agreement no 215995), namely to “bridge the gap between security experts and software developers”.

This Special Issue contains revised and extended versions of the top five papers at the 4th International Workshop on Secure Software Engineering (SecSE 2010), which was part of the Fifth International Conference on Availability, Reliability, and Security (ARES 2010) held 15-18 February 2010 in Krakow, Poland.

The papers in this special issue have all gone through an additional review by at least 3 international experts, and represent a significant extension of the workshop contributions. The papers cover many important facets of the software security field, spanning hot patching, buffer overflows, configuration fuzzing, usable security requirements engineering, and ending up with a critical look at agile software engineering methods from a software security perspective.

Bratus, Oakley, Ramaswamy, Smith and Locasto propose a solution to how mission-critical software can be updated without stopping the process or rebooting in “Katana: Towards Patching as a Runtime Part of the Standard Compiler-Linker-Loader Toolchain”. Shahriar and Zulkernine present a survey of approaches to counter buffer overflows in “Monitoring Buffer Overflow Attacks: A Perennial Task”. Dai, Murphy, and Kaiser introduce a novel approach to testing multiple configurations of complex software systems in “CONFU: Configuration Fuzzing Testing Framework for Software Vulnerability Detection”. Faily and Flechais explore the borderline between usability and security during requirements analysis in “Towards tool-
support for Usable Secure Requirements Engineering with CAIRIS”. Finally, Nicolaysen, Sassoon, Line, and Jaatun present results from interviews of agile developers contrasted with experiences from a case study in their article “Agile Software Development: The Straight and Narrow Path to Secure Software?”

We thank Editor-in-chief Khaled M. Khan for inviting us as guest editors, and the following reviewers for their efforts in improving the quality of this special issue.

REVIEWERS OF THIS SPECIAL ISSUE

Ruben Alonso, Visual Tools
Bart De Win, KU Leuven
Ivan Fléchais, University of Oxford
Khaled M. Khan, Qatar University
Per Håkon Meland, SINTEF
Khalid Azim Mughal, University of Bergen
Holger Peine, FH Hannover
Lillian Rostad, NTNU
Chunming Rong, University of Stavanger
Christoph Schuba, Sun Microsystems, Inc
Nahid Shahmehri, Linköpings Universitet
Panagiotis Trimintzios, ENISA
George Yee, Carleton University
Mohammad Zulkernine, Queens University

We will celebrate the fifth anniversary of SecSE by returning to historic Vienna, Austria, and we cordially invite you to submit your contributions. We look forward to many interesting discussions of state-of-the-art research and practice with the secure software engineering community!

Martin Gilje Jaatun
Per Håkon Meland
Guest Editors
IJSSE