New Proposed Method For Web Services Security

H. J. Muhasin
Department Of Computer Science, College Of Education Ibn AL-Haitham,University of Baghdad

Abstract

The Web service security challenge is to understand and assess the risk involved in securing a web-based service today, based on our existing security technology, and at the same time track emerging standards and understand how they will bc used to offset the risk in new web services. Any security model must illustrate how data can flow through an application and network topology to meet the requirements defined by thc business without exposing the data to undue risk. In this paper we propose a mechanism for the client to provide authentication data, based on the service definition , and for the service provider to retrieve those data . We also show how XML Digital Signatures and encryption can be exploited to achieve a level of trust .

Introduction

By security we mean the protection against unauthorized reading ,modification , or destruction of information . Obviously, there are many degrees of security, we consider here the level required for e-commerce where a system penetration could mean the loss of large amounts of money and of business prestige. There will also need to be a model for security, reliable messaging, quality of service, and management for the web services stack. The web service security challenge is to understand and assess the risk involved in securing a web-based service today based on our 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 . Any security model must illustrate how data can

flow through an application and network topology to meet the requirements defined by the business without exposing the data undue independent risk. A web service security model must support protocol -declarative security policies that web service providers can enforce and descriptive security policies attached to the service definitions that clients can use in order to securely access the service .

Web Services

One of the anticipated benefits of web services is that application – to – application communication can replace the humanend – user interaction in the current data entry web applications . web service standards have been defined to allow enterprise companies to rapidly define simple business – to – business interfaces and exchange messages . Additional requirements for a web services infrastructure include support for service context , conversation and activities , intermediaries , portal integration , and workflow and service flow management .

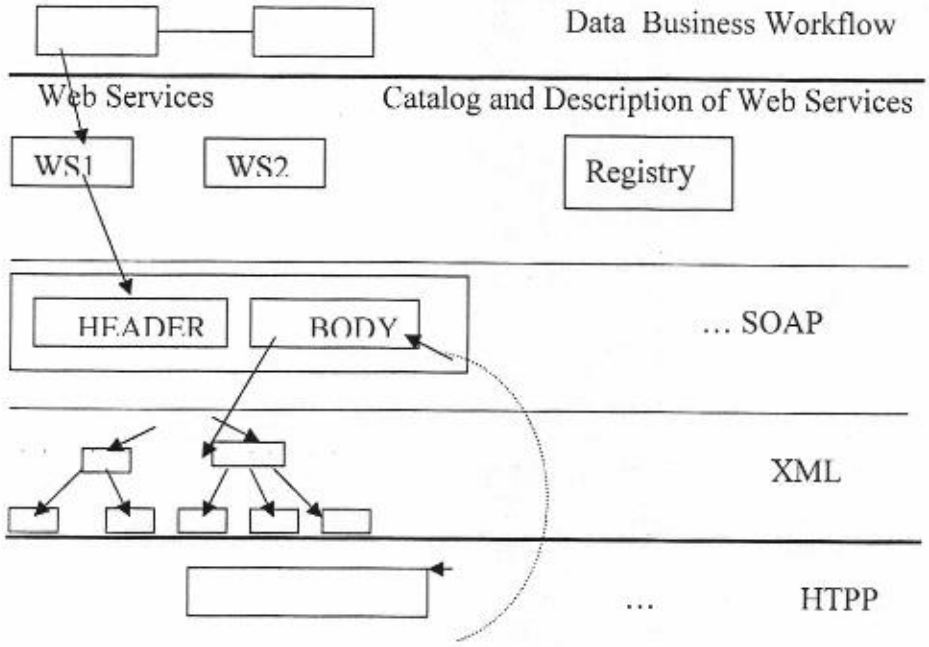
XML : Extensible Markup Language (XML)(1) began as an extensible data format to address the limitations of HTML (Hyper Text MarkupLanguage) , but has become a standard for communication between applications .With XML, an application defines markup tags to represent the different elements of data in text file so that the data can be read and processed by any application that uses XML .

using By XML , its possible for business persons to define policies and express them as XML documents . These documents can have sections that are encrypted and the documents themselves can be digitally signed , distributed , and then interpreted by the security mechanisms that configure the local software . This allows various implementations to map from the XML description to a local platform- specific policy enforcement mechanism without requiring changes to the infrastructure .

SOAP : Simple Object Access Protocol (SOAP) (2) is a simple ,lightweight , and extendable XML – based mechanism for exchanging structured data between network applications on top of widely used internet standards such as XML and HTTP (Hyper Text Transfer Protocol). SOAP consists of three parts :

an envelope that defines a framework for describing the contents of a message, a set of encoding rules for expressing instances of application – defined data types , and a convention for representing remote procedure calls (RPC) and responses .

The SOAP envelope is defined in XML and enables a large variety of meta – information to be added to the message, such as transactioni dentifiers , message routing information , and message security . The envelope consists of two parts : header and body . The header is a genenic mechanism for adding features to SOAP message . Allimmediate child elements of the SOAP header element are called header entries . The body is a container for application data intended for the ultimate recipient of the message . Thus , SOAP can be considered as another layer , between the transport layer (e.g. , HTTP) and the application layer (e.g. , business data) , that is a convenient place for conveying message meta – information . The following diagram shows one of these supporting layers , the HTTP layer :



WSDL: Web Services Description Language (WSDL) (3) is essentially an XML IDL (interface definition language) that provides a way to describe the function and interface of a service . It is a format for describing network services as a set of endpoints operating on messages containing either document–oriented or procedure–oriented information. The operations and messages are described abstractly , and then bound to a concrete network protocol and message format to define an endpoint . Related concrete endpoints are combined into abstract endpoints (services) . WSDL is extensible to allow description of endpoint and their messages regardless of what message formats or network protocols are used to communicate ; however , the only currently described bindings are for SOAP, HTTP GET / POST , and MIME (Multipurpose Internet Mail Extensions) .

The WSDL service information can be extracted from a UDDI(Universal Description , Discovery , and Integration) (4) business service entry, or may be obtained from other service repository sources.

Regardless of the source , both run – time and development tools can use WSDL to determine run – time bindings to a service . This information can be used to build the logic to access the service either directly from the client or through generated code stubs .

WSDL has the potential to be extended (5) to include the definition of the context needed by the business execution environment, including security. Without these extensions, users will make assumptions about security in the run – time environment of a Web service . Defining these
security assertions in XML will allow us to have a common interpretation of security attributes in different implementations . It will also facilitate searching .

Secure Web Services

There are fundamental business reasons underlying the existence of various security mechanisms . The authentication of the entity asserting an identity when requesting a service allows businesses to offer different classes of service to different customers. The business reason for data integrity is to ensure that each party in a transaction can have confidence in the business

transaction .There is also a business and legal need to have an audit trail and some evidence of non-repudiation to address liability issues . And more and more businesses are becoming aware of the internal threats to their applications by employees or others inside the firewall . Some business transactions require that confidentiality be provided on a service invocation or its data (such as credit card numbers) . Also, businesses on the Internet need to protect themselves from denial – of – service attacks . This is the environment in which we need to provide a security service model.

Moving forward from prototypes of web services will require an open security service model , based on standards , that can serve a heterogeneous " trust domain. " This security model should support interfaces for security services that :

1. Use XML data formats as the common representation of various security assertions .
2. Accept policy information expressed in XML to configure services (extending WSDL) .
3. Use XML messaging as the secure mechanism for exchanging the XML security assertions and also for servicing web service requests.

Security Technologies: To evolve from existing applications will initially involve wrapping legacy code with Web services . Some of the security constraints and trust models of existing applications will need to be carried forward and expressed within the early versions of web services. To accomplish this we will need to incorporate the work begun, in the various web services toolkits, on security technology from basic authentication XML digital signature support .

As these security technologies themselves become services , and workflow becomes the primary application paradigm for dynamic application integration, security services will evolve into core elements of a secure application workflow . A variety of security technologies are being adopted as standards.Following is a brief overview of these standards and how we can be used :

XML Digital Signatures :(6) is a standard for securely verifying the origins of messages . The XML signature specification allows XML documents to be signed in a standard way , with a variety of different digital signature algorithms. Digital signatures can be used for validation of messages and for non repudiation .

Security Assertion Markup Language(SAML):(7) is the first industry standard for secure e – commerce transactions using XML . SAML is being developed to provide a common language for sharing security services between companies engaged in business – to – business and business–to– consumer transactions. SAML allows companies to securely exchange authentication, authorization, and profile information between their customers, partners, or suppliers regardless of their security systems or e – commerce platforms. As a result SAML promotes the interoper – ability between disparate security systems , providing the framework for secure e – business transaction across company boundaries .

XML Encryption :(8) will allow encryption of digital content , such as Graphical Interchange Format (GIF) images , calable Vector Graphics (SVG) images , or XML fragments . XML Encryption allows the parts of an XML document to be encrypted while leaving other parts open , encryption of the XML itself , or the superencryption of data (i.e. , encrypting an XML document when some elements have already been encrypted) .

Application Patterns

Patterns are available for evolving to web services form existing web server applications . One is the browser – to server – pattern. This pattern wraps an existing application as a service , using a SOAP message as the service invocation . The web server provides a run – time execution container that defines its own security model , with policy information derived from a deployment descriptor configured by the deployer of the web server application . This pattern typically includes a mechanism for associating the identity of the invoking entity (the browser client) with the executing application instance,and allows the application to continue to function as it did before .

An advantage of this model is that the run – time code maintains the identity mapping , and name assertion life – time constraints and mechanisms for maintaining a valid token can be provided by the middleware. One of the disadvantages is that the model for "delegating" an identity requires that the delegated application – level identity is the same as the invocation identity of the intermediary (and hence the security context). This creates a coupling between middle ware and the application – level delegation logic and the requirement for the security context to support cascaded delegation , auditing , and nonrepudiation.

Another pattern involves rewriting the application with a modular design to create smaller tasks that can be combined in different ways to perform more complicated transactions . Each component is able to externalize its output into a message that the next component can use as input . This pattern uses SOAP messages to trigger each event. The messaging agents and message queues can be built into the run – time server below the application level. Sometimes the messaging agents become the " security – aware " part of the run – time code and control the flow of information along its path through components based on security attributes in the header of the message . Sometimes the security attributes get added into the message structure itself , as is the case with digital signatures . The trust model for this type of messaging relies on the specification of an explicit trust model along the lines of SAML. In the SAML – type name assertion, the trust will be explicit in the sense that the client and the server rely on coupling the identity information explicitly with the message , rather than on the underlying security context . Of course , this requires the service handler to be able to establish the identity of the caller based on the SAML – type name assertions and on trusting the entity that created these assertion tokens . Thus , in a SAML trust model an authentication / authorization authority has to be known and to digitally sign the assertions at the time of the authentication /authorization event.

A certificate associated with the signature can be used to identify the trusted authority and validate the signature , and a time stamp is included to indicate the assertion validity period . An *advantage* of this type of trust model is that the message can pass through multiple intermediaries . Authorization and delegation

decisions can be made in a standard way by the intermediaries without modifying the name assertion of the originator of the message request . If implemented in an " envelope ", it is also possible to build audit trails capable of asserting evidence of nonrepudiation , since each intermediary could wrap a message with its own name assertion . A *disadvantage* of this model is that the last component has to do some additional processing to make sure that the originator name assertion is valid from bot a trust and a time standpoint .

Both patterns implement security in the run – time code and both rely on a mapping of an external form of an identity into a run – time interpretation of that identity and into a set of rules about the identity and its capabilities . The difference has to do with where the mapping is done and whether the information is in an externalized form that can be middleware – independent and persistent .

Web Services Providers

Several companies that are specialize in component development are converting these into web services . They are responsible for the contents of the web services they provide . web services can be implemented in any language that can process XML, and may include a variety of functions . They may be quite complex and hide Trojan Horses or be infected with viruses or worms .Certifying that a program doesn't contain malicious software is what computer scientists call an undecidable problem , there is no method to guarantee that a given program is free of malicious code.

web Services will be trusted based on their origin and general fame, but there is no guarantee for the consumer . Certified software only proves the origin of the software and can guarantee a given functionality , there is no guarantee of the security of its contents .Naturally , vendors who develop their services carefully will be more trusted .

Conclusions

A web Services Security model should address security issues involved in request from an end client to a target service ,

including the intermediary services that route the service requests

This paper proposes a mechanism for the client to provide authentication data based on the service definition and , at the same time, for the service provider to retrieve those data . Because of the necessity for and complexity in establishing trust in the Web Services model, this paper also proposes how XML Digital Signatures and encryption can be exploited to achieve a level of trust. We have also shown that as part of its evolution , the web services requirements for application development can be seen as an opportunity to introduce a method of coupling security technologies (authentication , authorization , digital signatures , etc.) with business trust issues (public key infrastructure policy , role based access control , fire walls , etc.) and workflow into the creation of core web security services configured through policies expressed in XML .

We have looked at some of the aspects that have an effect on the security of Web Services . web services are indeed a very promising technology , and their use will continue to increase . We need , however , to be aware of the potential security problems that may occur . We have history as reference but undoubtedly there will be new problems . Currently , the Internet is not a safe place . On the positive side , we already have some promising security products and the basic frameworks for web services have sound security architecture . Further, cryptographic measures have solved some of the important security problems , such as authentication , message confide- ntiality , signatures , and non-repudiation , and all their power can be applied to web services as well .

References

1. W3C Recommendation , Extensible Markup Language (XML) 1.0 (Second Edition) ; see http: //www.w3.org/TR/ 2000/REC – xml – 20001006.html (2001) .
2. Box , D.; Ehnebuske , D. ; Kakivaya , G. ; Layman , A.; Mendelsohn ,N. ; Nielson , H. F.; Thatte , S. and Winter , D. Simple Object Access Protocol (SOAP) 1.1 , W3C Note (May 8, 2000); available at http:// www.w3.org/TR/SOAP/ .

3.Christensen ,E. ; Curbera , F. ; Merideth ,G. and Weerawarana , S. Web Services Description Language (WSDL) 1.1, W3CNote (March 15, 2001) ; see http://www.w3.org/TR/wsdl.html .
4.See http://www.uddi.org/ and http://www.uddi.org/faqs.html #who (2001).
5.Cover, R. (2000) Web Services Flow Language (WSFL) ; see http://xml. Coverpages.org/wsfl.html.
6.W3C XML Digital Signatures, see http://www.w3.org/ Signature and http://www.w3.org/TR/2000/CR-xmldsig-core-20001031/ .
7. Security Assertion Markup Language ; see http://www.oasisopen.org /committees/security/docs/draft-sstc-use-strawman-03.html (2001) .
8. W3C XML Encryption Syntax and Processing, see http://www.w3c. org/Encryption /2001/03/12-proposal.html .

مجلة ابن الهيثم للعلوم الصرفة والتطبيقية المجلد 19 (4) 2006

طريقة مقترحة لضمان امن وسرية الخدمات عبر شبكة Web

هيفاء جاسم محسن
قسم علوم الحاسبات ، كلية التربية – ابن الهيثم، جامعة بغداد

الخلاصة

لكي يتم اثبات امن وسرية الخدمات المقدمة عبر شبكة الانترنيت يجب أن نحدد ونفهم المخاطر التي تواجه الخدمات على هذه الشبكة اليوم بالاعتماد على تقنيات الامن والسرية الموجودة ، وفي نفس الوقت ابراز المسارات القياسية و تحديد كيفية الموازنة بينها و بين المخاطر في الخدمات الجديدة على شبكة الانترنيت . ان أي نموذج سرية جديد يجب ان يحدد كيفية سير البيانات خلال الشبكات لتنفيذ متطلبات الخدمات بدون التعرض للمخاطر.

في هذا البحث سوف نقدم طريقة مقترحة للمستفيد في الشبكة توفر الموثوقية للبيانات بالاعتماد على تعريف الخدمة المطلوبة ، ولمزود الخدمة في الشبكة لاسترجاع هذه البيانات . وسوف نوضح كيفية استخدام XML Digital Signature و Encryption لانجاز مستوى جيد من الضمان والموثوقية للبيانات.