Migrating from Shibboleth 1 to Shibboleth 2 IdP

Andy Swiffin, University of Dundee – 04/02/10 Rev 1.2

Note: ***These case studies, prepared by member organisations of the UK federation, are provided for information purposes only and reflect the particular arrangements and experience of those concerned.***

Introduction

The biggest problem with any migration of a live system is that there can be an awful lot of people using it. At one point recently, comparing the reported figures in the REFEDS wiki https://refeds.terena.org/index.php/Federations, The University of Dundee had a higher number of logins per day than The Netherlands had in a week. With that many hands, the construction of a gallows will be light work, should it all go wrong.

There were a number of aims:

i) Retention of the same eduPersonTargetedID
ii) Retention of the same EntityID
iii) Continuing to service both Shibboleth versions 1 and 2 endpoints simultaneously during the migration
iv) Invisibility of the migration to users
v) Minimal intervention by library staff to keep things working.
vi) Keep all configurations as "standard" as possible.

The order of the above by importance will change depending who you ask. Users may think that iv) is most important but get i) wrong and they will soon notice after the event, when any existing personalisation of resources is lost.

In the rest of this document I plan to look at:

a) the migration strategy used
b) the pre migration testing and configuration
c) the timeline of how the migration progressed after metadata change
d) any intervention used to mitigate problems
e) how problem SPs were diagnosed.

The Strategy

We already have a Shibboleth 1.3 IdP which was hosted in a Tomcat 5.5.25 container behind an Apache 2.2.6 front end running on SLES10 Linux. In fact there are two IdPs: idp1 and idp2. Normally, idp.dundee points to idp1 which is running on "real hardware", whereas idp2 is hosted on a VMware ESX infrastructure and provides a backup.

By default Shibboleth 1 uses endpoints of the form /shibboleth-idp/

e.g. https://idp.dundee.ac.uk/shibboleth-idp/AA

whereas Shibboleth 2 has endpoints of the form /idp/

e.g. https://idp.dundee.ac.uk/idp/profile/SAML1/SOAP/AttributeQuery

The Tomcat-Apache connector is used to send incoming requests with a particular endpoint to Tomcat and so for Shibboleth 1 the Apache (2.2) httpd.conf could contain:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

#original shibboleth 1 endpoint
ProxyPass /shibboleth-idp ajp://localhost:8009/shibboleth-idp
```

It was decided that the following strategy for migration would be used:

- IdP1 would be unchanged and would continue to act as a Shibboleth 1 IdP.
- IdP2 would be upgraded with new Java, Tomcat and Shibboleth 2.1 installed.
- The DNS entry point idp.dundee.ac.uk will continue to point to idp1.
- Shibboleth 1 endpoints /shibboleth-idp/ will continue to be serviced through the Tomcat on idp1 as above.
- Shibboleth 2 endpoints /idp/ will be passed to the Tomcat on idp2 through an additional proxypass in the apache httpd.conf on idp1.

In this way the Apache server on idp1 continues to be the point of entry for all Shibboleth traffic but with Shibboleth 1.3 requests being serviced locally and Shibboleth 2 requests by the other server. In this way we would be able to service both versions simultaneously. This is possible because Apache is being used as the initial point of entry. The Apache – Tomcat connector uses "proxypass" instructions to pass http endpoints into Tomcat but the Tomcat destination(s) need not be on the same machine; so the change to the Apache configuration is quite simple with just the one line below being added to httpd.conf (0.170 is idp2):

```
#new shibboleth 2 endpoint
ProxyPass /idp ajp://134.36.0.170:8009/idp
```

Initial Installation

Initially Shibboleth 2 was installed on idp2 with a different entityID for testing (https://idptest.dundee.ac.uk/shibboleth), the local Apache was setup to service idptest and a dns cname was setup to point here. Effectively this was a completely new IdP. However when the time came for migration its entityID, certificates and metadata were changed to make it the same entityID as the Shibboleth 1 IdP:

https://idp.dundee.ac.uk/shibboleth.

IdP2 had Java updated to JRE 1.6.0_17 and Tomcat updated to 6.0.20 but as both /usr/local/java and /usr/local/tomcat are symbolic links there were no changes required to any environment settings from the Shibboleth 1 configuration.

Tomcat was installed on IdP2, loosely following:

https://spaces.internet2.edu/display/SHIB2/IdPApacheTomcatPrepare

except that the section "Supporting SOAP Endpoints" was ignored as the Java keystore is not required when fronting Tomcat with Apache.

No configuration changes were made to Tomcat's server.xml. Anyone who has used Tomcat authorisation with Shibboleth 1 will remember the changes necessary to this and subsequently to Shibboleth's web.xml. None of this is now required as Shibboleth 2 handles the authentication.

Shibboleth was installed, following:

https://spaces.internet2.edu/display/SHIB2/IdPInstall and
http://www.ukfederation.org.uk/content/Documents/Setup2IdP

Shibboleth 1 had originally been installed in /usr/local/shibboleth-idp and to preserve consistency this was renamed and Shibboleth 2 was also installed here (the default is now to install in /opt/).

For the initial install the entityID was set differently to the production environment so that the IdP could be tested independently and so relyingparty.xml differed slightly from the final settings:

```
    <DefaultRelyingParty
provider="https://idptest.dundee.ac.uk/shibboleth"
                          defaultSigningCredentialRef="IdPCredential">


<security:Credential id="IdPCredential"
xsi:type="security:X509Filesystem">
      <security:PrivateKey>/usr/local/shibboleth-
idp/credentials/idp1.key</security:PrivateKey>
        <security:Certificate>/usr/local/shibboleth-
idp/credentials/idptest.cer</security:Certificate>
</security:Credential>
```

with entityID and certificates being for idptest.dundee.ac.uk rather than idp.dundee.ac.uk.

During installation, metadata is created in /usr/local/shibboleth-idp/metadata/idp-metadata.xml. This must be edited to contain the correct entityID as above and the certificate which is embedded was changed from the self-signed one (automatically generated) to a JANET Certificate Service one for idptest.dundee.ac.uk.

An extra, but optional, stage to the migration is to install a service provider on campus that you can configure as appropriate. This is not strictly necessary as there are sufficient diagnostics available from offsite resources, but the warm fuzzy glow that you get from seeing things work in something you have absolute control over is worth the effort. As it happened I already had the SP installed on both idp1 and idp2. On IdP2 I swapped its name (and certificates) to idptest and added the idptest SP to the local Dundee metadata I keep for the IdPs. This SP uses a simple PHP script to output the received environment.

The IdP needs to be configured to release attributes as appropriate: in Dundee's case, derived by LDAP from the campus directory. This was done so as to replicate the Shibboleth 1.3 functionality. One essential attribute is TargetedID and that was generated as a computedID using the same salt and attribute as had been done in 1.3.

The federation document
http://www.ukfederation.org.uk/content/Documents/Setup2IdP suggests:

```
    <resolver:AttributeDefinition id="eduPersonTargetedID.old"
xsi:type="Scoped" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
        scope="dundee.ac.uk" sourceAttributeID="computedID">
        <resolver:Dependency ref="computedID" />

        <resolver:AttributeEncoder xsi:type="SAML1ScopedString"
xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonTargetedID" />

    </resolver:AttributeDefinition>
```

should be used to generate the old style TargetedID, but this will only generate it when the SP is coming in with SAML1. In our case it was useful to have it released the same way in SAML2 as well and so the following was added:

```
   <resolver:AttributeEncoder xsi:type="SAML2ScopedString"
            xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                  name="urn:mace:dir:attribute-
def:eduPersonTargetedID"
            friendlyName="eduPersonTargetedID" />
```

which fixed a local problem by ensuring that eduPersonTargetedID will be released at all times in the old format.

Initial Testing

The IdP was first tested against the local SP whose configuration was changed to access the IdP with SAML1 and SAML2. This is controlled by changing the order of session initiators in the SP config:

```
 <SessionInitiator type="Chaining" Location="/Login" isDefault="true"
id="Intranet"
          relayState="cookie"
entityID="https://idp.dundee.ac.uk/shibboleth">

          <SessionInitiator type="SAML2" defaultACSIndex="2"
template="bindingTemplate.html"/>
          <SessionInitiator type="Shib1" defaultACSIndex="5"/>

   </SessionInitiator>
```

The SP attribute map was modified to output the attribute names differently depending on the version of SAML used.

The SP was then accessed by the SP in SAML1 mode

affiliation student@dundee.ac.uk;member@dundee.ac.uk
entitlement
U03103;MDNU;EF;urn:mace:ac.uk:sdss.ac.uk:entitlement:emol.sdss.ac.uk:restricted;urn:mace:dir:entitlement:common-lib-terms
persistent-id
https://idp.dundee.ac.uk/shibboleth!https://idptest.dundee.ac.uk/shibbolethSP!BZxI8TK+LLYTp5NHSyyioT6C3V4=
targeted-id BZxI8TK+LLYTp5NHSyyioT6C3V4=@dundee.ac.uk
unscoped-affiliation student;member

and in SAML2 mode

affiliation2 student@dundee.ac.uk;member@dundee.ac.uk
entitlement2
U03103;MDNU;EF;urn:mace:ac.uk:sdss.ac.uk:entitlement:emol.sdss.ac.uk:restricted;urn:mace:dir:entitlement:common-lib-terms
persistent-id
https://idp.dundee.ac.uk/shibboleth!https://idptest.dundee.ac.uk/shibbolethSP!BZxI8TK+LLYTp5NHSyyioT6C3V4=
targeted-id BZxI8TK+LLYTp5NHSyyioT6C3V4=@dundee.ac.uk
unscoped-affiliation2 student;member

The IdP will also tell us which version of SAML the SP used in its idp-process.log:

Compare:

13:13:30.068 - INFO [Shibboleth-Audit:675] -
20091216T131330Z|urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding|_941795b22230124a9fb7aed7c0413803|https://idptest.dundee.ac.uk/shibbolethSP|urn:mace:shibboleth:2.0:profiles:saml1:query:attribute|https://idptest.dundee.ac.uk/shibboleth|urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding|_e3575c0a30757599e1f1b6c9f3e29029|alswiffin||eduPersonScopedAffiliation,eduPersonTargetedID.old,transientId,eduPersonTargetedID,eduPersonEntitlement,eduPersonAffiliation,|_d0e5c64b2f83706f568b401d655cd44d|_62075e871d0654a08eb6d99e57fae23b,

with:

13:12:30.943 - INFO [Shibboleth-Audit:898] -
20091216T131230Z|urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect|_70b1e24a4c2492bbcd9f6472c8f4c7b5|https://idptest.dundee.ac.uk/shibbolethSP|urn:mace:shibboleth:2.0:profiles:saml2:sso|https://idptest.dundee.ac.uk/shibboleth|urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign|_36819a1929db74033be247814865bd24|alswiffin|urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport|eduPersonScopedAffiliation,eduPersonTargetedID.old,transientId,eduPersonTargetedID,eduPersonEntitlement,eduPersonAffiliation,

The warm fuzzy glow referred to earlier comes from knowing that the IdP will service SAML2 requests or, at least for the time being, the more frequent SAML1 requests, giving both the same attributes values.

For completeness the IdP was then tested against remote SPs by registering the new temporary entityID idptest into the federation metadata. The test SP https://target.iay.org.uk/secure/printenv.cgi allows one to see the attributes being released and again to confirm that the all important targetedID has remained the same.

HTTP_HOST target.iay.org.uk
HTTP_REFERER https://idptest.dundee.ac.uk/idp/Authn/UserPassword
HTTP_SHIB_ATTRIBUTES (value provided; see below)
HTTP_SHIB_AUTHENTICATION_INSTANT 2009-12-16T12:52:05.661Z
HTTP_SHIB_AUTHENTICATION_METHOD
urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
HTTP_SHIB_EP_AFFILIATION staff@dundee.ac.uk;member@dundee.ac.uk
HTTP_SHIB_EP_UNSCOPEDAFFILIATION staff;member
HTTP_SHIB_IDENTITY_PROVIDER https://idptest.dundee.ac.uk/shibboleth
HTTP_SHIB_INETORGPERSON_EMPLOYEENUM M81X123
HTTP_SHIB_NAMEIDENTIFIER _8d3f054cdc296e84548a11a02ea4cfb7
HTTP_SHIB_NAMEIDENTIFIER_FORMAT urn:mace:shibboleth:1.0:nameIdentifier
HTTP_SHIB_ORIGIN_SITE https://idptest.dundee.ac.uk/shibboleth

X3EskdMPy6dEHWmlSvwZvJNjDD4=@dundee.ac.uk
HTTP_SHIB_TARGETEDID2
https://idptest.dundee.ac.uk/shibboleth!urn:mace:ac.uk:sdss.ac.uk:provider:service:target.iay.org.uk!X3EskdMPy6dEHWmlSvwZvJNjDD4=

It is not necessary to expose the IdP in the WAYF (which may confuse users) as the URL can be subtly modified in order to reveal the previously hidden IdP.

This is achieved by changing (for example):

https://wayf.ukfederation.org.uk/shibboleth-wayf/uk.wayf?shire=https%3A%2F%2Fgeoshibb.edina.ac.uk%2FShibboleth.sso%2FSAML%2FPOST&time=1260969663&target=cookie&providerId=https%3A%2F%2Fgeoshibb.edina.ac.uk%2Fshibboleth

to

https://wayf.ukfederation.org.uk/shibboleth-wayf/ukfull.wayf?shire=https%3A%2F%2Fgeoshibb.edina.ac.uk%2FShibboleth.sso%2FSAML%2FPOST&time=1260969663&target=cookie&providerId=https%3A%2F%2Fgeoshibb.edina.ac.uk%2Fshibboleth

During tests against sites with personalisation I was initially dismayed to be treated as an unknown, but this is because even though my targeted ID is the same as before, SPs use a combination of targeted ID and entityID to create a user's identity.

Testing was performed in this way against a number of service providers and not surprisingly it worked flawlessly: basically, if it releases good attributes to one it should to all (attribute release policies permitting). So it was time to go live.

Switching the IdP on

Prior to going fully live the following had to be performed in preparation:

a) In relying-party.xml, switch the default relying party over to the correct entityID and
b) change the certificate references to the idp.dundee.ac.uk certificate.
c) In idp-metadata.xml, change the entityID and paste in the correct certificate.

As a final test before contacting EDINA through the helpdesk, the local SP was given metadata for idp.dundee as the Shibboleth 2 IdP and correct operation of the IdP with its final entityID was confirmed.

To activate it I sent the federation helpdesk a request to swap the metadata and a copy of the metadata the IdP was using. The contents of /metadata/idp-metadata.xml (also available from https://idp.dundee.ac.uk/idp/profile/Metadata/SAML) were best for their import tools. They were able to take this and I received an email to say that at 18:15 on Thursday 10th December it had gone live.

Amazingly the first connection was just over 35 minutes later at 18:51 from Science Direct. Unfortunately this was an error message generated, most probably, by someone already authenticated via 1.3 trying to then access this SP which was asking for attributes from the Shibboleth 2 IdP (which of course knew nothing about the connection).

```
From idp-process.log:
18:51:55.434 - WARN
[edu.internet2.middleware.shibboleth.idp.profile.saml1.AbstractSAML1P
rofileHandler:582] - Error resolving principal name for SAML request
from relying party 'https://sdauth.sciencedirect.com/'
```

Then at 19:00 the first real successful authentication happened for! Some of these SPs are pretty diligent about keeping up to date (or very lucky with their timing).

The Aftermath

The first hint that not everything was working was when I tried one particular resource, which said

"Your IdP isn't supplying the eduPersonTargetedID attribute to the service. This is required, please contact your IdP for further information."

From an email to their helpdesk, they later said:

"Our metadata wasn't updated until 11am this morning hence the error. This should work fine now that the metadata is in sync."

However, clearly some metadata had been updated prior to that as it had started going to the Shibboleth 2 IdP for authentication but was not requesting attributes from that IdP. From 11:00 everything worked normally.

This became a recurring theme with a few SPs over the next few days and I monitored both the Shibboleth 2 idp-process.log and the Apache ssl_access_log. When accessing some SPs I would see the SP authenticate through Shibboleth 2 but then a request would come in as a "POST /shibboleth-idp/AA" (Shibboleth 1 endpoint) to Apache. A lookup of the IP address would reveal that the incoming request would be from the site that had just authenticated via Shibboleth 2!

For example, the logs for one of these resources (anonymised to protect the not so innocent) revealed a Shibboleth 2 authentication:

00:30:48.161 - INFO [Shibboleth-Audit:714] -
20091211T003048Z|urn:mace:shibboleth:1.0:profiles:AuthnRequest||https://www.a.resource/shibbolet
h|urn:mace:shibboleth:2.0:profiles:saml1:sso|https://idp.dundee.ac.uk/shibboleth|urn:oasis:names:tc:SA
ML:1.0:profiles:browser-
post|_be3d5512d6af0eb475a1d53400c739d3|alswiffin|urn:oasis:names:tc:SAML:2.0:ac:classes:unspeci
fied||_847d125b8894b061fa1aada21484130f|_a5157df3849ea8a92a8793349e646dbd

Followed by a Shibboleth 1 request for attributes:

ddd.cc.bbb.aa - - [11/Dec/2009:00:30:48 +0000] "POST /shibboleth-idp/AA HTTP/1.1" 200 676

A lookup of the IP address proves this:

aa.bbb.cc.ddd.in-addr.arpa name = the.same.resource

A number of resources were problematic in this way. One resource was particularly odd in that a session initiator URL worked fine, albeit doing everything through Shibboleth 1, whereas going through the WAYF demonstrated the split personality.

Another resource also displayed this split personality but in reverse, i.e. going to the Shibboleth 1 IdP for authentication but then sending a request to the Shibboleth 2

endpoint for attributes. But in contrast, an email to their metadata technical contacts elicited a speedy response and a fix on the Friday afternoon.

By midday on Friday (18 hours after the metadata change) the library had identified the following situation:

- Shib 2 and not working: 2 resources
- Still Shib 1.3 and working: 10 resources
- Shib 2 and working: 25 resources

By 4pm on the Friday (22 hours after the changeover) the situation had improved somewhat with a further adoption of the Shibboleth 2 metadata

- Shib 2 and not working: 2 resources
- Still Shib 1.3 and working: 5 resources
- Shib 1.3 from the morning but now working via Shib 2: 6 resources

As can be seen there was a steady drift over to Shibboleth 2 throughout the day: however it was no problem to us as long as they stayed on one or the other.

After the weekend there were few remaining problems. One resource was still coming to the 1.3 IdP but an email to the metadata technical contact elicited a speedy response and the admission that they were still manually updating the metadata (which they did straight away!). Another resource was still accessible through the session initiator URL via 1.3 and this (along with the failure via the WAYF) was fixed by their reboot on Wednesday.

Conclusion

The ultimate goal of any migration is to swap from one version to another seamlessly. This approach nearly achieves that by allowing incoming requests from both Shibboleth versions to be serviced during the changeover period. While many SPs were quick to migrate, requests were still arriving at the 1.3 IdP up to 6 days after the metadata was changed, albeit from just one SP by then. The approach also offers the advantage that a completely standard configuration is maintained using standard Shibboleth 2 endpoints, hence any further software updates will be easy to perform.

The dual endpoint servicing became a little unstuck with the few SPs who seemed to be singing from the two different hymn sheets simultaneously but there is little we can do to protect ourselves from this.

The approach described here is particularly aimed at those running Shibboleth on a Linux platform and so assumes confidence in manually modifying configuration files with a text editor. It should however translate readily to the Windows environment although, again, some manual configuration would be required.

It has been pointed out to me that it would be possible to perform this migration on one single platform by running two instances of Tomcat listening on different ports or even by running the two IdPs within one Tomcat container. It was suggested that this would simplify the process. However, I believe this is more complex, requires a non standard configuration and is invasive to the original production service. I had

considered the approach but rejected it on these grounds in favour of the two server approach that was used.

If resilience is an issue at all times for a site, then the new 2.1 IdP will have to be brought up as a fully resilient service in parallel with the existing resilient 1.3 service. Whereas Shibboleth 1.3 provided resilience through HAShib, Terracotta is used with 2.1. There will of course be additional hardware requirements

Acknowledgements

I'd like very much to thank Adrian Barker of UCL who first suggested to me the idea that I could use Apache proxypass to direct some endpoints to a Tomcat on one server and some to a different server rather than just assuming they were both local. Thanks, Adrian, it worked well :-)