**SAML Bearer Assertion Flow**

Microsoft Documentation suggest 4 authentication flows but also supports SAML Bearer assertion flow. This is particularly useful in scenario when a user is trying to fetch data through Graph API which only supports delegated permissions. In this case, client credentials flow which is preferred for any background processes will not work. Also, when the information is being exchanged from few different systems then also the only resort is SAML Bearer flow.

What is the challenge in getting authenticated and access data through Office 365 and Azure AD was the question when i started working on this concept.

It all started with an intent of accessing the Graph API for different workloads in office 365 viz Accessing an outlook task,planner task or for that matter read Microsoft teams related information.

There were couple of approaches to take in this scenario and no matter which approach we take,it all boils down to fetch an OAuth Token as any resource accessible through Graph API is secured with Azure AD.

Basically the need is to to have the data in the background for the user, so the user should not be prompted for the credentials during the process, Hence as a first choice, [client credentials flow](https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow) is considered.This worked for quite a few API's, i believe mostly for all of the Graph API's which supports Application permission(for e.g SendMail,Read a sharepoint list item) while your register the App. However, for API's which does not support, its a complete dead end. To my research,there were no other flows as such which can help to fetch token/data without the prompt.

Ultimately, in the end i came to know that OAuth2 Token endpoint for AAD supports SAML assertion and SAML assertion is basically in the user context which means delegated permission will work.So in a nutshell below is the scenario.

|  |
| --- |
| [https://3.bp.blogspot.com/-fgYMrIOzlec/W7jlBxPQx7I/AAAAAAAAE1o/Zov4IKjdeBgTXoXDJvfjsyQ61uxh73HLACLcBGAs/s640/Blog.png](https://3.bp.blogspot.com/-fgYMrIOzlec/W7jlBxPQx7I/AAAAAAAAE1o/Zov4IKjdeBgTXoXDJvfjsyQ61uxh73HLACLcBGAs/s1600/Blog.png) |
| High level Flow to Fetch OAuth Token based on SAML Assertion |

Now let us understand on how we can actually fetch SAML Asserstion.programatically. this approach is tested with ADFS. However this works with any IDP which is supporting return of SAML assertion programatically.

#### OAuth 2.0 SAML Bearer Assertion Flow

The OAuth 2.0 SAML bearer assertion flow defines how a SAML assertion is used to request an OAuth access token. A SAML assertion is an XML security token issued by an identity provider and consumed by a service provider. The service provider relies on its content to identify the assertion’s subject for security-related purposes.

### Pre-Requisites

1. There is a trust relationship between the authorization server/environment- O365 and the issuer of the SAML 2.0 bearer assertion, which is the identity provider - ADFS. To configure ADFS for SSO and IDP you may refer to this [article](https://blogs.technet.microsoft.com/canitpro/2015/09/11/step-by-step-setting-up-ad-fs-and-enabling-single-sign-on-to-office-365/)
2. The application is registered in the office portal with below necessary configurations.Since we are using OAuth V2.0 endpoint,so need to register the application in App registration portal. Below are the steps
   * Login to the [App registration portal](http://apps.dev.microsoft.com/) (Please note that we are using the v2.0  endpoints for Graph API and hence need to register the application in this portal. Otherwise we could have used the registrations in Azure active directory)
   * Click "Add an App"
   * Enter an "Application Name"
   * Make a note of the Application id (this is referred as client id when required by the application)
   * Click "Generate New Password" under Application secrets section. Please note that this will be flashed only once in the screen and make sure you note and save it in a safe place 😀
   * Under section "Microsoft Graph Permissions", add delegated permission for "Tasks.read" since we intend to use the Outlook graph api.
   * Click Save for saving the Application related settings and data.
3. Postman Tool is required to test the Sample requests in three parts

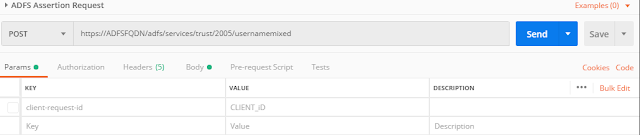
### Process to fetch the OAuth Token and Access Graph API

This can be achieved with the POSTMAN tool for POC in three parts as below. The same can be converted to code in any platform of your choice.

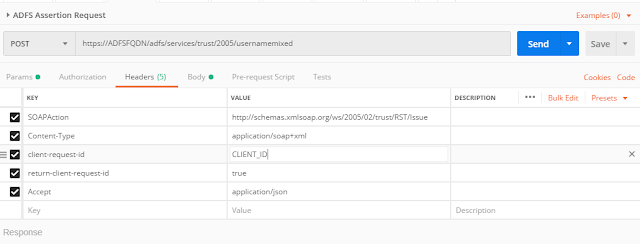
1. Get the SAML assertion from ADFS
2. Get the OAuth2 token using the assertion
3. Get the data with the Oauth token

#### Get the SAML assertion from ADFS

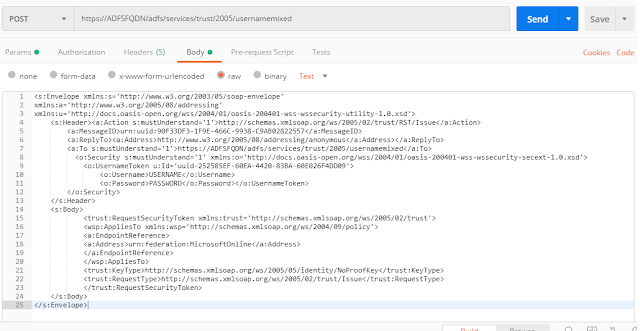
1.  Create a POST request to ADFS endpoint with SOAP envelope to fetch the SAML assertion

[](https://4.bp.blogspot.com/-k-7ZPuEQwEs/W_421HORP1I/AAAAAAAAE8Y/HaBy9o-Bg_gw92cBYGPPfIBQB8nY5BXZACLcBGAs/s1600/adfs-assertion.png)

2. Header values as below

[](https://1.bp.blogspot.com/-ejGy-B5YTRo/W_43LQSR3iI/AAAAAAAAE8g/Jfa3a3MBD4MXh3s4TpOlBcXajYf7viH0QCLcBGAs/s1600/adfs-headers.png)

3. ADFS request body

[](https://1.bp.blogspot.com/-CGA_Rt_cNT4/W_43nalNyuI/AAAAAAAAE8s/k06myogb5w84ghqTv6UJCMBd3KraqJ0fACLcBGAs/s1600/adfs-body.png)

Note : At the end of the post , i have attached the postman export files for reference

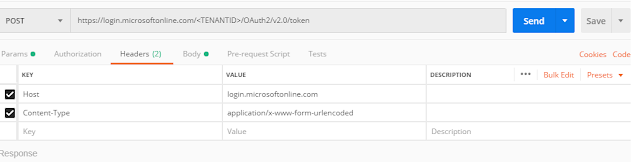
Once,this request is posted successfully, you should receive a SAML Assertion from ADFS  Only the **<SAML:Assertion>** tag data is required for further requests

Convert the same to base64 encoding to use in further requests.

#### Get the OAuth2 token using the assertion

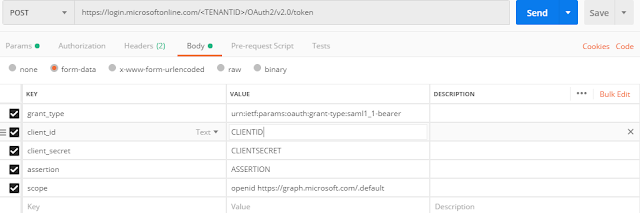
In this step ,we will fetch a OAuth2 token using the ADFS assertion response.

1. Create a POST request as shown below with the header values.

[](https://2.bp.blogspot.com/-ANy1i3HX2xY/W_472X-0NWI/AAAAAAAAE84/yxf6P1zZRyknuCETnZz7nET1sNoz6gPEgCLcBGAs/s1600/Oauth-Post.png)

2. In the Body of the request please do the following

Replace client\_id, client\_secret and assertion (base 64 encoded SAML assertion obtained in step #1)

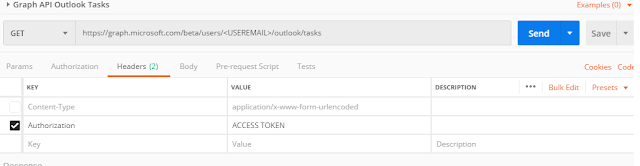
[](https://4.bp.blogspot.com/-juY5hayO44U/W_48VJKUQXI/AAAAAAAAE9A/KEN-OCwgJzAtOXDDuKmx0XDJRgWaThXBACLcBGAs/s1600/OAuth-Body.png)

3. Upon successful request you will receive a access token from Azure active directory.

#### Get the data with the Oauth token

1. For the POC we will call Graph API’s **(e.g outlook tasks in this example)**

Create a GET request as shown below with the access token fetched in the earlier step.

[](https://2.bp.blogspot.com/-y059Q0trnb0/W_49JEc9irI/AAAAAAAAE9M/h5uVab1p0Ckq8yHdUrfM_WBRnodOhK-vQCLcBGAs/s1600/Outlook-Req.png)

2. Upon successful request , you will receive a json response.

I would say, there is much more to Azure but i did not find enough documentation to this topic. So for the benefit of the community, I tried jotting down the thoughts. Hope this helps !