Shib2IdpTomcat6

Ezt a lapot össze kellene vonni ezzel, és az elavult infókat frissíteni

JVM beállítások

A Tomcat6 jelenleg nem működik együtt tökéletesen 6-os JVM-mel (egész pontosan a commons-dbcp csomag a JDBC API pici megváltozása miatt), ezért egyelőre ajánlott 5-ös JVM-mel futtatni - FIXME.

A Shibboleth2 IdP nem hajlandó elindulni a JVM-mel szállított Sun-féle Xerces implementációval, ezért az IdP csomaggal szállított Xerces és Xalan implementációkat be kell másolni a $JAVA_HOME/lib/endorsed könyvtárba, vagy a következő kapcsolóval kell indítani a Tomcat-et:

```
java -Djava.endorsed.dirs=/path/to/xerces-libs
```

Shibboleth IdP telepítése

A kitömörített bináris disztribúció könyvtárában adjuk ki a

```
sh ant.sh
```

parancsot, ami megkérdezi a telepítési könyvtárat (${SHIB_HOME}) és a hostnevet, majd elkészíti azt, legenerálja a teszt-céllal használható kulcsokat és tanúsítványokat (ezeket a credentials könyvtárba teszi idp.key, idp.crt néven).

Ezután a tomcat-et futtató felhasználónak (nálam: tomcat) írási jogot kell adni a log könyvtárra, majd telepíthetjük az idp webalkalmazást a tomcat webapps root alá (nálam: a /var/lib/tomcat-6/webapps/ROOT)

```
chown tomcat:tomcat ${SHIB_HOME}/logs
cp ${SHIB_HOME}/war/idp.war /var/lib/tomcat-6/webapps/ROOT
```

Új SAML SP felvétele

Egy új SP felvételéhez csak az SP metaadatára van szükségünk SAMLv2 szabványos XML formában. A metaadatot vagy fizikailag el kell helyezni a ${SHIB_HOME}/metadata könyvtárban, vagy egy URL-en elérhetővé kell tenni a Shibboleth IdP számára.

Metaadat-források megadása a ${SHIB_HOME}/conf/relying-party.xml -ben

```
<!-- több provider megadása láncolással -->
<MetadataProvider id="ShibbolethMetadata" xsi:type="ChainingMetadataProvider"
xmlns="urn:mace:shibboleth:2.0:metadata">
   <!-- Fájlrendszerből olvasott metaadat -->
   <!-- Fill in metadataFile attribute with deployment specific information -->
   <MetadataProvider id="sp1" xsi:type="FilesystemMetadataProvider"
xmlns="urn:mace:shibboleth:2.0:metadata"
      metadataFile="${SHIB_HOME}/metadata/sp1-meta.xml" maintainExpiredMetadata="true">
    <!--Metaadat aláírás ellenőrzése -->
    <!--MetadataFilter xsi:type="SignatureValidation"
trustEngineRef="shibboleth.MetadataTrustEngine" /-->
   </MetadataProvider>
</MetadataProvider>
```

SAMLv2 Profilok beállítása

A ${SHIB_HOME}/conf/relying-party.xml -ben kell a következő módosításokat eszközölni:

Először a SAMLv2 SSO Profil alapbeállításai

```
<!-- a saját IDP entityID-je, amivel minden SP-hez egyszerre beállíthatjuk mint provider -->
<DefaultRelyingParty
  provider="https://idp.example.com/idp/shibboleth"
  defaultSigningCredentialRef="IdPCredential">

 <!--
   5 perces assertion érvényesség (óraszinkronizálás IdP - SP között fontos!)
   A válaszok kötelező digitális aláírása
   Attribútum push
 -->
 <ProfileConfiguration xsi:type="saml:SAML2SSOProfile"
    includeAttributeStatement="true"
    assertionLifetime="300000"
    assertionProxyCount="0"
```

```
    signResponses="always"
    signAssertions="never"
    encryptAssertions="conditional"
    encryptNameIds="conditional" />
</DefaultRelyingParty>
```

SP esetén az alapértelmezett beállítások felülírása:

```
<RelyingParty
    id="https://sp1.example.com/shibboleth"
    provider="https://idp.example.com/idp/shibboleth"
    defaultSigningCredentialRef="IdPCredential">
   <ProfileConfiguration xsi:type="saml:SAML2SSOProfile" encryptAssertions="never"/>
</RelyingParty>
```

Attribútum kiadása (címtárból)

Egy attribútum kiadásához két dolgot kell beállítani: a resolver-t és a filter-t. Előbbi felelős az attribútum megszerzéséért és a session kontextusba helyezésért, utóbbi az SP felé történő kiadást szabályozza.

Új attribútum beolvasása címtárból (${SHIB_HOME}/conf/attribute-resolver.xml) és SAMLv1 illetve SAMLv2 AttributeStatement -be kódolása:

```
<resolver:AttributeDefinition id="email" xsi:type="Simple"
  xmlns="urn:mace:shibboleth:2.0:resolver:ad"
  sourceAttributeID="mail">
  <resolver:Dependency ref="myLDAP" />
  <resolver:AttributeEncoder xsi:type="SAML1String"
    xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
    name="urn:mace:dir:attribute-def:mail" />
  <resolver:AttributeEncoder xsi:type="SAML2String"
    xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
    name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="mail" />
</resolver:AttributeDefinition>
```

A resolver:Dependency adja meg azt a forrást, amiből az attribútum feloldásra kerül. Ez esetünkben a myLDAP:

```
<resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory"
   xmlns="urn:mace:shibboleth:2.0:resolver:dc"
   ldapURL="ldap://ldap.example.com" baseDN="ou=people,dc=example,dc=com"
   principal="userid=shibboleth,ou=systems,dc=example,dc=com"
   principalCredential="password">
     <FilterTemplate>
           <![CDATA[
               (uid=$requestContext.principalName)
           ]]>
     </FilterTemplate>
</resolver:DataConnector>
```

NameIdentifier leképzés

Szintén az attribute-resolver.xml -ben kell beállítani az Assertion Subject NameID -t, ami az IdP-SP közötti azonosítóért felel. Példaképp egy SAMLv2 Tranziens azonosítót a következőképp állíthatunk be:

```
<resolver:AttributeDefinition id="transientId" xsi:type="TransientId"
xmlns="urn:mace:shibboleth:2.0:resolver:ad">
  <resolver:AttributeEncoder xsi:type="SAML2StringNameID"
    xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
    nameFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:transient" />
</resolver:AttributeDefinition>
<resolver:PrincipalConnector xsi:type="Transient"
  xmlns="urn:mace:shibboleth:2.0:resolver:pc"
  id="saml2Transient"
  nameIDFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:transient" />
```

Változat #2
cziernorbert hozta létre 2026-04-14 13:22:13 CEST
cziernorbert frissítette 2026-04-14 13:24:23 CEST