Shibbing uPortal JA-SIG Session

This 2-hour session determined what development is necessary to fully integrate shibboleth within uPortal.

Integration areas

1. Authentication to the portal

2. Getting a user's groups and attributes

3. User proxying

Authentication to the portal

This is just a configuration exercise. Basically, use mod_shib in Apache to protect the portal's location, and within uPortal, grab the username from the servlet API (REMOTE_USER).

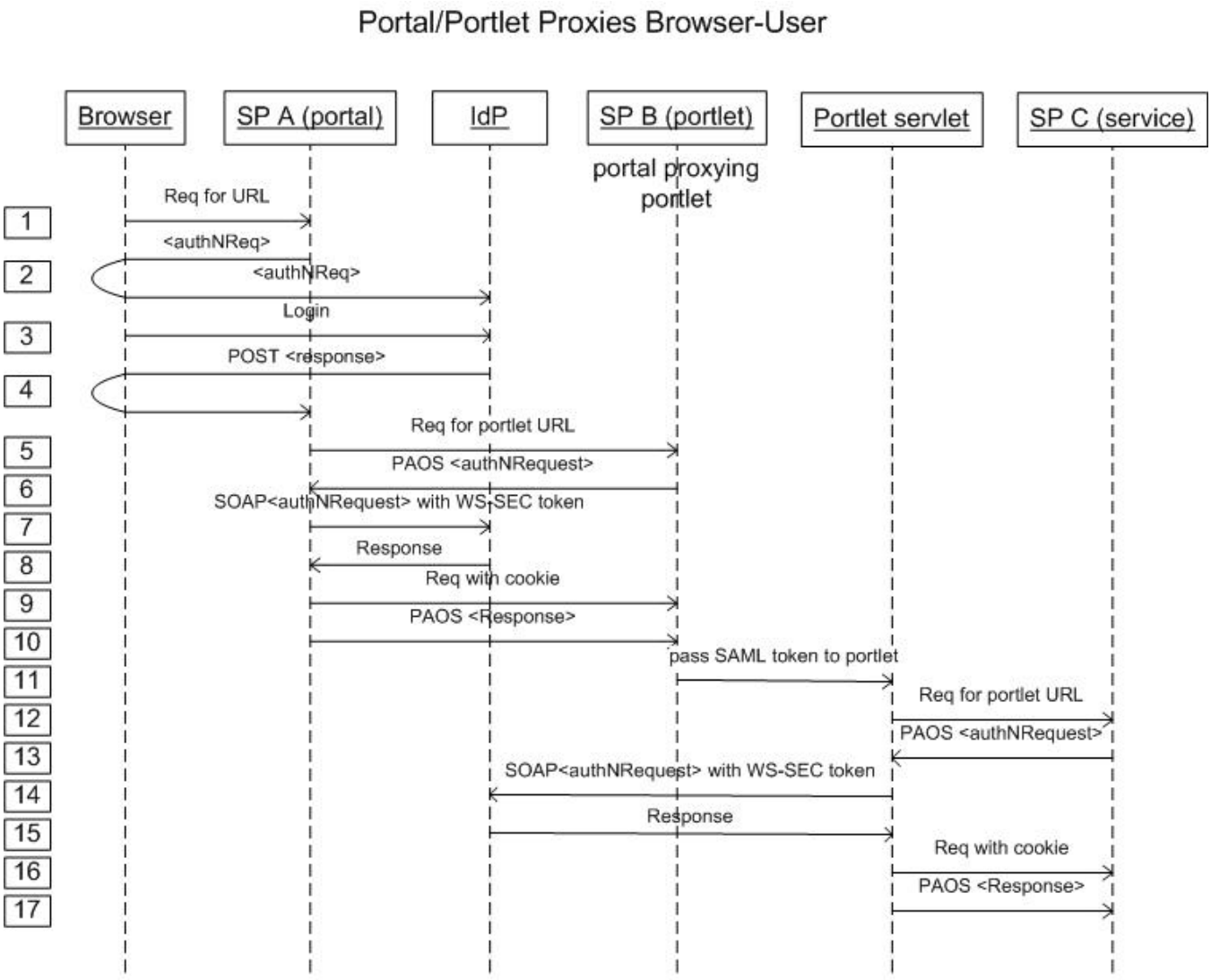
Getting a user's groups and attributes

- Use getRemoteUserContext() to get shib header values, ie, the SAML attributes
- PersonDirectory has an in-memory personDAO
    - Servlet filter not committed yet (Chris Doyle, jira issue persondir-37)
    - Done for shib+uPortal (@ JHU)
    - A few engineering issues remain with that servlet
    - Should be added to the PersonDirectory project
- PersonDirectory caches data for session lifetime
    - "username" is persisted in upUser table, maps username to internal key
    - If there are non-local users, need a globally unique "username"
    - PersonDirectory has no charset issues and default length limit of NNN
- Use PAGS to map SAML attributes to uPortal groups

User proxying

A high-level sequence diagram gives the general approach to doing the 4-tiers to proxy through the portal, a portlet, to external service. Essential underpinnings of this approach are

- Use of the SAML 2 ECP profile ("Enhanced Client or Proxy") that specifies how service to do service SAML flows.
- Use of standard SAML assertion attributes to constrain the use of a proxy token.



1-4. Browser-user authenticates to portal. SAML response includes list of entityIDs and URLs for portlets for which portal is authorized to delegate.

5-10. Portal follows ECP profile to obtain SAML token to be used by portlet. Occurs when portlet is “activated”, eg, when user switches to containing tab. SP B is a second instance of mod_shib running in same apache instance as SP A, leveraging the fact that portlet URL is in a URL namespace subordinate to portal’s.

11. Portlet API is used to pass SAML token to portlet, probably as a base64 encoded serialization of the token to ensure any signature remains valid.

12-17. Portlet follows ECP profile to obtain SAML token to access SP C as browser-user.

A visio of the above is available, in case you'd like to play!

Next steps

Scott Cantor (osu.edu) will draft initial specs for IdP enhancements needed to (1) support ECP and (2) add support for expressing policy that constrains delegation of proxy tokens.

Scott Cantor (osu.edu) will draft initial specs for the overall flow (of which the above is an inaccurate but indicative form).

~battags will review the above draft spec to ascertain degree of harmony with the existing CAS proxy flows.

Scott Cantor (osu.edu) will enhance the shibboleth SP to provide suitable logging of and policy control over acceptance of proxy tokens.

~edalquist@wisc.edu will draft initial specs for the work needed to complete the servlet filter mentioned above, as well as recommendations for using the PersonDirectory and PAGS for storing SAML attributes and mapping the user to uPortal groups.

~awp9 will review the various draft specs to ensure that together they produce a viable solution.

unknown will develop specs for a library (or whatever) to enable portlets to implement the ECP profile.

Tom Barton (uchicago.edu) will identify or provide a space in which to continue collaborative work on this topic, and will coordinate with appropriate Internet2, Unicon, U Chicago, and other people to keep this effort on track.

Tom Barton (uchicago.edu) will ensure that a portion of Unicon's engagement with U Chicago's uPortal deployment is assigned to this development activity.

Tom Barton (uchicago.edu) will ensure that JISC is brought in to learn of any interest they may have in this effort.