Securing Composite Web Services

Brandon Clopton

Abstract

Web services (WS) are a recent advancement in the way applications and the data involved upon execution are shared. Applications published online via WS can be accessed by many different types of systems. Businesses can take advantage of this ease of interoperability to expand the consumption of their online services. However, with no boundaries for accessing these resources, some sort of security must be implemented to protect them. A single web service can be annotated with security attributes. When composing a mesh of services, though, the composite web service’s security can be hard to assess, and even harder to implement effectively. The unique contribution of this research is the evaluation of the feasibility of WS security annotation to support dynamic WS composition.

1. Introduction

Service Oriented Architecture (SOA) relies on a group of individual software component’s functionalities to react with one another to perform a greater task. Each piece of software, or service, is able to be invoked via an interface [Sprott]. For SOA to be effective, its resources must be easily utilized. Different software applications, running on a variety of platforms and frameworks, need to be able to communicate. Therefore, protocols must be established by which services on various machines can interoperate in a standard form.

A web service is a software system that solves this interoperability dilemma over a network. Machines are allowed to interact with each other because they can process the format of the interface programmed with the eXtensible Markup Language (XML). Messages are sent between machines using the Simple Object Access Protocol (SOAP). The Web Services Description Language (WSDL) describes the functionality of a web service, thus helping to locate necessary web services. To further help with finding web services, the Universal Description, Discovery and Integration (UDDI) service holds those that register along with their WSDL descriptions [Booth]. After a web service is found, the service requestor and service provider must agree on the semantics of their interaction before it takes place. Composing a group of web services together is made possible by the Business Process Execution Language for Web Services (BPEL4WS) [Juric].

Each web service can be annotated to have incoming and outgoing security constraints, allowing the composition of adjacent services to be secure. With these in place, each service requestor’s capabilities must match up with the respective service provider’s constraints. This makes for an effective way to combine compatible web services [Carminati]. However, if a web service’s constraint level of security differs from that of its capability, a question arises as to what level of security the composite web service obtains.

The travel agent service, derived from [Xu], is a good example of how this works. A user accesses a single interface, entering in the information needed to book a flight, book a hotel room, and obtain maps from the area. Each of these three obligations will be met by its own web service. The problem here is that the user may submit some information that only needs to be used by certain parts of the service. For instance, a credit card is involved with airline and hotel reservations, but a website can provide maps for free. As a travel agency customer, one would want to know that the credit card information will be secure and accessible to only necessary
vendors. BPEL4WS does not provide this type of security in itself, which presents the problem for composing web services in a secure manner.

During this research, industry standards and web service technologies were explored to analyze the feasibility of web services security annotations to support dynamic web services composition. The rest of this paper is organized as follows: Section 2 presents other approaches to building secure web service compositions. Section 3 provides relevant material for security-conscious composition. Section 3.1 introduces industry standards. Section 3.2 gives an overview of security annotations. Section 3.3 offers future research possibilities.

2. Background

Previous attempts at securing composite web services were explored in hopes to gain a more intuitive outlook on the issue. The Organization for the Advancement of Structured Information Standards (OASIS) put forth the Security Assertion Markup Language (SAML) as a standard to define how security credentials are exchanged between online business partners [Hughes]. This concept relies on a service being previously used and documented correctly by participating partners. The dilemma here is that new web services have no asserting parties, and services still have to trust an outside identity provider.

Studies from Iowa State University [Pathak] show that composition techniques using a trust-based system have a limited security level. This method assumes that once a service is trusted, it can always be trusted. However, since web services are not always combined with the same services, the trusted service may unknowingly branch to a service that has not yet been trusted.

Another approach to securely composing web services together is to send a service model back to the consumer before executing. Upon invocation of the composite web service, the requestor sends in security policies that the service must uphold [Xu]. The downside to this proposal is that numerous consumer privacy obligations may not be achievable by one composite web service, reducing its usability.

3. Approach

Organizations, corporations, and the general public have all contributed to creating technological standards. These are sometimes enough to develop a desired project, but technology is continually advancing to get more out of its resources. Web services, for instance, can take care of redundant software, and moreover eliminate machine-to-machine incompatibility. Along with this innovation, though, comes a problem with securing a network of services. Therefore, the question is: can current industry standards support secure dynamic web service composition?

3.1 Industry Standards

BPEL4WS defines the behavior of a business process between multiple web services, or partners. Each partner has a description that encapsulates an interface, called a partner link, which lets other services know how an interaction must take place [Andrews]. To be able to do this, BPEL4WS is layered on top of other XML specifications. A WSDL message describes a web service in an attempt to advertise for potential partners. In particular, it expresses the operations that the web service offers. Also, a message exchange pattern is published so other services know how to invoke these operations [Chinnici]. XML Schema is typically used to
illustrate the structure of a document. XML documents form a tree structure, allowing machines
to process the data. Each node of the tree is specified by XML Schema as to what type of data it
can hold [Fallside]. In this case, the schema tells each service what kind of format is needed for
sending messages to one another within the business process. SOAP provides a messaging
framework in order for web services to send/receive messages with each other. A SOAP
message is sent in an envelope made up of a header and a body [Gudgin]. At its very basic,
SOAP has no way of securing messages.

To compensate for the lack of secure communication, extensions were added to XML.
XML Encryption is a good way to disguise information while it is being sent. The root of the
document can be encrypted, as well as a single node. The downfall here is that encryption relies
on cryptographic keys. When ciphering messages, each partner would have a public key and a
private key. Encrypting the information involves the public key, which is then sent along with
the message to let the recipient know which respective private key to use when decrypting it.
Thus, the keys have the same protection as a password for a personal account. As another
approach at protecting the integrity of a message, XML Signature was created. The receiving
partner of a message must check for a trusted certificate before accepting it [CISCO]. However,
a service must first know which certificate authorities can be trusted for this method to be
effective.

3.2 Security Annotation Overview

In the work done by [Bush], problems regarding security in service orchestration were
addressed. First, a way to assign individual web services a level of security was established.
The Web Ontology Language (OWL), created by the Naval Research Lab (NRL), was used.
OWL is specifically designed for use by applications that must process information within a
document, rather than the content being presented to humans [McGuinness]. Security attributes
were satisfactorily added to lone web services by [Bush]. To accomplish this, a framework to
annotate a language similar to BPEL4WS with the attributes was developed. Further
advancements in extending the business process language were theoretically explored to support
secure composite web services.

The model for annotating individual services with a proper security level is a great start.
This seems to work well when a web service performs a function on its own, allowing only users
with matching security capabilities access to the interface. [Bush] took it one step further and
showed that this framework implementation would allow two partners to be orchestrated
securely together. The example given is a data service communicating with a second warning
service. The data service holds information about military aircraft flight routes, and the warning
service is able to detect when two flights will intersect each other.

Again, this simplified model proves that the annotation of security attributes for
individual web services is a good starting point for securing dynamic composite web services.
The first obstacle to overcome is to make sure one service is able to recognize the other’s
security constraints and/or capabilities. The reason this is still just the beginning is because only
two services were orchestrated together. Given any two web services with security attributes, the
only way for them to be compatible is if they share the same level of security. Therefore, either
the two web services will be securely composed, or they will not be composed together at all.
When a third service is introduced to the network, the level of security has the ability to change
over the course of interoperations. A partner’s incoming security may not be the same as its
outgoing security, thus compromising the security level of the composite web service.
Figure A will include a model of the travel agency service annotated with security attributes.

Description of Figure A.

3.3 Future Research Directions

References


