Identity, Security and XML Web Services

Jorgen Thelin

Chief Scientist
Cape Clear Software Inc.

E-mail: Jorgen.Thelin@capeclear.com

Abstract

- The use of security credentials and concepts of single-sign-on and "identity" play a big part in Web Services as developers start writing enterprise-grade line-of-business applications. An overview is provided of the emerging XML security credential standards such as SAML, along with various "identity" standards such as Passport and Liberty. We examine how "identity aware" Web Service implementations need to be, and the value a Web Services platform can add in reducing complexity in this area, with lessons drawn from experiences using J2EE technology for real-world security scenarios.

Agenda

- The Concept of Identity
- Web Services and Identity
- Interoperable XML Security and Identity
- Examples of Security Credentials in SOAP
- Single-sign-on
- Identity Awareness in Web Services

A Definition of Identity

- Definition from Cambridge Dictionaries Online:
  - **Identity**
    - [noun]
    - Who a person is, or the qualities of a person or group which make them different from others
    - http://dictionary.cambridge.org/define.asp?key=identity*1+0

What is Identity?

- At its most basic, the concept of Identity is about:
  - Who you are
  - How you prove who you are
  - What that allows you to do

Identity – Who are you?

- An identity equates to a particular subject or principal
  - For example: Joe Bloggs …
  - … Who lives at 123 My Street, Your Town
- Usually equates to a person, but could also be a group, corporation, or even something like an automated software agent component
- Subjects must be distinguishable

Identity – Proof of identity

- How do you prove who you are?
- In real life, this is usually thru some official documents such as:
  - Driving License
  - Passport
- In computing terms, a user has a set of security credentials such as:
  - username + password
  - X509 certificates

Identity – Permissions

- What does this identity prove about us?
- What does this identity allow us to do?

- Some real life examples:
  - Holding a UK passport proves I am a UK Citizen
  - Losing my passport does not stop me being a UK Citizen; it just makes it harder to prove that I am.
  - A standard driving license shows I am allowed to drive a car
  - I am not allowed to drive a Heavy Goods Vehicle unless I hold a HGV Driving License

Identity – Permissions and Credentials

- The permissions and entitlements for an identity is ultimately determined by the set of credentials that were presented to assert that identity.

- Permissions and credentials are use to make policy enforcement decisions

  - Am I allowed to drive a Heavy Goods Vehicle?
  - Am I allowed to work in the UK?
  - Am I allowed to work in the US?

Web Services and Identity

- How does this affect Web Services?
- Security and Identity is a fundamental requirement of any real-world deployment of a Web Services application
- Ultimately all security policy decisions are based on the caller's identity
- The challenge is to how to represent and prove a caller's identity in an open and interoperable way.

Web Services and Identity 2

- Security and identity considerations for a Web Services application:
  - Authentication
    - Who is the caller?
    - How did they prove their identity?
    - Do we trust the source of these credentials?
  - Authorization
    - What is the caller allowed to do?
  - Attributes
    - What other facts do we know about the caller?
      - For example, e-mail address, department, employee number
    - How do we use this attribute information in the application?
      - For example, customizing the data returned based on display preferences

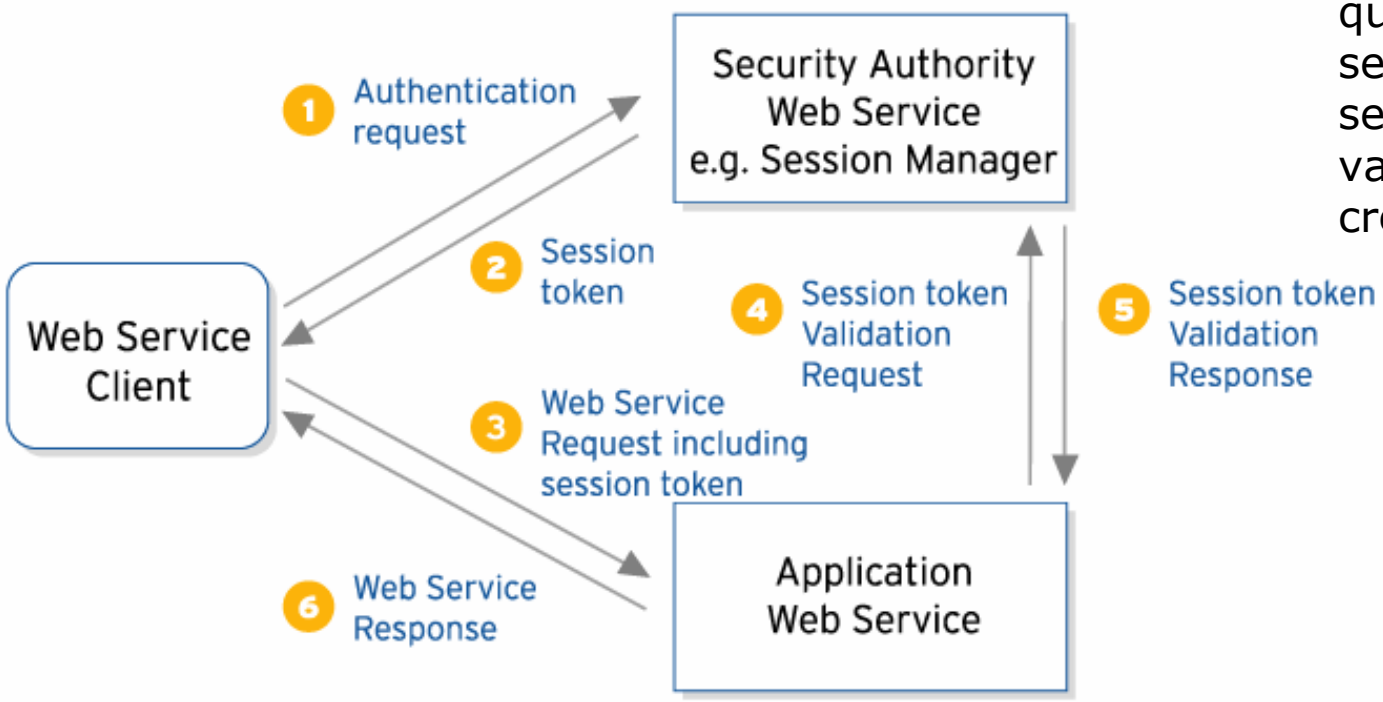
Web Services and Identity 3

- To achieve interoperable security and identity, web services require the following
- Standard ways to:
  - Representing security credential data in XML
    - Eg. SAML – Security Assertions Markup Language specification
  - Obtaining credential data
    - Eg. Single-sign-on services such as Microsoft Passport or Liberty Alliance specifications
  - Transport credential data in a SOAP message
    - Eg. SOAP header fields defined in the WS-Security specification

Types of Security Tokens

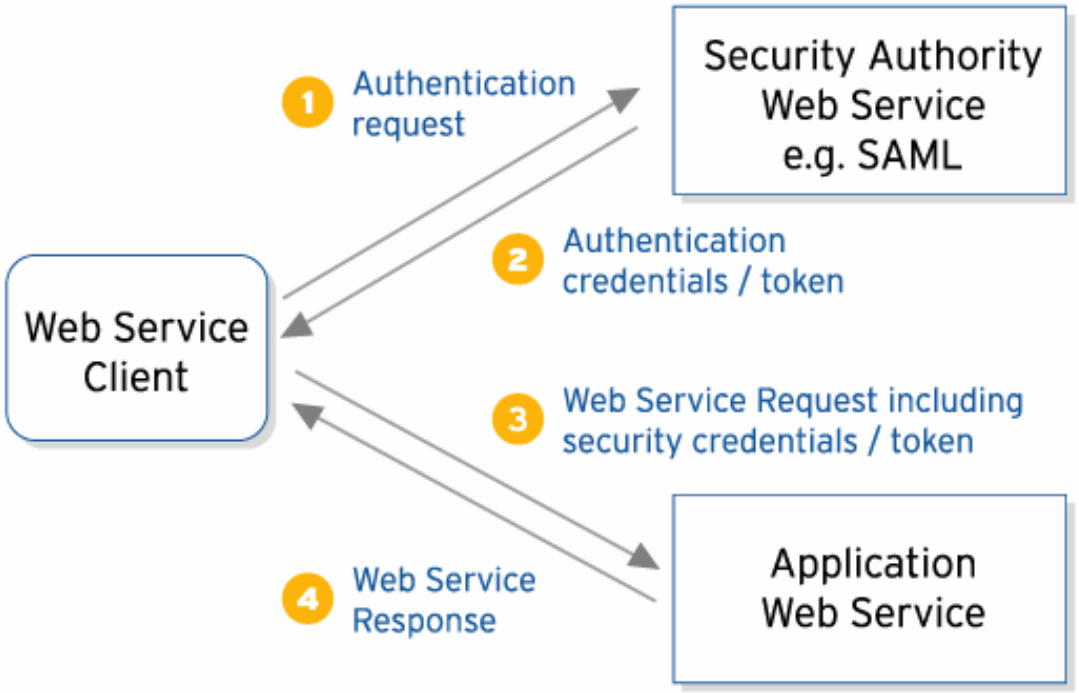
- The WS-Security specification set defines the following tokens:
  - Unsigned security tokens
    - Username
  - Signed security tokens
    - X.509 certificates (binary)
    - Kerberos tickets (binary)
  - XML security tokens
    - Any XML token, such as SAML
    - Usually self verifying / signed

Typical XML Security Dialogue – Non Self-Validating Credentials



Need to query the security service to validate the credentials

Typical XML Security Dialogue – Self Validating Credentials



No need to query the security service to validate the credentials.

Usually done by the security authority digitally signing the credentials.

SAML v1.0

- SAML – Security Assertions Markup Language
  - An XML-based framework for exchanging security information
  - A specification published by the OASIS organization
- The SAML specification defines:
  - How to represent security credentials (“Assertions” in SAML parlance) using XML
  - An XML message exchange protocol for querying a SAML Authority service
- SAML does not define:
  - How to obtain security credentials (“Assertions”) in the first place

SAML Assertion Types

- SAML Authentication Assertions
  - The results of an authentication action performed on a *subject* by a *SAML authority*
- SAML Attribute Assertions
  - Attribute information about a *subject*
- SAML Authorization Assertions
  - Authorization permissions that apply to a *subject* with respect to a specified *resource*

A Username Token in WS-Security SOAP Header

```
<SOAP:Envelope xmlns:SOAP="...">
  <SOAP:Header>

    <wsse:Security
      xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">

      <wsse:UsernameToken>
    <wsse:Username>jthelin</wsse:Username>
    <wsse:Password Type="wsse: PasswordDigest" >XYZabc123</wsse:Password>
        <wsse:Nonce>
     h52sI9pKV0BVRPUolQC7Cg==
    </wsse:Nonce>
   </wsse:UsernameToken>

      ...
    </wsse:Security>

   </SOAP:Header>

   <SOAP:Body Id="MsgBody">
     <!-- SOAP Body data -->
   </SOAP:Body>
  </SOAP:Envelope>
```

A Binary X509 Certificate in WS-Security SOAP Header

```
<wsse:Security
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">

  <wsse:BinarySecurityToken  Id="X509Token"
     xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext"
     ValueType="wsse:X509v3"
     EncodingType="wsse:Base64Binary"
  >
     MIIEZzCCA9CgAwIBAgIQEmtJZc0...
  </wsse:BinarySecurityToken>

  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

    <ds:SignedInfo> ... </ds:SignedInfo>
    <ds:SignatureValue> ... </ds:SignatureValue>

    <ds:KeyInfo>
      <wsse:SecurityTokenReference>
        <wsse: Reference URI="#X509Token" />
      </wsse:SecurityTokenReference>
    </ds:KeyInfo>
  </ds:Signature>

  ...
</wsse:Security>
```

A SAML Assertion in WS-Security SOAP Header

```
<wsse:Security
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/12/secext">

    <saml:Assertion
      xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
      MajorVersion="1"
      MinorVersion="0"
      AssertionID="SecurityToken-mc375268"
      Issuer="mycompany"
      IssueInstant="2002-07-23T11:32:05.6228146-07:00" >
      ...
    </saml:Assertion>

    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

      <ds:SignedInfo> ... </ds:SignedInfo>
      <ds:SignatureValue> ... </ds:SignatureValue>

      <ds:KeyInfo>
        <wsse:SecurityTokenReference>
          <saml:AssertionIDReference>
            SecurityToken-mc375268
          </saml:AssertionIDReference>
        </wsse:SecurityTokenReference>
      </ds:KeyInfo>
    </ds:Signature>

    ...
</wsse:Security>
```

Single-sign-on Services

- SSO Services provide:
  - a single point of logon and authentication
  - a standardized way to obtain suitable credentials to prove the authenticated identity
- The main contenders using XML are:
  - Liberty Alliance
  - Microsoft Passport
  - Proprietary security products such as Netegrity SiteMinder are adding direct SAML interfaces
  - WS-Trust – new spec for standardized XML interface
- Still remains an area needing standardization

Liberty Alliance

- The Liberty Alliance Project is a cross-industry group aiming to establish an open standard for federated network identity
- http://www.projectliberty.org/
- The Liberty specification v1.0 has two main facets:
  - Single sign-on
  - Identity federation

Microsoft .NET Passport

- Microsoft .NET Passport is a suite of Web-based services that makes using the Internet and purchasing online easier and faster for users.
- http://www.passport.com/
- .NET Passport provides users with
  - Single sign-in (SSI)
  - Fast purchasing capability at participating sites
- Microsoft is upgrading the current Passport facilities to
  - Provide an XML interface
  - Support federation
  - Use Kerberos v5 as the underlying mechanism for securely exchanging credentials

The Need for a Sign-on Standard – WS-Trust

- The need remains for a "sign-on standard" to avoid reliance on proprietary interfaces
- WS-Trust
  - A proposed specification in the WS-Security family
  - Provides a standardized interface for acquiring security tokens
  - Still very early in the standardization process, but the most likely candidate for a common interface
  - http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-trust.asp

Identity-awareness in Web Services

- Do web services themselves need to be identity-aware?
  - Not really, in most cases
  - A mature web services platform product such as Cape Clear Server can handle almost all the "boilerplate" work of authentication and enforcement of access control lists

Identity-awareness in Web Services - 2

- Most standard authentication and authorization functions are best done in a uniform manner by the platform, rather than being implemented on an application-by-application basis
  - Interceptor plugins allow this to be a deployment policy decision rather than an implementation decision
- Web Service application only needs to be Identity-aware if it needs to use attributes asserted for the caller
  - For example, reading the delivery address from the user's MS Passport record

Ultimate Web Services platform security

- Ultimate goal will be declarative security functions for web services just like EJB
  - So, having declarative statements of:
    - Permitted authentication realms / single-sign-on services
    - Required transport security attributes
      (for example, "Callers must use encrypted / SSL connections")
    - Required message security attributes
      (for example, "Messages must be digitally signed")
    - Role-based access control lists applied at the granularity of the operation / method call.
- This places control of security to application administrators rather than developers.

Summary

- "Identity" is one of the fundamental concepts in all Web Service security mechanisms
- Having a standard XML-based serialized form of credentials is vital for true end-to-end interoperability
- Standardization of specifications for credential exchange and single-sign-on using XML and SOAP are still incomplete, so true interoperability is not yet possible.
- Use a mature Web Services runtime platform such as Cape Clear Server to handle most "boilerplate" security tasks such as enforcing authentication and authorization requirements.