CRM2013, ADFS & OAuth: “Hey! Where’s the refresh token?”

by **Sonoma Partners** March 11, 2014

Today’s guest blogger is Matt Dearing, a Development Principal at Sonoma Partners.

Dynamics CRM 2013 added OAuth 2.0 support for authenticating with both the SOAP (Organization.svc/web) and ODATA endpoints. This works both online and on-premises (when using Claims Enabled IFD with ADFS*). Supporting OAuth opens the door to much better interoperability with non .NET clients integrating with CRM Dynamics 2013.

Recently, we were working on a Windows 8.1 app that integrates with CRM Dynamics 2013 and decided to try OAuth. The CRM Dynamics 2013 SDK has a very detailed section on setting up OAuth and registering your client (Walkthrough: Register a CRM app with Active Directory). We were targeting on-premises (IFD) and noticed we were not receiving refresh

tokens with the access tokens issued by ADFS. Refresh tokens are used to log a user back in, after their access token has expired, without prompting for credentials. We found out there are few other important settings that determine whether refresh tokens are issued. Running the following PowerShell commands showed the following:

> *PS C:\Windows\system32> Get-AdfsRelyingPartyTrust <insert Relying Party Trust for IFD here> | fl Name, IssueOAuthRefreshTokensTo, AlwaysRequireAuthentication, TokenLifetime*
>
> *Name : <insert Relying Party Trust for IFD here>*
>
> *IssueOAuthRefreshTokensTo : AllDevices*
>
> *AlwaysRequireAuthentication : False*
>
> *TokenLifetime : 960*
>
> *PS C:\Windows\system32> Get-AdfsProperties | fl SsoLifetime*
>
> *SsoLifetime : 960*

The settings worth noting are as follows:

IssueOAuthRefreshTokensTo
This can have one of three values

1. NoDevice = Never issue refresh tokens
2. AllDevices = Always issue refresh tokens
3. WorkplaceJoinedDevices = Only issue refresh tokens on workplace joined devices i.e. Ones that have been registered using the DRS service.

AlwaysRequireAuthentication
If this is true then the relying party will always require fresh credentials and no refresh token will be returned.

TokenLifeTime
Specifies the duration, in minutes, of the token issued by the Relying Party. This is also used for the OAuth access token's lifetime.

SSOLifetime
Specifies the duration, in minutes, of the token issued by ADFS after successful log in. Also used for the OAuth refresh token's lifetime.

If TokenLifeTime >= SSOLifetime then no refresh token will be issued as the access token would outlive it. As you can see above, both our TokenLifetime and SsoLifetime were set to

960 minutes which explained ADFS not issuing refresh tokens. After increasing the SSOLifetime, refresh tokens were returned. We could not find a way to extend a refresh token's life without increasing the SSOLifetime. It did not appear that new refresh tokens were issued by ADFS after successful refresh token logins, so the user would need to explictly log in with credentials once their refresh token is expired.

When deciding the correct duration for both TokenLifetime and SSOLifetime, it is important to note that both settings are used for more than just OAuth. They control the lifetime of your ADFS SSO token (SSOLifetime) and your relying parties token lifetime (TokenLifetime). So if you increased the SSOLifetime to a month and someone was successfully authenticated by ADFS, the token they received would be good for a month to then prove their identity to any configured relying party.

Special thanks to Travis Querec & Mahesh Hariharan (both of Microsoft) for pointing us in the right direction on these settings. Having OAuth support in CRM 2013 is very exciting and makes integrating with CRM 2013 much easier from non .NET clients.

*The newest version of ADFS that comes with Server 2012 R2. Sometimes referred to as ADFS 2.2 or ADFS 3.0