Shib2SP

Az SP-t a `shibboleth2.xml` állományon keresztül konfigurálhatjuk. Ebben a leírásban feltételezzük, hogy az SP konfigurációja a `/etc/shibboleth` könyvtárban van.

Előkészületek

- Telepítsük a shibbolethet
- Válasszunk egy egyedi azonosítót, ún. `entityID`-t az SP számára. Ez az azonosító URL formájú, létező hosztnév, egy - alapértelmezés szerint - /shibboleth path-szal. Pl: https://lipton.aai.niif.hu/shibboleth. Megfelelő konfiguráció után az entityID-t meghívva válaszul az adott entitás metaadatát kapjuk válaszul.

Működő példa konfiguráció 1

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">

    <!--
    By default, in-memory StorageService, ReplayCache, ArtifactMap, and SessionCache
    are used. See example-shibboleth2.xml for samples of explicitly configuring them.
    -->

    <!--
    To customize behavior for specific resources on Apache, and to link vhosts or
    resources to ApplicationOverride settings below, use web server options/commands.
    See https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPConfigurationElements for
help.
```

```
    For examples with the RequestMap XML syntax instead, see the example-shibboleth2.xml
    file, and the https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPRequestMapHowTo
topic.
    -->

    <!-- The ApplicationDefaults element is where most of Shibboleth's SAML bits are defined.
-->
    <ApplicationDefaults entityID="https://events.prace-ri.eu/shibboleth"
                         REMOTE_USER="eppn"

cipherSuites="ECDHE+AESGCM:ECDHE:!aNULL:!eNULL:!LOW:!EXPORT:!RC4:!SHA:!SSLv2">

        <!--
        Controls session lifetimes, address checks, cookie handling, and the protocol
handlers.
        You MUST supply an effectively unique handlerURL value for each of your applications.
        The value defaults to /Shibboleth.sso, and should be a relative path, with the SP
computing
        a relative value based on the virtual host. Using handlerSSL="true", the default, will
force
        the protocol to be https. You should also set cookieProps to "https" for SSL-only
sites.
        Note that while we default checkAddress to "false", this has a negative impact on the
        security of your site. Stealing sessions via cookie theft is much easier with this
disabled.
        -->
        <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                  checkAddress="false" handlerSSL="false" cookieProps="http">

            <!--
            Configures SSO for a default IdP. To allow for >1 IdP, remove
            entityID property and adjust discoveryURL to point to discovery service.
            (Set discoveryProtocol to "WAYF" for legacy Shibboleth WAYF support.)
            You can also override entityID on /Login query string, or in RequestMap/htaccess.
            -->

                <SSO discoveryProtocol="SAMLDS"
discoveryURL="https://mdx.eduid.hu/role/idp.ds">
                 SAML2 SAML1
```

```
                </SSO>


            <!-- SAML and local-only logout. -->
            <Logout>SAML2 Local</Logout>

            <!-- Extension service that generates "approximate" metadata based on SP
configuration. -->
            <Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>

            <!-- Status reporting service. -->
            <Handler type="Status" Location="/Status" acl="127.0.0.1 ::1"/>

            <!-- Session diagnostic service. -->
            <Handler type="Session" Location="/Session" showAttributeValues="false"/>

            <!-- JSON feed of discovery information. -->
            <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>

        </Sessions>

        <!--
        Allows overriding of error template information/filenames. You can
        also add attributes with values that can be plugged into the templates.
        -->
        <Errors supportContact="prace-indico-admin@niif.hu"
            helpLocation="/about.html"
            styleSheet="/shibboleth-sp/main.css"/>

        <MetadataProvider type="Dynamic" ignoreTransport="true">
              <Subst>https://mdx.eduid.hu/entities/$entityID</Subst>
              <MetadataFilter type="Signature" certificate="href-metadata-signer-2020.crt"/>
        </MetadataProvider>

        <!-- Example of remotely supplied batch of signed metadata. -->
        <!--
        <MetadataProvider type="XML" validate="true"
              uri="http://example.org/federation-metadata.xml"
              backingFilePath="federation-metadata.xml" reloadInterval="7200">
```

```
            <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
            <MetadataFilter type="Signature" certificate="fedsigner.pem"/>
            <DiscoveryFilter type="Blacklist" matcher="EntityAttributes" trimTags="true"
              attributeName="http://macedir.org/entity-category"
              attributeNameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
              attributeValue="http://refeds.org/category/hide-from-discovery" />
        </MetadataProvider>
        -->

        <!-- Example of locally maintained metadata. -->
        <!--
        <MetadataProvider type="XML" validate="true" file="partner-metadata.xml"/>
        -->

        <!-- Map to extract attributes from SAML assertions. -->
        <AttributeExtractor type="XML" validate="true" reloadChanges="false" path="attribute-
map.xml"/>

        <!-- Use a SAML query if no attributes are supplied during SSO. -->
        <AttributeResolver type="Query" subjectMatch="true"/>

        <!-- Default filtering policy for recognized attributes, lets other data pass. -->
        <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>

        <!-- Simple file-based resolver for using a single keypair. -->
        <CredentialResolver type="File" key="events-shib.key" certificate="events-shib.cert"/>

        <!--
        The default settings can be overridden by creating ApplicationOverride elements (see
        the https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApplicationOverride
topic).
        Resource requests are mapped by web server commands, or the RequestMapper, to an
        applicationId setting.

        Example of a second application (for a second vhost) that has a different entityID.
        Resources on the vhost would map to an applicationId of "admin":
        -->
        <!--
        <ApplicationOverride id="admin" entityID="https://admin.example.org/shibboleth"/>
```

```
        -->
    </ApplicationDefaults>

    <!-- Policies that determine how to process and authenticate runtime messages. -->
    <SecurityPolicyProvider type="XML" validate="true" path="security-policy.xml"/>

    <!-- Low-level configuration about protocols and bindings available for use. -->
    <ProtocolProvider type="XML" validate="true" reloadChanges="false" path="protocols.xml"/>

</SPConfig>
```

Működő példa konfiguráció 2

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    logger="syslog.logger" clockSkew="180">

    <!-- The OutOfProcess section contains properties affecting the shibd daemon. -->
    <OutOfProcess logger="shibd.logger">
        <!--
        <Extensions>
            <Library path="odbc-store.so" fatal="true"/>
        </Extensions>
        -->
    </OutOfProcess>

    <!-- The InProcess section conrains settings affecting web server modules/filters. -->
    <InProcess logger="native.logger">
    </InProcess>

    <!-- Only one listener can be defined, to connect in process modules to shibd. -->
    <UnixListener address="shibd.sock"/>
    <!-- <TCPListener address="127.0.0.1" port="12345" acl="127.0.0.1"/> -->

    <!-- This set of components stores sessions and other persistent data in daemon memory. --
>
```

```
    <StorageService type="Memory" id="mem" cleanupInterval="900"/>
    <SessionCache type="StorageService" StorageService="mem" cacheTimeout="3600"
inprocTimeout="900" cleanupInterval="900"/>
    <ReplayCache StorageService="mem"/>
    <ArtifactMap artifactTTL="180"/>

    <!-- This set of components stores sessions and other persistent data in an ODBC database.
-->
    <!--
    <StorageService type="ODBC" id="db" cleanupInterval="900">
        <ConnectionString>

DRIVER=drivername;SERVER=dbserver;UID=shibboleth;PWD=password;DATABASE=shibboleth;APP=Shibbole
th
        </ConnectionString>
    </StorageService>
    <SessionCache type="StorageService" StorageService="db" cacheTimeout="3600"
inprocTimeout="900" cleanupInterval="900"/>
    <ReplayCache StorageService="db"/>
    <ArtifactMap StorageService="db" artifactTTL="180"/>
    -->

    <!-- To customize behavior, map hostnames and path components to applicationId and other
settings. -->
    <RequestMapper type="Native">
        <RequestMap applicationId="default">
            <!--
            The example requires a session for documents in /secure on the containing host
with http and
            https on the default ports. Note that the name and port in the <Host> elements
MUST match
            Apache's ServerName and Port directives or the IIS Site name in the <ISAPI>
element
            below.
            -->
□□<Host name="wiki.aai.niif.hu" authType="shibboleth"
□□□  requireSession="false" applicationId="wiki.aai"
□□□  redirectErrors="https://wiki.aai.niif.hu/index.php/Kezd%C5%91lap">
□□□<Path name="secure" requireSession="true" />
```

```
□□</Host>
□□<Host name="www.aai.niif.hu" authType="shibboleth"
□□□  requireSession="false" applicationId="www.aai">
□□□<Path name="secure" requireSession="true" />
□□</Host>
        </RequestMap>
    </RequestMapper>

    <!--
    The ApplicationDefaults element is where most of Shibboleth's SAML bits are defined.
    Resource requests are mapped by the RequestMapper to an applicationId that
    points into to this section.
    -->
    <ApplicationDefaults id="default" policyId="default"
        entityID="https://lipton.aai.niif.hu/shibboleth"
        homeURL="https://lipton.aai.niif.hu/shib-error.php"
        REMOTE_USER="eppn persistent-id targeted-id"
        signing="false" encryption="false"
        >

        <!--
        Controls session lifetimes, address checks, cookie handling, and the protocol
handlers.
        You MUST supply an effectively unique handlerURL value for each of your applications.
        The value can be a relative path, a URL with no hostname (https:///path) or a full
URL.
        The system can compute a relative value based on the virtual host. Using
handlerSSL="true"
        will force the protocol to be https. You should also add a cookieProps setting of ";
path=/; secure"
        in that case. Note that while we default checkAddress to "false", this has a negative
        impact on the security of the SP. Stealing cookies/sessions is much easier with this
disabled.
        -->
        <Sessions lifetime="28800" timeout="3600" checkAddress="false"
            handlerURL="/Shibboleth.sso" handlerSSL="true"
            exportLocation="http://localhost/Shibboleth.sso/GetAssertion"
            idpHistory="false" idpHistoryDays="7">
```

```
            <!--
            SessionInitiators handle session requests and relay them to a Discovery page,
            or to an IdP if possible. Automatic session setup will use the default or first
            element (or requireSessionWith can specify a specific id to use).
            -->

            <!-- Directly to the IdP -->
            <SessionInitiator type="Chaining" Location="/Login" id="Intranet"
                    relayState="cookie" entityID="https://idp.niif.hu/shibboleth">
                <SessionInitiator type="SAML2" defaultACSIndex="1"
template="bindingTemplate.html"/>
                <SessionInitiator type="Shib1" defaultACSIndex="5"/>
            </SessionInitiator>

□□□<!-- Discovery Service -->
            <SessionInitiator type="Chaining" Location="/DS" id="DS"
□                       relayState="cookie" acsByIndex="false"
□□□      isDefault="true" >
                <SessionInitiator type="SAML2" template="bindingTemplate.html"
□□□□  defaultACSIndex="3" />
                <SessionInitiator type="Shib1" defaultACSIndex="5" />
                <SessionInitiator type="SAMLDS" URL="https://ds.niif.hu/"/>
            </SessionInitiator>
□    <SessionInitiator type="Chaining" Location="/SAML1DS" acsByIndex="false"
relayState="cookie"
□□□      id="Saml1Only">
            □<SessionInitiator type="Shib1" defaultACSIndex="6"/>
            □<SessionInitiator type="SAMLDS" URL="https://ds.niif.hu" />
            </SessionInitiator>

            <!--
            md:AssertionConsumerService locations handle specific SSO protocol bindings,
            such as SAML 2.0 POST or SAML 1.1 Artifact. The isDefault and index attributes
            are used when sessions are initiated to determine how to tell the IdP where and
            how to return the response.
            -->
            <md:AssertionConsumerService Location="/SAML2/POST" index="1"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
            <md:AssertionConsumerService Location="/SAML2/POST-SimpleSign" index="2"
```

```
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-SimpleSign"/>
            <md:AssertionConsumerService Location="/SAML2/Artifact" index="3"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>
            <md:AssertionConsumerService Location="/SAML2/ECP" index="4"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
            <md:AssertionConsumerService Location="/SAML/POST" index="5"
                Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"/>
            <md:AssertionConsumerService Location="/SAML/Artifact" index="6"
                Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>

            <!-- LogoutInitiators enable SP-initiated local or global/single logout of
sessions. -->
            <LogoutInitiator type="Chaining" Location="/Logout" relayState="cookie">
                <LogoutInitiator type="SAML2" template="bindingTemplate.html"/>
                <LogoutInitiator type="Local"/>
            </LogoutInitiator>

            <!-- md:SingleLogoutService locations handle single logout (SLO) protocol
messages. -->
            <md:SingleLogoutService Location="/SLO/SOAP"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
            <md:SingleLogoutService Location="/SLO/Redirect"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
            <md:SingleLogoutService Location="/SLO/POST" conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
            <md:SingleLogoutService Location="/SLO/Artifact"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>

            <!-- md:ManageNameIDService locations handle NameID management (NIM) protocol
messages. -->
            <md:ManageNameIDService Location="/NIM/SOAP"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
            <md:ManageNameIDService Location="/NIM/Redirect"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"/>
            <md:ManageNameIDService Location="/NIM/POST" conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
```

```
            <md:ManageNameIDService Location="/NIM/Artifact"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"/>

            <!--
            md:ArtifactResolutionService locations resolve artifacts issued when using the
            SAML 2.0 HTTP-Artifact binding on outgoing messages, generally uses SOAP.
            -->
            <md:ArtifactResolutionService Location="/Artifact/SOAP" index="1"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>

            <!-- Extension service that generates "approximate" metadata based on SP
configuration. -->
            <Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>

            <!-- Status reporting service. -->
            <Handler type="Status" Location="/Status" acl="127.0.0.1"/>

            <!-- Session diagnostic service. -->
            <Handler type="Session" Location="/Session"/>

        </Sessions>

        <!--
        You should customize these pages! You can add attributes with values that can be
plugged
        into your templates. You can remove the access attribute to cause the module to return
a
        standard 403 Forbidden error code if authorization fails, and then customize that
condition
        using your web server.
        -->
        <Errors session="sessionError.html"
            metadata="metadataError.html"
            access="accessError.html"
            ssl="sslError.html"
            localLogout="localLogout.html"
            globalLogout="globalLogout.html"
            supportContact="root@localhost"
```

```
            logoLocation="/shibboleth-sp/logo.jpg"
            styleSheet="/shibboleth-sp/main.css"/>

        <!-- Uncomment and modify to tweak settings for specific IdPs or groups. -->
        <!-- <RelyingParty Name="SpecialFederation" keyName="SpecialKey"/> -->

        <!-- Chains together all your metadata sources. -->
        <MetadataProvider type="Chaining">
           <MetadataProvider type="XML" uri="https://metadata.eduid.hu/current/href.xml"
               backingFilePath="href.xml" reloadInterval="7200">
              <SignatureMetadataFilter certificate="href-metadata-signer-2010.crt"/>
          </MetadataProvider>
           <MetadataProvider type="XML" uri="https://metadata.eduid.hu/current/niifi.xml"
               backingFilePath="niifi.xml" reloadInterval="7200">
              <SignatureMetadataFilter certificate="href-metadata-signer-2010.crt"/>
          </MetadataProvider>
        </MetadataProvider>

        <!-- Chain the two built-in trust engines together. -->
        <TrustEngine type="Chaining">
            <TrustEngine type="ExplicitKey"/>
            <TrustEngine type="PKIX"/>
        </TrustEngine>

        <!-- Map to extract attributes from SAML assertions. -->
        <AttributeExtractor type="XML" path="attribute-map.xml"/>

        <!-- Use a SAML query if no attributes are supplied during SSO. -->
        <AttributeResolver type="Query"/>

        <!-- Default filtering policy for recognized attributes, lets other data pass. -->
        <AttributeFilter type="XML" path="attribute-policy.xml"/>

        <!-- Simple file-based resolver for using a single keypair. -->
        <!-- <CredentialResolver type="File" key="example.key" certificate="example.crt"/> -->

        <!-- Example of a second application (using a second vhost) that has a different
entityID. -->
        <!-- <ApplicationOverride id="admin" entityID="https://admin.example.org/shibboleth"/>
```

```
-->
<ApplicationOverride id="wiki.aai" entityID="https://wiki.aai.niif.hu/shibboleth" >
<CredentialResolver type="File" key="wiki.aai.niif.hu.key"
    certificate="wiki.aai.niif.hu.crt"/>
</ApplicationOverride>
<ApplicationOverride id="www.aai" entityID="https://www.aai.niif.hu/shibboleth" >
<CredentialResolver type="File" key="www.aai.niif.hu.key"
    certificate="www.aai.niif.hu.crt"/>
</ApplicationOverride>
    </ApplicationDefaults>

    <!-- Each policy defines a set of rules to use to secure messages. -->
    <SecurityPolicies>
        <!--
        The predefined policy enforces replay/freshness, standard
        condition processing, and permits signing and client TLS.
        -->
        <Policy id="default" validate="false">
            <PolicyRule type="MessageFlow" checkReplay="true" expires="60"/>
            <PolicyRule type="Conditions">
                <PolicyRule type="Audience"/>
                <!-- Enable Delegation rule to permit delegated access. -->
                <!-- <PolicyRule type="Delegation"/> -->
            </PolicyRule>
            <PolicyRule type="ClientCertAuth" errorFatal="true"/>
            <PolicyRule type="XMLSigning" errorFatal="true"/>
            <PolicyRule type="SimpleSigning" errorFatal="true"/>
        </Policy>
    </SecurityPolicies>


</SPConfig>
```

Minimális beállítások

Környezeti beállítások

A konfigurációs fájl első negyedében lévő szekciókban elsősorban a Shibboleth futásával kapcsolatos beállítások találhatók, amelyek alapértelmezett értékei legtöbbször megfelelőek az általunk elvárt működéshez.

- `OutOfProcess`
- `InProcess`
- `UnixListener`
- `StorageService`
- `SessionCache`
- `ReplayCache`
- `ArtifactMap`

RequestMap

A RequestMap megadja azokat a címeket (Host és Path), amelyeket a Shibboleth SP kezelni fog. Szerkezete:

```
        <RequestMap applicationId="default">
□□<Host name="wiki.aai.niif.hu" authType="shibboleth"
□□□  requireSession="false" applicationId="wiki.aai"
□□□  redirectErrors="https://wiki.aai.niif.hu/index.php/Kezd%C5%91lap">
□□□<Path name="secure" requireSession="true" />
□□</Host>
□□<Host name="www.aai.niif.hu" authType="shibboleth"
□□□  requireSession="false" applicationId="www.aai">
□□□<Path name="secure" requireSession="true" />
□□</Host>
        </RequestMap>
```

A RequestMap több Host elemet is tartalmazhat, a Host elem 0 vagy több Path elemet tartalmazhat.

> **Figyelem**
>
> Ha 1-nél nagyobb mélységű könyvtárat (pl. a `/shibtest/shibreq` nevűt) szeretnénk védeni, akkor **nem** adhatjuk meg a *name* paraméterben a "shibtest/shibreq" értéket, hanem egymásba ágyazott Path elemeket kell használni. A *name* paraméter nem tartalmazhat '/' karaktert.

Az egyes elemeknél paraméterekkel szabályozhatjuk, hogy az SP milyen módon kezelje a hostot vagy az útvonalat. A paraméterek felüldefiniálhatók. A legfontosabb paraméterek az alábbiak (ezek ugyanúgy használhatók Host-nál mint Path-nál):

- `requireSession`: ha értéke "true", akkor az SP csak akkor továbbítja a HTTP request-et az alkalmazás ill. a webszerver felé, ha sikerült létrehozni egy autentikált session-t. Ha "false", akkor az alkalmazás felelős azért, hogy létrehozza a Shibboleth session-t (ún. lazy session) Alapértelmezés: "false"
- `exportAssertion`: ha értéke "true", akkor az SP átadja a teljes, IdP-től kapott Attribute Assertion-t az alkalmazásnak a SHIB_ATTRIBUTES HTTP mezőben (base64 kódolással). Alapértelmezés: "false"
- `applicationId`: lehetőség van arra, hogy bizonyos helyekre érkező kérésekre az SP más és más módon próbáljon meg session-t létrehozni, ezt ún. Shibboleth Application-ben konfigurálhatjuk. Ha nem adunk meg értéket, akkor a "default" application-nél megadott értékek vonatkoznak majd a session-re.
- `redirectError`: átirányítási hiba esetén a Shibboleth erre az oldalra irányít át - ennek az [[isPassive]] -ot használó oldalaknál van jelentősége

ApplicationDefaults

Ennél a szekciónál tudjuk megadni az általános, minden alkalmazásra érvényes alapbeállításokat. Ezek a beállítások természetesen minden egyes alkalmazás tekintetében felüldefiniálhatók.

Alapattribútumok

- `id` **(kötelező)**: a alkalmazás elsődleges belső azonosítója. Az alapbeállíásoknál (tehát itt) elvárt érték: `default`
- `policyId` **(kötelező)**: a vonatkozó id-jű `SecurityPoilicies` szekcióra mutat
- `entityID` **(kötelező)**: egyedi azonosító, amely egyértelműen azonosít egy SP-t. A külső alkalmazások csak ezt az azonosítót látják, belső id-t...stb nem. Többnyire URL formátumú.
- `homeURL`:
- `REMOTE_USER`: egy prioritási listát adhatunk meg, melynek elemei azok az attribútumok, melyek közül az az első nem NULL értékű kerül beállításra a HTTP_REMOTE_USER változóba
- `signing`: az XML üzenetek aláírtságára vonatkozó elvárások állíthatók be
- `encryption`: az XML üzenetek titkosítására vonatkozó elvárások állíthatók be

Sessions

Ennél a szekciónál állíthatjuk be, hogy az SP miként kezelje a Single Sign-on (SSO) folyamatának egyes részeit. Az alapparamétereken túl (session lejárati idő...stb) ún. handlerek találhatók benne. Természetesen az alapbeállítások alkalmazásonként felülírhatók az `<ApplicationOverride>` résznél.

Handlerekről

A handlerek az SP-n belül működnek, de a fő folyamatoktól leválasztva. Egy-egy speciális feladatot látnak el - mintegy szkript jelleggel. Egy handler a megfelelő URL meghívásával érhető el. Ezen

URL meghívásakor az SP felismeri, hogy mely handlert illeti az adott részfeladat megoldása, és átadja neki a feladat ellátásához szükséges paramétereket. Az SP-n belül egy "alaphandler" található, amely felel a handlereket illető feladatok kiosztásáért, ez jelenti majd a handlerek elérési útvonalában a gyökeret.

Alapattribútumok

- **`handlerURL`**: Az alaphandler elérési útja. Alapértelmezés szerint: `"/Shibboleth.sso"`.
- **`handlerSSL`**: Beállítható, hogy kizárólag titkosított csatornán keresztül történhessen a handlerekkel való kommunikáció. Alapértelmezés szerint: `true`
- **`lifetime`**: Beállítható az SP session maximális hossza. Alapértelmezés szerint ez 28800 másodperc. Fontos megjegyezni, hogy az SP session megszűnése nincs közvetlen hatással a Shibboleth által védett alkalmazás által generált sessionre
- **`timeout`**:
- **`checkAddress`**: Megadható, hogy az SP ellenőrizze-e, hogy a felhasználó IP címe egyezik-e az IdP által az asseirton-ben írttal. Alapértelmezés szerint: `true`
- **`exportLocation`**:
- **`idpHistory`**: Igaz érték esetén a SP beállít egy cookie-et, melyhez értékül adja azt az IdP-t, amelynél sikeres autentikáció történt. Alapértelmezés szerint: `false`
- **`idpHistoryDays`**: Megadhatjuk az `idpHistory` cookie érvényességi idejét napokban. Amennyiben nem kerül beállításra, akkor a cookie az adott munkamenet végén lejár

SessionInitiator

Ennél a szekciónál kerülnek beállításra azok a paraméterek, melyek meghatározzák, hogy az SP kihez-mihez irányítsa a felhasználót, mikor az érvényes session nélkül (tehát autentikáció előtt) próbálja elérni a Shibboleth által védett tartalmat.

Alapattribútumok

- **`type`**: Meghatározza a SessionInitiator típusát. A főbb típusokat lásd lejjebb.
- **`Location`**: Az URL, amely meghívásakor az adott SessionInitiator handler-e aktivizálódik.
- **`id`**: (opcionális) Az adott SessionInitiator-re lehet ezen id által hivatkozni egyéb beállításoknál
- **`entityID`**: Az SP az itt megadott értékben szereplő IdP-hez irányítja az autentikálni kívánó felhasználót
- **`relayState`**: meghatározza, hogy...
- **`acsByIndex`**: igaz érték esetén él a lehetőség, hogy a megfelelő AssertionConsumerService-hez ne teljes URI-val forduljunk, hanem elég legyen csak annak indexét megadnunk.
- **`defaultACSIndex`**: az `acsByIndex="true"` esetén beállítható, hogy alapértelmezés szerint mely indexxel rendelkező AssertionConsumerService-t használjuk

SessionInitiator főbb típusai

- **SAML2 SessionInitiator** (Protocol Handler):

`type"SAML2"`
SAML2-es autentikációs folyamatot kezdeményez, és érti a SAML2 szabványon alapuló paramétereket. Mindenképp szükséges, hogy kapjon egy `entityID` paramétert, értékében egy valós IdP entityID-jával.

- **SHIB1 SessionInitiator** (Protocol Handler):
  `type"SHIB1"`
  Shibboleth 1.x-es autentikációs folyamatot kezdeményez, és SAML 1.1 szabványon alapuló paramétereket ért. Mindenképp szükséges, hogy kapjon egy `entityID` paramétert, értékében egy valós IdP entityID-jával.
- **SAMLDS SessionInitiator** (Discovery Handler):
  `type"SAMLDS"`
  Az `url` attribútum értékeként megadott helyre irányítja a böngészőt, ahol SAML2 Discovery Service-t vár. A SAML2DS ismeri az isPassive-ot.
- **WAYF SessionInitiator** (Discovery Handler):
  `type"WAYF"`
  Az `url` attribútum értékeként megadott helyre irányítja a böngészőt, ahol Shibboleth WAYF szolgáltatást vár.
- **Chaining SessionInitiator**:
  `type"Chaining"`
  Egy Chaining típusú SessionInitiator elem további SessionInitiator elemeket tartalmazhat, melyek felveszik a keret elem attribútumaiban meghatározott tulajdonságokat.

MetadataProvider

Ennél a szekciónál kell beállítani, hogy az SP milyen forrásokból jut hozzá a szükséges metaadatokhoz.

A források 3 fő típusa

- XML MetadataProvider
  `type"XML"`
  SAML2 szabványos XML fájlt tölt be a rendszer. A fájl lehet lokális, vagy távoli, webszerveren keresztül elérhető. Leggyakrabban használt típus. Példa:

```
<MetadataProvider type="XML" uri="https://metadata.eduid.hu/current/href.xml"
    backingFilePath="href.xml" reloadInterval="7200">
    <SignatureMetadataFilter certificate="href-metadata-signer-2020.crt"/>
</MetadataProvider>
```

  A tanúsítvány innen szerezhető be: https://metadata.eduid.hu/certs/href-metadata-signer-2020.crt
- Chaining MetadataProvider További `MetadataProvider`-(eke)t tartalmazhat.
- dinamikus, MDQ

```
  <MetadataProvider type="MDQ" id="href-2020" ignoreTransport="true"
baseUrl="https://mdx.eduid.hu/">
      <MetadataFilter type="Signature" certificate="href-metadata-signer-2020.crt"/>
      <MetadataFilter type="RequireValidUntil" maxValidityInterval="864000"/>
  </MetadataProvider>
```

A tanúsítvány innen szerezhető be: https://metadata.eduid.hu/certs/href-metadata-signer-2020.crt

ApplicationOverride

Amennyiben az SP több alkalmazást kezel, és ezek között az alkalmazások között vannak olyanok, melyeknek valamely tulajdonsága nem egyezik az SP alapértelmezettként megadott tulajdonságaival (jellemzően ilyen lehet pl. az entityID), akkor ezeket ebben a szekcióban felül lehet definiálni.

Kiegészítő beállítások

POST preservation

Ha legalább 2.2-es verziójú Shibboleth SP-t használunk, úgy lehetőségünk van egy olyan funkció beállítására, amely lehetővé teszi, hogy ha egy felhasználó valamilyen formba ír (pl. egy wikibe), akkor a küldés gomb megnyomásakor a shibboleth egy átmeneti helyen eltárolja a beírt adatokat. Ennek jelentősége, hogy ha írás közben lejárt volna a felhasználó sessionje, így alapértelmezés szerint a bejelentkező oldalra dobná a rendszer, ami által elveszne, amit begépelt, úgy bekapcsolt post preservation esetén ezek az adatok megmenekülnek, nem kell őket újra beírni.

A funkció bekapcsolásához a `<Sessions>` elem attribútumaként kell megadni az alábbi két név-érték párt.

- `postData="ss:mem"` , az érték mondja meg, hogy a form adatait az SP mely, a konfigurációs fájl elején definiált Storage Service-en keresztül tárolja. Alapértelmezés szerint a memóriában, de lehetőség van külső tároló megadására is. További információ a Storage Service-ről
- `postTemplate="/etc/shibboleth/postTemplate.html"`

Hiányossága a funkciónak, hogy ha a form tartalmaz `file` típusú `input` mezőt, akkor nem fog működni.

HREF integráció

1. Az SP-t regisztrálni kell a Resource Registry-ben
2. Le kell tölteni a metadatához tartozó tanúsítványt a https://metadata.eduid.hu/current/ címről, és elmenteni a shibboleth kofigurációs fájljait tartalmazó könyvtárba
3. A Metadata beállításoknál meg kell adni a HREF metadata elérhetőségét:

   https://metadata.eduid.hu/current/href.xml
4. Az `attribute-map.xml` fájlban el kell távolítani a kommentjeleket azon attribútumok elől, melyeket az SP használni kíván.
5. Újra kell indítani a shibboleth démont.

Változat #6
cziernorbert hozta létre 2026-04-14 13:22:38 CEST
cziernorbert frissítette 2026-04-17 10:55:35 CEST