SCRIPT MOT DE PASSE

Contexte : instaurer une rotation automatique des mots de passe admin (locaux et du domaine) tous les 7 jours.

Le mot de passe doit contenir 25 caractères et être généré de façon aléatoire.

Il faut pouvoir récupérer les nouveaux mots de passe par mail ou console.

Pré-requis :

- PowerShell
- Mot de passe d'application Gmail

Mot de passe d'application Gmail

Pour obtenir un mot de passe d'application, il faut d'abord activer la double validation dans les paramètres du compte Gmail. Ensuite l'option apparaîtra.

> Aller sur votre compte Gmail (myaccount) > Sécurité et connexion > Activer la « double validation »

Récupération par mail

```
Import-Module ActiveDirectory

# ===== PARAMÈTRES =====
$User = "CN=Adminstrateur, CN=Users, DC=mathilde, DC=local"   # ou possible « idadminlocal », ex : rachel
$PasswordLength = 25
$Chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_=+"      #caractères modifiables

# Mail
$SmtpServer = "smtp.gmail.com"
$port      = 587
$From      = "nyaimathilde@gmail.com"
```

```
$To       = "nyaimathilde8@gmail.com"
$Subject    = "Rotation mot de passe Admin AD"



# Génération du mot de passe
$Password = -join (1..$PasswordLength | ForEach-Object {
   $Chars[(Get-Random -Minimum 0 -Maximum $Chars.Length)]
})

# Conversion sécurisée
$SecurePassword = ConvertTo-SecureString $Password -AsPlainText -Force

# Changement du mot de passe AD
Set-ADAccountPassword -Identity $User -Reset -NewPassword $SecurePassword

# Forcer la réplication AD (optionnel mais recommandé)
#repadmin /syncall /AdeP | Out-Null

# ===== Authentification Gmail =====
# IMPORTANT : Utiliser un mot de passe d'application Gmail
$Credential = Get-Credential



# Corps du mail
$Body = @"
Bonjour,

Le mot de passe du compte AD suivant a été modifié automatiquement :

Compte : $User
Date    : $(Get-Date -Format "yyyy-MM-dd HH:mm:ss")

NOUVEAU MOT DE PASSE :
$Password
```

Merci de le stocker dans un coffre-fort sécurisé.

-- Script automatique

"@

Envoi du mail

Send-MailMessage `

-SmtpServer $SmtpServer `

-Port $Port `

-UseSsl `

-Credential $Credential `

-From $From `

-To $To `

-Subject $Subject `

-Body $Body `

-Encoding UTF8

Récupération en console

Compte admin local

Import-Module ActiveDirectory

===== PARAMÈTRES =====

$User = "rachel"

$PasswordLength = 25

$Chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_=+".ToCharArray()

======================

try {

Génération du mot de passe

```
$Password = -join (1..$PasswordLength | ForEach-Object {
$Chars | Get-Random
    })

    # Conversion en SecureString
$SecurePassword = ConvertTo-SecureString $Password -AsPlainText -Force

# Changement du mot de passe AD
Set-ADAccountPassword -Identity $User -Reset -NewPassword $SecurePassword

# Affichage console
Write-Host "Mot de passe modifié avec succès" -ForegroundColor Cyan
Write-Host "Compte : $User"
Write-Host "Date   : $(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')"
Write-Host ""
Write-Host "NOUVEAU MOT DE PASSE :"
Write-Host $Password -ForegroundColor Cyan

 }
catch {
Write-Host "ERREUR lors du changement du mot de passe !" -ForegroundColor Red
Write-Host $_.Exception.Message
 }
```

Admin du domaine

```
Import-Module ActiveDirectory
# ===== PARAMÈTRES =====
$User = "CN=Administrateur, CN=Users, DC=mathilde, DC=local"
$PasswordLength = 25  #longueur modifiable
$Chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$
%^&*()-_=+".ToCharArray()   #caractères modifiables
# ======================
```

```
try {
# Génération du mot de passe
$Password = -join (1..$PasswordLength | ForEach-Object {
$Chars | Get-Random
    })

    # Conversion en SecureString
$SecurePassword = ConvertTo-SecureString $Password -AsPlainText -Force

# Changement du mot de passe AD
Set-ADAccountPassword -Identity $User -Reset -NewPassword $SecurePassword

# Affichage console
Write-Host "Mot de passe modifié avec succès" -ForegroundColor Cyan
Write-Host "Compte : $User"
Write-Host "Date   : $(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')"
Write-Host ""
Write-Host "NOUVEAU MOT DE PASSE :"
Write-Host $Password -ForegroundColor Cyan

 }
catch {
Write-Host "ERREUR lors du changement du mot de passe !" -ForegroundColor Red
Write-Host $_.Exception.Message
 }
```

→ Stocker les scripts dans un dossier à la racine. Sinon la racine n'exécutera rien.

Rotation automatique

Planification de la tâche

Au lancement du script via Powershell, les id et le mot de passe d'application Gmail sont demandés = mode interactif.

Lors du lancement de tâche et donc du script, la tâche échouera car l'id et le mot de passe d'application ne seront pas rentrés. La tâche est lancée en mode non-intéractif.

Sur le serveur SMTP, en PowerShell, taper :

Get-Credential | Export-Clixml "C:\Secure\smtp_cred.xml".

Ce dossier servira à stocker le mot de passe (ici en hash)

Modification du script :

A la place de : $Credential = Get-Credential, mettre > **$Credential = Import-Clixml « C:\secure\smtp_cred.xml »**

Sur le serveur SMTP > gestionnaires de tâches > planifier une tâche > nommer la tâche

Cocher dans l'onglet « général » :

- Exécuter même si l'utilisateur n'est pas connecté
- Exécuter avec les autorités maximales
- Configuré pour : « Windows Server 2022 »

Dans l'onglet « Déclencheurs » > nouveau > mettre les horaires/créneaux souhaités

Dans l'onglet « Actions » > nouveau

→ action = « démarrer un programme »

→ paramètres > programme/script :

« C:\Windows\Sustem32\WindowsPowerShell\v1.0\powershell.exe »

/

« C:\Windows\SYSTEM32\cmd.exe »

/

« C:\Users\Administrateur.MATHILDE\Desktop\cmd.bat » → celui là ok

Le dossier cmd.bat contient (le créer en amont dans le bureau): **cd « C:\script\ » & start cmd.lnk**

(facultatif > arguments : mettre le chemin complet du script)