Lagrad pseudonym identifierare

Installera mysql

```
# apt-get install mysql-server

.. under installationen sätts ett root-lösenord ..
```

Skapa databas och tabell

Skapa en databas...

```
# mysql -p
... använd lösenordet från installationen
mysql> SET NAMES 'utf8';
SET CHARACTER SET utf8;
CHARSET utf8;
CREATE DATABASE IF NOT EXISTS shibboleth CHARACTER SET=utf8;
USE shibboleth;
Query OK, 0 rows affected (0.00 sec)
```

Skapa en tabell...

```
mysql> CREATE TABLE IF NOT EXISTS shibpid (
  localEntity TEXT NOT NULL,
  peerEntity TEXT NOT NULL,
  principalName VARCHAR(255) NOT NULL default '',
  localId VARCHAR(255) NOT NULL,
  persistentId VARCHAR(36) NOT NULL,
  peerProvidedId VARCHAR(255) default NULL,
  creationDate timestamp NOT NULL default CURRENT_TIMESTAMP
  on update CURRENT_TIMESTAMP,
  deactivationDate timestamp NULL default NULL,
  KEY persistentId (persistentId),
  KEY persistentId_2 (persistentId, deactivationDate),
  KEY localEntity (localEntity(16), peerEntity(16),localId),
  KEY localEntity_2 (localEntity(16), peerEntity(16),
  localId, deactivationDate)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.00 sec)
```

skapa slutligen en user och ge rättigheter på tabellen. Denna user bör ha ett annat lösenord än `hemligt123`.

```
mysql> create user shibboleth identified by 'hemligt123';
Query OK, 0 rows affected (0.00 sec)
mysql> grant ALL on shibboleth.shibpid to 'shibboleth'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

Installera JDBC-connector

Hämta en JDBC-connector för mysql från http://dev.mysql.com/downloads/connector/j/ (tex mysql-connector-java-5.1.13.tar.gz). Packa upp i lämplig katalog och kopiera jar-filen (tex mysql-connector-java-5.1.13-bin.jar) till `lib`-katalogen för binär-paketet till shibboleth. Detta är katalogen med en `instal l.sh` och `install.bat`. Kör sedan `install.sh} (eller {{instal.bat` om du använder Windows) för att skapa en ny version av `idp.war` med mysql-connectorn instoppad. Starta sedan om din servlet-motor.

```
# cp mysql-connector-java-5.1.13-bin.jar /opt/jboss/server/default/lib
```

När du kör install.sh så får du frågan om du vill skriva över den existerande installationen. Svara 'N' (nej) på den frågan - annars försvinner dina inställningar. En ny war-fil skapas alltid.

Studera loggarna för din servlet-motor samt idp (idp-process.log). Om du får felmeddelanden om att det inte går att hitta mysql-connectorn i classpath så har du misslyckats med att installera connectorn - gå tillbaka och kontrollera att en ny `idp.war` faktiskt skapades.

Skapa DataConnector

Skapa följande `DataConnector` i `attribute-resolver.xml`:

```
<resolver:DataConnector id="StoredId"
        xsi:type="StoredId"
        xmlns="urn:mace:shibboleth:2.0:resolver:dc"
        generatedAttributeID="persistentId"
        sourceAttributeID="uid"
        salt="large random salt value">
   <resolver:Dependency ref="uid" />
   <ApplicationManagedConnection
            jdbcDriver="com.mysql.jdbc.Driver"
            jdbcURL="jdbc:mysql://localhost:3306/shibboleth?autoReconnect=true"
            jdbcUserName="shibboleth"
            jdbcPassword="hemligt123" />
</resolver:DataConnector>
```

Om du använder AD så kan det vara lämpligt att använda `sAMAccountName` som källa till `uid`. Det är dock fortfarande helt rimligt att @id för motsvarande `AttributeDefinition` är `uid`.

Här ska `myLDAP` ersättas med @id-attributet från den `DataConnector` som används för att hämta user-information. Ersätt "large random salt value" med ett stort (mellan 16 och 48 tecken) långt slumpmässigt lösenord. Ett sätt att generera ett sådant är programmet `apg` eller följande kommando:

```
# openssl rand -base64 36 2>/dev/null
```

Detta lösenord är mycket viktigt att spara - om det går förlorat eller behöver ändras kommer alla pseudonymer att ändras vilket betyder att alla SPer kommer att uppfatta inloggningar som "nya".

Skapa slutligen en `PrincipalConnector` i `attribute-resolver.xml` enligt:

```
<resolver:PrincipalConnector xsi:type="StoredId" xmlns="urn:mace:shibboleth:2.0:resolver:pc" id="
saml2Persistent"
        nameIDFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
        storedIdDataConnectorRef="StoredId" />
```

Attribut-definitioner

Börja med att se till att definitionen av attributet `uid` finns i `attribute-resolver.xml` och inte är utkommenterat. Om du använder AD så kan detta vara baserat på `sAMAccountName` istället men attributet kan fortfarande heta `uid`.

Skapa nu följande två attribut-definitioner i `attribute-resolver.xml`. Det första är legacy-attributet eduPersonPrincipalName och det andra den nya definitionen baserat på SAML 2.0 NameID. SWAMID rekommenderar att båda görs tillgängliga till alla SPer.

```
<resolver:AttributeDefinition id="eduPersonTargetedID" xsi:type="SAML2NameID"
        xmlns="urn:mace:shibboleth:2.0:resolver:ad"
        sourceAttributeID="persistentId">

   <resolver:Dependency ref="StoredId" />

   <resolver:AttributeEncoder xsi:type="SAML1XMLObject" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" />

   <resolver:AttributeEncoder xsi:type="SAML2XMLObject" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10" friendlyName="eduPersonTargetedID" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="persistentId" xsi:type="ad:PrincipalName"
                              xmlns="urn:mace:shibboleth:2.0:resolver:ad"
                              sourceAttributeID="persistentId">

   <resolver:Dependency ref="StoredId"/>

   <resolver:AttributeEncoder xsi:type="SAML1StringNameIdentifier"
                               xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                               nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" />

   <resolver:AttributeEncoder xsi:type="SAML2StringNameID"
                               xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                               nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" />
</resolver:AttributeDefinition>
```

Attribute-release

> Detta kommer att göra pseudonymer tillgängliga för alla SP:er vilket är SWAMIDs rekommendation. Om du inte vill lämna ut pseudonyer till alla så måste du ändra `PolicyRequirementRule` nedan

SWAMID rekommenderar att dessa attribut releasas till alla SP:er. Detta gör man enklast genom följande entry i `attribute-filter.xml`:

```
<AttributeFilterPolicy id="releasePermanentIdToAnyone">
   <PolicyRequirementRule xsi:type="basic:ANY" />
   <AttributeRule attributeID="persistentId">
      <PermitValueRule xsi:type="basic:ANY" />
   </AttributeRule>
   <AttributeRule attributeID="eduPersonTargetedID">
      <PermitValueRule xsi:type="basic:ANY" />
   </AttributeRule>
</AttributeFilterPolicy
```