Kiegészít? tudnivalók

- Intézmény átnevezés
- Erasmusplus ESI
- Tanúsítványok a föderációban
- Interoperabilitás mátrix
- AA Testing
- EduIDTest
- Publikációk
- SSP EntityCategories
- EntraID in eduID.hu
- AAI AzureADasAuthsource
- SamlSign

Intézmény átnevezés

Sok esetben merül fel az a kérdés, hogy eduID tagintézmény névváltozása esetén mi a teendő. Alább egy konkrét levelezés másolata következik, amely remélhetőleg általános információkat tartalmaz.

Kérdés

> Kecskemét és Szolnok egyesítésével létrejött a Pallasz Athéné Egyetem. Mit kell ilyenkor tennünk? Mert a weboldalakat át kéne neveznem az új dominre. Kell csinálnom új IDP-t és új SP-ket? vagy csak SP-ket?

Szerz?dés

Ha a PAE jogutódja a KeFo-nak, akkor nem kell új szerződést kötni. Egyébként igen.

Átnevezni vagy nem átnevezni?

entityID

Az entitások átnevezése problémás, ezért azt célszerű kerülni. A legtöbb gond az eduPersonTargetedId-vel van, ugyanis az "targeted" azonosító, ami azt jelenti, hogy akár az IdP, akár az SP entityID-je megváltozik, az azonosító értéke is megváltozik, azaz a felhasználó "élettörténete" újraindul az SP-nél. Ráadásul az azonosítók kézi átállítása is meglehetősen körülményes. Kérdés, hogy mely általatok használt SP-k használnak eduPersonTargetedId-t. A Videotorium például igen, ill. lásd még a href.xml-ben a RequestedAttributes elemet. Mivel az entityID nem jelenik meg a felhasználóknak (normális esetben), emiatt nem valószínű, hogy bárki arra kötelezne titeket, hogy változtassátok meg.

További probléma az entityID megváltoztatásával, hogy a kereskedelmi adatbázis-szolgáltatók jellemzően kézzel tartják karban azt a listát, hogy mely intézmények számára elérhető a szolgáltatás, és az entityID változtatása miatt az intézményünk kikerülhet ezekből a listákból.

Scope

Az eduPersonPrincipalName és az eduPersonScopedAffiliation használja a scope-okat. Az minden további nélkül működik, hogy újabb scope-okat **IS** elkezdjetek használni (új felhasználóknál), de ha egy létező felhasználó scope-ját lecserélitek, akkor ő a fentihez hasonló módon új felhasználóként jelenik meg az összes SP-nél. És eduPersonPrincipalname-et és/vagy eduPersonScopedAffiliationt nagyon sok SP használ, valamint a legtöbb belső SP is valószínűleg úgy van konfigurálva, hogy scope alapon végezze az autorizációt. Az autorizációs szabályok (pl. `require affiliation member@kefo.hu` vagy `require eppn bekre.pal@kefo.hu`) szabályok módosítása triviális. Az élettörténet megtartása sajnos már alkalmazásspecifikus, jellemzően kézi adatbázisműveleteket igényel. Annyival egyszerűbb ez, mint az eduPersonTargetedId módosítása, hogy pontosan tudod az átírási szabályt, pl:

```
s/([^@]+)@kefo.hu/$1@pae.hu/
```

A tanácsom az, hogy nézzétek át, hogy mely SP-k esetén fontos az élettörténet megtartása, és ezeket kérjétek meg a fenti változtatásra. (Pl. ha vannak olyan HBONE-adminisztrációs szolgáltatások, amiket eddig kefo.hu-s azonosítóval használtál, akkor azok ilyenek.)

mail

A mail cím megváltoztatása jellemzően nem jelent problémát. Azt érdemes észben tartani, hogy a Drupal megköveteli az e-mail címek egyediségét, azaz ha megváltozik az eppn, miközben nem változik a mail, akkor az "új" felhasználó létrehozása nem fog sikerülni.

IdP URL

Mellékhatások nélkül módosítható, hogy az IdP milyen URL-en azonosítsa a felhasználókat (vagy válaszoljon az AttributeQuery-kre, de ez nagyon ritka funkció). Ehhez a `SingleSignOnService` és `SingleLogoutService` végpontokat kell lecserélni a Resource Registry-ben a haladó beállítások között, valamint természetesen az IdP webszerverét megfelelően kell konfigurálni. Ez a módosítás kizárólag azt befolyásolja, hogy milyen URL jelenik meg a felhasználók böngészőjében, amikor azonosításra átirányítják őket az SP-k. Nyilván megfelelő SSL tanúsítvány is kell az új URL-re.

Discovery leírás

Ez is mellékhatások nélkül módosítható, és ez az, ami a leginkább szembetűnő a felhasználók számára. Ezt is a Resource Registry-ben, a Leírók-nál kell módosítani. (Légyszi küldjetek egy értesítést az info @ eduid.hu-ra, ha módosítjátok, mert a központi Discovery Service-t jelenleg még kézzel kell frissítenünk.)

Erasmusplus ESI

Erasmus+ és ESI

Az Európai Hallgatói Kártya Kezdeményezés részeként az Európai Bizottság 2021 júniusától kezdve digitális formában kívánja támogatni az Erasmus programhoz kapcsolódó összes szolgáltatást, mint például az Online Learning Agreement (OLA) és az Erasmus+ mobilalkalmazást. A digitalizálás alapvető része a hallgatók azonosítása, amely az eduGAIN-en keresztül fog történni.Az eduGAIN szövetség hazai képviselője a KIFÜ, aki Magyarországon az eduID.hu szövetséget működteti, hogy az oktatás, kutatás és közgyűjtemények szereplői az intézményi hitelesítő adataikkal bejelentkezhessenek az eduGAIN szolgáltatásokba. Az eduGAIN-ben a szolgáltatások sikeres azonosítás esetén azonnal megkapják a felhasználók helyes azonosításához és jogosultság kezeléséhez szükséges információkat. Az MyAcademicID működéséhez, az európai hallgatói azonosító (European Student Identifier) helyes létrehozásához és az Erasmus+ szolgáltatásokban szükséges attribútumok beállításhoz szükséges információk az alábbiak.

MyAcademicID proxy

Az Erasmus+ szolgáltatásokhoz való hozzáférés a MyAcademicID projektben létrehozott és a GEANT által kezelt proxyn keresztül történik. Ez azt jelenti, hogy az Erasmus+ összes szolgáltatásának egyetlen szolgáltatója van a következő entitásazonosítóval:

> “ https://proxy.prod.erasmus.eduteams.org/metadata/backend.xml

A Szolgáltatót az eduID.hu az eduGAIN-en keresztül teszi közzé, így ha az intézménye tagja az eduGAIN-nek is (lásd: Metaadatok), akkor nem kell további lépéseket tenni. Ellenkező esetben lépjen kapcsolatba először a KIFÜ eduID.hu csapatával, hogy kérje identitásszolgáltatójának exportálását az eduGAIN-ba, majd konfigurálja be az eduID.hu szolgáltatás által terjesztett eduGAIN metaadatokat (lásd: Metadata).

Attributumok

Az Erasmus + szolgáltatások eléréséhez szükséges attribútumok a következők.

Attributumok	Leírás	Példa

eduPersonPrincipalName	Állandó, nem célzott, nem újra kiosztható globálisan egyedi azonosító	`username@foobar.hu`
eduPersonTargetedID	Nem átlátszó, célzott (minden szolgáltatónál más) azonosító, amely nem osztható ki újra	`https://idp.foobar.hu/idp/shibboleth!https://sp.example.com/shibboleth!a1b2c3d4-789a-4ca7-8ff9-43216789bd`
displayName	A felhasználó megjelenítendő neve. **Szükséges, ha nincsen cn, vagy givenName+sn.**	
cn	A felhasználó teljes neve. **Szükséges, ha nincsen displayName vagy givenName+sn.**	`Gipsz Jakab Aladár`
givenName	Felhasználó keresztneve. **Szükséges, ha nincsen displayName vagy cn.**	`Jakab`
sn	Felhasználó családi neve. **Szükséges, ha nincsen displayName vagy cn.**	`Gipsz`
eduPersonAffiliation	Felhasználó és intézmény közti viszony leírása.	`[student](member,)`
eduPersonScopedAffiliation	Felhasználó és intézmény közti viszony leírása. scope-al	`[student@foobar.hu](member@foobar.hu,)`
schacPersonalUniqueCode	A szervezet által hozzárendelt személyes egyedi kód (URN). '''Az európai hallgatói azonosító kódolására és továbbítására szolgál.	`<nowiki>urn:schac:personalUniqueCode:int:esi:foobar.hu:123456789</nowiki>`
schacHomeOrganization	Szervezete domainje.	`foobar.hu`
schacHomeOrganizationType	Szervezet típusa (URN)	`<nowiki>urn:schac:homeOrganizationType:eu:higherEducationalInstitution</nowiki>`

European Student Identifier

Az European Student Identifier (ESI) globálisan egyedi, tartós, nem célzott azonosító (minden szolgáltatónál ugyanaz marad). Titkosítva kerül továbbításra a `schacPersonalUniqueCode` attribútumon keresztül. Az ESI két fő módját támogatott:

1. országos kód:ha rendelkezésre áll nemzeti hallgatói kód
2. intézményenként:jelenleg ez támogatható Magyarországi szinten

Az ESI három részből áll:

- változatlan URN névtér: `<nowiki>urn:schac:personalUniqueCode:int:esi:</nowiki>:`
- szervezet domainje: foobar.hu
- hallgatói azonosító kód (pl. belső nyilvántartási szám): 123456789

Az így kapott teljes ESI: `<nowiki>urn:schac:personalUniqueCode:int:esi:foo.bar:123456789</nowiki>`

Az ESI teljes specifikációja elérhető a GEANT wiki-n:

https://wiki.geant.org/display/SM/European+Student+Identifier

EWP adminisztrátori funkció

Az eduGAIN közösség 2022 őszén definiálta **EWP Admin szerepkört**. EWP Admin szerepkör (Erasmus Without Paper Administrator szerepkör) lehetővé teszi az Erasmus+ tevékenységeiben részt vevő felsőoktatási intézmények (HEI) felhatalmazott képviselői számára, hogy szabványos módon bejelentkezzenek az EWP-eszközökbe és EWP-információik és -beállításaik kezelése egységes legyen. Erről bővebb információ itt található:

https://wiki.geant.org/display/SM/EWP+Admin+Role

További információk

https://wiki.geant.org/display/SM/Identifier+in+Erasmus+mobility

Tanúsítványok a föderációban

SAML2 föderációkban az entitásoknak ismerniük kell egymás publikus kulcsát (amelyet X.509 tanúsítványokban osztanak meg egymással) ahhoz, hogy biztonságosan kommunikálhassanak egymással. Eközben a felhasználókkal is interakcióba lépnek, ami miatt könnyű összekeverni a két fajta tanúsítványt:

1. amit az IdP/SP a felhasználó felé használ;
2. amit másik entitások (SP/IdP) felé használ.

A **felhasználók felé** olyan tanúsítványt kell használni, amelyben a felhasználók böngészője megbízik. Ez a tanúsítvány nem szerepel a föderációs metadatában, ellenben a webszerver konfigurációjában hivatkozni kell rá. Jellemzően valamilyen jól ismert CA-val (pl. DigiCert vagy letsencrypt) aláírt tanúsítványt, ami azt is jelenti, hogy rendszeresen cserélni kell őket.

A **föderációs metadatában** szereplő tanúsítványt elsősorban a föderációs alkalmazás (Shibboleth , SimpleSAMLphp) konfigurációjában kell megadni, mert ez az, amivel alá tudja írni az általa küldött üzeneteket, illetve dekódolni tudja a fogadott titkosított adatokat. Ez a tanúsítvány lehetőség szerint hosszú (10+ éves) lejáratú, self-signed tanúsítvány legyen.

- A webszerver (Apache, Jetty) konfigurációban csak akkor szerepeljen a föderációs metadatában szereplő tanúsítvány, ha azt szeretnénk, hogy az IdP támogassa az attribútumok back-channel történő letöltését (a Shibboleth IdP ilyen), esetleg ha valamilyen oknál fogva önálló AttributeAuthority-t építünk. Ha valami fut a szabványos https porton (pl. az IdP SSO szolgáltatása vagy egy SP), akkor az AttributeAuthority szolgáltatást nem tehetjük ide, ezért az jellemzően a 8443-as porton szokott figyelni.

Interoperabilitás mátrix

Interoperabilitás mátrix

Tesztelt szoftverek

- Shibboleth 2.0 IdP
  - metadata: papigw-shibboleth2-idp.xml
  - telepítési útvonal: /usr/local/shibboleth-idp-2.0.0
  - protokollok: SAML1.1, Shibboleth1.3, SAML2.0
- Shibboleth 2.0 SP
  - metadata: papigw-shibboleth2-sp.xml
  - telepítés Debian csomagból, konfiguráció /etc/shibboleth/ alatt
  - protokollok: SAML1.1, Shibboleth1.3, SAML2.0
- OpenSSO/FAM 8.0 CVS build - IdP
  - metadata: maszat-opensso-idp.xml
  - host: maszat.sch.bme.hu
  - protokollok: SAML2.0
- simpleSAMLphp (?) - SP
  - entityID: https://papigw.aai.niif.hu/simplesaml
  - protokollok: Shibboleth1.3, SAML2.0

Tesztelt protokollok és bindingok

- SAML2.0 AuthnRequest/AuthnResponse protokoll (Web browser SSO profil)
  - HTTP-GET / HTTP-POST binding
  - HTTP-GET / HTTP-Artifact binding
- SAML2.0 AttributeQuery protokoll
- SAML2.0 Single Logout

SAML2.0 Interoperabilitás mátrix

Jelmagyarázat:

- Single Sign on - AuthnRequest/Response (Attribute push-sal együtt)

- HTTP-POST - SAML2.0 HTTP-Post binding
- HTTP-Artifact - SAML2.0 HTTP-Artifact binding
- Attribute query - SAML2.0 Attribute Query protocol
- Signing / encryption - az Assertion aláírása, aláírt és titkosított Assertion feldolgozása
- Metadata management - mennyire egyszerű megoldani hogy az IdP és az SP ismerje egymást

A zöld-del jelölt funkciók tökéletesen működnek, a narancssárgák nem triviálisan, de működésre bírhatók (ilyenkor mindig van megjegyzés is hozzájuk), a pirossal jelölt funkciók nem működtek. Az áthúzott funkciókat nem implementálja az adott szoftverpáros.

Sure, here's the HTML conversion of the provided MediaWiki table:

	Shibboleth2 SP	OpenSSO SP	simpleSAMLphp SP
Shibboleth2 IdP	**Shib2-Shib2** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management	**Shib2-OpenSSO** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management	**Shib2-SimpleSAMLphp** Single Sign on HTTP-POST ~~HTTP-Artifact~~ ~~Attribute query~~ Signing / encryption Metadata management
OpenSSO IdP	**OpenSSO-Shib2** Single Sign on HTTP-POST HTTP-Artifact ~~Attribute query~~ Signing / encryption Metadata management	**OpenSSO-OpenSSO** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management	**OpenSSO-simpleSAML** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management
simpleSAMLphp IdP	**simpleSAML-Shib2** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management	**simpleSAML-OpenSSO** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management	**simpleSAML-simpleSAML** Single Sign on HTTP-POST HTTP-Artifact Attribute query Signing / encryption Metadata management

AA Testing

The following shell script uses *curl* to query a SAML2 Attribute Authority.

You need a valid principal (eduPersonPrincipalName) and the X.509 credentials of an existing Service Provider to use this script.

Source

```
#!/bin/bash

basedir=$(dirname $0)

# URL of the Attribute Authority
AA_URI="https://hexaa.eduid.hu:8443/simplesaml/module.php/aa/attributeserver.php"

# Testing principal (subject)
Principal="bajnokk@niif.hu"

# HEXAA cert
AACert="$basedir/keys/hexaa.eduid.hu-aa.crt"

# EntityID and credentials of the SP on behalf of which
# the request is made
ReqSP="https://sp.hexaa.eduid.hu/test"
ReqCert="$basedir/keys/test.sp.hexaa.eduid.hu-fed.crt"
ReqKey="$basedir/keys/test.sp.hexaa.eduid.hu-fed.key"



usage () {
        cat <<EOS
Usage: $0 [options]

Options:
  -a uri       Attribute Authority URI. Defaults to '$AA_URI'
  -C certfile  Attribute Authority metadata certificate in PEM format. Defaults to '$AACert'.
```

```
  -p principal Testing principal (user name / subject). Defaults to '$Principal'.
  -s entity    EntityID of the SP on behalf of which the request is made. Defaults to '$ReqSP'
  -k keyfile   Key file in PEM format containing the key of the SP used for the request.
Defaults to '$ReqKey'
  -c certfile  Cert file in PEM format containing the certificate of the SP used for the
request. Defaults to '$ReqCert'
EOS
        exit 3
}

# Get command line arguments
while getopts "a:p:s:k:c:h" opt; do
        case $opt in
                a)
                        AA_URI=$OPTARG
                        ;;
                C)
                        AACert=$OPTARG
                        ;;
                p)
                        Principal=$OPTARG
                        ;;
                s)
                        ReqSP=$OPTARG
                        ;;
                k)
                        ReqKey=$OPTARG
                        ;;
                c)
                        ReqCert=$OPTARG
                        ;;
                h)
                        usage
                        ;;
                \?)
                        usage
                        ;;
        esac
done
```

```
DATE=$(date --utc +%FT%TZ)
ReqID=$(hexdump -n 16 -e '4/4 "%08x" 1 "\n"' /dev/urandom)


read -r -d '' REQ_XML <<EOS
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <samlp:AttributeQuery xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_$ReqID" IssueInstant="$DATE"
Version="2.0">
      <saml:Issuer>$ReqSP</saml:Issuer>
      <saml:Subject>
        <saml:NameID Format="urn:oid:1.3.6.1.4.1.5923.1.1.1.6">$Principal</saml:NameID>
      </saml:Subject>
    </samlp:AttributeQuery>
  </S:Body>
</S:Envelope>
EOS

#debug echo "$REQ_XML"

echo "$REQ_XML" | \
  curl --silent --show-error --cacert $AACert --cert $ReqCert --key $ReqKey \
       --header "Content-Type: text/xml;charset=UTF-8" --data @- $AA_URI
```

Validation of response

Signature validation:

```
xmlsec1 --verify --id-attr:ID "urn:oasis:names:tc:SAML:2.0:protocol:Response" --trusted-pem
$aacert $response 2>/dev/null
```

Content validation:

```
xmllint --xpath "//*['Attribute'](local-name()#bkmrk-)[@Name'$attribute']/*[local-
name()='AttributeValue']/text()" $response
```

EduIDTest

hosts file használata

A hosts file meglehetősen egyszerűen használható eszköz arra, hogy élesben működő szolgáltatás átalakítása során tudjunk. Mivel az eduID-ban elterjedten használt SAML profilokban az üzenetváltás a legtöbb esetben a böngészőn keresztül zajlik, ezért elegendő a böngészőt arra rávenni, hogy a tesztelni kívánt SP-t vagy IdP-t ne a világ által látott helyen, hanem a tesztelni kívánt ponton keresse. Ez a módszer **mind SP, mind IdP tesztelésre használható**.

Néhány tipp:

- A tesztelni kívánt IdP vagy SP föderációs konfigjában megadott tanúsítványok egyezzenek meg az éles rendszerben használt tanúsítvánnyal. Ennek hiányában az aláírás nem lesz valid, illetve az esetlegesen titkosítottan küldött adatokat nem lehet kibontani.
- Ha végül az új gép a régi nevén fog hallgatni, akkor nincs szükség a Resource Registry-ben módosítani az adatokon.
- *Bármely* SP-vel / IdP-vel tesztelhetjük az IdP-nket / SP-nket, hacsak nem rontunk el valamit (vö. első két pont) a távoli fél nem fogja észrevenni, hogy nem az "éles" partnerrel beszél.
- Tesztelésre privát IP cím is megfelelő, feltéve, hogy a saját gépünk eléri azt a tartományt.
- [SP]: HEXXA kapcsolat is tesztelhető a módszerrel, feltéve, hogy a tesztelt SP képes kapcsolatot nyitni a hexaa.eduid.hu 8443-as portjára. A HEXAA kizárólag a használt SSL tanúsítvány alapján azonosítja be a kérdezőt, így ugyanazt a tanúsítványt és entityID-t használva ugyanazt a választ kapjuk a tesztelt és az éles SP-n is.
- [IdP]: Az IdP-nk helyes működését például itt tudjuk ellenőrizni: https://attributes.eduid.hu

SP tesztelés

A szolgáltatásunk SAML integrációjának tesztelésére remekül használható eszköz a **https://samlidp.io**, amellyel pár kattintással készíthetünk magunknak teszt célokra IdP-t, amelynek megadható - akár a föderációban még nem regisztrált - SP metaadata is, így különböző felhasználói profilokkal próbálkozhatunk.

Publikációk

- AAI előadás a 2007-es Networkshopon
- AAI és Shibboleth (HBONE Workshop)

SSP EntityCategories

Ez a lap a simpleSAMLphp EntityCategories moduljának továbbfejlesztett változatához tartozó beállításokat ismerteti. Reményeink szerint a modul valamikor a simpleSAMLphp alapcsomagjának is része lesz, ám amíg ez nem történik meg, addig az alábbiak szerint érdemes eljárni, ha használni szeretnénk.

A modul forrása: https://github.com/sitya/simplesamlphp-module-entitycategories.git

Fontos, hogy a modul nem helyettesíti a core:AttributeLimit, vagy a niif:AttributeLimit modult, valamelyikkel együtt használandó!

Telepítés composeren keresztül

```
cd /path/to/simplesamlphp
composer require simplesamlphp/simplesamlphp-module-entitycategories:dev-master
composer config repositories.simplesamlphp/simplesamlphp-module-entitycategories vcs
https://github.com/sitya/simplesamlphp-module-entitycategories.git
composer update
```

Beállítás

Lévén egy AuthProc modullal van dolgunk, így a rendes authproc szekcióba kell elhelyeznünk valahol a sorban, a core:AttributeLimit, vagy niif:AttrobuteLimit előtt. Az alább részletezett alapbeállításokon túl (`default, strict, allowRequestedAttributes`) az egyes entityCategory-k URI-jait kell megadni, mint egy tömb kulcsát, és a hozzátartozó tömb értékeiként pedig a megengedett attribútumok URN-jeit. Tetszőleges számú entityCategory-t megadhatunk, a korábbi beállítások elsősorban azt szabályozzák, hogy mi történjen akkor, olyan entitással van dolgunk, akinek nincs a metadatájában megadva EntityCategory, vagy ha van, nem illeszkedik az általunk explicit beállított listában található EntityCategory-k valamelyikére.

M?köd? példa

```
75  => array(
       'class' => 'entitycategories:EntityCategory',
       'default' => true,
```

```
        'strict' => true,
        'allowRequestedAttributes' => true,
        'http://eduid.hu/category/registered-by-eduidhu' => array (),
        'http://www.geant.net/uri/dataprotection-code-of-conduct/v1' => array (),
        'http://refeds.org/category/research-and-scholarship' => array(
            'urn:oid:2.16.840.1.113730.3.1.241', #displayName
            'urn:oid:2.5.4.4', #sn
            'urn:oid:2.5.4.42', #givenName
            'urn:oid:0.9.2342.19200300.100.1.3', #mail
            'urn:oid:1.3.6.1.4.1.5923.1.1.1.6', #eduPersonPrincipalName
            'urn:oid:1.3.6.1.4.1.5923.1.1.1.9', #eduPersonScopedAffiliation
        ),
    ),

    80 => array(
        'class' => 'niif:AttributeLimit',
        'default' => true,
        'bilateralSPs' => array(
            'google.com' => array('mail'),
            'urn:federation:MicrosoftOnline' => array('IDPEmail', 'ImmutableID'),
        ),
    ),
```

A fenti példa az alábbiakat végzi:

1. a magyar föderáció által regisztrált entitások számára a Resource Registry-ben beállított attribútumok (`RequestedAttributes`) alapján történik az attribútum kiadás;
2. a GÉANT Code of Conduct entitás kategóriájú eduGAIN-es SP-k számára szintén `RequestedAttributes` alapján történik az attribútum kiadás;
3. a Research & Scholarship entitás kategóriájú eduGAIN-es SP-k számára kiadjuk a kategória által igényelt attribútumcsomagot
   - az ilyen SP-k `RequestedAttributes` alapján módosíthatnak az attribútum igényeiken
4. a fenti kategóriáknak nem megfelelő SP-k közül kizárólag a *bilateralSPs* tömbben megadott entitásoknak adunk ki attribútumot.

> **Warning**
>
> Az *entitycategories:EntityCategory* modul *strict* beállítása esetén a lokálisan felvett SP-k esetén nem lesz figyelembe véve az *attributes* tömbelem! Ez azt jelenti, hogy a nem a központi metaadatokból származó SP-ket fel kell venni az *niif:AttributeLimit* modul listájába. Ebből fakadóan ilyen esetben nem is használhatjuk a *core:AttributeLimit* modult.

Opciók

`default`

Logikai kapcsoló, `true/false` értékeket vehet fel. Amennyiben `true`, úgy a beállított EntityCategory-k alatt megadott attribútumkészletet akkor is kiadjuk az adott EntityCategory-val érkező SP-nek, ha az attribútumokat explicit, a `RequestedAttribute`-ok között nem kérte felsorolva. Ennek az R&S EntityCategorynál van különös jelentősége (a példában is ez szerepel), mert ott a specifikáció kimondja, hogy a meghatározott attributúmkészletet akkor is ki kell adni az R&S-t támogató IdP-nek, amennyiben nincsenek ezen attribútumok tételesen felsorolva, mint `RequestedAttributes`.

`strict`

Logikai kapcsoló, `true/false` értékeket vehet fel. Amennyiben `true`, úgy csak és kizárólag a modul konfigurációjában megadott EntityCategory-val érkező SP-k számára kerülnek attribútumok kiadásra, "ismeretlen" EntityCategory-val bíró, és EntityCategory nélküli SP-k számára nem kerül kiadásra semmi.

`allowRequestedAttributes`

Logikai kapcsoló, `true/false` értékeket vehet fel. Amennyiben `true`, úgy megengedjük, hogy egy SP a hozzátartozó EntityCategory konfigurációjában megadott attribútumlistán túl kért attribútumot kiadásra `RequestedAttributes`-on keresztül, `false` esetén csak a listában szereplő attribútumok kiadását engedi a modul. Amennyiben az érték `true` és a `strict` értéke `false`, úgy az "ismeretlen" EntityCategory-val bíró SP-k számára is a `RequestedAttributes` alapján kerülnek kiadásra az attribútumok.

EntraID in eduID.hu

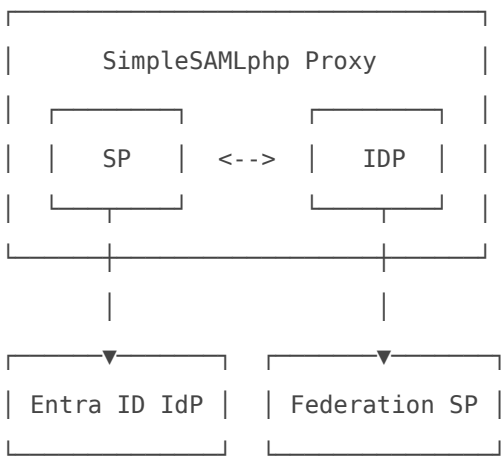
Overview

This document explains how to configure SimpleSAMLphp so that it uses Microsoft Entra ID as the authentication authority (Identity Provider) and then acts as a SAML Identity Provider (IdP) to external federated Service Providers (SPs). This pattern is commonly known as an **authentication proxy** or **IdP proxy**.

In plain terms:

- End users authenticate against **Entra ID**.
- SimpleSAMLphp receives this authentication and optionally enriches or transforms attributes.
- SimpleSAMLphp then issues SAML assertions to federation partners or internal applications.

The proxy setup looks like this:



Configure a SimpleSAMLphp SAML 2.0 Service Provider

To configure SimpleSAMLphp as a SAML 2.0 Service Provider, a new authentication source must be defined in the file `config/authsources.php`. This authentication source represents SimpleSAMLphp in its SP role towards Entra ID and is used to publish SP metadata.

The following example shows a minimal configuration suitable for use with Microsoft Entra ID:

```
$config = [
    /* ... */
    /* An authentication source that can authenticate against SAML 2.0 IdPs. */
    'entraid-sp' => [
        'saml:SP',
        // The entity ID of this SP.
        'entityID' => 'https://proxy.example.org/simplesaml',
        // The entity ID of the IdP this SP should contact.
        'idp' => 'https://sts.windows.net/<your-entra-tenant-id>/'
        'name' => ['en' => 'Microsoft Entra ID'],
        // certificates
        'certificate' => 'server.crt',
        'privatekey' => 'server.key',
        'privatekey_pass' => 'YourPrivateKeyPassphrase', /* you encrypt your private key,
right? */
        'authproc' => [
            /* authproc rules*/
        ],
        // fine tuning the auth source for Entra ID
        'sign.authnrequest' => true,
        'sign.logout' => true,
        'proxymode.passAuthnContextClassRef' => true,
        'disable_scoping' => true,
        'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
    ],
],
```

Configure SimpleSAMLphp SAML 2.0 Identity Provider

In order for SimpleSAMLphp to issue SAML assertions to downstream Service Providers, it must be configured as a SAML 2.0 Identity Provider. This configuration is defined in the file `metadata/saml20-idp-hosted.php`.

The IdP configuration references the previously defined authentication source, effectively chaining authentication to Entra ID.

```
$metadata['http://proxy.example.org/idp'] = [
   /*
    * The hostname of the server (VHOST) that will use this SAML entity.
    *
    * Can be '__DEFAULT__', to use this entry by default.
    */
   'host' => '__DEFAULT__',
   // X.509 key and certificate. Relative to the cert directory.
   'privatekey' => 'server.pem',
   'privatekey_pass' => 'YourPrivateKeyPassphrase',
   'certificate' => 'server.crt',
   /*
    * Authentication source to use. Must be one that is configured in
    * 'config/authsources.php'.
    */
   'auth' => 'entraid-sp', // proxy to Microsoft Entra ID
   'attributes.NameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
   'authproc' => [
   ],
];
```

Create a new Enterprise Application in Entra ID

1. Create a new Enterprise Application

A new **Enterprise Application** must be created in the Entra ID portal to represent SimpleSAMLphp in its role as a SAML Service Provider. This can be done by navigating to the **Enterprise applications** section of the Entra ID portal and creating a new application. During creation, the option to create a custom application that is not found in the gallery should be selected. A descriptive name and also select *Integrate any other application you don't find in the gallery*.

2. Configure SAML-based Single Sign-On

After the application has been created, SAML-based single sign-on must be enabled. This is done by opening the application, navigating to the **Single sign-on** section, and selecting **SAML** as the sign-on method. The trust relationship between Entra ID and SimpleSAMLphp is established by

uploading the SAML 2.0 SP metadata generated by SimpleSAMLphp. The metadata upload automatically populates the basic SAML configuration, including the entity ID and assertion consumer service URL.

3. Download Entra ID Federation Metadata

To finalise the SimpleSAMLphp side of the bilateral trust relationship between your Entra ID tenant and SimpleSAMLphp, copy your Enterprise Application's *App Federation Metadata*. Using SimpleSAMLphp's Metadata Converter (found on the *Federation* tab of SimpleSAMLphp's admin portal), convert your App Federation Metadata to SimpleSAMLphp's native PHP format. Once you have the converted metadata, paste it into the `metadata/saml20-idp-remote.php` file.

4. Configure Attribute Claims Rules

Attribute and claim mappings can be adjusted in the Entra ID application to ensure that the required user attributes are released to SimpleSAMLphp. These attributes will later be available for transformation, filtering, or enrichment before being sent to downstream Service Providers.

Attribute Mapping and Transformation

When authenticating against Microsoft Entra ID, user attributes are returned as SAML claims using Microsoft-specific or WS-Federation-style claim URIs. In most federation environments, these claims must be mapped to standard SAML or eduPerson attribute names before they are released to downstream Service Providers.

SimpleSAMLphp performs attribute mapping through authentication processing filters. Mapping rules are applied in the `authproc` section of the authentication source that represents Entra ID, ensuring that attributes are normalized as soon as they enter SimpleSAMLphp. These mappings can either reuse
[https://github.com/simplesamlphp/simplesamlphp/blob/master/attributemap/entra2name.php built-in attribute maps provided] by SimpleSAMLphp or be defined explicitly using custom rules.

Here is an example of using `core:AttributeMap` processing filter:

```
'authproc' => [
    /* ... */
    60 => [
        'class' => 'core:AttributeMap',
        /* there are several versions of the userprincipalname claim, you only need the one
you use */
        'http://schemas.xmlsoap.org/claims/UPN' => 'eduPersonPrincipalName',
```

```
        'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn' =>
'eduPersonPrincipalName',
        'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name' =>
'eduPersonPrincipalName',
        /* other possible attributes */
        'http://schemas.xmlsoap.org/claims/CommonName' => 'displayName',
        'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname' => 'givenName',
        'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname' => 'sn',
        'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress' => 'mail',
        'http://schemas.microsoft.com/ws/2008/06/identity/claims/groups' => 'memberOf',
    ],
    /* ... */
],
```

or

```
'authproc' => [
    60 => [
'class' => 'core:AttributeMap',
'attributemap' => 'entra2name',
    ],
],
```

Once mapped, attributes can be further filtered, enriched, or selectively released by additional authentication processing filters before being issued by the proxy IdP.

Configure SimpleSAMLphp to Use Entra ID as an Authentication Source

With the Enterprise Application configured, SimpleSAMLphp must be instructed to use Entra ID as its authentication source. This is done by setting the IdP entity ID in the entraid-sp authentication source to the Entra ID tenant identifier.

```
'idp' => 'https://sts.windows.net/<your-entra-tenant-id>/',
```

This configuration causes SimpleSAMLphp, acting as a Service Provider, to redirect authentication requests to Entra ID. After importing the Entra ID metadata, the corresponding entity ID should be visible under SAML 2.0 IdP metadata on the Federation tab of the SimpleSAMLphp admin interface.

Testing

You should now be able to go to the **Test** tab in the admin portal, log in to your `entraid-sp` authentication source, and be redirected to your Entra ID application's login page. Once logged in, it is worth verifying that SimpleSAMLphp is correctly receiving the attributes from Entra ID.

Sources

- https://nathansenblog.wordpress.com/2021/02/23/azure-ad-single-sign-on-with-simplesamlphp
- https://safire.ac.za/technical/resources/configuring-simplesamlphp-to-use-entra-id

AAI AzureADasAuthsource

Amennyiben Azure AD-ban tároljuk a felhasználói adatokat, úgy lehetőség van azt azonosítási forrásként is használni. A SimpleSAMLphp oldalon leírt telepítés után az alábbiak elvégzésére van szükség:

(ez csak egy példakonfiguráció, természetesen el lehet ettől térni)

Teend?k SimpleSAMLPHP (SSP) oldalon

Keressük ki az Azure AD-ból a Tenant ID-t. A beállítás során erre *TenantID*-val fogunk hivatkozni, oda minden esetben ezt az azonosítót kell behelyettesíteni. Az azonosítót jelenleg a https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/Overview oldalon keresztül lehet beszerezni (Forrás: https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-how-to-find-tenant)

A *DOMAIN* helyére a használni kívánt scope-ot szükséges behelyettesíteni (pl intezmeny.hu)

config/authsources.php fájlba

```
    'default-sp' => [
        'saml:SP',

        // The entity ID of this SP.
        // Can be NULL/unset, in which case an entity ID is generated based on the metadata
URL.
        'entityID' => null,

        // The entity ID of the IdP this SP should contact.
        // Can be NULL/unset, in which case the user will be shown a list of available IdPs.
        'idp' => 'https://sts.windows.net/_TenantID_/',

        // The URL to the discovery service.
        // Can be NULL/unset, in which case a builtin discovery service will be used.
```

```
        'discoURL' => null,
        'NameIDFormat' => 'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent',
        'simplesaml.nameidattribute' => 'eduPersonTargetedID',

        /*
         * The attributes parameter must contain an array of desired attributes by the SP.
         * The attributes can be expressed as an array of names or as an associative array
         * in the form of 'friendlyName' => 'name'. This feature requires 'name' to be set.
         * The metadata will then be created as follows:
         * <md:RequestedAttribute FriendlyName="friendlyName" Name="name" />
         */
        /*
        'name' => [
            'en' => 'A service',
            'no' => 'En tjeneste',
        ],

        'attributes' => [
            'attrname' => 'urn:oid:x.x.x.x',
        ],
        'attributes.required' => [
            'urn:oid:x.x.x.x',
        ],
        */
    ],
```

config/config-metarefresh.php fájlba

```
$config = [

    'sets' => [
        'azure' => [
            'cron' => ['hourly'],
            'sources' => [
                [
                    'src' =>
'https://login.microsoftonline.com/_TenantID_/federationmetadata/2007-
06/federationmetadata.xml',
                ],
```

```
            ],
            'expireAfter' => 34560060, // Maximum 4 days cache time (3600*24*4)
            'outputDir' => 'metadata/metarefresh-azure',
            'outputFormat' => 'flatfile',
        ],
    ],
];
```

metadata/saml20-idp-hosted.php

A

```
    'authproc' => [

        10 => [
              'class' => 'core:AttributeMap',
              'azure2name'
        ],

        15 => [
            'class' => 'core:AttributeCopy',
            'eduPersonPrincipalName' => 'schacPersonalUniqueCode',
        ],

        16 => ['class' => 'core:PHP',            'code' => '                $attributes[=
"urn:schac:personalUniqueCode:int:esi:_DOMAIN_:" .
$attributes["schacPersonalUniqueCode"]("schacPersonalUniqueCode"][0])[0];
            ',
        ],

        18 => [
            'class' => 'core:AttributeAlter',
            'subject' => 'eduPersonPrincipalName',
            'pattern' => '/^.*$/',
            'replacement' => '${0}@_DOMAIN_',
            'target' => 'eduPersonPrincipalName'
        ],

        20 => [
```

```
            'class' => 'core:AttributeAdd',
            'eduPersonEntitlement' => array('urn:mace:dir:entitlement:common-lib-terms')
        ],

        22 => [
            'class' => 'core:AttributeAdd',
            'schacHomeOrganization' => array('_DOMAIN_')
        ],

        23 => [
            'class' => 'core:AttributeAdd',
            'schacHomeOrganizationType' =>
array('urn:schac:homeOrganizationType:eu:higherEducationalInstitution')
        ],

// Azure AD-ban célszerű az affilation-t (intézményhez való viszonyt, pl hallgató, oktató,
dolgozó) security group alapján meghatározni. Ezeknek az objektum azonosítóját át fogja adni
az Azure AD, amit könnyen kicserélhetünk a kívánt affilation-re:

        31 => [
            'class' => 'core:AttributeAlter',
            'subject' => 'eduPersonAffiliation',
            'pattern' => '/^_eduID_Dolgozó_GroupID_$/', // _eduID_Dolgozó_GroupID_ értéket
cseréljük a megfelelő Objektum ID-ra
            'replacement' => 'faculty',
        ],

        32 => [
            'class' => 'core:AttributeAlter',
            'subject' => 'eduPersonAffiliation',
            'pattern' => '/^_eduID_Hallgató_GroupID_$/', // _eduID_Hallgató_GroupID_ értéket
cseréljük a megfelelő Objektum ID-ra
            'replacement' => 'student',
        ],

        33 => [
            'class' => 'core:AttributeAlter',
            'subject' => 'eduPersonAffiliation',
            'pattern' => '/^_eduID_Admin_GroupID_$/', // _eduID_Admin_GroupID_ értéket
```

```
cseréljük a megfelelő Objektum ID-ra
            'replacement' => 'staff',
        ],

        34 => [
            'class' => 'core:AttributeAdd',
            'eduPersonAffiliation' => array('member'),
        ],

        35 => [
            'class' => 'core:AttributeCopy',
            'eduPersonAffiliation' => 'eduPersonScopedAffiliation'
        ],

        36 => [
            'class' => 'core:AttributeAlter',
            'subject' => 'eduPersonScopedAffiliation',
            'pattern' => '/^.*$/',
            'replacement' => '${0}@$_DOMAIN_',
        ],

        50 => [
            'class' => 'core:TargetedID',
            'identifyingAttribute' => 'eduPersonPrincipalName',
            'nameId' => TRUE,
        ],

        60 => [
            'class' => 'core:AttributeMap',
            'name2oid'
        ],
        75  => [
            'class' => 'entitycategories:EntityCategory',
            'default' => true,
            'strict' => false,
            'allowRequestedAttributes' => true,
            'http://eduid.hu/category/registered-by-eduidhu' => [],
            'http://www.geant.net/uri/dataprotection-code-of-conduct/v1' => [
                'urn:oid:2.16.840.1.113730.3.1.241', # displayName
```

```
                'urn:oid:2.5.4.4', # sn
                'urn:oid:2.5.4.42', # givenName
                'urn:oid:0.9.2342.19200300.100.1.3', # mail
                'urn:oid:1.3.6.1.4.1.5923.1.1.1.6', # eduPersonPrincipalName
                'urn:oid:1.3.6.1.4.1.5923.1.1.1.9', # eduPersonScopedAffiliation
                'urn:oid:1.3.6.1.4.1.5923.1.1.1.1', # eduPersonAffiliation
            ],
            'http://refeds.org/category/research-and-scholarship' => [
                'urn:oid:2.16.840.1.113730.3.1.241', # displayName
                'urn:oid:2.5.4.4', # sn
                'urn:oid:2.5.4.42', # givenName
                'urn:oid:0.9.2342.19200300.100.1.3', # mail
                'urn:oid:1.3.6.1.4.1.5923.1.1.1.6', # eduPersonPrincipalName
                'urn:oid:1.3.6.1.4.1.5923.1.1.1.9', # eduPersonScopedAffiliation
                'urn:oid:1.3.6.1.4.1.5923.1.1.1.1', # eduPersonAffiliation
            ],
        ],

        90 => 'core:AttributeLimit',
    ],

    'simplesaml.nameidattribute' => 'urn:oid:1.3.6.1.4.1.5923.1.1.1.6',

    'attributeencodings' => array(
        'urn:oid:1.3.6.1.4.1.5923.1.1.1.10' => 'raw', /* eduPersonTargetedID with oid
NameFormat. */
    ),

    'sign.logout' => true
];
```

attributemap/azure2oid.php

```
<?php
$attributemap = [
    // displayName
    'http://schemas.microsoft.com/identity/claims/displayname' =>
'urn:oid:2.16.840.1.113730.3.1.241',
```

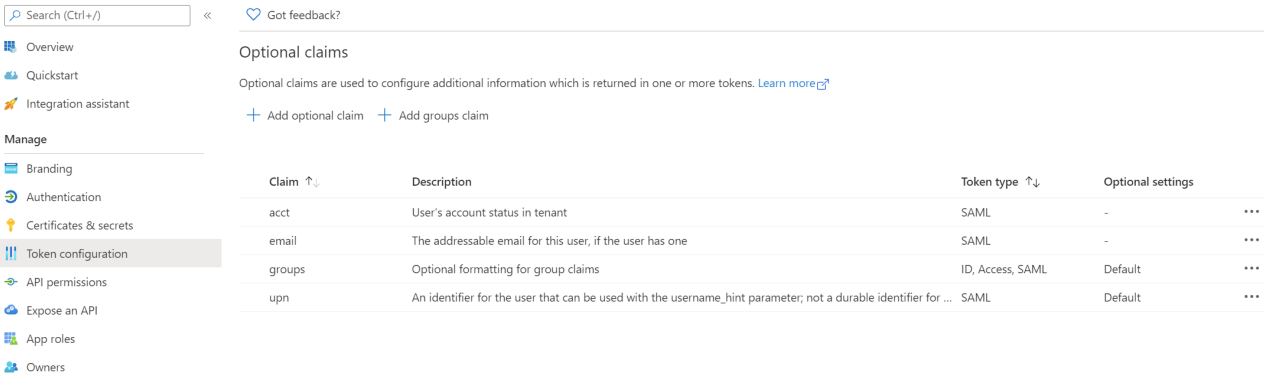
```
    // eppn
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name' =>
'urn:oid:1.3.6.1.4.1.5923.1.1.1.6',
    // givenName
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname' => 'urn:oid:2.5.4.42',
    // cn
    '://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname' => 'urn:oid:2.5.4.3',
    // surname
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname' => 'urn:oid:2.5.4.4',
    // mail
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress' =>
'urn:oid:0.9.2342.19200300.100.1.3',
    // o & organisation
    'http://schemas.microsoft.com/identity/claims/tenantid' => 'urn:oid:2.5.4.10',
];
```

attributemap/azure2name.php

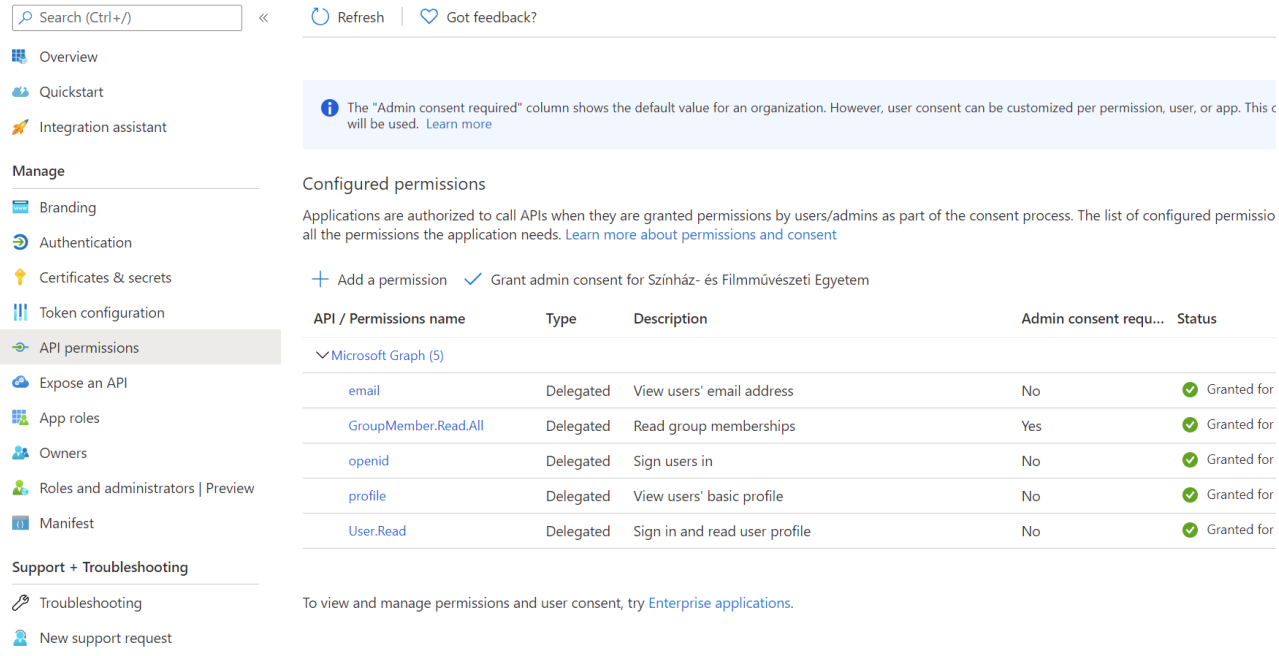
```
<?php
$attributemap = [
    // eppn
    'http://schemas.microsoft.com/identity/claims/objectidentifier' =>
'eduPersonPrincipalName',
    // mail
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress' => 'mail',
    // displayName
    'http://schemas.microsoft.com/identity/claims/displayname' => 'displayName',
    // givenName
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname' => 'givenName',
    // cn
    'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname' => 'sn',
    // affiliation
    'http://schemas.microsoft.com/ws/2008/06/identity/claims/groups' =>
'eduPersonAffiliation',
];
```

Teend?k Azure AD oldalon

1. A https://portal.azure.com/ oldalon jelentkezzünk be egy adminisztrátori fiókkal
2. Válasszuk az "App registrations"-t
3. "New registration"
4. "Redirect URI (optional)" -nál adjuk meg a telepített SSP default SP címét. Pl: https://idp.*DOMAIN*/simplesaml/module.php/saml/sp/metadata.php/default-sp
5. "Token configuration" # > "Add optional claim"> "Token type"-nál válasszuk a "SAML"-t és jelöljük ki az összes attribútumot, majd, "Add"
6. "Add groups claim", majd mentsük el



7. Állítsuk be az API persmissions-t az alábbi alapján:



Teszt

SamlSign

Parancssoros eszköz, melyhez debian alatt az `opensaml2-tools` csomagot kell telepíteni. A program kétféle üzemmódban képes működni: **metaadat aláírása** és **metaadat ellenőrzése**.

Metaadat aláírása

```
samlsign -s -k /path/to/mainkey.key -f /path/to/metadatatosign.xml
```

Alapértelmezés szerint a samlsign az eredményeket az alapértelmezett kimenetre írja ki (STDOUT), így célszerű ezt egy új fájlba átirányítani:

```
samlsign -s -k /path/to/mainkey.key -f /path/to/metadatatosign.xml >
/path/to/metadatasigned.xml
```

Metaadat ellenőrzése

```
samlsign -c /path/to/maincert.crt -f /path/to/metadatatosign.xml
```

Samlsign legfontosabb kapcsolói

- -s ez határozza meg, hogy aláírunk, vagy ellenőrzünk. Ha megadtuk kapcsolóként, akkor a program megpróbálja aláírni a megadott xml fájlt, ha nem, akkor ugyanezt a fájlt ellenőrizni fogja.
- -f az ellenőrzendő/aláírandó fájl elérhetősége **abszolút útvonallal** megadva
- -k a privát kulcs elérhetősége **abszolút útvonallal** megadva
- -c az ellenőrzésre használt publikus kulcs elérhetősége **abszolút útvonallal** megadva
- További részletes leírás a samlsign man oldalán

További fontos tudnivalók A samlsign nem szereti a metadatában szereplő `Organization`-nel kapcsolatos adatokat, mivel ilyen tag-ekben kötelezően megadandó `xml:lang` attribútumot `lang`-ra alakítja át, ami által viszont nem lesz érvényes (valid) maga a metaadat, így pl. a shibboleth sem fog tudni vele mit kezdeni. A **megoldás** (nem szép, de hasznos): az aláírás előtt álló metaadatokból ki kell szedni az `Organization`-nel kapcsolatos adatokat. Ezek után már gond nélkül aláírja és az eredmény is érvényes lesz.

Apró trükk a privát kulcs kinyerésére `jks`-ből

Tekintettel arra, hogy a `keytool` nem teszi lehetővé a privát kulcs kihalászását JavaKeystore-ból, így külső segítséget kell igénybe vennünk. A segédalkalmazás `ExportPrivateKey` névre hallgat, és innen letölthető egy darab zip fájl. Használata rendkívül egyszerű:

```
java -jar ExportPrivateKey.zip {jks fájl elérhetősége} JKS {jks jelszó} {alias} {célfájl}
```

Ezek után a létrehozott kulccsal már használhatjuk is a samlsign-t.