

WEB SERVICES SECURITY

February 2008

© The Government of the Hong Kong Special Administrative Region

The contents of this document remain the property of, and may not be reproduced in whole or in part without the express permission of the Government of the HKSAR.

Disclaimer: Whilst the Government endeavours to ensure the accuracy of the information in this paper, no express or implied warranty is given by the Government as to the accuracy of the information. The Government of HKSAR accepts no liability for any error or omission arising from or related to the use of the information.

TABLE OF CONTENTS

Summary	2
I. Web Services	3
What are Web Services?	3
Finding and Publishing a Web Service	3
A Description of Web Services.....	4
Invokeing a Web Service	5
Web Services in the Hong Kong Government.....	5
II. Security Considerations	7
Security Threats	7
Defence and Protection	8
Deployment Considerations	12
III. Conclusion.....	14

SUMMARY

Web services are software systems designed to support interoperable machine-to-machine interaction over a network. Recently, a number of new standards and protocols have been introduced, and Web services are finding a new role to play in a range of business applications. When deploying a web services project, security is one of the most important issues that need to be addressed. In this paper, we discuss the possible threats to Web services and suggest preventive measures.

I. WEB SERVICES

WHAT ARE WEB SERVICES?

According to the World Wide Web Consortium (W3C)¹, a Web service is defined as “*a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL[Web Services Definition Language]). Other systems interact with the Web service in a manner prescribed by its description using SOAP[Simple Object Access Protocol] messages, typically conveyed using HTTP with an XML serialisation in conjunction with other Web-related standards*”². In other words, a Web service provides a mechanism for disparate web applications to communicate with each other via standards such as WSDL³, SOAP⁴, XML and so on. APIs (Application Programming Interfaces) are usually available for application developers and programmers to access Web services hosted on a remote site over the network, irrespective of the platform or underlying technology.

FINDING AND PUBLISHING A WEB SERVICE

¹ <http://www.w3.org>

² <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>

³ <http://www.w3.org/TR/wsdl>

⁴ <http://www.w3.org/TR/soap/>

The Universal Description, Discovery and Integration (UDDI)⁵ specification provides a set of services that assist in discovering or inquiring about the availability of Web services. A UDDI registry is a directory of business and service information, of which there are two types: public and private. Before a Web service can be discovered, it must first be registered to a UDDI registry. The UDDI publishing APIs⁶ are designed to create and update Web services entries to the UDDI registry.

For .NET implementations, Microsoft provides an alternative technology called DISCO⁷, which is also designed to create and discover .NET deployed web services.

A DESCRIPTION OF WEB SERVICES

Each Web service has a machine processable description written in Web Services Description Language (WSDL), which is “*an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information*”⁸. This WSDL file can be sent directly to perspective users, or published in the UDDI registries. Upon a successful inquiry to a UDDI registry, the WSDL link about the target Web service will be returned to the requester, describing core information about the contents and providing information on how to communicate (or bind) with the target Web service.

⁵ <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm>

⁶ http://uddi.org/pubs/ProgrammersAPI_v2.htm

⁷ <http://msdn.microsoft.com/library/default.asp?url=/msdnmag/issues/02/02/xml/TOC.asp>

⁸ <http://www.w3.org/TR/wsdl>

INVOKING A WEB SERVICE

After obtaining WSDL descriptions of the Web service or services required, the requester can invoke those Web services by initiating a SOAP (Simple Object Access Protocol)⁹ call to the service provider. The SOAP specification provides information that can be used for exchanging structured and typed information between peers using XML in a decentralised, distributed environment. Web services are delivered by exchanging SOAP messages between the Web service requester and the service provider, typically using HTTP or SMTP protocols to transport messages.

WEB SERVICES IN THE HONG KONG GOVERNMENT

In Hong Kong, one of the critical issues facing many e-Government projects is the problem of interoperability arising from diverse and disparate legacy government IT systems. To solve this problem, effective tools and methodologies are required to provide easy and seamless connections between systems that were developed by different teams at different times, running in different environments and employing different software/hardware platforms. In this respect, Web services technologies can be beneficial in providing e-Government facilities to business and the public.

The Hong Kong Government plays at least two roles in the wider economy when it comes to the application of Web services technologies. First, because the Government provides a large number of business services, efficient delivery of e-Government functions in the form of Web services can greatly enhance the local e-commerce infrastructure. Second, as a major player in the economy, the Government can act as a leader in the adoption of new

⁹ <http://www.w3.org/TR/soap/>

technologies, and set a good example in the application of Web services technologies in solving interoperability issues with legacy IT systems¹⁰.

¹⁰ http://www.info.gov.hk/digital21/eng/knowledge/webservice_egov.html

II. SECURITY CONSIDERATIONS

When deploying Web services in business, security is one of the important issues that need to be addressed. In this section, we describe the common threats that may affect Web services. Deployment considerations are also discussed briefly.

SECURITY THREATS

In 2005, the Web Services-Interoperability Organisation (WS-I) published a paper entitled “*Security Challenges, Threats and Countermeasures*”¹¹ which identified a number of key threats facing Web services:

1. **Message alteration:** an attacker alters an original message by inserting, removing or modifying content created by the originator, and the faked message is then mistaken by the receiver as being the originator’s real intention. In addition, an attacker may also construct a new fake message to fool the receiver into believing it to have come from a valid sender.
2. **Loss of confidentiality:** an unauthorised person intercepts and reads a transmitted message.
3. **Man in the middle attack:** an attacker sits between the real sender and the real receiver and fools both participants by, for example, capturing and reading all communications from both the sender and receiver and then forwarding modified messages to each of the two parties.

¹¹ <http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0.pdf>

4. Replay of message parts: an attacker replays parts of the captured message to the receiver with the aim of gaining access to an unauthorised system, or causing the receiver to take unnecessary action. This is a variation of threat number 1 above.
5. Replay: an attacker resends a complete message that has been previously sent by some other source, including the attacker.
6. Denial of service: an attacker does a small amount of work on a message that causes the target system to devote all its resources to a specific task so that it cannot provide any services to valid requests.

These threats basically exploit weaknesses in confidentiality, integrity, authentication and availability protection within existing infrastructure.

DEFENCE AND PROTECTION

To prevent against the threats identified above, a number of Web services and HTTP standards have been drawn up. According to the guideline paper “*Guide to Secure Web Services*”¹² published by NIST, the standards that can help protect identified threats are:

1. W3C XML Encryption¹³: used to encrypt and decrypt digital content.

W3C’s XML Encryption Working group is developing a standard for encrypting or decrypting the content of XML documents. The working group is also creating XML syntax to represent encrypted content and the information for decryption. With this standard, an XML document would be

¹² <http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf>

¹³ <http://www.w3.org/Encryption/2001/>

partially encrypted which effectively means only the sensitive portions of the XML document are encrypted. Different portions can be encrypted with different keys so that the same XML documents can be distributed to various recipients. Once the XML document is encrypted this way, tags indicating the beginning and end of the encrypted information will appear within the document. As only the XML data is encrypted but not the whole XML file, the document is still recognised by XML parsers and handled accordingly. Once this standard is adopted, confidentiality can be assured. An example of an XML encryption can be found at <http://www.ibm.com/developerworks/xml/library/x-encrypt/listing2.html>.

2. W3C XML Signature¹⁴: used to provide integrity, signature assurance and non-repudiation.

The W3C's XML Signature Working group has also proposed the XML Signature standard, which specifies the syntax and processing rules for applying digital signatures to any XML data. According to the W3C, "*XML Signatures provide integrity, message authentication, and/or signer authentication services for data of any type, whether located within the XML that includes the signature or elsewhere.*" In other words, XML Signatures can be used to ensure that the content within an XML document has not been changed or altered in any way during the transaction process. An example of an XML signature can be found at <http://www.xml.com/pub/a/2001/08/08/xmlsig.html>.

XML Signatures rely heavily on a concept called canonicalisation, which has been developed by the W3C mainly to standardise data formats, and compensate for typographical variations in the same piece of data scanned by different file systems and parsers. When a signature is applied to XML content, canonicalisation creates a unique signature using the data and tags in

¹⁴ <http://www.w3.org/Signature/>

the XML file. This ensures that by applying the same canonicalisation method to the received message content, data integrity can be verified.

3. WS-Security¹⁵ Tokens: used to help the receiver of the message identity and verify the sender.

Security tokens provide a mechanism for conveying security information with a SOAP message, and the token itself is described in XML. The following security tokens are supported:

- a) Username Tokens: used as a means to identify the requestor by “username”, and an optional password;
- b) X.509 Tokens: uses an X.509 digital certificate to help authenticate a SOAP message or to identify a public key with a SOAP message that has been encrypted;
- c) SAML (Security Assertion Markup Language) Tokens: used to secure SOAP messages and SOAP message exchanges with the help of SAML assertions that bind the subjects (e.g. the sender) and statements of the assertions to a SOAP message with an XML signature. Three general kinds of assertion statements can be used: authentication, authorisation and attribute. These three statements are used at various times in an application to determine who the requester is, what they are requesting, and whether or not their request has been granted. In addition, SAML assertions enable the preservation of security restrictions across different security domains;
- d) Kerberos Tokens: used to allow a service to authenticate the Kerberos ticket and interoperate within existing Kerberos domains;

¹⁵ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

- e) Rights Expression Language (REL) Tokens: used to implement message level integrity and confidentiality using Rights Expressions as defined in ISO/IEC 21000-5.
4. W3C WS-Addressing¹⁶ : used to help protect against a message replay attack.

A uniquely identifiable message ID can be used to detect a message replay, but in order to detect a message replay, the message ID should contain data such as a timestamp so that any legitimate retransmission of the message would not be confused with a replay attack. In addition, the message ID should not be predictable.

5. Other standards used in more traditional Web technologies, including IETF SSL/TLS, SSL/TLS with client authentication, and IETF HTTP authentication methods can also help protect against weaknesses in confidentiality and authentication.

In addition, there are other specifications and standards that are designed to help support the protective measures outlined above. For example, the W3C XML Key Management Specification (XKMS)¹⁷ defines protocols for public key management. It defines a way to distribute and register public keys used by the XML Signature and XML Encryption specifications. XKMS comprises two sub-protocols: XML Key Registration Service Specification (X-KRSS) and XML Key Information Service Specification (X-KISS). X-KRSS is used for public key registration and X-KISS is used to resolve the keys provided in an XML Signature.

¹⁶ <http://www.w3.org/2002/ws/addr/>

¹⁷ <http://www.w3.org/TR/xkms/>

Extensible Access Control Markup Language (XACML) is another specification aimed at enhancing the access control capability of Web Services. XACML is based on the access control matrix model and allows for defining authorisation rules for each element of an XML document, or the document as a whole.

DEPLOYMENT CONSIDERATIONS

The standards described above form the groundwork for SOAP messaging security. All parties involved in message exchange can make use of these XML technologies in the following manner:

1. The message sender specifies the processing intermediaries in the SOAP message header.
2. The message sender can encrypt message headers and sign them using the XML Signature standard.
3. Each part of the SOAP message can be given a different signature that corresponds to the intended processing intermediary.
4. The message sender can utilise XKMS to distribute and register public keys for each processing intermediary
5. Upon receipt of the message, each processing intermediary inspects the signed SOAP headers using an XKMS public key and validates the signature.
6. After validation, each processing intermediary may then utilise XML encryption to decrypt the SOAP headers and the corresponding message component.

This process may be repeated right through the processing chain. Furthermore, each processing intermediary may encrypt and sign additional SOAP headers and message components that are intended for downstream processing.

As SOAP messages are transported using the HTTP protocol, traditional firewalls are not XML-aware and will usually permit all XML traffic to go through without further checking. That is, a network firewall cannot add extra protection to SOAP messages. It is advisable therefore, when implementing Web services without the support for WS-Security, to deploy XML gateways or firewalls that can perform XML checking.

In addition, as with normal applications, a secure audit log of all messages going in and out should be kept. The log should provide sufficient information to support comprehensive audits of the effectiveness of, and compliance with, the intended security measures.

III. CONCLUSION

As Web services are still relatively new in terms of their practical implementation, web architects and developers need to be careful in how they deploy Web services. In addition to the protective measures discussed in this document, standard recommendations for the security of web applications should also be followed. Some best practices are:

1. Harden underlying servers according to security guidelines;
2. Apply the latest security patches to all system components;
3. Ensure that strict validation is applied to all input;
4. Ensure proper authentication and authorisation is enforced to restrict privileges and access rights to only valid personnel.

In addition, when firewalls do not provide adequate security when it comes to the deployment of Web services, a WS-Security or XML-aware gateway should be considered.