Chapter 4

A Security Requirements Engineering Tool for Domain Engineering in Software Product Lines

Jesús Rodríguez
University of Castilla – La Mancha, Spain

Eduardo Fernández-Medina
University of Castilla – La Mancha, Spain

Mario Piattini
University of Castilla – La Mancha, Spain

Daniel Mellado
National Competition Commission, Spain

ABSTRACT

The concepts of Service-Oriented Architectures and Software Product Lines are currently being paid a considerable amount of attention, both in research and in practice. Both disciplines promise to make the development of flexible, cost-effective software systems possible and to support high levels of reuse, and may sometimes be complementary to each other. In both paradigms, security is a critical issue, although most of the existing product line practices do not comprise all the security requirements engineering activities or provide automated support through which to perform these activities, despite the fact that it is widely accepted that the application of any requirements engineering process or methodology is much more difficult without a CARE (Computer-Aided Requirements Engineering) tool, since it must be performed manually. Therefore, this chapter shall present a tool denominated as SREPPLineTool, which provides automated support through which to facilitate the application of the security quality requirements engineering process for software product lines, SREPPLine. SREPPLineTool simplifies the management of security requirements in product lines by providing us with a guided, systematic and intuitive manner in which to deal with them from the early stages of product line development, thus

DOI: 10.4018/978-1-60566-794-2.ch004
simplifying the management and the visualization of artefact variability and traceability links and the integration of security standards, along with the management of the security reference model proposed by SREPPLine.

INTRODUCTION

As software complexity grows and customers demand higher and higher quality software, non functional requirements can no longer be considered to be of secondary importance (Cysneiros & Yu, 2005). The tendency towards larger systems that are distributed over the Internet, such as those based on SOA (Service Oriented Architecture), has simultaneously introduced many new security threats (Opdahl & Sindre, 2008). Present-day information systems are therefore vulnerable to a host of threats and cyber-attackers, such as malicious hackers, code writers, cyber-terrorists, etc. (Choo et al., 2007).

Furthermore, there is currently an increase both in the demand for and the complexity of the software needed. Thus, in order to obtain high-quality systems along with higher productivity, software product line (SPL) based development has become the most successful approach in the reuse field, since it can help to significantly reduce both time-to-market and development costs (Bosh, 2000; Clements & Northrop, 2002), by increasing the reuse of all types of artefacts, thanks to the combination of coarse-grained components with a top-down systematic approach in which software components are integrated into a high-level structure. Moreover, existing SOA techniques can be used as an infrastructure on which increasingly complex software product line systems can be built with the aim of facilitating the emergence of a concurrent market in which atomic products from supplier product lines can be automatically integrated into a larger product line (Trujillo et al., 2008). This scenario, in which a product line consumes products that are supplied from third-party product lines, is called Service-Oriented Product Line (SOPL), an example of which might be web portals of portlets (Trujillo et al., 2008).

The complexity and the extensive nature of the SPL also make security and requirements engineering much more important for SPL based development, (particularly if this is a SOPL), than for the development of the information system since a security breach or vulnerability on the line can cause major long-term problems both to all SPL products and to third-party SPLs if the product in question was a SOPL.

Nevertheless, despite the fact that many requirements engineering practices must be appropriately tailored to the specific demands of product lines (Birk & Heller, 2007), SPL engineering software engineering methodologies and standard proposals have traditionally ignored security requirements and security variability issues. Thus, although several works dealing with security requirements management tools, similar to SREPPLineTool exist, none of them are either sufficiently specific or are tailored to the SPL development paradigm, mainly because they do not deal with security requirements variability.

This chapter will describe a security requirements management tool called SREPPLineTool which has been developed with the aim of providing SREPPLine (security requirements engineering process for software product lines) (Mellado et al., 2008c) with an automated support. This tool will facilitate the development of secure software product lines and could be applied to the development of secure service-oriented product lines, since the security requirements engineering concepts for SPL that SREPPLine manages do not change for this type of SPL, which is based on SOA techniques.