Shibboleth IdP telepítés Debian

!!! bug "Elavult információ"

```
**Figyelem**: a Shibboleth szoftver ezen változata már nem támogatott. Az új verziókhoz a leírások itt találhatók:

* [Shibboleth2_IdP](https://help.edu.hu/books/aai-4BA/page/shibboleth2-idp)
* [Shibboleth2_SP](https://help.edu.hu/books/aai-4BA/page/shibboleth2-sp)
```

Előkészületek

Tanúsítvány

Kell készíteni egy megfelelő SSL szerver tanúsítványt. Ha más nem szól ellene, érdemes ugyanazt a tanúsítványt használni a felhasználók felé, mint az SP-k felé.

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.

Tomcat

JDK telepítés

Sajnos Etch alatt a `sun-java5-jdk` csomag függ egy csomó X-es csomagtól, melyeket nem biztos, hogy szeretnénk telepíteni egy szerveren, érdemes lehet

- feltenni a `sun-java5-jre` csomagot ÉS
- kézzel telepíteni egy JDK-t, mondjuk a http://java.sun.com oldalról letöltve

Ez igazából egy nagy *hack*, ugyanis ahhoz, hogy a tomcat-et csomagból telepíteni tudjuk, kell a `java2-runtime` csomag, amelyet biztosít a JRE is, **viszont** a Tomcat-nek JDK kell, hogy JSP-t tudjon futtatni.

```
* <small>**Megj.:** Minden JSP-t első futtatáskor a konténer (Tomcat) lefordít Java kóddá, aztán byte-kóddá, ezért tart jó sokáig az - újraindítás utáni - első request. Ezután az eredményt elcache-eli, így csak akkor kell újrafordítania, ha a JSP megváltozik.</small>
```

A JDK telepítés elég egyszerű, letöltjük a java.sun.com oldalról a nekünk tetsző verziót, aztán kicsomagoljuk, mondjuk a `/usr/lib` alá, aztán csinálunk egy szimbolikus linket, hogy a `/usr/jdk` mindig a "jó" JDK-ra mutasson.

Tomcat telepítés

Ha minden rendben meg, akkor elegendő egy

```
apt-get install tomcat5.5
```

Ez felpakolja a tomcat különböző függőségeit is.

Ahhoz, hogy a Tomcat rendben elinduljon, szükséges neki megmondani, hogy hol találja a JDK-t. Ezért tegyük a `/etc/default/tomcat5.5` fájlba a következőt:

```
JAVA_HOME=/usr/jdk
```

Ne felejtsük el, hogy a Tomcat szerver "tomcat55" 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/tomcat5.5` fájlban:

```
TOMCAT5_SECURITY=no
```

Tomcat konfiguráció

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_jk konnektora érheti el).

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

Apache

IdP-t telepíthetünk "standalone" Tomcat környezetre is, ekkor nincs szükségünk Apache-ra. A leírást ide kérjük :)

Az IdP telepítéséhez szükségünk lesz az alap apache szerverre (Etch-ben 2.2-es verziójú) és néhány modulra:

- ~~`libapache-mod-ssl`~~: a `mod_ssl` az `apache2` csomag része.
- `libapache2-mod-jk`

A konfiguráció lépései:

- `mod_ssl` modul betöltése; figyelés a 8443-as porton is
- `mod_jk` modul betöltése, konfigurálása
- VirtualHost konfigurálása
- autentikáció konfigurálása

mod_ssl

`/etc/apache2/ports.conf`

```
Listen 443
Listen 8443
```

Engedélyezzük az SSL modult.

```
a2enmod ssl
```

mod_jk

A `mod_jk` telepítés után alapértelmezetten engedélyezve van, ha mégsem lenne, az `a2enmod jk` paranccsal engedélyezhetjük.

A `/etc/libapache2-mod-jk/workers.properties` file-ban állítsuk be a `workers.tomcat_home` és a `workers.java_home` paramétereket a Tomcat ill. a JDK telepítésénél használt értékekre.

(tomcat_home=/usr/share/tomcat5.5 az alapértelmezett telepítésnél.)

Már csak az van hátra, hogy bizonyos URI-kra érkező kéréseket a modul átküldje a Tomcat-nek. Ehhez az alábbi konfigurációs direktívákat kell megadnunk valahol a szerver konfigurációban (pl. `/etc/apache2/apache2.conf`)

```
<IfModule mod_jk.c>
   JkWorkersFile /etc/libapache2-mod-jk/workers.properties
   JkLogFile /var/log/apache2/mod_jk.log
   JkLogLevel info
   JkMount /shibboleth-idp/* ajp13_worker
</IfModule>
```

A fenti példában a **shibboleth-idp** az IdP servlet telepítése során (később) megadott URI. Ez azt jelenti, hogy a `/shibboleth-idp` URI alá jövő összes kérést a Tomcat fogja megkapni.

- RedHat ES4 disztribúció alatt az **ajp13_worker** helyett **ajp13**-t kellett használni.

VirtualHost

Nem feltétlenül szükséges külön VirtualHost-ban futtatni az IdP-t, de sok szempontból "tisztább" konfigurációt eredményez. Egy működő konfig:

```
<VirtualHost 193.224.163.21:443 [2001:738:0:600:216:3eff:fe00:18]:443>
  ServerName      papigw.aai.niif.hu
  ServerAdmin     root@niif.hu
  DocumentRoot    /var/www/papigw.aai.niif.hu/htdocs
  CustomLog       /var/log/apache2/papigw.aai.niif.hu.ssl_access.log combined
  ErrorLog        /var/log/apache2/papigw.aai.niif.hu.ssl_error.log
  SSLEngine       On
  SSLCertificateFile      /etc/apache2/ssl/papigw.aai.niif.hu.crt
  SSLCertificateKeyFile   /etc/apache2/ssl/papigw.aai.niif.hu.key

  <Location /shibboleth-idp/SSO>
        AuthType Basic
        AuthBasicProvider ldap
        AuthName "Login to PAPIGW Identity Provider"
        AuthLDAPURL ldaps://directory.iif.hu:636/ou=users,o=niifi,o=niif,c=hu?uid?one
        AuthLDAPBindDN uid=papigw.aai.niif.hu,ou=https,ou=applications,o=niifi,o=niif,c=hu
        AuthLDAPBindPassword ******
        AuthzLDAPAuthoritative off
```

```
        require valid-user
    </Location>
</VirtualHost>

<VirtualHost 193.224.163.21:8443 [2001:738:0:600:216:3eff:fe00:18]:8443>
    ServerName      papigw.aai.niif.hu
    ServerAdmin     root@niif.hu
    DocumentRoot    /var/www/papigw.aai.niif.hu/htdocs
    CustomLog       /var/log/apache2/papigw.aai.niif.hu.ssl_access.log combined
    ErrorLog        /var/log/apache2/papigw.aai.niif.hu.ssl_error.log
    SSLEngine       On
    SSLCertificateFile      /etc/apache2/ssl/papigw.aai.niif.hu.crt
    SSLCertificateKeyFile   /etc/apache2/ssl/papigw.aai.niif.hu.key
    SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
    SSLVerifyClient optional_no_ca
    SSLVerifyDepth 10
    SSLOptions +StdEnvVars +ExportCertData
</VirtualHost>
```

- **Megj.:** IPv6-on is figyelünk :)

Autentikáció

SSO URI

Ez az az URI, amelyre az SP átirányítja a *felhasználót*, általában a szabványos https porton érhető el. A példában LDAP-ból azonosítjuk a felhasználót, majd az azonosított felhasználónevet a REMOTE_USER változóban adjuk át a Shibboleth IdP servletnek.

A `<Location ...>` blokkban bármilyen azonosítást beállíthatunk (MySql, plain file, stb).

- **Megj.:** Az LDAP SSL használatához a leírás itt található

AA URI

Az *Attribute Authority* általában a 8443-as porton érhető el.

Ezen az URI-n az SP-k kapcsolódnak hozzánk, hogy a felhasználóról adatokat kérjenek. Az SP-ket mindig tanúsítvánnyal azonosítjuk. "Természetesen" a request-et utána továbbítani kell a Tomcatben futó IdP servletnek. (Ezt a mod_jk fejezetben mutatott példában a `JkMount /shibboleth-idp/*` megadásával értük el.)

IdP servlet telepítése

Az IdP innen tölthető le: http://shibboleth.internet2.edu/latest.html

A tar.gz állományt csomagoljuk ki, majd lépjünk be a létrejövő könyvtárba.

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 tomcat5.5 csomag használatakor a `/usr/share/tomcat5.5/common/endorsed` könyvtárba kell tenni a jar file-okat.

Installer

```
export JAVA_HOME=/usr/jdk
./ant
```

A telepítés során az alábbi paramétereket kell megadnunk:

- Shibboleth IdP alkalmazás neve: az URI, amelyre érkező kéréseket a Tomcat az IdP servletnek ad át. Default: `shibboleth-idp`
- Filesystem- vagy manager-alapú telepítést akarunk? (Javasolt: Filesystem)
- Az IdP alkalmazás könyvtára. Default: `/usr/local/shibboleth-idp`
- Tomcat home. Default: `/usr/local/tomcat`, Debian alatt a `/var/lib/tomcat5.5` könyvtárat érdemes használni.

Könyvtárak

A telepítő minden file-t (binárisok, konfiguráció, logok, stb) egyetlen könyvtár alatti struktúrába tenne, de valószínűleg jobban járunk, ha az alkalmazásunk konfigurációja a `/etc`, a logok pedig a `/var/log` alatt találhatók.

Például:

```
export IDP_HOME=/usr/local/shibboleth-idp
mv $IDP_HOME/etc /etc/`basename $IDP_HOME`
ln -s /etc/`basename $IDP_HOME` $IDP_HOME/etc
```

```
mv $IDP_HOME/logs /var/log/`basename $IDP_HOME`
ln -s /var/log/`basename $IDP_HOME` $IDP_HOME/logs
```

Mivel a Debianon a Tomcat "tomcat55" user nevében fut, a szükséges állományokhoz hozzá kell tudnia férni

```
chown tomcat55 /var/log/`basename $IDP_HOME`
chmod 755 $IDP_HOME/bin/*
```

Ezek után már csak újra kell indítani a Tomcat-et, és az IdP-nek működnie kell. Ellenőrizni pl. úgy tudjuk, hogy meghívjuk a https://hostnev/shibboleth-idp/Status URI-t, amelynek az "AVAILABLE" stringet kell visszaadni.

Forrás

- Shibboleth Identity Provider Installation
- Shibboleth IdP installation with Debian and Tomcat
- Shibboleth IdP telepítése Debian 4.0 / Ubuntu 7.04 alatt (német nyelvű)
- Más környezetekre vonatkozó telepítési leírások
  - SUSE 10
  - OpenSUSE 10.2 (német)
  - **Tomcat-only telepítési leírások**
  - "Hivatalos" Tomcat-only leírás
  - Debian + Tomcat

Változat #1
cziernorbert hozta létre 2026-04-14 13:22:29 CEST
cziernorbert frissítette 2026-04-14 13:24:39 CEST