DOES THE WEB SERVICES-SECURITY (WS-S) STANDARD SUITABLY ADDRESS THE NEEDS OF A HETEROGENEOUS B2B LANDSCAPE?

Allyson Bantom[a] **and Maree Pather**[b]

[a] *Port Elizabeth Technikon*

[b] *Port Elizabeth Technikon*

[a] Postal Address: Suite 84 Private Bag X27964
Greenacres
Port Elizabeth
6057
[a] Telephone Number: (041) 9944015 or 0834796571
[a] Fax Number: (041) 9945421
[a] Email: s9845651@student.petech.ac.za or bantom@vwsa.co.za
[b] Telephone Number: (041) 504 3580
[b] Email: Maree@petech.ac.za

ABSTRACT

Today's application designs are moving towards the ideal of centrally invoked reusable code components that are available over standard protocols such as HTTP and SMTP. Three role types can be found in a typical Web Services-oriented architecture: that of Web service provider, Web service requestor and Web service broker. The service provider could publish Web Services to a server cluster somewhere on the Internet. A Web service requestor could discover Web services on the server and request the use of one or more methods, which the server could then allow the requestor to invoke. In a world void of malicious intent, web users would have rapidly adopted such a framework a long time ago. Unfortunately, it is widely accepted that one of the main barriers to the adoption of Web Services is security.

The following three items are listed in the literature as the main challenges facing Web services security today:

1. Security pertaining to Web service requestors
2. Persistent security in a multi-hop request/response SOAP message
3. The requirement to separate Web services security from the underlying network.

Many XML-based standards and initiatives have been developed to address the basic requirements, such as authentication, authorisation, confidentiality etc. of Web services security. This study evaluates these standards (e.g. XML Digital Signature, XML Encryption) within the context of the WS-Security specification, initially developed by IBM, Microsoft and VeriSign and now driven by OASIS. The specifications contained therein (such as WS-Policy, WS-Privacy) will briefly be described. The paper is aimed at critically evaluating the WS-S specification in terms of its ability not only to provide the security requirements of Web services, but also to do so in the context of a heterogeneous business-to-business landscape.

KEY WORDS

Web services, XML, SOAP, WS-Security

DOES THE WEB SERVICES-SECURITY (WS-S) STANDARD SUITABLY ADDRESS THE NEEDS OF A HETEROGENEOUS B2B LANDSCAPE?

1. INTRODUCTION

Due to an increasing need to communicate and collaborate with their business partners, businesses are initiating projects that include the deployment of selected applications and the exposure of sometimes sensitive information outside of their 'safe' local-area network (LAN) infrastructures. They envisage the automation of many of their business processes, which could be realised if they were to find a way to enable their applications to provide functionality to or request functionality from that of their trusted business partners'. This requirement is usually met with many concerns, the two primary ones being security and interoperability. How will the fundamental information security requirements be implemented across this relatively 'open' arena? How will the various applications, written in different programming languages and running on various platforms seamlessly integrate with one another? How will a heterogeneous security technology landscape be co-coordinated? Attempts to provide comprehensive answers to these questions have resulted in the creation of many specifications and standards. This article deals with some of these with the intention of focusing on the security aspect of interoperable business applications specifically at the messaging level but first a framework for enabling a variable number of dissimilar applications to easily exchange information without the need for standardising on a specific programming language or platform or the creation of complex interfaces, will briefly be addressed.

2. WEB SERVICES DEFINED

Web services define a loosely-coupled modular approach to exposing application functionality (services) to other applications, that is both language neutral and platform independent (Berlind, 2002). Web services are based on XML, a standardised language for expressing data that enables data to be easily exchanged across different platforms (Microsoft, 2003). Web services facilitate integration by making use of industry-standard protocols such as XML, SOAP, WSDL and UDDI (Berlind, 2002).

Simple Object Access Protocol (SOAP) can be described as a protocol neutral messaging mechanism for sending data from one application to another (O' Neill et al, 2003).

In order for an entity to be able to invoke a Web service from any given location on a network (e.g. the Internet), it needs to know the type of communications protocol required to do so, the address of the Web service on the network, any encoding schemes that may need to employed in the service request message and any other details that the Web service provider may have stipulated. The requestor would also need to have an idea of the methods offered by the service and any parameters that would need to be supplied to these methods. Web Services Description Language (WSDL) is the means of expressing this information to a requestor in a standard format (XML) (Sun Microsystems Inc, 2004).

Universal Description, Discovery and Integration (UDDI) is implemented to serve a Web Service registry function that enables Web service providers to publish information about available Web services so that these services can be discovered by web service requestors. Requestors obtain the necessary information about how to use the Web service from the UDDI registry. This information is usually in the form of a WSDL document (Manes, 2002).

As previously mentioned, this article will focus on a number of specifications that attempt to address the security requirements of heterogeneous business-to-business interactions. Now that Web services and their chief supporting technologies have been defined, it is necessary to discuss some of the current Web service security requirements and concerns.

3. TODAY'S MAIN WEB SERVICES SECURITY CHALLENGES

Web services present many challenges, among them are the following:

1. The identity of the original service requestor needs to be known by a Web service.
2. Persistent security is required for multi-hop SOAP messages.
3. Web services security should be independent of the underlying communication protocols. (O'Neill et al, 2003).

These challenges will now be explained in more detail:

3.1. The identity of the original service requestor needs to be known by a Web service.

A Web service may be requested on behalf of a requesting entity. In such a scenario the identity of the original requestor and any associated attributes may not be conveyed to the Web service, instead data pertaining to the identity of the 'proxy' may be passed to the Web service. There is a requirement for the original requestor to be 'known' to the Web service. The conveyed message needs to include a representation of the identity and other required data pertaining to the original requestor.

3.2. Persistent security is required for multi-hop SOAP messages.

No security 'intermissions' should exist when a SOAP request is sent by a Web service requestor to a Web service, specifically when the message is routed to a number of other points (intermediaries) along the way before reaching the intended recipient (endpoint). Security technologies like SSL (Secure Sockets Layer) provide point-to-point exchanges; this is sufficient within a simplistic requestor to recipient, recipient to requestor security context but does not address a multi-hop message scenario where the security of specific data needs to be preserved right through to its destination (end-to-end security), regardless of any 'short stops' along the way.

3.3. Web services security should be independent of the underlying communications protocols.

It should not be taken for granted that SOAP messages would necessarily be sent over HTTP. SOAP is intended as a protocol neutral data wrapper and any proposed Web service security specification should ensure the security of SOAP messages over any standard communications protocol.

These security challenges are an expression of real world Web services security requirements that need to be addressed by any Web service security model that is to be considered as being a comprehensive security solution. The success of the model would depend on whether it can address these 'real world' requirements and whether what is proposed is practical to implement. A Web service security specification that attempts to offer a 'realistic' Web services security framework is WS-Security.

4 WS-SECURITY

WS-Security defines additional SOAP elements that can be included within the SOAP message header to provide persistent integrity and confidentiality in message exchanges between Web services applications (Apshankar, 2002). The specification also defines and describes how security tokens (which may be used for authentication and authorization) may be embedded into SOAP messages (O' Neill et al, 2003). A security token can be defined as a representation of a claim about an entity (principal). A claim is a statement (assertion) about an entity that may originate from the entity itself or from a trusted claim issuing authority (O'Neill et al, 2003).

WS-Security facilitates integration and interoperability between different security implementations by supporting a wide variety of security technologies, thereby enabling businesses to leverage their existing security investments for use in a business-to-business application integration model as well as being extensible enough to cater for the newer security technologies (Atkinson et al, 2002). It can thus be said that WS-Security offers a level of abstraction above the various models that may be implemented at various business entities (O'Neill et al, 2003). WS-Security also provides end-to-end security at the messaging level by defining how SOAP message integrity and confidentiality is preserved as a message is routed between intermediaries to the intended Web services.

Before we delve into the more technical aspects of WS-Security it is useful to mention that WS-Security is one of a number of other specifications outlined in a security roadmap for securing Web services created by IBM and Microsoft (Apshankar, 2002). These specifications include WS-Policy, WS-Trust, WS-Privacy, WS-SecureConversation, WS-Federation and WS-Authorisation. Together with WS-Security, the security roadmap specifications form a type of pyramid model, with some specifications being based on those that precede them and all the specifications being inter-related to form a comprehensive Web service security infrastructure model. The diagram below represents this.

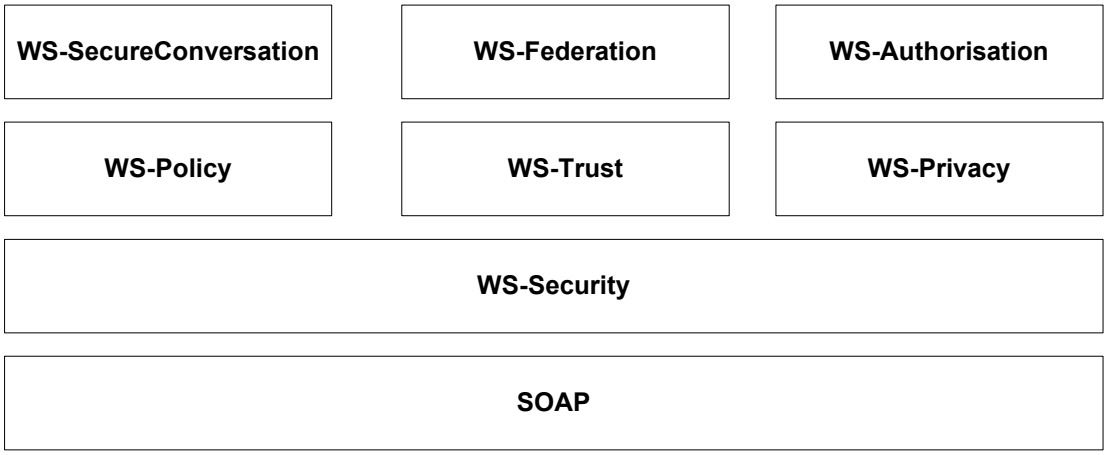

Figure 1 IBM and Microsoft Web services security specifications (O'Neill et al, 2003)

Each of the additional specifications will now briefly be described (O'Neill et al, 2003):

4.1 WS-Policy

Web service providers can describe the security requirements for successfully invoking their Web services using WS-Policy. Such requirements could include the types of encryption and digital

signature algorithms as well as supported security tokens that need to be applied within SOAP messages requesting services. The means for attaching (binding) these security requirements to a SOAP request is also specified.

4.2 WS-Trust

WS-Trust defines means for establishing trust relationships between entities. These trust relationships can either be established directly or brokered, in which case a trust proxy is used in order to request security tokens based on WS-Policy information. These tokens would be requested and obtained from some token issuer and then inserted into a SOAP request by the trust proxy on behalf of the requestor. The tokens would be digitally signed and encrypted to ensure their integrity and confidentiality en-route to a Web service.

4.3 WS-Privacy

WS-Privacy can be used to communicate privacy policies created by organizations for their Web services. SOAP requests for Web services would need to indicate conformance to these policies. This conformance would take the form of security tokens embedded using WS-Security. WS-Trust would evaluate the requests by comparing embedded privacy claims (within tokens) against privacy statements expressed within WS-Policy descriptions.

4.4 WS-SecureConversation

This specification allows for the creation of a secure 'session' between a Web service requestor and a Web service through a process of dual authentication using SOAP messages. This 'session' includes the derivation of session keys. WS-SecureConversation enables a Web service to receive more than one SOAP request over a session without having to evaluate embedded security tokens against policy statements every time a message is received. WS-SecureConversation is based on WS-Security and WS-Trust.

4.5 WS-Federation

WS-Federation defines how trust relationships are managed in a heterogeneous security environment. Using WS-Federation, a requestor authenticated to one Web service that consumes a particular type of authentication token can automatically be authenticated to another Web service that may consume a different authentication token, assuming that the two services have a trusted relationship. WS-Federation is based on WS-Security, WS-Policy, WS-Trust and WS-SecureConversation.

4.6 WS-Authorisation

WS-Authorisation enables a means for defining and managing access-control policies for Web services. It does not mandate the use of specific authorisation mechanisms, but is flexible enough to accommodate authorisation based on Access Control Lists (ACLs) and Role Based Access Control (RBAC).

The nature of the relationships between these Web services security specifications are depicted by Fig 1 and described in the preceding specification definitions.

As mentioned in the earlier discussion on WS-Security, this specification allows the association of security tokens with SOAP messages (Apshankar, 2002). These may be binary tokens such as X.509 certificates and Kerberos tickets or username/password combinations or XML-based assertions (Security Assertion Markup Language, SAML). These tokens need to be secured in transit to their intended recipient. The integrity of security tokens and of the SOAP message payload is ensured through the use of XML digital signature to sign the tokens and/or the entire message payload or selected elements of the payload. Confidentiality is implemented using XML Encryption to encrypt security tokens and/or the entire message payload or selected elements of the payload. In this way, only the intended message recipients are able to decrypt tokens and

data for consumption and any unauthorised modifications to tokens and data can be detected by comparing before and after hash values.

5. DOES WS-SECURITY ADDRESS THE SECURITY REQUIREMENTS OF WEB SERVICES?

So far, a basic picture of the WS-Security specification has emerged. It needs to be determined whether WS-Security is able to meet the high-level security requirements of application-to-application interactions across multiple networks. These security requirements include Authentication, Authorisation, Integrity, Confidentiality, Non-repudiation, Availability and Privacy (O'Neill et al, 2003).

WS-Security while defining a means for including one or more authentication and authorization tokens of varying formats within a SOAP message, does not provide a complete Authentication and Authorisation mechanism in an application-to-application interaction model (Apshankar, 2002). Similarly, while defining a means of providing integrity (XML Digital signature) and confidentiality (XML Encryption) at the messaging level, it does not stipulate how PKI functions are to be co-ordinated or how Non-repudiation and Auditing services are to be implemented. Web services consumed in a business-to-business application integration model ultimately reside on servers. It is vitally important that all servers participating in message exchanges are secured through the use of devices or software on which granular security policies are defined and all unnecessary entry points are closed. This is especially critical in a federated trust scenario. These are just some of the security considerations, but it is enough to come to the conclusion that WS-Security in isolation cannot be considered an all-encompassing Web services security model. The requirement for the creation of the other IBM and Microsoft roadmap specifications re-inforces this idea.

How then is Web services security to be ensured in this security context? The next section proposes a security approach that includes the use of WS-Security to address this question.

6. A PROPOSED WEB SERVICES SECURITY SOLUTION

The following represents an incipient approach to a possible solution. It is currently in a very primordial stage. As this is a research-in-progress paper, based on a Masters dissertation commenced in February of this year, extensive refinements will be required before the final model is arrived at.

Assume a simplistic business-to-business scenario prevails, where a number of organizations have agreed to expose selected Web services to one another. Each of these organizations could have differing security requirements, in line with the security technologies deployed on their networks. Entities wanting to connect to the exposed services would need to be aware of these to begin with. These requirements could be expressed in a standardised format using WS-Policy. The security policies of participating organisations could be stored centrally on a server. These policies would share a common format but could express different security requirements for the use of certain Web services. Requests to access or modify existing policies would need to be authenticated. Similarly, requests to publish policies to the server would also need to be authenticated. An authorisation mechanism would also need to be employed to ensure that authorised entities are only granted the access control rights that they require so that their range of operations is appropriately limited. A standard would need to created and communicated to all relevant parties, specifying the type of authentication and authorization tokens to be provided by entities trying to access the policy server. These tokens would be embedded and secured using WS-Security. It is important that timestamp

elements with a narrowly defined time window be included in each SOAP message header to avoid the occurrence of replay attacks. It is important that timestamp elements are encrypted to prevent tampering (Seely, 2002). This applies to all SOAP messaging in this model.

Once an entity has been successfully authenticated and authorized, it should have the option to view selected WSDL documents for Web services offered by its trading partners. These WSDL documents can be stored on a separate server, which can have a network firewall configured to allow only requests originating from the policy server through to the server.

Once a requestor has downloaded a policy, a SOAP message with the appropriate WS-Security elements can be constructed by the requestor or a proxy to be sent to a service provider. SOAP requests to services must at a minimum, provide proof that the message sender is indeed the requestor and that the requestor is a valid requestor. This can be provided by embedding a signed X.509 digital certificate or Kerberos ticket in the message using WS-Security and XML Digital Signature. The "proof" needs to be kept confidential; again WS-Security using XML Encryption can be employed here. SOAP messages may be altered by intermediaries. XML processors deployed on these intermediaries may remove white space between XML elements or add their own elements to the message. This would obviously render any applied digital signatures invalid. It is recommended that an exclusive canonicalisation method be applied to the XML data before it is passed to a digest algorithm and again at the recipient, before the digest is recalculated (O'Neill et al, 2003).

At the recipient Web server, once authentication and authorization tokens have been validated and the relevant data has been decrypted, the data should pass through an application-level firewall that has the appropriate authorization and application rules for Web services dynamically configured, (the firewall should be automatically updated as the Web services change) to validate that any parameters being passed to a service, are the expected parameters and that the parameters are of the correct data type and length. This is to prevent denial-of-service and buffer overflow attacks. Once valid data has been passed to the Web service/s being requested and the request has been processed, the response data should also be validated and then sent to the requestor using WS-Security over a secure channel. If unauthorised operations are detected at the recipient, it may not be wise to propagate error messages through to the requestor, as attackers could use the information included in error messages to attempt to launch attacks. IPSec can be used to ensure transport-level security throughout the model.

7. CONCLUSION

This article posed a question about the use of WS-Security in a heterogeneous business-to-business environment. Certainly WS-Security is ideal for the provision of Web services security in a heterogeneous security landscape. Through its support for the usage of various security technologies within SOAP messaging, the problem of security interoperability will be addressed in the final model. Hopefully, more meaningful statements will emanate from my final model. However, one singular fact gleaned from my work to date is that the WS-Security specification should be used for security in a business-to-business model but it should not be perceived as a complete Web services security solution.

REFERENCES

Apshankar, K. (2002, July 24). *WS-Security: Security for Web Services.* Retrieved February 18, 2003 from http://www.webservicesarchitect.com/content/articles/apshankar04.asp

Atkinson B., Della-Libera G., Hada S., et al. (2002, April). *Specification: Web Services Security (WS-Security) Version 1.0 05 April 2002.* Retrieved December 8, 2003, from

http://www-106.ibm.com/developerworks/webservices/library/ws-secure/

Berlind, D. (2002, February 11). *What are Web services anyway?* Retrieved March 15, 2004, from

http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2846997,00.html

DataPower Technology, Inc. (2003, May). *Essential XML Web Services Security Best Practices*. Retrieved August 12, 2003, from

http://www.datapower.com/docs/XML_WSS_BP0503.pdf

Manes, A. (2002, February 1). *What is UDDI?* Retrieved March 15, 2004 from http://searchwebservices.techtarget.com/ateQuestionNResponse/0,289625,sid26_cid446775_tax289201,00.html

Microsoft Inc. (2003, May 15). *What are Web Services?* Retrieved December 8, 2003, from http://www.microsoft.com/net/basics/webservices.asp

O' Neill et al. (2003). *Web Services Security.* McGraw-Hill/Osborne.

Sun Microsystems Inc. (2004). *Web Services Description Language (WSDL): An Intuitive View.* Retrieved May 12, 2004, from http://java.sun.com/dev/evangcentral/totallytech/wsdl.html