SAML V2.0 Basics

Eve Maler
eve.maler@sun.com
Sun Microsystems, Inc.

Updated 2 October 2006
This presentation may be copied and reused with attribution
Topics

• The big picture
• The standards landscape
• SAML concepts and terms
• SAML assertions
• Major SAML usage scenarios
• How you can get started
• Resources
The big picture
Opportunities with distributed identity

- **People can:**
  - Avoid authenticating repeatedly
  - Unify management of their identity information
  - Have better-personalized online experiences
  - Gain better privacy control

- **Services and applications can:**
  - Offload authentication and identity lookup tasks
  - Unify treatment of all “things with identities”
  - Provide finer-grained access control and differentiation

- **Organizations can:**
  - More securely outsource business functions
Essential roles in a distributed identity architecture

- Identity-attesting entities
- Identity-wielding entities
- Identity-relying entities

Identity providers, asserting parties, identity services, homesites...

Relying parties, service providers, membersites...

Principals, subjects... Is verifiable and interacts through a client of some kind
Identity data distribution flows and containers

Requests
- Identity-relying entities
- Identity-attesting entities
- Identity-wielding entities

Responses
- Identity-relying entities
- Identity-wielding entities
- SMTP, HTTP, SOAP...

Identity data
- Claims, assertions, credentials, attributes...
The basic use case for single sign-on (SSO)

1. Authenticate

2. Access protected resource

Sign-On Site (e.g., employer or university)

Business agreement

Site with Protected Content (e.g., research article)

Identity information (e.g., authn and attributes)
Fleshing out a scenario: web-based IM

• Specialized web IM application in a manufacturing environment
  – Used to notify repair personnel about, and let them discuss, equipment breakdown episodes

• Employees have IM access by virtue of:
  – A valid login to the company portal
  – A role of “repair_tech”

• An identity service can ping all online repair techs automatically to discuss malfunction triage situations

• Employees can log out of their portal, IM, and all their other work apps in one step
Technical challenges with distributed identity

- Distributing identity info across domain boundaries in the first place – privacy, security, accuracy, compliance...

- Getting the identity info semantics right – the syntax is the comparatively easy part

- Security solutions at the application layer never absolve you from providing security below
Requirements for distributed identity

- Standard, flexible **formats** for identity information
- **Protocols** that are standard, secure, privacy-enabled, technology-neutral, and interoperable for exchanging identity information between components of distributed applications
- A way to set up **trust relationships** between entities that share identity information within technical, business, and legal frameworks
The standards landscape
The overall identity landscape (YMMV)

Focusing on SAML

• The Security Assertion Markup Language in six words: “The universal solvent of identity information”

• Best supported and most thoroughly standardized, covering a wide range of distributed-identity scenarios
  – Reflects the convergence of several development streams
  – Enables privacy along various dimensions

• Many other specs and standards build on it
Liberty / SAML / Shibboleth: one degree of separation

- **SAML V1.0**
  - Liberty Alliance (Jul 2002)
  - OASIS SSTC (Nov 2002)

- **SAML V1.1**
  - OASIS Contribution (Sep 2003)
  - OASIS Participation (Mar 2005)

- **SAML V2.0**
  - OASIS Contribution (Apr 2005)
  - Under way

- **ID-FF V1.1 (Liberty Federation)**
  - ID-FF V1.2 (Liberty Federation)
  - Liberty Alliance (Jan 2003)
  - Nov 2003

- **“Phase 1”**
  - Jul 2002

- **Shib V1.1**
  - Shib 1.0/1.1 APIs (Jul/Aug 2003)
  - Shib APIs (Apr 2004)

- **Shib V1.2 APIs**
  - Apr 2004

- **ID-FF V1.2 (Liberty Federation)**
  - Apr 2005

- **OASIS**
  - Contribution
  - Participation

- **SSTC**

- **Internet2**

- **SAML V2.0 adoption as Liberty Federation**
  - Under way
SAML concepts and terms
SAML in a technical nutshell

• SAML in 15 words: “XML-based framework for marshaling security and identity information and exchanging it across domain boundaries”
  • It wraps existing security technologies rather than inventing new ones
  • Its profiles offer interoperability for a variety of use cases, but you can extend and profile it further

• At SAML's core: assertions about subjects
  • Authentication, attribute, entitlement, or roll-your-own

Identity data

Claims, assertions, credentials, attributes...
SAML components and how they relate to each other

Profiles
Combinations of assertions, protocols, and bindings to support a defined use case (also attribute profiles)

Bindings
Mappings of SAML protocols onto standard messaging and communication protocols

Protocols
Requests and responses for obtaining assertions and doing identity management

Assertions
Authentication, attribute, and entitlement information

Authentication Context
Detailed data on types and strengths of authentication

Metadata
Configuration data for identity and service providers
The SAML specifications map to them fairly closely.

0. Managers: start here!
1. Techies: start here...
2. ...then proceed to here...
3. ...and move leftward...
4. ...as necessary.

N.B.: Always work from the errata composite version of the spec if there is one.
Language about subjects

• **Entity (or system entity):** An active element of a computer/network system

• **Principal:** An entity whose identity can be authenticated

• **Subject:** A principal in the context of a security domain
Language about identities

- **Identity**: The essence of an entity, often described by one's characteristics, traits, and preferences
  - **Anonymity**: Having an unknown/concealed identity

- **Identifier**: A data object that uniquely refers to a particular entity
  - **Pseudonym**: A privacy-preserving identifier
(More) language about identities

• **Federated identity:** Existence of an agreement between providers on a set of identifiers and/or attributes to use to refer to a principal
  
  – **Account linkage:** Relating a principal's accounts at two providers so they can communicate about it
Language about (more) entities

• **Asserting party (SAML authority):** An entity that produces SAML assertions
  
  - **Identity provider:** An entity that creates, maintains, and manages identity information for principals and provides principal authentication to other service providers
(More) language about (more) entities

- **Relying party**: An entity that decides to take an action based on information from another system entity
  - **Service provider**: An entity that provides services to principals or other entities
SAML assertions
Assertion basics

• An assertion is a claim made by someone about someone
• SAML assertions are structured as a series of statements about a subject:
  – **Authentication** statement: “Sam authenticated with a smartcard PKI certificate at 9:07am today”
  – **Attribute** statement (which can contain multiple attributes): “Sam is a manager and has a $5000 spending limit”
  – **Authorization decision** statement (now deprecated): “Yes, Sam can read that web page”
  – Your own customized statements...
Example of an assertion's common portions

```xml
<saml:Assertion
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  Version="2.0"
  IssueInstant="2006-07-28T14:01:00Z">
  <saml:Issuer>
    www.emeffgee.com
  </saml:Issuer>
  <saml:Subject>
    <saml:NameID
      Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
      J.Handy@emeffgee.com
    </saml:NameID>
  </saml:Subject>
  <saml:Conditions
    NotBefore="2006-07-28T14:00:05Z"
    NotOnOrAfter="2006-07-28T14:05:05Z">
  ... statements go here ...
  </saml:Conditions>
</saml:Assertion>
```
Overall assertion element structure

1. I'm telling you
   - `saml:Issuer`

2. (yes, it's really me)
   - `ds:Signature`

3. about this guy/gal/thing.
   - `saml:Subject`

4. Make sure to follow these rules in using this information.
   - `saml:ConditionsType`
     - `saml:Condition`
     - `saml:AudienceRestriction`
     - `saml:OneTimeUse`
     - `saml:ProxyRestriction`
     - `NotBefore`
     - `NotOnOrAfter`

5. By the way, did you know that...?
   - `saml:AdviceType`
     - `saml:AssertionIDRef`
     - `saml:AssertionURIRef`

6. Okay, so here's what you need to know about this guy/gal/thing:
   - `saml:Statement`
     - `saml:AuthnStatement`
     - `saml:AuthzDecisionStatement`
     - `saml:AttributeStatement`
Subject element structure

1. Here's his/her/its unique identifier

2. (which might be scrambled).

3. Here's a way to securely confirm that the guy you got this from is the same as the guy I'm telling you about.

4. For example, making him prove he has a specific key would suffice...
Example of an authentication statement

<saml:Assertion ...
  common info goes here ...
>
  ... and here ...

  <saml:AuthnStatement
    AuthnInstant="2006-07-28T14:00:05Z"
    SessionIndex="0">
    <saml:AuthnContext>
      <saml:AuthnContextClassRef>
        urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
      </saml:AuthnContextClassRef>
    </saml:AuthnContext>

  </saml:AuthnStatement>

  ... more statements might appear here ...

</saml:Assertion>
Authentication statement element structure

- AuthnInstant
- SessionIndex
- AuthnContext

When it happened
Just in case he/she/it has more than one session going at once

The type of authentication (how “strong” it was)
Authentication context classes

SAML comes with a healthy set of predefined identifiers for typical authentication scenarios:

- Internet Protocol
- Internet Protocol Password
- Kerberos
- Mobile One Factor Unregistered
- Mobile Two Factor Unregistered
- Mobile One Factor Contract
- Mobile Two Factor Contract
- Password
- Password Protected Transport
- Previous Session
- Public Key – X.509
- Public Key – PGP
- Public Key – SPKI
- Public Key – XML Signature
- Smartcard
- Smartcard PKI
- Software PKI
- Telephony
- Nomadic Telephony
- Personalized Telephony
- Authenticated Telephony
- Secure Remote Password
- SSL/TLS Cert-Based Client Authentication
- Time Sync Token
- Unspecified

You can also create or customize your own authentication context classes...
Example of an attribute statement

<saml:Assertion ... common info goes here ... >
  ... and here ...
  <saml:AttributeStatement>
    <saml:Attribute
      NameFormat="http://emeffgee.com" Name="Role">
      <saml:AttributeValue>repair_tech</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute
      NameFormat="http://emeffgee.com">
      Name="Certification"
      <saml:AttributeValue xsi:type="emeffgee:type">
        <emeffgee:CertRecord language="EN">
          <Course>
            <Name>Structural Repair</Name>
            <Credits>3</Credits>
          </Course> ...
        </emeffgee:CertRecord>
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
Attribute statement element structure

1. Here's an attribute name/value pair
2. (which could be one of many).
3. The whole thing could be provided in scrambled form.
4. Here's the name and how to interpret it.
5. Here's the value
6. (which could be one of many)
7. and have an arbitrary string or XML value).
Attribute profiles

• Basic
  – Simple string-based SAML attribute names

• X.500/LDAP
  – Common standardized convention for SAML attribute naming using OIDs, expressed as URNs

• UUID
  – SAML attribute names as UUIDs, expressed as URNs

• DCE PAC
  – Representation of DCE realm, principal, and group membership information in SAML attributes

• XACML
  – How to map SAML attributes cleanly to XACML attribute representation

• XPath (draft)
  – XPath expression pointing to the attribute values within an XML document as the attribute name – has utility in identity services

• Your own customized attribute profiles...
So far, no interchange – just format

- SAML assertions are becoming the way to marshal packets of identity information
  - They wrap existing authentication and attribute (and authorization) semantics rather than inventing new ones

- Getting them from point A to point B has two interesting aspects:
  - **Why?** What purpose is being served in sending and getting them?
  - **How?** Along what channels do they flow?
  - There are security and privacy implications for both
Request/response protocols

- Assertions are requested, provided as input, and returned as output in the course of doing these jobs.
- SAML defines various XML request/response protocol message pairs
  - All based on a hierarchy of complex data types in the protocol schema.
- The messages can be conveyed using various communications protocols through SAML bindings.
Major SAML usage scenarios
Key use cases covered by SAML profiles

• Single sign-on
  – Using standard browsers and enhanced clients (such as handheld devices)

• Federating identities
  – Using a well-known identifier or a privacy-preserving pseudonym

• Attribute services

• Single logout

• You can create your own profiles...
  – E.g., WS-Security defines a SAML Token Profile for securing web services
The vanilla Web SSO profile

- **Goal:** J. Handy, repair tech, signs in only once whenever using the company portal *and* the IM app

- **Requirement:** The portal has to prove to the IM app that J. is authenticated, and also provide attributes that will let the portal make an authorization decision

- **The players:**
  
  - **EmEffGee portal** (identity-asserting) 
    - IdP
  
  - **J. Handy, repair tech** (identity-wielding)
  
  - **EmEffGee web IM/chat** (identity-relying) 
    - SP
A (mockup of a)n IM conversation

- J. Handy has logged in either at the IM app prompt or the company portal
- Engine #6 itself alerted all online techs to the overheating
- Jamie has determined through presence/location services where the other techs are
- Meebo might be providing infrastructure that EmEffGee hosts itself, or alternatively, it's an outsourced service
Several options for information flow

• Does J. visit the portal or the IM app first?
  – If J. tries to use IM first, the IM app has to explicitly request info from the portal
  – The SSO assertion has to be conveyed from IdP to SP regardless, using a response message

• If the IM app makes a request, does it push (HTTP POST), allow to be pulled (“artifact”), or use HTTP redirect for the request?

• Does the portal push (HTTP POST) or allow to be pulled (“artifact”) the response?

• Let's see...carry the two...that's eight options
  – But some are more common than others
SP-initiated flow with redirect and POST bindings

1. User or UA action
2. Supply resource
3. Challenge for credentials
4. User login
5. GET using <AuthnRequest>
6. POST signed <Response> in HTML form
7. Redirect with <AuthnRequest>

Service Provider www.abc.com
- Resource
- Access check
- Assertion Consumer Service

Identity Provider www.xyz.com
- Single Sign-On Service

Browser
- User or UA action
Example of an authentication request

```xml
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    Version="2.0"
    ID="f0485a7ce95939c093e3de7b2e2984c0"
    IssueInstant="2006-07-28T14:01:05Z"
    Destination="https://www.emeffgee.com/IdP/"
    AssertionConsumerServiceIndex="1"
    AttributeConsumingServiceIndex="0">
    <samlp:RequestedAuthnContext>
        <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        </saml:AuthnContextClassRef>
    </samlp:RequestedAuthnContext>
    <samlp:NameIDPolicy
        Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
    </samlp:NameIDPolicy>
</samlp:AuthnRequest>
```
Authentication request element structure

1. I'm asking you
2. (yes, it's really me)
3. about this guy/gal/thing.
4. Here is the kind of identifier I want,
5. and the type of authentication I want used.
6. the conditions the assertion must meet,
7. Please adhere to these constraints on IdP proxying.
8. Don't forget these other instructions...

Don't forget these other instructions...
IdP-initiated flow with the POST binding

Service Provider
www.abc.com

Resource

Assertion Consumer Service

Identity Provider
www.xyz.com

Single Sign-On Service

Access check

User or UA action

Browser

1. Challenge for credentials
2. User login
3. Signed <Response> in HTML form
4. Select remote resource
5. POST signed <Response>
6. Supply resource
7. User or UA action
SSO using an enhanced client

• SAML defines an “Enhanced Client or Proxy” SSO profile for:
  – Proxy servers such as WAP gateways in front of limited-ability mobile devices
  – Clients that can't use HTTP redirects
  – Accommodating the inability of the IdP and SP to communicate, for whatever reasons
    • It might be an architectural choice

• In some circumstances the web client is smarter than the average bear
  – An ECP client can use the PAOS binding to communicate cleverly via SOAP and HTTP
  – It may also be clever about where to find IdPs
  – It can even be an IdP
ECP use cases

Examples: handsets, medical devices, set-top boxes

Enhanced client

Identity Provider

Service Provider

Enhanced proxy

Example: WAP gateway

Identity Provider

Service Provider
SSO using ECP

Service Provider
www.emeffgee.com/IM

Identity Provider
www.emeffgee.com/portal

Resource

Access check

Assertion Consumer Service

Single Sign-On Service

<AuthnRequest> in PAOS request

<Response> in PAOS response

<AuthnRequest> in SOAP request

<Response> in SOAP response

Enhanced Client

SOAP intermediary

Access resource

Supply resource

1

2

3

4

5

6
Account linking with privacy and flexibility

• SSO involves only one-way information flow
  – The IM/chat app need not have a “local account” for J. Handy at all
  – Typically, however, it does because its relationship with J. is non-trivial

• Two-way flow of information is often desired to synchronize identity data stores

• The two apps could become IdPs for each other by “opening the kimono” and sharing J.'s identifier for correlation (federation)
  – But it's J.'s kimono!
The basic use case for pairwise identity federation

- Book flight logged in as **johndoe**
- Prepare to rent car logged in as **jdoe**; accept offer of federation with AirlineInc.com
- Prepare to book hotel logged in as **johnd**; accept offer of federation with AirlineInc.com

Agree on **azqu3H7** for referring to John (neither knows the local ID used on the other side)

Agree on **f78q9c0** for referring to John (neither knows the local ID used on the other side)

No correlation of John's activities across these sites
SAML's name identifier management profile

• Providers can set up pairwise-unique nicknames for J. Handy
  – One option is a **persistent pseudonym**, for an ongoing portal+IM app relationship
  – Another is a **transient pseudonym**, e.g. for single-session access granted to groups based on attributes

<table>
<thead>
<tr>
<th>EmEffGee portal</th>
<th>EmEffGee web IM/chat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local username: <a href="mailto:J.Handy@EmEffGee.com">J.Handy@EmEffGee.com</a></td>
<td>Local username: jamie_handyMFG2006</td>
</tr>
<tr>
<td>Local password: ALiST0e5440</td>
<td>Local password: I'mRich$$!</td>
</tr>
</tbody>
</table>

Opaque handle: a42b3543af
Out-of-band federation

Service Provider
www.CarRentallInc.com

Identity Provider
www.AirlineInc.com

Browser

User with local ID john at both providers

Normal SAML web single sign-on profile

1. Supply resource
2. Pass along signed <Response> about john
3. Convey signed <Response> about john
4. User login
5. Challenge for credentials
6. Pass along <AuthnRequest>
7. Convey <AuthnRequest>
8. Access resource

Link CarRentallInc's john with AirlineInc's john outside of SAML

Identity store

Identity store

Access check

Assertion Consumer Service

Single Sign-On Service

Resource

Assertion Consumer Service

Pass along <AuthnRequest>

Challenge for credentials

Convey signed <Response> about john

Convey signed <Response> about john

Pass along signed <Response> about john
Federation with a persistent pseudonym

Browser

Service Provider www.CarRentalInc.com

Identity Provider www.AirlineInc.com

User with local ID john at AirlineInc and local ID jdoe at CarRentalInc
Federation with a transient pseudonym

Browser

Service Provider www.CarRentalInc.com

Identity Provider www.AirlineInc.com

Local ID IdP Linked ID
n/a AirlineInc 294723

Linked ID SP Local ID
294723 CarRentalInc john

Cache

Transient pseudonym (NameID= "294723") and attributes

Identity store

Access resource

Supply resource

Pass along signed <Response>

Convey <AuthnRequest> asking for transient pseudonym

Convey signed <Response> about 294723

User login as john

Challenge for credentials

User with local ID john at AirlineInc

Local ID     IdP       Linked ID
n/a      AirlineInc    294723

Linked ID       SP       Local ID
294723    CarRentalInc   john

Assertion Consumer Service

Single Sign-On Service

Access check
Federation termination

<table>
<thead>
<tr>
<th>Local ID</th>
<th>IdP</th>
<th>Linked ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>jdoe</td>
<td>AirlineInc</td>
<td>61611</td>
</tr>
<tr>
<td>jdoe</td>
<td>BankingInc</td>
<td>71711</td>
</tr>
<tr>
<td>mlamb</td>
<td>AirlineInc</td>
<td>81811</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linked ID</th>
<th>SP</th>
<th>Local ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>61611</td>
<td>CarRentalInc</td>
<td>john</td>
</tr>
<tr>
<td>61612</td>
<td>HotelStayInc</td>
<td>john</td>
</tr>
<tr>
<td>61621</td>
<td>CarRentalInc</td>
<td>mary</td>
</tr>
</tbody>
</table>

1. Terminate 61611 linking
2. Affirmative response
SP-initiated single logout

Service Provider
CarRental.com

Identity Provider
AirlineInc.com

User or UA action

Browser

Request global logout

Logged out

<LogoutRequest>

<LogoutResponse>

Logged out

SP-initiated single logout

Service Provider
CarRental.com

Identity Provider
AirlineInc.com

Single Logout Service

User or UA action

Browser

Request global logout

Logged out

<LogoutRequest>

<LogoutResponse>

Logged out
How you can get started
Development vs. deployment

• You shouldn't have to implement SAML support from scratch in applications
  – Open-source implementations for various languages and platforms
    • However, SAML V2.0 is still new in "roadmap" terms
    • OpenSAML is out in front on support
  – Free trials of products
    • E.g., Sun's Access Manager/Federation Manager are available for free through the "Red October" program

• Your lawyers and privacy advocates shouldn't have to start from scratch either
  – Use Liberty Alliance guidelines in building federation relationships
A real-world case study: Sun-BIPAC federation

- Sun provides an employee benefit: access to BIPAC, which provides insight on the U.S. political scene
  - BIPAC offers a web application for personalized information lookup by Congressional district
  - Some personalization is restricted by U.S. Law related to privacy

- Benefits of federation (N.B.: it uses Liberty ID-FF):
  - Cross-domain SSO from the Sun IdP to the BIPAC SP, in the course of which stronger authentication has been deployed
  - Privacy-enabled attribute exchange to allow anonymous – yet personalized – experiences

- See Enterprise Outsourcing paper for interesting deployment considerations and lessons
Resources
Some helpful resources

- SAML specs and outreach info:
  http://www.oasis-open.org/committees/security

- Liberty deployment guidelines:
  http://projectliberty.org/resources/guidelines.php

- SAML/Liberty Federation adoption info:
  http://projectliberty.org/about/marketadoption.php

- The IIW map in full resolution:

- Paper on Liberty Federation in Enterprise Outsourcing:
  http://www.idealliance.org/proceedings/xml05/abstracts/paper154.html

- Aggregation of many popular identity weblogs:
  http://www.planetidentity.org

- Some open-source projects involving SAML:
  http://ZXID.org
Any questions?

Thank you for your attention

Eve Maler
eve.maler@sun.com
Pushing String @ http://www.xmlgrrl.com/blog