# 5. Profile Configurations

## Profile Configurations

**Section Topics**

- SAML and OIDC profile configurations
- Profile configuration options
- Default vs. RP-specific profile configurations

### SAML and OIDC profile configurations

**Default (SAML) profile configurations**

- The profile configuration file is */opt/shibboleth-idp/conf/relying-party.xml*

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:util="http://www.springframework.org/schema/util"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:c="http://www.springframework.org/schema/c"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org
/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context http://www.springframework.org
/schema/context/spring-context.xsd
                           http://www.springframework.org/schema/util http://www.springframework.org
/schema/util/spring-util.xsd"
       default-init-method="initialize"
       default-destroy-method="destroy">

    <!--
    Unverified RP configuration, defaults to no support for any profiles. Add <ref> elements to the
list
    to enable specific default profile settings (as below), or create new beans inline to override
defaults.

    "Unverified" typically means the IdP has no metadata, or equivalent way of assuring the identity
and
    legitimacy of a requesting system. To run an "open" IdP, you can enable profiles here.
    -->
    <bean id="shibboleth.UnverifiedRelyingParty" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
            <!-- <bean parent="SAML2.SSO" p:encryptAssertions="false" /> -->
            </list>
        </property>
    </bean>

    <!--
    Default configuration, with default settings applied for all profiles, and enables
    the attribute-release consent flow.
    -->
    <bean id="shibboleth.DefaultRelyingParty" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
                <bean parent="Shibboleth.SSO" p:postAuthenticationFlows="attribute-release" />
                <ref bean="SAML1.AttributeQuery" />
                <ref bean="SAML1.ArtifactResolution" />
                <bean parent="SAML2.SSO" p:postAuthenticationFlows="attribute-release" />
                <ref bean="SAML2.ECP" />
                <ref bean="SAML2.Logout" />
                <ref bean="SAML2.AttributeQuery" />
                <ref bean="SAML2.ArtifactResolution" />
                <ref bean="Liberty.SSOS" />
            </list>
        </property>
    </bean>


...
```

**Default OIDC profile configurations**

- The OIDC profile configuration file is */opt/shibboleth-idp/conf/oidc-relying-party.xml*

```
...

    <!-- OIDC Profile Configurations. -->
    <bean id="OIDC.SSO" class="org.geant.idpextension.oidc.config.OIDCCoreProtocolConfiguration"
        p:securityConfiguration-ref="%{idp.security.oidc.config:shibboleth.oidc.
DefaultSecurityConfiguration}"
        p:iDTokenLifetime="%{idp.oidc.idToken.defaultLifetime:PT1H}"
        p:accessTokenLifetime="%{idp.oidc.accessToken.defaultLifetime:PT10M}"
        p:authorizeCodeLifetime="%{idp.oidc.authorizeCode.defaultLifetime:PT5M}"
        p:refreshTokenLifetime="%{idp.oidc.refreshToken.defaultLifetime:PT2H}"
        p:servletRequest-ref="shibboleth.HttpServletRequest"
        p:tokenEndpointAuthMethods="%{idp.oidc.tokenEndpointAuthMethods:client_secret_basic,
client_secret_post,client_secret_jwt,private_key_jwt}" />

    <bean id="OIDC.UserInfo" class="org.geant.idpextension.oidc.config.OIDCUserInfoConfiguration"
        p:securityConfiguration-ref="%{idp.security.oidc.config:shibboleth.oidc.
DefaultSecurityConfiguration}"
        p:servletRequest-ref="shibboleth.HttpServletRequest" />

    <bean id="OIDC.Registration" class="org.geant.idpextension.oidc.config.
OIDCDynamicRegistrationConfiguration"
        p:securityConfiguration-ref="%{idp.security.oidc.config:shibboleth.oidc.
DefaultSecurityConfiguration}"
        p:servletRequest-ref="shibboleth.HttpServletRequest"
        p:tokenEndpointAuthMethods="%{idp.oidc.dynreg.tokenEndpointAuthMethods:client_secret_basic,
client_secret_post,client_secret_jwt,private_key_jwt}" />

    <bean id="OIDC.Configuration" class="org.geant.idpextension.oidc.config.
OIDCProviderInformationConfiguration"
        p:securityConfiguration-ref="%{idp.security.oidc.config:shibboleth.oidc.
DefaultSecurityConfiguration}"
        p:servletRequest-ref="shibboleth.HttpServletRequest"/>

    <bean id="OAUTH2.Revocation" class="org.geant.idpextension.oauth2.config.
OAuth2TokenRevocationConfiguration"
        p:securityConfiguration-ref="%{idp.security.oidc.config:shibboleth.oidc.
DefaultSecurityConfiguration}"
        p:servletRequest-ref="shibboleth.HttpServletRequest"/>

...
```

**Relying party configuration**

- The main configuration file is */opt/shibboleth-idp/conf/relying-party.xml*

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:util="http://www.springframework.org/schema/util"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:c="http://www.springframework.org/schema/c"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org
/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context http://www.springframework.org
/schema/context/spring-context.xsd
                           http://www.springframework.org/schema/util http://www.springframework.org
/schema/util/spring-util.xsd"
       default-init-method="initialize"
       default-destroy-method="destroy">

    <import resource="oidc-relying-party.xml"/>

    <bean id="shibboleth.UnverifiedRelyingParty"  p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
                <bean parent="OIDC.Registration" />
                <bean parent="OIDC.Configuration" />
            </list>
        </property>
    </bean>

    <bean id="shibboleth.DefaultRelyingParty" p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
                <bean parent="SAML2.SSO" p:postAuthenticationFlows="attribute-release" />
                <ref bean="SAML2.Logout" />
                <bean parent="OIDC.SSO" p:postAuthenticationFlows="attribute-release" />
                <bean parent="OIDC.UserInfo"/>
                <bean parent="OAUTH2.Revocation"/>
            </list>
        </property>
    </bean>

...
```

## Profile configuration options

**Profile configuration options**

- https://github.com/CSCfi/shibboleth-idp-oidc-extension/wiki/ProfileConfigurations
- https://wiki.shibboleth.net/confluence/display/IDP30/RelyingPartyConfiguration
    - Shared options with all configurations
        - Standard security configuration and our extensions.
            - Used in various ways, depending on the context
                - OIDC.Configuration: signing + encryption configuration (credentials, algorithms) for openid-configuration
                - OIDC.Registration: which signing + encryption configuration details are supported
                - OIDC.SSO: which signing + encryption configuration is enabled
        - Inbound interceptor flows
        - Outbound interceptor flows
    - Client authenticable configuration options for OIDC.SSO, OIDC.Registration and OAUTH2.Revocation
        - Endpoint authentication methods
    - Flow-aware configuration options for OIDC.SSO and OIDC.Registration
        - Flags to enable implicit, hybrid and authorization code flows
        - Flag to enable refresh tokens
    - Flow-specific options
        - Multiple options especially for OIDC.SSO and OIDC.Registration (lifetimes, etc)
        - Post authentication flows, default authentication methods for OIDC.SSO

## Default vs. RP-specific profile configuration

**Profile configuration vs client metadata - overlapping configurations**

- Default profile configurations enable wide set of features
- Standard *shibboleth.RelyingPartyOverrides* mechanism can be used with OIDC RPs too

**Snippet of /opt/shibboleth-idp/conf/relying-party.xml**

```
...

    <util:list id="shibboleth.RelyingPartyOverrides">
        <bean parent="RelyingPartyByName"  p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" c:relyingPartyIds="test_rp">
            <property name="profileConfigurations">
                <list>
                    <bean parent="OIDC.SSO" />
                </list>
            </property>
        </bean>
    </util:list>

...
```

- Some of the profile configuration options have overlapping claims in the client metadata
    - E.g. token endpoint authentication methods

## Exercises

**Exercise 5.1 - Modifying default profile configuration**

1. Add additional audience ***test_api*** for all authenticated relying parties

**Snippet of /opt/shibboleth-idp/conf/relying-party.xml**

```
...

    <bean id="shibboleth.DefaultRelyingParty" p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" parent="RelyingParty">
        <property name="profileConfigurations">
            <list>
                <bean parent="SAML2.SSO" p:postAuthenticationFlows="attribute-release" />
                <ref bean="SAML2.Logout" />
                <bean parent="OIDC.SSO" p:postAuthenticationFlows="attribute-release" p:
additionalAudiencesForIdToken="test_api" />
                <bean parent="OIDC.UserInfo"/>
                <bean parent="OAUTH2.Revocation"/>
            </list>
        </property>
    </bean>

...
```

2. Verify that the additional audience is visible in the *id_token.*

**Hints, Tips and Result**

```
[OIDC_CLAIM_aud] => test_rp,test_api
```

**Exercise 5.2 - Modifying RP-specific profile configuration**

1. Remove *postAuthenticationFlows* and *additionalAudiencesForIdToken* settings for *test_rp*.

**Snippet of /opt/shibboleth-idp/conf/relying-party.xml**

```
...

    <util:list id="shibboleth.RelyingPartyOverrides">
        <bean parent="RelyingPartyByName"  p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" c:relyingPartyIds="test_rp">
            <property name="profileConfigurations">
                <list>
                    <bean parent="OIDC.SSO" />
                </list>
            </property>
        </bean>
    </util:list>

...
```

2. Are the additional audiences now visible for *test_rp* as they are defined in **shibboleth.DefaultRelyingParty**? Why?

**Hints, Tips and Result**

```
[OIDC_CLAIM_aud] => test_rp


They are not, because the settings from OICD.SSO defined in oidc-relying-party.xml are inherited, not
OIDC.SSO settings from shibboleth.DefaultRelyingParty.
```

3. What happens if you configure that only *private_key_jwt* is accepted as the token endpoint authentication method for *test_rp*?

**Snippet of /opt/shibboleth-idp/conf/relying-party.xml**

```
...

    <util:list id="shibboleth.RelyingPartyOverrides">
        <bean parent="RelyingPartyByName"  p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" c:relyingPartyIds="test_rp">
            <property name="profileConfigurations">
                <list>
                    <bean parent="OIDC.SSO" p:tokenEndpointAuthMethods="private_key_jwt" />
                </list>
            </property>
        </bean>
    </util:list>

...
```

**Snippet of /opt/shibboleth-idp/logs/idp-process.log**

```
...
2018-10-04 12:45:49,839 - DEBUG [org.geant.idpextension.oidc.profile.impl.
InitializeRelyingPartyContext:170] - Attaching RelyingPartyContext for rp test_rp
2018-10-04 12:45:49,839 - DEBUG [org.geant.idpextension.oidc.profile.impl.
InitializeRelyingPartyContext:175] - Profile Action InitializeRelyingPartyContext: Setting the rp
context verified
2018-10-04 12:45:49,840 - DEBUG [net.shibboleth.idp.profile.impl.SelectRelyingPartyConfiguration:136]
- Profile Action SelectRelyingPartyConfiguration: Found relying party configuration EntityNames
[test_rp,] for request
2018-10-04 12:45:49,843 - WARN [org.geant.idpextension.oidc.profile.impl.
ValidateEndpointAuthentication:250] - Profile Action ValidateEndpointAuthentication: The requested
method client_secret_basic is not enabled
2018-10-04 12:45:49,843 - WARN [org.geant.idpextension.oidc.profile.impl.
ValidateEndpointAuthentication:230] - Profile Action ValidateEndpointAuthentication: Unsupported
client authentication method client_secret_basic
2018-10-04 12:45:49,853 - WARN [org.opensaml.profile.action.impl.LogEvent:105] - A non-proceed event
occurred while processing the request: AccessDenied
...
```

**Exercise 5.3 - Advanced access-control configuration with context-check**

The goal of this exercise is to configure the **test_rp** application to be only accessible for **teppo2** user. Shibboleth IdP provides context-check interceptor for this purpose.

1. Add *context-check* post authentication flow to the relying party configuration

**Snippet of /opt/shibboleth-idp/conf/relying-party.xml**

```
...

    <util:list id="shibboleth.RelyingPartyOverrides">
        <bean parent="RelyingPartyByName"  p:responderIdLookupStrategy-ref="
profileResponderIdLookupFunction" c:relyingPartyIds="test_rp">
            <property name="profileConfigurations">
                <list>
                    <bean parent="OIDC.SSO" p:postAuthenticationFlows="context-check"/>
                </list>
            </property>
        </bean>
    </util:list>

...
```

2. Edit */opt/shibboleth-idp/conf/intercept/context-check-intercept-config.xml* for your needs. HINT! The existing file contains good basis, find out from attribute-resolver which is the username in your configuration.

**Hints, Tips and Result**

```
...


    <bean id="shibboleth.context-check.Condition" parent="shibboleth.Conditions.AND">
        <constructor-arg>
            <list>
                <bean parent="shibboleth.Conditions.RelyingPartyId" c:candidates="#{{'test_rp'}}" />
                <bean class="net.shibboleth.idp.profile.logic.SimpleAttributePredicate"
                        p:useUnfilteredAttributes="true">
                    <property name="attributeValueMap">
                        <map>
                            <entry key="uid">
                                <list>
                                    <value>teppo2</value>
                                </list>
                            </entry>
                        </map>
                    </property>
                </bean>
            </list>
        </constructor-arg>
    </bean>


...
```

3. Restart IDP service and try to access the test RP with *teppo* and *teppo2* (same password). You can logout the user via */idp/profile/Logout* -endpoint.

**Snippet of /opt/shibboleth-idp/logs/idp-process.log**

```
...
2018-10-05 01:20:37,187 - INFO [Shibboleth-Audit.SSO:276] -
20181005T012037Z|AuthenticationRequest||test_rp|http://csc.fi/ns/profiles/oidc/sso/browser|https://192.
168.0.150|||teppo|||||
...
```