

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.

Előkészületek

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 webservert
- 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 önálárírt, 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 mentjük 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 felejtünk 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 tanúsítványt használni!

- A SimpleSAMLphp alapértelmezetten a tanúsítványt a **cert** mappában keresi.
- Az alábbi paranccsal egy 10 éves [self-signed tanúsítvány](#) generálunk a SimpleSAMLphp 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 autentikációs 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 SimpleSAMLphp 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.org/simplesaml/module.php/admin/federation>), és az SPn-nek releváns dokumentációt kell követni.

Service Provider (SP) beállítás

Alapbeállítások

A telepített alkalmazásunk által kezelt SP-eket 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 metaadatban 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 kéri
    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.

Változat #6

cziernorbert hozta létre 2026-04-14 13:22:23 CEST

cziernorbert frissítette 2026-04-17 10:48:40 CEST