information

Editorial

Introduction to the Special Issue on Evaluating the Security of Complex Systems

Eduardo B. Fernandez

Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA; fernande@fau.edu; Tel.: +1-561-297-3466

Academic Editor: Willy Susilo
Received: 8 July 2016; Accepted: 9 July 2016; Published: 18 July 2016

Abstract: Recent security breaches show the need to secure large, distributed, complex systems. A fundamental, but little discussed aspect of security is how to evaluate when a complete system is secure. Purely formal methods cannot handle this level of complexity. Code checking does not consider the interaction of separate modules working together and is hard to scale. Model-based approaches, such as patterns and problem frames, can be effective for handling large systems. Their use in evaluating security appears promising. A few works in this direction exist, but there is a need for more ideas. This Special Issue focuses on global, model-based, architectural, and systems-oriented evaluation methods.

Keywords: security evaluation; systems security; reference architectures; security patterns; privacy

Patterns have a big potential to build secure systems, which have not yet been exploited in industry. There are several reasons for this but one of the most important is that there are no evaluations of the security of the systems built using security patterns. We have found very few papers attempting to evaluate security, not just in systems using patterns, but on any other type of system [1–4]. This is especially paradoxical for systems built with security patterns since, as we discussed in our previous works [2,5], patterns provide a reference to perform this evaluation by analyzing if the patterns provide a way to control the identified threats or as a basis for proofs. Our intention was also to identify if approaches using other artifacts could lead to some type of evaluation. We invited 12 colleagues of whom we had a high opinion and who all work on similar subjects, and we were able to convince five of them to submit papers, which were carefully reviewed and revised before being approved for publication. We hope these papers give an enlightening view of some of the current issues of security evaluation.

The five papers in this issue are related in a rather loose way to the subject of security evaluation, focusing on different aspects and using, not only patterns, but other related artifacts. We did not try to enforce strict compliance with the call for papers, as we thought that as far as their general objectives were consistent with our objectives, it was worthwhile to present their ideas.

The paper by Heckman and Schell [6] leverages patterns to design systems that can be proven to be secure. As a basic pattern they use a Reference Monitor, the implementation of which is a multilevel model security kernel. Roger Schell is one of the pioneers of the concept of the security kernel, which he has applied to a variety of systems and the paper summarizes his experience. Heckman and Schell's paper emphasizes a fundamental point: we need secure designs, not just secure implementations. A strong design can assure the overall security of a system, even if some parts may be compromised [7]. Patterns are a way to enforce good designs, potentially allowing proofs or at least analysis of security properties. The paper shows clearly how the application of principles through patterns can stop subversion attacks, which are impossible to control through good coding and patching.

Yoshizawa et al. are concerned with implementation aspects of security patterns [8]. They indicate that developers may apply the patterns in ways that introduce vulnerabilities and propose a test template to determine if an applied pattern has introduced vulnerabilities. Their approach also helps correct vulnerabilities that are found.

Zalewski et al. [9] consider measuring security in cyberphysical systems. They discuss the measurement process and present five approaches to perform this evaluation, including one using security patterns. Cyberphysical systems are used in a variety of applications, including electricity generation, transportation, medical systems, and many others. These are often critical applications where security is of fundamental importance but there is relatively little work on evaluating their security.

With vehicles having more and more computational power, their security is becoming critical and manufacturers are worried about making their products secure which in this case implies also safety; take, for example, a security attack that can make a car go off of a road. The paper by Beckers et al. [10] considers how to evaluate the security of vehicles by extending security standards to the automotive realm.

For some applications, security is not the only or the most important requirement, but their main goal is protecting the privacy of their users. In this case, instead of evaluating security we need to evaluate privacy, which although very close, has its own constraints. Meis and Heisel describe a semi-automatic way to identify privacy requirements in an application [11], using a health application as a case study. The identified requirements can then be used to evaluate the final implementation by looking for inconsistencies between privacy requirements or between privacy requirements and the domain model.

A *pattern* is a solution to a recurring problem in a specific context [12]. A pattern embodies the knowledge and experience of software developers and can be reused in new applications; carefully-designed patterns implicitly apply good design principles. Patterns are also good for communication between designers and to evaluate and reengineer existing systems. While initially developed for software, patterns can describe hardware, physical entities, and combinations of these. *Security patterns* provide solutions to security problems, usually on how to stop a threat [5]. *Aspect-oriented software* development focuses on the identification, specification, and representation of cross-cutting concerns in software units and their modularization into separate functional units as well as their automated composition [13]. In general, aspects are code oriented while patterns are model oriented but there are model-oriented aspects. *Problem frames* are a requirements engineering approach proposed by Jackson [14]. The system-to-be (called machine) and its interfaces to the environment, which consists of domains, are represented in a context diagram; this method is used in [11].

We acknowledge the contributions of the MDPI editors, as well as the paper reviewers.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Breu, R.; Innerhofer-Oberperfler, F.; Yautsiukhin, A. Quantitative assessment of enterprise security system. In Proceedings of the Third International Conference on Availability, Security and Reliability (ARES), Barcelona, Spain, 4–7 March 2008; pp. 921–928.
2. Fernandez, E.B.; Yoshioka, N.; Washizaki, H.; VanHilst, M. Measuring the level of security introduced by security patterns. In Proceedings of the 4th Workshop on Secure Systems Methodologies Using Patterns (SPattern 2010), in Conjunction with ARES 2010, Krakow, Poland, 15–18 February 2010.
3. Halkidis, S.T.; Tsantalkis, N.; Chatzigeorgiu, A.; Stephanides, G. Architectural risk analysis of software systems based on security patterns. *IEEE Trans. Depend. Secure Comput.* **2008**, *5*, 129–142. [CrossRef]
4. Heyman, T.; Scandariato, R.; Huygens, C.; Joosen, W. Using security patterns to combine security metrics. In Proceedings of the Third International Conference on Availability, Security and Reliability (ARES), Barcelona, Spain, 4–7 March 2008; pp. 1156–1163.

5. Fernandez, E.B. *Security Patterns in Practice: Building Secure Architectures Using Software Patterns*; Wiley: Chichester, UK, 2013.
6. Heckman, M.R.; Schell, R.R. Using Proven Reference Monitor Patterns for Security Evaluation. *Information* **2016**, *7*, 23. [CrossRef]
7. Fernandez, E.B.; Yoshioka, N.; Washizaki, H.; VanHilst, M. An approach to model-based development of secure and reliable systems. In Proceedings of the Sixth International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, 22–26 August 2011.
8. Yoshizawa, M.; Washizaki, H.; Fukazawa, Y.; Okubo, T.; Kaiya, H.; Yoshioka, N. Implementation Support of Security Design Patterns Using Test Templates. *Information* **2016**, *7*, 34. [CrossRef]
9. Zalewski, J.; Buckley, I.A.; Czejdo, B.; Drager, S.; Kornecki, A.J.; Subramanian, N. A Framework for Measuring Security as a System Property in Cyberphysical Systems. *Information* **2016**, *7*, 33. [CrossRef]
10. Beckers, K.; Dürrwang, J.; Holling, D. Standard Compliant Hazard and Threat Analysis for the Automotive Domain. *Information* **2016**, *7*, 36. [CrossRef]
11. Meis, R.; Heisel, M. Computer-Aided Identification and Validation of Privacy Requirements. *Information* **2016**, *7*, 28. [CrossRef]
12. Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerland, P.; Stal, M. *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*; Wiley: Chichester, UK, 1996.
13. Wikipedia: Aspect-oriented software development. Available online: https://en.wikipedia.org/wiki/Aspect-oriented_software_development (accessed on 9 July 2016).
14. Jackson, M. *Problem Frames: Analyzing and Structuring Software Development Problems*; Addison-Wesley: Boston, MA, USA, 2001.