

Pilot resources

SPs to test IdP attribute release and returned authentication context:

- CILogon: <https://test.cilogon.org/testidp>
- ELIXIR: <https://perun.elixir-czech.cz/refeds-af-demo/> (select "Add new institution" from the bottom of the page to give a try with any eduGAIN IdP)
- EGI Check-in: <https://snf-666522.vm.okeanos.grnet.gr/ssp/module.php/core/authenticate.php?as=dev-egi-sp> (TBD - only works with social logins as registration with eduGAIN is still pending)
- SWITCHaai's common test SP: <https://attribute-viewer.aai.switch.ch/aai/>

Configuration examples - Shibboleth

The below config seems to be working swimmingly for me.

- If the SP doesn't ask for a specific aCC ref the IdP will choose the authentication method
- If mfa session is valid for any reason aCC ref = mfa is returned unless something else is requested
- IdP respects any aCC ref that is requested by executing that auth method and returning the appropriate aCC ref

There is a caveat and it is that most of the time the IdP will return a aCC ref that is not a part of the SAML2 spec: "<https://refeds.org/profile/sfa>" or "<https://refeds.org/profile/mfa>". This shouldn't be that big of a deal I think because if the SP cares about the aCC ref it should ask for something specific.

There is a lot going on in the config examples that are not specific to the RAF pilot. They are left there to give ideas about how to implement more complex login flows. Nothing should be taken as copy-paste configs ready to be deployed as-is.

Specifically there is also "authn/storage" and "authn/store" login methods that implement a rolling time window authentication piggybacking of a conventional one (TOTP in our case). We are a beta tester of this feature so those configs, too, will probably change in the future.

conf/idp.properties:

```
idp.authn.flows= MFA
```

conf/authn/general-authn.xml:

```
<util:list id="shibboleth.AvailableAuthenticationFlows">

    <bean id="authn/SPNEGO" parent="shibboleth.AuthenticationFlow"
        p:nonBrowserSupported="false"
        p:activationCondition-ref="shibboleth.SPNEGO.ActivationCondition">
        <property name="supportedPrincipals">
            <list>
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:federation:authentication:windows" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/sfa" />
            </list>
        </property>
    </bean>

    <bean id="authn/RemoteUser" parent="shibboleth.AuthenticationFlow"
        p:nonBrowserSupported="false"
        p:activationCondition-ref="shibboleth.X509.ActivationCondition">
        <property name="supportedPrincipals">
            <list>
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:X509" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/sfa" />
            </list>
        </property>
    </bean>

    <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.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/sfa" />
            </list>
        </property>
    </bean>

    <bean id="authn/storage" parent="shibboleth.AuthenticationFlow"
        p:nonBrowserSupported="false"
        p:passiveAuthenticationSupported="true"
        p:forcedAuthenticationSupported="true">
        <property name="supportedPrincipals">
            <list>
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/mfa" />
            </list>
        </property>
    </bean>

    <bean id="authn/store" parent="shibboleth.AuthenticationFlow"
        p:passiveAuthenticationSupported="true"
        p:forcedAuthenticationSupported="true" />

    <bean id="authn/SocialUserOpenIDConnect" parent="shibboleth.AuthenticationFlow"
        p:nonBrowserSupported="false"
        p:forcedAuthenticationSupported="true">
        <property name="supportedPrincipals">
            <list>
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/mfa" />
            </list>
        </property>
    </bean>

    <bean id="authn/MFA" parent="shibboleth.AuthenticationFlow"
        p:passiveAuthenticationSupported="false"
        p:forcedAuthenticationSupported="true">
        <property name="supportedPrincipals">
            <list>
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:federation:authentication:windows" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:X509" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient" />
                <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.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/sfa" />
                <bean parent="shibboleth.SAML2AuthnContextClassRef"
                    c:classRef="https://refeds.org/profile/mfa" />
            </list>
        </property>
    </bean>

</util:list>

<util:map id="shibboleth.AuthenticationPrincipalWeightMap">
    <entry>
        <key>

```

```

        <bean parent="shibboleth.SAML2AuthnContextClassRef"
            c:classRef="https://refeds.org/profile/mfa" />
    </key>
    <value>2</value>
</entry>
<entry>
    <key>
        <bean parent="shibboleth.SAML2AuthnContextClassRef"
            c:classRef="https://refeds.org/profile/sfa" />
    </key>
    <value>1</value>
</entry>
</util:map>

```

conf/authn/mfa-authn-config.xml:

```

<util:map id="shibboleth.authn.MFA.TransitionMap">
    <entry key="">
        <bean parent="shibboleth.authn.MFA.Transition" p:nextFlowStrategy-ref="firstFactor" />
    </entry>

    <entry key="authn/RemoteUser">
        <bean parent="shibboleth.authn.MFA.Transition">
            <property name="nextFlowStrategyMap">
                <map>
                    <entry key="ReselectFlow" value="authn/Password" />
                    <entry key="proceed" value="authn/storage" />
                </map>
            </property>
        </bean>
    </entry>

    <entry key="authn/SPNEGO">
        <bean parent="shibboleth.authn.MFA.Transition">
            <property name="nextFlowStrategyMap">
                <map>
                    <entry key="ReselectFlow" value="authn/Password" />
                    <entry key="proceed" value="authn/storage" />
                </map>
            </property>
        </bean>
    </entry>

    <entry key="authn/Password">
        <bean parent="shibboleth.authn.MFA.Transition" p:nextFlow="authn/storage" />
    </entry>

    <entry key="authn/storage">
        <bean parent="shibboleth.authn.MFA.Transition">
            <property name="nextFlowStrategyMap">
                <map>
                    <!-- If we do not have a storage event we perform second factor -->
                    <entry key="ReselectFlow" value-ref="secondFactor" />
                </map>
            </property>
        </bean>
    </entry>

    <entry key="authn/SocialUserOpenIDConnect">
        <bean parent="shibboleth.authn.MFA.Transition" p:nextFlow="authn/store" />
    </entry>

</util:map>

<util:map id="customObjectsForMFAFlow">
    <entry key="SPNEGOActivationCondition" value-ref="shibboleth.SPNEGO.ActivationCondition" />
    <entry key="X509ActivationCondition" value-ref="shibboleth.X509.ActivationCondition" />
</util:map>

```

```

<bean id="firstFactor" parent="shibboleth.ContextFunctions.Scripted" factory-method="inlineScript"
  p:customObject-ref="customObjectsForMFAFlow">
  <constructor-arg>
    <value>
      <![CDATA[
        logger = Java.type( "org.slf4j.LoggerFactory" ).getLogger( "firstFactor" );
        nextFlow = "authn/Password";

        isX509Activated = custom.get( "X509ActivationCondition" ).apply( input );
        isSPNEGOActivated = custom.get( "SPNEGOActivationCondition" ).apply( input );

        if ( isX509Activated ) {
          logger.debug( "Selected first factor: X509" );
          nextFlow = "authn/RemoteUser";
        } else if ( isSPNEGOActivated ) {
          logger.debug( "Selected first factor: SPNEGO" );
          nextFlow = "authn/SPNEGO";
        } else {
          logger.debug( "Selected first factor: Password" );
        }

        nextFlow;
      ]]>
    </value>
  </constructor-arg>
</bean>

<bean id="secondFactor" parent="shibboleth.ContextFunctions.Scripted" factory-method="inlineScript"
  p:customObject-ref="shibboleth.AttributeResolverService">
  <constructor-arg>
    <value>
      <![CDATA[
        nextFlow = "authn/store";
        logger = Java.type( "org.slf4j.LoggerFactory" ).getLogger( "secondFactor" );

        authCtx = input.getSubcontext( "net.shibboleth.idp.authn.context.AuthenticationContext" );
        mfaCtx = authCtx.getSubcontext( "net.shibboleth.idp.authn.context.
MultiFactorAuthenticationContext" );
        if ( mfaCtx.isAcceptable() ) {
          logger.debug( 'Second factor auth does not need to run' );
        } else {
          logger.debug( "Second factor auth needs to run" );
          nextFlow = "authn/SocialUserOpenIDConnect";

          usernameLookupStrategyClass = Java.type( "net.shibboleth.idp.session.context.navigate.
CanonicalUsernameLookupStrategy" );
          usernameLookupStrategy = new usernameLookupStrategyClass();
          resCtx = input.getSubcontext( "net.shibboleth.idp.attribute.resolver.context.
AttributeResolutionContext", true );
          resCtx.setPrincipal( usernameLookupStrategy.apply( input ) );
          resCtx.getRequesteIdPAttributeNames().add( "stepupUID" );
          resCtx.getRequesteIdPAttributeNames().add( "stepupEPN" );
          resCtx.getRequesteIdPAttributeNames().add( "stepupMobile" );
          resCtx.resolveAttributes( custom );

          // Pass the resolved attributes to context
          suCtx = authCtx.getSubcontext( "fi.okm.mpass.idp.authn.impl.
SocialUserOpenIdConnectContext", true );
          suCtx.setResolvedIdPAttributes( resCtx.getResolvedIdPAttributes() );
          input.removeSubcontext( resCtx );
        }

        nextFlow;
      ]]>
    </value>
  </constructor-arg>
</bean>

```

SimpleSAMLphp in Proxy mode

There is a need to patch the SimpleSAMLphp in order to pass the authnContextClassRef from the SP behind the proxy to the upper IdP: <https://github.com/simplesamlphp/simplesamlphp/pull/833>. The patch has been tested with ELIXIR AAI. Test can be done at <https://perun.elixir-czech.cz/refeds-af-demo/>.