

Table of contents

1	Introduction.....	2
1.1	Executive summary.....	2
1.2	Scope	2
1.3	Definitions.....	2
2	Script to extract selected OAuth grants	3
2.1	Get-OAuth2PemrissionGrants.ps1	3
2.1.1	Description.....	3
2.1.2	Script Parameters.....	3
	Script do not require any parameters.....	3
2.1.3	Pre-requisites.....	3
2.1.4	Console Input	4
2.1.5	Input File.....	4
2.1.6	Output File	4
2.1.7	Console Output.....	4
3	Extracting Azure AD Graph, and MS Graph grants	5
3.1	Available API list in APIConfig.xml file.....	5
3.2	Prerequisite Steps	6
3.3	Extracting target OAuth grants	6

1 Introduction

1.1 Executive summary

Script presented in this documentation is used to extract Azure AD Graph and MS Graph grants.

1.2 Scope

This document contains the procedure for running dedicated script, and fetching OAuth permissions related to selected APIs only (Azure AD Graph, and Microsoft Graph).

1.3 Definitions

Azure tenant	A tenant is a representation of an organization. It's a dedicated instance of Azure Active Directory that an organization or application developer receives when the organization or application developer creates a relationship with Microsoft - like signing up for Azure, Microsoft Intune, or Microsoft 365.
OAuth	OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. OAuth provides to clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner.
Delegated Permissions	Delegated permissions are used by apps that have a signed-in user present. For these apps, either the user or an administrator consents to the permissions that the app requests, and the app is delegated permission to act as the signed-in user when making calls to the target resource. Some delegated permissions can be consented to by non-administrative users, but some higher-privileged permissions require administrator consent.
Application Permissions	Application permissions are used by apps that run without a signed-in user present - for example, apps that run as background services or daemons. Application permissions can only be consented by an administrator.

2 Script to extract selected OAuth grants

2.1 Get-OAuth2PermissionGrants.ps1

2.1.1 Description

This script is the main script to extract OAuth grants. Based on selected API list, it searches thru delegated grants, and is looking for application permissions by iterating all Service Principals in target tenant.

2.1.2 Script Parameters

Script do not require any parameters.

Parameter Name	Type	Required	Default Value	Values
N/A	N/A	N/A	N/A	N/A



2.1.3 Pre-requisites

Script requires administrator to be already authenticated with target Azure tenant, and requires at least Security Reader role enabled at PIM (Privileged Identity Management) level.

Script requires APIConfig.xml file to be present under current directory or under. This file is used to customize initial dialog with target API list.

Script requires AzureAD or AzureADPreview module installed on the system where script will be executed.

Folder view with script:

Name	Date modified	Type	Size
 APIConfig.xml	2019-02-26 1:26 PM	XML Document	4 KB
 Get-OAuth2PermissionGrants.ps1	2020-02-25 1:59 PM	Windows PowerS...	7 KB

2.1.4 Console Input

None.

2.1.5 Input File

None

2.1.6 Output File

The script generate single .CSV log file with detected OAuth grants.

File name	Content
OAuthGrants-<tenant>_<timestamp>.csv	Detected delegated, and application grants related to selected APIs.

2.1.7 Console Output

The script will provide initially target API selector (based on APIConfig.xml file), which enables administrator to select target APIs to work with, then will process data from current Azure tenant.

3 Extracting Azure AD Graph, and MS Graph grants

Note: This procedure might be executed on a regular basis to view OAuth grants from target Azure tenant.

3.1 Available API list in APIConfig.xml file.

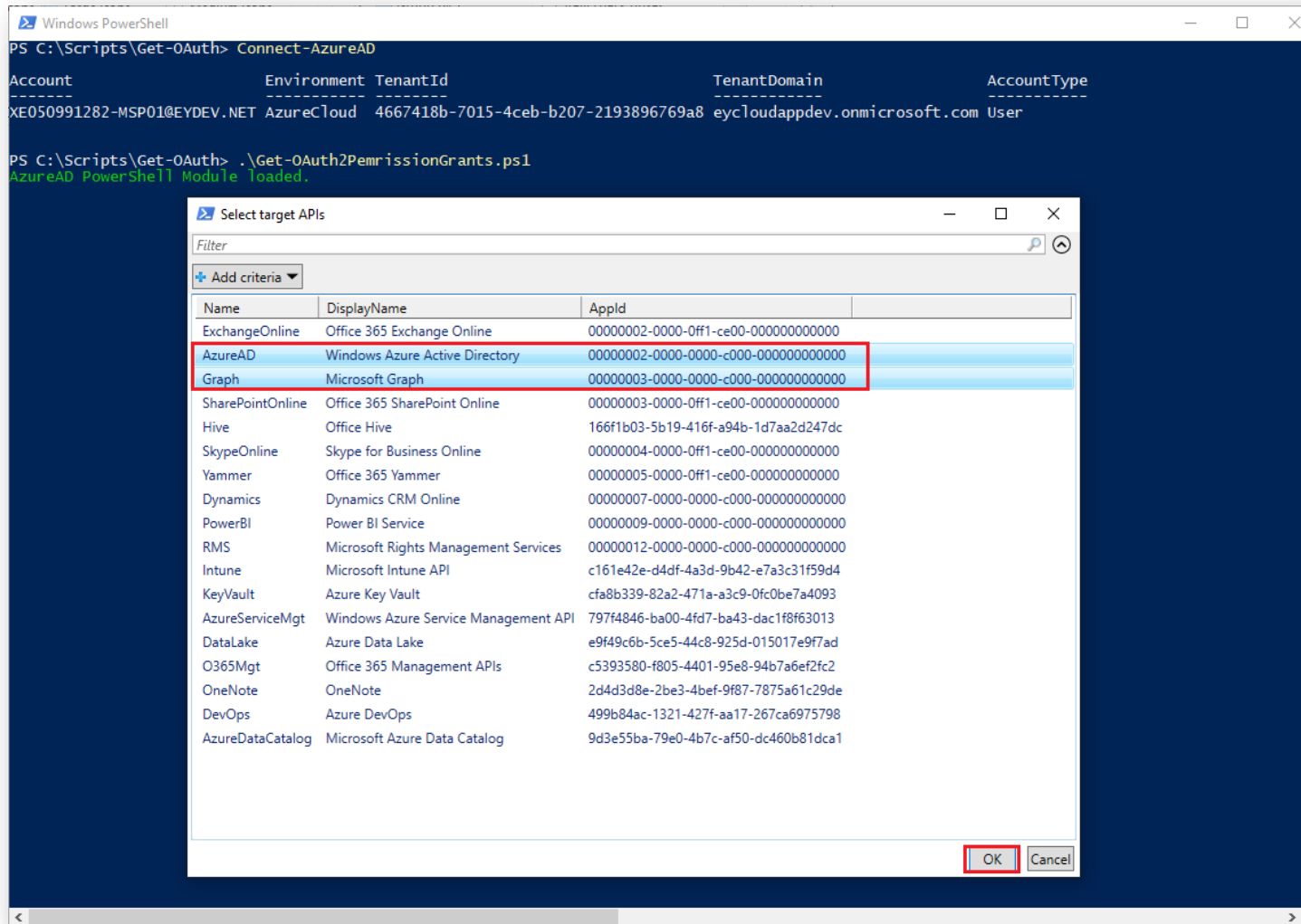
Name	DisplayName	AppId
ExchangeOnline	Office 365 Exchange Online	00000002-0000-0ff1-ce00-000000000000
AzureAD	Windows Azure Active Directory	00000002-0000-0000-c000-000000000000
Graph	Microsoft Graph	00000003-0000-0000-c000-000000000000
SharePointOnline	Office 365 SharePoint Online	00000003-0000-0ff1-ce00-000000000000
Hive	Office Hive	166f1b03-5b19-416f-a94b-1d7aa2d247dc
SkypeOnline	Skype for Business Online	00000004-0000-0ff1-ce00-000000000000
Yammer	Office 365 Yammer	00000005-0000-0ff1-ce00-000000000000
Dynamics	Dynamics CRM Online	00000007-0000-0000-c000-000000000000
PowerBI	Power BI Service	00000009-0000-0000-c000-000000000000
RMS	Microsoft Rights Management Services	00000012-0000-0000-c000-000000000000
Intune	Microsoft Intune API	c161e42e-d4df-4a3d-9b42-e7a3c31f59d4
KeyVault	Azure Key Vault	cfa8b339-82a2-471a-a3c9-0fc0be7a4093
AzureServiceMgt	Windows Azure Service Management API	797f4846-ba00-4fd7-ba43-dac1f8f63013
DataLake	Azure Data Lake	e9f49c6b-5ce5-44c8-925d-015017e9f7ad
O365Mgt	Office 365 Management APIs	c5393580-f805-4401-95e8-94b7a6ef2fc2
OneNote	OneNote	2d4d3d8e-2be3-4bef-9f87-7875a61c29de
DevOps	Azure DevOps	499b84ac-1321-427f-aa17-267ca6975798
AzureDataCatalog	Microsoft Azure Data Catalog	9d3e55ba-79e0-4b7c-af50-dc460b81dca1

3.2 Prerequisite Steps

1. *IAM Administrator* - verify if APIConfig.xml file contains expected API list.

3.3 Extracting target OAuth grants

1. *IAM Administrator* - Authenticate against target Azure tenant using *Connect-AzureAD*, providing MSP credentials.
2. *IAM Administrator* - starts **Get-OAuth2PemrissionGrants.ps1** script, selects Windows Azure Active Directory, and Microsoft Graph as target APIs (holding down Ctrl key enables multi-selection).



3. *IAM Administrator* - waits for script execution, and verify if output .CSV file is present.

```
Windows PowerShell
PS C:\Scripts\Get-OAuth> Connect-AzureAD

Gathering information about delegated grants ...
49% Complete
[ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo]

PS C:\Scripts\Get-OAuth> .\Get-OAuth2PermissionGrants.ps1
AzureAD PowerShell Module loaded.

* Processing Delegated grants ...
** Searching for delegated Windows Azure Active Directory grants ...
** Searching for delegated Microsoft Graph grants ...
*** Total number of delegated grants from all selected APIs : 1706
```



```
Windows PowerShell
PS C:\Scripts\Get-OAuth> Connect-AzureAD

Account                Environment TenantId                TenantDomain                AccountType
-----
XE050991282-MSP01@EYDEV.NET AzureCloud 4667418b-7015-4ceb-b207-2193896769a8 eycloudappdev.onmicrosoft.com User

PS C:\Scripts\Get-OAuth> .\Get-OAuth2PermissionGrants.ps1
AzureAD PowerShell Module loaded.

* Processing Delegated grants ...
** Searching for delegated Windows Azure Active Directory grants ...
** Searching for delegated Microsoft Graph grants ...
*** Total number of delegated grants from all selected APIs : 1706

* Processing Application grants ...
** Searching for all Application service principals ...
*** Total number of Application Service Principals to process : 1706

* Processing completed.

Output file : OAuthGrants_eycloudappdev.onmicrosoft.com_20200225152857.csv

PS C:\Scripts\Get-OAuth>
```