Shib2IdpAuth

Autentikáció nativan, Shib2Idp-b?l

LDAP-alapon

Szerkesszük a `${SHIB_HOME}/conf/login.config` -ot:

```
ShibUserPassAuth {
  edu.vt.middleware.ldap.jaas.LdapLoginModule required
     host="ldap.example.com"
     base="ou=people,dc=example,dc=com"
     ssl="false"
     serviceUser="userid=example-system,ou=systems,dc=example,dc=com"
     serviceCredential="password"
     userField="uid";
}
```

A serviceUser és a serviceCredential kihagyható, ekkor anonymous bind történik (azonban ilyen esetben a helytelen név / jelszó megadása LDAP Exception-t okoz és nem a jól értelmezhető hibás név / jelszó üzenetet adja a felhasználónak)

Ezután be kell állítani, hogy ezt a bekonfigurált autentikációt használja a Shibboleth (${SHIB_HOME}/conf/handlers.xml)

```
<LoginHandler xsi:type="UsernamePassword"
                 authenticationDuration="240"
                 jaasConfigurationLocation="file://${SHIB_HOME}/conf/login.config">

<AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</Authe
nticationMethod>
</LoginHandler>
<!-- SSO-hoz kell hogy az előző session-t át tudja venni -->
<LoginHandler xsi:type="PreviousSession">

<AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession</AuthenticationMe
```

```
thod>
</LoginHandler>
```

Az authenticationDuration paraméter elhagyása esetén az IdP 30 perc érvényességgel állítja ki a munkamenetet (`<AuthnStatement SessionNotOnOrAfter="...">`), tehát akár aktív tevékenység esetén is 30 perc múlva lejár a felhasználó session-je. Ezt érdemes tehát átállítani magasabb értékre.

A FORM-ot az idp.war -ban tudjuk testreszabni (login.jsp). A szállított login.jsp által beállított FORM tag és INPUT tag-ek tartalmát ne módosítsuk!

Ha a bejelentkezés nem sikerül jó felhasználónév/jelszó párral sem, a `logging.xml` szerkesztésével tudjuk megjeleníteni a debug üzeneteket (a `edu.vt.middleware.ldap` loggert kell átkonfigurálni).

Kerberos alapon

> **A szócikk vagy fejezet még megírásra vár**
>
> Ha ki tudod egészíteni, megköszönjük!

SQL (JDBC) alapon

A Shibboleth 2 IdP képes az autentikálást szabványos JAAS (Java Authentication and Authorization Service) modulokkal elvégezni, ezért lehetőség van relációs adatbázist használó autentikációs modul használatára is. Számos JAAS modul létezik adatbázisos autentikációra is, azonban ezek vagy túl bonyolultak, vagy nem kellően rugalmasak. Az alábbiakban az NIIF által fejlesztett (a Tagish/JDBC kódbázisán alapuló) modul beállítását mutatjuk be:

- JAAS/JDBC modul megfelelő verziójának letöltése
- jaas-jdbc-VERSION.jar és adatbázis driver jar bemásolása az idp webalkalmazás (idp.war) WEB-INF/lib könyvtárába
  - a MySQL Connector/J letölthető a MySQL oldalról
  - MySQL esetén a mysql-connector-java-{verzio}-bin.jar fájlra van szükségünk
- handler.xml -ben UsernamePassword login handler engedélyezése és RemoteUser login handler tiltása
- login.config ShibUserPassAuth-ban a JDBCLoginModul engedélyezése (a többi JAAS modul legyen kikommentezve!)
- adatbázis kapcsolattal összefüggő beállítások:
  - "hagyományos" megoldás
    - **dbDriver**: JDBC Driver osztály neve
    - **dbURL, dbUser, dbPassword**: adatbázis elérési paraméterek
  - JNDI használata esetén

    - **jndiResourceName**: DataSource API-t támogató JNDI név (bővebben lásd: Connection pool leírás)
- egyéb beállítások
    - **usersPreparedStatement**: egy olyan lekérdezés, ami a tárolt elhashelt jelszót kérdezi le egy felhasználónévhez (a felhasználónév helyén a ? karakter kell álljon, a lekérdezés egy vagy nulla sort kell visszaadjon!)
    - **passwordHashMethod**: a hasheléshez alkalmazott metódus (a használható metódusakat a Java Cryptography Architecture dokumentáció írja le).

A JAAS modul konfigurációja a login.config fájlban:

```
hu.niif.middleware.jaas.JDBCLoginModule required
  dbDriver="com.mysql.jdbc.Driver"
  dbURL="jdbc:mysql://databaseHost:3306/databaseName"
  dbUser="dbuser"
  dbPassword="randomsecret"
  usersPreparedStatement="SELECT password FROM users where username=?"
  passwordHashMethod="MD5";
```

LDAP-ból ellen?rzött X.509 tanúsítvánnyal

Ezen autentikációs mód a konténer (pl. Apache) által a klienstől elkért klienstanúsítványt veti össze a felhasználó LDAP bejegyzésében tárolt tanúsítványokkal (`userCertificate`). A modul használatának előfeltételei:

- a konténernek támogatnia kell a kliens-tanúsítványokat, azonban a CA ellenőrzés nem követelmény, a felhasználók self-signed tanúsítvánnyal is igénybe vehetik az autentikációs szolgáltatást
- az IdP-nek a kérésből el kell érnie a klienstanúsítványt
- a tanúsítványban szerepelnie kell a felhasználónévnek (mégpedig az `UID` mezőben)

A modul dokumentációja a ezen az oldalon érhető el.

Autentikáció konténer által

MySQL Autentikáció Apache-on keresztül

Az alábbiakban leírt Apache beállítások elsőre nyakatekertnek tűnhetnek, de az Apache 2.2-es sorozatában előforduló - ez idáig érdemben nem javított - bug miatt ez a megoldás működik csak.

Telepíteni kell a MySQL autentikációs Apache modult:

```
apt-get install libapache2-mod-auth-mysql
```

Engedélyezni kell a modul használatát

```
a2enmod auth_mysql
```

Az Apache adott hosthoz tartozó configjában meg kell adni:

```
Auth_MySQL_Info <host> <DB_user> <DB_password>
```

Ugyanitt meg kell adni az adott Location-re vonatkozó beállításokat - itt látja majd a webszerver, hogy ha az adott url-re érkezett kérés, akkor MySQL-ből kell autentikálnia az itt megadott paraméterek szerint

```
<Location /whereTo/Authn/RemoteUser>
 AuthType Basic
 AuthName "You can login here"
 AuthUserFile /dev/null
 AuthBasicAuthoritative Off

 AuthMySQL on
 AuthMySQL_Authoritative off
 AuthMySQL_DB VHOtools
 AuthMySQL_Password_Table vho_Users
 AuthMySQL_Password_Field password
 AuthMySQL_Encryption_Types PHP_MD5
 require valid-user
</Location>
```

MySQL Autentikáció TomCat-en keresztül

> **A szócikk vagy fejezet még megírásra vár**
>
> Ha ki tudod egészíteni, megköszönjük!

LDAP Autentikáció Apache-on keresztül

> **A szócikk vagy fejezet még megírásra vár**

> Ha ki tudod egészíteni, megköszönjük!

Single Sign-on

IdP Session

Az IdP-ben a session-nek nincs rögzített lifetime-ja, hanem aktivitásfüggő timeout értéket lehet beállítani. A jelenlegi IdP-ben az IdP session timeout független a fent megadott `authenticationDuration` értéktől, ezt az `internal.xml` állományban állíthatjuk be:

```
    <bean id="shibboleth.SessionManager"
          class="edu.internet2.middleware.shibboleth.idp.session.impl.SessionManagerImpl"
          depends-on="shibboleth.LogbackLogging">
        <constructor-arg ref="shibboleth.StorageService" />
        <constructor-arg value="<b>28800000</b>" type="long" />
    </bean>
```

A példában a StorageService konstruktorának értéke *ezredmásodpercben* értendő.

Single Logout

Változat #3
cziernorbert hozta létre 2026-04-14 13:22:15 CEST
cziernorbert frissítette 2026-04-14 13:24:26 CEST