Chapter 4
Security in Service Oriented Architectures: Standards and Challenges

Anne V.D.M. Kayem
German Research Center for Artificial Intelligence (DFKI GmbH), Germany

ABSTRACT

Service Oriented Architectures (SOAs) have become the defacto standard for defining interoperable architectures on the web with the most common implementation of this concept being in the form of web services. Information exchange is an integral part of SOAs, so designing effective security architectures that ensure data confidentiality and integrity is important. However, selecting a security standard for the architecture is challenging because existing solutions are geared toward access control in relatively static scenarios rather than dynamic scenarios where some form of adaptability is needed. Moreover, when services interact across different domains interoperability becomes a problem because of the lack a consistent security model to handle service interactions. This chapter presents a comparative analysis of SOA security standards. The authors discuss the challenges SOA security architecture designers face, in relation to an example travel agent web services scenario, and outline potential mitigation strategies.

1. INTRODUCTION

A Service Oriented Architecture (SOA) can be defined as an approach to distributed computing that allows loosely coupled components to interact seamlessly. On the Internet, in particular, the SOA concept has emerged as a simple but effective way of addressing the communication requirements of loosely coupled, standards-based, and protocol-independent distributed components that often belong to different domains (Papazoglou & Van den Heuel, 2007). Yet, this quality of seamless communications in an open environment raises issues pertaining to data security, privacy, and trust. Service providers need security protocols that allow them to design architectures to protect the services and information they make available to clients (users), while users want firm guarantees

DOI: 10.4018/978-1-4666-2136-7.ch004

of privacy. The problem is made more complex by the fact that task delegations, third party interactions and service compositions can occur between services with different security constraints and/or implementations (Cover, 2002 & Tang et al., 2007).

Composing an offered service requires the SOA to process and spread confidential data to the participating services in ways that enforce the security requirements of all the participants. However, in these cases, decomposing the security requirements of the individual participating services and composing security guarantees does not necessarily result in an overall secure system (Epstein & Matsumoto, 2006; Hutter & Volkamer, 2006; Sidharth & Liu, 2007; Alonso & Larrucea, 2008).

Implementations of Web Services have successfully manifested the SOA concept, and cases of interacting services are handled by negotiation models that establish some form of trust between the participating services (Cover, 2002 and Kleijnen & Raju, 2003). A web service is generally modeled around three key service components namely, the service provider, the client (Requester), and the registry. The service provider publishes available web services in a registry that clients (requesters) can query. Direct communications between a service provider and service requester are handled, by the semantic matchmaker.

Research in securing SOAs has aimed at, and continues to seek, ways of enforcing policies that overcome the challenges of protecting information from exposure during service execution (Sidharth & Liu, 2007; Crispo, et al., 2007; Thuraisingham et al., 2007). Each standard, model, and/or framework targets aspects of message security like confidentiality, integrity, and availability but fails to provide a method of dynamically excluding malicious and/or compromised services from the execution process (Epstein & Matsumoto, 2006; Sidharth & Liu, 2007; Alonso & Larrucea, 2008).

The aim of this chapter is to present a comparative analysis of the state of the art in security standards, models, and frameworks for SOAs. We use an example of a travel reservation web services scenario to discuss the challenges that SOA security architecture designers continue to face in spite of the existing security standards. The example of a travel-reservation web service is aimed at illustrating a case of interacting, but distributed, web services that work together to yield a required result. In such a case, the interacting web services need to operate in ways that enforce pre-set security policies (Buecker et al., 2007; Hutter & Volkamer, 2006). Each of the challenges evoked is supported by an example of an attack possibility as well as a proposition of potential mitigation strategies.

The rest of the chapter is structured as follows. In Section 2, we briefly explain the concept of web services since they are a practical example of an implementation of the SOA paradigm, and continue to give an example of a travel reservation scenario in order to portray a case of interacting services. The section concludes with an overview of SOAs and the impact of security standards in guiding secure SOA design. Section 3, gives a comparative analysis of SOA security standards while Section 4, presents the challenges they face and proposes potential mitigation strategies. We present some ideas for future research in Section 5 and offer concluding remarks in Section 6.

2. BACKGROUND

In order to facilitate the understanding of the rest of the chapter, this section explains the architecture and terminology underlying Service Oriented Architectures (SOAs). For simplicity, we use the concept of web services to explain how self-contained and self-describing components interact to provide a service. This is due to the fact that web services are a practical manifestation of how SOAs can be built successfully. Moreover, web services are composed of self contained, self-describing functions that can be published, located, and programmatically invoked over the