UW IdP 1.3 to 2.1 Migration

Introduction

This is the University of Washington's 1.3 to 2.1 migration experience.

Our migration plan provided for:

- Consistent response to Shibboleth 1.3 queries, including consistent eduPersonTargetedIDs (ePTIDs)
- Seamless migration with no interruption of service
- Clustered systems throughout the migration
- Rollback to version 1.3 if needed

New hardware, new name

Our 1.3 IdP had two names: hs.so.cac.washington.edu for authn and aa.so.cac.washington.edu for authz. At one time that seemed to make sense but no longer does. Our IdP hardware is out of warranty. We decided to install shib 2.1 on a couple of new boxes, with the new IdP cluster name of idp.u.washington.edu, and with a new certificate from InCommon. The new certificate brings one complication: some sites will get our new metadata but, because their local wayf has hardcoded our old IdP's url, they will continue to send users to hs.so.. As long as our metadata contains both the old and new credentials - and we push attributes - it will not matter to which idp users are sent. (see "things that can go wrong")

New IdP in pre-production

We installed and configured the 2.1 IdP, including the Terracotta clustering option; imported our 1.3 ePTID database; and converted ARPS to filters and some local resolver mods to scriptlets and etc. Our ePTIDs are computed when needed and stored in databases (one DB per IdP) which are lazily kept in sync. As a new ePTID's value is generated identically on all systems, including the old 1.3, so we can run redundent databases without much concern for replication.

At this point the new, 2.1 IdP was up and running and could be used for production by any SP that had the metadata. Google mail SSO was our first app. To test some 'real' load we manually switched a couple of local wikis to the new IdP. That proved it could run under at least a moderate load.

An emulation step

(A placate the Architect step, actually)

We configured a 2.1 IdP lookalike service on our 1.3 IdPs as a fallback if the 2.1 IdP completely failed. (See Scott's how-to.) Didn't expect to use it, and didn't, but it worked OK.

Production

All that was left was to update our InCommon metadata, and wait for SPs to flock to the new IdP.

There were a few thing that could go wrong:

- An SP doesn't update its metadata. This is OK. They just keep using the old idp until we are able to convince them to move forward.
- An SP updates metadata but uses a fixed wayf link to the old IdP. This is OK as long as the old credentials are in the new metadata, and the old IdP pushes attributes, AND the SP is not such an old SP that it doesn't know to use the pushed attributes but instead sends an attribute query anyway. This attribute query goes to the new IdP where it is rejected. We did not consider the 1.3-2.1 clustering option. In the latter case this is not OK. The wayf needs to be fixed.
- Our IdP is misconfigured and doesn't correctly respond to an SP. This can happen since there are a lot of configuration changes between 1.3 and 2.1. Checked for this by sending hand-crafted SSO requests to the IdP to make sure it responded correctly. Did this for all the SPs we really cared about. The aacli program is useful here as well - to make sure attribute generation and filtering is correct.
- The IdP not quite configured for production use. It is surprising how much debug configuration you can leave lying around .
- Users have bookmarked pages all throughout the old login path. These sort themselves out sooner or later.

Conclusion

We had a couple of the old-wayf issues, both quickly fixed with a call to the SP administrators. A couple SPs, who's administrators are on vacation, are still using the 1.3 IdP. Other than that we seem to have moved ahead quite seamlessly.

Postscript

1. Our first 2.1 IdP used port 7443, instead of the standard 443, in its SSO URL. Partly I didn't think it mattered; partly I needed a non-standard port to make the fallback idp on the old 1.3 system work. It turns out that some proxies and firewalls block port 7443 (just out of meanness I suppose). So it is important to use the 443 port. Not needing a fallback anymore we are switching our SSO URL back to the standard.