Office 365

Introduction

- Office 365 does not expose any web interface for setting up Single Sign-On, you must issue a few Powershell commands.
- The following command require a Windows computer with Powershell ≥ 5.0 installed.

Setup Powershell environment

- Start Powershell as administrator and enter the following command:

```
Install-Module MSOnline
```

Connect to Azure AD

- In Powershell, enter the following command and enter your Office 365 administrator credentials:

```
connect-msolservice
```

Change Office federation settings

- Issue the following command to load the certificate:

```
$cert = "MIIDXXX...XXXZWCxicZzKAgV"
```

The contents of the certificat is available on the setup page of your Trustelem application

- Choose a federation brand name for your organization, for instance:

```
$FederationBrandName = "mydomain.com"
```

- Execute the following commands (adapt the **DomainName**, the **URLs** and keep the backquotes characters `):

```
Set-MsolDomainAuthentication -DomainName mydomain.com -Authentication managed
Set-MsolDomainAuthentication       -DomainName mydomain.com `
-FederationBrandName             $FederationBrandName `
```

```
-Authentication                  Federated `
-PassiveLogOnUri                 https://mydomain.trustelem.com/app/34XXX/sso `
-SigningCertificate              $cert `
-IssuerUri                       https://mydomain.trustelem.com/app/34XXX/mydomain.com `
-LogOffUri                       https://mydomain.trustelem.com/app/34XXX/slo `
-PreferredAuthenticationProtocol SAMLP
```

Note for Azure AD users

⚠Using an external IdP like Trustelem (via SAML) to federate Azure AD / Office 365 for users that exist only in the cloud leads to several critical issues and is strongly discouraged:

- **Azure passwords no longer work — authentication is fully offloaded to the IdP.**
- **Users can't be created directly with federated domains — PowerShell is required.**
- **Each user needs a manually set onPremisesImmutableId via PowerShell.**
- **No automated provisioning, and more complex support.**

The consequences are the following:

- After this setup, your Azure users will not have the possibility to use their Azure AD password anymore : they have to use a Trustelem password instead.
  Go to your Azure AD directory on Trustelem > Enabled **Use Trustelem as password source**

- If the synchronized users only exist on an Azure which is not linked to an AD, then you'll need to verify if they have an **onPremisesImmutableId**. You also need to add this attribute to Trustelem:
  Go to your Azure AD directory on Trustelem > tick **Advanced options** > enter **onPremisesImmutableId** in **Custom attributes**

Powershell script example to add onPremisesImmutableId to existing users:

```
# Install the Microsoft Graph PowerShell module
Install-Module Microsoft.Graph -Scope CurrentUser

# Connect to Microsoft Graph with the necessary scopes
Connect-MgGraph -Scopes "User.ReadWrite.All", "Directory.AccessAsUser.All"

# Replace with your temporary fallback domain (e.g., yourdomain.onmicrosoft.com)
$tmpUPN = "yourdomain.onmicrosoft.com"
```

```
# Retrieve all users who don't have an OnPremisesImmutableId set
$users = Get-MgUser -All | Where-Object { -not $_.OnPremisesImmutableId }

foreach ($user in $users) {
    $currentUPN = $user.UserPrincipalName
    $initialDomain = $currentUPN.Split("@")[1]
    $newUPN = $currentUPN.Replace("@$initialDomain", "@$tmpUPN")

    # Temporarily change UPN to a domain that allows ImmutableId update
    Update-MgUser -UserId $user.Id -UserPrincipalName $newUPN

    # Generate a new unique ImmutableId (Base64-encoded GUID)
    $newImmutableId = [System.Convert]::ToBase64String([Guid]::NewGuid().ToByteArray())

    # Assign the ImmutableId to the user
    Update-MgUser -UserId $user.Id -OnPremisesImmutableId $newImmutableId

    # Revert UPN back to the original domain
    Update-MgUser -UserId $user.Id -UserPrincipalName $currentUPN
}

# List users who still don't have an ImmutableId (if any)
$usersWithoutImmutableId = Get-MgUser -All | Where-Object { -not $_.OnPremisesImmutableId } |
Select-Object UserPrincipalName

Write-Output "Users without OnPremisesImmutableId:"
$usersWithoutImmutableId.UserPrincipalName
```

Powershell script example to create a new user with onPremisesImmutableId:

```
Connect-MgGraph -Scopes "User.ReadWrite.All", "Directory.AccessAsUser.All"

# Password profile as plain hashtable
$passwordProfile = @{
    Password = "TemporaryPassword123!"
    ForceChangePasswordNextSignIn = $true
}

# Build full user creation parameters in a hashtable (sûr et lisible)
$params = @{
```

```
    DisplayName      = "Peter Doe"
    GivenName        = "Peter"
    Surname          = "Doe"
    UserPrincipalName = "peter.doe@your_domain.onmicrosoft.com"
    MailNickname     = "peterdoe"
    PasswordProfile  = $passwordProfile
    AccountEnabled   = $true
}

# Create user
$newUser = New-MgUser @params

# If created, assign ImmutableId and switch UPN
if ($newUser -and $newUser.Id) {
    $immutableId = [System.Convert]::ToBase64String([Guid]::NewGuid().ToByteArray())
    Update-MgUser -UserId $newUser.Id -OnPremisesImmutableId $immutableId
    Update-MgUser -UserId $newUser.Id -UserPrincipalName "peter.doe@your_federated_domain.fr"
} else {
    Write-Error "User creation failed. Aborting further operations."
}

Disconnect-MgGraph
```

Revision #6
Created 1 July 2022 09:03:03 by WALLIX Admin
Updated 1 April 2025 09:59:26 by WALLIX Admin