CAROLINA TECHNOLOGY CONSULTANTS

Tech Brief

October, 2009 *Architectural Issues in IT and Data Communications*

Topic

Web Services Security v1.1

Prepared By

Tyler R. Johnson

Summary

Web Services Security (WS-Security) is a set of related specifications for handling message integrity and privacy of communications for web services applications. WS-Security is implemented as extensions to SOAP messaging.

Discussion

WS-Security version 1.1 is defined by the core specification in [1] which outlines the overall framework for authentication, message integrity and privacy. Supplementary profiles [2] – [7] address protocol-specific implementations of the framework such as Kerberos and SAML. WS-Security Version 1.1 is not backward compatible with version 1.0 of the specification.

The specification is intended to be a building block upon which higher level and more complete security frameworks can be implemented. It includes awareness of multiple domains and extensibility for a variety of security token types, including binary and XML as well as opaque keys.

WS-Security

Provides:
* Extensible security framework
* Support for multiple token/key types
* End to end message content security
* Message integrity and privacy

Does NOT provide:
* Transport Level Security
* Trust determination
* Non-repudiation

WS-Security provides a rich vocabulary for describing security assertions and binding them to identities. It is the responsibility of the receiver to validate those assertions and keys by ensuring that received values are not malformed and that digital signatures are valid with the appropriate certificate authority. Developing drafts within OASIS provide support for federated security models which aid this process, though these are not yet standardized.

Headers

Security information is included in header blocks marked with <wsse:Security> tags. Multiple tags may be included in a SOAP message. Because a SOAP message may traverse a path which includes a number of intermediaries, each header block

may address security concerns to specific intermediaries. Therefore, multiple security header blocks may be present in a single SOAP message. Because each header block will be signed by the intermediary processing it, a chain of activity is captured in the ultimate message that provides detailed information about its journey. Intermediaries need not utilize the same signature algorithms in a given message.

Security tokens are inserted directly into the header if those token are XML based. Alternately, the header may contain a token reference to a URI that contains the token. References may be direct, embedded or represented by key identifiers.

Encryption

WS-Security leverages the XML Encryption standard to allow body and header blocks and body elements to be encrypted with shared keys or embedded keys. SOAP includes a mustUnderstand tag in headers that is essential for processing. Because fully encrypted headers would prohibit this, an <wsse11:EncryptedHeader> element is introduced to address this problem.

Because header and body blocks can be encrypted differently, implementers should thoroughly review the XML Encryption specification and ensure that the actual payload data is handled in an application-appropriate manner. For example, while SSL transport between SOAP actors may be adequate, direct encryption of payloads may be advisable to prevent actors from having access to payload data.

Timestamps

WS-Security includes vocabulary to support the timestamping of messages. Because timestamp assertions are protected as a part of WS-Security's message integrity characteristic, receivers can detect forged timestamps.

It is the responsibility of the receiver to protect itself against replay attacks by establishing a window of acceptable timestamps. For example, in a real time application, timestamps of seconds may be considered valid and timestamps of hours considered stale. However, an invoice processing application may have a SOAP message that has been signed and queued for weeks or even months awaiting processing. Business logic for assessing the level of trust in the source timestamp and length of validity are left to the implementer.

Error Messages

WS-Security utilizes SOAP's Fault mechanism to provide standardized error codes. However, it is recognized that error messages detailing the specific ways in which security assertions fail can themselves be used to compromise security. Accordingly, implementers may wish to use only very general fault messages.

Strategy Considerations

WS-Security is a reasonable framework for securing web services applications. Its strength is in giving implementers the ability to secure messages as they traverse various intermediary processing agents.

WS-Security by itself does not represent a full security solution. Rather, it provides a set of building blocks that can be used with additional architectural discretion and specification to define a complete security solution for a given web services application.

For Further Information

1. OASIS Standard WS-Security: SOAP Message Security 1.1, A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker, February 2006.
2. OASIS Standard Web Services Security UsernameToken Profile 1.1, A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker, February 2006.
3. OASIS Standard Web Services Security SAML Token Profile 1.1, R. Monzillo , C. Kaler, A. Nadalin, P. Hallam-Baker, February 2006.
4. OASIS Standard Web Services Security X.509 Token Profile 1.1, A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker, February 2006.
5. OASIS Standard Web Services Security Kerberos Token Profile 1.1, A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker, February 2006.
6. OASIS Standard Web Services Security Rights Expression Language (REL) Token Profile 1.1, A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker, February 2006.
7. OASIS Standard Web Services SOA Messages with Attachments (SwA) Profile 1.1, F. Hirsch, February 2006.
8. Full Web Services Security v1.1 documents with errata and schema.
9. U.S. Department of Homeland Security with Carnegie Mellon University and Cigital, Inc. Security Concepts, Challenges, and Design Considerations for Web Services Integration, G. Peterson, H. Lipson, 2006.