View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2004

Web Services and Web Services Security

Robert Boncella
Washburn University

Follow this and additional works at: http://aisel.aisnet.org/amcis2004

Recommended Citation

Boncella, Robert, "Web Services and Web Services Security" (2004). *AMCIS 2004 Proceedings*. 565.
http://aisel.aisnet.org/amcis2004/565

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Web Services and Web Services Security

Robert J. Boncella
Washburn University
bob.boncella@washburn.edu

ABSTRACT

Web services are self-contained modular applications that provide a computation upon request. These services can be described, published, located, and invoked over a network, generally the Internet but intranets, extranets, and LANs can be used as well. Most often the World Wide Web (WWW) will be the means for the request of services.

In using web services for its information systems needs, a firm may open access to its information assets. This action can become an attractive target for malicious hackers, industrial espionage, and fraud. The assurance of security of web services is necessary for a firm to be willing to adopt this technology as a means of creating its information systems

Keywords

web services, ws-security, web service security

INTRODUCTION

The purpose of this tutorial is to provide foundation for an understanding of the need and techniques of web services security. To provide this foundation a review of the concept of web services is presented as well as its related ideas. The security requirements for web services are discussed and finally the techniques that provide these security requirements are presented.

WEB SERVICES

Definition and Implications of Web Services

A reasonable definition of *web services* is "web services are self-contained modular applications that provide a computation upon request". These services can be described, published, located, and invoked over a network, generally the Internet but intranets, extranets, and LANs can also be used. Most often the World Wide Web (WWW) will be the means for the request of services. Furthermore these services are independent of the underlying systems providing the computation.

The implications of this paradigm of computing cannot be underestimated. Potentially, the web services architecture can provide universal interoperability of software applications. Every software application in the world could interact with every other software application in the world. This interaction is independent of geographical location, system hardware, operating system, and programming languages of the software application.

The web services paradigm of computing could provide a de facto standard for any particular type of computation. For example any number of computing systems can provide the verification of a customer's shipping address. If each of these systems provide this service publicly upon request that service becomes a commodity. And as a commodity it will allow for the substitutability of services. It is possible a firm can develop an information system for its use based on the computational services provides by the web services paradigm of computing. That information system can be developed at least cost because of the substitutability of services provided by the defacto standardization of computation. This method of development impacts the cost of maintaining the information system. The firm is not responsible for the process that provides the service but the service provider is and hence is required to maintain of the quality of that service.

Many services that a firm uses in its business operations have this characteristic of web services. For example, a firm that needs to ship an item from New York City to Laramie, Wyoming can use a number of service providers, including UPS, USPS, and FedEx, and DHL. The firm is interested in the results of service not how the provider implements the service.

If a firm is to use the advantages of web services it must have trust in the security of the web services. In using web services for its information systems needs, a firm will provide access to its information assets. This action can become an attractive target for malicious hackers, industrial espionage, and fraud. The assurance of security of web services is necessary for a firm to be willing to adopt this technology as a means of creating its information systems. To appreciate the challenges of web services security an overview of the architecture is in order.

Web Services Architecture

The foundation of web services is the request/response paradigm. This paradigm involves a client requesting a service via a specific protocol to a service provider. That service provider responds with either the service or a notification of why it cannot provide that service using a specific protocol. An example of this would be a web browser using the HTTP protocol to request a web page from a web server. T he details of this paradigm can be found in (Boncella 2000). This concept is used to implement the *service oriented architecture* (SOA) which implements web services. This architecture contains three entities and three operations and is illustrated in Figure 1.

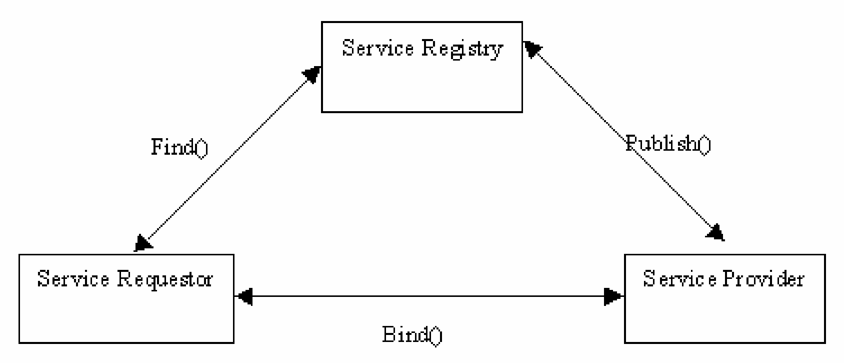

Figure 1

The *service requestor* is an application that requires a service (computation) from another application or applications. If the requesting application does not know where this service is located it sends a request to the *service registry* using the Find() operation.

The *service registry* is a well know application that responds to a Find() request sent by a service requestor. The service requestor provides search criteria for a particular service with the Find() operation. The service registry will return information about the requested service and where to find that service.

The *service providers* Publish() the specifications of the services they provide and where they are provided to the service registry.

If the specifications of service provider's service match the requirements of a the service requestor the service requestor uses the location information provide in the service registry to Bind() to the service provider. Once bound together the service requestor and service provider can engage in the request/respond paradigm to complete a computation.

For these entities and operations to carry out their intended functions they use a set of specifications and standards that allow uniform message passing between the entities. The following specifications are the components that are used to implement web services computing paradigm.

- HTTP/1.1
- RFC 2956:HTTP State Management Mechanism (Cookies)
- SOAP 1.1
- UDDI ver. 2.04 API
- WSDL 1.1
- XML 1.0
- XML Schema Part 1: Structures
- XML Schema Part 2: Datatypes

A brief description of each specification will follow along with references where a more detailed can be found.

HTTP/1.1 - Hypertext Transfer Protocol is the protocol that specifies the request/response process that take places between a web client and a web server. See http://www.ietf.org/rfc/rfc2616.txt and (Boncella 2000) for details.

RFC 2956 - Cookies specifies a mechanism that creates a stateful session when using HTTP requests and responses. See http://www.ietf.org/rfc/rfc2965.txt and (Boncella 2000) for details.

SOAP 1.1 - Simple Object Access Protocol is a message protocol that enables requests and responses to be sent in XML format from client to a server. SOAP defines an *envelope* that contains a *header* and a *body*. The SOAP body contains the *payload*. The payload contains the request for the service from the client and then the response to the request from the server. See http://www.w3.org/TR/soap/ for more details.

UDDI ver. 2.04 API - Universal Description, Discovery, and Integration is a specification of the registry that lists web services that are of interest to a service requestor entity. It uses taxonomies that categorize web services in a way meaningful to clients. See http://www.uddi.org/ for more details.

WSDL 1.1 - Web Services Description Language is a specification that details how to describe a web service. A WSDL document for a service is an XML document that contains information a programmer needs in order to contract for that service. See http://www.w3.org/TR/wsdl for more details.

XML 1.0 - XML is a tag-oriented language whose tags can be user defined and are used to describe the data contained in the document. See http://www.w3.org/XML/ for more details.

XML Schema Part 1: Structures XML Schema: Structures can be used to define, describe and catalogue XML vocabularies for classes of XML documents. See http://www.w3.org/TR/xmlschema-1/ for more details.

XML Schema Part 2: Datatypes XML Schema: Datatypes can be used to define datatypes in XML vocabularies and documents. See http://www.w3.org/TR/xmlschema-2/ for more details.

An Example

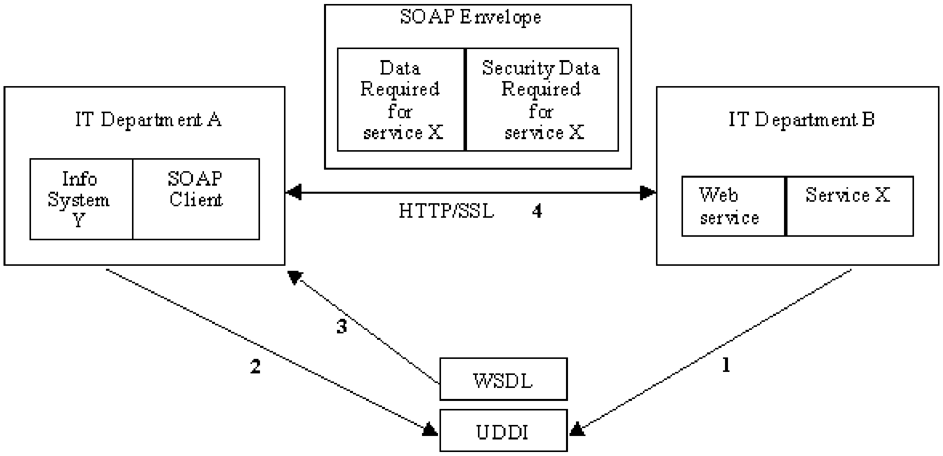

Figure 2

Step 1 - IT department B creates service X. Publishes it using UDDI specifications in a public service registry using the publish() procedure. This site contains the URL of the WSDL document that gives the specifications for service X.

Step 2 - IT department A is creating information system Y and needs service X. They consult the public registry via the find() procedure. They find IT department B's entry for service X.

Step 3 - IT department A uses the URL posted in the public registry to download a copy of the WSDL specification for service X. Using this specification they build a SOAP client to request service X.

Step 4 - IT department A's SOAP client builds a SOAP envelope that contains the required data for service X and the requisite security data for using service X provided by IT department B.

WEB SERVICES SECURITY

If a firm is going to use web services for its information systems it must be assured that its data will be secure. To understand web services security an overview of information security is provided below.

Information Security Requirements

The five requirements for that defines information security are:

- *Confidentiality* - This requirement assures user privacy and prevents the theft of information both in transit and stored. Symmetric and asymmetric encryption are used to create cipher text for transmission to and from clients and servers and for information held in storage.
- *Integrity* - This requirement assures that information either in transit or in storage was not modified, either intentionally or unintentionally. An encrypted message digest - a digital signature - assures message integrity.
- *Nonrepudiation* - This requirement assures that the sender of a message cannot legitimately claim they did not send the message. Digital Certificates and Public Key Infrastructure (PKI) are used to assure nonrepudiation.
- *Authentication* - This requirement assures that the sender and receiver are who the claim to be. PKI as well as smartcards and user name/password authentication methods can be used to assure authentication.
- *Authorization* - This requirement assures that an authentication entity can access only those information resources they are required to have either to request or provide a service. Once authenticated an entity authorization will be determined. Generally an authenticated entity has an associated access control list (ACL)
- *Availability* - This requirement assures that uninterrupted service is provided to authenticated and authorized users. In addition, interruptions of service by denial-of-service attacks are controlled.

Currently SSL (Secure Sockets Layer), PKI (Public Key Infrastructure), and firewalls are able to meet these requirements for conventional web traffic using HTTP (Boncella 2000, Boncella 2003). However SSL and firewalls are inadequate to assure these requirements for web services.

Inadequacies of SSL for Web Services Security

Figure 3 depicts the case where a web service is provided indirectly to the user. From a security view two sets of information security requirements need to be assures. These are referred to as security contexts.

1. The security context that covers the user to the web site.

2. The security context covers the user to the web service provider. This context is referred to as *persistent security*.

Persistent security requires the security of the SOAP request/response message be assured over more than one client/server connection. The security of the SOAP message extends beyond the first request/response interaction. SSL is inadequate for the persistent security requirement

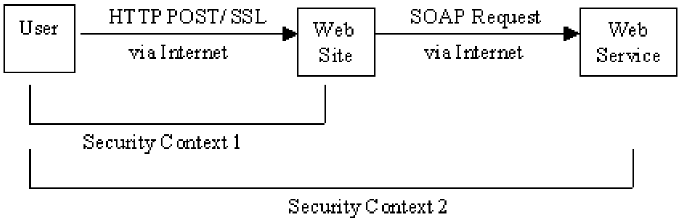

Figure 3

SSL was not designed to handle this type of process. SSL does encrypt the data stream but it does not provide end-to-end confidentiality. In particular SSL leaves the data exposed between the web site and the application providing the service.

SSL will not be able to assure integrity since it does not provide digital signatures.

Most often, implementations of SSL only ensures one-way authentication - the user is assured of the provider but the provider is not assured of the user. In Figure 3 it is possible that the user is not assured of the service provider since it is an indirect service provider. If authentication cannot be met, the authorization will be suspect.

Finally SSL does not support an end-to-end audit trail from service request to service response.

Web Services Security Requirements

The requirements for information security (such as confidentiality and integrity) remain the same for web services. The means by which they are assured is different from SSL and firewalls because of the additional requirement of persistent security. To assure persistent security, SOAP messages must include information about the message's security requirements. WS-Security is a de facto method of including security data into SOAP messages.

WS-Security Technology

WS-Security defines placeholders in the SOAP header to insert security information. It also specifies how encryption and digital signatures are to be inserted into SOAP messages. WS-Security also provides the specification for inserting any required security tokens. See http://www-106.ibm.com/developerworks/webservices/library/ws-secure/ for detailed information.

Listed below are brief descriptions of how each information security requirement is assured for web services using WS-Security.

Confidentiality for Web Services

XML Encryption is a specification produced by the World Wide Web Consortium (W3C). XML Encryption is used to encrypt portions of XML documents. XML Encryption is used to assure confidentiality in the case of a security context that ranges beyond a simple HTTP/SSL connection (Figure 2). In the case of a security context ranging over several SOAP intermediaries, portions of the SOAP document can be kept confidential from any SOAP intermediary en route as the message makes its way from the user to the web service provider and back. See http://www.w3.org/Encryption/2001/ for detailed information.

Integrity for Web Services

XML signature is a specification produced jointly by W3C and the Internet Engineering Task Force (IETF). An XML signature is the XML equivalent of a digital signature. It can be used to digitally sign selected portions of an XML document. In particular it can be used to sign data and thereby assure its integrity. An XML signature is used within SOAP messages. See http://www.w3.org/Signature/ for detailed information.

Authentication and Authorization Web Services

A web services user can request services from any number of service providers. That user should be authenticated on each service providers system. This authentication should then be to determine the resources that user is authorized to access on a particular service provider's system. Rather than prompting a user for authentication each time a request for service is made in a sequence of requests it is desirable to have a single sign on (SSO) process. In SSO, when the initial web service provider authenticates a user, any subsequent requests generated by that user to other service provider's systems will be automatically authenticated on that system and the user's authorization will be determined.

The two approaches to implementing SSO are:

1.. Include authentication information for each web service in the initial SOAP message.

2. maintain a user's authentication list in a central repository.

Security Assertions Markup Language (SAML) and XML Access Control Markup Language (XACML) can work together to implement the first approach. SAML information can be inserted into SOAP messages. This information is about user authentication and authorization as well as information about the user. XACML express access control rules in XML format. For detailed information about SAML see http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security and for XACML see http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

Microsoft's Passport scheme and Sun's Liberty Alliance Project use the centralized repository approach to user authentication.

Nonrepudiation - PKI for Web Services

XML Key Management specification (XKMS) provides PKI services (registering, locating, and validating keys) through XML. This service is provided as SOAP based web service. See http://www.w3.org/TR/xkms/ for detailed information.

Web Services Security Threats

A potential security threat introduced by the use of web services is possibility of a SOAP message containing malicious data that would cause the web service application to execute in an unintended mode. Or the SOAP message could contain a request for a service that is not advertised on that site is provided. That unadvertised service could compromise the service provider. SOAP messages easily pass through firewalls. What is needed is firewalls that filter the content of SOAP messages requesting passage through the firewall. See (Albrecht 2004) for more details on this vulnerability.

SUMMARY

As stated in the Introduction, The purpose of this tutorial is to provide foundation for an understanding of the need for and techniques of web services security. In addition an overview of the uses of web services is provided as is a discussion of its architecture and its components.

REFERENCES

1. Albrecht, C. (2004) *How Clean Is the Future of SOAP?, Communication of the ACM*, 47,2,Feb. 2004.
2. Atkinson, B., Della-Libera, G., Hada, S., Hondo, M., Hallam-Baker, P., Klein, J., LaMacchia, B., Leach, P., Manferdelli, J., Maruyama, H., Nadalin, A., Nagaratnam, N., Prafullchandra, H., Shewchuk, J., Simon, D. (2002) "Web Services Security (WS-Security)", http://www-106.ibm.com/developerworks/webservices/library/ws-secure/ (current Feb. 22, 2004)
3. Biron, P.V. and Malhotra, A. (2001) "XML Schema Part 2: Datatypes" http://www.w3.org/TR/xmlschema-2/ (current Feb. 22, 2004)
4. Boncella, R. (2000) "Web Security for E-Commerce", *Communications of the AIS*, 4, 11, Nov. 2000.
5. Boncella, R. (2003) *SSL* in *The Internet Encyclopedia*, Hossein Bidgoli (Editor), New York, New York, J. Wiley, 2003.
6. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. (2001) " Web Services Description Language (WSDL) 1.1", http://www.w3.org/TR/wsdl (current Feb. 22, 2004)
7. Fielding, R., Gettys, J., Mogul, J.,. Frystyk, H., Masinter,. L.,. Leach, P., and Berners-Lee, T. (1999) "Hypertext Transfer Protocol HTTP/1.1", http://www.ietf.org/rfc/rfc2616.txt (current Feb. 22, 2004).
8. Eastlake, D., and Reagle, J (2001) " XML Signature", http://www.w3.org/Signature/ (current Feb. 22, 2004)
9. Ford, W., Hallam-Baker, P., Fox, B., Dillaway, B., LaMacchia, B., Epstein, J., Lapp, J., (2001) " XML Key Management Specification (XKMS)", http://www.w3.org/TR/xkms/ (current Feb. 22, 2004)
10. Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., Nielsen H. F. (2003) "SOAP Version 1.2 Part 1: Messaging Framework", http://www.w3.org/TR/soap/ (current Feb. 22, 2004)
11. Kristol, D, and Montulli, L. (2000), " HTTP State Management Mechanism", http://www.ietf.org/rfc/rfc2965.txt (current Feb. 22, 2004)
12. OASIS (2001) (Organization for the Advancement of Structured Information Standards), " Universal Description, Discovery and Integration", http://www.uddi.org/ (current Feb. 22, 2004)
13. OASIS (2003), (Organization for the Advancement of Structured Information Standards), " eXtensible Access Control Markup Language", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.(current Feb. 22, 2004)
14. OASIS (2004), (Organization for the Advancement of Structured Information Standards), "Security Assertion Markup Language (SAML)", http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security (current Feb. 22, 2004)
15. Reagle, J. (2001) " XML Encryption", http://www.w3.org/Encryption/2001/ (current Feb. 22, 2004)
16. Thompson, H.S.,Beech, D., Maloney, M., Mendelsohn N. (2001) " XML Schema Part 1: Structures", http://www.w3.org/TR/xmlschema-1/ (current Feb. 22, 2004)
17. W3C (1996) ' Extensible Markup Language (XML)", http://www.w3.org/XML/ (current Feb. 22, 2004)