SAML2 Bearer Assertion Profile for OAuth 2.0

One of the very first complements to the OAuth 2.0 specification is the SAML2 Bearer Assertion Profile for OAuth 2.0. This page talks about the following two scenarios for the SAML2.0 Assertion.

- Using SAML2.0 assertions as authorization grants
- Using SAML2.0 assertions for client authentication

Among the above two scenarios, the former is the one that is more commonly needed in enterprises. The WSO2 Identity Server supports this process.

WSO2 Identity Server as an OAuth 2.0 Authorization Server can accept SAML2 Assertions from OAuth 2.0 clients as means of resource owner authentication and authorization. Additionally, it can exchange it with OAuth 2.0 access tokens in order to access protected resources on behalf of the resource owner.

Many existing enterprises that have implemented SOA, rely on SAML. In the case of WSO2 Identity Server, SAML is used in its Web SSO feature and STS feature. Such enterprises could face a situation where they now need to consume OAuth protected resources through APIs. OAuth and OAuth 2.0 in particular are more recent specifications compared to SAML. An enterprise that has already got a working SAML2.0 based SSO infrastructure between itself and the Service Provider (SP) would prefer to use the existing trust relationship between the Identity Provider (IDP) and the Service Provider, even if the OAuth Authorization Server is entirely different from the IDP. Especially if there could be a cut down in the number of steps performed in the OAuth 2.0 dance in obtaining an access token due to the fact that the clients have already authenticated themselves to the IDP and the resource owner has authenticated himself and authorized the client, enterprises are going to love it. The SAML2 Bearer Assertion Profile for OAuth 2.0 is the answer to the question of how we leverage on the existing trust relationship between the SP and the IDP, by presenting the SAML 2.0 token to the authorization server and exchanging it directly to an OAuth 2.0 access token.

1. A user initiates logging in to an enterprise application.

2. The application being a SAML2.0 SP redirects the user to the SAML2.0 IdP for login. The user enters his credentials at the IdP and is redirected back to the SP with a SAML2.0 bearer token signed by the IdP. The SP verifies the token and logs the user to the system. The SAML2.0 token is stored saved in the user's session by the SP.

3. At some point of time the SP needs access to an OAuth2.0 protected resource. The SP, this time acting as a OAuth2.0 client, makes a request to the OAuth2.0 Authorization Server (AS) to exchange the SAML2.0 bearer token for an OAuth2.0 access token. The AS validates the assertion and returns the access token.

4. The OAuth2.0 client application sets the OAuth2.0 access token to the Authorization header of its API call to the OAuth2.0 Resource Server (RS). An entity intercepting the call to the RS would extract the token and validate it with the AS and based on the result would allow or deny access to the resource.
### Related Topics

- To try the SAML2.0 Assertion grant type with WSO2 Identity Server, see [SAML2 Bearer Assertion Profile for OAuth 2.0 with WSO2 Travelocity](#).
- For more information on how to use SAML2.0 assertions for client authentication, see [SAML 2.0 Web SSO](#).