Software Factories for Security-by-Design

PhD Subject Proposal – Labex UCN@Sophia – January 2014
Topic: security and privacy protection

Thesis supervision:

- **Main supervisor: Yves Roudier** (**EURECOM**)
- **Philippe Collet** (**I3S**)
- Please note that, even though Yves Roudier does not yet hold an HDR, he undertakes to defend it within the current year (2014). This will be formally confirmed when required.

Context

Software is pervading most systems nowadays, from traditional IT systems to embedded systems or even ubiquitous computing artifacts. The increasing importance of software is however hampered by the multiplication of security vulnerabilities found in software that reveal the poor understanding of secure software engineering by business application developers. Programming mistakes that endanger the safe execution of the application and that might be exploited by attackers are well known, and consist for instance in buffer overflows, argument parsing issues, etc. It seems that either the lack of education of business programmers with respect to secure programming or the complexity of software prevents these problems from being solved without some form of automation or at least assistance. It is also difficult for business application developers to understand the security and privacy requirements that they have to address. Similarly, security experts that mandate such requirements have a hard time pushing them to the design and implementation stages of classical software engineering methodologies. Such interaction problems generally result in security design flaws in the software produced. For instance, security experts might ignore the use of a middleware layer by application developers, and will consider authentication at the network level, but not at the middleware level, thus allowing spoofing attacks on middleware commands.

This PhD work is thus proposed in the context of software engineering for security, and focuses on a security-by-design approach. We claim that in order to take advantage of the progress made in the past 20 years in software engineering, this field has to incorporate security concerns early on in the design of software-based systems. Conversely, the integration of security issues can only be successful in the future if the gap between business application programmers and security experts can be somehow bridged using adapted design and programming tools that integrate with software engineering concerns like reuse, modularity, agility, etc.

Challenges

Facing the increasing size and complexity of developed software systems requires to systematize the collaboration about the intricate relationships of security and business concerns. Security experts especially work either upstream, based on the verification of an abstraction of the system of which they will rarely see the implementation in software components, or downstream, with security testing techniques that aim at validating *a posteriori* the implementation of security mechanism within the application code. It must also be noted that the security of an application is still often considered after its design, in the light of many successful attacks.

The different phases of the lifecycle of secure software have already been improved with different approaches being proposed, but they are all focused on separate phases and no holistic approach has been undertaken so far. For instance, the security requirements engineering phase is generally disconnected from the development of software components, and even more from their security testing activities. While the generation of secure code or of security policies for configuring security mechanisms is nothing new, agility has not really been considered in secure software design so far. Yet agility is essential in terms of software engineering for the incremental development and refinement of security and privacy requirements and the implementation of the security mechanisms satisfying them. This is especially true when new threats become known, in order to understand to what extent security properties are still valid. This is also due to the once-and-for-all approach prevalent in the secure-by-design paradigm, which is essentially dealt with using a formal verification of the software design. So far, it is only the advent of concolic testing [SMA05] that has started mapping design with the latter phase of security testing, but it mostly ignores any feedback to the initial phase expression of security requirements, especially in complex component-based software.

Approach and Expected Results

The PhD will aim at providing foundations, techniques, and tooling to support software factories dedicated to the development of secure-by-design software with security and privacy requirements being the artefacts of prime importance. To meet these objectives, the research work will rely on previous of both concerned research groups, resp. MODALIS/I3S on SPL engineering and NSTEAM/EURECOM on security by design.

The work will extend the SysML-Sec framework [RIA13, ApRo13a, ApRo13b] for security requirements engineering with the refactoring capabilities provided by the SPL approach [ACL+13, UMB+13]. Software Product Lines (SPL) engineering is a recent approach in the software engineering field to effectively relate requirements in a domain to the development of software artefacts. The SPL paradigm is a relevant area of research that aims at addressing the challenges of complexity and variability in software development by producing a family of related software variants, called an SPL, for a given domain. It is then necessary to design and maintain the appropriate models, languages and techniques to produce multiple but similar software products. In different domains, such as the mobile (Nokia, Sony-Ericsson, etc.), automotive (General Motors) or avionics (Boeing) industries, it has already allowed companies to reduce development cost and time-to-market, and to increase the quality of software. In this context a software factory is a SPL that configures extensive tools, processes, and content using a template to automate the development and maintenance of the variants generally based on framework-based components. This approach has however, to our knowledge, never been applied to secure programming problems.

More specifically, the thesis will first explore whether the existing algorithms are adapted to representing security concerns as "features" on the one hand, and to guiding the design of the software architecture based on security best practices on the other hand. One important question relates to security requirements expressivity: how to capture complex combinations of elementary security properties over composed components? How to relate these to patterns expressing a potential vulnerability and an associated best practice to address it? How to select specific mechanisms, like bit commitments or computations with encrypted polynomials, based on the architecture and abstract requirements? This work will be guided by security use cases where software adopts a service oriented architecture. For instance, we intend to take advantage of a use case where we studied the introduction of security mechanisms based on the Oauth framework into multi-party interactions [CDR+13]. An inline reference monitor use case will also be used [ISR+12] as well as different use cases about parameter parsing safety and middleware-based security [Ser13]. We also intend to introduce risk analysis as a central element of a security-enabled SPL, in order to be able to use the modelling of attacks and attackers as a central element for the selection and correct configuration of appropriate components of an existing base library.

The work will also focus on code generation issues. The thesis will address the need for agility and for feedback from security testing through the introduction of requirements traceability mechanisms throughout the software engineering process, as well as through the design of threat coverage tools that will assess the impact of software vulnerabilities or the capabilities of specific attackers over software components based on the selection of a given variant in the SPL. Furthermore, we plan to increase the abstraction and hence reusability for the security expert through the use of behavior skeletons related to various aspects of security mechanisms (monitoring, filtering, encryption). The use of such skeletons to generate security mechanisms will be generalized, especially through new aspect-oriented pointcuts using or extending the proposals made in [Ser13] and [CSC13].

We finally expect that this thesis will produce a software engineering framework for assisting joint work over the specification, design, development, and testing of a secure software by business application developers and security experts. One of the challenges here is to produce intelligible security abstractions for guiding developers in the tasks that will not be automated. Another challenge related to such a framework will be to provide expressive enough, yet still simple abstractions for secure code generation and testing. The framework might for instance get some inspiration from other model-driven engineering frameworks like the Papyrus, AADL, or EADL environments, and to adopt an domain-specific language to describe how to map security properties and mechanisms to components. The HiPOLDS language [DSI+12], which we designed, also suggests another approach that would be more specific to software engineering for information flow security. An appropriate methodology should also be defined to take full advantage of this framework and to guide developers.

Short Bios

Ass. Prof. Yves Roudier

is a member of the Network & Security Department of EURECOM, and more precisely the NSTEAM group, whose aim is to identify security and privacy requirements raised by emerging areas in communication and computer systems. Yves Roudier focuses on the design of secure software and cryptographic protocols for service-oriented architectures, as for instance found in cloud computing systems, and for automotive embedded software. As to secure software, his interests relate to security-by-design. His team has made contributions about security requirements engineering [RIA13, ApRo13a, ApRo13b, Idr12], model-driven engineering, aspect-oriented programming, and in general programming issues for the implementation of security mechanisms [Ser13, CDR+13, DSI+12, ISR+12, MoRo00] as well as software verification of cryptographic protocols [PIA+11]. He has co-authored more than 50 journal, conference, or workshop papers and has taken part in many national or european research projects.

Prof. Philippe Collet

is a member of the MODALIS research group of the I3S laboratory. The MODALIS team notably focuses on the definition of model-driven approaches supporting the definition of large-scale distributed systems, and variability modelling is an area of expertise since 2008. The team has developed the FAMILIAR DSL to handle large scale feature models [ACL+13] and has successfully applied innovative SPL techniques in different contexts (*e.g.*, video surveillance, scientific workflows, architectural models) [ACL+11,ACC+13,ACG+12]. Research interests of Philippe Collet range from software product lines engineering (composition of feature models, relationships to software architectures, reverse engineering of variability models, domain-driven multiple software product lines [UMB+13]) to Model-Driven Engineering (separation of concerns, lightweight DSL engineering, application to self-adaptiveness for large scale distributed systems). He has been involved in many industrial projects and has led the ANR project SALTY from 2009 to 2013. In January 2014, Philippe Collet is general chair of the VaMoS'2014 workshop, the reference event on variability modeling [CWW14]. He also served as PC chair of VaMoS'2013 [GCK13].

References

[ACC+13] Mathieu Acher, Anthony Cleve, Philippe Collet, Philippe Merle, Laurence Duchien, Philippe Lahire. *"Extraction and Evolution of Architectural Variability Models in Plugin-based Systems"* in Software and Systems Modeling, Springer, 2013

[ACL+13] Mathieu Acher, Philippe Collet, Philippe Lahire, Robert France. "FAMILIAR: A Domain-Specific Language for Large Scale Management of Feature Models" (special issue: programming languages in Science of Computer Programming (SCP), 78 (6), pages 657-681, Elsevier, June 2013, 0167-642

[ACG+12] Mathieu Acher, Philippe Collet, Alban Gaignard, Philippe Lahire, Johan Montagnat, Robert France. *"Composing Multiple Variability Artifacts to Assemble Coherent Workflows"* (special issue: Quality Engineering for Software Product Lines) in Software Quality Journal, 20 (3-4), pages 689-734, Springer, September 2012

[ACL+11] Mathieu Acher, Philippe Collet, Philippe Lahire, Sabine Moisan, Jean-Paul Rigault. *"Modeling Variability from Requirements to Runtime"* in Proceedings of the 16th International Conference on Engineering of Complex Computer Systems (ICECCS'11), pages 77-86, IEEE, Las Vegas, 27-29 April 2011

[ApRo13a] Ludovic Apvrille, Yves Roudier. SysML-sec: A sysML environment for the design and development of secure embedded systems. In Proceedings of APCOSEC 2013, Asia-Pacific Council on Systems Engineering Conference, September 8-11, 2013, Yokohama, Japan

[ApRo13b] Ludovic Apvrille, Yves Roudier. SysML-Sec: A model-driven environment for developing secure embedded systems. In Proceedings of SAR-SSI 2013, 8ème Conférence sur la Sécurité des Architectures Réseaux et des Systèmes d'Information, 16-18 Septembre 2013, Mont-de-Marsan, France

[CDR+13] Ronan-Alexandre Cherrueau, Rémi Douence, Jean-Claude Royer, Mario Südholt, Anderson Santana De Oliveira, Yves Roudier, Matteo Dell'Amico. Reference monitors for security and interoperability in OAuth 2.0. In Proceedings of SETOP 2013, 6th International Workshop on Autonomous and Spontaneous Security, 12-13 September 2013, Rhul, Egham, UK

[CSC13] R.-A. Cherrueau, M. Südholt, O. Chebaro. Adapting workflows using generic schemas: application to the security of business processes. The 5th IEEE International Conference on Cloud Technology and Science (CloudCom'13), Security and Privacy track, Dec. 2013.

[CWW14] Philippe Collet, Andrzej Wasowski, Thorsten Weyer, editors. Proceedings of the Eighth International Workshop on Variability Modelling of Software-Intensive Systems (VaMoS 2014) ACM, Nice, France, 22-24 January 2014, 978-1-4503-2556-1

[DSI+12] Matteo Dell'Amico, Gabriel Serme, Muhammad Sabir Idrees, Anderson Santana de Oliveira, Yves Roudier. HiPoLDS: A hierarchical security policy language for distributed systems. Information Security Technical Report, ISSN: 1363-4127. December 2012.

[GCK13] Stefania Gnesi, Philippe Collet, Klaus Schmid, editors. Proceedings of the Seventh International Workshop on Variability Modelling of Software-Intensive Systems (VaMoS 2013) ACM, Pisa, Italy, 23-25 January 2013, 978-1-4503-1541-8

[Idr12] Muhammad Sabir Idrees. A requirements engineering driven approach to security architecture design for distributed embedded systems. PhD Thesis. September 2012.

[ISR+12] Muhammad Sabir Idrees, Gabriel Serme, Yves Roudier, Anderson Santana De Oliveira, Hrevé Grall, Mario Sudholt. Evolving security requirements in multi-layered Service-Oriented-Architectures. SETOP 2011, 4th International Workshop on Autonomous and Spontaneous Security, in conjunction with the 16th annual European research event in Computer Security (ESORICS 2011) symposium, September 15-16, 2011, Leuven, Belgium / Also published in "Lecture Notes in Computer Science", 2012, Volume 7122/2012

[MoRo00] Refik Molva, Yves Roudier. A distributed access control model for Java. ESORICS 2000, European Symposium On Research In Computer Security, 4-6 Octobre 2000, Toulouse, France / Also published as LNCS, Volume 1895/2000

[PIA+11] Gabriel Pedroza, Muhammad Sabir Idrees, Ludovic Apvrille, Yves Roudier. A formal methodology applied to secure over-the-air automotive applications. VTC-Fall2011, IEEE 74th Vehicular Technology Conference, 5-8 September 2011, San Francisco, USA.

[RIA13] Yves Roudier, Muhammad Sabir Idrees, Ludovic Apvrille. Towards the model-driven engineering of security requirements for embedded systems. MODRE 2013, International Workshop on Model-Driven Requirements Engineering, 15 July 2013, Rio de Janeiro, Brazil.

[Ser13] Gabriel Serme. Modularization of security software engineering in distributed systems. PhD Thesis, Telecom ParisTech. November 10th, 2013.

[SMA05] Koushik Sen, Darko Marinov, Gul Agha (2005). CUTE: a concolic unit testing engine for C. Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering. New York, NY: ACM. pp. 263–272. ISBN 1-59593-014-0.

[UMB+13] Simon Urli, Sébastien Mosser, Mireille Blay-Fornarino, Philippe Collet. "How to Exploit

Domain Knowledge in Multiple Software Product Lines?" in Proceedings of the Fourth International Workshop on Product LinE Approaches in Software Engineering at ICSE 2013 (PLEASE 2013), ACM, pages 4, San Francisco, USA, 20 May 2013