SETER:
Towards Architecture-Model Based Security Engineering

Ayda Saidane, University of Luxembourg, Luxembourg

Nicolas Guelfi, University of Luxembourg, Luxembourg

ABSTRACT

The quality of software systems strongly depends on their architecture. For this reason, taking into account security requirements at the architecture level is crucial for the success of secure software development. Today, systems are permanently evolving due to customer needs, technology evolution or maintenance constraints. Thus, a resilient secure system is expected to evolve towards more satisfaction of its security requirements (Guelfi 2011). In particular, such evolution process should identify and eliminate faults and vulnerabilities during the development process or runtime. This study focuses on the design phases and aims to propose a resilient software engineering process guaranteeing the development of secure systems that satisfy their critical requirements. During the development process, the system is expected to evolve until reaching satisfactory compliance against its requirements. The satisfaction computation is based on the quantification of failures and degradations. In this paper, the authors propose a novel architecture model-based security testing approach for identifying faults and vulnerabilities. The originality of the proposal resides in the usage of the architecture model for security testing and in coupling security requirements with threat model for generating both security functional test cases and malicious test cases. The assessment of the security requirements' satisfaction and the overall system resilience is based on the test traces analysis. Throughout this study, a client-server system is used as a running example for illustrating the approach.

Keywords: *Architecture Analysis and Description Language (AADL), Architecture Model, Engineering, Model-Based Security Testing, Security Engineering*

1. INTRODUCTION

Engineering resilient systems requires explicit and consistent reasoning of the systems' functional and non-functional requirements, which must be maintained regardless of any changes taking place in the system or its environment. In the relevant literature, we find many works addressing software systems' resilience. Most of them propose operational frameworks addressing resilience at different abstraction levels (Hawes & Reed, 2006; ter Beek, Faconti, Massink, Palanque, & Winckler, 2009; Górski et al., 2006). Moreover, they focus on runtime aspects and they are strongly influenced by intrusion tolerance and dependability approaches. We lack proposals for rigorous resilience engineering methodologies that can be used to

DOI: 10.4018/jsse.2012070102

design and evaluate resilient systems during the development cycle.

In Guelfi (2011), we proposed a formal framework called DREF (Dependability and Resilience Framework) for modeling and evaluating resilient and dependable systems. In particular, it quantitatively defines resilience and satisfaction against some functional or non-functional properties of interest. In our case, we focus on the evaluation of the security requirements satisfaction at both design and deployment phases. Particularly, we are designing a novel architecture-model based security testing methodology as an operational framework associated to DREF. In fact, we propose to use the interpretation of the test traces for experimentally evaluating the satisfaction of the security requirements by the system under test (SUT).

The model-driven engineering paradigm (MDE) is based on the specification, transformation and validation of models representing a system at different levels of abstractions. Specifically, we focus our study around the architecture level in order to take into account the security and resilience requirements early in the software engineering process. Model-based testing is the application of MDE for designing and executing the necessary artifacts of software testing. This is achieved by having a model that describes all aspects of the test target, mainly the test cases and the test execution environment. Usually, the test model is derived in whole or in part from a system model that describes some aspects of the SUT. However, the threat model should be incorporated in the SUT test model when testing security properties in order to generate malicious test cases aiming at violating the requirements and consequently testing the robustness of the system to malicious attacks. The interpretation of results from security functional test cases and malicious test cases is different and so are their generation and selection activities.

We propose a test-driven security modeling framework based on the Architecture Analysis and Description Language (AADL). AADL is an SAE (Society of Automotive Engineers) (http://www.sae.org/standards/) (standard that is widely used in industry and supported by major avionics, automotive and telecommunications providers.

The paper is structured as follows: Section 2 defines motivations and contributions of this work; Section 3 presents the basic concepts of DREF and AADL in addition to an overview of the model based security testing; Section 4 defines the running example; Section 5 presents an AADL based security modeling framework; Section 6 describes test generation and traces analysis steps; finally, Section 7 presents the SETER approach from security modeling to test execution and resilience assessment.

2. MOTIVATIONS AND CONTRIBUTIONS

Software architecture description languages provide a detailed view on the system's components, their interfaces and their interactions. We have in such models enough information to derive test cases relevant for the security properties of interest. What could be expected from testing the architecture model is the elicitation of attack scenarios exploiting some architecture level threats, like covert channels, and also lower level vulnerabilities, such as unchecked user input, by locating their activation and manifestation points. More importantly, architecture level vulnerabilities can be identified only by exploiting the architecture model for test generation as they don't appear in the detailed design test models. However, we don't find in the literature any research work on security test generation from the architecture model.

The quality of the generated test cases is strongly dependent on the quality of the test model. Specifically, we need to propose a modelling framework covering the functional aspect together with the malicious aspect existing in the system and its environment. In MAFTIA (2003), a security fault model has been proposed; it consists of 3 classes of interrelated internal and external security threats: