|  |  |
| --- | --- |
| **Topic:** | License audit Script |

|  |  |
| --- | --- |
| **Description:** | Script that searches Active directory for end dated users and returns users with licenses. |

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Changes Made** | **By (User)** |
| 1.0 | 10/09/2024 | Initial document creation | Mathew Gallehawk |

# Requirements

* Requires Powershell graph. (will set up when script is run)
* Powershell in Admin mode.
* Admin account with authorisation.

# Process

1. Open Powershell / Termianl as administrator.
2. Navigate to this directory using cd command:  
   cd "C:\Users\e103719\G.James Australia Pty Ltd\Team ICT Service Desk - General\tools\License Audit"
3. Run the script using .\Licence\_audit.ps1
4. You may get some security prompts, select open if there is a popup or R if prompted in terminal.
5. The first time you run this you may have to install graph api which involves selecting Y repeatedly.
6. A windows login popup will appear, and authentication will be required for execution.
7. This script takes a long time, but when completed a table will populate in this folder with a list of licensed terminated users.

# Script Review

### Script Overview

1. \*\*Module Check and Installation\*\*:

- The script first checks if the Microsoft Graph module is installed. If not, it installs the required modules.

2. \*\*User License Retrieval\*\*:

- It retrieves a list of disabled users and checks their licenses to identify specific types of licenses (E5, Telsa, Project, Visio).

3. \*\*Output to CSV\*\*:

- The script outputs the list of users with specific licenses to a CSV file.

4. \*\*Main Execution\*\*:

- The main function orchestrates the execution of the above steps.

### Detailed Explanation

#### 1. Module Check and Installation

```powershell

function graphCheck {

$exists = Get-Module -ListAvailable -Name Microsoft.Graph

$loop = $true

while ($loop) {

if (!$exists) {

write-host "Graph module not installed, installing now"

Install-Module Microsoft.Graph -Scope CurrentUser -Repository PSGallery -Force

Install-Module Microsoft.Graph.Beta -Repository PSGallery -Force

$exists = Get-Module -ListAvailable -Name Microsoft.Graph

}

else {

write-host "Graph module is installed"

$loop = $false

}

}

}

```

- \*\*graphCheck\*\*: This function checks if the Microsoft Graph module is installed. If not, it installs the module and its beta version. It loops until the module is confirmed to be installed.

#### 2. User License Retrieval

```powershell

function licensedArray {

$arr = @()

$users = Get-MgUser -All -Property id, userPrincipalName, accountEnabled | Where-Object { $\_.accountEnabled -eq $false } | select-object id, userPrincipalName, accountEnabled

class user {

[string]$id

[string]$upn

[string]$enabled

[Boolean]$e5

[Boolean]$Telsa

[Boolean]$project

[Boolean]$visio

}

$users | ForEach-Object {

$licenses = Get-MgUserLicenseDetail -UserId $\_.id | select-object SkuPartNumber

$user = [user]@{

id = $\_.id

upn = $\_.userPrincipalName

enabled = $\_ | select-object -ExpandProperty accountEnabled

e5 = if ($licenses | where-object { $\_.skuPartNumber -like "\*E5\*" } ) { $true } else { $false }

Telsa = if ($licenses | where-object { $\_.skuPartNumber -like "\*MCOPSTNEAU2\*" } ) { $true } else { $false }

project = if ($licenses | where-object { $\_.skuPartNumber -like "\*PROJECT\*" } ) { $true } else { $false }

visio = if ($licenses | where-object { $\_.skuPartNumber -like "\*VISIO\*" } ) { $true } else { $false }

}

if ($user.e5 -eq $true -or $user.Telsa -eq $true -or $user.project -eq $true -or $user.visio -eq $true) {

$arr += $user

}

}

return $arr

}

```

- \*\*licensedArray\*\*: This function retrieves a list of disabled users and their licenses. It creates a custom user object for each user and checks for specific licenses (E5, Telsa, Project, Visio). Users with any of these licenses are added to an array.

#### 3. Output to CSV

```powershell

function outputarray($arr, $fileName) {

$arr | Export-Csv -Path "$psscriptroot\$filename.csv" -NoTypeInformation -Append

}

```

- \*\*outputarray\*\*: This function exports the array of users with licenses to a CSV file.

#### 4. Main Execution

```powershell

function main {

graphCheck

Connect-MgGraph -NoWelcome

$userArray = licensedArray

$licensed\_retired = Retired-with-licenses $userArray

outputarray $licensed\_retired "LicenceAudit retired users with licences"

Disconnect-MgGraph

write-host "Script complete"

start-sleep -s 3

}

main

```

- \*\*main\*\*: This function orchestrates the script execution. It checks for the required modules, connects to Microsoft Graph, retrieves the list of users with licenses, exports the data to a CSV file, and then disconnects from Microsoft Graph.

### Summary

This script automates the process of auditing licenses for disabled users in an organization by:

1. Ensuring the necessary modules are installed.

2. Retrieving and filtering users based on their license details.

3. Exporting the filtered data to a CSV file for further analysis or action.