WSIT Security Profiles

Sang Shin
Michèle Garoche
www.javapassion.com
“Learning is fun!”

Agenda

- NetBeans Security Profiles
- NetBeans WSIT Security Profiles
  - Username Authentication with Symmetric Keys
  - Mutual Certificates Security
  - Transport Security (SSL)
  - Message Authentication over SSL
  - SAML Sender Vouches with Certificates
  - SAML Holder of Key
  - STS Issued Token
  - STS Issued Token with Service Certificate
  - STS Issued Endorsing Token
- Resources

NetBeans Security Profiles

Why NetBeans Security Profiles

- WSIT makes securing web services easier by including a set of pre-configured security profiles that can be applied to a web service or a web service operation simply by selecting it from a list
  - You can use NetBeans' configuration wizard to customize the security mechanism to the needs of your application.
- Security feature can be set on an endpoint or per-operation basis
- Standard headers in the SOAP message or XPath-identified elements may be signed and encrypted.

WSIT, WS-SecurityPolicy and WSDL

- After the security mechanism is chosen, appropriate WS-Security Policy assertions are generated in the WSIT configuration file.
- This configuration file is then used to generate the WSDL with correct policy assertions.

NetBeans Security Profiles

- NetBeans IDE provides a set of security profiles for configuring security on service endpoint and client
  - These profiles allow a developer to select the integrity and confidentiality of a message
  - A developer can also choose token format

NetBeans WSIT Security Profiles

NetBeans WSIT Security Profiles

- Username Authentication with Symmetric Keys
- Mutual Certificates Security
- Transport Security (SSL)
- Message Authentication over SSL
- SAML Authorization over SSL
- Endorsing Certificate
- SAML Sender Vouches with Certificates
- SAML Holder of Key
- STS Issued Token
- STS Issued Token with Service Certificate
- STS Issued Endorsing Token

Criteria to Use for Selecting a Security Profile

- Security layer
  - Transport or message layer
- Type of client credentials
  - Username/password, x509 certificate, SAML assertion or issued token from a third party trust authority (STS)
- The role of the client credential played in securing the messages
  - As a supporting token or as a primary securing token.
  - In the case of supporting token, the messages are usually secured with server's X509 certificate.

Username Authentication using Symmetric Keys

NetBeans WSIT Security Profiles

Username Authentication with Symmetric Keys

- Supports integrity and confidentiality using symmetric key
  - Symmetric key cryptography relies on a single, shared secret key that is used to encrypt a message
  - The shared, symmetric key is generated at runtime and encrypted using the service's certificate.
  - The client must specify the alias in the truststore by identifying the server's certificate alias.
- Supports authentication using username/password
  - The client does not possess any certificate/key of his own

Mutual Certificates Security

NetBeans WSIT Security Profiles

Mutual Certificates Security

- Supports integrity and confidentiality messages, supports authentication
- When using mutual certificates, a keystore and truststore file must be configured for both the client and server sides of the application
  - A keystore is a database of private keys and their associated X.509 certificate chains authenticating the corresponding public keys.
  - A truststore is a database of trusted entities and their associated X.509 certificate chains authenticating the corresponding public keys.

Transport Security (SSL)

NetBeans WSIT Security Profiles

Transport Security (SSL)

- Security is provided by the transport mechanisms used to transmit data over the wire between clients and providers
  - Relies on secure HTTP transport (HTTPS) using Secure Sockets Layer (SSL)

Transport Security (SSL)

- Transport security is a point-to-point security mechanism that can be used for authentication, integrity, and confidentiality.
- When running over an SSL-protected session, the server and client negotiate an encryption algorithm and cryptographic keys
- Security is "live" from the time the data leaves the consumer until it arrives at the provider, or vice versa.
  - The problem is that it is not protected once it gets to its destination.
  - For protection of data after it reaches its destination, use one of the security mechanisms that uses SSL and also secures data at the message level.

Transport Security (SSL)

- Digital certificates are necessary when running secure HTTP transport (HTTPS) using Secure Sockets Layer (SSL).
  - The HTTPS service of most web servers will not run unless a digital certificate has been installed.
  - Digital certificates have already been created for GlassFish, and the default certificates are sufficient for running this mechanism, and are required when using Atomic Transactions.

Message Authentication over SSL

NetBeans WSIT Security Profiles

Message Authentication over SSL

- Attaches a cryptographically secured identity or authentication token with the message and use SSL for confidentiality
- By default, a Username Supporting Token will be used for message authentication.
  - To use an X.509 Supporting Token instead, click the Configure button and select X509.

SAML Sender Vouches with Certificates

NetBeans WSIT Security Profiles

SAML Sender Vouches with Certificates

- Protects messages with mutual certificates for integrity and confidentiality and with a Sender Vouches SAML token for authorization.
  - The attesting entity provides the confirmation evidence that will be used to establish the correspondence between the subject of the SAML subject statements (in SAML assertions) and SOAP message content.
  - The attesting entity, presumed to be different from the subject, vouches for the verification of the subject.

SAML Sender Vouches with Certificates

- The receiver has an existing trust relationship with the attesting entity.
- The attesting entity protects the assertions (containing the subject statements) in combination with the message content against modification by another party.
- For this mechanism, the SAML token is included as part of the message signature as an authorization token and is sent only to the recipient.

SAML Sender Vouches with Certificates

- The message payload needs to be signed and encrypted.
- The requestor is vouching for the credentials (present in the SAML assertion) of the entity on behalf of which the requestor is acting.
- The initiator token, which is an X.509 token, is used for signature.
- The recipient token, which is also an X.509 token, is used for encryption.
- For the server, this is reversed, the recipient token is the signature token and the initiator token is the encryption token. A SAML token is used for authorization.

SAML Holder of Key

NetBeans WSIT Security Profiles

SAML Holder of Key

- Protects messages with a signed SAML assertion (issued by a trusted authority) carrying client public key and authorization information with integrity and confidentiality protection using mutual certificates.
- The Holder-of-Key (HOK) method establishes the correspondence between a SOAP message and the SAML assertions added to the SOAP message.
- The attesting entity includes a signature that can be verified with the key information in the confirmation method of the subject statements of the SAML assertion referenced for key info for the signature.

SAML Holder of Key

- Under this scenario, the service does not trust the client directly, but requires the client to send a SAML assertion issued by a particular SAML authority.
- The client knows the recipient's public key, but does not share a direct trust relationship with the recipient.
- The recipient has a trust relationship with the authority that issues the SAML token.
- The request is signed with the client's private key and encrypted with the server certificate. The response is signed using the server's private key and encrypted using the key provided within the HOK SAML assertion.

STS Issued Token

NetBeans WSIT Security Profiles

STS Issued Token

- This security mechanism protects messages using a token issued by a trusted Secure Token Service (STS) for message integrity and confidentiality protection.
- An STS is a service that implements the protocol defined in the WS-Trust specification
  - This protocol defines message formats and message exchange patterns for issuing, renewing, canceling, and validating security tokens.

STS Issued Token

- Service providers and consumers are in potentially different managed environments but use a single STS to establish a chain of trust.
- The service does not trust the client directly, but instead trusts tokens issued by a designated STS.
  - In other words, the STS is taking on the role of a second service with which the client has to securely authenticate.
  - The issued tokens contain a key, which is encrypted for the server and which is used for deriving new keys for signing and encrypting.

STS Issued Token with Service Certificate

NetBeans WSIT Security Profiles

STS Issued Token with Service Certificate

- This security mechanism is similar to the one discussed in STS Issued Token, with the difference being that in addition to the service requiring the client to authenticate using a SAML token issued by a designated STS, confidentiality protection is achieved using a service certificate.
- A service certificate is used by a client to authenticate the service and provide message protection.
  - For GlassFish, a default certificate of s1as is installed.

STS Issued Endorsing Token

NetBeans WSIT Security Profiles

STS Issued Endorsing Token

- This security mechanism is similar to the one discussed in STS Issued Token, with the difference being that the client authenticates using a SAML token that is issued by a designated STS.
  - An endorsing token is used to sign the message signature.
- In this mechanism, message integrity and confidentiality are protected using ephemeral keys encrypted for the service. Ephemeral keys use an algorithm where the exchange key value is purged from the cryptographic service provider (CSP) when the key handle is destroyed. The service requires messages to be endorsed by a SAML token issued by a designated STS.

Resources

WSIT Security

- WSIT Security Mechanisms document
  - https://wsit-docs.dev.java.net/releases/m6/WSIT_Security4.html
- Project Tango overview by Arun Gupta
  - https://wsit.dev.java.net/docs/tango-overview.pdf
- WSIT Security Configuration Demystified blog
  - https://xwss.dev.java.net/articles/security_config.html

Thank you!

Sang Shin
Michèle Garoche
www.javapassion.com
"Learning is fun!"