Security-Aware Service Specification for Healthcare Information Systems

Khaled M. Khan
Qatar University, Qatar

INTRODUCTION

With the rapid advancement of Web-based technologies, healthcare information systems are becoming increasingly heterogeneous in terms of their architecture, composition, and runtime characteristics. A healthcare system can be composed of several stand-alone service components, such as Web services available from various distributed sources for runtime execution. We use the terms *Web services* and *service* interchangeably in this chapter to refer to the same concept. A healthcare application system can be composed of multiple autonomous geographically dispersed software services. A healthcare software service is autonomous as it has its own executable code and uses its own data or files. The composition of a healthcare system can be dynamic or static, depending on how services are connected to each other to provide the services. Some of the services are downloaded directly from the Internet and executed dynamically with the application system. The use of independent services in the healthcare information system is appealing because it supports reusability of code and far efficient utilization of network resources, and it might be cost efficient.

Despite the benefits and usefulness of service computing, the characteristics of third-party services also present tremendous challenges for healthcare information security. As services are reused in far greater scale, there is a need for ensuring the security properties of third-party services to the composite healthcare system. Security is a systemwide property referring to the entire system composed of several services. Security architecture alone at the individual service level cannot protect a system unless the conformity of the security properties is checked with the enclosing application system in terms of required and ensured security properties. This issue of conformity of security properties between the individual service and the enclosing healthcare system has far-reaching implications because failure of this would seriously undermine the privacy of the patients and the reliability of the healthcare providers. Let us give a brief scenario (Han & Khan, 2006) in order to magnify the seriousness of the issue.

Consider a Web service that can generate various test data such as X-ray data, MRI images, CT scan images of the patients, and so forth. Another Web service uses these data as input and generates diagnosis reports to other Web services, which could ultimately produce prescriptions and prognoses of the patients' problems. Finally, the prescriptions are sent to another Web services that supplies medicines to the patients. In this scenario, we can see that the confidentiality and integrity issues of patient data are critical. Similarly, the authenticity of the service provider is an important issue, because the generated output of one service determines the outcome of the treatment produced by another service. Note that all these services are completely independent in terms of their development, their formation, and, most importantly, their security provisions. Different Web services have different security requirements as well as assurances during run-time processing of various sensitive patient data. The fundamental question is how service computing can cater various security requirements of different healthcare services in a composed federated e-healthcare system.

To address this question, this chapter proposes a framework of composing security properties for services used in healthcare information systems. It proposes a generic architecture of our proposed security-aware services and discusses the related issues. In contrast to most of the current initiatives that focus on incorporating existing security techniques into service computing, we see a great need and potential to develop a new framework for security-aware service composition in healthcare systems. The main objective of this chapter is to define a process that would enable us to compose various security properties of a federated healthcare information system that is assembled from various independent third-party software services.

BACKGROUND

The current practice of using software services in healthcare systems does not have any compositional process for properly evaluating the conformity of security properties between third-party services. This could dangerously lead to compromise to the enterprisewide security requirements of the healthcare systems, such as the confidentiality of patient data and the reliability of the services. Consequently, this risky practice of compositing systems of third-party services without due attention to the conformity of security properties could result in the degradation of system security. This practice virtually forces healthcare systems integrators to take undue risks by composing a system in order to achieve global services. Generally speaking, most systems integrators have neither the time nor the resources to examine the security properties of candidate software services. The required security functions that a healthcare system requires may not comply with the provided security profiles of a service.

A thorough examination of related literature reveals that very few research works in this area have been reported. Most research initiatives focus on how to make the individual services secure or how to make the enterprise systems more secure. In our opinion, these cannot solve the problem of "mutual" security conformity among various services

Web services paradigm is based on the Simple Object Access protocol (SOAP), the Web Services Description language (WSDL), and the Universal Description Discovery and Integration (UDDI). The Extensible Markup Language (XML) is the basic mechanism behind all of these technologies. Attempts have been made to extend WSDL to encode quality of services (QoS) properties (Curbera, Khalaf, Mukhi, Tai & Weerawarana, 2003) such as security, performance, and reliability. The aim was not to compose QoS properties along with the service composition, but rather to describe the QoS properties of Web services. None of the standards for XML-based definition language, such as Business Process Execution Language for Web Services (BPEL4WS) (Andrews et al., 2003), XML Process Definition Language (XPDL) (Kappel et al., 1995), Web Services Security Languages or WS-Security (Seely 2002), and Business Process Modeling Language (BPML) (BPMI, 2003), addresses the issue of the composition of security services to engineer the security properties of the composed Web services. WS-Security addresses Web service security by using existing security standards and specifications.

OUR FRAMEWORK

A healthcare software service may provide several functionalities; each may have quite different types of security properties from the others. For a software integrator, it is hard to predict what security properties a functionality supports unless they are well expressed with the interface of the functionality. Security properties are used for various reasons such as to authenticate a system, to authorize, and to ensure confidentiality and integrity. Examples of security properties can be passwords, private keys, secret keys, public keys, shared keys, and digital signatures. The protection of the service and its data and instructions is usually implemented with one or more security functions.

The security properties used to protect a functionality may have distinct security properties for a particular scenario. Security properties could also be grouped into two types based on their role in a functionality: security *precondition* and security *postcondition*. In order to verify whether the security precondition is met by the services user who wants to use the functionality, a reasoning engine is needed.

Based on these preliminaries, we can define the following types of *attributes* of a software service: (i) *the service has one or more functionality*; (ii) *a functionality may be protected with security functions*; (iii) *the security function is composed of security properties*; (iv) *the security properties can be classified as precondition or postcondition of the functionality*; and (v) *a reasoning engine is needed to verify the compliance of the security precondition and postcondition*.

Addressing the compositional security concerns discussed in the previous sections, we propose a framework for specifying these attributes at the architectural level of a service, as depicted in Figure 1.

The framework proposed in Figure 1 shows five distinct tasks. The task for identifying the functionalities of a service involves enlisting the associated functionalities of the service. For example, a *prepare diagnosis report service* may include functionalities such as *receiving pathological data*, *providing diagnosis report,* and so forth. In the next task, each of the identified functionalities is associated with its security properties. For example, *receiving pathological data*