OpenIDSupport

This work is no longer being pursued as there was not enough interest in it after the release of the test plugin for the IdP

Work is underway to add native OpenID support to the Shibboleth 2.x Identity Provider. The current goal is for a working deliverable by the end of the 2009 calendar year. This page will document what can be expected in the deliverable.

Because this work is being done without any specific use cases in mind, and because there is some time before it will actually be deployed, we're taking a few liberties with regards to protocol support. Rather than attempting to support the OpenID protocol and community as it exists today, we're looking ahead at where the technology is going, and using that as our guide. As we near our completion, or as more specific use cases are presented, we may go back and selectively add support for some technologies that were initially omitted.

What will be delivered?

There will be at least three deliverables: two generic Java libraries, and a Shibboleth IdP extension.

XRD Java Library

The Extensible Resource Descriptor (XRD) specification is currently being drafted within the XRI Technical Committee of OASIS. This is the specification future versions of OpenID and OAuth discovery are scheduled to use for service discovery. It is a complete replacement for both XRDS (currently used by OpenID 2.0) and XRDS-Simple (currently used by OAuth Discovery 1.0).

The first deliverable will be a generic library which implements the parsing, processing, and publishing of XRD documents. It will provide basic implementations for verifying the signatures and trust of these documents.

OpenID Java Library

The second deliverable will be a new OpenID protocol library. This library will provide OpenID request and response handling, signature verification, and limited support for certain OpenID extensions. This library will NOT include a full OpenID provider or relying party... it is designed to implement the OpenID protocol, but not all of the associated business logic. Specific notes with regards to supported features:

- OpenID Authentication 2.0 will be supported. There are currently no plans to support OpenID Authentication 1.1
- There are currently no plans to support XRDS for discovery. Those portions of the library will instead use XRD, as that specification is developed.
- The current plan is to support XRI identifiers. That may be pushed to a later version of the library... I'm not sure.
- OpenID Provider driven identifier selection will be supported
- True directed identity will be possible (generating a unique identifier per user, per relying party). However, the library will likely not provide the full support for this... implementers will need to do a little more work
- Support for the following OpenID Extensions are planned: Attribute Exchange 1.0, PAPE 1.0, Simple Registration 1.0.

Shibboleth IdP Plugin

The final deliverable will be a Shibboleth IdP plugin which takes the above libraries and adds XRD and OpenID support to Shibboleth. This plugin will include the appropriate Shibboleth protocol handlers, attribute encoders, etc to make OpenID feel just as native to Shibboleth as SAML does.

Outstanding Questions

I (Will) am still working on finding the balance of how much functionality is provided in the lower level libraries vs the IdP Extension. For example, the OpenID library will likely contain several discreet sub-systems that handle specific tasks like message encoding, trust verification, etc. In order to allow for the greatest amount of flexibility, these systems will be loosely coupled such that any of them could be replaced with a custom implementation without an inordinate amount of work. Perhaps even more importantly, this separation of functionality also allows the different sub-systems to be directly hooked in to different portions of the software which is using the library. For example, the Shibboleth IdP will need to plug different parts of the OpenID library into very specific components, so having this level of flexibility is an absolute requirement. Shibboleth is very likely the exception to the rule in this regard and most users of the library will not need such flexibility. So the question remains as to how much bootstrapping library provides itself, wiring all the sub-systems together. The goal is most certainly **not** to provide an out-of-the-box OpenID provider or consumer, but we've yet to find the happy medium in between.