X.509 Login Handler

- *The x509-login-handler provides an X.509 user certificate login (authentication) using SSL client authentication by the certificate within a users web browser.*
- *The x509-login-handler can be used as an alternative to or in conjuction with the UsernamePasswordLoginHandler.*
- *The x509-login-handler implements an authentication handler for the Shibboleth IdP, propagates the **subject** to the IdP and set the authentication context class "**urn:oasis:names:tc:SAML:2.0:ac:classes:X509**".*

Installation and configuration

Download

Either download the latest release of the X.509 login handler for Shibboleth from the project site:

https://forge.switch.ch/redmine/projects/x509-handler/files

or alternatively, get the latest source code from trunk:

Get latest code from trunk

```
svn export https://subversion.switch.ch/svn/general/aai/java-idp-x509-login-handler/trunk/ java-idp-x509-login-handler
cd java-idp-x509-login-handler
mvn package
```

Installation

Locate the x509-login-handler JAR file and copy it to the library directory of the Shibboleth Identity Provider installation directory.

```
cp x509-login-handler-X.Y.Z.jar $IDP_INSTALL_DIR/lib
```

Configuration

Login pages

You will want to edit the login pages for the X.509 login. It is also possible to add an X.509 login button on the Username/Password login page.

Certificate only login

- You can use `java-idp-x509-login-handler/examples/x509-login.jsp` as an example.
- Adjust the form action URLs.
- Place the modified file in the Identity Provider webapp directory `$IDP_INSTALL_DIR/src/main/webapp/`

Certificate login included in UsernamePassword login page

- You can use `java-idp-x509-login-handler/examples/login.jsp` as an example.
- Adjust the form action URLs.
- Place it in `the Identity Provider webapp directory $IDP_INSTALL_DIR/src/main/webapp/` (Backup your orginal login.jsp before this step!)

Caution: This may result in unexpected behaviour of the IdP from the perspective of the SP if it specifically required username/password authentication method and the user logs in with x509 authentication method.

Web application

Enable the the X.509 login servlets in `$IDP_INSTALL_DIR/src/main/webapp/WEB-INF/web.xml`:

```
<webapp>
    <!-- ... -->
    <!-- X509 login handler -->
    <servlet>
      <servlet-name>X509LoginHandler</servlet-name>
      <servlet-class>ch.SWITCH.aai.idp.x509.X509LoginHandler</servlet-class>
    </servlet>

    <servlet-mapping>
      <servlet-name>X509LoginHandler</servlet-name>
      <url-pattern>/Authn/X509</url-pattern>
    </servlet-mapping>

    <servlet>
      <servlet-name>X509LoginServlet</servlet-name>
      <servlet-class>ch.SWITCH.aai.idp.x509.X509LoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>
      <servlet-name>X509LoginServlet</servlet-name>
      <url-pattern>/Authn/X509/Login</url-pattern>
    </servlet-mapping>

    <!-- x509 login page -->
    <servlet>
        <servlet-name>x509_jsp</servlet-name>
        <jsp-file>/x509-login.jsp</jsp-file>
    </servlet>

    <servlet-mapping>
        <servlet-name>x509_jsp</servlet-name>
        <url-pattern>/x509-login</url-pattern>
    </servlet-mapping>
    <!-- ... -->
</webapp>
```

Apache configuration

```
<Location /idp/Authn/X509/Login>
    SSLCACertificateFile /etc/ssl/certs/client-authn.crt
    SSLVerifyClient require
    SSLVerifyDepth 5
    SSLOptions -StdEnvVars +ExportCertData
    SSLRequire ( %{SSL_CLIENT_I_DN_CN} eq "Example Organization Personal CA" and \
                 %{SSL_CLIENT_I_DN_O} eq "Example Organization" and \
                 %{SSL_CLIENT_S_DN_Email} =~  m/^.+@example\.org$/ )
</Location>
```

Handler configuration

In `$IDP_CONFIG_DIR/handler.xml`, add the xsd schema in the `<ProfileHandlerGroup>` and the login handler:

```
<ProfileHandlerGroup xmlns="urn:mace:shibboleth:2.0:idp:profile-handler" xmlns:x509="http://www.switch.ch/aai
/idp/x509"
                     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                     xsi:schemaLocation="urn:mace:shibboleth:2.0:idp:profile-handler
                         classpath:/schema/shibboleth-2.0-idp-profile-handler.xsd
                         http://www.switch.ch/aai/idp/x509 classpath:/schema/x509-login-handler.xsd">
    <!-- ... -->
    <!-- Login Handlers -->
    <!-- X509 Login Handler -->
    <!-- configuration attributes: -->
    <!-- loginPageURL (required): URL of JSP page with login form -->
    <!-- authenticationServletURL (required): Client AuthN protected page -->
    <!-- cookieDomain (optional): set domain of login context cookie for -->
    <!--     spefic environments, e.g. if authenticationServlet runs under -->
    <!--     a different domain name than the IdP -->
    <LoginHandler xsi:type="x509:X509"
                  loginPageURL="/x509-login"
                  authenticationServletURL="/Authn/X509/Login">
        <AuthenticationMethod>
            urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        </AuthenticationMethod>
    </LoginHandler>
    <!-- ... -->
</ProfileHandlerGroup>
```

Attribute resolver configuration

In `$IDP_CONFIG_DIR/attribute-resolver.xml`:

- add new attribute definitions to extract the principal from the certificate
- add dependency on the attributes and filter in data connector

```
<AttributeResolver>
    <!-- Provides the subjectDN from the certificate as attribute -->
    <resolver:AttributeDefinition xsi:type="Script"
                                  xmlns="urn:mace:shibboleth:2.0:resolver:ad"
                                  dependencyOnly="true"
                                  id="x500Principal">
        <Script><![CDATA[
            importPackage(Packages.edu.internet2.middleware.shibboleth.common.attribute.provider);
            importPackage(Packages.javax.security.auth.x500);

            x500Principal = new BasicAttribute("x500Principal");
            subject = requestContext.getUserSession().getSubject();
            if (subject != null) {
                x500Principal.getValues().addAll(subject.getPrincipals(X500Principal("").getClass()));
            }
        ]]></Script>

    </resolver:AttributeDefinition>

    <!-- Provides the subjectAltNames of type rfc822Name from the certificate as attribute -->
    <resolver:AttributeDefinition xsi:type="Script"
                                  xmlns="urn:mace:shibboleth:2.0:resolver:ad"
                                  dependencyOnly="true"
                                  id="x500SubjectAltNameEMailPrincipal">
        <Script><![CDATA[
            importPackage(Packages.edu.internet2.middleware.shibboleth.common.attribute.provider);
            importPackage(Packages.ch.SWITCH.aai.idp.x509.principals);

            x500SubjectAltNameEMailPrincipal = new BasicAttribute("x500SubjectAltNameEMailPrincipal");
            subject = requestContext.getUserSession().getSubject();
            if (subject != null) {
                x500SubjectAltNameEMailPrincipal.getValues().addAll(subject.getPrincipals(EMailPrincipal("").
getClass()));
            }
        ]]></Script>

    </resolver:AttributeDefinition>

    ...

    <resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory" ... >
      <resolver:Dependency ref="x500Principal" />
      <resolver:Dependency ref="x500SubjectAltNameEMailPrincipal" />
        <!-- Example for using X.509 in conjuntion with UsernamePassword Login Handler
             Using CN or one of the provided E-Mail addresses -->
        <FilterTemplate><![CDATA[
            #if( !$x500Principal.Empty )
                #set( $dn = $x500Principal.get(0).name.split(",") )
                #foreach( $item in $dn )
                    #if( $item.startsWith("CN=") )
                        #set( $cn = $item.substring(3) )
                    #end
                #end

                (| #foreach($mail in $x500SubjectAltNameEMailPrincipal) (mail=$mail) #end #if($cn) (cn=$cn)
#end)

            #else
                (uid=$requestContext.principalName)
            #end
        ]]></FilterTemplate>
    </resolver:DataConnector>

    ...

</AttributeResolver>
```

Logging configuration

In `$IDP_CONFIG_DIR/logging.xml`, add logging configuration for the x509 login handler:

```
<!-- ... -->
<!-- Logs X.509 LoginHandler messages -->
<logger name="ch.SWITCH.aai.idp.x509">
    <level value="WARN"/>
</logger>
<!-- ... -->
```

Deployment

Backup your IdP configuration before re-deploying the IdP web app

```
$IDP_INSTALL_DIR/install.sh
```

Bugs & Comments

Please reports bugs on the X.509 login handler issue tracker or send comments to aai@switch.ch.