Chapter 3.12
Identification of Vulnerabilities in Web Services Using Model-Based Security

Sebastian Höhn
Albert-Ludwig University, Germany

Lutz Lowis
Albert-Ludwig University, Germany

Jan Jürjens
Open University, UK

Rafael Accorsi
Albert-Ludwig University, Germany

ABSTRACT

In a service-oriented architecture, business processes are executed as composition of services, which can suffer from vulnerabilities. These vulnerabilities in services and the underlying software applications put at risk computer systems in general and business processes in particular. Current vulnerability analysis approaches involve several manual tasks and, hence, are error-prone and costly. Service-oriented architectures impose additional analysis complexity as they provide much flexibility and frequent changes within orchestrated processes and services. Therefore, it is inevitable to provide tools and mechanisms that enable efficient and effective management of vulnerabilities within these complex systems. Model-based security engineering is a promising approach that can help to fill the gap between vulnerabilities on the one hand, and concrete protection mechanisms on the other. The authors present an approach that integrates model-based engineering and vulnerability analysis in order to cope with the security challenges of a service-oriented architecture.

INTRODUCTION

Information systems consist of a plethora of different applications, services and components. The complex interplay between these system parts is one of the main challenges for the establishment of reliable and secure service oriented architectures (SOA). Among the prominent requirements for

DOI: 10.4018/978-1-60566-950-2.ch001

enterprise information systems is the ability to react to changes quickly and flexibly. To this end, a SOA is deployed in many different application scenarios. It allows the orchestration of services and the implementation of complex business processes without implementing the basic functions over and over again.

Security concepts for SOA heavily rely on model-based technologies. This is due to two prominent reasons: (1) model-based mechanisms work reliably and fast even in complex industrial settings, and (2) SOA itself is a model-based architecture. The deployment and the execution of business processes in a SOA are based on executable business process models mostly written in BPEL. The description of atomic services and their composition to higher-order services is also done in BPEL-Models, together with a WSDL description of the implemented interfaces.

To this end, we propose the integration of model-based security mechanisms for SOA. Current approaches (as explained in the next chapter) neglect the fact that vulnerabilities are major source for security incidents. In classical systems, vulnerability analysis and integration of appropriate counter-mechanisms is a mainly manual task. This is possible because these systems are quite static: they are deployed once and used for longer period in time. In a SOA, systems are composed and re-composed frequently and it becomes infeasible to manually interact with specific instances of business processes or high-order services. For example, they might be part of a complex orchestration. While it might seem a strong assumption that processes and services are orchestrated for unique tasks, systems exist that allow for dynamic integration of additional steps into existing processes (Reichert et al., 2006): by integrating individually required steps, a unique process arises that is executed exactly once.

These scenarios clearly show that security information and vulnerability information must be prepared for automated processing. If users can integrate additional process steps into existing business processes on the fly, it is inevitable to automatically evaluate the security implications. Several security properties of the resulting processes can be evaluated automatically. The following section will provide a motivation for and an overview of these mechanisms. Afterwards, we present a model-based extension for UMLsec that allows for the automated evaluation of vulnerabilities and their effects in a SOA.

Model-Based Security Analysis

Challenges for Computer Security

Attacks against computer systems, on which the infrastructures of modern society and modern economies rely, cause substantial financial damage. Due to the increasing interconnection of systems, such attacks can be waged anonymously and from a safe distance. Thus networked computers need to be secure. The high-quality development of security-critical systems is difficult. Still, many systems are developed, deployed, and used over years that contain significant security weaknesses. Causes: While tracing requirements during software development is difficult enough, enforcing security requirements is intrinsically subtle, because one has to take into account the interaction of the system with motivated adversaries that act independently. Thus security mechanisms, such as security protocols, are notoriously hard to design correctly, even for experts. Also, a system is only as secure as its weakest part or aspect. Security is compromised most often not by breaking dedicated mechanisms such as encryption or security protocols, but by exploiting weaknesses in the way they are being used (Anderson & Long, 2001). Thus it is not enough to ensure correct functioning of security mechanisms used. They cannot be "blindly" inserted into a security-critical system, but the overall system development must take security aspects into account in a coherent way (Saltzer & Schroeder, 1975). In fact, according to (Schneider, 1998), 85% of Computer Emergency Response Team (CERT) security advisories could not have been prevented just by making use of