Chapter 14
A Tool Support for Secure Software Integration

Khaled M. Khan
Qatar University, Qatar

Jun Han
Swinburne University of Technology, Australia

ABSTRACT

This paper presents a tool for the integration of security-aware services based applications that is constructed on the principles of security characterization of individual software services. The tool uses the technique of reasoning between the ensured security properties of the services and the security requirements of the user's system. Rather than reporting the research outcomes, in this paper the authors describe the architecture and capabilities of the tool for secure software integration. The main objective of this paper is to show that an automatic tool support could assist the process of security-aware service based software integration.

INTRODUCTION

In a service oriented system, an application system can be composed of several stand-alone software services developed by third parties. These services are available from Internet based sources. The development paradigm of service oriented applications is appealing to the software engineers as it promises maximum benefits of reusability, productivity, and efficient utilization of Internet. It provides software engineers with an opportunity to compose application systems with pre-fabricated services which are provided by stand alone software components. This paradigm represents the LEGO block style of software composition which provides quicker plug and play application development. In a service oriented system a service providing software component is usually developed, owned, and managed by third parties.

However, the conformity between service consumers' security requirements and security assurances of services over the Internet has become an important issue. In a highly open Internet environment, service consumers are virtually forced to

DOI: 10.4018/978-1-4666-1580-9.ch014

consume services of which they have only partial or no knowledge about their underlying security properties (Khan & Han, 2003). When services are discovered from the Internet and composed with the application system of the consumer, it is not always possible to verify the conformity of security properties between the application system and the third part services. A service is actually offered by a software component or an entity. For simplicity, in this paper we use component to refer to a service providing software entity. In a service oriented system, we need tools and techniques that assist us to check the security compatibility between the selected services and the security requirements of the consumers' application system.

To illustrate the main focus of this paper, let us consider a fictitious distributed healthcare scenario. A number of individual healthcare components provide independent services. Assume a consumer's system **y** running on a machine at a general practitioner's (GP) office connects with a component **s** that provides specialist prescription based on diagnosis report. The service **s** is selected from many such services running at various service providers machines. Component **y** provides a patient's diagnosis report to **s** to get a prescription. After receiving the prescription from **s**, **y** sends it electronically to another component **p** residing on a pharmacist's system for a price quotation. In this case developers would independently develop many such **p** and **s**, and make them available from their various distributed sources which are potentially able to deliver the services that **y** wants. However, component **y** is not only interested in specific services but also wants to know upfront the security properties that the components **s** and **p** could provide with the services.

In this scenario, two issues need to be addressed: (i) how to know the security assurances provided by a service; and (ii) how to verify that the required security properties of the client of a service are complied with the ensured security provided by the services. For example, a component offering pathological services may ensure confidentiality through secure storage and transmission of diagnosis reports. The component **y** (GP) may require confidentiality provided with specific encryption schemes with specific key size. The pathology component may or may not satisfy the general practitioner's (GP) security requirements, depending on how the confidentiality is realized by the service providing component.

The current practices and research provide limited supports for software components and systems security at the composition level. While the existing security technologies have made some progresses in addressing security issues of services, however, they have primarily been focusing on system security at the software infrastructure level. A key consideration missing from all these is how to reason about the security compatibility between a service and an application system. No matter how advanced the security techniques used at the individual service level, these would remain useless if the security properties are inconsistent with the required security of the client's system.

In order to facilitate the security-aware service composition, an automatic tool support is required. The tool could be used by the software engineers to identify the suitable services along with the security profiles of the providing software components, and reason about their compatibility with the service consumer's application system. Based on our approaches reported in (Khan & Han, 2003; Khan & Han, 2005), we have developed a simple tool to characterize the security properties of the services, and verify the security compliance between the service and the consumer's application system. This paper reports the architecture, capabilities, and limitations of the security characterization tool. This paper is organized as follows. Next section outlines the