T-110.5140 Network Application Frameworks and XML

Service Federation

22.03.2010

Sasu Tarkoma

Contents

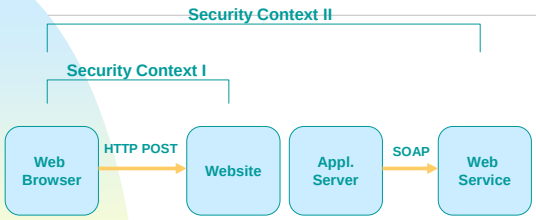
Security and Trust

- We are going towards identity-based service access
  - A number of identities per host
  - Pseudonyms, privacy issues
  - Delegation and federation are needed
- Decentralization: the user has the freedom of choosing who manages identity and data
- Solutions for authentication
  - Web-based standard (top-down)
    - ID-FF
  - Web-based practice (bottom-up)
    - OpenID and oAuth
  - Web services
    - SAML 2.0

Security Contexts in Web Services

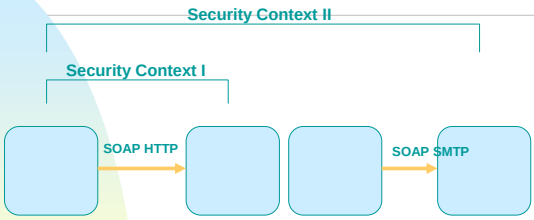
- Remember Web Services goals:
  - Re-use existing services
  - Combine services from several domains
- Security result: Must support several security domains
  - SOAP intermediaries
  - Reusing security tokens from one message in another message

Example 1: Pass subject details



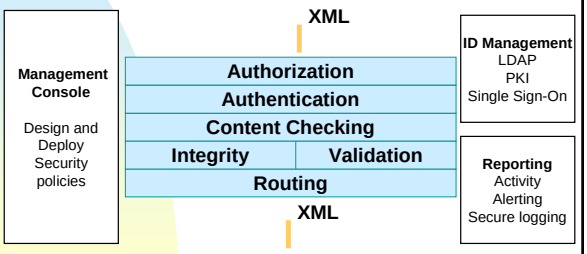
Main Point: We need security within AND between security contexts!

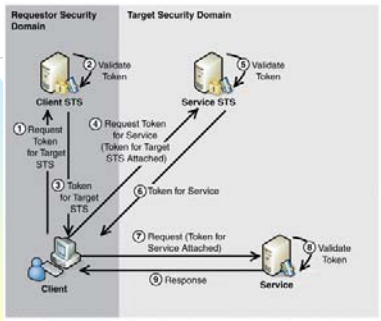
Example 2: SOAP Routing



Main Point: We need XML validation, encryption, and authentication between security contexts!

Functional point of view





Source: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpag2/html/wssp.asp

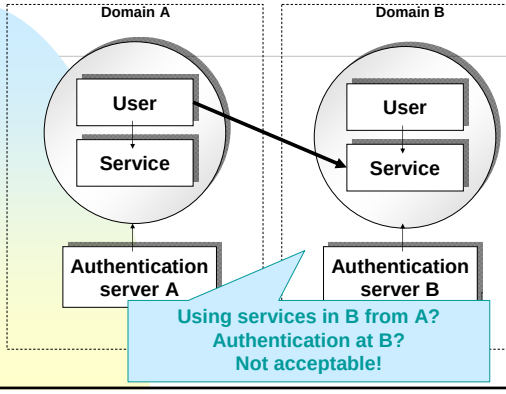
SAML

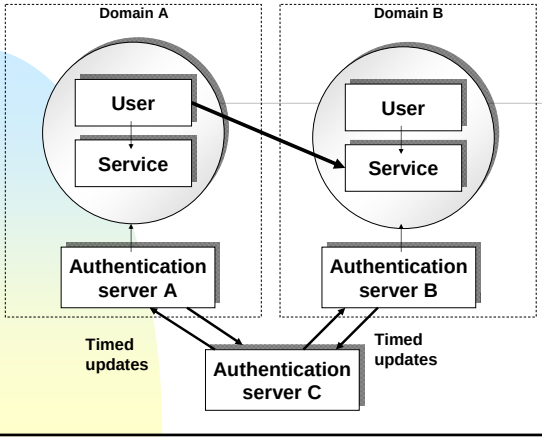
- SAML (Security Assertion Markup Language)
  - A XML-based framework (schemas) for the exchange of authentication and authorization information
  - A standard message exchange protocol
    - How you ask and receive information
- Mainly for integration, up to relying parties to decide to what authentication authority to trust
- Assertions can convey information about authentication acts performed by subjects, attributes of subjects, and authorization decisions about whether subjects are allowed to access certain resources
  - Authentication statements merely describe acts of authentication that happened previously
- Specified by OASIS

SAML in a nutshell

- XML-based framework for exchanging security information
  - XML-encoded security assertions
  - XML-encoded request/response protocol
  - Rules on using assertions with standard transport and messaging frameworks
- SAML & WS-Security allow a SOAP message to include information about the end-user's authentication status

SAML Motivation: Portable Trust





SAML assertions

- An assertion is a declaration of fact about a subject, e.g. a user
  - According to some assertion issues
- SAML has three kinds, all related to security:
  - Authentication
  - Attribute
  - Authorization decision
- You can extend SAML to make you own kinds of assertions
- Assertions can be digitally signed

All assertions have some common information

- Issuer and issuance timestamp
- Assertion ID
- Subject
  - Name plus the security domain
  - Optional subject information, 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

Authentication assertion

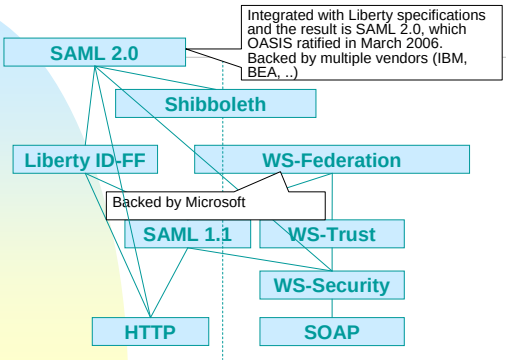
- An issuing authority asserts that:
  - Subject S
  - was authenticated by means M
  - at time T
- Caution: actually checking or revoking of credentials is not in the scope of SAML!
  - Password exchange
  - Challenge-response
  - Etc.
- It merely lets you link back to acts of authentication that took place previously

Example authentication assertion

```
<saml:Assertion
  MajorVersion="1" MinorVersion="0"
  AssertionID="127.0.0.1.1234567"
  Issuer="Example Corp"
  IssueInstant="2005-04-04T09:00:00Z">
  <saml:Conditions
    NotBefore="2005-04-04T09:00:00Z"
    NotAfter=""2005-04-04T09:05:00Z"/>
  <saml:AuthenticationStatement
     AuthenticationMethod="password"
     AuthenticationInstant="2005-04-04T09:01:00Z">
     <saml:Subject>
       <saml:NameIdentifier
         SecurityDomain="example.com"
         Name="johndoe"/>
       </saml:Subject>
   </saml:AuthenticationStatement>
</saml:Assertion>
```

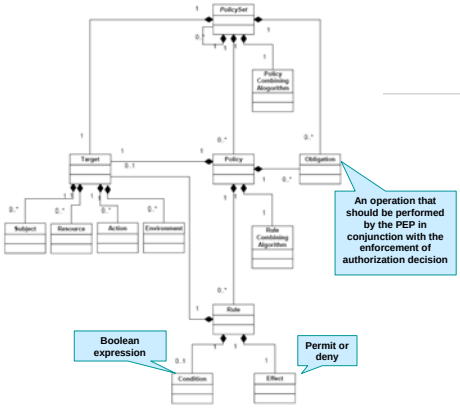
Assertion type	Description
Authentication Assertion	Asserts that subject S was authenticated by means M at time T
Attribute Assertion	Asserts that subject S is associated with attributes A1, A2,… with values V1,V2,...
Authorization Decision Assertion	Should the request to subject S for access type A be granted to resource R given evidence E

Overview of SSO Technologies

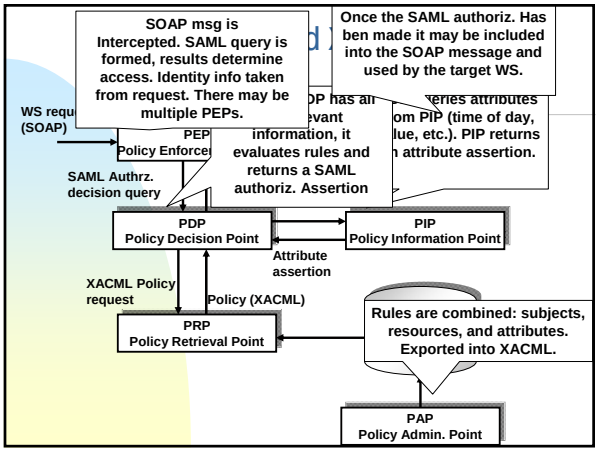


XACML

- XACML 2.0 and all the associated profiles were approved as OASIS Standards on 1 February 2005.
- XACML defines three top-level policy elements: <Rule>, <Policy> and <PolicySet>. The
  - <Rule> element contains a Boolean expression that can be evaluated in isolation, but that is not intended to be accessed in isolation by a ***PDP***. So, it is not intended to form the basis of an ***authorization decision*** by itself. It is intended to exist in isolation only within an XACML ***PAP***, where it may form the basic unit of management, and be re-used in multiple ***policies***.
  - The <Policy> element contains a set of <Rule> elements and a specified procedure for combining the results of their evaluation. It is the basic unit of ***policy*** used by the ***PDP***, and so it is intended to form the basis of an ***authorization decision***.
- Defines algorithms arriving at an authorization decision given the input rules and policies



Source: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf



Implementations

- Trust Services Integration Kit (TSIK), Verisign
  - Java API for creating trusted services, includes a SAML API
  - http://www.xmltrustcenter.org/developer/verisign/tsik/index.htm
- Apache XML-Security, Apache Software Foundation
  - XML Digital Signature and XML Encryption (Java, C++)
  - http://xml.apache.org/security/
- Web Services Enhancements 2.0, Microsoft
  - .NET implementation of various WS Security specs.
  - http://msdn.microsoft.com/webservices/building/wse/
- Microsoft Passport, Microsoft
  - Single sign-on support
- XML Security Suite, IBM
  - XML Digital Signature, XML Encryption and XML Access Control Language (Java)
  - http://www.alphaworks.ibm.com/tech/xmlsecuritysuite
- SunONE Identity Server, Sun Microsystems
  - Supports Liberty's federated identity and SAML

Web Services Enhancements 3.0

- Implements many of the rules of the WS-* specifications
- Works with HTTP and SOAP (SoapExtensions)
- Supported specifications
  - WS-Security, WS-SecurityPolicy, WS-SecureConversation, WS-Trust, WS-Referral, WS-Addressing, WS-Policy, WS-Attachments
  - 3.0 supports WS-Security 1.1
- Supports signing/encrypting message elements and policies
- Overview
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwse/html/newwse3.asp

WSE turnkey security profiles

- Common scenarios/patterns for securing messaging
  - UsernameOverTransport (username+pass&SSL)
  - UsernameForCertificate (username+pass&X.509 server auth)
  - AnonymousForCertificate(X.509 server auth)
  - MutualCertificate10 (X.509 client&server auth WS-S 1.0)
  - MutualCertificate11 (X.509 client&server auth WS-S 1.1)
  - Kerberos (Windows)
- Implemented using policy files
- Tokens and web farms
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/sctinfarm.asp

```
<mutualCertificate11Security
 clientActor
 establishSecurityContext="true|false"
 messageProtectionOrder="Signature and encryption order"
 renewExpiredSecurityContext="true|false"
 requireDerivedKeys="true|false"
 requireSignatureConfirmation="true|false"
 serviceActor
 ttlInSeconds >
 <clientToken/>
 <serviceToken/>
 <protection/>
</mutualCertificate11Security >
```

Note that both the client and server need to share part of the profile.

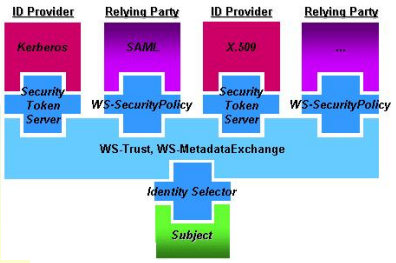
Passport and Live ID

- Intended to solve two problems
  - to be an identity provider to MSN
  - identity provider for the Internet
- First goal
  - over 250 million active Passport accounts and
  - 1 billion authentications per day
- Second goal
  - What is the role of the identity provider in transactions?
  - Passport no longer stores personal information other than username/password credentials
- Authentication service for sites
- Proprietary technology
- Roadmap: towards identity card (CardSpace)
  - Interface for identity based authentication and authorization
  - Identity cards that people can choose (Identity Metasystem)
  - Integration with Web sites
  - Consistent user interface
- Windows Live ID
  - Unified login service for Microsoft sites such as Hotmail, MSNBC, MSN, ..
  - Used also for ad targeting with adCenter
  - Has been opened for Web site developers (August, 2007)

Identities

- CardSpace (Microsoft)
  - Multiple identities
  - Interface for identity based authentication and authorization
  - Identity cards that people can choose
  - Integration with Web sites
  - Consistent user interface
  - Microsoft plans to implement this
    - ActiveX, WS-*
- http://www.identityblog.com/

IdentityCard



Source: http://www.identityblog.com/

Liberty Alliance ID-FF

- Liberty Alliance Identity Federation Framework (ID-FF)
- Basic case: Web direction
  - Redirect to IDP for credentials, redirect back to service, verification with IDP
  - Uses SAML requests and assertions
- Mandatory features for an identity provider
  - Single sign on and federation
  - Single sign out
  - Federation termination
  - Affliliations
  - Dynamic proxying of Identity Providers
- Circle of trust implemented using
  - SAML assertions, requests, redirection, and validation

ID-FF specs

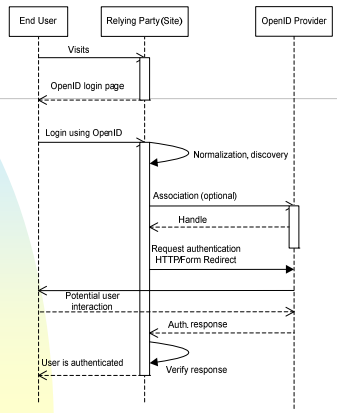
- Liberty ID-FF
  - Identity Federation Framework
  - A forerunner to the SAML 2.0 specification. All of the functionality in ID-FF has been incorporated into SAML 2.0
- Liberty ID-WSF
  - Identity Web Services Framework
  - Builds on WS-Security and SAML 2.0
- Liberty ID-SIS
  - Identity Services Interface Specifications
  - High-level web service interfaces that support particular use cases like data/profile, geolocation, contact book, and presence services.

OpenID

- OpenID is a decentralized sign-on system for the Web
  - Not a real single sign-on solution, does not support authorization
- Instead of usernames and passwords, users need to have an account with some identity provider
- The user has the choice of selecting a suitable identity provider
- Support: AOL, Orange, FireFox, Microsoft planning support in Vista, LiveJournal, Wikitravel, Zooomr, Ma.gnolia
- Estimated 120 million OpenIDs on the Internet
- OpenID 2.0 supports discovery
  - Yadis provides a mechanism for determining the services that are available with a given identifier
- Identity aggregation: ClaimID
  - Claim Web resources under your OpenID (must have write permission)

OpenID URL

- There are two ways to obtain an OpenID-enabled URL that can be used to login on all OpenID-enabled websites.
  - To use an existing URL that one's own control (such as one's blog or home page), and if one knows how to edit HTML, one can insert the appropriate OpenID tags in the HTML code following instructions at the OpenID specification.
  - The second option is to register an OpenID identifier with an identity provider. They offer the ability to register a URL (typically a third-level domain) that will automatically be configured with OpenID authentication service.

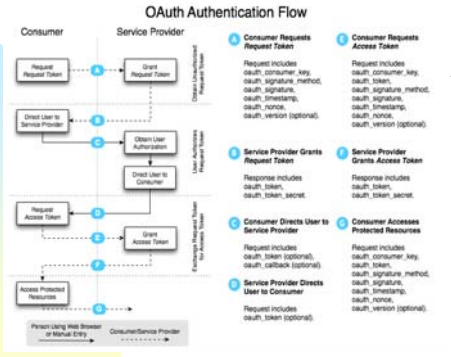


oAuth

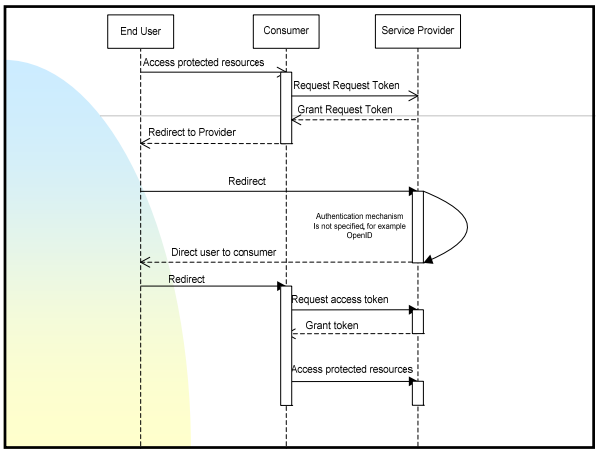
- oAuth is an open protocol to allow clients to access protected data
- Intended for desktop and web applications
- Example: a printing service printer.example.com, oAuth provides mechanisms for the printer to access user photos on photos.org without requiring users to provide credentials to printer.example.com.
- A solution for publish and interact with protected data
- Does not require a specific user interface or pattern, nor does it specify how service providers authenticate users
  - Can be used with OpenID
- Attempt to collect best practices from existing protocols
  - BBAuth (Yahoo), FacebookAuth, FlickrAuth, AuthSub (Google), OpenAuth (AOL) ..
- Contributors from many Web companies: Google, Flickr, Ma.gnolia, sixapart, Jaiku
- oAuth 1.0 Draft 3 was released September 28, 2007
- More information: http://oauth.net

Authentication with oAuth

- Entities: User, Consumer (accessing data), Service Provider (keeps the data)
- Tokens:
  - Request token: used by the consumer to ask the user to authorize access
  - Access token: used by the consumer to access the protected resources on behalf of the user
- OAuth Authentication is done in three steps:
  - The Consumer obtains an unauthorized Request Token.
  - The User authorizes the Request Token.
  - The Consumer exchanges the Request Token for an Access Token.



Source: http://oauth.googlecode.com/svn/spec/branches/1.0/drafts/3/spec.html



Lecture Summary

- Security contexts
  - Security needed within and between contexts
  - **XML validation, encryption, and authentication needed between security contexts!**
- WS security standard revisited
  - SOAP header carries security information (and other info as well)
  - Selective processing
- SAML
  - Statements about authorization, authentication, attributes
  - SAML & WS-Security & XACML
- OpenID and Live ID
- Implementations available