Asian Journal of Information Technology 6 (6): 674-680, 2007

The Web Services Securing: Basic Technologies

[1]Marzouk S. Mokbel, [1]Maan Younis Abdullah and [2]Le Jiajin
[1]School of Computer Science and Technology, University of Donghua
[2]School of Computer Science and Technology,
Central South University, Hunan-Changsha, China

Abstract: Securing the web services is of major concern when implementing critical business transaction with web services. A problem with web services security standards are appeared due to the fact that several organizations are involved in developing such standards caused many possibilities of unsecured usage. The present work summarizes the security model for WS-security and its related specifications. The most important one is the WS-security which defines a message-based security model for WS-S that is suitable for achieving end-to-end security in environments with multiple trust domains. Additionally, WS-security describes how to encode binary security tokens. Specifically, the specification describes how to encode X.509 certificates and Kerberos tickets as well as how to include opaque encrypted keys. This study also introduces the different levels of security and gives an overview of the current status of these concepts concerning standardization.

Key words: Web services, WS-security, HTTP, PKI, XML signature, XML encryption

INTRODUCTION

Security is a key factor for companies in adopting web services in their critical business transactions. Without security cared, companies could not use web services in the insecure internet environment. The loose coupling of services-the web services' emblem-can play an active part in this insecure environment (Takeshi and Michiaki, 2005). WS-security is flexible and designed to be used as the basis for the construction of a wide variety of security models including PKI, Kerberos and SSL. Specifically, the WS-security provides support for multiple security tokens, multiple trust domains, multiple signature formats and multiple encryption technologies. Some security aspects of web services are currently being standardized in OASIS. For example, the WS-security specification describes how to use existing W3C security recommendations such as XML signature and XML encryption, to ensure the integrity and confidentiality of SOAP messages. Another work also describes how the existing digital credentials and their associated trust semantics can be securely associated with SOAP messages (Abbie and Carling, 2003). Like this specifications can form the foundations of the overall architecture that can be used to secure web services. However, the main goal of this document is to introduce the fundamental principles of the proposed security model and to illustrate its benefits. To be successful in business scenarios, the web services have to be suitable for secure communication. Yet the original SOAP specification contains no solutions to solve the security problem. This study introduces the different levels of security and gives an overview of the current status of these concepts concerning standardization. The present research is aimed firstly to provide an overview of the standards for the basic technologies, such as SSL HTTPS, HTTPD, SPKM and PKI and secondly to introduce the basic technologies and motivates supports for secure web service implementation and security architecture that can be used at various layers in the network to ensure the security of the network and services.

TECHNOLOGIES FOR SECURING WEB SERVICES

In this study, we discuss the basic technologies of security architecture of Web Services like SSL, Reliable HTTP, Secure HTTP, SPKM, X.509 and anther technology.

Secure sockets layer: The Secure Sockets Layer (SSL) is industry standard security protocol. It was developed by Netscape and is supported by netscape and internet explorer. Usually, web sites that need to collect confidential information from their customers, such as credit card numbers for online purchases, use SSL.

- SSL provides point-to-point security or operates between end-points (and not applications), but for web services the end-to-end security is needed in which multiple intermediate nodes could exist

Corresponding Author: Marzouk S. Mokbel, School of Computer Science and Technology, University of Donghua, China

between the two end-points. In a web services environment, multiple XML-based business documents going through multiple intermediary nodes could be available and it will be difficult for such nodes to participate in security operations in an integrated fashion.

- SSL operates at the transport level and not at the message level. In other words, messages are protected only while in transit. It means saving message later to prove that it hasn't been modified can't be performed.
- SSL doesn't support nonrepudiation. Using SSL, a communicating partner can't prove that the other party has performed a particular transaction. Whereas, SSL doesn't support an end-to-end audit trail from service request to service response.
- SSL doesn't support element-wise signing and encryption. Giving larger XML order document, user may want to only sign or encrypt the credit card information and that is difficult in SSL. This is due to the reason that, SSL is a transport-level security scheme as opposed to a message-level scheme (Fensel, 2000).

Standards-based solutions: Interpeak is committed to provide standards-based security and TCP/IP solutions. An extensive testing is performed to verify that interpeak SSL is fully compliant with other important standard SSL applications from companies such as Microsoft and Netscape (Fig. 1). The SSL protocol implementation and its underlying cryptographic components and utilities comply with all the relevant security standards and specifications. In fact, as open SSL is used in both the Apache Web Server and the Opera Browser, it is literally tested millions of times each day. It is thereby, guaranteed to be fully compliant with the standards and products that adhere to it, e.g., browsers such as Netscape and Microsoft Internet Explorer. Interpeak SSL also strictly adheres to all the standards involved for the numerous algorithms, ciphers, certificates, etc., which are included in the Interpeak SSL release (ISNS, 2005).

Secure HTTP, Reliable HTTP and HTTP digest

Secure HTTP: Secure HTTP (HTTPS or S-HTTP) is a secure message-oriented communications protocol designed for use in conjunction with HTTP. S-HTTP is a superset of HTTP, which allows messages to be encapsulated in various ways. Encapsulations can include encryption, signing, or MAC based authentication. This encapsulation can be recursive and a message can have several security transformations applied to it. Also, there are several proposed extensions to HTTP that cannot be represented `within' another protocol, such as chunked-transfer-encoding and multiplexing. SEA attempts to bring S-HTTP's logical security model into HTTP/1.x development.

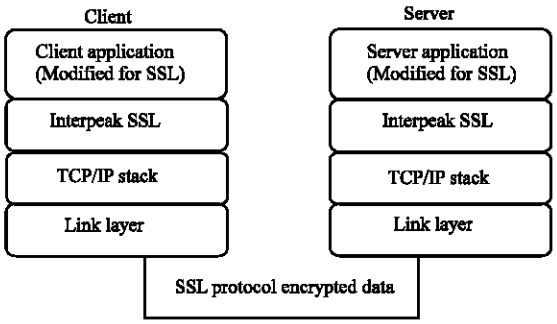

Fig. 1: SSL Client/Server application using Interpeak SSL for secure Internet communication

Reliable HTTP: Reliable HTTP (HTTPR) is a protocol which offers a reliable delivery of HTTP packets between server and client. This solves a number of issues that are evident in current HTTP and opens the way to reliable messaging between web services. HTTPR is a protocol for the reliable transport of messages from one application program to another over the internet, even in the presence of failures either of the network or the agents on either end. It is layered on top of HTTP. Specifically, HTTPR defines how metadata and application messages are encapsulated within the payload of HTTP requests and responses. HTTPR also provides some protocol rules which make it possible to ensure that each message is delivered to its destination application exactly once or is reliably reported as undeliverable.

HTTP digest: HTTP digest authentication defines a protocol which allows the client to prove to the server that it knows the correct password without sending the password itself to server. The client does an irreversible computation, using the password and a random value supplied by the server as the input values. The result is transmitted to the server who performs the same computation and authenticates the client if arrives at the same value. Since the computation is irreversible, an eavesdropper can't obtain the password (However, it should be mentioned that the matter is actually more complicated than that simplified above).

Simple PublicKey Mechanism (SPKM): The Simple Public-Key GSS-API Mechanism (SPKM) protocol is employed by peers implementing the Generic Security Service Application Program Interface (GSS-API) when

using SPKM. The SPKM is based on a public-key, rather than a symmetric-key, infrastructure. SPKM provides authentication, key establishment, data integrity and data confidentiality in an online distributed application environment using a public-key infrastructure (MSDN Library, 2005). The SPKM is an instance of the latter type of document and is therefore termed a "GSS-API Mechanism". This mechanism provides authentication, key establishment, data integrity and data confidentiality in an on-line distributed application environment using a public-key infrastructure. SPKM can be used as a drop-in replacement by any application which makes use of security services through GSS-API calls (for example, any application which already uses the Kerberos GSS-API for security). The use of a public-key infrastructure allows digital signatures supporting non-repudiation to be employed for message exchanges and provides other benefits such as scalability to large user populations (Mahmoud, 2005).

Kerberos: Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/ server applications by using secret-key cryptography. Kerberos was created by MIT as a solution to these network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to server (and vice versa) across an insecure network connection. If the client and server are used Kerberos to prove their identity, then they can also encrypt all of their communications to assure privacy and data integrity as they go about their business. Kerberos is freely available from MIT, under copyright permissions very similar to those used for the BSD operating system and the X Window System. MIT provides Kerberos in source form so that anyone who wishes to use it may look over the code for themselves and assure themselves that the code is trustworthy. In addition, for those who prefer to rely on a professionally supported product, Kerberos is available as a product from many different vendors (Khare, 1996).

Public key infrastructure: Public Key Infrastructure (PKI) is the term generally used to describe the laws, policies, standards and software that will regulate or manipulate certificates and public and private keys. The primary function of PKI is to allow the distribution and use of public keys and certificates with security and integrity. A PKI is a foundation on which other applications and network security components are built. Systems that often require PKI-based security mechanisms include email, various chip card applications, value exchange with e-commerce (e.g., debit and credit cards), home banking and electronic postal systems. A PKI has many usages and applications. A PKI enables the basic security services for such varied systems as:

- SSL, IPsec and HTTPS for communication and transactional security;
- S/MIME and PGP for email security;
- SET for value exchange;
- Indentures for B2B (Joel and Sun, 2001).

X.509: The X.509 standard defines what information can go into a certificate and describes how to write it down (the data format). In order to ensure a consistent processing model across all types supported by WSS: SOAP Message Security, the <wsse:SecurityToken Reference> element shall be used to specify all references to X.509 types in signature or encryption elements that comply with this profile.

Web services security standards: In this study an overview of the web services security will be submitted. The WS-security developed by IBM, Microsoft and Verisign, enhances SOAP with methods used to protect message integrity and confidentiality and to exchange security information. The WS-security standard for web services was ratified by OASIS on 2004 (Lannon, 2005). The web services security is also developed by Microsoft and IBM in order to provide core facilities for protecting the integrity and confidentiality of a message as well as mechanisms for associating security-related claims with the message (Westbridge Technology, 2003). The web service security challenge is to understand and assess the risk involved in securing web-based service today, based on the existing security technology and at the same time track emerging standards and understand how they will be used to offset the risk in new web services (Hondo *et al*., 2002). The standard describes enhancements of SOAP messaging in order to provide quality of protection through message integrity, message confidentiality and single message authentication (Lannon, 2005). In the study a description of how XML digital signatures and encryption can be exploited to achieve a level of trust is shown. The WS-security describes how to attach signature, encryption and security tokens to SOAP messages Specifically, the WS-Security profile specifications describes how to encode Username Tokens, X.509 Tokens, SAML Tokens, REL Tokens and Kerberos Tokens as well as how to include opaque encrypted keys as a sample of different binary token types.

XML security standards: XML schemas convey the data syntax and semantics for various application domains,

such as business-to-business transactions, medical records and production status reports. However, these schemas seldom address security issues, which can lead to a worst-case scenario of systems and protocols with no security at all. At best, they confine security to transport level mechanisms such as Secure Sockets Layer (SSL). On the other hand, the omission of security provisions from domain schemas opens the way for generic security specifications based on XML document and grammar extensions. These specifications are orthogonal to domain schemas but integrate with them to support a variety of security objectives, such as confidentiality, integrity and access control (Martin, 2003). Three of these are pretty mature already and shall therefore be presented within this paper stated as:

- XML signature-a standard supports various digital signature configurations (W3C recommendation)
- XML encryption-a standard supports different encryption types (W3C recommendation)
- XML Key Management Specification 2.0 (XKMS)-a collection of protocols for key management via web service (W3C working draft).

In addition, there are several other mechanisms which are explained in the following.

The Security Assertion Markup Language (SAML) standard defines a framework for exchanging security information between online business partners. It was developed by the Security Services Technical Committee (SSTC) of the standards organization OASIS (the Organization for the advancement of Structured Information Standards) (Hughes *et al*., 2005). Some of SAML objectives create an authentication and authorization exchange mechanism that is protocol-and platform-independent (also defined as Single Sign-On, or SSO), this should be independent of the deployment environment and should work with centralized, decentralized and federated deployment scenarios, the SAML framework should be XML-based (Fensel, 2000).

XML signature: Basically, an XML signature is comprised of four main components or elements: <SignedInfo>, <SignatureValue>, <KeyInfo> and <Object>. The <SignedInfo> element includes all of the contents or resources to be signed with each item having a corresponding <Reference> element, which identifies the content and a digest over it. The <Reference> elements are digested and cryptographically signed in a manner similar to signing when using a standard digital signature. The resulting signature value is stored in the <SignatureValue> element. The <KeyInfo> and <Object> elements are optional. XML digital signatures are represented by the Signature element, which has a structure given as in Fig. 2.

```
■ * Represents zero or more occurrences.
■ + Represents one or more occurrences.
■ ? Represents zero or one occurrences.
<Signature ID>
  <SignedInfo>
    <CanonicalizationMethod/>
      <SignatureMethod/>)
        (<Reference URI?>
          (<Transforms>)?
            <DigestMethod>
              <DigestValue>
            </Reference>)+
          </SignedInfo>
        <SignatureValue>
      (<KeyInfo>)?
    (Object ID?)*
</Signature>.
```

Fig. 2: XML digital signature structure

```
<EncryptedData Id? Type? MimeType? Encoding?>
  <EncryptionMethod/>?
   <ds:KeyInfo>
    <EncryptedKey>?
     <AgreementMethod>?
      <ds:KeyName>?
        <ds:RetrievalMethod>?
        <ds:*>?
        </ds:KeyInfo>?
       <CipherData>
      <CipherValue>?
     <CipherReference URI?>?
   </CipherData>
  <EncryptionProperties>?
</EncryptedData>.
```

Fig. 3: XML encryption

XML encryption: The second major XML security specification from the W3C is the XML encryption. The standard provides a flexible, XML based methodology for encrypting XML documents and for representing encrypted data in a XML format. It builds on the concepts originally created within XML signature (Fig. 3). The standard defines the following major components:

Granularity definitions for describing what will be encrypted; Syntax for representing encrypted data in an XML document and processing rules that describe how to encrypt and decrypt XML documents (Konstantin *et al*., 2005). XML encryption supports the encryption of an entire XML document or only selected portions of an XML document. It supports the super-encryption of data. That is, already encrypted data can be encrypted. The XML encryption also provides for the identification or transfer of decryption key information (Konstantin *et al*., 2005).

XMLKey Management Specification 2.0 (XKMS): The XML Key Management Specification (XKMS) is described as a web service which provides an interface

between XML application and Public Key Infrastructure (PKI). XKMS greatly simplifies the deployment of enterprise strength public key infrastructure by transferring complex processing tasks from the client application to a trust service (Phillip *et al*., 2001).

WS-security: The WS-security specification (OASIS, 2004) addresses single-message, end-to-end security. The web services security model is shaping up quite significantly. A new series of specifications explain how web services security can be implemented in a platform-independent and loosely-coupled manner in terms of establishing secured communications, defining policies for how services interact and defining rules of trust between domains of services (Box *et al*., 2002). WS-security provides a general-purpose mechanism for associating security tokens with messages. No specific type of security token is required by WS-security. It is designed to be extensible (e.g., support multiple security token formats). For example, a client might provide proof of identity and proof that they have a particular business certification (Schneier, 2001). Also WS-security describes a mechanism for encoding binary security tokens. Specifically, the specification describes how to encode X.509 certificates and Kerberos tickets as well as how to include opaque encrypted keys. It also includes extensibility mechanisms that can be used to further describe the characteristics of the security tokens that are included with a message (SWSW, 2002). The mechanisms described in the standard can be used in a multitude of security and encryption models and technologies. WS-security also provides a general-purpose mechanism for extensible association of security tokens with messages. This is important in situations where a client has to provide identification and proof for a specific business certification (Lannon, 2005).

Web services policy framework (WS-Policy): The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe and communicate the policies of a web service. WS-Policy defines the basic set of construction that can be used and extended by other web services specifications to describe a broad range of service requirements, preferences and capabilities (Box *et al*., 2002).

WS-policy provides a flexible and extensible grammar for expressing the capabilities, requirements and general characteristics of entities in XML web services-based system. WS-Policy defines a framework and a model for the expression of these properties as policies. Policy expressions allow for both simple declarative assertions as well as more sophisticated conditional assertions (Box *et al*., 2002). The policy framework provides a set of XML structure elements to indicate how a domain specific web service will express the associated policy. It might for example specify that the service in question can only be used with a security token of type Kerberos (Konstantin *et al*., 2005).

Web Services Trust language (WS-Trust): The recently updated Web Services Trust Language (WS-Trust) uses the secure messaging mechanisms of WS-security to define additional primitives and extensions for security token exchange to enable the issuance and dissemination of credentials within different trust domains (Phillip *et al*., 2001). As its name suggests, the WS-Trust is motivated by more than enabling interoperability between the multiple formats for security tokens that might be used in a WS-security protected message. It also addresses the issue of trust interoperability. Even if a given security token's format is acceptable to a recipient of a WS-Security-protected SOAP message, interoperability at the syntax level is no guarantee that the recipient will be able to trust the token. For example, for a SOAP Service to support Kerberos tokens does not mean that it would be able to accept Kerberos tickets from arbitrary Kerberos Key Distribution Centers the service would not have the necessary trust (in the form of shared symmetric keys) with these KDCs in order to decrypt and verify such tickets (Box *et al*., 2002).

Web services federation language (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 (SWSW, 2002). WS-Federation differentiates between *passive* and *active* requestor profiles. Passive requestors might for example be web browsers supporting HTTP whereas active ones would be SOAP-enabled applications. The idea-for both of these-then is to give a user the ability to browse through and access a sequence of applications without having to sign on every single time he accesses another related application. This can be achieved by providing a federated single sign on identity to the user. Realizing this can be done

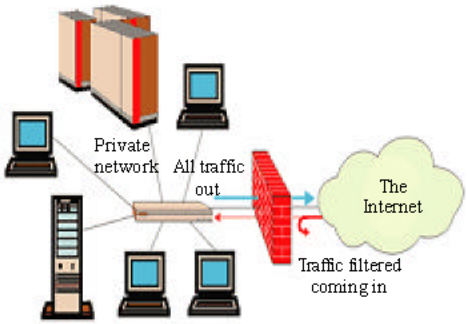

Fig. 4: The operation of firewall

by building upon the mechanisms discussed earlier in this study, e.g., by using STSs to issue valid security tokens (Berin, 2004).

Web services firewalls: A firewall is a system that implements and enforces an access control (or security) policy between two networks; it usually guards an internal private network from an external public one, isolating an intranet from the internet. Essentially a firewall connects two or more networks but only allows specified forms of traffic to flow between them. The firewall is a means by which a security policy can be enforced. Web services firewall protects against attacks targeting XML, SOAP and WSDL applications. The web services firewall leverages imperial's dynamic profiling technology to create a dynamic positive security model of allowed application usage and structure, including XML URLs, SOAP actions, XML elements and XML attributes (SSAS, 2006). Figure 4 demonstrates graphically the operation of firewall in protecting the physical boundaries of a network: There is a physical boundary of the private network and the only way to get into the network is through the firewall. While packets of network traffic and messages pass through a firewall, they are authenticated and checked for possible intrusion or malicious attacks. Only this brief explanation of firewalls will be presented here, intended to emphasize their relatively coarse-grained protection (Madsen, 2003).

CONCLUSION

This study has given a short overview about the technologies used for securing web service messages. WS-security provides a flexible and extensible framework to support a variety of security scenarios. The study provides a review of current efforts on developing security techniques for web services, presenting an integrated security architecture which can be used by organization to secure web services. The XML signature and XML encryption are mature standards, implemented in a large number of applications and libraries and used as core building blocks for other standards. The ability to supply a complete web service environment, in which risk assessment and policy enforcement are an integral component, will depend on several initiatives continuing to evolve as de facto standards. First, the workflow needs an integrated security model as part of its processing model. Second, analysis is needed to determine whether XML schemas can be used to formalize security models through the definition of security types. As web services are applied more broadly, as application topologies continue to evolve to support intermediaries such as firewalls, load balancers and messaging hubs and as awareness of the threats organizations face becomes more well understood, the need for additional security specifications for web services grows clear. Security for web services is a necessity and can be deployed.

REFERENCES

Abbie, B. and A. Carling, 2003. Web services security: An enabler of semantic web services, K2h 8E9, Ontario, Canada.

Berin, L., 2004. Introduction to XML Encryption and XML Signature, J. Inf. Security Technical Report, 9: 6-18.

Box, D., F. Curbera, M. Hondo, C. Kaler, D. Langworthy and A. Nadalin *et al*., 2002. Web Services Policy Framework (WS-Policy) Version 1, BEA Systems Inc., International Business Machines Corporation, Microsoft Corporation, SAP AG, USA.

Fensel, D., 2000. The Semantic Web and Its Languages, IEEE. Intelligent Sys., 15: 67-73.

Hondo, M., N. Nagaratnam and A. Nadalin, 2002. Securing Web services, IBM. Sys. J., 41: 288-241.

Http://nwc.securitypipline.com/

http://xml.coverpages.org/ni2002-04-11-b.html

Hughes, J., A. Origin and E. Maler, 2005. Security Assertion Markup Language (SAML) 2.0, Technical Overview report, sstc-saml-tech-overview-2.0-draft-03.http://www.oasis-open.org/committees/documents.php?wg_ abbrev=security.

Interpeak Secure Networking Software (ISNS), 2005. Secure Socket Layer Version 1.21-r5. InterpeakAB. www.interpeak.com/products/ssl. html.

Joel, W. and P.S. Sun, 2001. Public Key Infrastructure Overview, Sun Microsystems, Inc. California, U.S.A.

Khare, R., 1996. A Security Extension Architecture for HTTP/1.x,http://www.w3.org/TR/WD-http-sea.html#_Toc345790647.

Konstantin, B.A., J. Donald, B. Flinn, K.C. Shirley and H.D. Bret, 2005. Introduction to Web services and their security, J. Inf. Security Technical Report, 10: 2-14.

Lannon, R., 2005. Security in a web services world, Principal Security Consultant, Phoenix Group. Network Computing USA.

Madsen, P., 2003. WS-Trust: Interoperable Security for Web Services, Internet Report.

Mahmoud, Q.H., 2005. Securing Web Services and the Java WSDP 1.5 XWS-Security Framework, Sun Microsystems Inc. http://java.sun.com/developer/technicalArticles/WebServices/security/

Martin, N., 2003. Standards for XML and Web Services Security, ABB Corporate Research No. Dept. of Computer Science, University of Maryland at College Park, Security-Computer, pp: 96-98.

MSDN Library, 2005. Simple Public-Key GSS-API Mechanism (SPKM), http://msdn2.microsoft.com/en-us/library/ ms818754.aspx

Phillip, M., B. Hallam and F. Warwick, 2001. XML Key Management Specification (XKMS).VeriSign, Microsoft, web methods. http://xmltrustcenter.org/

Schneier, B., 2001. Secrets and Lies: Digital Security in a Networked World, John Wiley and Sons.

SecureSphere Appliance Specifications (SSAS), 2006. Web Application Firewall the Industry's Only Automated Web Application. Firewal lwww.imperva.com

Security in a Web Services World (SWSW), 2002. A Proposed Architecture and Roadmap A joint security whitepaper from IBM Corporation and Microsoft Corporation. http://www.eitsec.co.uk/firewall Deployment.asp, Firewall deployment.

Takeshi, I. and T. Michiaki, 2005. Patterns for Securing Web Services Messaging, IBM Research, Research Laboratory, Shimotsuruma, Yamato-shi, Tokyo, Japan, pp: 1623-14.

Westbridge Technology, W.B., 2003. XML Application Firewalls for Securing and Monitoring XML Web Services, http://www.westbridgetech.com.