Extract Deleted Users information using Graph API

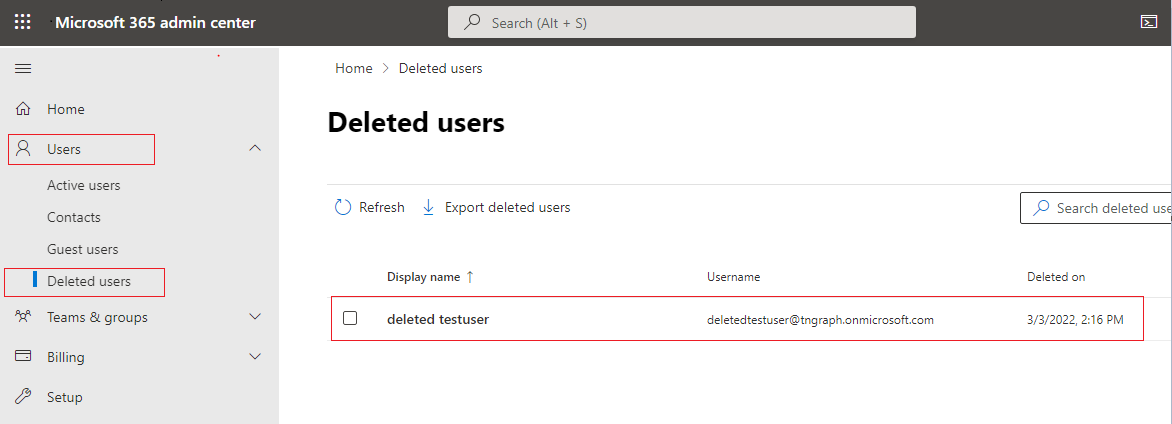
Microsoft keeps deleted users’ information (Soft delete) in the tenant for 30 days and then it will be deleted permanently (Hard Delete). So, if you have deleted a user accidentally then you can restore the user within 30 days and if you are an Microsoft365 administrator then you might need to keep a record of all deleted users (It can be used for calculating attrition rate of the organization).

To automate the report for deleted users you can use graph API in your organization, and it has several advantages as

* No need of a service account to avoid MFA.
* No need to run (manual) your script on weekends/holidays to avoid data loss.

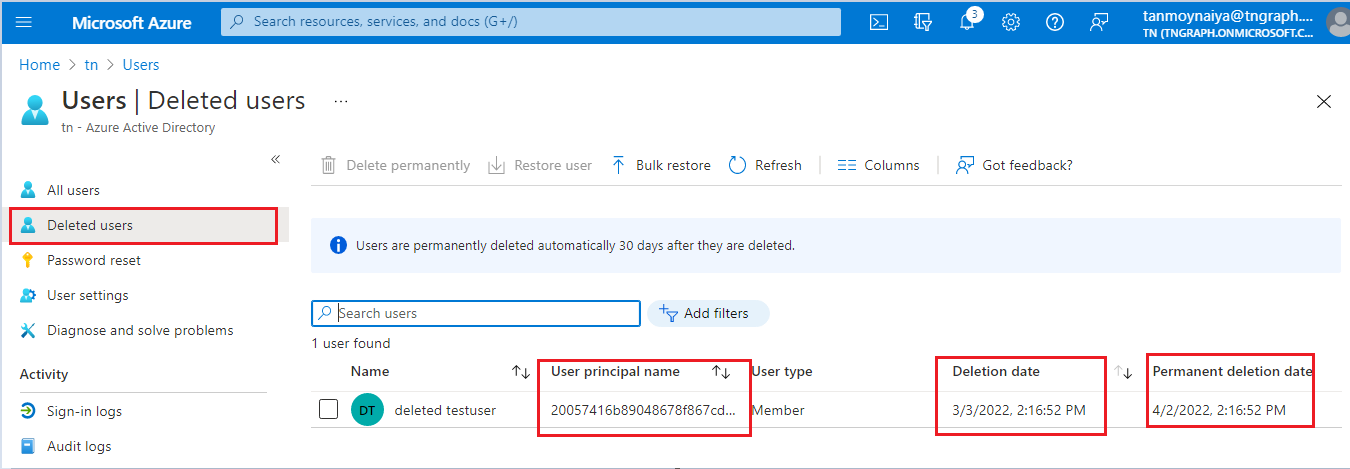
# How Deleted users looks like in the tenant

To get the list of deleted users’ information you need to login to O365 Admin portal (<https://admin.microsoft.com/Adminportal/Home?source=applauncher#/homepage>) -> Users -> Deleted Users.



Here you can clearly see the userprincipalname of the deleted user in the username field but if you do not have the admin access in the O365 portal then you have to login in Azure portal where you cannot see the userprincipalname of the deleted users.

To get the list of deleted users from azure portal, you need to login to azure (<https://portal.azure.com/#home>) -> Azure Active Directory -> Users -> Deleted Users.



Here you can see that the userprincipalname is changed (Not user-friendly manner) also there is information regarding the deletion date and the date when Microsoft will permanently delete the record and there is no way to restore the user once the record is deleted permanently from your tenant.

# getting the deleted users information using Graph API

To get the deleted user’s information through Graph API you must get below permission in your application. We will consider Application Permission for this blog.

Permissions:

| **Permission type** | **Permissions (from least to most privileged)** |
| --- | --- |
| Delegated (work or school account) | User.Read.All, User.ReadWrite.All, Directory.Read.All, Directory.ReadWrite.All, Directory.AccessAsUser.All |
| Delegated (personal Microsoft account) | Not supported. |
| Application | User.Read.All, User.ReadWrite.All, Directory.Read.All, Directory.ReadWrite.All |

## HTTP request

GET /directory/deletedItems/microsoft.graph.user

Script for getting the details as below

# Getting the token for the Application

$clientId = 'Application ID'

$tenantId = 'Tenant ID'

$clientSecret = 'Client Secret'

# Construct URI

$uri = "https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token"

# Construct Body

$body = @{

client\_id = $clientId

scope = "https://graph.microsoft.com/.default"

client\_secret = $clientSecret

grant\_type = "client\_credentials"

}

# Get OAuth Token

$tokenRequest = Invoke-WebRequest -Method Post -Uri $uri -ContentType "application/x-www-form-urlencoded" -Body $body -UseBasicParsing

# Access Token

$token = ($tokenRequest.Content | ConvertFrom-Json).access\_token

# Construct URL to get deleted users details

$url = “<https://graph.microsoft.com/beta/directory/deletedItems/microsoft.graph.user>”

$user = Invoke-RestMethod -Method GET -Uri $url -Headers @{Authorization = "Bearer $token"}

# take the result in a object to select required field

$obj = [pscustomobject]@{

UserPrincipalName = $user.value[0].userPrincipalName

UsageLocation = $user.value[0].usageLocation

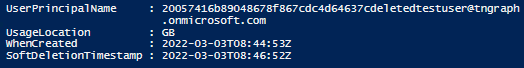
WhenCreated = $user.value[0].createdDateTime

SoftDeletionTimestamp = $user.value[0].deletedDateTime

}

$obj | fl

Output :



Looking at the above output you understand the userprincipalname has changed for the deleted user, but we need real userprincipalname of the deleted user to create a report. Luckily, we can get it from the above value.

# Get the Userprincipalname form Deleted user information

If you look closely then you will find that Microsoft simply concatenate the ObjectID of the user(When not deleted) and the previous UserPrincipalName and we all know that ObjectID is a 32 character hexadecimal number so if we simply break it from 32 character then we will get real UserPrincipalName of the deleted user as below.

PS C:\windows\system32> $user.value[0].userPrincipalName.Substring(32)

deletedtestuser@tngraph.onmicrosoft.com

# Complete Script to get all deleted users

# Getting the token for the Application

$clientId = 'Application ID'

$tenantId = 'Tenant ID'

$clientSecret = 'Client Secret'

# Construct URI

$uri = "https://login.microsoftonline.com/$tenantId/oauth2/v2.0/token"

# Construct Body

$body = @{

client\_id = $clientId

scope = "https://graph.microsoft.com/.default"

client\_secret = $clientSecret

grant\_type = "client\_credentials"

}

# Get OAuth Token

$tokenRequest = Invoke-WebRequest -Method Post -Uri $uri -ContentType "application/x-www-form-urlencoded" -Body $body -UseBasicParsing

# Access Token

$token = ($tokenRequest.Content | ConvertFrom-Json).access\_token

# Construct URL to get deleted users details

$url = <https://graph.microsoft.com/beta/directory/deletedItems/microsoft.graph.user>

$user = Invoke-RestMethod -Method GET -Uri $url -Headers @{Authorization = "Bearer $token"}

do{

$user = Invoke-RestMethod -Method GET -Uri $url -Headers @{Authorization = "Bearer $token"}

$url = $user.'@odata.nextLink'

$count = $user.value.Count

for($i=0;$i -lt $count;$i++)

{

$obj = [pscustomobject]@{

UserPrincipalName = $user.value[$i].userPrincipalName.Substring(32)

UsageLocation = $user.value[$i].usagelocation

WhenCreated = $user.value[$i].createdDateTime

SoftDeletionTimestamp = $user.value[$i].deletedDateTime

}

$obj | fl

}

}

until(!$url)

# OUTPUT :

UserPrincipalName : deletedtestuser@tngraph.onmicrosoft.com

UsageLocation : GB

WhenCreated : 2022-03-03T08:44:53Z

SoftDeletionTimestamp : 2022-03-03T08:46:52Z

PS C:\windows\system32>

TIPS : To avoid the data loss automate the above script on weekly basis (filter on date) so the deleted users information will be extracted before it got deleted permanently from the tenant.

N.B – Please go through the script carefully before running. We are not responsible for any unintentional modification in your environment.