Lycée pergaud SIO2 SISR, Mr. David

Mission 1 Bis, Powershell/GPO

FICHOT. Benjamin
19/10/2023

Sommaire

1. Création de nouveaux compte utilisateurs dans un services qualité

Nous allons créer un script en powershell, contrairement à la toute première mission qui était en batch.

Voici le script:

```
Install-Module NTFSSecurity

$usersQualite = Import-Csv "liste_utilisateurs.csv" -Delimiter ","
$groupesQualite = $usersQualite.groupe | sort | get-unique

New-Item -ItemType Directory -Path C:\dossier_utilisateur\qualite\
New-ADOrganizationalUnit -Name "Qualite" -Path "DC=gsb15,DC=local"

# Création des groupes
foreach ($groupe in $groupesQualite) {
        if(Get-ADGroup -Filter "SamAccountName -eq '$groupe'") {
        Write-Warning "Le groupe $groupe existe deja !"
    } else {
        New-ADGroup -Name $groupe -Path "ou=Qualite,dc=gsb15,dc=local" -GroupScope Global
    }
}

if(Get-ADGroup -Filter "SamAccountName -eq 'qualite'") {
    Write-Warning "Le groupe qualite existe deja !"
} else {
    New-ADGroup -Name qualite -Path "ou=Qualite,dc=gsb15,dc=local" -GroupScope Global
}


# Création utilisateur + attribution des groupes + paramétrage expiration de compte au bout de 6 mois
foreach ($User in $usersQualite) {
        $firstname = $user.Prenom
        $lastname = $user.Nom
        $username = "$firstname" + "_$lastname"
        $groupe = $user.Groupe
        $samid = $lastname.ToLower()
        $tmp = $firstname.SubString(0,1).ToLower()
        $samid = "$samid" + "$tmp"

    if(Get-ADUser -Filter "SamAccountName -eq '$samid'") {
        Write-Warning "L'utilisateur $samid existe deja !"
    } else {
        New-ADUser -Name "$firstname $lastname" `
        -DisplayName "$firstname $lastname" `
        -GivenName $firstname `
        -Surname $lastname `
        -SamAccountName $samid `
        -UserPrincipalName "$samid@$((Get-ADDomain).DNSRoot)" `
        -Description "Dans le groupe $groupe" `
        -Path "ou=Qualite,dc=gsb15,dc=local" `
        -AccountPassword (ConvertTo-SecureString "J'aime..." -AsPlainText -Force) `
        -ChangePasswordAtLogon $false `
        -Enabled $true `
        -HomeDrive A: `
        -HomeDirectory "\\BF-LABANNU\qualite\$groupe\$samid"

        Add-ADGroupMember $groupe $samid
```

```
        Add-ADGroupMember qualite $samid
        Set-ADAccountExpiration -Identity $samid -DateTime (Get-Date).AddMonths(6)
    }
}



if (Get-SmbShare -Name "qualite") {
        Write-Warning 'Le partage "qualite" existe deja !'
} else {
    $ParametresSMB = @{
        Name = "qualite"
        Path = "C:\dossier_utilisateur\qualite\"
        FullAccess = "gsb15.local\Administrateur"
        ChangeAccess = "Utilisateurs"
    }
    New-SmbShare @ParametresSMB
}



# PERMISSION
foreach ($User in $usersQualite) {
        $firstname = $user.Prenom
        $lastname = $user.Nom
        $username = "$firstname" + "_$lastname"
        $groupe = $user.Groupe
        $samid = $lastname.ToLower()
        $tmp = $firstname.SubString(0,1).ToLower()
        $samid = "$samid" + "$tmp"

    # Création du dossier personnel
    New-Item -ItemType Directory -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"

    # Desactiver l'héritage tout en copiant les autorisations NTFS héritées
    Get-Item "C:\dossier_utilisateur\qualite\$groupe\$samid" | Disable-NTFSAccessInheritance

    # Ajout des autorisations NTFS
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
"$samid@gsb15.local" -AccessRights FullControl
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
"$samid@gsb15.local" -AccessRights Delete -AccessType Deny
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
"$samid@gsb15.local" -AccessRights ChangePermissions -AccessType Deny
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account "Administrateurs"
-AccessRights FullControl

    # Modifier le proprietaire sur le dossier
    Set-NTFSOwner -Path "C:\dossier_utilisateur\qualite\$groupe\$samid" -Account "$samid@gsb15.local"

    # Supprimer des autorisations NTFS
    Remove-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account "Utilisateurs"
-AccessRights FullControl

} # foreach ($User in $usersQualite)



# Dossier partagé/Permission
foreach ($groupe in $groupesQualite) {

    # Création du dossier personnel
    New-Item -ItemType Directory -Path "C:\dossier_utilisateur\qualite\$groupe\partage"
```

```
    # Désactiver l'héritage tout en copiant les autorisations NTFS héritées
    Get-Item "C:\dossier_utilisateur\qualite\$groupe\partage" | Disable-NTFSAccessInheritance

    # Ajout des autorisations NTFS pour tous les groupes
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
"qualite@gsb15.local" -AccessRights ReadData
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
"qualite@gsb15.local" -AccessRights Delete -AccessType Deny

    # Ajout des autorisations NTFS pour le groupe concernant le partage
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
"$groupe@gsb15.local" -AccessRights FullControl
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
"$groupe@gsb15.local" -AccessRights Delete -AccessType Deny
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
"$groupe@gsb15.local" -AccessRights ChangePermissions -AccessType Deny
    Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account "Administrateurs"
-AccessRights FullControl


    # Modifier le proprietaire sur le dossier
    Set-NTFSOwner -Path "C:\dossier_utilisateur\qualite\$groupe\partage" -Account "Administrateurs"

    # Supprimer des autorisations NTFS
    Remove-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account "Utilisateurs"
-AccessRights FullControl
}
```

Oula, c'est vraiment pas cool à lire… On va voir ce que ce beau monde veut bien dire…

1. Fichier csv avec la liste des utilisateurs

On va commencer simple, avec:

```
1.  $usersQualite = Import-Csv "liste_utilisateurs.csv" -Delimiter ","
2.  $groupesQualite = $usersQualite.groupe | sort | get-unique
```

En 1 importer le fichier csv dans un tableau $userQualite délimiter par une , ca nous donnera une liste de chaque utilisateur avec son nom/son prénom et sont groupe
En 2 ne prend que les groupes qu'il faudra créer dans une liste, ça sera plus simple pour la création des groupes automatisé dans faire de calculs inutile au script

2. Création de l'OU qualité et du dossier qualité

```
1.  New-Item -ItemType Directory -Path C:\dossier_utilisateur\qualite\
2.  New-ADOrganizationalUnit -Name "Qualite" -Path "DC=gsb15,DC=local"
```

On crée d'abord le dossier qualite sur notre serveur puis on crée notre nouvelle Unité d'organisation dans gsb15.local
Nous verrons pour les permissions plus tard.

3. Création des groupes dans l'OU qualite

```
1.  foreach ($groupe in $groupesQualite) {
2.      if(Get-ADGroup -Filter "SamAccountName -eq '$groupe'") {
3.          Write-Warning "Le groupe $groupe existe deja !"
4.      } else {
5.          New-ADGroup -Name $groupe -Path "ou=Qualite,dc=gsb15,dc=local" -GroupScope Global
6.      }
7.  }
8.
9.  if(Get-ADGroup -Filter "SamAccountName -eq 'qualite'") {
10.     Write-Warning "Le groupe qualite existe deja !"
11. } else {
12.     New-ADGroup -Name qualite -Path "ou=Qualite,dc=gsb15,dc=local" -GroupScope Global
13. }
```

On créer une boucle pour vérifier si les comptes ne sont toujours pas créés, si il ne le sont pas alors on les crée. Bien sûr si il existe déjà on prévient l'utilisateur avec Write-Warning

4. Création des utilisateurs et attribution de leurs droit

```
1.  foreach ($User in $usersQualite) {
2.          $firstname = $user.Prenom
3.          $lastname = $user.Nom
4.          $username = "$firstname" + "_$lastname"
5.          $groupe = $user.Groupe
6.          $samid = $lastname.ToLower()
7.          $tmp = $firstname.SubString(0,1).ToLower()
8.          $samid = "$samid" + "$tmp"

9.      if(Get-ADUser -Filter "SamAccountName -eq '$samid'") {
10.         Write-Warning "L'utilisateur $samid existe deja !"
11.     } else {
12.         New-ADUser -Name "$firstname $lastname" `
13.         -DisplayName "$firstname $lastname" `
14.         -GivenName $firstname `
15.         -Surname $lastname `
16.         -SamAccountName $samid `
17.         -UserPrincipalName "$samid@$((Get-ADDomain).DNSRoot)" `
18.         -Description "Dans le groupe $groupe" `
19.         -Path "ou=Qualite,dc=gsb15,dc=local" `
20.         -AccountPassword (ConvertTo-SecureString "J'aime..." -AsPlainText -Force) `
21.         -ChangePasswordAtLogon $false `
22.         -Enabled $true `
23.         -HomeDrive A: `
24.         -HomeDirectory "\\BF-LABANNU\qualite\$groupe\$samid"

25.         Add-ADGroupMember $groupe $samid
26.         Add-ADGroupMember qualite $samid
27.         Set-ADAccountExpiration -Identity $samid -DateTime (Get-Date).AddMonths(6)
28.     }
29. }
```

Nouvelle boucle ce coup-ci pour créer les utilisateurs, nous allons déjà stocker leur nom, prénom et groupe, puis nous allons générer leur samid et leur username dans des variables avec la ligne 4, 6, 7 et 8.
La variable $samid stock le nom puis la première lettre du prénom.
La variable $username stock sont prénom et nom séparer d'un "_".

Maintenant nous allons créer les utilisateurs avec la commande New-ADUser:

- **-Name**(12) Son nom d'objet dans l'AD
- **-DisplayName**(13) Son nom d'affichage
- **-GivenName**(14) Son prénom,
- **-Surname**(15) Son nom,
- **-SamAccountName**(16) permet de lui donner un identifiant de connexion,
- **-UserPrincipalName**(17) Donne un nom principal à l'utilisateur(car pourquoi pas ?)
- **-Description**(18) Donne une description à l'utilisateur
- **-Path**(19) son chemin sur l'AD
- **-AccountPassword**(20) permet de définir un mot de passe pour la première connexion,
- **-ChangePasswordAtLogon**(21) permet de forcer la création d'un mot de passe personnalisé à l'utilisateur,
- **-Enabled**(21) permet d'activer ou non le compte,
- **-HomeDrive**(23) défini la lettre attribuée au répertoire utilisateur
- **-HomeDirectory**(24) chemin du répertoire utilisateur,

Puis nous allons définir leurs groupes et leur date d'expiration, chaque compte doit expirer après 6 mois avec ses 3 lignes:

```
1. Add-ADGroupMember $groupe $samid
2. Add-ADGroupMember qualite $samid
3. Set-ADAccountExpiration -Identity $samid -DateTime (Get-Date).AddMonths(6)
```

En 1 et 2 on ajoute l'utilisateur à son groupe et au groupe qualite.
Et en 3 on définit l'expiration du compte à 6 mois.

5. Création du partage qualité

Celui ci rapide, on créer un partage smb qualite:

```
1.  if (Get-SmbShare -Name "qualite") {
2.          Write-Warning 'Le partage "qualite" existe deja !'
3.  } else {
4.      $ParametresSMB = @{
5.          Name = "qualite"
6.          Path = "C:\dossier_utilisateur\qualite\"
7.          FullAccess = "gsb15.local\Administrateur"
8.          ChangeAccess = "Utilisateurs"
9.      }
10.     New-SmbShare @ParametresSMB
    }
```

Bien sûr on donne toutes les permission à Administrateur et on n'autorise que des modification/ajout et suppression pour utilisateurs, on affinera leur permission plus tard.

6. Création du dossier utilisateur et paramétrage des permissions

Bon, voici probablement la partie la plus longue… (la 7 aussi est longue)

```
1.    foreach ($User in $usersQualite) {
2.            $firstname = $user.Prenom
3.            $lastname = $user.Nom
4.            $username = "$firstname" + "_$lastname"
5.            $groupe = $user.Groupe
6.            $samid = $lastname.ToLower()
7.            $tmp = $firstname.SubString(0,1).ToLower()
8.            $samid = "$samid" + "$tmp"

9.        # Création du dossier personnel
10.       New-Item -ItemType Directory -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"

11.       # Désactiver l'héritage tout en copiant les autorisations NTFS héritées
12.       Get-Item "C:\dossier_utilisateur\qualite\$groupe\$samid" | Disable-NTFSAccessInheritance

13.       # Ajout des autorisations NTFS
14.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
      "$samid@gsb15.local" -AccessRights FullControl
15.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
      "$samid@gsb15.local" -AccessRights Delete -AccessType Deny
16.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
      "$samid@gsb15.local" -AccessRights ChangePermissions -AccessType Deny
17.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
      "Administrateurs" -AccessRights FullControl

18.       # Modifier le proprietaire sur le dossier
19.       Set-NTFSOwner -Path "C:\dossier_utilisateur\qualite\$groupe\$samid" -Account
      "$samid@gsb15.local"

20.       # Supprimer des autorisations NTFS
21.       Remove-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\$samid"  -Account
      "Utilisateurs" -AccessRights FullControl

22.   } # foreach ($User in $usersQualite)
```

Donc de 1 à 8, on récupère les mêmes informations que pour la boucle qui créer les utilisateurs, ensuite en 10 on créer le dossier utilisateur dans le dossier de son groupe.

En 12 on désactive l'héritage des permissions tout en copiant les autorisations NTFS déjà héritées;
De 14 à 21 on définit les permissions avec la commande Add-NTFSAccess, Set-NTFSOwner et Remove-NTFSAccess, ces commande sont disponible grace à l'import du module NTFSSecurity que l'on installe grâce à la commande `Install-Module NTFSSecurity` commande que l'on retrouve en première ligne du fichier.

En tout, on autorise l'utilisateur à créer, modifier et supprimer des fichiers dans son répertoire personnel, on ajoute le contrôle totale à administrateur, on définit l'utilisateur comme étant propriétaire de son répertoire personnel et on enlève les accès à tout autre utilisateurs.

7. Création dossier partagé et paramétrage des permissions

Bon, voici probablement la partie la plus longue…

```
1.    foreach ($groupe in $groupesQualite) {
2.
3.        # Création du dossier personnel
4.        New-Item -ItemType Directory -Path "C:\dossier_utilisateur\qualite\$groupe\partage"
5.
6.        # Désactiver l'héritage tout en copiant les autorisations NTFS héritées
7.        Get-Item "C:\dossier_utilisateur\qualite\$groupe\partage" | Disable-NTFSAccessInheritance

8.        # Ajout des autorisations NTFS pour tous les groupes
9.        Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
      "qualite@gsb15.local" -AccessRights ReadData
10.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
      "qualite@gsb15.local" -AccessRights Delete -AccessType Deny

11.       # Ajout des autorisations NTFS pour le groupe concernant le partage
12.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
      "$groupe@gsb15.local" -AccessRights FullControl
13.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
      "$groupe@gsb15.local" -AccessRights Delete -AccessType Deny
14.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
      "$groupe@gsb15.local" -AccessRights ChangePermissions -AccessType Deny
16.       Add-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
      "Administrateurs" -AccessRights FullControl

17.
18.       # Modifier le proprietaire sur le dossier
          Set-NTFSOwner -Path "C:\dossier_utilisateur\qualite\$groupe\partage" -Account
      "Administrateurs"
19.
20.       # Supprimer des autorisations NTFS
          Remove-NTFSAccess -Path "C:\dossier_utilisateur\qualite\$groupe\partage"  -Account
21.   "Utilisateurs" -AccessRights FullControl
      }
```

Bien sûr on donn

8. Le script de démarrage de session

```
net use V: \\BF-LABANNU\equipe_normes\partage /PERSISTENT:YES
net use W: \\BF-LABANNU\equipe_procedures\partage /PERSISTENT:YES
net use X: \\BF-LABANNU\equipe_qualite-de-la-documentation\partage /PERSISTENT:YES
net use Y: \\BF-LABANNU\equipe_qualite-des-produits\partage /PERSISTENT:YES
net use Z: \\BF-LABANNU\equipe_qualite-des-services\partage /PERSISTENT:YES
```

Pour chaque dossier de partage de chaque groupe, on l'ajoute avec la commande net use en précisant la lettre que l'on va lui attribuer et sont chemins réseau, l'option **PERSISTENT:YES** nous permet de garder le lecteur réseau après redémarrage de la session.