

Building Strong Web Service Authentication Using Security Assertion Markup Language (SAML)

Jahan Moreh
Chief Security Architect
Sigaba
jmoreh@sigaba.com
www.sigaba.com

Objectives

- Articulate the problem of integrating authentication into distributed applications
- List the individual specifications that make up SAML 2.0 standard
- Describe the structure of a SAML *Assertion*
- Describe SAML's *Authentication Statement* element
- Demonstrate an understanding of SAML Bindings
- Understand the use of Assertions, Statements and Bindings in the Browser Single Signon Profiles
- Demonstrate an understanding of Web Services Security SAML Token Profile

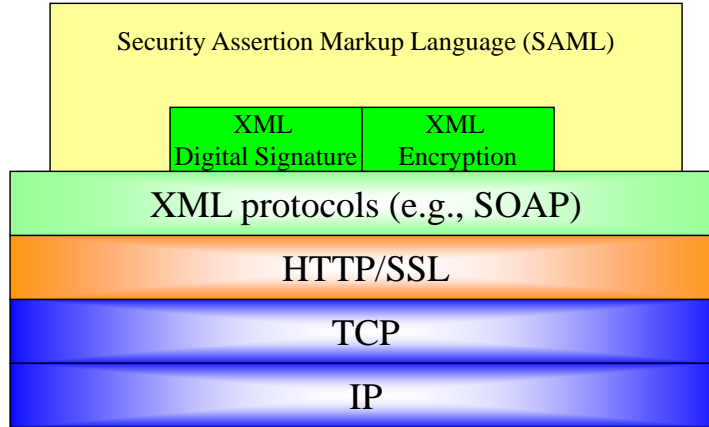
Integration of Security into Enterprise Applications

- Traditional approach
 - **Enable application using the underlying security infrastructure**
- Disadvantage
 - **Application is exposed to the security infrastructure**
 - **Integration is specific to security infrastructure**
 - **Little or no *re-use***
 - **Example:**
 - Kerberizing
 - PKI-enabling

New Approach: Abstract Integration

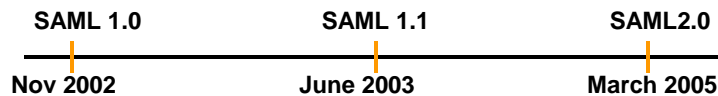
- Define a set of security services at an abstract level
 - **Authentication Assertion**
 - **Attribute Assertion**
 - **Authorization Decision**
- Advantages
 - **Independent of language, operating system, and communication protocol**
 - **Allows reuse of security functions integrated within an application**
 - **Allows migration of security infrastructure without affecting applications**
 - **Supports federated security model**

SAML: Total Picture



What is SAML

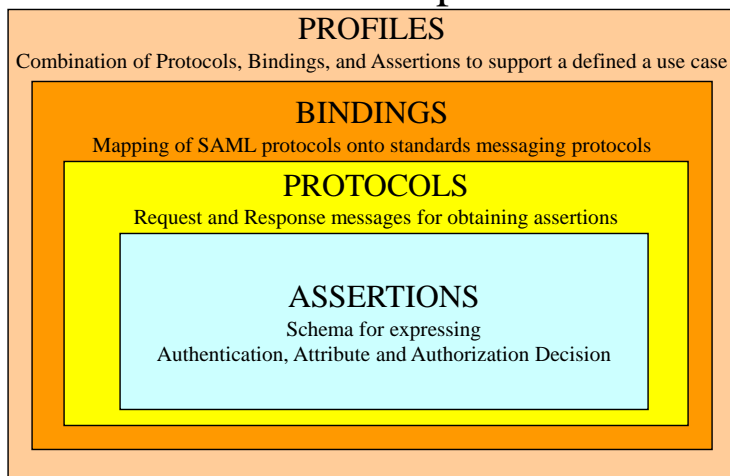
- OASIS Open Standard
- Security *assertions* expressed in XML
- Promotes interoperability
 - Applications and authentication systems
 - Applications and policy engines
 - Applications and attribute repositories
- SAML 2.0 and Liberty specification have converged



Components of SAML

- Assertions and Protocols
- Bindings
- Profiles
- Metadata
- Authentication Context
- Conformance
- Security Considerations
- Glossary

Relationship Between SAML Components



Structure of SAML Assertions

- One or more *Statements* regarding a *Subject*
- Three kinds of statements
 - **Authentication**
 - **Attribute**
 - **Authorization Decision**
- SAML allows other types of concrete statements
- Assertions may be digitally signed using XML DSIG

SAML Assertion Schema

- **Namespaces**
 - `urn:oasis:names:tc:SAML:2.0:assertion`
 - `urn:oasis:names:tc:SAML:2.0:protocol`
- **Abstract types**
 - **Capture common elements and attributes**
 - **Enable extensibility**
- **Concrete types**
 - **Instances of abstract types**
 - **Use XSD extension element**

Generalizing Assertions

- Condition
 - Additional information from a validation service
 - Valid only if the Service Provider is a member of specific audience
- Advice
 - Assertions that were used to make the policy decision

SAML Assertion

- Required attributes
 - **MajorVersion** and **MinorVersion**: **string**
 - **ID**: **ID**
 - **IssueInstant**: **dateTime**
- Required elements
 - **Issuer**: **saml:NameID** (extension of **string**)
- Zero or more *Statements*
- Zero or one *Subject*
- Optional other elements
 - **Condition**
 - **Advice**
 - **Signature**: XML digital signature using *enveloped* technique

SAML Subject Element

- Describes the principal that is the subject of an assertion
- A Subject has the following elements and attributes
 - An optional ID element
 - Zero or more *SubjectConfirmation* Elements

Example of a SAML Assertion Containing an Authentication Statement

```
<Assertion MajorVersion="2" MinorVersion="0"
  ID="urn:SigAuth:e587bfe6c7@SIGABA.COM:3d"
  IssueInstant="2004-06-19T11:19:00.000-08:00" >
  <Issuer> mechanism/SigAuth@SIGABA.COM </Issuer>
  <Subject>
    <NameID
      NameQualifier="sigaba.com"
      Format="urn:oasis:names:tc:SAML:1.1:nameidformat:emailAddress">
      jmoreh@sigaba.com
    </NameID>
  </Subject>
  <AuthnStatement
    AuthnInstant="2004-06-19T11:18:00.000-08:00" >
    <AuthnContext>
      <AuthnContextClassRef
        urn:oasis:names:tc:SAML:2.0:ac:classes:Password
      </AuthnContextClassRef>
    </AuthnContext>
  </AuthnStatement>
</Assertion>
```



Authentication

Statement with Conditions

```
<Assertion MajorVersion="2" MinorVersion="0"
  ID="urn:SigAuth:e587bfe6c7@SIGABA.COM:3d"
  IssueInstant="2004-06-19T11:19:00.000-08:00" >
  <Issuer> mechanism/SigAuth@SIGABA.COM </Issuer>

  <Conditions NotBefore="2004-6-19T11:19:00.000-08:00Z"
    NotOnOrAfter="2004-6-19T12:19:00.000-08:00Z" >
    <AudienceRestriction>
      <Audience>http://kserver1.sigaba.com</Audience>
      <Audience>http://globserv.org/cps.txt</Audience>
    </AudienceRestriction>
  </Conditions>

  <Subject>
    .....
  </Subject>
  <AuthnStatement>
    .....
  </AuthnStatement>
</Assertion>
```



Jahan Moreh

Strong Web Service Authentication Using SAML

Session: E-F2, April 28, 2008



Example of a SAML Assertion Containing an an Attribute Statement

```
<Assertion MajorVersion="2" MinorVersion="0"
  ID="urn:SigAuth:e587bfe6c7@SIGABA.COM:3d"
  IssueInstant="2004-06-19T11:19:00.000-08:00" >
  <Issuer> mechanism/SigAuth@SIGABA.COM </Issuer>
  <Subject>
    <NameID
      NameQualifier="sigaba.com"
      Format="urn:oasis:names:tc:SAML:1.1:nameidformat:emailAddress">
        jmoreh@sigaba.com
      </NameID>
  </Subject>
  <AttributeStatement>
    <Attribute
      Nameformat="urn:oasis:names:tc:SAML:1.1:nameidformat:emailAddress"
      Name="alias">
      <AttributeValue>jahan@unex.ucla.edu</AttributeValue>
    </Attribute>
  </AttributeStatement>
</Assertion>
```



Jahan Moreh

Strong Web Service Authentication Using SAML

Session: E-F2, April 28, 2008



SAML Protocol

- SAML defines the schema for request/response messages
- SAML defines query/reply protocols that use these messages
- Example of protocols:
 - Authentication Query: used to query about existing Assertions that contain an Authentication Statement about a Subject
 - Authentication Request: used to request an Identity Provider to authenticate the Subject and produce an Assertion containing an Authentication Statement
 - Artifact Resolve: used to obtain an Assertion by reference
 - Attribute Query: used to request a set of attributes about the Subject

Example of Authentication Request

<Request

```
MajorVersion="2" MinorVersion="0"  
ID="urn:SigAuth:298cd2f311@SIGABA.COM:f0"  
IssueInstant="2004-06-19T11:18:59.000-08:00">
```

continued on next page...

Example of Authentication Request

```

<AuthnRequest
  MajorVersion="2" MinorVersion="0" ID="urn:SigAuth:298cd2f311@SIGABA.COM:f0"
  IssueInstant="2004-06-19T11:18:00.000-08:00" >
  <Subject>
    <NameID
      NameQualifier="sigaba.com"
      Format="urn:oasis:names:tc:SAML:1.1:nameidformat:emailAddress">
      jmoreh@sigaba.com
    </NameID>
  </Subject>
  <RequestedAuthnContext>
    <AuthnContextClassRef>
      urn:oasis:names:tc:SAML:2.0:ac:classes:Password
    </AuthnContextClassRef>
  </RequestedAuthnContext>
</AuthnRequest>
</Request>

```



Jahan Moreh

Strong Web Service Authentication Using SAML

Session: E-F2, April 28, 2008



Example of Authentication Response

```

<Response
  MajorVersion="2" MinorVersion="0"
  ID="urn:SigAuth:98ec2c590f@SIGABA.COM:26"
  InResponseTo="urn:SigAuth:298cd2f311@SIGABA.COM:f0"
  IssueInstant="2004-06-19T11:19:00.000-08:00">
  <Status>
    <StatusCode
      Value="urn:oasis:names:tc:SAML:2.0:status:Success">
    </StatusCode>
  </Status>
  <Assertion MajorVersion="2" MinorVersion="0"
    ID="urn:SigAuth:e587bfe6c7@SIGABA.COM:3d"
    IssueInstant="2004-06-19T11:19:00.000-08:00" >
    <Issuer> mechanism/SigAuth@SIGABA.COM </Issuer>

```

Continued on the next page.....



Jahan Moreh

Strong Web Service Authentication Using SAML

Session: E-F2, April 28, 2008



Example of Authentication Response (cont.)

```

<Subject>
  <NameID
    NameQualifier="sigaba.com"

    Format="urn:oasis:names:tc:SAML:1.1:nameidformat:emailAddress">
    jmoreh@sigaba.com
  </NameID>
</Subject>
<AuthnStatement
  AuthnInstant="2004-06-19T11:18:00.000-08:00" >
  <AuthnContext>
    <AuthnContextClassRef>
      urn:oasis:names:tc:SAML:2.0:ac:classes:Password
    </AuthnContextClassRef>
  </AuthnContext>
</AuthnStatement>
</Assertion>
</Response>

```

SAML Protocol Bindings

- Protocol Binding specify how SAML protocol messages are carried
- Variety of bindings are envisioned
- SAML specifies the following bindings
 - SOAP
 - Reverse SOAP (POAS)
 - HTTP Redirect
 - HTTP POST
 - HTTP Artifact
 - SAML URI
- Some bindings define a *RelayState* for conveying and preserving state information across the request/reply protocol

SAML Profiles

- Profiles define the use of SAML assertions and request-response messages in communications protocols and frameworks
- SAML envisions a variety of profiles
- SAML specifies the following profiles
 - A set of SSO profiles including
 - Web Browser SSO
 - Enhanced Client or Proxy (ECP)
 - Identity Provider Discovery
 - Single Logout
 - Name Identifier Management
 - Artifact Resolution
 - Assertion Query/Request
 - Name Identifier Mapping
 - A set of Attribute Profiles

SAML

Web Browser SSO Profile

- Sets of rules describing how to embed SAML assertions into and extract them from HTTP and SOAP-based protocols
- Support Browser Single Sign-on
- Subject Confirmation Method
 - urn:oasis:names:tc:SAML:2.0:cm:bearer
- Assertions are recommended to be *single use*
- Rely on SSL and/or signed assertions

Artifact-specific Rules

- The Service Provider and the Identity Provider must establish a mutually authenticated session
- Identity Provider enforces the following rules:
 - Assertion is one-time-use
 - Assertion will be returned only to a requester for whom the assertion was intended
 - Requester must be authentic (e.g., via SSL client authentication)

POST-specific Rules

- The SAML <Response> **must** be digitally signed
- Service Provider enforces the following rules:
 - Assertion is one-time-use
 - Response's digital signature must be valid
 - Response's <Recipient> attribute must match the Service Provider

Web Services Security Header

- WSS Security provides a new extension for the SOAP header
 - `<wsse:Security>`
- The security header is targeted at a specific SOAP *role/actor*
- There may be multiple `<wsse:Security>` elements
 - Two security header blocks **MUST NOT** target the same role

WSS SAML Token Profile

- SAML Assertions can be carried and referenced in `<wsse:security>`
- Receiver must establish the relationship between:
 - The *subject* in the SAML statement
 - The entity providing the SAML assertion

Confirming Assertions

- WSS envisions two confirmation methods from SAML
 - *holder-of-key*
 - *sender-vouches*
- XML Signature is strongly recommended as means of confirmation

Example of SAML Token Profile

```
<wsse:Security>
  <saml:Assertion>
    AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
    IssueInstant="2003-04-17T00:46:02Z"
    Issuer="www.opensaml.org"
    MajorVersion="1"
    MinorVersion="1"
    xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
      <saml:Conditions>
        NotBefore="2003-07-19T16:53:33.173Z"
        NotOnOrAfter="2003-07-19T17:08:33.173Z"
      </saml:conditions>
    </saml:assertion>
  </saml:Assertion>
</wsse:Security>
```

continued on next page.....

Example of SAML Token Profile

```

<saml:AuthenticationStatement>
  <saml:Subject>
    <saml:NameIdentifier
      NameQualifier=ucla.edu
      Format="#emailaddress">
      Jmoreh@ucla.edu
    </saml:NameIdentifier>
    <saml:SubjectConfirmation>
      <saml:ConfirmationMethod>
        urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
      </saml:ConfirmationMethod>
      <ds:KeyInfo>
        <ds:KeyValue>...</ds:KeyValue>
      </ds:KeyInfo>
    </saml:SubjectConfirmation>
  </saml:Subject>
</saml:AuthenticationStatement>
<ds:Signature>...</ds:Signature>
</saml:Assertion>

```

Example of SAML Token Profile

```

<ds:Signature>
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#" />
<ds:SignatureMethod
  Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1" />
<ds:Reference URI="#MsgBody">
.....
<wsse:SecurityTokenReference wsu:id="STR1">
<wsse:Reference wsu:id="..."
  wsse:ValueType="http://www.docs.oasis-open.org/wss/.../SAMLAssertion-1.0"
  wsse:URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc" />
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>

```

Conclusions

- There is no single definitive standard for securing web services
- XML and non-XML security technologies are equally applicable to Web Services
- Web Services can be secured at multiple levels
- Use of specific profiles enhances interoperability
- SOA/Web Service architects should seriously consider using SAML for integrating security into applications