Deployment Guide

Oversharing Script

Prepared by

**Anuj Tanti**

Click here to enter text.

Table of Contents

[1 Introduction 4](#_Toc161840655)

[1.1 Oversharing script architecture 4](#_Toc161840656)

[1.2 Intended Audience 5](#_Toc161840657)

[2 Deployment Guidance 6](#_Toc161840658)

[2.1 Prerequisites 6](#_Toc161840659)

[2.1.1 Machine (If running from a local server) 6](#_Toc161840660)

[2.1.2 PowerShell 7.2 6](#_Toc161840661)

[2.1.3 PowerShell Modules 6](#_Toc161840662)

[2.1.4 Required Azure Resources (Optional) 7](#_Toc161840663)

[2.2 Configure Solution Components 7](#_Toc161840664)

[2.2.1 Authentication 7](#_Toc161840665)

[2.2.2 SharePoint Configuration When using App Secret to connect to SPO/ODFB 13](#_Toc161840666)

[2.3 Access Deployment Artifacts 13](#_Toc161840667)

[2.4 Oversharing script execution 13](#_Toc161840668)

[2.4.1 Using M365 App ID and Client Secret 14](#_Toc161840669)

[2.4.2 Using M365 App ID and certificate 15](#_Toc161840670)

[3 User and Group script execution 19](#_Toc161840671)

[3.1.1 Using M365 App ID and Client Secret 19](#_Toc161840672)

[3.1.2 Using M365 App ID and certificate 20](#_Toc161840673)

[4 FAQs 22](#_Toc161840674)

1. Introduction

Oversharing script is PowerShell based solution that allows an M365 tenant admin/ SPO Administrator to report on documents that are overshared. This solution scans SharePoint Online as well as OneDrive for Business (ODFB).

Below are the sharing scenarios that this script reports on:

|  |  |
| --- | --- |
| **Target Object** | **Scenario** |
| Site | Shared with the default SPO “Everyone”, “Everyone except external” or security group which has everyone within the organization |
| Site | Viewable by external users |
| Site | Viewable by anonymous users |
| Library | Shared with the default SPO “Everyone”, “Everyone except external” or security group which has everyone within the organization |
| Library | Viewable by external users |
| Library | Viewable by anonymous users |
| Item | Shared with the default SPO “Everyone”, “Everyone except external” or security group which has everyone within the organization |
| Item | Accessible via “Anyone” sharing link |
| Item | Viewable by external users |

This document describes the steps required to deploy and verify the installation and execution of Oversharing Script.

* 1. Oversharing script architecture



Oversharing script is a simple implementation to discover overshared information within an M365 tenant. The script uses SharePoint online native search Apis to query data from SPO/ODFB and outputs in CSVs. Since the implementation utilizes SPO Search & query-based engine:

1. Script only can process information which is made searchable within a tenant. It won’t surface information from sites for which Search has been disabled i.e.

A screenshot of a computer

Description automatically generated

Fig. Site Collection Search Setting

A screenshot of a computer

Description automatically generated

Fig. SharePoint Library Search Setting

1. Script can be customized to target different scenarios for ex: only discover information from specific sites, sites with specific metadata attached to it, specific file types & others
   1. Intended Audience

This document is intended for the team responsible for managing M365 artefacts, such as SharePoint Online (SPO) sites & One Drive for Business (ODFB). It assumes a basic working knowledge of M365, SharePoint Online, ODFB, PowerShell and PnP.PowerShell.

1. Deployment Guidance

The following sections document the steps required to deploy “Oversharing Solution”.

* 1. Prerequisites

This solution has various prerequisites that should be confirmed as ready before installation if possible. Pre-requisites are below.

* + 1. Machine (If running from a local server)

If the script is running from a server, then, a machine with the hardware and software configuration below is required:

1. **Hardware Configuration**
   1. Processor: 1 gigahertz (GHz) or faster
   2. RAM: 4 gigabytes (GB)
   3. Storage: 64 GB
2. **Software Requirements**
   1. OS: Windows 10 1607+, Windows 11, Windows Server 2016, 2019, 2022
3. Good internet connectivity
   * 1. PowerShell 7.2+

This solution uses the latest **PnP.PowerShell** PowerShell module which requires PowerShell 7.2+. More Info <https://pnp.github.io/powershell/articles/installation.html>

* + 1. PowerShell Modules

The below PowerShell modules are required:

* Install the PnP.PowerShell Module (min v2.3)

Install-Module PnP.PowerShell -MinimumVersion 2.3.0

* + 1. Required User Account

Please create a dummy user account (**No M365 license assignment required**) in Azure Entra which we will be using to run the oversharing script to scan and report on items which are accessible to everyone with organisation.

**NOTE:** Make sure this user account is not given access to any M365 resources.

* + 1. Authentication

The solutions use latest Microsoft Entra App-only authentication architecture to authenticate against the SPO/ODFB when retrieving data which is accessible to externals/anonymous users. ([More Information](https://learn.microsoft.com/en-us/sharepoint/dev/solution-guidance/security-apponly-azuread)).

**Note: If reporting on SPO resources accessible to externals/anonymous is not required then we don’t need to do below mentioned steps.**

* + - 1. Microsoft Entra App Registration & Permission configuration
* Navigate to <https://entra.microsoft.com/> and click on “App Registration”.

A screenshot of a computer

Description automatically generated

* Click “New Registration” and fill in the details and click “Register”.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* *Copy the “Application ID” as we would need this for our Oversharing Script execution.*

A screenshot of a computer

Description automatically generated

* Navigate to “API Permissions”, “Add a permission”, and then “SharePoint”

A screenshot of a computer

Description automatically generated

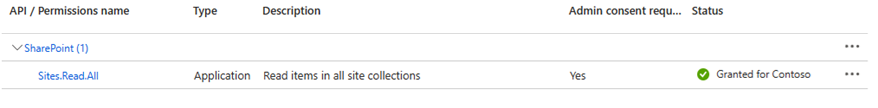
* Click on “Application permissions” à “Sites.Read.All” and click “Add permission”

A screenshot of a computer

Description automatically generated

* Click “Grant admin Consent for ……” and complete the consent process.

Your API permissions should now display as below:



* + - 1. Microsoft Entra App certificate

When wanting to use certificate-based authentication for M365 Entra App to connect to SPO/ODFB, follow steps defined in below link to generate a self-signed certificate.

[Granting access via Azure AD App-Only](https://learn.microsoft.com/en-us/sharepoint/dev/solution-guidance/security-apponly-azuread)

Once certificate is generated, upload the “**.cer**” file to the App registered in step#2.2.1.1

A screenshot of a computer

Description automatically generated

* 1. Oversharing script execution

Oversharing script has 2 operation modes which are mentioned below:

* Scan all SPO resources accessible to everyone within the Organisation
* Scan all SPO resources accessible to anonymous and external users
  + 1. Scan all SPO resources accessible to everyone within the Organisation

For this scenario, oversharing script will require below mandatory information

|  |  |
| --- | --- |
| Parameter Name | Details |
| SPTenantName | SharePoint Tenant Name. If you SPO Admin Url is <https://contoso-admin.sharepoint.com>, then tenant name is “Contoso” |
| PnPEnterpriseAppId | Provide the PnP Enterprise App ID as per the new authentication model added in PnP.PowerShell module when using interactive login ([more info](https://pnp.github.io/powershell/articles/registerapplication)) |
| ScanAccessibleToEveryone | Switch to specify scanning to be done for all data accessible to everyone within organization |
| outputreportpath | Local path on the machine where the reports will be downloaded |
| Logpath | Local path on the machine to output the script execution log |

**Example:**

.\GenerateOversharedDataReportV3.ps1 -SPTenantName "contoso" -TenantID "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"  -ScanAccessibleToEveryone -PnPEnterpriseAppId "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -outputreportpath "c:\sharingreport\" -logpath "c:\sharingreport\"

* + 1. Scan all SPO resources accessible to anonymous and external users

For this scenario, oversharing script will use M365 AppID and certificate thumbprint to connect to SPO/ODFB to discover spo resources accessible to external/anonymous users. To use certificate-based authentication to connect to SPO/ODFB, make sure the “.pfx” file generated during step#2.1.5.2 is installed on the local machine from where the oversharing script will be executed. To install the “.pfx” file, follow below steps

* Double click “.pfx” file

A screenshot of a certificate

Description automatically generated

A screenshot of a computer

Description automatically generated

* Enter the password that was set at the time of certificate generation.

A screenshot of a login box

Description automatically generated

A screenshot of a certificate

Description automatically generated

A screenshot of a certificate

Description automatically generated

For above scenario, oversharing script will require below mandatory information

|  |  |
| --- | --- |
| Parameter Name | Details |
| SPTenantName | SharePoint Tenant Name. If you SPO Admin Url is <https://contoso-admin.sharepoint.com>, then tenant name is “Contoso” |
| ClientID | M365 App Entra ID created in step#2.1.5.1 |
| Thumbprint | M365 App Certificate thumbprint created in step#2.1.5.2 |
| TenantID | M365 Tenant ID |
| outputreportpath | Local path on the machine where the reports will be downloaded |
| Logpath | Local path on the machine to output the script execution log |

**Example:**

.\GenerateOversharedDataReportV3.ps1 -SPTenantName "contoso" -TenantID "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"  -ScanAccessibleToExternalsAndAnonymous -ClientID "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" -Thumbprint "911D70A3C206A14EDC5342D9EDAJSGHEEHJ" -outputreportpath "c:\sharingreport\" -logpath "c:\sharingreport\"

1. FAQs
2. Can we run schedule this script to run using Azure Automation?

Yes, this script can be scheduled to run using Azure automation with minor tweaks.

1. Does this script output all the information into single CSV file?

No, the script is designed to generate multiple CSVs files and each CSV by default will have 10,000 rows of data (this value can be changed if number of rows needed per csv >10,000).

1. How to fast track the data discovery process using the script?

The best possible approach is to run the script in parallel targeting different areas within SPO/ODFB to speed up the overshared data discovery process. For example, one script can scan only the most sensitive sites that you may have, another can scan just ODFB and so on.

1. How to target the script to just scan specific areas within SharePoint or ODFB?

Since the oversharing script uses SharePoint KQL search queries, it gives us flexibility to tailor the query to limit scanning to designated areas within SPO. One would need KQL skills to build the query.

1. Can I extract additional metadata from the oversharing script?

Yes, if the metadata that needs to be extracted is crawled by SPO Search APIs and is available, we should be able to extract that into the output CSVs.

1. Do we know how much time the script will take to report on oversharing?

The script is highly efficient but it’s difficult to estimate how much time it will take. Since each tenant is different in terms of size, it becomes increasingly difficult to estimate on time it will take. There are other factors that affect the performance of the script i.e.

* 1. Internet connection speed
  2. Network configuration
  3. Number of metadata retrieved