Introducing Shibboleth

MPG-AAI Workshop
2009-07-09
MPG-AAI

- MPG-AAI
  - a MPG-wide Authentication & Authorization Infrastructure
  - for access control to web-based resources
  - using Shibboleth Framework
  - plus own extensions

- What is it about

- Why MPG-AAI

- What is Shibboleth

- How does it work?
Authentication is about

- *Identity* - “who am I”
- and *Trust* - how can I prove it

“Real World” Analogies

- electronic ID card - your passport
- prove it's yours - photo, biometric data...
- trust the issuer
Authorization is about

- **Roles** - “What” am I
- and **Rights** - what am I *entitled to do* therefore
- and **Trust** again – who says so

“Real World” Analogies: Going to the Movies

- your Role: moviegoer
- your Rights: watch the film
- proved by
  - your obtained ticket
  - issued by the cinema box office
IP (-Range) based “Authorization”

- IP based means “Where are you from”
- weak compensation for the central questions
  - who are you & what are you allowed to do
- leads to jungle in network structure
  - Uni Groups at MPG Campus or vice versa
  - Spin-Off Companies on MPG Campus
    (while most contracts are for non-commercial use only)
- Subnets must be segregated
  - changes are not notified
  - overlaps not reflected
  - hard to maintain
  - error prone
Certificates

- reliable, very good concerning Trust, but
- hard to obtain (require PKI, show your ID)
- mainly intended for authentication
- relatively static (changes in user roles require a new certificate)
- Usability many users can't cope with certificates
Access Control at Service Level

- Allows fine grained access control based on Roles & Rights
- No (false) assumptions about locality of users
- Flexible and good to maintain (direct control)

Problems
- (too) huge user space
- each application has its own Identity Management (IDM)
- Central Identity Management
  - single point of failure
  - hard to maintain (changes not notified etc.)

Solution: Federated Approach - MPG-AAI with Shibboleth
A ‘shibboleth’ is a custom, phrase or use of language that acts as a test of belonging to a particular social group or class. By definition, it is used to exclude those deemed unsuitable to join this group.

- Control by Division and Segregation.

Doris Salcedo, "Shibboleth", 2007, Installation at Tate Modern, London (Foto: Tate Modern)
What is Shibboleth for?

- **Authentication**
  - federated Single Sign On (web browser)
  - exploiting existing 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 Technology")
- **based on Security Assertion Markup Language (SAML)**
  - sophisticated abstraction of use-cases to support several communication profiles for exchanging security information
- **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)
Flow Chart

MPG-AAI - Introduction
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
...how it works

- **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>_dc5233802465379cee216e874bcb569</td>
</tr>
<tr>
<td>Shib-Identity-Provider</td>
<td>idp2_rzg_mpg.de</td>
</tr>
<tr>
<td>Shib-Authentication-Instant</td>
<td>2009-02-19T09:18:59.5817</td>
</tr>
<tr>
<td>Shib-AuthnContext-DecId</td>
<td>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</td>
</tr>
<tr>
<td>epAffiliation</td>
<td>• staff • member</td>
</tr>
<tr>
<td>epEntitlement</td>
<td>• urn:geant:dfn.de:mpg:am1:tcsttc:cottcc:rzg</td>
</tr>
<tr>
<td></td>
<td>• urn:geant:dfn.de:dfn-pki:slc</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>eppn</td>
<td><a href="mailto:megger@rzg.mpg.de">megger@rzg.mpg.de</a></td>
</tr>
<tr>
<td>eppAffiliation</td>
<td>• <a href="mailto:staff@rzg.mpg.de">staff@rzg.mpg.de</a> • <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 • Max Planck Gesellschaft zur Förderung der Wissenschaften</td>
</tr>
<tr>
<td></td>
<td>• Max Planck Society for the Advancement of Science</td>
</tr>
<tr>
<td>ou</td>
<td>• rzg.mpg.de • Rechenzentrum Garching</td>
</tr>
<tr>
<td></td>
<td>• Computing Center Garching</td>
</tr>
<tr>
<td>persistentID</td>
<td>idp2_rzg_mpg_de_idsp2_rzg_mpg_de\nKmyVoXx5Ev0n1073NcN\ldir\hd</td>
</tr>
<tr>
<td>sn</td>
<td>Egger</td>
</tr>
<tr>
<td>HTTP_SSH_SESSION_ID</td>
<td>_dc5233802465379cee216e874bcb569</td>
</tr>
<tr>
<td>HTTP_SSH_IDENTITY_PROVIDER</td>
<td>idp2_rzg_mpg_de</td>
</tr>
<tr>
<td>HTTP_SSH_AUTHENTICATION_METHOD</td>
<td>idp2_rzg_mpg_de</td>
</tr>
<tr>
<td>HTTP_SSH_AUTHENTICATION_INSTANT</td>
<td>2009-02-19T09:18:59.5817</td>
</tr>
<tr>
<td>HTTP_SSH_AUTHNCONTEXT_CLASS</td>
<td>idp2_rzg_mpg_de</td>
</tr>
<tr>
<td>HTTP_SSH_AUTHNCONTEXT_DECL</td>
<td>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</td>
</tr>
<tr>
<td>HTTP_SSH_ASSERTION_COUNT</td>
<td>0</td>
</tr>
<tr>
<td>HTTP_SSH_APPLICATION_ID</td>
<td>Default</td>
</tr>
</tbody>
</table>
The MPG-AAI Federation
IdP-Proxy

- Connects MPG-AAI Federation with other Federations like DFN-AAI
- Proxy mechanism allows "delegated" authentication to
  - other IdPs
  - other Identity-Management-Services (IDM) like LDAP, databases etc...
- all retrieved SAML assertions are signed by the IdP(-Proxy) and provided to the requesting SP
- other Federations or external SPs need to maintain just this IdP as single relying-party
- Attributes might be further filtered or adapted by the IdP-proxy at the crossover to other federations (DFN-AAI)
IP/Web-Proxy

- Bridge between the MPG-AAI Federation and external Service Providers
  - who don't have a Shibboleth interface
  - and who's authorization is based on IP-addresses/ranges.
- maps users' institute to specific IP-address
- solves “traveling scientist” problem
  enables users to access such IP-protected service providers from remote locations (field-trip, from home, ...)

![Diagram of IP/Web-Proxy with nodes labeled MP1, IP, MPI, ezproxy, and squid, connecting through publishers with IP-authZ and IP authZ]
Central Services Layout

Central (virtual) IP as access point for Users & Services
active
pound-loadbalancer listens on this virtual IP
Failover:
if "active" pound fails "passive" one takes over (managed by keepalived)

Loadbalancer passes load/requests to actual services, using encrypted/secure communication-channel

High-Availability Loadbalancer
Layer-7 session-aware

GWDG

services.aai.mpg.de (ports 80, 443)
pub.aai.mpg.de (ports 2048, 2443)

aai.gwdg.de
pound (active)
keepalived

aai2.gwdg.de
pound (passive)

RZG

aai.rzg.mpg.de
DiscoveryService
aai.rzg.mpg.de/ds
IdP-Proxy
aai.rzg.mpg.de/idp
IP-Proxy
(port 2048, 2443)
ezproxy
squid

GWDG

aai.gwdg.de
DiscoveryService
aai.gwdg.de/ds
IdP-Proxy
aai.gwdg.de/idp
IP-Proxy
(port 2048, 2443)
ezproxy
squid

aai2.gwdg.de
DiscoveryService
aai2.gwdg.de/ds
IdP-Proxy
aai2.gwdg.de/idp
IP-Proxy
(port 2048, 2443)
ezproxy
squid

SSH-Tunnel
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
  urn:oid:
  e.g. "urn:oid:1.3.6.1.4.1.5923.1.1.1.7"

- use/register your own urn namespace
  urn:mace:, urn:geant:
  e.g. "urn:mace:dir:attribute-def:eduPersonEntitlement"

- create own URLs
  e.g. "https://myfederation.org/attributes/myOwnAttbName"
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/SAMLDiffs
  - 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 - request/response pairs

Assertions
“bare xml” - package of security information

- IdP Discovery, Web-Browser SSO, SLO ...
- Http-POST, SAML SOAP ...
- Authn.Request, Assertion Query ...
- Issuer, Subject ...
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
### Metadata

<table>
<thead>
<tr>
<th>Identity Provider MD</th>
<th>Service Provider MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>• EntityID</td>
<td>• EntityID</td>
</tr>
<tr>
<td>• IDPSSODescriptor</td>
<td>• SPSSODescriptor</td>
</tr>
<tr>
<td>- ArtifactResolutionService</td>
<td>- AssertionConsumerService</td>
</tr>
<tr>
<td>- NameIDFormat</td>
<td>- ManageNameIDService</td>
</tr>
<tr>
<td>- SingleSignOnService</td>
<td>- SingleLogOut</td>
</tr>
<tr>
<td>- KeyDescriptor, Certificate</td>
<td>- KeyDescriptor, Certificate</td>
</tr>
<tr>
<td>• AttributeAuthorityDescriptor</td>
<td>- Extension: DiscoveryResponse</td>
</tr>
<tr>
<td>- AttributeService</td>
<td></td>
</tr>
<tr>
<td>- KeyDescriptor, Certificate</td>
<td></td>
</tr>
</tbody>
</table>

**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
- Thank You for Your Attention -

Questions, Ideas, Discussion