Shib2IdpInstall

Előkészületek

entityID

Fontos, hogy az entityID **egyedi** és **állandó** legyen. Javasolt forma: `https://idp.intezmenyneve.hu/idp/shibboleth`.

Tűzfal

Be kell engedni a 443-as és a 8443-as portokat. Ha nagyon szigorúan vesszük, akkor a 8443-as portot elegendő csak a szóbajöhető SP-kről beengedni, de ezzel általában nem vagyunk tisztában, ezért célszerű a "nagyvilágból" beengedni. Biztonsági szempontból nem sok különbség van a 443-as és a 8443-as porton elérhető alkalmazások között.

JDK

Debian Lenny alatt a `sun-java6-jdk` csomagot kell feltelepíteni. Telepítés előtt érdemes az aptitude-ban kikapcsolni az opcionális függőségek telepítését.

```
aptitude install sun-java6-jdk
```

Állítsuk be, a `JAVA_HOME` környezeti változót!

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

> **Figyelem**
>
> Az `openjdk-6-jdk` csomag használata esetén ne felejtsük feltenni a `ca-certificates-java` csomagot, anélkül ugyanis hibát fogunk kapni az IdP indításakor!

Shibboleth security provider

> **Info**
>
> A Shibboleth Security Provider csak akkor szükséges, ha a Java alkalmazásszerver (Tomcat) önállóan fog kéréseket feldolgozni és nem kerül elé proxy (Apache)

Be kell másolni a `lib/shib-jce-1.0.jar` állományt a `$JAVA_HOME/jre/lib/ext` könyvtárba. Ha az `ext/` könyvtár nem létezik, akkor hozzuk létre.

```
cp lib/shib-jce-1.0.jar $JAVA_HOME/jre/lib/ext
```

Ezek után be kell állítani, hogy a JRE használni is tudja ezt a providert. Ehhez a `$JAVA_HOME/jre/lib/security/java.security` fájlban keressük meg az ún. "security provider"-eket, és írjuk hozzá a következő sort:

> security.provider.**7**
> =edu.internet2.middleware.shibboleth.DelegateToApplicationProvider

- **Megj.:** a "security.provider." után következő szám mindig a megelőzőnél legyen eggyel nagyobb!

Bouncy Castle JCE

> **Bizonytalan információ!**
>
> Az alábbi leírás az 5-ös JVM-hez készült, 6-os JVM esetén erre nem biztos hogy szükség van.

A JVM-mel jövő Java Cryptography Engine (JCE) nem támogatja az összes kriptográfiai algoritmust, amelyre az Identity Providernek szüksége lehet (pl. XML Digital Signature, XML Encryption). A Bouncy Castle JCE ezek mellett még olyan algoritmusokat is tartalmaz (általában hatékonyabb és szabványosabb formában), amelyek benne vannak a Java JCE-ben.

Ehhez először le kell tölteni a Bouncy Castle JCE-t. A JCE állományok a Provider oszlopban, a "Signed Jar Files" részben találhatók. (A nevük `bcprov-jdk-VERZIO.jar`.) Letöltés után a .jar fájlokat a `$JAVA_HOME/jre/lib/ext` könyvtárba kell tenni.

```
wget http://www.bouncycastle.org/download/bcprov-jdk15-141.jar
cp bcprov-jdk15-141.jar $JAVA_HOME/jre/lib/ext
```

Ezek után be kell állítani, hogy a JRE használni is tudja ezt a providert. Ehhez a `$JAVA_HOME/jre/lib/security/java.security` fájlban keressük meg az ún. "security provider"-eket, és írjuk hozzá a következő sort:

> security.provider.**8**=org.bouncycastle.jce.provider.BouncyCastleProvider

- **Megj.:** a "security.provider." után következő szám mindig a megelőzőnél legyen eggyel nagyobb!

Tomcat 6

A 2.1.3 és újabb IdP megköveteli a Tomcat6 használatát (Tomcat5.5 -tel bizonyos böngészők esetén nem működik rendesen). A Debian Lenny nem tartalmazza a Tomcat6-ot, ezért a testing ágból kell feltelepíteni.

A Tomcat6-ra való frissítésről további - vázlatos - információkkal ez az oldal szolgál.

Telepítés

Ha minden rendben meg, és a squeeze source beállításra került, akkor elegendő egy

```
aptitude install tomcat6
```

parancs kiadása. Ez felpakolja a tomcat különböző függőségeit is - az ajánlott függőségek (tomcat6-admin, -docs, stb.) feltelepítése nem szükséges.

Ne felejtsük el, hogy a Tomcat szerver "tomcat6" user nevében fog futni! Mivel a Shibboleth servletnek szüksége van arra, hogy hozzáférjen a filerendszerhez, a Java Security Manager-t ki kell kapcsolni a `/etc/default/tomcat6` fájlban:

```
TOMCAT6_SECURITY=no
```

Ahhoz, hogy a Tomcat számára üzembiztosan elegendő memóriát biztosítsunk, ugyanebbe a fájlba (`/etc/default/tomcat6`) adjuk meg:

```
JAVA_OPTS="-Xms256M -Xmx512M -XX:-DisableExplicitGC "
```

Beállítás

A `/etc/tomcat6/server.xml` fájlt kell szerkesztenünk

Ha a Tomcat Apache mögött fut

A 8009-es porton figyelő Connector elem konfigurációjához hozzá kell adni, hogy a `tomcatAuthentication` értéke "false" legyen, ezen kívül a hozzáférést korlátozhatjuk a localhost-ra is (hiszen a Connector-t csak a helyben futó Apache mod_proxy_ajp konnektora érheti el).

```
<Connector port="8009" address="127.0.0.1" tomcatAuthentication="false"
               enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```

Ha a Tomcat önállóan, Apache nélkül fut

Ha a Tomcat Apache nélkül fut, akkor be kell állítani, hogy az SP-vel való kommunikációra fenntartott 8443-as porton egyből a Tomcat figyeljen.

```
<Connector port="8443"
           maxHttpHeaderSize="8192"
           maxSpareThreads="75"
           scheme="https"
           secure="true"
           clientAuth="want"
           SSLEnabled="true"
           sslProtocol="TLS"
           keystoreFile="IDP_HOME/credentials/idp.jks"
           keystorePass="PASSWORD"
           truststoreFile="IDP_HOME/credentials/idp.jks"
           truststorePass="PASSWORD"
           truststoreAlgorithm="DelegateToApplication"/>
```

Ahol az `IDP_HOME` az IdP alapkönyvtára, a `PASSWORD` pedig az IdP telepítésekor megadandó jelszó lesz.

> **Info**
>
> Amennyiben a Tomcat önállóan fut, szükség van a Shibboleth Security Provider telepítésére!

További információ angolul

Apache beállítás

Tanúsítványok beszerzése és bemásolása `/etc/ssl` vonatkozó alkönyvtárai alá.

Meg kell adni, hogy az Apache figylejen a 443-as és 8443-as portokon. Az alábbiak kerüljenek a `/etc/apache2/ports.conf` fájlba

```
Listen 443
Listen 8443
```

SSO URL (443-as port)

Be kell állítani a virtuális hosztot, amelyhez az IdP-t rendeltük. Először a 443-as portot konfiguráljuk.

```
<VirtualHost _default_:443>
  ServerName aai.example.org:443
  SSLEngine On
  SSLCertificateFile /etc/ssl/certs/aai.example.org.crt
  SSLCertificateKeyFile /etc/ssl/private/aai.example.org.key
  SSLCertificateChainFile /etc/ssl/certs/aai.example.org.crt
  ProxyRequests Off
  <Proxy ajp://localhost:8009>
    Allow from all
  </Proxy>
  ProxyPass /idp ajp://localhost:8009/idp retry=5
</VirtualHost>
```

Ezen a porton valamilyen széles körben ismert tanúsítványt kell használni, mivel a felhasználók böngészőjének ismerniük kell(ene) a kibocsátót.

AA ill. Artifact (8443-as port)

Ezen keresztül az SP és az IdP közvetlenül kommunikálnak egymással. Ide arra a tanúsítványra van szükség, amely a föderációs metadatában szerepel - az aláírója nem érdekes.

A csatorna felépítésekor az IdP és az SP is autentikálja magát. Az SP autentikációját az Apache végzi, ami nem végez kibocsátó-ellenőrzést (`optional_no_ca`). Ez utóbbit az IdP alkalmazás végzi el, ezért nagyon fontos, hogy a kliens tanúsítványát az Apache továbbadja az alkalmazásnak (`ExportCertData`).

> **Figyelem**
>
> Az Apache a tanúsítvány-ellenőrzésnél ellenőrzi a tanúsítvány típusát. Ezért az SP tanúsítványának vagy kliens tanúsítványnak kell lennie, vagy nem lehet benne típus információ.

```
<VirtualHost _default_:8443>
  ServerName aai.example.org:8443
  SSLEngine On
  SSLCipherSuite ALL:!ADH:!EXPORT56:!EXPORT40:RC4+RSA:!SSLv2:+HIGH:+MEDIUM:+LOW:+EXP
  SSLCertificateFile /etc/ssl/certs/aai-aa.example.org.crt
  SSLCertificateKeyFile /etc/ssl/private/aai-aa.example.org.key
  SSLVerifyDepth 10
  SSLVerifyClient optional_no_ca
  SSLOptions -StdEnvVars +ExportCertData
  ProxyRequests Off
  <Proxy ajp://localhost:8009>
    Allow from all
  </Proxy>
  ProxyPass /idp ajp://localhost:8009/idp retry=5
</VirtualHost>
```

A virtuális hoszt engedélyezése után be kell tölteni az `ssl` és `proxy_ajp` modulokat, majd újra kell indítani az apache-ot.

Shibboleth 2.x IdP servlet telepítés

Letöltés

A hivatalos IdP kiadás innen innen tölthető le

Alternatívaként az NIIF által patchelt, ezáltal SLO képes IdP kiadást ajánljuk, ami a NIIF AAI oldalról érhető el. A Single Logout-képes IdP-ről további információ itt.

Kicsomagolás

A `shibboleth-idp-2.x.x-bin.zip` fájl tartalma kicsomagolás után a `/usr/local/shibboleth-idp` könyvtár alákerül

```
cd /usr/local
jar -xf shibboleth-idp-2.x.x-bin.zip
```

Endorsed jar állományok

Sajnos - legalábbis a cikk írásakor - a "kincstári" Sun-os Tomcat (Java?) JAXP parser egy ismert memóriaszivárgást tartalmaz, ezért a disztribúcióban az `endorsed/` könyvtárban található .jar file-okat kézzel be kell másolni a Tomcat `endorsed/` könyvtárába.

- A Debian alatti tomcat6 csomag használatakor a `/usr/share/tomcat6/common/endorsed` könyvtárba kell tenni a jar file-okat (ezt a könyvtárt létre is kell hozni).
  mkdir /usr/share/tomcat6/endorsed cp endorsed/*.jar /usr/share/tomcat6/endorsed/

Installer

```
export JAVA_HOME=/usr/jdk
cd /usr/local/shibboleth-idp
chmod 755 install.sh
./install.sh
```

A telepítés során az alábbi kérdésekre kell választ adnunk:

> Is this a new installation? Answering yes will overwrite your current configurat ion. [yes|no]
> yes

Új telepítés, vagy sem.

> Where should the Shibboleth Identity Provider software be installed? /
> opt/shibboleth-idp-2.0.0
> /usr/local/shobboleth-idp

Itt található a letöltött és kicsomagolt shibboleth programcsomag

>

> What is the hostname of the Shibboleth Identity Provider server?
>
> idp.example.org
> idp.example.org

Shibboleth IdP alkalmazás URI alapú azonosítója.

> A keystore is about to be generated for you. Please enter a password that will be used to protect it.
> changeme

Feljegyzendő jelszó :)

Befejezés

Környezeti változó beállítása

```
IDP_HOME=/usr/local/shibboleth-idp
export IDP_HOME
```

Szimbolikus linkek megadása - az egyértelműség és konvenció kedvéért...

```
mv $IDP_HOME/conf /etc/`basename $IDP_HOME`
ln -s /etc/`basename $IDP_HOME` $IDP_HOME/conf
mv $IDP_HOME/logs /var/log/`basename $IDP_HOME`
ln -s /var/log/`basename $IDP_HOME` $IDP_HOME/logs
mkdir /var/lib/`basename $IDP_HOME`
mv $IDP_HOME/metadata /var/lib/`basename $IDP_HOME`/metadata
ln -s /var/lib/`basename $IDP_HOME`/metadata $IDP_HOME/metadata
```

Jogosultságok beállítása - hogy a `tomcat6` felhasználó hozzáférhessen az alábbi könyvtárakhoz

```
chown -R tomcat6 /var/log/`basename $IDP_HOME` /var/lib/`basename $IDP_HOME`
```

További, már telepített IdP-től függő tomcat beállítás

```
cd /var/lib/tomcat6/
mkdir -p conf/Catalina/localhost
```

Az így létrehozott könyvtárban készítsünk egy `idp.xml` nevű (a név legyen azonos a idp webalkalmazás nevével) fájlt az alábbi tartalommal:

```
<Context
        docBase="/usr/local/shibboleth-idp/war/idp.war"
        privileged="true"
        antiResourceLocking="false"
        antiJARLocking="false"
        unpackWAR="false" />
```

Naplófájlok rotálása

Az alapértelmezett logging.xml nem törli a régi állományokat, ezért ezek egy idő után megtöltik a diszket.

Erre a korrekt megoldás az (lenne), ha a Logback alrendszert utasítjuk, hogy az N (a példában 90) napnál régebbi fájlokat rotálja ki. Ehhez a `logging.xml`-ben adjuk meg a `maxHistory` paramétert az *összes* `rollingPolic`y-nál, valahogy így:

```
 <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
   <FileNamePattern>/usr/local/shibboleth-idp/logs/idp-access-%d{yyyy-MM-
dd}.log</FileNamePattern>
   <maxHistory>4</maxHistory>
 </rollingPolicy>
```

Sajnos azonban jelenleg a logback csak egy állományt töröl, a régi file-okat megtartja (pl. akkor is, ha több, mint egy napig nem futott az IdP. Amíg ez nincs megoldva, addig kerülő megoldás lehet cron-ból törölni a régi file-okat

```
sudo crontab -u tomcat6 -e


MAILTO=mail@example.com
#m h  dom mon dow   command
52 18 *   *   *     find /var/log/shibboleth-idp/ -mtime +90 -delete
```

Teszt

Ahhoz, hogy kiderítsük, működik-e (ill. fut-e :)) az IdP webalkalmazásunk, ahhoz böngészőben hívjuk meg az alábbi urlt: `https://idp.example.org/idp/profile/Status`, amennyiben az oldalon egy

`ok`-t látunk, akkor az alkalmazásunk fut, és elkezdhetjük beállítani az attribútumok feloldását és kiadását.

Ha nem működik a webalkalmazás, akkor az alábbi naplófájlokban kezdjünk el keresgélni:

- `/var/log/shibboleth/idp-error.log`
- `/var/log/shibboleth/idp-process.log`

A naplózás mélységét a `/etc/shibboleth/logging.xml` fájlban állíthatjuk be. Hibakereséshez érdemes a `<ErrorLog>` értékét `DEBUG`-ra állítani.

Shibboleth 2.0 IdP beállítás

Metadata beállítása

Metadata aláírás ellenőrzés beállítása

Az IdP-be beállított metaadat(ok) valódiságának ellenőrzéséhez szükséges egy ún. `TrustEngine` beállítása. Ezt a `relying-party.xml` -ben kell megtenni a `Security Configurations` részben:

```
<security:TrustEngine
   id="shibboleth.MetadataTrustEngine" xsi:type="security:StaticExplicitKeySignature">
 <security:Credential id="HREFSigner" xsi:type="security:X509Filesystem">
  <security:Certificate>/path/to/idp/credentials/href-metadata-signer-
2011.crt</security:Certificate>
 </security:Credential>
</security:TrustEngine>
```

A konfigurációban hivatkozott `href-metadata-signer-2011.crt` elérhető innen: https://metadata.eduid.hu/href-metadata-signer-2011.crt, SHA-1 lenyomata a következő: `FE:AE:0B:E8:FB:59:ED:F7:CB:7F:69:DF:19:4F:8B:6D:C7:F6:96:66`

HREF föderációs metadata beállítása az IdP-ben

A HREF metadata állomány elérhetősége:

- http://metadata.eduid.hu/current/href.xml

A Shibboleth IdP `relying-party.xml` konfigurációban a következőképpen lehet beállítani a HREF metaadatot (fontos hogy az előző pontban leírt `TrustEngine` is be legyen állítva):

```
<MetadataProvider id="HREF-Metadata"
     xsi:type="FileBackedHTTPMetadataProvider" xmlns="urn:mace:shibboleth:2.0:metadata"
     metadataURL="http://metadata.eduid.hu/current/href.xml"
     backingFile="/path/to/idp/metadata/href.xml"
     maxRefreshDelay="PT1H">
  <MetadataFilter xsi:type="SignatureValidation"
trustEngineRef="shibboleth.MetadataTrustEngine" />
</MetadataProvider>
```

Tovább a föderációba

Amennyiben a telepítendő IdP-t szeretnénk a HREF-be integrálni, úgy ennél a pontnál küldjünk egy levelet az aai@niif.hu címre, amely nyomán, ha minden rendben van, az IdP regisztrálásra kerül a Resource Registry-ben, s a válasz e-mail tartalmaz majd két hivatkozást, melyekről letölthetők az `attribute-filter.xml` és `attribute-resolver.xml` fájlok. Ezek már testreszabva tartalmazzák az IdP igényeit, az első fájlt elég csak bemásolni, a másodikban pedig - értelemszerűen - az egyes, a helyi erőforrásokra vonatkozó felhasználóineveket és jelszavakat kell kicserélni a megfelelőre.

További információk a Resource Registry-be történő felvételről

Attribútum filter automatikus frissítése
A Resource Registry automatikusan generálja minden egyes föderációban szereplő IdP számára a saját, testre szabott attribútum filterét, így célszerű úgy beállítani az IdP-t, hogy ezt a fájlt automatikusan töltse le. Ehhez hozzunk létre egy `confcache` nevű könyvtárat, adjunk rá írásjogot a `tomcat6` felhasználónak, majd szerkesszük a `conf/service.xml` fájlt. Az XML felső harmadában kerül megadásra az `AttributeFilterEngine`, melyet az alábbiak alapján kell átírni.

```
<Service id="shibboleth.AttributeFilterEngine"
       xsi:type="attribute-afp:ShibbolethAttributeFilteringEngine"
       configurationResourcePollingFrequency="3600000"
       configurationResourcePollingRetryAttempts="128">
      <ConfigurationResource url="https://rr.aai.niif.hu/gen_attribute-
filter.php/href/IDP_NEVE_AZ_RRBEN/attribute-filter.xml"
                             file="/path/to/shibboleth-idp/confcache/attribute-filter.xml"
xsi:type="resource:FileBackedHttpResource" />
</Service>
```

Több attribútum filter használata
Hasznos lehet, ha a föderációs szűrőkön kívül további irányokba kívánunk IdP-nkből attribútumokat kiadni. Elterjedt, hogy pl. különböző google szolgáltatásokhoz lehet az intézményen keresztül autentikálni, amely beállítási részleteket értelemszerűen a Resource Registry nem tartalmazza, így a letöltött friss, és a régi, helyi adatokat is tartalmazó fájlok egyesítése bosszantó plusz munka

lehet. Ezt elkerülendd dolgozhatunk több attribútum filterfájlból. Ehhez ismét a `conf/service.xml` fájlt kell szerkeszteni. Alább a fenti kódrészlet kiegészítése.

```
<Service id="shibboleth.AttributeFilterEngine"
       xsi:type="attribute-afp:ShibbolethAttributeFilteringEngine"
       configurationResourcePollingFrequency="3600000"
       configurationResourcePollingRetryAttempts="128">
      <ConfigurationResource url="https://rr.aai.niif.hu/gen_attribute-
filter.php/href/IDP_NEVE_AZ_RRBEN/attribute-filter.xml"
                             file="/path/to/shibboleth-idp/confcache/attribute-filter.xml"
xsi:type="resource:FileBackedHttpResource" />
      <ConfigurationResource file="/path/to/shibboleth-idp/conf/attribute-filter-local.xml"
xsi:type="resource:FilesystemResource" />
</Service>
```

Ezek a lépések természetesen kihagyhatók, ha nincs szándékunkban a föderáció tagjaivá válni, ekkor érdemes az alább részletezett attribútumokhoz kapcsolódó tudnivalókkal folytatni.

Statisztika küldés

Autentikáció beállítása

Attribútum feloldás beállítása

Attribútum kiadás beállítása

Változat #6
cziernorbert hozta létre 2026-04-14 13:22:18 CEST
cziernorbert frissítette 2026-04-17 10:56:29 CEST