OpenSSO

OpenSSO

OpenSSO telepítése

Az OpenSSO szerver letölthető a projekt oldaláról. Telepítéshez servlet-api támogató alkalmazásszerver szükséges. (Tomcat-et nem ajánlják a futtatásra, inkább érdemes nagyobb alkalmazásszervereken futtatni, pl. Glassfish vagy Oracle AS)

Az alkalmazásszerver konfigurációjában a Heap size-ot 1G méretűre kell állítani (-Xmx1G), különben a telepítés hibát dob. Valamint érdemes a JVM hotspotot -server módban futtatni. (/path/to/glassfish/domains/domain/config/domain.xml)

Glassfish V2UR1 alatt így zajlik a telepítés:

```
$ /path/to/asadmin deploy --name opensso --contextroot opensso /path/to/opensso.war
```

Ezután webes felületen történik a konfigurálás: meg kell adni az admin felhasználó nevét, jelszavát, a konfigurációs címtár paramétereit (érdemes a beágyazott címtárat használni), a felhasználókat tároló címtár paramétereit (host, port, admin bind paraméterek és DIT gyökér).

Sikeres telepítés esetén a webes felületen állíthatjuk be ízlésünk szerint a kért szolgáltatásokat.

Realm-ek

Az OpenSSO egyszerre képes több szervezetet kiszolgálni, ezeknek a konfigurációját külön-külön 'realm'-ekben kezeli. Minden realmhez megadhatóak az authentikációs modulok, a felhasználói adatokat tartalmazó címtár elérésének paraméterei, egyedileg szerkeszthetők a hozzáférési szabályok, és a federációs konfiguráció is teljesen külön van minden szervezetnél.

Hosztolt IDP beállítása

Legegyszerűbben a nyitóoldalon elhelyezett gyorslinkekkel hozhatunk létre új IDP-t ('Create hosted Identity Provider'). Ekkor meg kell adni a realm-et, és hogy melyik Circle-of-trust részévé kívánjuk tenni az IDP-t. Ha még nem hoztunk létre COT-ot, akkor azt megtehetjük itt.

Ezután a 'Federation' fül alatt találjuk az összes beállítási paramétert. A hosztolt IDP minden beállítását elvégezhetjük adminfelületről: aláíró és titkosító kulcsok (ezeket keystore-ból veszi, amit a keytool parancssal menedzselhetünk parancssorból), támogatott NameID formátumok,

attríbutum mappelés akár saját osztállyal, és persze a támogatott SAML2 bindingok - Redirect, POST, Artifact - paraméterei.

A beállított metaadatok XML formátumban a http://host:port/opensso/saml2/jsp/exportmetadata.jsp URL-en lesznek elérhetőek. (az exportmetadata.jsp a következő paramétereket tudja fogadni: realm, entityID)

Amennyiben digitálisan aláírt metaadatra van szükségünk, azt az adminkonzolból tudjuk csak exportálni, illetve ezen patch segítségével az exportmetadata.jsp -ből is a signMetadata=true paraméter megadásával (https://opensso.dev.java.net/issues/show_bug.cgi?id=2680)

Új SP hozzáadása

Szintén a taszkok között található link új távoli SP hozzáadására ('Register remote Service Provider'). Itt a SAML2 metaadat URL-jét kell megadni, vagy feltölteni azt. Ezen kívül egy listában megadható, hogy milyen attribútum-leképzést igényel az SP. Természetesen az SP-t is hozzá kell adni egy COT-hoz.

Miután felvettük az SP-t, néhány attribútumot a Federation fülön átállíthatunk utólag is.

Interoperabilitás

- OpenSSO IdP - Shibboleth2 SP
- OpenSSO SP - Shibboleth2 IdP
- OpenSSO IdP - SimpleSAMLphp SAML2 SP

Problémák

Hibás metaadat hozzáadása (vannak hibák, amiket az import parancs nem vesz észre) után a hozzáadott entity-t nem lehet törölni sem, ilyenkor a teljes federation fül működésképtelenné válik. Megoldás: újraindítás és a famadm delete-entity parancs használata. Ha ez sem megy, akkor a konfigurációs címtárból ki kell törölni az entity-t.

Az SP metaadatának tartalmazni kell a NameID attribútumot, az IDP anélkül nem képes a federációra. Ezt a ManageNameIDService után, és az AssertionConsumerService elé kell beírni, pl. így

```
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
```

OpenSSO IdP - Shibboleth2 SP

Cél

http://maszat.sch.bme.hu -n futó OpenSSO-n IdP létrehozása, és a https://sandbox.aai.niif.hu/shibboleth alatt futó Shibboleth 2 SP-vel való federáció.

OpenSSO IdP létrehozása

lásd: OpenSSO

Cél Realm: /niif-teszt IDP entityID: https://idp.sch.bme.hu/niif-teszt Legalább a "signing certificate alias" -t állítsuk be

A /opensso/famadm.jsp -> export-entity parancssal tudjuk XML-ként exportálni az létrehozott IDP metaadatát. (Vagy, a /opensso/saml2/jsp/exportmetadata.jsp?entityID=https://idp.sch.bme.hu/niif-teszt&realm=/niif-teszt URL-ről közvetlenül elérhetjük).

Shibboleth 2 SP beállítása

/etc/shibboleth/shibboleth2.xml:

```
...
<SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="teszt"
               relayState="cookie" entityID="https://idp.sch.bme.hu/niif-teszt">
  <SessionInitiator type="SAML2" defaultACSIndex="1" template="bindingTemplate.html"/>
</SessionInitiator>
...
<MetadataProvider type="XML" file="maszat-idp.xml"/>
...
```

Shibboleth SP Metadata importálása OpenSSO-ba

A lementett XML-ből töröljük ki az `<md:Extension>` node-ot, mivel a parszolás során hibaüzenetet dob, ami a teljes SAML2Meta konfigurációt megfekteti.

```
ERROR: SAML2MetaManager.getEntityDescriptor
javax.xml.bind.UnmarshalException: Unexpected end of element
{urn:oasis:names:tc:SAML:2.0:metadata}:Extensions
...
```

Ha ez a hiba előjön, akkor a konfigurációs címtárból kell törölni az entity-t, ugyanis ilyenkor teljesen használhatatlan lesz a felület (ez egy bug sajnos)

```
$ ldapmodify ...
dn:
ou=<entityid>,ou=default,ou=OrganizationConfig,ou=1.0,ou=sunFMSAML2MetadataService,ou=services
,
 o=<realm>,ou=services,<basedn>
changetype: delete
```

A problémát az okozza, hogy amikor a metadata xml-t beolvassa a SAX parszer, akkor JAXB-vel mappeli java objektumokra, és ezt marshallolja ki a konfigurációs store-ba. Viszont az Extensions belseje egy teljesen külön névtér és külön séma, ezért arra nincsenek generált osztályok. Ezt úgy veszi a JAXB, hogy egy `<Extensions/>` -t ír ki, ami viszont unmarshal időben nem felel meg már a sémának (ugyanis xsd szerint ott legalább egy elemnek lennie kell).

Egyelőre azt a megoldást javasolták, hogy kézzel szedjem ki az `<Extensions>` -t, aminek továbbra sem örülök az xml aláírás és a kézi hegesztés miatt. Idővel egy olyan javítást fognak készíteni, amivel user osztálykönyvtárat meg lehet adni az opensso-nak, hogy a metaadat bizonyos részeit (pl. extensions belseje) arra mappelje le. Illetve tettem egy olyan javaslatot, hogy ilyen esetben amikor kinullozza a node belsejét a JAXB, akkor a teljes `<Extensions>` node-ot töröljék, így legalább az xml valid marad. Talán azt is javítják, hogy ilyenkor a teljes metaadat rész összeomlik és semmilyen műveletet nem lehet végezni. (

https://opensso.dev.java.net/issues/show_bug.cgi?id=2736)

A ManageNameIDService ill. AssertionConsumerService közé írjuk be a következő node-okat:

```
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
```

Ha nem adunk meg NameIDFormat -ot, akkor az OpenSSO IdP vissza fogja utasítani a kérést. Megj. https://opensso.dev.java.net/issues/show_bug.cgi?id=2172 hosszútávon ezt a viselkedést valószínű megváltoztatják.

Ezt a metaadatot a Federation fül Entities -> Import Entity parancssal tudjuk importálni. Itt ki kell választani a /niif-teszt realm-et, és feltölteni az XML-t (vagy megadni az URL-jét). Ezután felül a Circle of Trust-nál hozzá tudjuk már adni a SAML2 SP-t.

Ha mindezt végigcsináltuk, máris működik minden.

Problémák

Ha Shibboleth2 SP-ben külön application-be tesszük a metaadatot, akkor nem találja meg a SAML Response issuer-alapján.

Az OpenSSO nem jelzi a saml attribútum névformátumát (`<saml:Attribute NameFormat="...">`). A Shibboleth pedig csak az urn:oasis:names:tc:SAML:2.0:attrname-format:uri típusú nevet fogadja el.

Emiatt a következő probléma adódik:

```
shibd.log:

2008-05-14 18:33:25 INFO Shibboleth.AttributeExtractor : creating mapping for Attribute
urn:mace:dir:attribute-def:mail
...
<saml:Attribute Name="urn:mace:dir:attribute-def:mail">
   <saml:AttributeValue
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="xs:string">hege@*********</saml:AttributeValue>
</saml:Attribute>
...
2008-05-14 18:33:31 INFO Shibboleth.AttributeExtractor [1]: skipping unmapped SAML 2.0
Attribute
with Name:  urn:mace:dir:attribute-def:mail, Format:urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecifiedd
```

Itt egy patch OpenSSO-hoz ami megoldja a problémát:

https://opensso.dev.java.net/issues/show_bug.cgi?id=2775 Ezen patch használatával a következő módon kell megadni az IDP attribútum mappelést az IDP > Attribute Processing fülön:

```
# [NameFormat|]SAMLAttributeName=LocalAttributeName
urn:oasis:names:tc:SAML:2.0:attrname-format:uri|urn:mace:dir:attribute-def:mail=mail
```

OpenSSO IdP - SimpleSAMLphp SAML2 SP

Cél

OpenSSO hosztolt IdP és SimpleSAMLphp SP összekapcsolása a SAML2 protokoll segítségével.

SimpleSAMLphp telepítése

[[http://rnd.feide.no/content/installing-simplesamlphp]]

Konfigurációs paraméterek (config/config.php)

A következő paramétereket érdemes beállítani kezdésképp:

```
secretsalt: egy titkos 32 bájtos véletlenszám, amit a titkosításhoz használni fog a
simplesamlphp
technicalcontact_name,email: az üzemeltető technikai kapcsolattartója
logging_handler: file / syslog
debug: bekapcsolva minden saml kérés és válasz megjelenik a webes felületen (kényelmes!)
enable.saml20-sp, enable.saml20-idp, enable.shib13-sp, enable.shib13-idp
default-saml20-idp: Discovery Service megkerülése és fix IdP választása
```

IdP metaadat beállítása

metadata/saml20-idp-remote.php:

```
'https://idp.sch.bme.hu/niif-teszt' =>  array(
  'name'                                  =>      'NIIF Test at idp.sch.bme.hu',
  'description'                   => 'Log in via idp.sch.bme.hu',
```

```
  'SingleSignOnService'   =>
'http://maszat.sch.bme.hu:58080/opensso/SSORedirect/metaAlias/niif-teszt/idp',
  'SingleLogoutService'   =>
'http://maszat.sch.bme.hu:58080/opensso/IDPSloRedirect/metaAlias/niif-teszt/idp',
  'base64attributes'              =>      false,
  'request.signing' => false,
  'certificate' => "maszat-idp.crt",
  'certFingerprint' => "DE:F1:8D:BE:D5:47:CD:F3:D5:2B:62:7F:41:63:7C:44:30:45:FE:33",
  'saml2.relaxvalidation' => array('noattributestatement')
)
```

A cert könyvtárba mentsük le a maszat-idp.crt-t (például a maszat idp metaadatból kimásolva).

A fenti konfiguráció HTTP/Redirect bindingot használ a SAML Requestre, a választ pedig HTTP/Post-on keresztül kapja. Fontos, hogy a base64attributes ki legyen kapcsolva, ugyanis az OpenSSO IdP nem kódolja base64-be az attribútumokat a SAML Response-ban.

SP metaadat beállítása

metadata/saml20-sp-hosted.php:

```
'https://maszat.sch.bme.hu/simplesamlphp/sp/niif-teszt' => array(
  'host'  => 'maszat.sch.bme.hu',
/*'privatekey' => 'server.pem',
  'certificate' => 'server.crt',
  'request.signing' => true,*/
  'NameIDFormat' => 'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent'
)
```

Ezután a /simplesamlphp/saml2/sp/metadata.php?output=xml URL-en keresztül tudjuk elérni az SP metaadatot. Fontos, hogy ebben a generált metaadatban nem tükröződik pl. a signing certificate és a NameIDFormat beállítás, ezért ezeket kézzel kell beleszerkeszteni.

Miután kijavítgattuk a metaadat fájlt, az OpenSSO adminfelület Federation -> Import Entity parancsával tudjuk importálni a megfelelő Realm-be. Importálás után a Circle of Trust konfigurációhoz is hozzá kell adni a SimpleSAMLphp SP-t.

Problémák

Nincsenek :)