Chapter XIII
Approaches and Best Practices in Web Service Style, XML Data Binding, and Validation:
Implications to Securing Web Services

David Meredith
STFC E-Science Centre, UK

Asif Akram
STFC E-Science Centre, UK

ABSTRACT

This chapter shows how the WSDL interface style (RPC / Document), strength of data typing, and approach to data binding and validation have important implications on application security (and interoperability). This is because some (common) bad-practices and poor implementation choices can render a service vulnerable to the consequences of propagating loosely bound or poorly constrained data. The chosen Web service style and strength of data typing dictate how SOAP messages are constructed and serialized, and to what extent SOAP messages can be constrained and secured during validation. The chosen approach to binding and validation dictates how and where the SOAP-body and SOAP-header (which includes the security constructs) are handled in the application, and also determines the reliability of message parsing. The authors show how these Web service styles and implementation choices must be carefully considered and applied correctly by providing implementation examples and best practice recommendations.

INTRODUCTION: HOW WSDL STYLE, STRENGTH OF DATA TYPING, BINDING, AND VALIDATION ARE IMPORTANT FOR WEB SERVICE SECURITY

In this chapter, we show how the WSDL interface style (RPC / Document), strength of data typing ("loosely typed" or "strongly typed") and method used for data binding and validation have important implications to application security and interoperability, as each play important roles in securing and constraining how the message data is handled (or propagated) by an application. Some common bad-practices and poor implementation styles can cause serious security implications. These Web service implementation styles and choices must be carefully considered and applied correctly in order to ensure; a) The correct choice of Web service style, which dictates how the SOAP message is constructed, serialized, and to what extent messages can be constrained during validation; b) The complete binding and validation of a SOAP payload before the business logic is invoked. This includes both *how* and *where* the SOAP Body and SOAP header (which contains the security constructs) are handled in the application; The chosen binding and validation style also dictates the reliability of data parsing, which is especially important when parsing complex security constructs associated with Web service security specifications; c) Data contract compliance between client and service; and d) Interoperability and robustness.

If implemented incorrectly, a Web service can be vulnerable to the consequences of accepting (or propagating) loosely bound or poorly validated data. For example, a service that defines only "loose" WSDL typing with only a limited or "weak" binding and validation strategy would be vulnerable to routing erroneous, restricted, or even malicious message content to business logic and to third-parties. The potential consequences are similar for the client. These consequences can easily compromise security, hinder reliability, and breach service and consumer trust. In this chapter, we provide some implementation best practices and recommendations based on our experimentation with Web service styles and binding/validation frameworks. Examples are provided using current SOAP standards for Java, including JAX-RPC and JAX-WS. We assess the advantages and disadvantages associated with "loose" verses "strong" data typing and when decoupling the binding/validation framework from the SOAP engine. For the most part, we recommend the use of the Document/ literal wrapped Web service style coupled with a dedicated binding and validation framework. These choices leverage the advanced capabilities of XML schema for precisely declaring/describing, constraining, and validating data and security constructs.

BACKGROUND

At the heart of Web service style, interoperability, and data binding and validation is the Basic Profile 1.0 (Basic Profile Version 1.0, 2004), published by the Web services Interoperability Organization (*http://www.ws-i.org/*). The Basic Profile 1.0 details how to use primary Web service related specifications together, including; XML (Extensible Markup Language, 2006), WSDL (Web Services Description Language, 2001), SOAP (Simple Object Access Protocol, 2000), and UDDI (Universal Description, Discovery and Integration, 2004). In doing this, the Basic Profile provides important guidelines that should be observed in order to create fully interoperable and well defined/constrained services. The recommendations and best-practice examples provided throughout this chapter are WS-I compliant and coverage of the Basic Profile is an important part of this chapter. Of critical importance to both the interoperability and effective binding and valida-