ShibIdPUpgradeHowTo

IdP software upgrade

If you are running IdP v1.3 consider the following before upgrading the software. IdP v2.x is substantially different from 1.3 code base.
Upgrade details are available at the wiki. I am listing a few.

Customization

Did you customize IdP 1.3 in any way? Most installers have customized to some degree. If you did, do the same in 2.x

Authn response post

IdP v1.3 used a jsp (IdP.jsp?) to post Authn assertion back to SP's, where as v2 uses velocity templates. If you customized the jsp, you may want to customize the velocity templates as well.

Velocity templates are bundled as part of the jar.
Customize and copy them to $IDP_WEBAPP/WEB-INF/classes/templates. This will override the default templates that is bundled in the jar.

Authentication

Integrating IdP with campus Auhectication may be different at each campus.
UCLA uses custom authentication service hosted by a different group in the campus. We used RemoteUSerAuthetication handler. If you are using LDAP or some other authn, consult Shibboleth wiki/forum.

handler.xml

```
..
..
<LoginHandler xsi:type="PreviousSession" authenticationDuration="PT15M">
        <AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession</AuthenticationMethod>
    </LoginHandler>

    <LoginHandler xsi:type="RemoteUser" authenticationDuration="PT15M">
        <AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport<
/AuthenticationMethod>
    </LoginHandler>
```

web.xml

Added a filter AuthnRequestFilter. This is the integration point with campus authn system.

```
<!-- UCLA custom  -->
<filter>
          <filter-name>authnrequest</filter-name>
          <filter-class>edu.ucla.iamucla.idp.custom.AuthnRequestFilter</filter-class>
        </filter>
        <filter-mapping>
                <filter-name>authnrequest</filter-name>
                <url-pattern>/Authn/RemoteUser</url-pattern>
        </filter-mapping>

..
..
..

<!-- Servlet protected by container user for RemoteUser authentication -->
    <servlet>
        <servlet-name>RemoteUserAuthHandler</servlet-name>
        <servlet-class>edu.internet2.middleware.shibboleth.idp.authn.provider.RemoteUserAuthServlet</servlet-
class>
        <load-on-startup>3</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>RemoteUserAuthHandler</servlet-name>
        <url-pattern>/Authn/RemoteUser</url-pattern>
    </servlet-mapping>
```

Convert ARP

In 2.x AFP replaces ARP. Schema is completely different. Handcoding of AFP or converting ARP to AFP is an arduous tasks if you have many policies. UCLA had 200+ custom release policies.
We wrote a script to convert the ARP to AFP.

Metadata

1.3 Metadata should be reusable in v2.x. SPs will continue to use SAML 1.1 protocol. Do not advertise new features (for ex, SAML 2 end points). Get the new version working for few days and then start rolling out new features of 2.x software.

How will you push metadata/AFP changes and refresh them frequently? How will you keep up with InCOmmon metadata changes? It was not seamless in early versions of IdP2 (when I started). We pushed changes during non-peak hours. It probably gotten better in the new release.

ePTID

Our 1.3 implementation was buggy. We took a chance and implemented standard algorithm. No one complained so far.

Is any of your relying party dependent on ePTID? Implementation may be different in 2.x. Make sure same algorithm is used to generate ePTID.

ePTID may be stored in a database. We thought using database is an additional dependency. We chose to generate on the fly, at the expense of run time performance (which is negligible now a days)

Will this upgrade cause service interruption?

There will be an interruption to IdP services during the upgrade. Keep SP's, users informed. Length of intrruption depnds on your deployment plan.

We were able to keep it down to 2 minutes. We brought up v2.x on a new set of servers. We switched from 1.3 to 2.x in the Content switch (load balancer).

Impact on users

SSO service may not be available during rollout/deployment.
Users will lose their SSO session. When users move from one application to the next, they may be asked to sign in again. SSO does not work across two versions of the software.

Inform the users early.

Impact on Shibboleth Service Providers

Upgrade should be transparent to SP's. No configuration change is mandated on the SP side.
Do you have SP specific customization, specially login page, logout page, help etc.? Plan for it.

Keep the SP's informed.

Session clustering and Terracotta

This may be the biggest challenge going from 1.3 to v2.x.

Do you need to cluster the sessions for load balancing and failover? You will have to use Terracotta to achieve this. There is a steep learning curve to configuring and using Terracotta.

Due to the complexity, IdP designers decided to discontinue Terracotta and use a new clustering solution in v3.0. Is it worth investing the time and effort in Terracotta now, knowing that it will go away in a year?

On the other hand how important is it to provide smooth failover? These are the trade offs you have to think about.

It is possible to run 2.x without Terrcotta clustering. See notes.

If you are using Terracotta I can help you with configuration. TC configuration from the wiki didn't work for us.

If you don't want to deal with the complexity, run IdP in stateless mode. See wiki at https://wiki.shibboleth.net/confluence/display/SHIB2/IdPStatelessClustering

Testing

If you are using Terracotta, set up test environment that mimics production, with ACTIVE ad STANDBY Terracotta instances. Test Terracotta fail over scenarios.

Backout

Do you have a backout plan if the unexpected happens? How quickly can you restore 1.3?

Apache/Tomcat

Are you fronting tomcat(IdP) with Apache? Will you host the IdP at the same location/URL as before (service endpoints in the published metadata)?
To guarantee smooth transition (for SP's) I advise not to change IdP SSO & AA endpoints (in the metadata) that is distributed to SPs.

Logout

Did you support IdP session logout in any way in 1.3? How will you do the same in 2.x?
We simply remove the session cookies.

SLO is a different ballgame. We do not support yet.

Build/Install

How will you build, install and maintain the software, specially if you have customization?

I chose to build shibboleth-common, shibboleth-idp, OpenSAML, Xmltooling etc. per instructions at https://wiki.shibboleth.net/confluence/display/SHIB2/SourceAccess. I created a maven web project and included all dependencies (including shibboleth-common, opensaml..) and added customizations in this project. Customization includes custom authn filter, velocity templates, overriding common library classes.

For upgrade strategy see notes at https://wiki.shibboleth.net/confluence/display/SHIB2/IdPUpgrades. Choose th eone that best works for you.