Introducing Shibboleth

Clarin AAI Hands On Workshop
Oxford eResearch Center
2009-02-25
Federated Authentication & Authorization Middleware in a Distributed Environment

Components & Entities

How it works together, Flow Chart

Security Assertion Markup Language (SAML)

Metadata

Security: SSL & Certificates

Shibboleth 2.x vs. 1.3.x
What is Shibboleth for?

Authentication
- federated Single Sign On (web browser)
- using local authentication system

Authorization
- attribute based access control

in a distributed environment
- decoupled service protection from identity management
- no domain- or security-realm boundaries
  (“Internet Technologie”)
- sophisticated abstraction of use-cases
  to support several communication profiles
  for exchanging security information
- based on Security Assertion Markup Language (SAML)

Encryption & Signing by x.509 certificates
Components

Service Provider (SP)
  ensure login
  request attributes
  access control

Identity Provider (IdP)
  authentication & SSO
  provide attributes

Discovery Service (DS)
  resolve users home site – their hosting Identity Provider

Attributes, SAML & Assertions
  exchange security information

Federation through Metadata
  declare involved entities and communication endpoints
Flow Chart (simple)
How it works...

User requests a protected resource from a Service Provider

Service Provider
- selects session initialization based on protection configuration
  - decides which IdP or DS to use
  - decides which communication profile to use
  - redirects user to destined IdP or DS, respectively

Discovery Service
- user selects his “home site”, his hosting IdP
- DS sets a cookie ('_saml_idp') with chosen IdP information
- and directs the requesting SP to the destined IdP (SSO Service) using redirects and said cookie
Identity Provider

SSO-Service checks requests and in case redirects to configured LoginHandler
SSO-Service authenticates user using existing IDM at the hosting site
Attribute-Authority resolves attributes filters attribute according to release policy
posts SAML assertions with authentication- & attribute-statement via an auto-submit post form to requesting SP

Service Provider

processes the posted SAML assertions retrieves attributes, mapping to httpd variables, filtering due to policy
redirect to originally requested resource
access control decision (mod_shib) based on access-rules & provided attributes
<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shib-Application-ID</td>
<td>Default</td>
</tr>
<tr>
<td>Shib-Session-ID</td>
<td>cdc5233802465379c3e216e874bcb569</td>
</tr>
<tr>
<td>Shib-Identity-Provider</td>
<td>idp2_rzg.mpg.de</td>
</tr>
<tr>
<td>Shib-Authentication-Instant</td>
<td>2009-02-19T09:18:59.581Z</td>
</tr>
<tr>
<td>epAffiliation</td>
<td>• staff</td>
</tr>
<tr>
<td></td>
<td>• member</td>
</tr>
<tr>
<td>epEntitlement</td>
<td>• urn:geant:dfn.de:mpg:aoi:test:costcenter:rzg</td>
</tr>
<tr>
<td></td>
<td>• urn:geant:dfn.de:dfn-pki:slcs</td>
</tr>
<tr>
<td>epOrgDN</td>
<td>O=Max Planck Society,DC=mpg,DC=de</td>
</tr>
<tr>
<td>epOrgUnitDN</td>
<td>OU=Computing Center Garching, O=Max Planck Society, DC=rzg, DC=mpg, DC=de</td>
</tr>
<tr>
<td>app</td>
<td><a href="mailto:megger@rzg.mpg.de">megger@rzg.mpg.de</a></td>
</tr>
<tr>
<td>epsAffiliation</td>
<td>• <a href="mailto:staff@rzg.mpg.de">staff@rzg.mpg.de</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="mailto:member@rzg.mpg.de">member@rzg.mpg.de</a></td>
</tr>
<tr>
<td>mail</td>
<td><a href="mailto:matthias.egger@rzg.mpg.de">matthias.egger@rzg.mpg.de</a></td>
</tr>
<tr>
<td>o</td>
<td>• MPG</td>
</tr>
<tr>
<td></td>
<td>• Max Planck Gesellschaft zur Förderung der Wissenschaften</td>
</tr>
<tr>
<td></td>
<td>• Max Planck Gesellschaft for the Advancement of Science</td>
</tr>
<tr>
<td>ou</td>
<td>• rzg.mpg.de</td>
</tr>
<tr>
<td></td>
<td>• Rechenzentrum Garching</td>
</tr>
<tr>
<td></td>
<td>• Computing Center Garching</td>
</tr>
<tr>
<td>persistentID</td>
<td>idp2_rzg.mpg.de</td>
</tr>
<tr>
<td>sn</td>
<td>Egger</td>
</tr>
<tr>
<td>HTTP_SHIB_SESSION_ID</td>
<td>cdc5233802465379c3e216e874bcb569</td>
</tr>
<tr>
<td>HTTP_SHIB_IDENTITY_PROVIDER</td>
<td>idp2_rzg.mpg.de</td>
</tr>
<tr>
<td>HTTP_SHIB_AUTHENTICATION_METHOD</td>
<td></td>
</tr>
<tr>
<td>HTTP_SHIB_AUTHENTICATION_INSTANT</td>
<td>2009-02-19T09:18:59.581Z</td>
</tr>
<tr>
<td>HTTP_SHIB_AUTHNCONTEXT_CLASS</td>
<td></td>
</tr>
<tr>
<td>HTTP_SHIB_AUTHNCONTEXTDECL</td>
<td>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</td>
</tr>
<tr>
<td>HTTP_SHIB_ASSERTION_COUNT</td>
<td></td>
</tr>
<tr>
<td>HTTP_SHIB_APPLICATION_ID</td>
<td>Default</td>
</tr>
</tbody>
</table>

Show Shibboleth assertions | Show all HTTP variables

provided by SWITCHaai
Attributes

basically simple string key-values pairs

distributed architecture requires naming convention for exchanged attributes

URIs as attribute names for uniqueness and namespace control

leverage existing LDAP object class identifier

\texttt{urn:oid:}

\begin{verbatim}
  e.g. "urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
\end{verbatim}

use/register your own urn namespace

\texttt{urn:mace:, urn:geant:}

\begin{verbatim}
  e.g. "urn:mace:dir:attribute-def:eduPersonEntitlement"
\end{verbatim}

create own URLs

\begin{verbatim}
  e.g. "https://myfederation.org/attributes/myOwnAttbName"
\end{verbatim}
Security Assertion Markup Language

XML standard by OASIS
for exchange of authentication & authorization information

Short History
- SAML 1.0 as OASIS Standard in November 2002
- SAML 1.1 as OASIS Standard in September 2003
- SAML 2.0 as OASIS Standard in March 2005

SAML and Shibboleth
- SAML 1.1 implemented in Shibboleth 1.x
- SAML 2.0 implemented in Shibboleth 2.x (rel. March 2008)

Compatibility
- SAML 2.0 substantially different to SAML 1.1
  https://spaces.internet2.edu/display/SHIB/SAMLDiff
- Shibboleth 2.x is fully downward compatible
to Shibboleth 1.3.x / SAML 1.1
SAML Components Stack

Profiles
“use cases” - specify how assertions, protocols & bindings combine to handle concrete use cases

Bindings
“how” - map SAML protocols to actual message/communication protocols

Protocols
“what” - specify elements of SAML assertion requests/responses

Assertions
“bare xml” - package of security information
Federation & Metadata

Federation

- through metadata
- requires schema definition (policy)
- requires Discovery Service

Metadata

- define participating entities
- define communication endpoints, SAML profile bindings
- certificates/public keys for encryption & signing
Identity Provider MD

EntityID
IDPSSODescriptor
  ArtifactResolutionService
  NameIDFormat
  SingleSignOnService
  KeyDescriptor, Certificate
AttributeAuthorityDescriptor
  AttributeService
  KeyDescriptor, Certificate

Service Provider MD

EntityID
SPSSODescriptor
  AssertionConsumerService
  ManageNameIDService
  SingleLogOut
  KeyDescriptor, Certificate
  Extension: DiscoveryResponse

Sample Metadata
Shibboleth 2

Shibboleth 2.0 released March 2008
uses SAML 2.0, compatible to SAML 1.1
fully downward compatible to 1.3.x
using attribute-push model
usage of LDAP object class identifier (urn:oid:" namespace)
as (default) attributes names
Shibboleth 1.3.x announced for end-of-life
by June 30, 2010
Thanks & Discussion

- Thank You for Your Attention -

Questions, Ideas, Discussion
mpgaai workshop

Shibboleth Service Provider

Clarin AAI Hands On Workshop
Oxford eResearch Center
2009-02-25
About Shibboleth ServiceProvider

Shibboleth ServiceProvider & Components
Installation & Initial Setup
Join Federations (Session Initialization)
Using Attributes (Mapping & Policy)
Protect Web Locations
Single Logout (SLO)
Other Applications & Services
Shibboleth & Grid
Service Provider: Components

Session Initiator
- checks authentication status
- establishes (SSO) session with IdentityProvider, in case via DiscoveryService

Assertion Consumer Service (ACS)
- controls SAML Assertions
- requests attribute from IdentityProvider (pull model)
- validating the attributes based on Attribute-Acceptance-Policy

Resource Manager (RM)
- actual access control based on attributes
Installation Layout

Shibboleth Service Provider
- apache22.config
- mod_shib_22.so
- shibboleth2.xml

Apache httpd
- httpd.conf
- mod_ssl
- certificate (public/private keys)

log4shib
xerxes-c
xml-security-c
openSaml
xmlTooling

metadata.xml

g++
libcurl-dev
libssl-dev

openssl

load
include
ref.
Apache - virtual ssl host

httpd-vhost.conf

...  
Listen 443
<VirtualHost _default_:443>
  Include /opt/mpgaai/shibboleth-sp2/etc/shibboleth/apache22.config

  ServerName https://idp.mydomain.de
  UseCanonicalName On

  SSLEngine on
  SSLCipherSuite ALL
  SSLVerifyDepth 10
  SSLOptions +StdEnvVars +ExportCertData

  SSLCertificateFile /opt/mpgaai/httpd/conf/ssl.crt/sp.crt
  SSLCertificateKeyFile /opt/mpgaai/httpd/conf/ssl.key/sp.key
  SSLCACertificatePath /etc/ssl/certs
...

Listen 443
<VirtualHost _default_:443>
  SSLEngine on
  #...
  # sample virtual ssl proxy configuration
  SSLProxyEngine on
  RewriteEngine On
  RewriteOptions Inherit
  RewriteRule ^/sp/(.*) https://internalserver:443/sp/$1 [P]
  SetEnv force-proxy-request-1.0 1
  SetEnv proxy-nokeepalive 1
  ...
</VirtualHost>
Installation HowTos 4 Shibboleth-SP

Internet2
https://spaces.internet2.edu/display/SHIB2/Installation

SWITCH
https://www.switch.ch/aai/support/serviceproviders/

DFN (de)
https://www.aai.dfn.de/dokumentation/
  service-provider/installation-opensuse-10-3/

LRZ
http://www.lrz-muenchen.de/~hommel/
  shibboleth/sp20_unter_sles10sp1.html
Shibboleth SP – Main Config Files

shibboleth2.xml
core configuration of Shibboleth

attribute-map.xml
  definition of applicable attributes
mapping of attributes to httpd-environment- and (optional) http-header-variables

attribute-policy.xml
filtering of accepted attributes:
checks for attribute names and values

metadata.xml
describes the federation (participating entities)
Initial Setup

shibboleth2.xml

Windows Installation: set IIS Site
https://spaces.internet2.edu/display/SHIB2/NativeSPISAPI

RequestMap/Host.name
ApplicationDefaults.entityID, (~.homeUrl)
Sessions.handlerURL
SessionInitiators
MetadataProvider
Credentials
SAML profile bindings

Metadata Generator
shibboleth2.xml – SessionInitiator

https://spaces.internet2.edu/display/SHIB2/NativeSPSessionInitiator

```xml
<SessionInitiator
    id="mpgaai" type="Chaining"
    Location="/Login" isDefault="true" relayState="cookie" >

    <SessionInitiator type="SAMLDS"
        URL="https://shib-idp.gwdg.de/ds/WAYF" />

    <SessionInitiator type="WAYF"
        URL="https://shib-idp.mpg.de/ds/WAYF"
        defaultACSIIndex="5" />

    <SessionInitiator type="SAML2"
        defaultACSIIndex="1" acsByIndex="false"
        template="bindingTemplate.html" />

    <SessionInitiator type="Shib1" defaultACSIIndex="5" />

</SessionInitiator>
```

SAML2 - DiscoveryService
SAML/Shib1 - Where-R-U-From
SAML2 - Session
SAML/Shib1 - Session
Multiple Federations / IdPs

```
<SessionInitiator id="mpgaai" type="Chaining"
    Location="/Login" isDefault="true" relayState="cookie" >
    <SessionInitiator type="SAMLDS" ...
</SessionInitiator>

<SessionInitiator id="deisa" type="Chaining"
    Location="/DeisaFed" isDefault="false" relayState="cookie" >
    <SessionInitiator type="SAMLDS" URL=...
</SessionInitiator>

<SessionInitiator id="idpBoth" type="Chaining"
    Location="/TestIDP" isDefault="false"
    entityID="testidp2_rzg_mpg_de" >
    <SessionInitiator type="SAML2" template="bindingTemplate.html"
        defaultACSIndex="1" acsByIndex="false" />
    <SessionInitiator type="Shib1" defaultACSIndex="5" />
</SessionInitiator>

<SessionInitiator id="idpSaml2" type="SAML2"
    Location="/SampleIDP" isDefault="false"
    entityID="testidp2_rzg_mpg_de"
    defaultACSIndex="1" acsByIndex="false"
    template="bindingTemplate.html" />
```
Using Attributes - Mapping

attribute-map.xml

https://spaces.internet2.edu/display/SHIB2/NativeSPAddAttribute
https://spaces.internet.internet2.edu/display/SHIB2/NativeSPAttributeDecoder

maps attribute name (URN)
to attribute ID: httpd-environment-, http-header-variable

```
<Attribute id="epEntitlement"
    name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" />

<Attribute id="epAffiliation"
    name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" >
    <AttributeDecoder
        xsi:type="StringAttributeDecoder"
        caseSensitive="false"/>
</Attribute>

<Attribute id="epScopedAffiliation"
    name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9">
    <AttributeDecoder
        xsi:type="ScopedAttributeDecoder"
        caseSensitive="false"/>
</Attribute>
```
attribute-policy.xml
filters attributes due to configured attribute acceptance policy

AttributeFilterPolicy
defines several policies according to your requirements
https://spaces.internet2.edu/display/SHIB2/AFPAttributeFilterPolicy

PolicyRequirementRule
specifies when an AttributeFilterPolicy applies
https://spaces.internet2.edu/display/SHIB2/AFPPolicyRequirementRule

AttributeRule
specifies the rules for each attribute
https://spaces.internet2.edu/display/SHIB2/AFPAttributeRule

PermitValueRule
defines which values of (which) attributes are accepted
https://spaces.internet2.edu/display/SHIB2/AFPPermitValueRule
<afp:AttributeFilterPolicy>
  <afp:PolicyRequirementRule xsi:type="ANY"/>

  <afp:AttributeRule attributeID="email">
    <afp:PermitValueRule xsi:type="basic:ANY"/>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="epScopedAffiliation">
    <afp:PermitValueRule xsi:type="AND">
      <RuleReference ref="epAffiliationValues"/>
      <RuleReference ref="ScopingRules"/>
    </afp:PermitValueRule>
  </afp:AttributeRule>

  <afp:AttributeRule attributeID="*">
    <afp:PermitValueRule xsi:type="ANY"/>
  </afp:AttributeRule>

</afp:AttributeFilterPolicy>
Attribute Policy – shared rules

<afp:PermitValueRule id="epAffiliationValues" xsi:type="OR">
  <Rule xsi:type="AttributeValueString" value="faculty"/>
  <Rule xsi:type="AttributeValueString" value="student"/>
  <Rule xsi:type="AttributeValueString" value="staff"/>
  <Rule xsi:type="AttributeValueString" value="alum"/>
  <Rule xsi:type="AttributeValueString" value="member"/>
  <Rule xsi:type="AttributeValueString" value="affiliate"/>
  <Rule xsi:type="AttributeValueString" value="employee"/>
  <Rule xsi:type="AttributeValueString" value="library-walk-in"/>
</afp:PermitValueRule>

<afp:PermitValueRule id="ScopingRules" xsi:type="AND">
  <Rule xsi:type="NOT">
    <Rule xsi:type="AttributeValueRegex" regex="@"/>
  </Rule>
  <Rule xsi:type="saml:AttributeScopeMatchesShibMDScope"
       xmlns:saml="urn:mace:shibboleth:2.0:afp:mf:saml"/>
</afp:PermitValueRule>
httpd.conf

```xml
<VirtualHost _default_:443>
   Include /opt/mpgaai/shibboleth-sp2/etc/shibboleth/apache22.config
   ...

   # protected sites:  
   # sample to require authenticated/logged-in user 
   <Location /secure>
      AuthType shibboleth
      ShibRequireSession On
      ShibUseHeaders on
      require valid-user
   </Location>

   # sample to require certain attribute with certain value 
   <Location /libonly>
      AuthType shibboleth
      ShibRequireSession On
      require entitlement common-lib-terms
   </Location>

</VirtualHost>
```

anything under /secure is shibboleth protected and requires an authenticated user

only users with attribute 'entitlement' of value 'common-lib-terms' are authorized
<VirtualHost _default_:443>
    Include /opt/mpgaai/shibboleth-sp2/etc/shibboleth/apache22.config
...
<Location /secure>
    AuthType shibboleth
    ShibRequireSession On
    require valid-user
</Location>

# sample of free sub locations
<Location /secure/free>
    Order Deny,Allow
    Satisfy Any
    Allow from all
</Location>

# lazy session
# e.g. for mediawiki with shib-plugin
<Location /mediawiki>
    AuthType shibboleth
    Require shibboleth
</Location>

</VirtualHost>

all locations under /secure/free are unprotected although /secure/* is generally shibboleth protected

"lazy session" no login required, shibboleth attributes are present, though
Internet2:
https://spaces.internet2.edu/display/SHIB2/SLOIssues

often requested feature, realization not trivial, though
implemented in Service-Provider 2.x

not yet implemented in Identity-Provider 2.1.x
on the roadmap for IdP-2.2

Challenge
- proper feedback (success/fail) of all affected entities
- how to treat services (SP) temporary not available
- user comprehension: users should understand SSO & SLO

Own Solutions: e.g. Haka Federation (Finland)
http://www.csc.fi/english/institutions/haka/technology/information/haka_logout
Shibboleth ServiceProvider for Apache Webserver only

webapplication may make use of shibboleth attributes from the httpd header

OIOSAML.java
http://www.softwareborsen.dk/projekter/softwarecenter/
  brugerstyring/oio-saml-java

mainly a servlet filter for protecting webapps
uses opensaml2 implementation

plugins & extensions to integrate Shibboleth or SAML

Zope
https://mams.melcoe.mq.edu.au/zope/mams/pubs/
  Installation/shibbolized-zope

Plone http://tid.ithaka.org/software

Mediawiki http://www.mediawiki.org/wiki/Extension:Shibboleth_Authentication
common grid middleware like Unicore or Globus Toolkit are based on x.509 certificates

GridShib project integrates Globus Toolkit with Shibboleth-1.3 / SAML-1.1

Shibboleth-enabled Short Lived Credentials Service – SLCS

VOMS - Virtual Organization Membership Service
with VASH - VOMS Attributes from Shibboleth
http://www.switch.ch/grid/vash/about/about_long.html
short lived credential service (slcs)

1. login

2. authenticate

3. handle
   (Browser POST)

4. SAML attributes

5. DN + authZ token

6. CSR

7. PKCS#10

8. PKCS#7

9. Certificate

10. store

Key + X509

Client

Shibboleth IdP

Online CA

SLCS
Shibboleth SP

audit

Data store

clarin aai hands on workshop
- Thank You for Your Attention -

Questions, Ideas, Discussion