Delegated Authentication with Shibboleth

Andrew Petro
Software Developer
Unicon, Inc.

Fall 2010 Internet2 Membership Meeting
Atlanta, GA
03 November 2010
This Talk

● Delegated authentication using SAML and Shibboleth
  - Why you need it
  - What it is
  - Software for implementing it
Goals for this talk

- Understand delegation use case in abstract
- See why this is important in enterprise portals
- Understand that standard support and implementation of relevant features are in Shibboleth IdP and SP today
- Awareness of Java library and example code making use of this
Agenda

1. Introduction
2. Use Case
3. How It Works
4. Software
5. Next Steps
Introduction
Introduction

• Andrew Petro; Software Developer; Unicon, Inc.
  - Jasig CAS steering committee
  - Jasig uPortal committer
  - etc.
My employer: Unicon

- Jasig CAS Solutions Provider
- InCommon Affiliate

www.unicon.net
Use Case
Delegated Authentication

• System B authenticates to System C on behalf of Person A

• That is, A authenticates to B and delegates authority to B for the purpose of authenticating to C as “B on behalf of A”
Delegation example

uPortal

Email Preview Portlet

IMAP Server
Delegation example *

* Warning: this is a bad example because IMAP is an anemic protocol. This slide motivates delegation concept, not implementation.
Credential Replay

- Special (blunt) case of delegated authentication
- System B can authenticate on behalf of Person A because B borrows the credentials (password!) of A
Authenticating Services to Services

- Credential replay?
- Service credentials and trust relationships?
- Topological restrictions?

- Sure, but what about the “on behalf of a user” part?
RDBMS / JDBC Example

• How 'bout a portlet that reflects library account?

<table>
<thead>
<tr>
<th>My Library Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>You have 6 checked out books. 4 of these are overdue, accumulating $1.00 in fines each day. You should return them.</td>
</tr>
<tr>
<td>You currently owe the library $10.25. You should pay this fine or, like, you won't graduate.</td>
</tr>
</tbody>
</table>
My Library Account
You have 6 checked out books.

```
SELECT * FROM CHECKED_OUT_BOOKS WHERE PATRON_ID = ?;
```
**RDBMS / JDBC Example**

My Library Account

You have **6** checked out books.

```
SELECT * FROM CHECKED_OUT_BOOKS
WHERE PATRON_ID = ?;
```

Portlet asserts identity of user.
RDBMS / JDBC Example

My Library Account
You have 6 checked out books.

SELECT * FROM CHECKED_OUT_BOOKS WHERE PATRON_ID = ?;

Portlet \textit{arbitrarily} asserts identity of user.
Authenticating Services to Services

- Credential replay?
- Service credentials and trust relationships?
- Topological restrictions?

- Sure, but what about the “on behalf of a user” part?
Authenticating only service?

- Service credentials and trust relationships?

My Library Account
You have 6 checked out books.

SELECT * FROM CHECKED_OUT_BOOKS
WHERE PATRON_ID = ?;
CAS proxy tickets

- CAS proxy tickets authenticate service on behalf of user.
Delegated SAML Assertions

- Delegated SAML assertions also authenticate a service on behalf of a user
Enterprise Portal

- Dashboards
- Service delivery platform
- Portlets using delegated authentication to access backing services on user's behalf is a common pattern in enterprise portals
Proxy CAS

* This slide grossly simplifies some nice considerations not shown.
Delegated SAML Authentication

* This slide grossly simplifies some nice considerations not shown.
How It Works
Short version

• Attempt to access resource (WSP), get authentication request
• Modify authentication request with original SAML and present to IdP
• IdP responds with a new SAML assertion successfully responsive to authentication request from WSP
• Present new assertion to WSP, get original resource, set some headers to continue connecting
Portal layer

- User logs in to portal with SAML assertion
- Portal gets raw SAML assertion from SP
- uPortal selectively releases SAML assertion to portlets
Opting in a portlet

In portlet.xml:

```xml
<User-Attribute>
  <Description>
    SAML Assertion
  </Description>
  <Name>samlAssertion</Name>
</User-Attribute>
```
PortletRequest request;
Map userInfo = (Map)
    request.getAttribute(
        PortletRequest.USER_INFO);

String samlAssertion = (String)
    userInfo.get("samlAssertion");
SAML Delegation Java Library

- Abstracts getting from IdP a delegated SAML assertion from the raw initial SAML assertion
- Abstracts using delegated SAML assertion (via HttpClient abstraction)
Attempts to get the resource

- Response, presumably from the Shibboleth SP, is a request for authentication

- “PAOS”
Process WSP response

- Changes the authentication request response from WSP per Enhanced Client Profile
- Removes some elements
- Adds original SAML assertion that authenticated user to portal
Presents modified request to IdP

- Presents modified request for authentication, including embedded original SAML assertion authenticating user to portal, to IdP

- This authenticates the portal to IdP (via certificate)

- This authenticates the context to IdP (on behalf of the user authenticated by the prior SAML assertion)
IdP responds with SAML assertion

- IdP responds with a SAML assertion suitable for presentation to the backing WSP, authenticating the portal and the delegation
Present delegated assertion to WSP

• Library presents the SAML assertion to the WSP, successfully responding to the authentication request, and finally accessing the originally requested resource.

• Result: an HttpClient instance that will continue setting the appropriate headers and responding to authentication requests by the WSP
Configure SP to accept delegation

```xml
<PolicyRule type="Delegation" match="oldest">
  <del:Delegate>
    <saml:NameID>
      https://portal.example.org/shibboleth
    </saml:NameID>
  </del:Delegate>
</PolicyRule>
```
Quite a bit of detail and formality was just glossed over.
Software
Shibboleth

- IdP (support for vending the delegated assertions)
- SP (releases initial SAML assertion to portal, support for consuming the delegated assertions)
Java Delegation Support Library

- Implements using SAML assertion to interact with IdP to get new delegated SAML assertion
- Implements using delegated SAML assertion to retrieve one or more https:// resources from a backing service
uPortal extensions

- Implements support in uPortal for (selectively) making SAML assertion available to portlets so they can successfully use that Java library
Example portlet

- Demonstrates using the uPortal extension and the shibboleth-delegation Java library
Next Steps
Further Test, Use in the Real World

- Needs (more) adopters
- Improve documentation
- Attendant code maturity issues (this code is not bad, but it isn't honed through use either)
- Iterations and release march
Enhance Shibboleth SP

- (This point stolen from Shib SP roadmap discussion, cf. Scott Cantor)
- Move functionality from the Java library into the SP
- Allows maintaining that functionality closer to the rest of the SP code
- Eases implementing delegation support in more languages
Improve uPortal-Shib Story?

- uPortal already supports Shibboleth
  - Authentication
  - User attributes
  - And with this, delegated authentication
- Needs better documentation (what doesn't?)
- Certainly needs better marketing
Resources
Shib-uPortal Wiki Space

Shibboleth IdP and SP modules
Java Library

- Implements interaction with IdP to get delegated SAML assertion
- And basic retrieval of a resource via HTTPS using the assertion

https://source.jasig.org/sandbox/delegated-saml-authentication/
uPortal extensions

• Bridges from SP into the portal framework
• Delivers SAML assertion (selectively) to portlets as user attribute “samlAssertion”

https://source.jasig.org/sandbox/ShibbolethuPortalIntegration/
Portlet demonstrating use


Seems like ridiculously little code (Spring PortletMVC and use of Java library)

That's kind of the point.
Questions & Answers

Andrew Petro
Software Developer
Unicon, Inc.

apetro@unicon.net
www.unicon.net/blog/apetro