SimpleSAMLphp

- SimpleSAMLphp
- SimpleSAMLphp proxy vidyo portálhoz
- Alkalmazások samlizálást segítő teszt IdP simplesamlPHP segítségével
- SimpleSAMLphp NIIF ldap séma mapping
- Single Logout in Shibboleth IdP
- Attribute Conversion for simpleSAMLphp
- SimpleSAMLMixedMetadata
- SSP2

SimpleSAMLphp

Az alábbi lapon megkíséreljük összefoglalni a legfontosabb lépéseket, melyek általános esetben elegendőek ahhoz, hogy működő SimpleSAMLphp (SSP) alkalmazást állítsunk üzembe.

Telepítés

A leírás a forrásból történő telepítés lépéseit írja le. Az itt részletezetten kívül a SimpleSAMLphp telepíthető Debian (vagy más) operációs rendszer csomagjából, de ebben az esetben ne telepítsunk composerrel third-party (pl. általunk készített) modulokat!

El?készületek

Ahhoz, hogy problémamentesen telepíthessük SSP alkalmazásunkat, az alábbi szoftverkomponenseknek kell működniük szerverünkön.

- PHP futtatására alkalmas webszerver
- PHP környezet (>=5.4)
- A következő PHP kiterjesztéseket engedélyezni kell
  - `date`, `dom`, `hash`, `libxml`, `openssl`, `pcre`, `SPL`, `zlib`
  - LDAP-ból történő autentikáció esetén: `ldap`
  - Adatbázisból történő autentikáció esetén a megfelelő adatbázis-csatolót `mysql`, `pgsql`
  - RADIUS szerveren keresztül történő autentikáció esetén: `radius`
  - Assertion-ök titkosítása esetén: `mcrypt`
  - Memcache használata esetén: `memcache`
  - HEXAA integrációhoz (SP): `soap`

Debian 9 / Ubuntu 16.04 LTS csomagok

```
sudo apt install php php-dom mcrypt php-xml php-mbstring
```

RHEL / CentOS 7 csomagok

A **php-mcrypt** csomaghoz engedélyezni kell az "*epel-release*"-t.

```
sudo yum install epel-release
sudo yum update
sudo yum install php php-dom php-mcrypt php-xml php-mbstring php-common mod_ssl
```

Composer

A **composer** PHP csomagkezelőt is telepíteni kell (akár forrásból, akár csomagból), hogy telepíteni lehessen a SimpleSAMLphp futásához szükséges PHP library-ket.

Letöltés

A GitHubról történő telepítés előnye, hogy a simplesamlphp könnyen frissíthető marad, csak a third party modulokat kell újratelepíteni. Az utolsó stabil verzió számát a

https://simplesamlphp.org/download oldalról tudhatjuk meg.

```
cd /var
git clone
[https://github.com/simplesamlphp/simplesamlphp.git](https://github.com/simplesamlphp/simplesa
mlphp.git)
cd simplesamlphp
git checkout tags/v1.16.2 -b v1.16.2
composer install --no-dev
```

Apache konfigurálás

A webről csak a `/var/simplesamlphp/www` könyvtárat kell elérni. **Tilos** a teljes simplesamlphp könyvtárat a DocumentRoot alá tenni!

```
Alias /simplesaml /var/simplesamlphp/www
<Directory /var/simplesamlphp/www>
  Require all granted
</Directory>
```

Alapbeállítások

Konfigurációs fájlok másolása

Mielőtt aktiváljuk valamelyik főszolgáltatását (IdP,SP...) a telepített alkalmazásnak, néhány beállítást meg kell adnunk a `config/config.php` és `config/authsources.php` konfigurációs fájlokban.

- **config.php** másolása a **config-templates** mappából cp config-templates/config.php config/
- **authsources.php** másolása a **config-templates** mappából cp config-templates/authsources.php config/

A **config.php** és **authsources.php** fájlok másolása utána ellenőrizzük, hogy a SimpleSAMLphp működik-e, a https://example.org/simplesaml oldalon.

Konfigurációs fájlok szerkesztése

Adminisztrációs adatok beállítása

Amennyiben az SimpleSAMLphp kezdőlapja hiba nélkül megjelent, akkor nyissuk meg a **config/config.php** fájl szerkesztésre és végezzük el az alábbi beállításokat.

- **Az "admin" felhasználó jelszavát, mellyel webes felületen keresztül be tud lépni a települő SSP-be.**
  'auth.adminpassword' => 'ujjelszotirdide',
- **Titkosítási feladatokhoz szükséges "salt", azaz véletlenszerűen összeálló karaktersorozat**

  ```
  'secretsalt' => 'randombytesinsertedhere',
  ```

  A karaktersorozat előállításában segíthet az alábbi parancs:

  ```
  tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32 count=1 2>/dev/null;echo
  ```

- **Elérhetőségeket, amely adatok bekerülnek majd a generált metaadatba**

  ```
  'technicalcontact_name'     => 'Gipsz Jakab',
  'technicalcontact_email'    => 'jakab.gipsz@example.org',
  ```

- **Nyelv és időzóna adatok**

  ```
  'language.default'      => 'hu',
  'timezone' => 'Europe/Budapest',
  ```

Az alapadatok megadása után mentsük és zárjuk be a **config.php**-t.

Naplózás beállítása

Alapértelmezetten a SimpleSAMLphp a **syslog**-ba irányítja a naplózást.

Ha fájlba akarunk naplózni, akkor a megfelelő könyvtárhoz biztosítsunk írás jogot a webszerver felhasználónak, és ne felejtsünk el gondoskodni a naplófájlok rotálásáról!

- **log** mappa létrehozása és jogosultság beállítása

  ```
  sudo mkdir log; sudo chown www-data:adm log; sudo chmod 755 log
  ```

- Naplózási szint beállítása a **config/config.php**-ban

```
'debug' => array(
    'saml' => true,
    'backtraces' => true,
    'validatexml' => false,
),
'logging.level' => SimpleSAML\Logger::DEBUG,
'logging.handler' => 'file',
```

A "SimpleSAML\Logger::DEBUG" a legrészletesebb naplózási beállítás, éles rendszernél nem ajánlott csak hiba keresés esetén.

Tanúsítvány készítése

Nem ajánlott a SimpleSAMLphp-hoz és webszerverhez ugyanazt a tanúsítvány használni!

- A SimpleSAMLphp alapértelmezetten a tanúsítványt a **cert** mappában keresi.

```
mkdir cert
```

- Az alábbi paranccsal egy 10 éves self-signed tanúsítvány generálunk a SimpleSMALphp számára.

```
openssl req -new -newkey rsa:2048 -x509 -days 3652 -nodes -out cert/saml-example-org.crt -keyout cert/saml-example-org.key
```

A fingerprint az alábbi módon kérdezhető le a legegyszerűbben

```
openssl x509 -fingerprint -noout -in cert/saml-example-org.crt
```

Telepítés kész

Amennyiben elkészültünk a fenti lépésekkel, úgy a https://service.example.org/simplesaml/ címen elérjük a telepített SSP-nk webes adminfelületét, ahol megejthetjük a további beállítások nagy részét.

Identity Provider (IdP) beállítás

Alapbeállítások

IdP engedélyezése: a **config/config.php** fájlban kell a saml20 idp-t "true"-re állítani.

```
'enable.saml20-idp' => true,
```

LDAP autentikáció

Meg kell adni, hogy az IdP milyen módon azonosítsa a felhasználót, amennyiben alapértelmezés szerint nem engedélyezett modult szeretnénk használni, úgy a megfelelő modult a `modules` könyvtár alatt engedélyezni kell. Az alábbi példában az LDAP alapú azonosítást mutatjuk be, amely külön modult nem igényel, alapértelmezés szerint része a telepített alkalmazásnak.

Javasolt az LDAP-ban egy olyan bejegyzést létrehozni az IdP számára, amely olvasni tudja a felhasználóknak a föderációban használt attribútumait. Az azonosítás alapértelmezett módon a felhasználó nevében történő újra bind-olással történik, így a jelszóhoz nem kell hozzáférést adni.

Ahhoz, hogy megadhassuk az LDAP-hoz tartozó beállításokat, a `config/authsources.php` fájlt kell szerkesztenünk. Az alábbi kódrészletet elegendő beszúrni, és az egyes változóknak a helyi LDAP-nak megfelelő adatokat értékül adni.

```
'example-ldap' => array(
    'ldap:LDAP',

    /* The hostname of the LDAP server. */
    'hostname' => 'ldap.example.org',

    /* Whether SSL/TLS should be used when contacting the LDAP server. */
    'enable_tls' => FALSE,

    /*
     * Which attributes should be retrieved from the LDAP server.
     * This can be an array of attribute names, or NULL, in which case
     * all attributes are fetched.
     */
    'attributes' => NULL,

    /*
     * The pattern which should be used to create the users DN given the username.
     * %username% in this pattern will be replaced with the users username.
     *
     * This option is not used if the search.enable option is set to TRUE.
    'dnpattern' => 'uid=%username%,ou=people,dc=example,dc=org',
```

```
     */

    /*
     * As an alternative to specifying a pattern for the users DN, it is possible to
     * search for the username in a set of attributes. This is enabled by this option.
     */
    'search.enable' => TRUE,

    /*
     * The DN which will be used as a base for the search.
     * This can be a single string, in which case only that DN is searched, or an
     * array of strings, in which case they will be searched in the order given.
     */
    'search.base' => 'ou=people,dc=example,dc=org',

    /*
     * The attribute(s) the username should match against.
     *
     * This is an array with one or more attribute names. Any of the attributes in
     * the array may match the value the username.
     */
    'search.attributes' => array('uid', 'mail'),

    /*
     * The username & password the simpleSAMLphp should bind to before searching. If
     * this is left as NULL, no bind will be performed before searching.
     */
    'search.username' => 'cn=simplesamlphp,dc=example,dc=org',
    'search.password' => 'servicepassword',

    'priv.read' => TRUE,
    // The DN & password the SimpleSAMLphp should bind to before
    // retrieving attributes. These options are required if
    // 'priv.read' is set to TRUE.
    'priv.username' => 'cn=simplesamlphp,dc=example,dc=org',
    'priv.password' => 'servicepassword;,
  ),
```

Metaadat alapok

A beállítandó IdP alapvető paraméterei a `metadata/saml20-idp-hosted.php` fájlban állíthatók. Az alábbi kódrészlet egy minimális, de már működő példát mutat.

```
$metadata['__DYNAMIC:1__'] = array(
    /*
     * The hostname for this IdP. This makes it possible to run multiple
     * IdPs from the same configuration. '__DEFAULT__' means that this one
     * should be used by default.
     */
    'host' => '__DEFAULT__',

    /*
     * The private key and certificate to use when signing responses.
     * These are stored in the cert-directory.
     */
    'privatekey' => 'saml-example-org.key',
    'certificate' => 'saml-example-org.crt',

    /*
     * The authentication source which should be used to authenticate the
     * user. This must match one of the entries in config/authsources.php.
     */
    'auth' => 'example-ldap',
);
```

Megfelelő beállítások után a dinamikusan generált metadata a `/saml2/idp/metadata.php` útvonalon érhető el.

Tesztelés

Legegyszerűbben a telepített SSP felületén tudjuk ellenőrizni, hogy a beállított autentikációs forrás működik-e. A felületen az Authentication fül alatt található egy 'Test authentication sources' link, amelyre kattintva látható minden beállított autentikációs forrás is, így a két alapértelmezett, teszt célokat szolgáló alatt a most beállított example-ldap névre hallgatónak is meg kell jelenni. (A közvetlen url erre az oldalra: simplesaml/module.php/core/authenticate.php) Ez utóbbira kattintva a beállított LDAP-ból autentikálva be kell tudnunk jelentkeznünk; siker esetén az LDAP-ból kinyerhető attribútumokat láthatjuk.

Service Provider (SP) beállítás

Alapbeállítások

A telepített alkalmazásunk által kezelt SP-ket a **config/authsources.php** fájlban tudjuk beállítani. Az *entityID, idp, discoURL* értékeket most hagyjuk *NULL* értéken és adjuk hozzá a **privatekey** / **certificate** részt.

A SimpleSAMLphp a tanúsítvány fájlokat a korábban létrehozott **cert** mappában fogja keresni, a fájlokat elég relatív elérési úttal megadni.

```
 'default-sp' => array(
        '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 should SP should contact.
        // Can be NULL/unset, in which case the user will be shown a list of available IdPs.
        'idp' => NULL,

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

        'privatekey' => 'saml-example-org.key',
        'certificate' => 'saml-example-org.crt',

 ),
```

A fenti beállítások alapján az SP entityID-ja megegyezik a metadata elérési útvonalával (szokásos megoldás SSP-nél), nem adunk meg alapértelmezett idp-t, ezáltal az SSP választási lehetőséget kínál fel számunkra, mikor az SP-re szeretnénk bejelentkezni, ill. most még Discovery Service URL-t sem adunk meg, hanem a beépítettet használjuk. Ez utóbbit majd a HREF-be történő integrációkor meg kell változtatni - lásd lejjebb.

Az SP készen áll arra, hogy összekössük egy IdP-vel (ez jellemzően szintén egy SimpleSAMLphp alkalmazás). Ehhez szükséges, hogy SP oldalon beállítsuk az IdP metadata-t és IdP oldalon is beállítsuk az SP metadata-t.

Metadata

Metadata fájlok

A különböző metadata template fájlok a **metadata-templates** mappában találhatóak. A nekünk szükséges template fájlt másoljuk át a metadata mappába.

- **SP** oldalon lennie kell egy **metadata/saml20-idp-remote.php** fájlnak. Ez a fájl tartalmazza az IdP eléréséhez szükséges adatokat.

```
cp metadata-templates/saml20-idp-remote.php metadata
```

- **IdP** oldalon lennie kell egy **metadata/saml20-sp-remote.php** fájlnak. Ez a fájl tartalmazza az SP eléréséhez szükséges adatokat.

```
cp metadata-templates/saml20-sp-remote.php metadata
```

Metadata letöltés

Ezen az oldalon megtaláljuk az SP vagy IdP-re vonatkozó **metadata**-t, **XML** és **PHP** formátumban:

https://example.org/simplesaml/module.php/saml/sp/metadata.php/default-sp?output=xhtml

SP metadata beállítás IdP oldalon

A metadata **simplesaml** kezdőlapon, az alábbi helyen érhető el:

- Magyar nyelv esetén: "Föderáció" fül / "SAML 2.0 SP Metaadatok" pont alatt a "Mutasd a metaadatokat" linkre kattintva juthatunk el a fenti menüponthoz.
- Angol nyelv esetén: "Federation" fül / "SAML 2.0 SP Metadata" pont alatt a "Show metadata" linkre kattintva juthatunk el a fenti menüponthoz.

A "*SimpleSAMLphp fájl formátumban - akkor használható, ha a másik oldalon SimpleSAMLphp van*" mezőből tegyük a vágólapra az alábbi PHP kódot:

```
$metadata['https://example.org/simplesaml/module.php/saml/sp/metadata.php/default-sp'] = array
(
  'SingleLogoutService' =>
  array (
    0 =>
    array (
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
      'Location' => 'https://example.org/simplesaml/module.php/saml/sp/saml2-
logout.php/default-sp',
    ),
  ),
  'AssertionConsumerService' =>
```

```
  array (
    0 =>
    array (
      'index' => 0,
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST',
      'Location' => 'https://example.org/simplesaml/module.php/saml/sp/saml2-acs.php/default-
sp',
    ),
    1 =>
    array (
      'index' => 1,
      'Binding' => 'urn:oasis:names:tc:SAML:1.0:profiles:browser-post',
      'Location' => 'https://example.org/simplesaml/module.php/saml/sp/saml1-acs.php/default-
sp',
    ),
    2 =>
    array (
      'index' => 2,
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact',
      'Location' => 'https://example.org/simplesaml/module.php/saml/sp/saml2-acs.php/default-
sp',
    ),
    3 =>
    array (
      'index' => 3,
      'Binding' => 'urn:oasis:names:tc:SAML:1.0:profiles:artifact-01',
      'Location' => 'https://example.org/simplesaml/module.php/saml/sp/saml1-acs.php/default-
sp/artifact',
    ),
  ),
  'contacts' =>
  array (
    0 =>
    array (
      'emailAddress' => 'admin@example.org',
      'contactType' => 'technical',
      'givenName' => 'Example Corp. IT Dept.',
    ),
  ),
  'certData' =>
```

```
'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==',
);
```

A vágólapra másolt kódot IdP oldalon, a **metadata/saml20-sp-remote.php** fájl végére illesszük be.

IdP metadata beállítás SP oldalon

A metadata **simplesaml** kezdőlapon, az alábbi helyen érhető el:

- Magyar nyelv esetén: "Föderáció" fül / "SAML 2.0 IdP Metaadatok" pont alatt a "Mutasd a metaadatokat" linkre kattintva juthatunk el a fenti menüponthoz.
- Angol nyelv esetén: "Federation" fül / "SAML 2.0 IdP Metadata" pont alatt a "Show metadata" linkre kattintva juthatunk el a fenti menüponthoz.

A "*SimpleSAMLphp fájl formátumban - akkor használható, ha a másik oldalon SimpleSAMLphp van*" mezőből tegyük a vágólapra az alábbi PHP kódot:

```
$metadata['https://idp.example.org/simplesaml/saml2/idp/metadata.php'] = array (
  'metadata-set' => 'saml20-idp-remote',
  'entityid' => 'https://idp.example.org/simplesaml/saml2/idp/metadata.php',
  'SingleSignOnService' =>
  array (
    0 =>
    array (
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
      'Location' => 'https://idp.example.org/simplesaml/saml2/idp/SSOService.php',
    ),
  ),
  'SingleLogoutService' =>
  array (
    0 =>
    array (
      'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
      'Location' => 'https://idp.example.org/simplesaml/saml2/idp/SingleLogoutService.php',
    ),
  ),
  'certData' => 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==',
  'NameIDFormat' => 'urn:oasis:names:tc:SAML:2.0:nameid-format:transient',
);
```

A vágólapra másolt kódot SP oldalon, a **metadata/saml20-idp-remote.php** fájl végére illesszük be.

Tesztelés

A fent elvégzett alapbeállítások után már tudjuk tesztelni a, hogy a felépített IdP - SP kapcsolat működik-e.

SP oldalon nyissuk meg a **simplesaml** kezdőlapot:

- Magyar nyelv esetén: "Azonosítás (autentikáció)" fül / "Azonosítási (autentikációs) beállítások tesztelése" link / "default-sp" link-re kattintva tudjuk tesztelni az IdP - SP kapcsolatot.
- Angol nyelv esetén: "Authentication" fül / "Test configured authentication sources" link / "default-sp" link-re kattintva tudjuk tesztelni az IdP - SP kapcsolatot.

A legördülő menüben az IdP-nk "nevére" kattintva, be kell tudnunk jelentkezni (az IdP-n keresztül). Ha működik, akkor az IdP visszairányít az SP-re, kiírja az azonosított felhasználó attribútumait.

Az alapvető lépsekkel kész vagyunk, van egy működő SP-nk és egy működő IdP-nk.

HREF-integráció

Metadata beállítása (IdP és SP is)

Javasolt dinamikus metaadatforrást (MDX) használni, opcionálisan kiegészítve statikus állományokkal. Részletes leírás itt: SimpleSAMLMixedMetadata

IdP

Amennyiben van SSP alapú IdP-nk, melyet szeretnénk a föderáció részévé tenni, úgy a teendők a következők.

- (Az adminisztratív teendőktől itt most eltekintünk, a csatlakozás folyamata itt van leírva)
- Kell küldeni egy levelet a info@eduid.hu címre, benne néhány mondat mellett az IdP metaadatának URL-jével (https://example.org/simplesamlphp/saml2/idp/metadata.php)
- Ha minden rendben megy, akkor az IdP bekerül a Resource_Registry-be, ezáltal a föderációs metaadatba is.
- Az előző pontban leírt módon be kell állítani a központi metadata feldolgozását.
- Amennyiben a föderációs metaadatban már szerepel a mi IdP-nk is, úgy a föderáció valamelyik, tesztelési célokat szolgáló SP-jénél ki is próbálhatjuk a bejelentkezést.

- **Fontos**, hogy a föderációs Discovery Service óránként generálja újra az IdP-k listáját, így ennyi idő mindenképp szükséges, hogy az új IdP megjelenjen itt, az egyes SP-k pedig két óránként töltik újra a metaadatot, így előfordulhat, hogy azonnal nem fog minden működni, de néhány óra alatt várhatóan beindul. :)
- Tesztelésre használható oldal: https://attributes.eduid.hu
- Ahhoz, hogy a Resource Registry-be is be tudjunk lépni és az IdP további, a föderációra vonatkozó beállításait meg tudjuk ejteni, ehhez az IdP-nek ki kell adnia az alábbi attribútumokat:
  - mail - ez belépés után, manuálisan is beállítható
  - eduPersonPrincipalName
  - schacHomeOrganizationType (az attribútumot hamarosan kivezetjük a kötelező attribútumok közül)
  - eduPersonScopedAffiliation

Attribútumok kezelése

Beállított IdP-nk alapértelmezés szerint azokat az attribútumokat adja ki, melyeket a metaadat alapján az SP kért (Lásd a metadatában a RequestedAttribute elemeket), és egyúttal alapból meg tudta szerezni a felhasználói adatbázisból, esetünkben az LDAP-ból. Mivel néhány attribútum nem szerepel az LDAP-ban, hanem az IdP-ben kell előállítani, így pár helyen módosítanunk kell az alapértelmezett konfiguráción.

A `metadata/saml20-idp-hosted.php` fájlba szerkesszük be az alábbi kódrészlet értelemszerűen módosított változatát. Az `'auth' => 'example-ldap',` sor alatt kezdjük. Fontos, hogy egyúttal a config.php `authproc.idp` részét kikommentezzük, nehogy az ottani sorszámokkal megadott default feladatok bekavarjanak.

```
'AttributeNameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
 'userid.attribute' => 'uid', // Itt adjuk meg, hogy mely, az LDAPból származó attribútum
alapján fogja az IdP kiszámítani az eduPersonTargetedID-t
 'authproc' => array(
                10 => array(
                        'class' => 'core:AttributeMap',
                        'uid' => 'eduPersonPrincipalName'
                //Itt az 'uid' az az attribútum az LDAP-ban, amely a felhasználó azonosítóját
tartalmazza, mert ebből képezzük az eduPersonPrincipalName-t.
                ),
                # 20 => array(
                #         'class' => 'core:AttributeAdd',
                #         'schacHomeOrganizationType' =>
array('urn:schac:homeOrganizationType:hu:university')
                # //Kötelező statikus attribútum az
```

```
[[HREFAttributeSpec#schacHomeOrganizationType|intézmény jellegének]] megfelelően
                # ),
                30 => array(
                        'class' => 'core:AttributeAlter',
                        'subject' => 'eduPersonPrincipalName',
                        'pattern' => '/^.*$/',
                        'replacement' => '${0}@intezmenydomain.hu',
                // Itt adjuk hozzá az intézményi scope-ot az eduPersonPrincipalName már
meglévő értékéhez
                ),
                40 => array(
                        'class' => 'core:AttributeAlter',
                        'subject' => 'eduPersonAffiliation',
                        'pattern' => '/^.*$/',
                        'replacement' => '${0}@intezmenydomain.hu',
                // Itt adjuk hozzá az intézményi scope-ot az eduPersonAffiliation már meglévő
értékéhez
                ),
                50 => array(
                        'class' => 'core:AttributeMap',
                        'eduPersonAffiliation' => 'eduPersonScopedAffiliation'
                // Az LDAP-ból eduPersonAffiliation-ként érkező attribútumból föderációs
elvárásoknak megfelelően eduPersonScopedAffiliationt készítünk
                ),
                60 => array(
                        'class' => 'core:AttributeAdd',
                        'eduPersonScopedAffiliation' => array('member@intezmenydomain.hu')
                // Az eduPersonScopedAffiliation-ben tesztelés céljából kiadhatjuk member
értéket,
                // így ha LDAP-ból nem jön érték, akkor is láthatjuk, hogy működik az
attribútum kiadás
                ),
                61 => array(
                        'class' => 'core:TargetedID',
                        'nameId' => TRUE,
                ),
               // Itt állítjuk be, hogy az IdP előállítson és kiadhasson állandóazonosítóként
eduPersonTargetedID-t, ha kérik
                70 => array('class' => 'core:AttributeMap',
                        'name2oid'
```

```
                // Az LDAP-os attribútum nevekből itt kreálunk szabványos urn:oid
formátumúakat
                ),
                80 => 'core:AttributeLimit',
              ), // .authproc
       'simplesaml.nameidattribute' => 'eduPersonPrincipalName',
       'attributeencodings' => array(
               'urn:oid:1.3.6.1.4.1.5923.1.1.1.10' => 'raw',
        ),
        'sign.logout' => true
```

- További tudnivalók a Resource Registry-ről, ill. a Föderációs attribútum specifikációról.
- Ha minden rendben ment, akkor a Resource Registry-ben regisztrált IdP-hez tartozó adminisztrációs jogok átkerülnek az IdP technikai gazdájához, s ezzel a folyamat kész is.

SP

Amennyiben IdP-t is beállítottunk, és be is tudunk lépni a Resource Registry-be, úgy nincs más dolgunk, mint az RR-ben új SP-t hozzáadni a föderációhoz, amely a megfelelő átfutási idő után a föderáció minden tagjánál látható is lesz.

Ellenkező esetben (nincs IdP, és nem is tervezünk beállítani), akkor az IdP hozzáadásánál részletezett pontokon kell végig menni a metaadat betöltéséig, s a továbbiakat az említett e-mail címen megbeszélni.

Attribútum scopeok használata

A HREF föderáció IdP-i ún. scopeolt attribútumokat is használnak. Ez a scopeolás azt jelenti, hogy minden egyes IdP csak a saját scopejában ad ki attribútumokat, és a Shibboleth SP-k ezt ellenőrzik is. A scope és az attribútum valódi értéke egy '@' karakterrel kerül elválasztásra (ilyen attribútumok jelenleg: eduPersonScopedAffiliation illetve eduPersonPrincipalName).

A SimpleSAMLphp alapértelmezett telepítése nem szűri a hibásan scopeolt értékeket. Kiegészítő modulként szűrésre használható az NIIF által fejlesztett attributescope modul, ami reményeink szerint rövid távon a hivatalos SimpleSAMLphp kiadás része lehet.

A telepítésről és konfigurációról bővebben itt lehet olvasni: https://github.com/NIIF/simplesamlphp-module-attributescope

- Az `attributescope` modul használata esetén a következőképp kell módosítani a `config/config.php` fájlt:

```
authproc.sp =  array(
                     ...
                    // 49 => array('class' => 'core:AttributeMap', 'oid2name'),
                    50 => array(             'class' => 'attributescope:FilterAttributes'
                    ),
                    ...
                ),
```

Figyeljünk arra, hogy mire a modulhoz ér a vezérlés, az attribútumok nevei *friendlyName* alakúak legyenek (ne pedig *oid*-ok). A példában erre utal a 49-es sor.

SimpleSAMLphp proxy vidyo portálhoz

Vidyo Portal Authentication Proxy

A vidyo portál utolsó fejlesztései lehetővé tették a SAML alapú authentikációt, és authorizációt.

Az implementáció nem teljesen fedi le a SAML feature-öket, az SP implementáció csak egy IdP-vel képes kapcsolatot létesíteni.

A portált a simpleSAMLphp proxy-ként való telepítésével tehetjük egy föderáció tagjává.

simpleSAMLphp telepítése

A simplesamlphp telepítését elvégezzük a dokumentáció szerint.

SSP IdP oldalának konfigurálása, illesztés a Vidyo portál felé

Legelőször is engedélyezni kell az IdP funkciót

config/config.php

```
'enable.saml20-idp' => true,
```

Gyártsuk le az IdP certificate-jét, és rakjuk a *cert* könyvtárba *idp.pem*, illetve *idp.crt* néven.

```
cd cert
openssl req -newkey rsa:2048 -new -x509 -days 3652 -nodes -out idp.crt -keyout idp.pem
```

metadata/saml20-idp-hosted.php

```
'auth' => 'default-sp',
'privatekey' => 'idp.pem',
'certificate' => 'idp.crt',
),
```

A vidyo portál admin felületéről le kell tölteni a portál metaadatát, és el kell menteni a metadata könyvtárba.

```
metadata/vidyo-sp.xml
```

Erre hivatkozni kell a *config/config.php*-ben is:

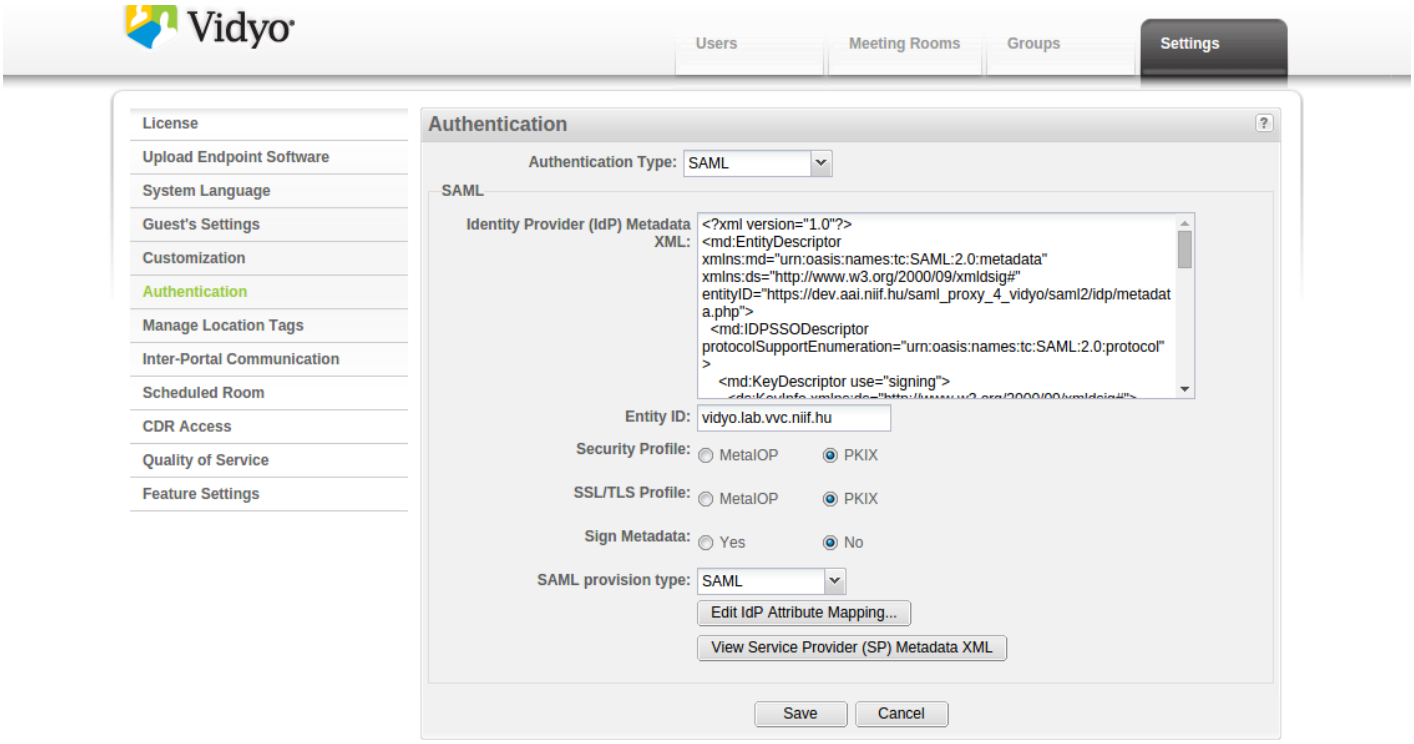
```
'metadata.sources' => array(
    ...
    array('type' # > 'xml', 'file' > 'metadata/vidyo-sp.xml'), // vidyo sp
    ... ),
```

Vidyo admin portál

A portálon be kell állítani,

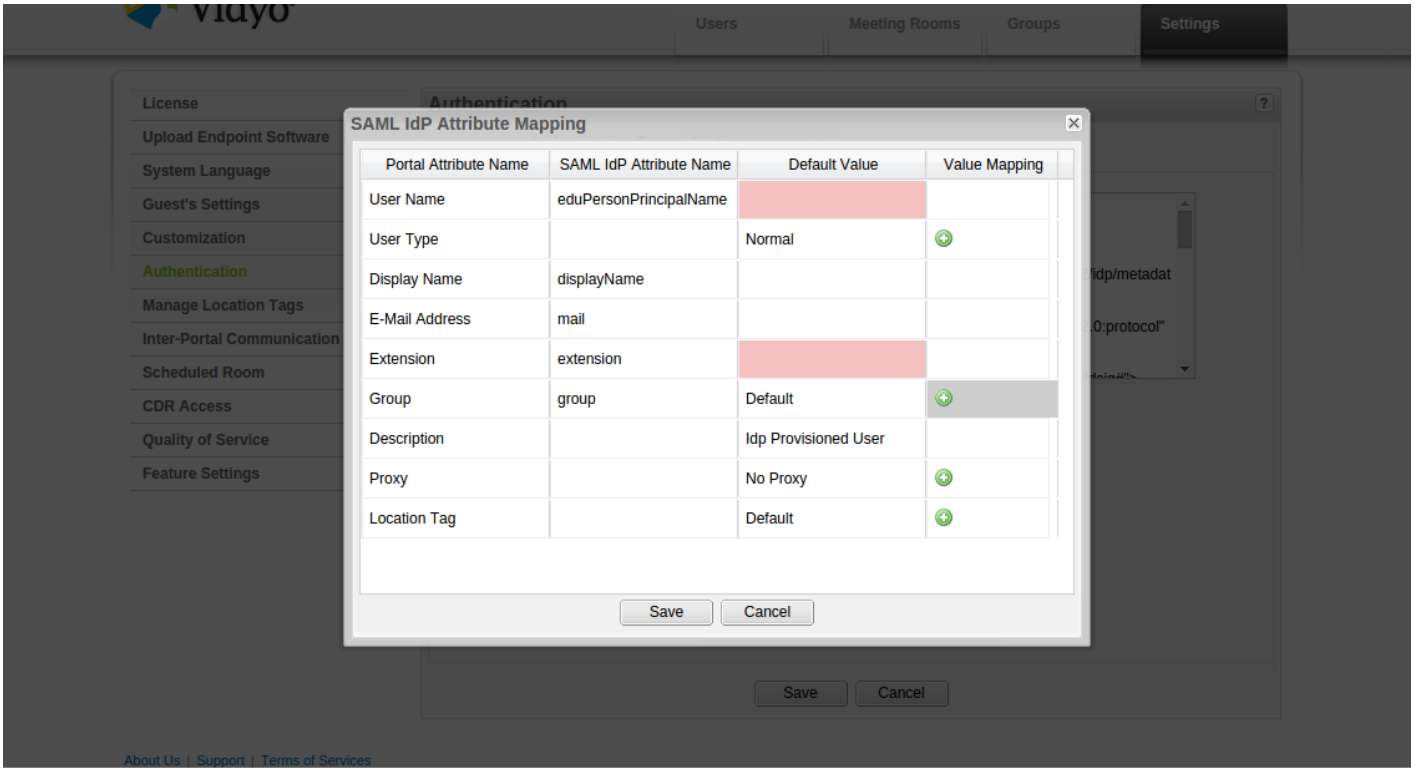
- hogy az azonosítás SAML alapú legyen, *Authentication Type*
- fel kell tölteni az IdP metaadatát, ezt az ssp telepítés *saml2/idp/metadata.php* oldaláról tölthetjük le. *Identity Provider (IdP) Metadata XML*
- be kell állítani az auto provisioninget, *SAML provision type*

Az előző fejezetben említett portál metaadatát ezen az oldalon érjük el. *View Service Provider (SP) metadata XML* Össze kell illeszteni a SAML rétegből jövő attribútumokat a Vidyo portál által használt adatmodellel. *Edit IdP Attribute Mapping...*

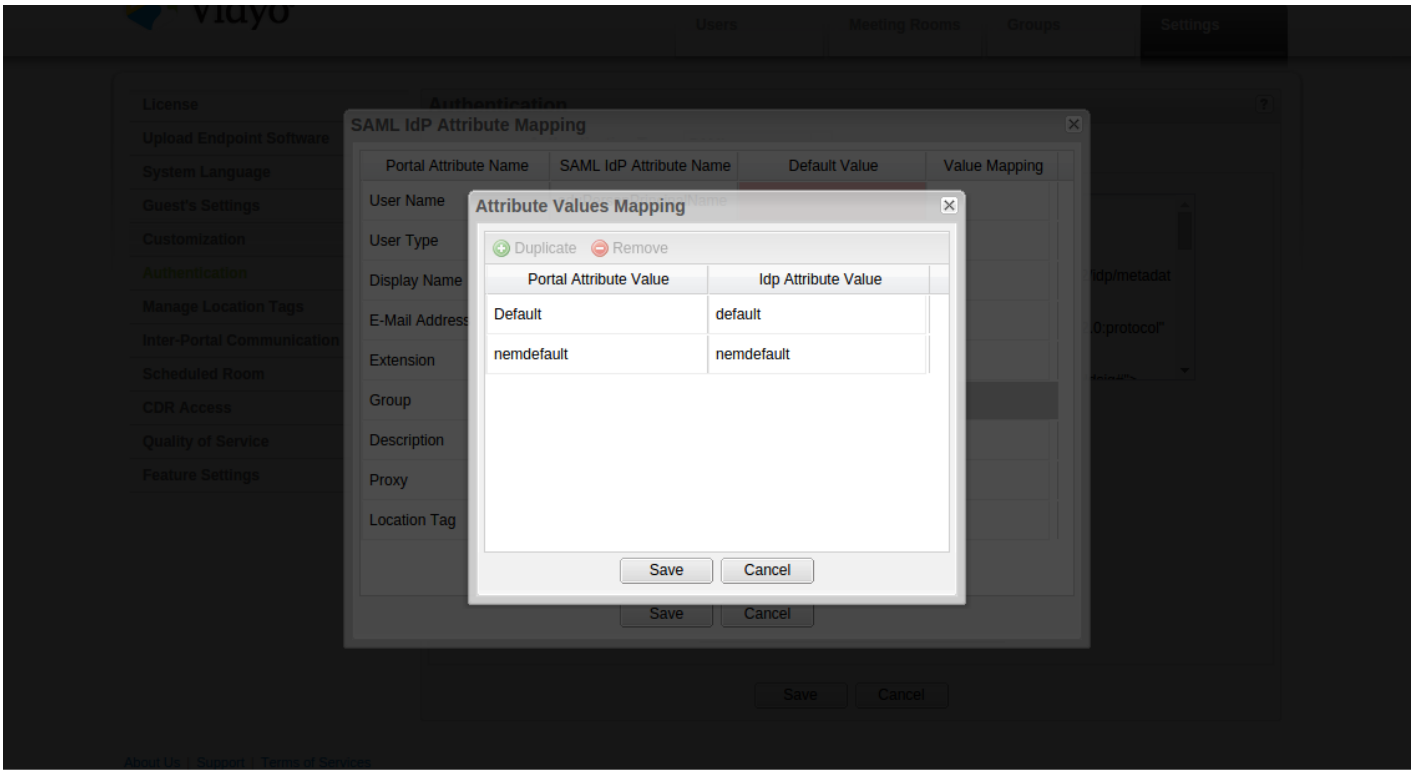


A *SAML IdP Attribute Name* oszlopokba az SSP-től kapott attribútum neveket kell írni. Ha a proxy IdP oldalán a példa szerint állítottuk be az *AttributeMap* szűrőt, akkor itt az attribútumok friendly nevét kell beírnunk. Tipp:

https://github.com/simplesamlphp/simplesamlphp/blob/master/attributemap/name2oid.php



Bizonyos attribútumoknál lehetőség van érték mapping-re is, tipikusan csoport, vagy típus jellegű attribútumoknál, ahol a kapott attirbútumok értéke alapján történik a megfeleltetés.



SSP SP oldalának konfigurálása, illesztés a föderációba

A proxy egyik oldala a föderáció felé, mint SP viselkedik. Az authsource-ot 'default-sp'-nek nevezzük el, erre kell hivatkozni a későbbiekben az IdP konfigurációban.

A *config/config.php* file-ba

Hogy a vidyo portál, és egyé authentikációs szűrők futtatásakor az attribútum megfeleltetéseknél ne okozzanak gondot az oid formátumú attribútum nevek, mielőtt kiadjuk őket, az *AttributeMap* szűrő segítségével alakítsuk át az attribútum neveket.

config/config.php

```
'authproc.sp' => array(
   ...
   200 # > array('class' > 'core:AttributeMap', 'oid2name'),
   ...
),
```

Ha még nem tettük meg, rakjunk ide is certificate-et.

```
cd cert
openssl req -newkey rsa:2048 -new -x509 -days 3652 -nodes -out sp.crt -keyout sp.pem
```

és könyveljük be a *config/authsrouces.php* -ba

```
'default-sp' => array(
    'saml:SP',
 ...
    'privatekey'  => 'sp.pem',
    'certificate' => 'sp.crt',
 ...
```

metadata

Az SP-t regisztráljuk be a kívánt föderációba a föderáció által megadott szabályok alapján.

metarefresh

Hogy a metadadatok mindig napra készek legyenek, gondoskodjunk a metarefresh és cron modul beállításáról.

A konfigurációs file-okat a config könyvtárba kell elhelyezni a sablonokat a modulok config-templates alkönyvtáraiban találjuk meg.

A modulok bekapcsolásáról a rendszer konfigurációban rendelkezhetünk a legegyszerűbben.

config/config.php

```
'module.enable' => array(
        'cron' => TRUE,
        'metarefresh' => TRUE,
),
```

Alkalmazások samlizálást segít? teszt IdP simplesamlPHP segítségével

1. Telepítünk egy SSP egyedet valamelyik szerverünkre.
2. Engedélyezzük a **userpass** auth forrás modult.

```
touch modules/exampleauth/enable
```

3. Elrendezzük a metadatákat.
4. Az authsources.php file-t valahogy így alakítjuk ki:

```
<?php

$config = array(

        // This is a authentication source which handles admin authentication.
        'admin' => array(
                // The default is to use core:AdminPassword, but it can be replaced with
                // any authentication source.

                'core:AdminPassword',
        ),

        'multi' => array(
            'multiauth:MultiAuth',

            /*
             * The available authentication sources.
             * They must be defined in this authsources.php file.
             */
            'sources' => array('projekt1', 'projekt2'),
        ),
        'projekt1' => array(
                'exampleauth:UserPass',
                'tesztuser1:tesztuser1' => array(
```

```
                        'eduPersonPrincipalName' => array('tesztuser1@example.com'),
                        'uid' => array('tesztuser1'),
                        'cn' => "Nemecsek Ernő",
                        'sn' => "Nemecsek",
                        'givenName' => "Ernő",
                        'mail' => "devnull@example.com",
                        'homePostalAddress' => "1234 Budapest, Nincsisilyen utca 2.",
                        'schacDateOfBirth' => "19751221",
                        'schacPlaceOfBirth' => "Budapest",
                        'niifPersonMothersName' => "Törőcsik Mari",
                        'niifPersonResidentalAddress' => "3456 Nagyabajom, Mitírjakide utca
2.",
                ),
                'tesztuser2:tesztuser2' => array(
                        'eduPersonPrincipalName' => array('tesztuser2@example.com'),
                        'uid' => array('tesztuser2'),
                        'cn' => "Nemecsek Ernő 2",
                        'sn' => "Nemecsek",
                        'givenName' => "Ernő",
                        'mail' => "devnull@example.com",
                        'homePostalAddress' => "1234 Budapest, Nincsisilyen utca 2.",
                        'schacDateOfBirth' => "19751221",
                        'schacPlaceOfBirth' => "Budapest",
                        'niifPersonMothersName' => "Törőcsik Mari",
                        'niifPersonResidentalAddress' => "3456 Nagyabajom, Mitírjakide utca
2.",
                ),
        ),
       'projekt2' => array(
                'exampleauth:UserPass',
                'tesztuser1:tesztuser1' => array(
                        'eduPersonPrincipalName' => array('tesztuser1@example.com'),
                        'uid' => array('tesztuser1'),
                        'cn' => "Nemecsek Ernő",
                        'sn' => "Nemecsek",
                        'givenName' => "Ernő",
                        'mail' => "devnull@example.com",
                        'homePostalAddress' => "1234 Budapest, Nincsisilyen utca 2.",
                        'schacDateOfBirth' => "19751221",
                        'schacPlaceOfBirth' => "Budapest",
```

```
                        'niifPersonMothersName' => "Törőcsik Mari",
                        'niifPersonResidentalAddress' => "3456 Nagyabajom, Mitírjakide utca
2.",
                ),
                'tesztuser2:tesztuser2' => array(
                        'eduPersonPrincipalName' => array('tesztuser2@example.com'),
                        'uid' => array('tesztuser2'),
                        'cn' => "Nemecsek Ernő 2",
                        'sn' => "Nemecsek",
                        'givenName' => "Ernő",
                        'mail' => "devnull@example.com",
                        'homePostalAddress' => "1234 Budapest, Nincsisilyen utca 2.",
                        'schacDateOfBirth' => "19751221",
                        'schacPlaceOfBirth' => "Budapest",
                        'niifPersonMothersName' => "Törőcsik Mari",
                        'niifPersonResidentalAddress' => "3456 Nagyabajom, Mitírjakide utca
2.",
                ),
        ),
);
```

SimpleSAMLphp NIIF ldap séma mapping

A simpleSAMLphp különböző attribútum mappinget használ az attribúmnevek átfordításaihoz. A href ldap sémához még nincs, ezt a két file tartalmazza az oid - name oda-vissza mapping-et. Az attributemap könyvtárban van a helyük. A config.php authproc szabályai között kell felvenni őket, amikor szükség van rá.

config/config.php

```
...
        'authproc.sp' => array(
...
                11 => array(
                        'class' => 'core:AttributeMap', 'oid-href'
                ),
...
   ),
...
```

attributemap/href-oid.php

```
<?php

/**
*  Hungarian Research and Education Federation AttributeSchema representation
*  source: [https://wiki.aai.niif.hu/images/3/35/99-
niifschema.ldif](https://wiki.aai.niif.hu/images/3/35/99-niifschema.ldif)
*  @author: Szabó Gyula, aai.sztaki.hu  <gyufi@sztaki.hu>
*
*/

$attributemap = array(
        'niifPersonCityOfBirth' => 'urn:oid:1.3.6.1.4.1.11914.0.1.155',
        'niifPersonDateOfBirth' => 'urn:oid:1.3.6.1.4.1.11914.0.1.152',
        'niifPersonActivityStatus' => 'urn:oid:1.3.6.1.4.1.11914.0.1.153',
        'niifPersonJoinDate'  => 'urn:oid:1.3.6.1.4.1.11914.0.1.169',
```

```
        'niifPersonOrgID' => 'urn:oid:1.3.6.1.4.1.11914.0.1.154',
        'niifCertificateSubjectDN' => 'urn:oid:1.3.6.1.4.1.11914.0.1.151',
        'niifEduPersonFacultyDN' => 'urn:oid:1.3.6.1.4.1.11914.0.1.161',
        'niifPersonPosition' => 'urn:oid:1.3.6.1.4.1.11914.0.1.167',
        'niifStatus' => 'urn:oid:1.3.6.1.4.1.11914.0.1.1',
        'niifPersonIdentityNumber' => 'urn:oid:1.3.6.1.4.1.11914.0.1.158',
        'niifTitle' => 'urn:oid:1.3.6.1.4.1.11914.0.1.2',
        'niifCertificateSHA1Fingerprint' => 'urn:oid:1.3.6.1.4.1.11914.0.1.173',
        'niifEduPersonAttendedCourse' => 'urn:oid:1.3.6.1.4.1.11914.0.1.164',
        'niifEduPersonArchiveCourse' => 'urn:oid:1.3.6.1.4.1.11914.0.1.171',
        'niifEduPersonHeldCourse' => 'urn:oid:1.3.6.1.4.1.11914.0.1.172',
        'niifPrefix' => 'urn:oid:1.3.6.1.4.1.11914.0.1.0',
        'niifPersonDegree' => 'urn:oid:1.3.6.1.4.1.11914.0.1.166',
        'niifEduPersonFaculty' => 'urn:oid:1.3.6.1.4.1.11914.0.1.160',
        'niifEduPersonMajor' => 'urn:oid:1.3.6.1.4.1.11914.0.1.162',
        'niifPersonQuitDate' => 'urn:oid:1.3.6.1.4.1.11914.0.1.170',
        'niifPersonMothersName' => 'urn:oid:1.3.6.1.4.1.11914.0.1.157',
        'niifEduPersonAcademicYear' => 'urn:oid:1.3.6.1.4.1.11914.0.1.163',
        'niifPersonCountyOfBirth' => 'urn:oid:1.3.6.1.4.1.11914.0.1.156',
        'niifUniqueId' => 'urn:oid:1.3.6.1.4.1.11914.0.1.3',
        'niifPersonPrefix' => 'urn:oid:1.3.6.1.4.1.11914.0.1.165',
        'niifActiveMemberOf' => 'urn:oid:1.3.6.1.4.1.11914.0.1.168',
        'niifPersonResidentalAddress' => 'urn:oid:1.3.6.1.4.1.11914.0.1.159',
        'niifIDPrefix' => 'urn:oid:1.3.6.1.4.1.11914.0.1.100',
);
?>
```

/attributemap/oid-href.php

```
<?php

/**
*  Hungarian Research and Education Federation AttributeSchema representation
*  source: [https://wiki.aai.niif.hu/images/3/35/99-
niifschema.ldif](https://wiki.aai.niif.hu/images/3/35/99-niifschema.ldif)
*  @author: Szabó Gyula, aai.sztaki.hu  <gyufi@sztaki.hu>
*
*/

$attributemap = array(
```

```
        'urn:oid:1.3.6.1.4.1.11914.0.1.155' => 'niifPersonCityOfBirth',
        'urn:oid:1.3.6.1.4.1.11914.0.1.152' => 'niifPersonDateOfBirth',
        'urn:oid:1.3.6.1.4.1.11914.0.1.153' => 'niifPersonActivityStatus',
        'urn:oid:1.3.6.1.4.1.11914.0.1.169' => 'niifPersonJoinDate' ,
        'urn:oid:1.3.6.1.4.1.11914.0.1.154' => 'niifPersonOrgID',
        'urn:oid:1.3.6.1.4.1.11914.0.1.151' => 'niifCertificateSubjectDN',
        'urn:oid:1.3.6.1.4.1.11914.0.1.161' => 'niifEduPersonFacultyDN',
        'urn:oid:1.3.6.1.4.1.11914.0.1.167' => 'niifPersonPosition' ,
        'urn:oid:1.3.6.1.4.1.11914.0.1.1' => 'niifStatus',
        'urn:oid:1.3.6.1.4.1.11914.0.1.158' => 'niifPersonIdentityNumber',
        'urn:oid:1.3.6.1.4.1.11914.0.1.2' => 'niifTitle',
        'urn:oid:1.3.6.1.4.1.11914.0.1.173' => 'niifCertificateSHA1Fingerprint',
        'urn:oid:1.3.6.1.4.1.11914.0.1.164' => 'niifEduPersonAttendedCourse',
        'urn:oid:1.3.6.1.4.1.11914.0.1.171' => 'niifEduPersonArchiveCourse',
        'urn:oid:1.3.6.1.4.1.11914.0.1.172' => 'niifEduPersonHeldCourse',
        'urn:oid:1.3.6.1.4.1.11914.0.1.0' => 'niifPrefix',
        'urn:oid:1.3.6.1.4.1.11914.0.1.166' => 'niifPersonDegree' ,
        'urn:oid:1.3.6.1.4.1.11914.0.1.160' => 'niifEduPersonFaculty',
        'urn:oid:1.3.6.1.4.1.11914.0.1.162' => 'niifEduPersonMajor',
        'urn:oid:1.3.6.1.4.1.11914.0.1.170' => 'niifPersonQuitDate' ,
        'urn:oid:1.3.6.1.4.1.11914.0.1.157' => 'niifPersonMothersName',
        'urn:oid:1.3.6.1.4.1.11914.0.1.163' => 'niifEduPersonAcademicYear',
        'urn:oid:1.3.6.1.4.1.11914.0.1.156' => 'niifPersonCountyOfBirth',
        'urn:oid:1.3.6.1.4.1.11914.0.1.3' => 'niifUniqueId',
        'urn:oid:1.3.6.1.4.1.11914.0.1.165' => 'niifPersonPrefix' ,
        'urn:oid:1.3.6.1.4.1.11914.0.1.168' => 'niifActiveMemberOf' ,
        'urn:oid:1.3.6.1.4.1.11914.0.1.159' => 'niifPersonResidentalAddress',
        'urn:oid:1.3.6.1.4.1.11914.0.1.100' => 'niifIDPrefix',
);
?>
```

Single Logout in Shibboleth IdP

Important notes on third party cookies

In some browsers, the IFrame-driven front-channel logout doesn't work due to the browser blocking third party cookies.

Every cookie which is sent to a foreign domain via img, iframe, script, etc. tags is considered to be third party, so the session cookie of the SP software in a foreign domain is third party cookie when it is sent in an IFrame. By blocking these cookies, the SP doesn't receive the session cookie and thus it could stop processing the logout request at this point.

Additional links:

- Shibboleth-dev thread on the issue
- How to disable third party cookies in firefox
- Additional explanation in Mozilla Bugzilla
- Same origin policy for cookies
- Further information on third party cookie handling

"Although any third-party cookie restrictions are not a sufficient method to prevent cross-domain user tracking, they prove to be rather efficient in disrupting or impacting the security of some legitimate web site features, most notably certain web gadgets and authentication mechanisms."

Why service providers might need the session cookie

Most of the services do not need the session cookie itself, they only need the NameIdentifier, which is carried by the logout request, so back-channel logout requests are enough for them. But there might be service providers which do not implement back-channel bindings (eg. SimpleSAMLphp), or need front-channel application notification.

Why not fully back-channel?

SAML profiles specification (section 4.4.3.1) clearly states that front-channel should be preferred when sending the logoutrequest to the session authority (IdP). If the user interface is generated by the IdP, it could inform the user about the whole logout process, and each SP response. If the SP

would use back-channel logoutrequest, the IdP's response would only contain minimal information (ie. success or failure), and this is not desirable. Also, the IdP would need to execute back-channel requests in parallel and synchronize them with the originating request, so this would make the processing code much more complex.

Technical solution

Our proposal is to prefer back-channel endpoints at the service provider side, unless your application needs to be notified via front-channel. For example,

- if your application behind your SP needs the session cookie with the notification, use only front-channel bindings in the SP metadata,
- otherwise use only back-channel binding in the SP metadata.

By these mutually exclusive endpoint sets, the SP can clearly advise the IdP which binding it should use when contanting this SP. Thus on the IdP side, both implementations need to be available.

Non-technical solution

Another option would be to add a new requirement for your end users. You can claim that banning third-party cookies is unsupported (because it breaks SLO), just like disabling all cookies (which breaks SSO). Convincing your users (and the Shibboleth developers to accept this solution) might be dubious, though.

Features

- Implements SAML2 Single Logout profile
- If initiated by an SP, user must confirm the single logout process: one can choose to logout only from the initiating SP and the IdP.
- Highly customizable front-channel logout interface which leverages javascript and asynchronous operations in order to provide a clean, simple UI.
- UI is usable with javascript disabled.
- Supports localized SP name lookup via Organization elements in SAML metadata .
- Fallback to back-channel logout if front-channel is not supported by the SP.
- Supports Shibboleth SSO sessions (if the SP initiates sessions using Shibboleth1.3 protocol, but supports SAML2 logout).
- Supports full back-channel operation.
- Supports IdP-initiated Single Logout.

UI customization

The UI is located in two JSP files:

- `sloQuestion.jsp` the user chooses whether she wants to logout from all service providers or just from the provider where she came from.
- `sloController.jsp` is the logout UI where every session participant and their corresponding logout status is shown. At the end of the logout process, the user is notified if the single logout was completed.

How it works

SLOServlet

The heart of the logout process is the `SLOServlet`. This servlet is responsible for these actions:

- rendering the logout question and controller page
- initiating front-channel or back-channel logout to one SP (`SLOServlet?action&entityID=...`)
- returning the logout status as a JSON string (`SLOServlet?status`)

With javascript

The controller renders a page where an iframe is placed for every active session participant. This iframe calls the `SLOServlet?action&entityID=...` URL where the logout request is issued for the given session participant. If the request is front-channel, the iframe will make a front-channel SAML message exchange with the peer (using HTTP-Redirect or POST bindings).

The status of the single logout process is queried via asynchronous requests. The status response from `SLOServlet` is a JSON array. This JSON array contains one object with the `entityID` and `logoutStatus` properties for each session participant.

The logout status can be one of the followings:

- `LOGGED_IN`: logout is not initiated for this participant yet.
- `LOGOUT_ATTEMPTED`: logout was initiated.
- `LOGOUT_FAILED`: logout failed.
- `LOGOUT_UNSUPPORTED`: SAML2 logout is not supported by the participant (the metadata does not contain the necessary endpoints).
- `LOGOUT_TIMED_OUT`: timed out waiting for a response.
- `LOGOUT_SUCCEEDED`: logout was successful.

Status queries are issued using exponential backoff timing, until the timeout is reached. Please see the `sloController.jsp` for the exact timing used.

Without javascript

Controller renders an HTML page with `<noscript>` tags. There is one link for each session participant opening up in new window/tab, which can initiate the logout process for that particular

SP. Depending on the current logout status, several other controls are enabled on the page:

- `Refresh` button, which will reload the controller HTML with the current status icons to follow the overall logout process.
- `Logout failed` message when logout process was finished, and there was at least one failed session participant.
- `Logout succeeded` message when logout process was finished, and all session participants completed the logout.

IdP-initiated Logout (available since v2.1.3-slo2)

The user can initiate their logout process from the IdP (the URL is `/idp/Logout`). IdP-initiated logout has a clear advantage over SP-initiated logout, because the URL and the UI is fully independent from the current SP software used, thereby providing a unique logout URL for all users of the given IdP.

Non-trivial settings

Security

SAML Single Logout Profile requires the logout requests and responses to be signed or otherwise authenticated. Without this, a user session could be DOS-ed knowing the NameID.

You have two choices

- instruct the SP to sign messages
- configure the IdP not to require authentication of logout messages (and bear with possible DOS-attacks)

Signing messages

Signing can be turned on by setting the **`signing`** property to **`front`** (for front-channel only) or **`true`** in the `ApplicationDefaults` or `ApplicationOverride` element in shibboleth2.xml.

!!! note

```
Signing messages is normally unnecessary for back-channel, as the transport is usually
authenticated with the certificates in the metadata. However, for back-channel logout it is
the IdP who initiates the HTTP connection to the SP, and it is the **webserver**, who answers
the request. Because of the different needs, the webserver almost always uses a different
certificate (a server certificate signed by a well-known CA) than the SP (possibly self-
signed, client certificate). Therefore the SP must sign back-channel messages as well to
authenticate itself to the IdP. Unfortunately, you can only enable signing all (otherwise
```

```
transport protected) messages, and this may affect performance.
```

Not requiring peer authentication

Message issuer authentication can be turned off by changing the security policy of processing Single Logout messages. You can do this by commenting out the following line from the block `SAML2SLOSecurityPolicy` at `relying-party.xml`:

```
<security:Rule xsi:type="security:MandatoryMessageAuthentication" />
```

Session lifetime

IdP session lifetime must be longer than any SP session lifetime. Otherwise, if an SP session outlives the IdP session and the user reauthenticates for a new session for other SPs, logout would not terminate session at the first SP.

The IdP can limit the maximum lifetime of the SP session by using the (optional) `SessionNotOnOrAfter` property in the SAML2 authentication statement. SAML1.1 does not have this feature, so **you cannot limit the session lifetime for SPs using Shibboleth SSO protocol.**

> This can be set in the `relying-party.xml` by specifying the number of milliseconds in the `maximumSPSessionLifetime` attribute of the `SAML2SSOProfile` configuration.

Required changes in the IdP

Name identifier caching in IdP session

In the LogoutRequest the IdP must reference the current user's name identifier. This name identifier is issued as part of the SSO process. In order to efficiently retrieve this information, the IdP should cache the name identifier in the IdP session information object.

Associated ticket: SIDP-336

Session indexing

On receiving a LogoutRequest from a session participant, the IdP must be able to retrieve the IdP session associated with the principal. Session participants use the issued name identifier to identify the principal. The IdP session object can be indexed (and then retrieved of course) by any arbitrary

unique key, so we use the name identifier value to index the session.

Associated ticket: SIDP-338

IdP Session invalidation

Currently there is a bug in the IdP implementation which causes the IdP sessions to outlive the session removal.

Associated ticket: SIDP-333

How to use

How to build

- install the maven2 build tool
- source code is available from our git repository
  - you can use the convenient snapshot links below to start playing
  - if you are brave enough, feel free to clone the whole repository and track our development branches (frontchannel-slo for the idp project and slo-configuration branch for the shibboleth-common project)
- compile the shib-java-common project first with the `mvn -DskipTests install` command (the first build might take quite a long time if you haven't used maven before)
- compile the java-idp project with the same maven command
- install the `java-idp/target/shibboleth-identityprovider-{version}-bin.zip` binary package the same way as you'd install a vanilla Shibboleth IdP bundle

Released versions

- download the latest binary snapshot version from our software distribution site

v2.2.0-slo10

- fix configuration templates
- source code snapshots
  - shibboleth-common-1.2.0-slo2
  - shibboleth-identityprovider-2.2.0-slo10

v2.2.0-slo9

- allow EncryptedID to be used in the initiating request (patch contributed by Michael Simon from Karlsruher Institut für Technologie)
- expose method for programatical back-channel logout
- source code snapshots
  - shibboleth-common-1.2.0-slo2
  - shibboleth-identityprovider-2.2.0-slo9

v2.1.5-slo7

- use AttributeConsumingService/ServiceName to feed the logout interface
- source code snapshots
  - java-shib-common-1.1.4-slo2
  - java-idp-2.1.5-slo7

v2.1.5-slo6

- skip session-indexing under error conditions
- source code snapshots
  - java-shib-common-1.1.4-slo2
  - java-idp-2.1.5-slo6

v2.1.5-slo5

- fixed NullPointerException with non-existent or filtered NameIdentifiers
- fixed a flaw in session-indexing logic, use the whole NameIdentifier as the index, not just the value
- source code snapshots
  - java-shib-common-1.1.4-slo2
  - java-idp-2.1.5-slo5

v2.1.5-slo4

- upstream version bump
  - java-shib-common
  - java-idp

v2.1.4-slo4

- updated Shibboleth-core
- fixed NullPointerException introduced by an erroneous merge in v2.1.4-slo3
  - java-shib-common
  - java-idp

v2.1.3-slo3

- support Terracotta clustering
- source code snapshots
  - java-shib-common
  - java-idp

v2.1.3-slo2

- support IdP initiated logout
- source code snapshots
  - java-shib-common
  - java-idp

v2.1.3-slo1

- support SP initiated front- and back-channel logout

Hints

> - Don't forget to include Single Logout endpoints in the IdP metadata
> - Shibboleth SP prior to 2.1 did not include NameID properly in the LogoutRequest, therefore you cannot initiate SLO with Shibboleth SPs older than 2.1
> - Shibboleth SP prior to 2.2.1 answered with Partial logout for back-channel requests due to a bug
> - Shibboleth SP (currently released versions) do not distinguish between Success and Partial logout when showing the UI (see this report for details). This is not needed unless you are using back-channel logout.
> - If you plan to upgrade a clustered IdP to this version, don't forget to check the new tc-config.xml and rebuild the terracotta boot jar

Missing features

- Administrative logout
- Logout the user in the underlying JAAS provider

Attribute Conversion for simpleSAMLphp

This page describes the features of Attribute Conversion and Filtering library for simpleSAMLphp

Introduction

eduGAIN uses Bridging Elements for interconnecting federations. To provide attribute translation and filtering services, an attribute 'mangling' library was developed for the Java-based bridging elements. As simpleSAMLphp can also be used as an eduGAIN bridging element, the conversion and filtering library was ported to PHP.

Beyond eduGAIN, you can use this module for every IdP or SP operating mode (shib13 SP/IdP, saml2 SP/IdP) of simpleSAMLphp in order to provide more powerful attribute conversion and filtering capabilities.

Download and support

You can download the module from here. The module is in beta stage, it needs broader community review. It is not yet recommended for production environments.

If you have any questions regarding the module, please write to '*aai aT niif _dOt hu*.

For changelogs please visit the project repository.

Compatibility

eduGAIN

This library is intended to be configuration-compatible with the eduGAIN Attribute_Conversion_for_eduGAIN Java library. The module can read the eduGAIN converter and filter engine XML configuration files and should operate the same way as the Java one.

Configuration files

The eduGAIN attribute converter and filter module defines its own XML schema for attribute conversion and attribute filtering purposes. See the Attribute_Conversion_for_eduGAIN page for more information on attribute rules.

Using the module

This module has a working name `edugain`. As this module only addresses the attribute translation part of the 'eduGAIN-problem', it might be renamed later.

Enabling the simpleSAMLphp module

This module depends on the **xsl** php extensions (more specifically, the *XSLTProcessor* class), so make sure it is properly configured.

The module can be enabled by creating an empty file named `modules/edugain/default-enable`.

simpleSAMLphp module configuration

EduGAIN is available for simpleSAMLphp as an authentication processing filter: *edugain:Attributes*. The Attributes processing filter takes the following configuration properties:

```
'authproc' => array(
  50 => array(
   'class' => 'edugain:Attributes',
   'mode' => 'idp',
   'converterconfig' => '/path/to/AttributeConverter.xml',
   'filterconfig' => '/path/to/AttributeFilter.xml',
   'cache' => true
  )
)
```

Configuration parameters for the module

- **class** (required): defines the eduGAIN filter for simpleSAMLphp.
- **mode** (required): configures the way this module operates (`idp` or `sp`). See below for more information on operating modes
- **converterconfig** (optional): configures the path of the attribute converter configuration xml file.

- **filterconfig** (optional): configures the path of the attribute filter configuration xml file.
- **cache** (optional, default: true): enables or disables the internal configuration cache. See the Configuration_cache section below for more.

!!! info

```
If either `converterconfig` or `filterconfig` is omitted, than the relevant part of the module (conversion or filtering respectively) is disabled. Note that **disabling filter means you let all the attributes through**.
```

Operating modes

EduGAIN module can operate in two modes, **idp** or **sp**. This mode affects two behaviors: the conversion-filtering order, and the provider matching.

- in **idp** mode, attribute filter is run **after** conversion, and the RemoteProvider match is done against the SP (or R-BE in eduGAIN bridged environment) which initiated the SSO session .
- in **sp** mode, attribute filter is run **before** conversion, and the RemoteProvider match is done against the IdP (or H-BE in eduGAIN bridged environment) which released the attributes to our simpleSAMLphp SP.

Configuration file

The simpleSAMLphp eduGAIN module reads the eduGAIN XML configuration format and transforms it into php arrays using XSL transformation. The submodules (*edugain:SplitMerge* and *edugain:Filter*) are configured automatically by the edugain:Attributes class.

PHP configuration interface for these filters are not supported at the moment and may be subject to change, so please use the XML configuration.

Configuration cache

The XML reading is very time-consuming but conversion and filtering rules should be evaluated on every request. Because of that, the eduGAIN module can cache the XML configuration into a serialized PHP array, which is stored locally in a directory named `cache`. If the XML file is not updated since the last cache file generation then the cache is used and the XML parsing part is skipped. Cache file name is computed according to the following:

```
md5(full_configuration_file_path).cache.php
```

!!! info

```
Enabling the cache is strongly recommended in production environments.
```

Differences between the Java and the PHP implementations

- There is no **CustomRule** for attribute conversion. One can use simpleSAMLphp authentication processing filter API to implement arbitrary conversion rules.
- **LocalProvider** matching is unsupported in simpleSAMLphp. Unfortunately when simpleSAMLphp is in bridging mode (using the SP module to protect an IdP), the IdP processing filters do not see the peer entity of the SP module. However, you can achieve the correct behavior by putting one *edugain:Attributes* processing filter in the SP configuration and use **RemoteProvider** matches to filter and convert attributes there.
- You don't need to use a separate attribute name mapper, because simpleSAMLphp contains built-in **name2oid**,**oid2name**, **name2urn** and **urn2name** methods, which provide the same functionality.
- Regular expressions are somewhat different in PHP. The eduGAIN module uses perl-compatible regular expressions (see preg_match documentation for details).

SimpleSAMLMixedMetadata

Ez a vázlatos leírás egy olyan SimpleSAMLphp IdP (vagy SP) konfigurálásában kíván segítséget nyújtani, amely az alábbi metaadatforrásokat használja:

- kézzel szerkesztett saml20-sp-remote (vagy saml20-idp-remote) például Google Apps vagy Office365 használatához;
- az intézményi metadata halmazt, azaz a belső SP-k (esetleg teszt IdP-k) halmazát;
- a magyar eduID.hu föderációban levő entitásokat;
- és az eduGAIN-ben levő entitásokat.

Ez utóbbi kettőt a föderáció **MDX** szolgáltatása biztosítja leghatékonyabban.

Metaadatforrások beállítása

A **config/config.php** állományban cseréljük le a *metadata.sources* részt az alábbira:

```
'metadata.sources' => array(
        array('type' => 'flatfile'),
        array('type' => 'flatfile','directory' => 'metadata/metarefresh'),
        array('type' => 'mdx', 'server' => 'http://mdx.eduid.hu', 'cachedir' =>
'/var/simplesamlphp/mdx-cache', 'cachelength' => 7200,
'validateFingerprint' => '91:81:AD:2B:F1:C1:4E:47:93:A2:9D:49:34:B7:77:62:4F:2F:98:43'
),
),
```

Az MDX cache könyvtárat és a statikus metaadatok könyvtárát hozzuk létre, és tegyük a webszerver által írhatóvá:

```
sudo mkdir /var/simplesamlphp/{mdx-cache,metadata/metarefresh}
sudo chgrp www-data /var/simplesamlphp/{mdx-cache,metadata/metarefresh}
sudo chmod g+w /var/simplesamlphp/{mdx-cache,metadata/metarefresh}
```

Statikus metaadatok

A fenti szakaszban a dinamikus metaadatok konfigurációját már megadtuk, a statikus metaadatokat viszont a *cron* modul által meghívott *metarefresh* program végzi, így ezeket is

konfigurálni kell.

> **Note**
>
> Ha nem használunk belső SP-ket, akkor a szakaszban leírt konfigurációra nincs szükségünk, **készen vagyunk**!

config/config-metarefresh.php:

```
<?php

$config = array(
   'sets' => array(
       'href' => array(
           'cron'      => array('hourly'),
           'sources'   => array(
               array(
                   'src' => 'http://metadata.eduid.hu/2011/niifi.xml',
                   'validateFingerprint' =>
'FE:AE:0B:E8:FB:59:ED:F7:CB:7F:69:DF:19:4F:8B:6D:C7:F6:96:66',
               ),
           ),
           'expireAfter'   => 60*60*24*7, // Maximum 4 days cache time.
           'outputDir'     => 'metadata/metarefresh/',
           'outputFormat'  => 'flatfile',
       ),
   ),
);
```

config/module_cron.php:

```
<?php
/*
 * Configuration for the Cron module.
 */

$config = array (

        'key' => 'eztajelszotcsereldle',
        'allowed_tags' => array('daily', 'hourly', 'frequent'),
```

```
        'debug_message' => TRUE,
        'sendemail' => FALSE,

);
```

Engedélyezzük a *cron* és a *metarefresh* modulokat:

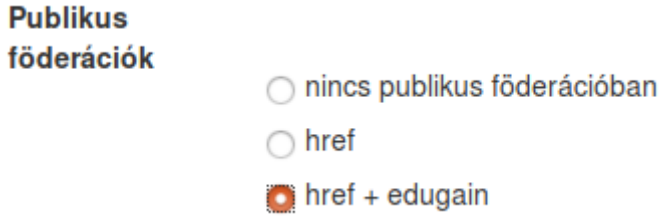
```
sudo touch /var/simplesamlphp/modules/{cron,metarefresh}/enable
```

Hozzuk létre azt a rendszer cron bejegyzést, amely időzítve meghívja a SimpleSAMLphp cron modulját:

```
echo '03 * * * * www-data curl --silent
"https://idp.example.org/simplesamlphp/module.php/cron/cron.php?key=eztajelszotcsereldle&tag=h
ourly" \
>/dev/null 2>&1' > sudo tee /etc/cron.d/simplesamlphp
```

Ezzel az IdP-nk készen áll az eduGAIN konföderációba való publikálásra. A haladó attribútumkiadás-konfiguráció leírása itt található: SSP EntityCategories.

A Resource Registry felületen belépve állítsuk át az entitást az eduGAIN-ben való publikálásra:



SSP2

Az alábbi lapon megkíséreljük összefoglalni a legfontosabb lépéseket, melyek általános esetben elegendőek ahhoz, hogy működő SimpleSAMLphp (SSP) alkalmazást állítsunk üzembe.

Telepítés

Először is nagyvonalakban leírásra kerülnek az előkészületek ezt követően pedig maga a szoftver telepítése és beállítása.

El?készületek

Ahhoz, hogy problémamentesen telepíthessük SSP alkalmazásunkat, az alábbi szoftverkomponenseknek kell működniük szerverünkön.

- A következő könyvtárakat kiegészítőket telepíteni kell: `wget openssl unzip build-essential libldap2-dev libldap-common`
- PHP futtatására alkalmas webszerver
- PHP környezet (>=8.0)
- A következő PHP kiterjesztéseket engedélyezni kell
  - `posix, date , dom , fileinfo , filter , hash , json , libxml , mbstring , openssl , pcre , session , simplexml , sodium , SPL, zlib, ldap`
  - Adatbázisból történő autentikáció esetén a megfelelő adatbázis-csatolót `mysqli, pdo, pdo_mysql`
  - RADIUS szerveren keresztül történő autentikáció esetén: `radius`
- Információk, certek:
  - Bár a szoftver képes futni mariadb, és redis nélkül, ezek vagy hasonló megoldások használata éles környezetben ajánlott.
    - Amennyiben ezeket szándékozunk használni database connection stringgel kell rendelkezni, redis elérhetőséggel és jelszóval rendelkezni.
    - Az adatbázis szerkezetének kialakítása a használt moduloktól függ, a leggyakoribb a consent modul, ott van is dokumentáció az adatbázis inicializálásáról.
  - Szükséges 2 certpár, az egyik az apachenak a másik az SSP-nek ezutóbbi lehet önalárírt, és nem ajánlott ugyan azt használni.

Composer

A **composer** PHP csomagkezelőt is telepíteni kell (akár forrásból, akár csomagból), hogy telepíteni lehessen a SimpleSAMLphp futásához szükséges PHP library-ket.

Telepítés

Elvégezhető composerrel például a `/var/` mappában:

`composer create-project simplesamlphp/simplesamlphp:2.1.3`

Mappaszerkezet módosítása:

```
mv simplesamlphp simplesamlphp-prod #Tetszőleges átnevezés
mkdir -p simplesamlphp-prod/cert
mkdir -p /var/simplesamlphp-prod/log/stats/
mkdir -p /var/simplesamlphp-prod/mdx-cache/
chown -R www-data:www-data /var/simplesamlphp-prod/mdx-cache/
chown -R www-data:www-data /var/simplesamlphp-prod/log/
chown -R www-data:www-data /var/simplesamlphp-prod/metadata/
```

Composerrel a szükséges modulok telepítése (például LDAP, REDIS, vagy consent):

```
composer require simplesamlphp/simplesamlphp-module-ldap:v2.3.2
composer require predis/predis:2.2.2
composer require simplesamlphp/simplesamlphp-module-consent:1.3.2
```

Ezzel a telepítés, és a szükséges kiegészítők telepítése megtörtént.

Apache konfigurálás

A webről csak a `/var/simplesamlphp-prod/public` könyvtárat kell elérni. **Tilos** a teljes simplesamlphp könyvtárat a DocumentRoot alá tenni!

```
Alias /simplesaml /var/simplesamlphp-prod/public
<Directory /var/simplesamlphp-prod/public>
  Require all granted
</Directory>
```

Simplesamlphp Alapbeállítások

Konfigurációs fájlok

Amennyiben korábbi verziókat használtunk,

jó megközelítés lehet diff segítségével ellenőrizni a különbségeket a korábbi `config.php` fileunk és a `config.php.dist` között.

Ha ilyennel nem rendelkezünk akkor lehet rögtön alapul venni a `config.php.dist`-et és hasonlóképp a metadata-templates mappát.

Konfigurációs fájlok szerkesztése

Baseurlpath beállítása

- Állítsuk be a baseurlpath opciót. Mutasson a telepítés URL-jére, ahol a SimpleSAMLphp elérhető:

`'baseurlpath' => 'https://your.canonical.host.name/simplesaml/',`

Adminisztrációs adatok beállítása

- **Az "admin" felhasználó jelszavát, mellyel webes felületen keresztül be tud lépni a települő SSP-be.**

  ```
  'auth.adminpassword'        => 'ujjelszotirdide',
  ```

- **Titkosítási feladatokhoz szükséges "salt", azaz véletlenszerűen összeálló karaktersorozat**

  ```
  'secretsalt' => 'randombytesinsertedhere',
  ```

  A karaktersorozat előállításában segíthet az alábbi parancs:

  ```
  tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32 count=1
  2>/dev/null;echo
  ```

- **Elérhetőségeket, amely adatok bekerülnek majd a generált metaadatba**

  ```
  'technicalcontact_name'     => 'Gipsz Jakab',
  'technicalcontact_email'    => 'jakab.gipsz@example.org',
  ```

- **Nyelv és időzóna adatok**

  ```
  'language.default'      => 'hu',
  'timezone' => 'Europe/Budapest',
  ```

Az alapadatok megadása után mentsük és zárjuk be a **config.php**-t.

Naplózás beállítása

Alapértelmezetten a SimpleSAMLphp a **syslog**-ba irányítja a naplózást.

Ha fájlba akarunk naplózni, akkor a megfelelő könyvtárhoz biztosítsunk írás jogot a webszerver felhasználónak, és ne felejtsünk el gondoskodni a naplófájlok rotálásáról!

- Naplózási szint beállítása a **config/config.php**-ban

```
'debug' => array(
       'saml' => true,
       'backtraces' => true,
       'validatexml' => false,
),
'logging.level' => SimpleSAML\Logger::DEBUG,
'logging.handler' => 'file',
```

A "SimpleSAML\Logger::DEBUG" a legrészletesebb naplózási beállítás, éles rendszernél nem ajánlott csak hiba keresés esetén.

Modulok engedélyezése

```
'module.enable' => [
 'exampleauth' => true,
 'saml' => false, //
 'core' => null, // Alapértelmezett érték
 'ldap' => true, // 2.x verzióban külön telepíteni és engedélyezni kell az ldap modult.
 'admin' => true, // Ezt szükséges engedélyezni hogy elérhessük az adminisztrációs felületet
],
```

Tanúsítvány készítése

Nem ajánlott a SimpleSAMLphp-hoz és webszerverhez ugyanazt a tanúsítvány használni!

- A SimpleSAMLphp alapértelmezetten a tanúsítványt a **cert** mappában keresi.
- Az alábbi paranccsal egy 10 éves self-signed tanúsítvány generálunk a SimpleSMALphp számára.

```
openssl req -new -newkey rsa:3072 -x509 -days 3652 -nodes -out cert/saml-example-org.crt -keyout cert/saml-example-org.key
```

A fingerprint az alábbi módon kérdezhető le a legegyszerűbben

```
openssl x509 -fingerprint -noout -in cert/saml-example-org.crt
```

Telepítés kész

Amennyiben elkészültünk a fenti lépésekkel, úgy a https://service.example.org/simplesaml/admin címen elérjük a telepített SSP-nk webes adminfelületét.

Identity Provider (IdP) beállítás

Alapbeállítások

IdP engedélyezése: a **config/config.php** fájlban kell a saml20 idp-t "true"-re állítani.

```
'enable.saml20-idp' => true,
```

Metaadat alapok

A beállítandó IdP alapvető paraméterei a `metadata/saml20-idp-hosted.php` fájlban állíthatók. Az alábbi kódrészlet egy minimális, de már működő példát mutat.

```
<?php
$metadata['https://example.org/saml-idp'] = [
   /*
    * The hostname for this IdP. This makes it possible to run multiple
    * IdPs from the same configuration. '__DEFAULT__' means that this one
    * should be used by default.
    */
   'host' => '__DEFAULT__',
   /*
    * The private key and certificate to use when signing responses.
    * These can be stored as files in the cert-directory or retrieved
    * from a database.
    */
   'privatekey' => 'example.org.pem',
   'certificate' => 'example.org.crt',
   /*
    * The authentication source which should be used to authenticate the
    * user. This must match one of the entries in config/authsources.php.
    */
   'auth' => 'example-ldap',
];
```

A fentebb hivatkozott certeket korábban létrehoztuk, de az example-ldap auth forrást még nem:

LDAP autentikáció

Javasolt az LDAP-ban egy olyan bejegyzést létrehozni az IdP számára, amely olvasni tudja a felhasználóknak a föderációban használt attribútumait. Az azonosítás alapértelmezett módon a felhasználó nevében történő újra bind-olással történik, így a jelszóhoz nem kell hozzáférést adni.

Ahhoz, hogy megadhassuk az LDAP-hoz tartozó beállításokat, a `config/authsources.php` fájlt kell szerkesztenünk. Az alábbi kódrészletet elegendő beszúrni, és az egyes változóknak a helyi LDAP-nak megfelelő adatokat értékül adni.

```
'example-ldap' => [
        'ldap:Ldap',

        /**
         * The connection string for the LDAP-server.
         * You can add multiple by separating them with a space.
         */
        'connection_string' => 'ldap.example.org',

        /**
         * Whether SSL/TLS should be used when contacting the LDAP server.
         * Possible values are 'ssl', 'tls' or 'none'
         */
        'encryption' => 'ssl',

        /**
         * The LDAP version to use when interfacing the LDAP-server.
         * Defaults to 3
         */
        'version' => 3,

        /**
         * Set to TRUE to enable LDAP debug level. Passed to the LDAP connector class.
         *
         * Default: FALSE
         * Required: No
         */
        'debug' => false,
```

```
/**
 * The LDAP-options to pass when setting up a connection
 * See [Symfony documentation]
 */
'options' => [
    /**
     * Set whether to follow referrals.
     * AD Controllers may require 0x00 to function.
     * Possible values are 0x00 (NEVER), 0x01 (SEARCHING),
     *   0x02 (FINDING) or 0x03 (ALWAYS).
     */
    'referrals' => 0x00,

    'network_timeout' => 3,
],

/**
 * The connector to use.
 * Defaults to '\SimpleSAML\Module\ldap\Connector\Ldap', but can be set
 * to '\SimpleSAML\Module\ldap\Connector\ActiveDirectory' when
 * authenticating against Microsoft Active Directory. This will
 * provide you with more specific error messages.
 */
'connector' => '\SimpleSAML\Module\ldap\Connector\Ldap',

/**
 * Which attributes should be retrieved from the LDAP server.
 * This can be an array of attribute names, or NULL, in which case
 * all attributes are fetched.
 */
'attributes' => null,

/**
 * Which attributes should be base64 encoded after retrieval from
 * the LDAP server.
 */
'attributes.binary' => [
    'jpegPhoto',
    'objectGUID',
    'objectSid',
```

```
        'mS-DS-ConsistencyGuid'
    ],

    /**
     * The pattern which should be used to create the user's DN given
     * the username. %username% in this pattern will be replaced with
     * the user's username.
     *
     * This option is not used if the search.enable option is set to TRUE.
     */
    'dnpattern' => 'uid=%username%,ou=people,dc=example,dc=org',

    /**
     * As an alternative to specifying a pattern for the users DN, it is
     * possible to search for the username in a set of attributes. This is
     * enabled by this option.
     */
    'search.enable' => false,

    /**
     * An array on DNs which will be used as a base for the search. In
     * case of multiple strings, they will be searched in the order given.
     */
    'search.base' => [
        'ou=people,dc=example,dc=org',
    ],

    /**
     * The scope of the search. Valid values are 'sub' and 'one' and
     * 'base', first one being the default if no value is set.
     */
    'search.scope' => 'sub',

    /**
     * The attribute(s) the username should match against.
     *
     * This is an array with one or more attribute names. Any of the
     * attributes in the array may match the value the username.
     */
    'search.attributes' => ['uid', 'mail'],
```

```
        /**
         * Additional filters that must match for the entire LDAP search to
         * be true.
         *
         * This should be a single string conforming to [RFC 1960]
         * and [RFC 2544]. The string is appended to the search attributes
         */
        'search.filter' => '(&(objectClass=Person)(|(sn=Doe)(cn=John *)))',

        /**
         * The username & password where SimpleSAMLphp should bind to before
         * searching. If this is left NULL, no bind will be performed before
         * searching.
         */
        'search.username' => null,
        'search.password' => null,
    ],
```

Megfelelő beállítások után a dinamikusan generált metadata a `/saml2/idp/metadata.php` útvonalon érhető el.

Tesztelés

A usereknek már nincs adminisztrációs felület azonban adminként még be lehet lépni amennyiben a core modulok között engedélyeztük az admin felületet:

https://service.example.org/simplesaml/admin/

A belépést követően a teszt fülre kattintva tesztelhetjük az authentikációs forrásokat:

https://idp.niif.hu/simplesaml/module.php/admin/test

Kézi metadata csere, élesített SP-vel

Az IdP metadata valamint metadata eszközök megtalálhatók a következő oldalon (admin fiók szükséges): https://idp.example/simplesaml/module.php/admin/federation

Amennyiben az SP is simplesamlphp használhatjuk a fentihez hasonló elérési úton található SimplesSAMLphp SP Metadatát, ez php formátumban van, ellenkező esetben pl shibboleth sp meg kell keresnünk a metaadat XML-t majd a fentebb említett federation oldalon található XML →simplesamlphp metadata konvertert használni.

Az így kapott php formátumú metaadatokat pedig be kell illeszteni az IdP-n a `metadata/saml20-sp-remote.php` fileba a következő példához hasonlóképp:

```
<?php

$metadata['https://sp.example.org/simplesaml/module.php/saml/sp/metadata.php/default-sp'] = [
    'AssertionConsumerService' => 'https://sp.example.org/simplesaml/module.php/saml/sp/saml2-acs.php/default-sp',
    'SingleLogoutService' => 'https://sp.example.org/simplesaml/module.php/saml/sp/saml2-logout.php/default-sp',
];
```

A másik (SP) oldalon amennyiben szintén simplesamlphp van, az idp metaadatokat hasonlóképp hasonlóképp érjük el majd illesszük be a metadata/saml20-idp-remote.php fileba.

```
<?php
$metadata['https://example.org/saml-idp'] = [
    'SingleSignOnService'  => 'https://example.org/simplesaml/saml2/idp/SSOService.php',
    'SingleLogoutService'  => 'https://example.org/simplesaml/saml2/idp/SingleLogoutService.php',
    'certificate'          => 'example.pem',
];
```

A certet a config php-ban beállított certdir-ben keresi (alapértelmezetten /cert) Amennyiben más SP-t használunk az idp XML metadatájára lesz szükség amely szintén föderáció fül alatt érhető el (https://idp.example/simplesaml/module.php/admin/federation), és az SPn-nek releváns dokumentációt kell követni.

Service Provider (SP) beállítás

Alapbeállítások

A telepített alkalmazásunk által kezelt SP-ket a **config/authsources.php** fájlban tudjuk beállítani. A SimpleSAMLphp a tanúsítvány fájlokat a korábban létrehozott **cert** mappában fogja keresni, a fájlokat elég relatív elérési úttal megadni.

```
<?php
$config = [
```

```
    /* This is the name of this authentication source, and will be used to access it later. */
    'default-sp' => [
        'saml:SP',
        'entityID' => 'https://myapp.example.org/',
        'privatekey' => 'saml.pem',
        'certificate' => 'saml.crt',
        'idp' => 'https://example.org/saml-idp', //Alapértelmezett IdP beállítása
    ],

];
```

Tesztelés

A fent elvégzett alapbeállítások után már tudjuk tesztelni a, hogy a felépített IdP - SP kapcsolat működik-e.

SP oldalon nyissuk meg a **admin** teszt felületet:

https://idp.niif.hu/simplesaml/module.php/admin/test

Itt kattintsunk a default SP-re

HREF-integráció

Metadata beállítása (IdP és SP is)

Javasolt dinamikus metaadatforrást (MDX) használni, opcionálisan kiegészítve statikus állományokkal. Részletes leírás itt: SimpleSAMLMixedMetadata()

IdP

Amennyiben van SSP alapú IdP-nk, melyet szeretnénk a föderáció részévé tenni, úgy a teendők a következők.

- (Az adminisztratív teendőktől itt most eltekintünk, a csatlakozás folyamata itt van leírva)
- Kell küldeni egy levelet a info@eduid.hucímre, benne néhány mondat mellett az IdP metaadatának URL-jével (https://example/org/simplesaml/module.php/saml/idp/metadata)

- Ha minden rendben megy, akkor az IdP bekerül a Resource_Registry-be, ezáltal a föderációs metaadatba is.
- Az előző pontban leírt módon be kell állítani a központi metadata feldolgozását.
- Amennyiben a föderációs metaadatban már szerepel a mi IdP-nk is, úgy a föderáció valamelyik, tesztelési célokat szolgáló SP-jénél ki is próbálhatjuk a bejelentkezést.
- **Fontos**, hogy a föderációs Discovery Service óránként generálja újra az IdP-k listáját, így ennyi idő mindenképp szükséges, hogy az új IdP megjelenjen itt, az egyes SP-k pedig két óránként töltik újra a metaadatot, így előfordulhat, hogy azonnal nem fog minden működni, de néhány óra alatt várhatóan beindul. :)
- Tesztelésre használható oldal: https://attributes.eduid.hu
- Ahhoz, hogy a Resource Registry-be is be tudjunk lépni és az IdP további, a föderációra vonatkozó beállításait meg tudjuk ejteni, ehhez az IdP-nek ki kell adnia az alábbi attribútumokat:
  - mail - ez belépés után, manuálisan is beállítható
  - eduPersonPrincipalName
  - schacHomeOrganizationType (az attribútumot hamarosan kivezetjük a kötelező attribútumok közül)
  - eduPersonScopedAffiliation

Attribútumok kezelése

Beállított IdP-nk alapértelmezés szerint azokat az attribútumokat adja ki, melyeket a metaadat alapján az SP kért (Lásd a metadatában a RequestedAttribute elemeket), és egyúttal alapból meg tudta szerezni a felhasználói adatbázisból, esetünkben az LDAP-ból. Mivel néhány attribútum nem szerepel az LDAP-ban, hanem az IdP-ben kell előállítani, így pár helyen módosítanunk kell az alapértelmezett konfiguráción.

A `metadata/saml20-idp-hosted.php` fájlba szerkesszük be az alábbi kódrészlet értelemszerűen módosított változatát. Az `'auth' => 'example-ldap',` sor alatt kezdjük. Fontos, hogy egyúttal a config.php `authproc.idp` részét kikommentezzük, nehogy az ottani sorszámokkal megadott default feladatok bekavarjanak.

```
 'AttributeNameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:uri',
 'userid.attribute' => 'uid', // Itt adjuk meg, hogy mely, az LDAPból származó attribútum
alapján fogja az IdP kiszámítani az eduPersonTargetedID-t
 'authproc' => array(
                10 => array(
                        'class' => 'core:AttributeMap',
                        'uid' => 'eduPersonPrincipalName'
                //Itt az 'uid' az az attribútum az LDAP-ban, amely a felhasználó azonosítóját
tartalmazza, mert ebből képezzük az eduPersonPrincipalName-t.
                ),
```

```
                # 20 => array(
                #         'class' => 'core:AttributeAdd',
                #         'schacHomeOrganizationType' =>
array('urn:schac:homeOrganizationType:hu:university')
                # //Kötelező statikus attribútum az
[[HREFAttributeSpec#schacHomeOrganizationType|intézmény jellegének]] megfelelően
                # ),
                30 => array(
                        'class' => 'core:AttributeAlter',
                        'subject' => 'eduPersonPrincipalName',
                        'pattern' => '/^.*$/',
                        'replacement' => '${0}@intezmenydomain.hu',
                // Itt adjuk hozzá az intézményi scope-ot az eduPersonPrincipalName már
meglévő értékéhez
                ),
                40 => array(
                        'class' => 'core:AttributeAlter',
                        'subject' => 'eduPersonAffiliation',
                        'pattern' => '/^.*$/',
                        'replacement' => '${0}@intezmenydomain.hu',
                // Itt adjuk hozzá az intézményi scope-ot az eduPersonAffiliation már meglévő
értékéhez
                ),
                50 => array(
                        'class' => 'core:AttributeMap',
                        'eduPersonAffiliation' => 'eduPersonScopedAffiliation'
                // Az LDAP-ból eduPersonAffiliation-ként érkező attribútumból föderációs
elvárásoknak megfelelően eduPersonScopedAffiliationt készítünk
                ),
                60 => array(
                        'class' => 'core:AttributeAdd',
                        'eduPersonScopedAffiliation' => array('member@intezmenydomain.hu')
                // Az eduPersonScopedAffiliation-ben tesztelés céljából kiadhatjuk member
értéket,
                // így ha LDAP-ból nem jön érték, akkor is láthatjuk, hogy működik az
attribútum kiadás
                ),
                61 => array(
                        'class' => 'core:TargetedID',
                        'nameId' => TRUE,
```

```
                ),
               // Itt állítjuk be, hogy az IdP előállítson és kiadhasson állandóazonosítóként
eduPersonTargetedID-t, ha kérik
                70 => array('class' => 'core:AttributeMap',
                        'name2oid'
                // Az LDAP-os attribútum nevekből itt kreálunk szabványos urn:oid
formátumúakat
                ),
                80 => 'core:AttributeLimit',
              ), // .authproc
       'simplesaml.nameidattribute' => 'eduPersonPrincipalName',
       'attributeencodings' => array(
               'urn:oid:1.3.6.1.4.1.5923.1.1.1.10' => 'raw',
        ),
        'sign.logout' => true
```

- További tudnivalók a Resource Registry-ről, ill. a Föderációs attribútum specifikációról.
- Ha minden rendben ment, akkor a Resource Registry-ben regisztrált IdP-hez tartozó adminisztrációs jogok átkerülnek az IdP technikai gazdájához, s ezzel a folyamat kész is.

SP

Amennyiben IdP-t is beállítottunk, és be is tudunk lépni a Resource Registry-be, úgy nincs más dolgunk, mint az RR-ben új SP-t hozzáadni a föderációhoz, amely a megfelelő átfutási idő után a föderáció minden tagjánál látható is lesz.

Ellenkező esetben (nincs IdP, és nem is tervezünk beállítani), akkor az IdP hozzáadásánál részletezett pontokon kell végig menni a metaadat betöltéséig, s a továbbiakat az említett e-mail címen megbeszélni.

Attribútum scopeok használata

A HREF föderáció IdP-i ún. scopeolt attribútumokat is használnak. Ez a scopeolás azt jelenti, hogy minden egyes IdP csak a saját scopejában ad ki attribútumokat, és a Shibboleth SP-k ezt ellenőrzik is. A scope és az attribútum valódi értéke egy '@' karakterrel kerül elválasztásra (ilyen attribútumok jelenleg: eduPersonScopedAffiliation illetve eduPersonPrincipalName).

A SimpleSAMLphp alapértelmezett telepítése nem szűri a hibásan scopeolt értékeket. Kiegészítő modulként szűrésre használható az NIIF által fejlesztett attributescope modul, ami reményeink szerint rövid távon a hivatalos SimpleSAMLphp kiadás része lehet.

A telepítésről és konfigurációról bővebben itt lehet olvasni: https://github.com/NIIF/simplesamlphp-module-attributescope

- Az `attributescope` modul használata esetén a következőképp kell módosítani a `config/config.php` fájlt:

```
authproc.sp =  array(
                    ...
                   // 49 => array('class' => 'core:AttributeMap', 'oid2name'),
                   50 => array(             'class' => 'attributescope:FilterAttributes'
                   ),
                   ...
               ),
```

Figyeljünk arra, hogy mire a modulhoz ér a vezérlés, az attribútumok nevei *friendlyName* alakúak legyenek (ne pedig *oid*-ok). A példában erre utal a 49-es sor.