

3. TechEx2019 - Configuring Authentication

Configuring Authentication

Section Topics

- OIDC Authentication Request
- Authentication flow selection for OIDC authentication request

OIDC Authentication Request

Authentication Request

http://openid.net/specs/openid-connect-core-1_0.html#AuthRequest

http://openid.net/specs/openid-connect-core-1_0.html#IndividualClaimsRequests

- Requested Authentication Context Class Reference values are defined by acr_values request parameter and acr claims request.
 - Request may be essential or non-essential.
- prompt has options
 - none, The Authorization Server MUST NOT display any authentication or consent user interface pages. Translates to isPassive.
 - login, The Authorization Server SHOULD prompt the End-User for reauthentication. Translates to forceAuthn.
 - consent, The Authorization Server SHOULD prompt the End-User for consent before returning information to the Client
 - select_account, The Authorization Server SHOULD prompt the End-User to select a user account. Not supported by current implementation.
- max_age, specifies the allowable elapsed time in seconds since the last time the End-User was actively authenticated by the OP.

Authentication flow selection for OIDC authentication request

OIDC Extension Authentication Configuration

<https://github.com/CSCfi/shibboleth-idp-oidc-extension/wiki/AuthenticationConfiguration>

Exercises

Exercise 3.1 - ACR value

ACR value

1. RP does not request for ACR by default. Modify the RP to request for password authentication.

```
nano +418 /etc/httpd/conf.d/auth_openidc.conf

OIDCAuthRequestParams acr_values=password

systemctl restart httpd
```

2. Run authentication sequence and verify from the logs that password authentication is being requested

Hints, Tips and Result

Parameters:

```
scope:openid profile email address phone
acr_values:password
response_type:code
state:UeWkLZsv4qfSh5kmzN2lTsHqj0E
redirect_uri:https://195.148.31.24:8443/protected/redirect_uri
nonce:dC3KenQHX-8RleSXrkNQajljL0NuxpafDOiAZ-5vHnk
client_id:test_rp
```

3. OP does not seem to respond with ACR claim value - there is no [OIDC_CLAIM_acr] on your landing page. The authentication method principals are set in `/opt/shibboleth-idp/conf/authn/general-authn.xml`. See <https://github.com/CSCfi/shibboleth-idp-oidc-extension/wiki/AuthenticationConfiguration> on how to add "password" authentication method principal for password flow.

Hints, Tips and Result

```
<bean id="authn/Password" parent="shibboleth.AuthenticationFlow"
      p:passiveAuthenticationSupported="true"
      p:forcedAuthenticationSupported="true" >
  <property name="supportedPrincipals">
    <list>
      <bean parent="shibboleth.SAML2AuthnContextClassRef"
            c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes>PasswordProtectedTransport" />
      <bean parent="shibboleth.SAML2AuthnContextClassRef"
            c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes>Password" />
      <bean parent="shibboleth.SAML1AuthenticationMethod"
            c:method="urn:oasis:names:tc:SAML:1.0:am:password" />
      <bean parent="shibboleth.OIDCAuthnContextClassReference"
            c:classRef="password" />
    </list>
  </property>
</bean>
```

- a. Verify the ACR claim value from your landing page.

Hints, Tips and Result

```
[OIDC CLAIM acr] => password
```

- b. The value is set in action AddAcrToldToken. Verifv from the logs the action has taken place.

Hints, Tips and Result

```
grep AddAcrToIDToken /opt/shibboleth-idp/logs/idp-process.log
```

```
2018-09-10 04:52:49,068 DEBUG [org.geant.idpextension.oidc.profile.impl.AddAcrToIDToken:58] - Profile Action AddAcrToIDToken: Setting acr to id token
2018-09-10 04:52:49,079 - DEBUG [org.geant.idpextension.oidc.profile.impl.AddAcrToIDToken:60] - Profile Action AddAcrToIDToken: Updated token {"sub":"VUG4777YP3NNU5KRFESX6SKRAPXLE4MI","aud":["_443085776b9c4370eeb8b7481b99d8e31"],"acr":{"password"},"auth_time":1536555148,"iss":"https://192.168.0.150"},"exp":1536558769,"iat":1536555169}
```

- c. Client receives the ACR claim in ID Token. Locate ID Token from Token Response (field "id_token") in the logs. Decode the ID Token and verify the existence and value of ACR claim. Use for instance <https://jwt.io/> to decode the ID Token.

Hints, Tips and Result

```
grep id_token /opt/shibboleth-idp/logs/idp-process.log
```

```
Content:{"access_token":"AAdzZWNYZXQxV9G7fr...tJ9","id_token":  
eyJraWQiOiJ0ZXN0a2V5UmlLCjhbGciOiJSUzIiNiJ9.  
eyJhdF9oYXNoIjoiaW9wTFlpbjNSYmVmUmXJamdjWkNQQSIsInNlYiI6IlZVRzQ3NzdZUDNOTVU1SJGRVNyNlNLUkFQWEeF  
NElJIiwiaXYvkiJoixzQMza4NTc3NmI5YzkyZnBLWI4Yjc0ODFiOTlkYmYUzIiwiaWF0IjoiYWNyYjoicGFzc3dvcmQmClJhdXRoX3Rp  
bWUiOjEiMzYlNTUxNDgsImVzcyI6ImhhOmR2bWVzC8xOTUuMTUy4LjaUmtUWlwiIiwiaXhwIjoxNmTM2NTU4NzY5LClUpYXQiojE1  
MzYlNTUxNjk5Im5vbmlNIjoiaUmntPNjhSmVHbEoybD0xaOU9XamVCWWJidlNiNVlU3Mza3ZwMldastVOQSJ9.  
kIF8PKmVVEULbas6gsaiOcho51W_614V6vwXmfIraw2RosTamSoyFVjCT_RxpO5RRHQVbpHVVB9Q91QcwEOZjnyTU4VCP2ZQ  
N7EqLHP8gTLQ_SihooQ8CB5hp_w-ijd4Pzw2SSdk44X2mdfOJv0W0TFtwzRFkrprp8pleYqz1YIMh0DH31pvU8AbasmZ-  
K7ehJYegln35rWzz9ve7tqFTblAB0Gj4syGYCCr6oAKPrMmo9LiYrpe92viFFAGirZFthz9Mjd1THZOLWNd62tpJJLgm6RqDD  
PMolM2NrPahcYvdHakErM9S5Xp0hHoyn6J4ihArDGqWNrgI3y9_6GU35g"},"token_type":"Bearer","expires_in":  
600}
```

```
{
  kid: "testkeyRS",
  alg: "RS256"
}. {
  at_hash: "iopLYin3RbefRlIjgcZCPA",
  sub: "VUG4777YP3NMU5KRFESEX6SKRAPXLE4MI",
  aud: "_443085776b9c4370eeb8b7481b99dbe3",
  acr: "password",
  auth_time: 1536555148,
  iss: "https://\//192.168.0.150",
  exp: 1536558769,
  iat: 1536555169,
  nonce: "RfO68gJeG1J2oLZ9OWjeBYbbvSb5R5Ss3kvp3WZI5NA"
}. [signature]
```

Exercise 3.2 - Flow selection

Select flow by ACR value

1. Modify client to request for ACR value "ipaddress".

```
nano +418 /etc/httpd/conf.d/auth_openidc.conf

OIDCAuthRequestParams acr_values=ipaddress

systemctl restart httpd
```

2. Run authentication sequence.
3. Follow log entries, identify requested ACR and what is actually sent back as ACR in the response. Can you explain why they do not match?

Hints, Tips and Result

Request:

Parameters:

```
scope:openid email
acr_values:ipaddress
response_type:code
state:PunYlKqqJA6b57C9DPKbt5cvdq8
redirect_uri:https://192.168.0.150:8443/protected/redirect_uri
nonce:4kcflIwolApMDMMfKxYte9IvU5aIULIWc38dW_XG5lc
client_id:_443085776b9c4370eeb8b7481b99dbe3
```

```
2018-09-10 07:24:54,250 - DEBUG [org.geant.idpextension.oidc.profile.impl.ProcessRequestedAuthnContext:
184] - Profile Action ProcessRequestedAuthnContext: Created preferred principal context
```

```
2018-09-10 07:25:00,905 - DEBUG [org.geant.idpextension.oidc.profile.impl.AddAcrToIDToken:60] -
Profile Action AddAcrToIDToken: Updated token {"sub":"VUG4777YP3NMU5KRFESX6SKRAPXLE4MI","aud":
["_443085776b9c4370eeb8b7481b99dbe3"],"acr":"password","auth_time":1536564300,"iss":"https://192.
168.0.150","exp":1536567900,"iat":1536564300}
```

4. Activate IPAddress authentication, configure it for acr "ipaddress" and verify it is actually used as preferred authentication flow.

Hints, Tips and Result

```
nano /opt/shibboleth-idp/conf/authn/ipaddress-authn-config.xml
```

```
#Modify the existing IPAddress authentication mapping
```

```
<util:map id="shibboleth.authn.IPAddress.Mappings">
  <entry key="teppo">
    <list>
      <value>0.0.0.0/0</value>
      <value>:::1/128</value>
    </list>
  </entry>
</util:map>
```

```
nano /opt/shibboleth-idp/conf/authn/general-authn.xml
```

```
#Modify the existing IPAddress authentication
```

```
<bean id="authn/IPAddress" parent="shibboleth.AuthenticationFlow"
  p:passiveAuthenticationSupported="true"
  p:lifetime="PT60S" p:inactivityTimeout="PT60S">
  <property name="supportedPrincipals">
    <list>
      <bean parent="shibboleth.SAML2AuthnContextClassRef"
        c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol" />
      <bean parent="shibboleth.OIDCAuthnContextClassReference"
        c:classRef="ipaddress" />
    </list>
  </property>
</bean>
```

```
nano +121 /opt/shibboleth-idp/conf/idp.properties
```

```
#Add IPAddress as active authentication option
```

```
idp.authn.flows=IPAddress|Password
```

After restart and performing new authentication sequence you should see:

```
2018-09-10 15:15:01,007 - DEBUG [org.geant.idpextension.oidc.profile.impl.AddAcrToIDToken:60] -
Profile Action AddAcrToIDToken: Updated token {"sub":"VUG4777YP3NMU5KRFESX6SKRAPXLE4MI","aud":
["_443085776b9c4370eeb8b7481b99dbe3"],"acr":"ipaddress","auth_time":1536592497,"iss":"https://192.
168.0.150","exp":1536596100,"iat":1536592500}
```

5. Create essential claims request for ACR values urn:mace:incommon:iap:silver and urn:mace:incommon:iap:bronze. This is now request that must (but will not) be met.

```
nano +418 /etc/httpd/conf.d/auth_openidc.conf

OIDCAuthRequestParams claims=%7B%22id_token%22%3A%7B%22acr%22%3A+%7B%22essential%22%3A+true%2C%
22values%22%3A+%5B%22urn%3Amace%3Aincommon%3Aiap%3Asilver%22%2C%22urn%3Amace%3Aincommon%3Aiap%3Abronze%
22%5D%7D%7D%7D

systemctl restart httpd
```

Start a new authentication sequence and see the result. This behavior should be something you are used to in SAML2 world.

Hints, Tips and Result

```
2018-09-10 15:30:58,586 - DEBUG [net.shibboleth.idp.authn.impl.SelectAuthenticationFlow:395] - Profile
Action SelectAuthenticationFlow: Specific principals requested with 'exact' operator:
[AuthenticationContextClassReferencePrincipal{authnContextClassReference=urn:mace:incommon:iap:
silver}, AuthenticationContextClassReferencePrincipal{authnContextClassReference=urn:mace:incommon:iap:
bronze}]
2018-09-10 15:30:58,586 - DEBUG [net.shibboleth.idp.authn.impl.SelectAuthenticationFlow:411] - Profile
Action SelectAuthenticationFlow: No active results available, selecting an inactive flow
2018-09-10 15:30:58,586 - DEBUG [net.shibboleth.idp.authn.impl.SelectAuthenticationFlow:432] - Profile
Action SelectAuthenticationFlow: Checking for an inactive flow compatible with operator 'exact' and
principal 'urn:mace:incommon:iap:silver'
2018-09-10 15:30:58,587 - DEBUG [net.shibboleth.idp.authn.impl.SelectAuthenticationFlow:432] - Profile
Action SelectAuthenticationFlow: Checking for an inactive flow compatible with operator 'exact' and
principal 'urn:mace:incommon:iap:bronze'
2018-09-10 15:30:58,588 - INFO [net.shibboleth.idp.authn.impl.SelectAuthenticationFlow:453] - Profile
Action SelectAuthenticationFlow: None of the potential authentication flows can satisfy the request
2018-09-10 15:30:58,650 - DEBUG [org.geant.idpextension.oidc.profile.impl.
AbstractBuildErrorResponseFromEvent:123] - Profile Action BuildAuthenticationErrorResponseFromEvent:
No mapped event found for RequestUnsupported, creating general invalid_request
2018-09-10 15:30:58,651 - DEBUG [org.geant.idpextension.oidc.profile.impl.
AbstractBuildErrorResponseFromEvent:131] - Profile Action BuildAuthenticationErrorResponseFromEvent:
ErrorResponse successfully set as the outbound message
2018-09-10 15:30:58,652 - DEBUG [org.geant.idpextension.oidc.encoding.impl.NimbusResponseEncoder:148]
- Outbound response
Headers:
    Location:https://192.168.0.150:8443/protected/redirect_uri?error_description=Invalid+request%
3A+RequestUnsupported&state=T_2Ez6onqjVzJFT8T9d3zjzpkps&error=invalid_request
```

6. Let's return to the original state

```
nano +418 /etc/httpd/conf.d/auth_openidc.conf

#OIDCAuthRequestParams claims=%7B%22id_token%22%3A%7B%22acr%22%3A+%7B%22essential%22%3A+true%2C%
22values%22%3A+%5B%22urn%3Amace%3Aincommon%3Aiap%3Asilver%22%2C%22urn%3Amace%3Aincommon%3Aiap%3Abronze%
22%5D%7D%7D%7D

service httpd restart

nano +121 /opt/shibboleth-idp/conf/idp.properties

#Remove IPAddress as active authentication option
idp.authn.flows=Password

systemctl restart shibboleth-idp
```

Exercise 3.3 - Prompt

prompt=consent and other prompt parameters

1. Set the consent to be requested

```
nano +417 /etc/httpd/conf.d/auth_openidc.conf

OIDCAuthRequestParams prompt=consent

systemctl restart httpd
```

2. Authenticate the user few times and verify this feature actually works as expected.
3. Try different combinations of the parameter