

Web Services Security: Challenges and Techniques

Anoop Singhal
Computer Security Division, NIST
psinghal@nist.gov

Abstract

Web Services based computing is currently an important driver for the software industry. While several standards bodies (such as W3C and OASIS) are laying the foundation for Web Services Security, several research problems must be solved to make secure Web Services a reality. This paper describes techniques for Web Services Security and some of the challenges for the future.

1. Introduction

The advance of Web services technologies promises to have far-reaching effects on the Internet and enterprise networks. Web services based on the eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP), and related open standards, and deployed in Service Oriented Architectures (SOA) allow data and applications to interact without human intervention through dynamic and ad hoc connections. Web services technology can be implemented in a wide variety of architectures, can co-exist with other technologies and software design approaches, and can be adopted in an evolutionary manner without requiring major transformations to legacy applications and databases.

2. Web Services Security Stack The open standards communities that created Web services developed a number of security standards for Web services. Figure 1 illustrates a notional reference model for Web services security standards. This reference model maps the different standards to the different functional layers of a typical Web service implementation. These layers are modeled after the OSI Reference Model but are not intended to be interpreted as strictly hierarchical.

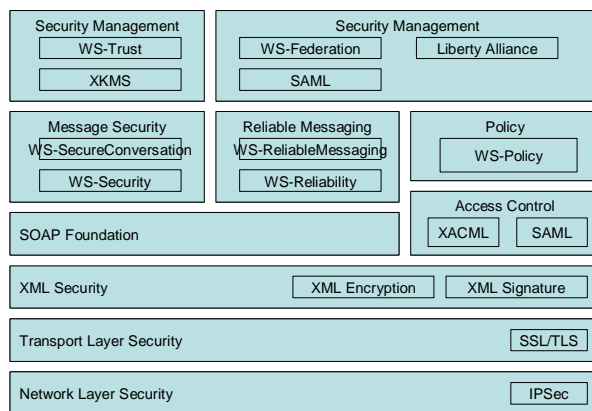


Figure 1: The Web Services Security Stack

3. Challenges

This section discusses some of the important challenges in the area of web services security.

1. Discovery

In Web services discovery, participants identify and compose Web Services Description Language (WSDL) specific services based on definitions in a UDDI registry. Due to the potentially large number of service candidates in the registry, performance rankings for algorithms used to search, match and compose services can vary from case to case.

2. End to End Quality of Service and Protection

Most Web services deployed do not provide guarantees for Quality of Service (QoS) or Quality of Protection (QoP). QoS is important in defining what the expected level of performance a particular Web service will have.

3. Availability and Protection from Denial of Service Attacks

Availability enables a Web Services Application to detect a Denial of Service (DOS) attack, to continue operation as long as possible and then to gracefully recover and resume operations after a DOS attack.

4. Recommendations

This section describes some of our recommendations for secure Web Services.

1. Replicate Data and Services to Improve Availability

Since Web Services are susceptible to Denial of Service (DOS) attacks it is important to replicate data and applications in a *robust* manner.

2. Use Logging of Transactions to Improve Accountability

Non-repudiation and accountability require logging mechanisms involved in the entire SOA transaction. As of this writing, there are few implemented logging standards that can be used across an entire SOA. In particular, the level of logging provided by various UDDI registries, identity providers, and individual Web services varies greatly. Where the provided information is not sufficient to maintain accountability and non-repudiation, it may be necessary to introduce additional software or services into the SOA to support these security properties.

3. Use Threat Modeling and Secure Software Design Techniques to Prevent Attacks

The objective of secure software design techniques is to ensure that the design and implementation of Web Services Software does not contain defects that can be exploited. Threat modeling and risk analysis techniques should be used to protect the Web Services application from attacks. Used effectively, threat modeling can find security strengths and weaknesses, discover vulnerabilities and provide feedback into the security life cycle of the application. Software security testing should include security oriented code reviews and penetration testing. By using threat modeling and secure software design techniques Web services can be implemented to withstand a variety of attacks.

4. Use Performance Analysis and Simulation Techniques for End to End Quality of Service and Quality of Protection

Queuing networks and simulation techniques have long played critical roles in designing, developing and managing complex information systems. Similar techniques need to be used for quality assured and highly available web services. In addition to QoS of a single service, end-to-end QoS is critical for most composite services. For

example, enterprise systems with several business partners must complete business processes in a timely manner to meet real time market conditions. The dynamic and compositional nature of web services makes end-to-end QoS management a major challenge for service oriented distributed systems.

5. Use XML Firewalls

One way to mitigate risks from Web based applications is to use XML firewalls. This filters XML documents and it uses contents in XML documents to make decisions. For example, it can filter malicious content, or specially crafted parameters that can be used to generate a SQL query by the attacker. Additionally, if it is configured properly, it can act like a WS-Security end point, to verify user signatures and validate user identity.

6. Use Penetration Testing

Web Services interfaces should always be penetration tested before they are opened up for external access. These interfaces can be quite complex and even experienced development teams may make mistakes which can have security implications.

5. Conclusions

Web Services based computing is currently an important driver for the software industry. The primary goal of Service Oriented Computing is to make a collection of software services accessible via standardized protocols whose functionality can be automatically discovered and integrated into applications. While several standards bodies (such as W3C and OASIS) are laying the foundation for Web Services, several research problems must be solved to make secure Web Services a reality. Service description, automatic service discovery as well as QoS, reliability and protection considerations of atomic and composite Web Services are the important problems that need to be solved.