Distributed Services
Security using the SAML

Krishna Sankar
ksankar@cisco.com
Agenda

- Introduction To XML Security
- SAML
  - Assertions
  - Protocols
- Use Cases
  - Transitive Trust For Distributed Systems
  - Distributed Assertion Exchange
  - Distributed AuthC & AuthZ Single Sign-on

Thanks to Eve Maler (Sun), Prateek Mishra (Netegrity), RLBob Morgan (UWash/Internet2), and Darren Platt (RSA) for a lot of material in this presentation
XML Security Standards – What for?

**Assertions Exchange**
- Transport Neutral
- Interoperable

**Identity of sender & receiver**
- Is From the right entity
- Got to the right entity
- Assurance of delivery
  - They got it & receipt recvd
  - non repudiation
XML Security Standards – What for?

- Transport security
  - Confidentiality
  - Integrity
  - Availability
  - Handle Exceptions in transport, applications, intermediaries

- Audit, legal issues

- Web Services Security/Trust Model
OASIS - SAML

Security Assertions Mark-up language

“The purpose of the XML-Based Security Services TC (SSTC) is to define an XML framework for exchanging authentication and authorization information”

What is SAML?

- XML based Framework
- A set of XML vocabularies for:
  - Authentication Assertion
  - Attribute Assertion
  - Session Assertion (Future)
  - Credential Assertion (Future)
- So that data traveling on the wire is standardized
What is SAML?

- A standard message exchange protocol
- Clarity in orchestrating how you ask for and get the information you need
- Rules for how the messages ride “on” and “in” transport protocols
- For better interoperability
What problems does SAML try to solve?

- Security data is shared in mostly proprietary ways
  - The different systems that generate and use security data are very tightly coupled
  - Integrating new security features may require developing a lot of new code

- Web-based applications show the need for more federation
  - We need to cross domains more easily
  - Internal & External Domains
In Short … SAML is …

A standard way of exchanging security & related data across heterogeneous, distributed systems crossing domain (geographical, namespace, temporal, spatial, organizational,…) boundaries
SAML assertions

 Assertions are declarations of fact, according to someone

 SAML assertions are compounds of one or more of three kinds of “statement” about “subject” (human or program):
 - Authentication
 - Attribute
 - Authorization decision

 You can extend SAML to make your own kinds of assertions and statements

 Assertions can be digitally signed
All statements in an assertion share common information

- Issuer ID and issuance timestamp
- Assertion ID
- Subject
  - Name plus the security domain
  - Optional subject confirmation, e.g. public key
- “Conditions” under which assertion is valid
  - SAML clients *must reject* assertions containing unsupported conditions
  - Special kind of condition: assertion validity period
- Additional “advice”
  - E.g., to explain how the assertion was made
Assertion structure
Example common information for an assertion

```xml
<Assertion
    MajorVersion="1" MinorVersion="0"
    AssertionID="128.9.167.32.12345678"
    Issuer="Smith Corporation"
    IssueInstant="2001-12-03T10:02:00Z">
    <Conditions
        NotBefore="2001-12-03T10:00:00Z"
        NotOnOrAfter="2001-12-03T10:05:00Z">
        <AudienceRestrictionCondition>
            <Audience>...URI...</Audience>
        </AudienceRestrictionCondition>
    </Conditions>
    <Advice>
        ...a variety of elements can go here...
    </Advice>
    <Advice>
        ...statements go here...
    </Advice>
</Assertion>
```
Authentication statement

- An issuing authority asserts that subject S was authenticated by means M at time T
- Targeted towards SSO uses
- **Caution:** Actually checking or revoking of credentials is not in scope for SAML!
- It merely lets you link back to acts of authentication that took place previously
Authentication statement structure
Example assertion with authentication statement

```xml
<saml:Assertion ...>
  <saml:AuthenticationStatement
    AuthenticationMethod="password"
    AuthenticationInstant="2001-12-03T10:02:00Z">
    <saml:Subject>
      <saml:NameIdentifier
        SecurityDomain="smithco.com"
        Name="joeuser" />
      <saml:ConfirmationMethod>
        http://...core-25/sender-vouches
      </saml:ConfirmationMethod>
    </saml:Subject>
  </saml:AuthenticationStatement>
</saml:Assertion>
```
Attribute statement

- An issuing authority asserts that subject S is associated with attributes A, B, … with values “a”, “b”, “c”…
- Useful for distributed transactions and authorization services
- Typically this would be gotten from an LDAP repository
  - “john.doe” in “example.com”
  - is associated with attribute “Department”
  - with value “Human Resources”
Attribute statement structure
Example assertion with attribute statement

```xml
<saml:Assertion ...>
  <saml:AttributeStatement>
    <saml:Subject>...</saml:Subject>
    <saml:Attribute
        AttributeName="PaidStatus"
        AttributeNamespace="http://smithco.com">
          <saml:AttributeValue>
            PaidUp
          </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
        AttributeName="CreditLimit"
        AttributeNamespace="http://smithco.com">
          <saml:AttributeValue>
            <my:amount currency="USD">500.00</my:amount>
          </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```
Authorization decision statement

- An issuing authority decides whether to grant the request by subject S for access type A to resource R given evidence E
- Useful for distributed transactions and authorization services
- The subject could be a human or a program
- The resource could be a web page or a web service, for example
Authorization decision statement structure
Example assertion with authorization decision statement

```xml
<saml:Assertion ...>
    <saml:AuthorizationStatement
        Decision="Permit"
    
        Resource="http://jonesco.com/rpt_12345.htm">
            <saml:Subject>...</saml:Subject>
            <saml:Actions
                ActionNamespace="http://...core-25/rwedc">
                    <saml:Action>Read</saml:Action>
                </saml:Actions>
            </saml:AuthorizationStatement>
        </saml:Assertion>
```
SAML protocol

- Asserting Party
- SAML
  - Request for Assertion of Certain Type
  - Response Assertion
- Relying Party

Diagram showing the flow of SAML protocol with the parties and their interactions.
Assertions are normally provided in a SAML response

- Existing tightly coupled environments may need to use their own protocol
  - They can use assertions without the rest of the structure

- The full benefit of SAML will be realized where parties with no direct knowledge of each other can interact
  - Via a third-party introduction
Requests can take several forms

- You can query for specific kinds of assertion/statement
  - Authentication query
  - Attribute query
  - Authorization decision query

- You can ask for an assertion with a particular ID
  - By providing an ID reference
  - By providing a SAML “artifact”
Authentication query

“Please provide the authentication information for this subject, if you have any”

It is assumed that the requester and responder have a trust relationship
- They are talking about the same subject
- The response with the assertion is a “letter of introduction” for the subject
Authentication query structure
Example request with authentication query

```xml
<samlp:Request
  MajorVersion="1" MinorVersion="0"
  RequestID="128.14.234.20.12345678">
  <samlp:AuthenticationQuery>
    <saml:Subject>
      <saml:NameIdentifier
        SecurityDomain="smithco.com"
        Name="joeuser"/>
    </saml:Subject>
  </samlp:AuthenticationQuery>
</samlp:Request>
```
**Attribute query**

“Please provide information on the listed attributes for this subject”

If you don’t list any attributes, you’re asking for all available ones

If the requester is denied access to some of the attributes, only the allowed attributes would be returned

(This situation is indicated in the status code of the response)
Attribute query structure
Example request with attribute query

```xml
<samlp:Request ...
  <samlp:AttributeQuery>
    <saml:Subject>
      <saml:NameIdentifier
        SecurityDomain="smithco.com"
        Name="joeuser" />
    </saml:Subject>
    <saml:AttributeDesignator
      AttributeName="PaidStatus"
      AttributeNamespace="http://smithco.com">
    </samlp:AttributeDesignator>
  </samlp:AttributeQuery>
</samlp:Request>
```
Authorization decision query

“Is this subject allowed to access the specified resource in the specified manner, given this evidence?”

This is is a yes-or-no question

- The answer is not allowed to be “no, but they’re allowed to access these other resources”
- Or “yes, and they’re also allowed to perform these other actions”
Authorization decision query structure
Example authorization decision query

<
samlp:Request ...
<samlp:AuthorizationQuery

Resource="http://jonesco.com/rpt_12345.htm">
<saml:Subject>
<saml:NameIdentifier
    SecurityDomain="smithco.com"
    Name="joeuser" />
</saml:Subject>
<saml:Actions
    ActionNamespace="http://...core-25/rwedc">
    <saml:Action>Read</saml:Action>
</saml:Actions>
<saml:Evidence>
    <saml:Assertion>...</saml:Assertion>
</saml:Evidence>
</samlp:AuthorizationQuery>
</samlp:Request>
Responses just contain a set of assertions

- Or one or more assertions can be returned with status information
- If something went wrong, no assertions are returned, just status
  - Status information can have a complex structure
- Currently the status codes are:
  - Success
  - VersionMismatch
  - Receiver
  - Sender
- Responses are expected to be signed
Response structure
Example response

```xml
<response
    MajorVersion="1" MinorVersion="0"
    RequestID="128.14.234.20.90123456"
    InResponseTo="128.14.234.20.12345678"
    StatusCode="Success">
<assertion
    MajorVersion="1" MinorVersion="0"
    AssertionID="128.9.167.32.12345678"
    Issuer="Smith Corporation">
<conditions
    NotBefore="2001-12-03T10:00:00Z"
    NotAfter="2001-12-03T10:05:00Z" />
<authenticationStatement ...>...
</authenticationStatement>
<assertion>
</response>
```
Use Cases

- Transitive Trust For Distributed Systems
- Distributed Assertion Exchange
- Distributed AuthC & AuthZ
Transitive Trust For Distributed Systems

Browser based single-sign-on Applications:
- Eco Systems
- Consortiums
- AOL/Passport/LA based applications
“While PKI will authenticate and control user access to the front-end application, its capabilities do not extend to the connection between the front-end application and the back-end system,” Foody said. “Security is a challenge for integrated e-business systems that are multi-tiered and link multiple applications.”

http://www.techrepublic.com/printerfriendly.jhtml;jsessionid=L3MAO24ZPS2TGCTEAAKSFFA?id=r00820010810swm01.htm
Distributed Assertion Exchange

- Session handling
- Context for web services
- Applications:
  - B2B Eco Systems
  - Web Services
SOAP Profile:
Exchanging SAML assertions with pre-established relationship.
CA

Sender | SAML Processor | KeyStore/PKI/AMPS | Receiver | SAML Processor | KeyStore/PKI/AMPS

Create Assertion() -> getKey() -> Generate Short-lived Certificates() -> Create Assertion incl embedded SAML Certificate() -> Attach SAML assertions to SOAP message() -> SOAP Message w/ Assertions()

E.g.: Context between web services as SAML attribute assertions

SOAP Fault()

<Fault>
  <Faultcode>Client.SAML</Faultcode>
  <Faultstring>Cannot resolve key</Faultstring>
</Fault>

SOAP Fault()

<Fault>
  <Faultcode>Client.SAML</Faultcode>
  <Faultstring>subject not authorized</Faultstring>
</Fault>

SOAP Profile:
Exchanging SAML assertions based on short-lived certificates, requires no established relationships
Widely Distributed AuthC & AuthZ

- Collaboration across multiple, independent and geographically dispersed stakeholders
- Stakeholders able to enforce policies even when controlled by different administrative domains
  - Traditional: ACLs
    - Cannot scale. Cause too many errors
- Multiple layers of management would impose restrictions

Courtesy: DOE report: LBNL-42928: Certificate-based Access Control for widely distributed resources
OASIS - XACML

**eXtensible Access Control Markup Language**

An XML specification for expressing policies for information access over the Internet

Example: DRM, Clinical Records, Workflow, Attribute dependent access control (temporal, different roles, different groups, geographies)

**Issues to be addressed include, but are not limited to:**

- Fine grained control, The nature of the requestor,
- The protocol over which the request is made,
- Content introspection, The types of activities authorized.

**http://www.oasis-open.org/committees/xacml/index.shtml**
Access Groups are Defined by Several Required Attributes

3

Stakeholder_2 requires Attribute_3 and Attribute_4

Stakeholder_3 requires Attribute_5 and Attribute_6

set of users who posses Attribute_3 and Attribute_4

set of users who posses Attribute_5 and Attribute_6

sets of required attributes define access-groups

set of users who posses Attribute_1 and Attribute_2

Stakeholder_1 requires Attribute_1 and Attribute_2

separate attributes may be required for specific actions on the resource

Stakeholder_4 requires Attribute_6 for Action_1

Access-group possessing the attributes required by all stakeholders
(1) Use-conditions are Imposed by Independent Stakeholders

Stakeholders provide and maintain and use-conditions

DOE-HQ

LBNL

ALS

ALS Medical Beamline

STOP

access control gateway

(2) Users have Attributes that Match the Use-conditions

SAML

XACML

User Attributes Match the Required Use-Conditions

Access is Granted after Verifying

Country

LBNL Employee or Visitor

Building 101

Approved Access

Control Model

Certificate for Widely Distributed Access Control

Courtsey: DOE report: LBNL-41349: Authorization & Attribute Certificates for Widely Distributed Access Control
Questions ?
Questions ?