Elavult / Archív

- Shibboleth IdP konfigurációja
- Attribútum kiadás
- Alkalmazások illesztése
- AAIInterop-Shib2SimpleSAMLphp
- AAIInterop-OpenSSOShib2
- Shibboleth IdP telepítés Debian
- Shibboleth IdP
- Resource Registry
- UApprove

Shibboleth IdP konfigurációja

> **Elavult információ**
>
> **Figyelem**: a Shibboleth szoftver ezen változata már nem támogatott. Az új verziókhoz a leírások itt találhatók:
>
> • Shibboleth2_IdP
> • Shibboleth2_SP

Az IdP alkalmazást az `idp.xml` állományon keresztül konfigurálhatjuk. Ebben a leírásban feltételezzük, hogy az IdP alkalmazás konfigurációs állományai a `/etc/shibboleth-idp` könyvtárban vannak.

M?köd? példa konfiguráció

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<IdPConfig
        xmlns="urn:mace:shibboleth:idp:config:1.0"
        xmlns:cred="urn:mace:shibboleth:credentials:1.0"
        xmlns:name="urn:mace:shibboleth:namemapper:1.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:idp:config:1.0 ../schemas/shibboleth-
idpconfig-1.0.xsd"
        AAUrl="https://idp.niif.hu:8443/shibboleth-idp/AA"
        resolverConfig="file:/etc/shibboleth-idp/resolver.ldap.xml"
        defaultRelyingParty="urn:niif.hu:aai:HREF"
        defaultAuthMethod="urn:oasis:names:tc:SAML:1.0:am:password"
        providerId="https://idp.niif.hu/shibboleth">

        <RelyingParty name="urn:niif.hu:aai:HREF" signingCredential="href_cred">
                <NameID nameMapping="shm"/>
        </RelyingParty>
        <RelyingParty name="urn:geant:niif.hu:niifi:sp:register.ca.niif.hu"
                     signingCredential="href_cred"
```

```
                    forceAttributePush="true">
                <NameID nameMapping="shm"/>
        </RelyingParty>

        <ReleasePolicyEngine>
                <ArpRepository
implementation="edu.internet2.middleware.shibboleth.aa.arp.provider.FileSystemArpRepository">
                        <Path>file:/etc/shibboleth-idp/arps/</Path>
                </ArpRepository>
        </ReleasePolicyEngine>

        <Logging>
                <ErrorLog level="DEBUG" location="file:/var/log/shibboleth-idp/shib-error.log"
/>
                <TransactionLog level="INFO" location="file:/var/log/shibboleth-idp/shib-
access.log" />
        </Logging>

        <NameMapping
                xmlns="urn:mace:shibboleth:namemapper:1.0"
                id="shm"
                format="urn:mace:shibboleth:1.0:nameIdentifier"
                type="SharedMemoryShibHandle"
                handleTTL="28800"/>

        <ArtifactMapper
implementation="edu.internet2.middleware.shibboleth.artifact.provider.MemoryArtifactMapper" />

        <Credentials xmlns="urn:mace:shibboleth:credentials:1.0">
                <FileResolver Id="href_cred">
                        <Key>
                                <Path>file:/etc/httpd/conf/ssl.key/idp.niif.hu.key</Path>
                        </Key>
                        <Certificate>
                                <Path>file:/etc/httpd/conf/ssl.crt/idp.niif.hu.crt</Path>
                        </Certificate>
                </FileResolver>
        </Credentials>
```

```
        <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.ShibbolethV1SSOHandler">
                <Location>https?://[^:/]+(:(443|80))?/shibboleth-idp/SSO</Location> <!-- regex
works when using default protocol ports -->
        </ProtocolHandler>
        <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.SAMLv1_AttributeQueryHandler"
>
                <Location>.+:8443/shibboleth-idp/AA</Location>
        </ProtocolHandler>
        <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.SAMLv1_1ArtifactQueryHandler"
>
                <Location>.+:8443/shibboleth-idp/Artifact</Location>
        </ProtocolHandler>
        <ProtocolHandler
implementation="edu.internet2.middleware.shibboleth.idp.provider.Shibboleth_StatusHandler">
                <Location>https://[^:/]+(:443)?/shibboleth-idp/Status</Location>
        </ProtocolHandler>

        <MetadataProvider
type="edu.internet2.middleware.shibboleth.metadata.provider.XMLMetadata"
                uri="file:/etc/shibboleth-idp/href-metadata.xml"/>
</IdPConfig>
```

XML elemek magyarázata

IdPConfig

Az IdPConfig elem attribútumai közül az xmlns: és xsi: attribútumokat nem szabad megváltoztatni, de van néhány, amit kötelező:

- *defaultRelyingParty*: ez adja meg, hogy melyik RelyingParty-t kell használni, ha a request alapján nem állapítható meg. Ha nincs ehhez tartozó RelyingParty elem, akkor az IdP nem indul el.
- *providerID*: ez adja meg az IdP egyedi azonosítóját a föderációban. Általában URN vagy URL formában adjuk meg.
- *resolverConfig*: az attribútum feloldás konfigurációs állományát adja meg.

- *AAUrl*: az Attribute Authority elérhetősége. (Erre csak a Shibboleth 1.1-el való kompatibilitás megőrzése érdekében van szükség. Nem biztos, hogy kötelező megadni...)

Általában nem szükséges megadni:

- *authHeaderName*: itt kell megadni, ha az SSO Handler más változóban kapja meg a felhasználó azonosítóját (principal), mint a REMOTE_USER szerver változó
- *defaultAuthMethod*: megadható, hogy az elkészített SAML Assertion milyen autentikációs metódust tartalmazzon. A lehetséges értékek a SAML 1.1 specifikáció 7.1-es szakaszában találhatók. Ha nincs megadva, akkor az értéke `urn:oasis:names:tc:SAML:1.0:am:unspecified`. A `defaultAuthMethod` értéke RelyingParty szintjén felülbírálható
- *maxSigningThreads*: az üzenet aláírására és egyéb műveletekre indított thread-ek maximális száma. Az IdP teljesítménye hangolható ezzel.
- *passThruErrors*: boolean változó, amely azt szabályozza, hogy a hibákat az IdP továbbadja-e az SP felé

Az IdP konfigurációban a többi XML Element az IdPConfig gyereke.

RelyingParty

Egy IdP tetszőleges mennyiségű RelyingParty-t kezelhet.

A legfelső szintű alapértelmezett beállításokon kívül minden egyes RelyingParty-ra beállíthatjuk az alábbi értékeket:

- *name* (kötelező): a RelyingParty neve. Ha nem egyezik meg az SP által küldött providerId-vel, akkor az IdP a metadata segítségével próbálja megállapítani, hogy az SP-re melyik RelyingParty definíció vonatkozik.
- *providerId*: az a providerId, amelyet az IdP használ a RelyingParty-k felé.
- *signingCredential*: az Assertion-ök és az SSL sessionben használt SSL kulcsokra vonatkozó FileResolver elem ID-jét adhatjuk meg itt.
- *AAUrl*: az Attribute Authority elérhetősége.
- *defaultAuthMethod*: megadható, hogy a RelyingParty számára elkészített SAML Assertion milyen autentikációs metódust tartalmazzon. A lehetséges értékek a SAML 1.1 specifikáció 7.1-es szakaszában találhatók. Ha nincs megadva, akkor az értéke az IdPConfig element-nél megadott érték, ill. `urn:oasis:names:tc:SAML:1.0:am:unspecified`.
- *passThruErrors*: boolean változó, amely azt szabályozza, hogy a hibákat az IdP továbbadja-e az SP felé. Alapértelmezett érték: false
- *signAssertions*: boolean változó, amely azt szabályozza, hogy az IdP aláírja-e a kiállított Assertion-öket. Leginkább akkor van rá szükség, ha az Assertion-t más alkalmazásnál is fel akarjuk használni. Alapértelmezett érték: false

- *forceAttributePush*: boolean változó, ennek segítségével ki lehet kényszeríteni az Attribute Push használatát. Alapértelmezett érték: false

A RelyingParty element NameID gyermeke segítségével állítható be a használt NameID kezelés.

ReleasePolicyEngine

Itt adhatjuk meg az attribútum kiadás implementációját (ezt általában nem kell változtatni) és az ARP állományok elérhetőségét.

Logging

A Logging element szabályozza a naplózási szintet, ill. a naplófile-ok helyét. Részletesebb beállításokra a Log4J-t is használhatjuk. (Lásd még: Értelmes naplóüzenetek (IdP))

NameMapping

Ebben az elemben adható meg a NameMapper implementációja, illetve az assertionökben használt azonosító (Subject Identifier) formátuma.

- Az alapértelmezett értékek az esetek többségében megfelelők, csak akkor módosítsd, ha tudod, mit csinálsz!

Attribútumok:

- *id*: egyedi név, erre lehet hivatkozni a NameID elementben.
- *format* (URI): ez határozza meg a Subject Identifier formátumát. Tetszőleges URN használható, amiben az IdP és az SP megegyezik. Néhány gyakrabban használt formátum:
  - `urn:mace:shibboleth:1.0:nameIdentifier`: alapértelmezett Shibboleth azonosító (tranziens, átlátszó)
  - `urn:oasis:names:tc:SAML1.1:nameid-format:X509SubjectName`: X.509 tanúsítvány DN. A GridShib használja ezt a formátumot.
  - `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`: email-cím, használata nem javasolt
  - `http://schemas.xmlsoap.org/claims/UPN`: MS UPN, az ADFS integrációhoz használható
- *class*: a NameMapper implementációjának a javaclass útvonala. (HAShib használatához módosítani kell.)
  A további attribútumok csak az alapértelmezett implementáció esetén értelmezhetők.
- *handleTTL*: azt határozza meg, hogy az IdP mennyi ideig őrizze a Session Cache-ében a kiosztott azonosítókat. (Csak `urn:mace:shibboleth:1.0:nameIdentifier` formátum esetén értelmezhető.) Ezt követően erre az azonosítóra történő hivatkozás már nem lesz

megengedett, a felhasználónak esetleg újra kell azonosítania magát.

- *type*: azt adja meg, hogy az SSO Handler és az Attribute Authority között milyen formában utazzanak az azonosítók. Lehetséges értékek:
  - `CryptoHandleGenerator`: szimmetrikus kódolással titkosított azonosítók használata
  - `Principal`: az SSO Handler-től megkapott azonosító átadása az Attribute Authority-nek
  - `SharedMemoryShibHandle`: (alapértelmezett) megosztott, memóriában tárolt session cache. Ha az SSO Handler és az Attribute Authority egy konténerben futnak, ezt érdemes használni.

ArtifactMapper

Itt adható meg az ArtifactMapper implementációja. HAShib használata esetén át kell állítani.

Credentials

Ebben az elemben adhatók meg a használt titkos kulcsok és tanúsítványok. Több is megadható, az *id* attribútum értékével hivatkozhatunk rájuk, pl a RelayingParty konfigurációban.

ProtocolHandler

Itt adhatók meg az egyes handler servletek elérhetőségei. Általában nem szükséges felülírni!

Forrás

** Shibboleth Wiki**

- IdP fő konfiguráció
- Relying Party konfiguráció
- NameMapping

Attribútum kiadás

Elavult információ

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

- Shibboleth2_IdP
- Shibboleth2_SP

Az Attribute Release Policy (ARP) határozza meg, hogy az attribútum feloldás után rendelkezésre álló attribútumok közül mely attribútumokat lehet az SP-nek kiadni. Egy ARP vonatkozhat a teljes IdP-re ("site" ARP), illetve az azonosított felhasználóra is. A site-ARP-k általában a **arps/arp.site.xml** állományban, a felhasználói ARP-k pedig az **arps/arp.user.$PRINCIPAL.xml** állományban találhatók, ahol $PRINCIPAL megegyezik a REMOTE_USER változóban megkapott értékkel.

M?köd? példa

```
<?xml version="1.0" encoding="UTF-8"?>
<AttributeReleasePolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="urn:mace:shibboleth:arp:1.0"
      xsi:schemaLocation="urn:mace:shibboleth:arp:1.0 shibboleth-arp-1.0.xsd" >
 <Description>Not The Simplest Possible ARP.</Description>
 <Rule>
  <Description>Mindenkire vonatkozó szabályok</Description>
  <Target>
   <AnyTarget/>
  </Target>
  <Attribute name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation">
   <AnyValue release="permit"/>
  </Attribute>
  <Attribute name="urn:mace:dir:attribute-def:eduPersonOrgDN">
   <AnyValue release="permit"/>
  </Attribute>
 </Rule>
```

```
<Rule>
<Description>NIIFI által üzemeltetett SP-kre vonatkozó szabályok</Description>
<Target>
<Requester
matchFunction="urn:mace:shibboleth:arp:matchFunction:regexMatch">.*\.n?iif\.hu\/.*</Requester>
</Target>
<Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName">
<AnyValue release="permit"/>
</Attribute>
<Attribute name="urn:mace:dir:attribute-def:mail">
<AnyValue release="permit"/>
</Attribute>
<Attribute name="urn:mace:dir:attribute-def:cn">
<AnyValue release="permit"/>
</Attribute>
<Attribute name="urn:mace:dir:attribute-def:eduPersonEntitlement">
<Value release="permit"
    matchFunction="urn:mace:shibboleth:arp:matchFunction:regexMatch">
^urn:niif.hu:services:aai:entitlement:.*
</Value>
</Attribute>
</Rule>
</AttributeReleasePolicy>
```

ARP feldolgozás menete

Az Attribute Authority a rendelkezésre álló ARP-kből (tehát site és felhasználói ARP-kből) egy ún. **effective ARP**-t állít elő.

1. Meghatározza, hogy melyik ARP file-okat kell feldolgozni.
2. Meghatározza, hogy melyek azok a szabályok, amelyek az attribútum lekérdezéshez kapcsolódnak
   - Minden ARP szabály, amely alapértelmezettnek vannak megjelölve, automatikusan bekerül az ARP-be, anélkül, hogy az illesztési függvényeket (matchFunction) végrehajtaná
   - Minden nem alapértelmezett szabály illesztési függvénye alapján megállapítja, hogy a providerId alapján vonatkozik-e a kérést indító félre.
3. Attribútum filter létrehozása
   1. Minden attribútumhoz megállapítja a vonatkozó Rule-ok listáját

2. Ebből a listából az összes olyan attribútum értéket kiveszi, amelyre *deny* szabály vonatkozik
3. Ha egy szabály úgy rendelkezik, hogy minden érték kiadható, akkor az egyes értékekre vonatkozó *deny* szabályok szűkítik a kiadható értékek listáját. Ha egy szabály az attribútumok összes értékének kiadását megtiltja, akkor az egyes értékekre vonatkozó engedélyek figyelmen kívül lesznek hagyva.

ARP Rule

Az ARP szabályok különböző illeszkedési vizsgálatok segítségével megállapítják, hogy egy SP-nek egy-egy attribútum milyen feltételekkel adható ki.

matchFunction

Ez az attribútum adja meg, hogy milyen illesztési eljárást kell használni az illeszkedési vizsgálatnál. Lehetséges értékei:

- **urn:mace:shibboleth:arp:matchFunction:stringMatch**: *true*, ha két karakterlánc pontosan megegyezik (ez az alapértelmezett illesztési függvény)
  - ugyanezt jelenti: **urn:mace:shibboleth:arp:matchFunction:exactShar**
- **urn:mace:shibboleth:arp:matchFunction:stringNotMatch**: *true*, ha két karakterlánc eltér
- **urn:mace:shibboleth:arp:matchFunction:regexpMatch**: *true*, ha a karakterlánc megfelel a paraméterként megadott reguláris kifejezésnek
- **urn:mace:shibboleth:arp:matchFunction:regexpNotMatch**: *true*, ha a karakterlánc nem felel meg a paraméterként megadott reguláris kifejezésnek
- **urn:mace:shibboleth:arp:matchFunction:anyValueMatch**: tetszőleges nem üres string esetén *true*

Target

A Target elemnek kétféle gyermeke lehet:

- `**AnyTarget**`: minden SP-re vonatkozik a szabály (az azonosítatlan SP-kre is!)
- `**Requester**`: a szabály akkor kerül az effective ARP-be, ha az SP providerId-je illeszkedik

Attribute

Egy Rule 0 vagy több Attribute elemet tartalmazhat. Tartalmaznia kell egy `name` paramétert, amely az attribútum teljes neve (általában URN, lásd az attribútum feloldás leírását). Az elemnek kétféle gyermeke lehet:

- `**AnyValue**`: az attribútum bármely értékére vonatkozik a szabály
- `**Value**`: ebben az esetben kötelezően szerepel egy `**release**` paraméter, melynek értéke *permit* vagy *deny* lehet. Itt is opcionálisan megadható a `**matchFunction**` paraméter.

Constraint

A Constraint-ek használatával attribútumok kiadását más attribútumok értékéhez is köthetjük, így pl. megtehetjük, hogy a "hozzajarulasBeszerezve" nevű attribútum `true` értékéhez kössük az attribútumok kiadását.

A megkötések konfigurációjához lásd: https://spaces.internet2.edu/display/SHIB/ArpConstraint

Tesztelés

A Shibboleth IdP-hez tartozik egy **resolvertest** névre hallgató program, amellyel ellenőrizhetjük az attribútumok kiadását is. Használatához először be kell állítani a telepítésnek megfelelően az IDP_HOME és a JAVA_HOME változókat.

Példa: `/usr/local/shibboleth-idp/bin/resolvertest --idpXml=file:///etc/shibboleth-idp/idp.xml --user=bajnokk` (Azonosítatlan SP-nek kiadott attribútumok)

```
<Attribute xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
□□   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
□□   AttributeName="urn:mace:dir:attribute-def:eduPersonScopedAffiliation"
□□   AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
  <AttributeValue Scope="niif.hu">employee</AttributeValue>
</Attribute>

<Attribute xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
□□   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
□□   AttributeName="urn:mace:dir:attribute-def:eduPersonOrgDN"
□□   AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
  <AttributeValue>o=niifi,o=niif,c=hu</AttributeValue>
</Attribute>
```

Példa 2.:`/usr/local/shibboleth-idp/bin/resolvertest --idpXml=:///etc/shibboleth-idp/idp.xml --user=bajnokk --requester=https://dev.aai.niif.hu/shibboleth` (Azonosított SP-nek kiadott attribútumok.)

```
...
<Attribute xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
□□   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
□□   AttributeName="urn:mace:dir:attribute-def:eduPersonPrincipalName"
□□   AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
  <AttributeValue Scope="niif.hu">bajnokk</AttributeValue>
</Attribute>
...
```

Hivatkozások

- https://spaces.internet2.edu/display/SHIB/IdPARPConfig
- https://spaces.internet2.edu/display/SHIB/AttributeReleaseRule
- https://spaces.internet2.edu/display/SHIB/ArpConstraint
- ShARPE ARP Editor

Alkalmazások illesztése

- Drupal shib_auth module (angolul / in English)
- Drupal illesztése Shibboleth-hez (elavult)
- MediaWiki illesztése Shibboleth-hez
- Webmail szoftverek illesztése Shibboleth-hez
- Moinmoin illesztése Shibboleth-hez
- Könyvtári rendszerek illesztése
- Office 365 illesztése Shibboleth IdP-hez

AAIInterop-Shib2SimpleSAMLphp

!!! bug "Elavult információ"

```
**Figyelem**: ez a szakasz vagy szócikk elavult információkat tartalmazhat!
```

Shibboleth2 IdP - SimpleSAMLphp SP Interoperabilitás

- IdP: papigw-shibboleth2-idp.xml
- SP: papigw-simplesaml-saml2-sp.xml

SAML2.0 Single Sign on

HTTP-Post

- Működik
- A SimpleSAMLphp nem támogatja a NameID titkosítását, csak az egész assertion titkosítását (FIXME)
  - ezért be kell állítani hogy a NameID-t ne titkosítsa az IdP, ugyanígy kényszeríteni kell az Attribute Push használatát is

```
<RelyingParty id="https://papigw.aai.niif.hu/simplesaml"
   provider="https://papigw.aai.niif.hu/idp/shibboleth"
   defaultSigningCredentialRef="IdPCredential">
   <ProfileConfiguration xsi:type="saml:SAML2SSOProfile"
      encryptNameIds="never" includeAttributeStatement="true"/>
</RelyingParty>
```

- A SimpleSAMLphp SP metaadatába kézzel kell beletenni az aláíró és titkosító publikus kulcsokat, mivel a kiexportált metaadat ezeket nem tartalmazza (ráadásul a php-s konfigurációban szereplő certificate/privatekey paramétereket nem lehet abszolut elérési úttal hivatkozni, mindenképp a cert/ könyvtárban kell lenniük)

HTTP-Artifact

- A SimpleSAMLphp nem támogatja a HTTP-Artifact bindingot (és általában a SOAP-ot használó bindingokat)

Attribute push

- Működik

Attribute pull

- A SimpleSAMLphp nem támogatja az AttributeRequest protokollt (a SOAP binding miatt)

NameIDFormat

- Általában urn:oasis:names:tc:SAML:2.0:nameid-format:transient

SAML2.0 Single Log out

- A SimpleSAMLphp támogatná az SLO-t, de a shibboleth2 IdP nem. A metaadatnál panaszkodik is hogy a Shibboleth metadata nem tartalmaz SingleLogoutService-t.

AAIInterop-OpenSSOShib2

> **Elavult információ**
>
> **Figyelem**: ez a szakasz vagy szócikk elavult információkat tartalmazhat!

OpenSSO IdP - Shibboleth2 SP Interoperabilitás

- IdP: maszat-opensso-idp.xml
- SP: papigw-shibboleth2-sp.xml{.download="papigw-shibboleth2-sp.xml"}

SAML2.0 Single Sign on

- SP oldali SAML2 bindingot támogató AttributeConsumerService-ek:
  - 1: /SAML2/POST urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
  - 2: /SAML2/POST-SimpleSign urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign
  - 3: /SAML2/Artifact urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact
  - 4: /SAML2/ECP urn:oasis:names:tc:SAML:2.0:bindings:PAOS

HTTP-Post

- Működik
- https://papigw.aai.niif.hu/saml2interop/opensso-post/
- SP oldalon

```
<SessionInitiator type="Chaining" Location="/Login" id="maszat-opensso-post"
    relayState="cookie" entityID="https://idp.sch.bme.hu/niif-teszt">
    <SessionInitiator type#"SAML2" defaultACSIndex="1" template="bindingTemplate.html"/>
</SessionInitiator>
```

HTTP-Artifact

- Az OpenSSO nem figyel arra hogy az SP milyen AttributeConsumerService-t kér, így az IDP oldalon kell konfigurálni az SP tulajdonságait úgy, hogy az alapbeállítás ne a POST legyen.
- TODO - Trust management a back-channel kommunikációnál a tanusítványt ismernie kell az SP-nek.
- JVM beállítása a kliens tanúsítvány használatához - https://opensso.dev.java.net/issues/show_bug.cgi?id=1409

Attribute push

- Gyári buildekkel nem működik, ugyanis az OpenSSO urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified formátumban küldi az attribútumokat, amiket a Shibboleth2 SP nem fogad el.
- Saját patch használata esetén bekonfigurálható az uri típusú NameFormat is, azzal működik probléma nélkül.
  - https://opensso.dev.java.net/issues/show_bug.cgi?id=2775

Attribute pull

- Az OpenSSO-ban nem lehet kikapcsolni az attribute push-t. FIXME

NameIDFormat

- A Shibboleth2 SP által generált metaadatba kézzel be kell illeszteni a NameIDFormat node-ot
- Az OpenSSO támogatja a perzisztens és a tranziens SAML2 azonosítót is.

SAML2.0 Single Log out

- A shibboleth2 nem támogatja (még) a Single Log-out protokollt, lásd: https://wiki.shibboleth.net/confluence/display/SHIB2/SLOIssues

Metaadat problémák

- A Shibboleth2 SP metaadatból el kell távolítani az `<md:Extensions>` node-ot az összes gyerekelemével együtt.
  - Erre nagyon figyelni kell, mert összeomlik tőle az OpenSSO, és csak címtár-módosítással ('hibás' metaadat törlése) állítható helyre.
  - Sajnos nem triviális javítani ezt a viselkedést...

- A metaadatba ágyazott certificate esetén csak a `<ds:X509Certificate>` node szerepelhet, semmi más
  - Ehhez írtam patch-et ami ezt javítja.
  - https://opensso.dev.java.net/issues/show_bug.cgi?id=2985

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

El?készületek

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.

T?zfal

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

Shibboleth IdP

Elavult információ

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

- Shibboleth2_IdP
- Shibboleth2_SP

Telepítési leírások

- Shibboleth IdP telepítés (Debian)

Konfigurációs leírások

- IdP alkalmazás konfigurációja
- Attribútumok használata
- Attribútum kiadás konfigurációja
- Felhasználó azonosítás

- Új IdP hozzáadása a föderációhoz

Resource Registry

> **Elavult információ**
>
> **Figyelem:** ez a szócikk elavult, a Resource Registry megújult egy ideje!

Alapfogalmak

- **Attribútum**: felhasználóra vonatkozó tulajdonság. A föderációban használt attribútumok listája **itt érhető el**.
- **SP: Service Provider - Szogáltatás**: Webes alkalmazás, amelynek felhasználóit föderatívan valamilyen IdP által autentikáltatja
- **IdP: Identity Provider - Azonosító szervezet**: Feladata a felhasználó azonosítása, felhasználó attribútumainak kiadása SP-k részére
- **Föderáció**: olyan intézmények halmaza, amelyek között lehetséges az azonosítási-információk átadása. Az intézmények - szabályozott keretek között - megbíznak a másik intézmény által kiállított azonosítási-információkban.
- **Entitás**: föderációt alkotó elem (IdP, SP)

Áttekintés

A Resource Registry az alakuló magyarországi felsőoktatási és kutatási föderáció (HREF) központi eleme, mellyel a föderációban résztvevő szolgáltatások (SP-k) és azonosító szervezetek (IdP-k) adminisztrálását lehet egy letisztult környezetben, elosztott jogosultságokkal végezni. A rendszer az egyes entitásokért felelős adminisztrátorok számára készül.

A rendszer a svájci SWITCH Intézet által fejlesztett rendszer alapjaira épül, PHP nyelven íródott, adatbázisként MySQL-t használ.

Funkciók

A rendszer saját adatbázisból dolgozik, minden funkciójának kimenete ezeken az adatokon alapul.

- Föderáció-szintű feladata, hogy a központi metaadatot óránként generálja, melyet a résztvevő "entitások" használnak, ezzel garantálva egyfelől a föderáció egységességét, másfelől a megfelelő formátumú metaadat alkalmazásával megteremtse a lehetőséget, hogy a föderáció kiegészítő alkalmazásai (Discovery Service), ill. nemzetközi szintű együttműködésben - bizonyos keretek közt - más föderációk is dolgozhassanak ebből.

- Az attribútum-szabályzat szintén föderációs szinten állítható, melyeket kiegészítve, az egyes IdP-k megadhatják, hogy mely attribútumokat milyen feltételekkel adják ki, ill. az egyes SP-k is deklarálhatják, hogy milyen attribútumok megléte esetén tudnak egyáltalán működni, mindezt az egyes intézmények adatvédelmi felelősei által kontrolálva.
- Egyénileg, az adott entitás adminisztrátorai által használhatók az egyes IdP-k, SP-k telepítését és konfigurálását megkönnyítő funkciók, melyek a megfelelő beállításokat webes felületen megadva letölthetővé teszik az ezen beállítások alapján automatikusan generált, és jó eséllyel minimális további kézi konfigurációt igénylő fájlokat. Fontos, hogy ezeket a fájlokat nem kötelező használni, ám segítséget jelenthetnek.

Shibboleth 2.x SP-hez:

- `shibboleth2.xml`

Fontos, hogy ezt akkor lehet szinte egy az egyben használni, amennyiben az adott SP csak egy alkalmazást véd. Amennyiben több alkalmazás is igényel Shibbolethet egyazon hoszton, úgy kézzel kell szerkeszteni az xml-t.

- `attribute-map.xml`
- `attribure-policy.xml`

Ez a két fájl egy az egyben használható letöltés után, további konfigurációt alapesetben nem igényel.

Shibboleth 2.x IdP-hez:

- `attribute-resolver.xml`

Ez csak egy keret fájl, a legtöbb olyan elem szerepel benne, amelyeket a helyi viszonyokra szabva már működhet az IdP, ám itt muszáj kézzel is szerkeszteni, pl. LDAP adatok...

- `attribure-filter.xml`

A fájl egyből használható - további információk a fájl előállításának menetéről.

SimpleSamlPHP-hoz:

- `AttributeFilter.xml`

A fájl egyből használható - további információk a fájl előállításának menetéről.

Bejelentkezés a rendszerbe

A Resource Registry a https://rr.eduid.hu címen érhető el, és bejelentkezni csak föderatív azonosítás után lehet. A nyitóképernyőn a bejelentkezési lehetőségen túl mindössze általános,

nyilvános információk érhetők el a föderáció aktuális állapotával kapcsolatban, ill. a rendszer használatához található segítség.

A rendszerbe történő bejelentkezéshez elengedhetetlen, hogy a felhasználót azonosító IdP az alábbi attribútumokat átadja a Resource Registry-nek.

- eduPersonPrincipalName
- schacHomeOrganizationType
- eduPersonScopedAffiliation
- email - ez belépés után, manuálisan is beállítható

Szerepek a rendszerben

A Resource Registrybe csak föderatív azonosítás után lehet belépni.

Felhasználó

- Lehetősége van rá, hogy a föderációhoz szolgáltatást (SP-t) regisztráljon, amely jóváhagyás után élesedhet.

SP adminisztrátor

- Ő felelős egy, vagy több, már jóváhagyott SP-ért, ill. elbírálhat felhasználói SP ajánlásokat.

IdP adminisztrátor

- Az általa regisztrált és karbantartott IdP-ért felelős.

RR adminisztrátor

- A Resource Registry-n belül tevékenykedő felhasználók jogosultságaiért felelős, ő adhat hozzá egyes entitásokhoz újabb adminisztrátorokat, ill. bírálhat el IdP-ket, és SP-ket.

Alapértelmezés szerint, aki be tud lépni, a legegyszerűbb felhasználói jogosultságokat kapja, bármilyen magasabb szintű szerepkört RR adminisztrátor delegálhat számára, a magasabb jogkörrel járó felelősségi kört pontosan körülhatárolva. (Pl. A RR adminsiztrátor a felhasználót az őt azonosító IdP adminisztrátorává tehet, amely nyomán a felhasználó pontosan ezt az egy IdP-t hangolhatja, de felhasználói jogosultságokat már nem oszthat tovább)

Fontos, hogy az RR-ben az egyes szerepek egy-egy intézmény adminisztrálásához köthetők, ily módon megvalósítva az elosztott jogosultságkezelést. Egy példán keresztül bemutatva ez azt jelenti, hogy egy felhasználó ha szeretne SP-t felvenni a föderációba, akkor azt csak az őt azonosító intézmény hatókörébe teheti meg, ami azt is jelenti, hogy az adott intézményhez tartozó, RR adminisztrátor jogosultsággal rendelkező felhasználó hagyhatja ezt a regisztrációs-, vagy módosítási kérelmet jóvá. Ugyanez az adminisztrátor csak a saját intézményéhez tartozó

entitásokra vonatkozóan oszthat, vagy vonhat vissza jogosultságokat.

Természetesen létezik Power User, aki mindent lát, mindenhez van jogosultsága, de csak nem várt esemény esetén aktivizálódik valahol az NIIF AAI környékén :), amúgy rendeltetésszerű működés esetén a szubszidiaritás elvét képviselve az intézményeké az őket érintő ügyekben a döntési jog.

Folyamatok

SP regisztráció

Bárki, akit a rendszer föderatív azonosítással beléptetett, kezdeményezheti egy SP föderációba történő felvételét, ehhez az „SP adminisztráció“ oldalon az „Új SP regisztrálása“ c. menüpontot kell választani. Varázsló segít a regisztrációban, melynek mindössze a telepített SP metaadatának nyilvánosan elérhető url-jét kell megadni (alapértelmezés szerint: https://#HOSTNAME#/Shibboleth.sso/Metadata), majd az automata a lehető legtöbb beállítási paramétert megpróbálja kiolvasni az xml-ből, és egyből beírni az adatbázisba az új SP adatai közé. Mivel minden adatot nem lehetséges az alapértelmezett metaadatból kinyerni, így a regisztráló felhasználónak néhány további adatot kell megadnia ahhoz, hogy véglegesíthesse az SP regisztrációs kérelmet. Ezeket hat csoportra lehet osztani.

- **Alapinformációk**: itt kerülnek megadásra az alapvető, leíró információk, melyek az SP nevét, leírását tartalmazzák, ill. a legfontosabb azonosító, az entityID. Az adatok egy része (pl: entityID) kiderül már a metaadatból is, így a beviteli mezőt már az automata kitöltötte.
- Itt tudjuk meghatározni első körben azt is, hogy az adott SP nyilvános, vagy belső SP legyen. Ennek szellemében kell a megadott feltételes mezőket kitöltenünk. Ha belső SP, akkor csak a legszükségesebb adatok megadása elvárt.
- **Kapcsolattartók**: ha a metaadatból nem derül ki, akkor kézzel kell megadni technikai, adminisztratív, általános...stb. kapcsolattartó személyeket, akik adatai a központi metaadatban is szerepelni fognak.
- **SP Service Locations**: különböző bindingok elérhetőségei – ezt az automata az esetek nagy hányadában jól kiolvassa a metaadatból, emberi módosítást a legritkább esetben igényel. Kivételt képez a *NameIdFormat* meghatározása, mely kapcsán három opció közül választhatunk.
  - Tranziens opciót kell választanunk, ha SP-nk számára nem fontos, hogy ki a felhasználó, hiszen nem ez alapján dől el, hogy milyen erőforrásokat érhet el, hanem az alapján hogy milyen, a felhasználóra vonatkozó, pl. eduPersonScopedAffiliation attribútumot állnak az SP rendelkezésére.
  - Perzisztens opciót kell választanunk, ha SP-nk számára fontos, hogy ki a felhasználó ÉS sz SP által védett alkalmazásaink is felkészültek arra, hogy persistent-idt fogadjanak, ezzel dolgozzanak.
  - Nem meghatározott opciót kell választanunk, amennyiben az SP által védendő alkalmazás mind persistent, mind transient NameID fogadására alkalmas.

- *Megjegyzés* Amennyiben most alakítjuk ki az AAI infrastruktúránkat, újonnan állítjuk be az SP-t annak érdekében, hogy valamilyen alkalmazást védjen, akkor *ajánlott*, hogy támogassa a perzisztens azonosítók használatát.
- **Tanúsítványok**: az SP által használt tanúsítványokat kell PEM formátumban megadni – ehhez is segítséget nyújt varázsló helyben, amelynek az SP metadatájának URL-jét címét kell megadni, ami után az automata beolvassa a tanúsítvány(oka)t.
- **Kötelező attribútumok**: itt lehetséges azon attribútumok megadása, melyek kiadása elvárt az IdP-től, amelyek nélkül az SP által védett alkalmazás nem használható.
- Amennyiben egy attribútumot megkövetel az SP használatához, az azt jelenti, hogy az attribútum kiadása nélkül az SP nem lesz használható. Ha egy attribútumot ajánlottnak jelöl, akkor az IdP kiadja, amennyiben implementálta, ám az SP-nek e nélkül is működnie kell. Ha olyan attribútumot jelöl kötelezőnek, amely a föderációs szabály szerint csak ajánlott, vagy opcionális, úgy könnyen előfordulhat, hogy az IdP nem implementálta, így nem is tudja kiadni, aminek következtében a felhasználó nem fogha tudni használni az Ön SP-jét.
- Ideális esetben nincs szükség külön szabályzásra, amennyiben mégis, úgy törekedjen rá, hogy minél kevesebb attribútumot szabályozzon külön!
- **Hallgatóság**: megadhatók, milyen jellegű IdP-k érhetik el az adott SP-t, ill. amennyiben ez a szabályzás nem lenne elegendő, úgy egyesével is megadhatók IdP-k aszerint, hogy felhasználói használhatják-e az SP-t, vagy sem.
- Amennyiben belső SP-t regisztráltunk, itt állíthatjuk be, hogy minden intézmény jelleget tiltunk, és egy kivételt megadunk: a saját IdP-nket. Ily módon más IdP nem is fog tudni erről az SP-ről, tehát annak ellenére, hogy szerepel a föderációs metaadatban, csak belső használatra lesz alkalmas.

SP módosítás

Egy jóváhagyott SP-t csak a megfelelő jogosultsággal rendelkező felhasználó tud módosítani. Általában egy-egy SP-hez tartozó adminisztrációs jogot az intézmény *RR adminisztrátor* jogkörrel rendelkező felhasználója osztja ki az adott SP jóváhagyásával egy ütemben.

A módosítás folyamata teljesen analóg a regisztrációéval, ami a funkciókat illeti.

FONTOS: Akár regisztráltunk, akár módosítottunk, a változásokat jóvá kell hagynia az Önt azonosító intézet RR adminisztrátorai közül valakinek. Amíg ezek a változtatások bekerülnek a föderációs metaadatba, az legfejjebb egy óra, ám amíg minden föderációs entitás frissíti a metaadatot, így értesül a változásról.

IdP regisztráció

A föderációba új IdP-t - mivel a regisztrálandó IdP-t üzemeltető kolléga még nem tud belépni a Resource Registry-be - egy, a föderációs adminisztrációért felelős kolléga tud regisztrálni. Ehhez szükséges, hogy az IdP, minden kapcsolódó programmal együtt telepítve legyen, és az alapértelmezett, telepítéskor generált metaadat egy meghatározott url-en elérhető legyen.

A telepítéshez - *Shibboleth esetében* - pl. a Shib2IdpInstall wikilapon található leírás szolgál segítségül. Attribútumokat, autentikációt konfigurálni nem kell, elegendő, ha a Teszt pontnál látjuk a megnyugtató `ok` szöveget, és minimális beállításokat megejtettük

Amennyiben ez működik, úgy írni kell egy e-mailt az aai@niif.hu címre, valaki a föderációs adminisztrátorok közül regisztrálja az IdP-t, és a válasz e-mailben elküld két linket, amelyek tartalmaznak két linket az `attribute-resolver.xml` és `attribute-filter.xml` már testreszabott konfigurációs fájlokra mutatva. Ezeket letöltve, bemásolva az IdP-nek már működnie kell alapszinten, így már lehetségessé válik a Resource Registry-be történő belépés. Sikeres belépés után az intézményhez tartozó RR jogosultságokat átadjuk, s a továbbiakban mehet minden a maga utáján, intézményi szinten.

Nagyon fontos, hogy az IdP-n bármilyen módosítás **azonnal érvénybe lép**, így rossz beállítás esetén akár az IdP által hitelesíthető felhasználók belépése is ellehetetlenülhet.

A beállítási lehetőségek az alábbiak

- **Alapinformációk**: IdP neve, leírása, jellege – technikai ismereteket nem igénylő, leíró jellegű információk
- **Technikai információk**: EntityID, és különböző bindingok elérhetőségei – ezt az automata az esetek nagy hányadában jól kiolvassa a metaadatból, emberi módosítást a legritkább esetben igényel.
- **Tanúsítványok**: az IdP által használt tanúsítványokat kell PEM formátumban megadni – ehhez is segítséget nyújt varázsló helyben, amelynek a webszerver címét kell megadni, ami után az automata beolvassa a tanúsítvány(oka)t.
- **Kapcsolattartók**: ha a metaadatból nem derül ki, akkor kézzel kell megadni technikai, adminisztratív, általános...stb. kapcsolattartó személyeket, akik adatai a központi metaadatban is szerepelni fognak.

A további négy beállítás némi hozzáértést igényel, lévén az alapértelmezett metaadatból nem olvashatók ki. A rendszer ezeket az értékeket a föderációs szabályoknak, megállapodásoknak megfelelően készíti elő, legtöbb esetben nincs szükség módosításra, ám ha mégis, bármilyen speciális igény okán, akkor nagy odafigyeléssel kell beállítani.

- **Támogatott attribútumok**: beállítandó, hogy az IdP mely attribútumokat, milyen formában támogatja. A föderációs szinten kötelezőket mindenképp támogatnia kell.
- **Általános attribútum kiadási szabályok**: beállítható, hogy amennyiben egy SP az adott attribútumot kötelezően, ill. opcionálisan kiadandóként kéri, akkor az IdP hogyan viselkedjen. Ezek a szabályok általánosak, minden, a föderációban résztvevő SP-r vonatkoznak.
- **Speciális attribútum kiadási szabályok**: beállíthatók külön-külön egyes SP-kkel való viselkedés, amennyiben indokolt az általános szabályoktól való eltérés.
- **Telepítési és környezeti információk**: leíró információk, amelyek pl. hiba esetén segítséget adnak a hiba elhárítójának, hogy milyen rendszerrel lesz dolga. Emellett statisztikai célokat is szolgál.

Attribútumok kezelése

A föderációban használható attribútumok részletes listája a föderációs attribútum specifikációban található.

Az egyes attribútumokkal kapcsolatban négy irányból lehetséges beállításokat eszközölni

- Minden SP meghatározhatja, hogy mely attribútumok kiadását követeli meg, és melyek kiadását ajánlja (SP beállítások - Kötelező attribútumok menüpont)
- Minden SP meghatározhatja, hogy mely IdP-ktől hajlandó attribútumokat elfogadni (SP beállítások - Hallgatóság menüpont)
- Minden IdP meghatározhatja általánosságban, hogy ha egy SP tőle egy bizonyos attribútum kiadását megköveteli, vagy ajánlja, akkor azt az attribútumot kiadja-e, vagy sem. (IdP beállítások - Általános attribútum kiadási szabályok menüpont)
- Minden IdP meghatározhat SP-specifikus szabályokat, tehát egy-egy SP-re, vagy egy-egy SP egy-egy attribútumára vonatkozólag megadhat az általános beállításaitól eltérő szabályokat - pl. az eduPersonPrincipalName-t ha általában ajánlva kérik az SP-k, akkor kiadja, de XY SP-nek semmiképp nem adja ki. (IdP beállítások - Egyedi attribútum kiadási szabályok menüpont)

További információ az attribútumok implementációjáról, kapcsolódó fogalmakról **A fenti beállítások eredőjeként generálódik az IdP-k által használandó XML alapú attirbútum filter fájl**

A generált attribútum filter alapvetően tiltó jellegű, tehát az IdP pontosan azokat az attribútumokat és pontosan azoknak az SP-knek adhatja ki, melyek megadásra kerültek a beállításoknál, egyébként semmit.

Néhány példa a "leképződésre"

- Ha egy SP kiadásra ajánl egy attribútumot, de az IdP (akár az általános-, akár az egyedi attribútum kiadási szabálya miatt) nem adja ki azt, akkor az XML fájlban komment formájában jelenik meg, hogy ezt és ezt az attribútumot igényelné az SP, de nem kerül kiadásra
- Ha egy SP hallgatóságából kitilt egy IdP-t, akkor az IdP attribútum filterében az adott SP-re vonatkozólag nem jelenik meg semmi, amelynek következtében nem is kerül számára kiadásra semmi

XML alapú filter

- Shibboleth IdP-nél: `attribute-filter.xml`
- simpleSAMLphp IdP-nél: `AttributeFilter.xml`

A filter fájlok Resource Registry által előállított változatát használni nem kötelező, de fokozottan ajánlott, hiszen ezzel garantálható egyfelől, hogy az IdP-n keresztül autentikáló felhasználók azokat az SP-ket, melyeket el kell tudniuk érni, jól fogják, megfelelő attribútumokkal "a zsebükben" fogják tudni elérni, másfelől így tudnak érvényesülni a feljebb részletezett korlátozó szabályzások.

Adatvédelmi szempontok

Ha egy SP megváltoztatja attribútum igényeit pozitív irányba (új attribútumokat kér), úgy a változtatás csak akkor fog belekerülni az IdP-k attirbútum filterébe, amennyiben ezt a változtatást tudomásul veszi az IdP oldaláról az illetékes adatvédelmi felelős. Amennyiben egy SP-nél ilyen jellegű változás történik, a rendszer e-mailben értesíti az érintett IdP-k gazdáit, adatvédelmi felelőseit.

Gyakorlati ajánlás

Kihasználandó a Resource Registry által biztosított lehetőségeket ajánljuk, hogy az IdP-hez tartozó generált attribútum filter fájlt automatikusan töltsék le az IdP-k, bizonyos időközönként (óránként, naponta párszor...), hiszen ezekbe csak úgy kerülhet változtatás, ha azt az IdP adatvédelmi felelőse jóváhagyta, akkor viszont egyből átvezetődik a változtatás, nem szükséges kézzel letölteni, ill. újraindítani az IdP-t. (Shibboleth esetében be kell állítani egy kapcsolót, SSP-nél automatikusan újratölti a friss XML-t)

A filter elérhetősége

- Shibboleth IdP: https://rr.eduid.hu/gen_attribute-filter.php/href/IDP_NEVE/attribute-filter.xml
- simpleSAMLphp IdP: https://rr.eduid.hu/gen_attribute-filter-ssp.php/href/IDP_NEVE/attribute-filter.xml

Útmutató a beállításhoz

- Shibboleth
- simpleSAMLphp

UApprove

> **Warning**
>
> Ez jelen formájában egy elavult lap, hamarosan frissítésre kerül

uApprove

Felépítés

Az uApprove a SWITCH.ch által fejlesztett alkalmazás, ami a Shibboleth2 IdP-vel együttműködve képes a felhasználótól attribútum-kiadás hozzájárulást kérni.

A uApprove két részből áll. Egy szervlet filterből (IdP plugin), ami a Shibboleth2 IdP webalkalmazásba beépülve elkapja és elemzi a kéréseket, illetve egy különálló webalkalmazásból (Viewer), ami a felhasználói hozzájárulást kéri el.

A felhasználói hozzájárulásnak két szintje van: a globális felhasználási feltételek elfogadása, illetve minden SP esetén egy ún. digitális identitás elfogadása. Ez utóbbi felület lehetőséget ad a felhasználó számára hogy lássa az adott SP felé kiadandó attribútumait, és beleegyezzen azok kiadásába.

A hozzájárulások XML fájlban vagy relációs adatbázisban is tárolhatók. A uApprove lehetőséget ad arra, hogy az SP-hez történő hozzáféréseket naplózza, az attribútum-kiadástól függetlenül (tehát elég az IdP plugin komponenst használni ahhoz hogy a naplózás működjön - ezt Monitoring módnak hívjuk).

uApprove viewer webalkalmazás telepítése

```
cd uApprove-2.1.4/
unzip viewer-2.1.4-bin.zip
sudo cp viewer-2.1.4/conf-template/* /path/to/uApprove/uApprove
sudo cp -r viewer-2.1.4/webapp /path/to/tomcat/webapps/uApprove
```

A uApprove közös konfiguráció a common.properties fájlban található:

```
storageType = Database
databaseConfig = /path/to/uApprove/database.properties
#storageType = File
#flatFile = /path/to/uApprove/uApprove-log.xml

# A globális felhasználási feltételeket tároló fájl
# Ebben az XML-ben több verzió is tárolható,
# verzióváltás esetén a felhasználónak újra el kell fogadnia.
termsOfUse = /path/to/uApprove/terms-of-use.xml

# Kommunikáció titkosításához
SharedSecret = some-very-random-string
```

database.properties (a támogatott RDBMS a MySQL):

```
driver = com.mysql.jdbc.Driver
url = jdbc:mysql://localhost:3306/*dbname*
user = *username*
password = *password*
sqlCommands = /path/to/ArpViewer/mysql.commands
```

viewer.properties:

```
# amennyiben a böngésző nem küld lokalizációs információt, ezen beállítás jut érvényre
useLocale = HU_hu
# felajánljuk-e a felhasználónak az sp-től független beleegyezést
globalConsentPossible=true
loggingConfig = /path/to/uApprove/logging.xml
```

attribute-list: az attribútumok sorrendezését és egyes nem kívánt attribútumok (pl transientid) elrejtését állíthatjuk be benne. Az attribútumnevek a Shibboleth2 attribute-resolver.xml -ben deklaráltaknak kell megfeleljenek.

web.xml (/path/to/tomcat/webapps/uApprove/WEB-INF):

```
 <context-param>
  <param-name>Config</param-name>
  <param-value>
  /path/to/uApprove/viewer.properties;
```

```
  /path/to/uApprove/common.properties;
  </param-value>
 </context-param>
```

IdP plugin beállítása

Csomagoljuk ki a plugint és másoljuk a konfigurációt illetve a szükséges library fájlokat a helyükre

```
cd uApprove-2.1.4/
unzip idp-plugin-2.1.4-bin.zip
sudo cp idp-plugin-2.1.4/conf-template/* /path/to/uApprove
sudo cp idp-plugin-2.1.4/lib/* /path/to/idp-setup/src/main/webapp/WEB-INF/lib/
```

Ezután szerkesszük az IdP web.xml konfigurációját

```
<web-app>
  ...

  <filter>
    <filter-name>uApprove IdP plugin</filter-name>
    <filter-class>ch.SWITCH.aai.uApprove.idpplugin.Plugin</filter-class>
    <init-param>
      <param-name>Config</param-name>
      <param-value>
        /path/to/uApprove/idp-plugin.properties;
        /path/to/uApprove/common.properties;
      </param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>uApprove IdP plugin</filter-name>
    <url-pattern>/profile/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
  </filter-mapping>

</web-app>
```

Ezután a Shibboleth IdP setup.sh szkriptjével készítsük el a webarchívumot (idp.war), amit telepítsünk újra.

A uApprove IdP plugin egyedi beállításai az idp-plugin.properties-ben találhatóak:

```
# Azon SP-hez amihez nem akarunk ArpFiltert használni
spBlacklist = /path/to/config/sp-blacklist
# SP logolás
LogProviderAccess = false
# Csak SP logolás
MonitoringOnly = false
# ArpViewer alkalmazás helye
uApproveViewer = https://idp.example.org/uApprove/Controller
# Támogassuk-e a passzív authentikációt
isPassiveSupport = false
```

Attribútumnevek beállítása

Az attribútumnevek helyes megjelenítéséhez az ArpViewer a Shibboleth2 IdP attribute-resolver.xml fájlt használja. Ebben a fájlban kell beállítani a lokalizált attribútumneveket a következőképpen:

Shibboleth2 IdP conf/attribute-resolver.xml:

```
 <resolver:AttributeDefinition id="postalAddress"
xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
sourceAttributeID="postalAddress">

<resolver:Dependency ref="myLDAP" />
<resolver:DisplayName xml:lang="en">Business postal address</resolver:DisplayName>
<resolver:DisplayName xml:lang="hu">Hivatalos postai cím</resolver:DisplayName>
<resolver:DisplayDescription xml:lang="en">Business postal address: Campus or office
address</resolver:DisplayDescription>
<resolver:DisplayDescription xml:lang="hu">Az intézmény hivatalos postai
címe</resolver:DisplayDescription>
```

Megjegyzés: a Resource_Registry által előállított attribute-resolver.xml template fájl tartalmazza az attribútumok magyar nyelvű leírásait.

ArpFilter before the profile servlet - support for other authentication modules

Lásd: ArpFilterProposal (angol)