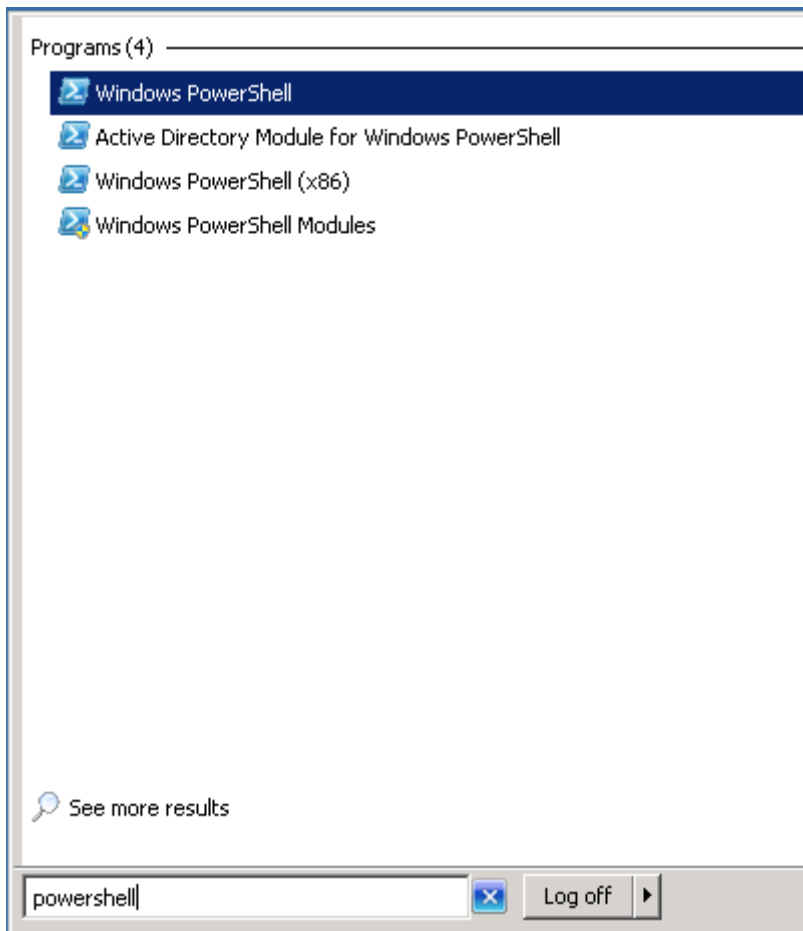


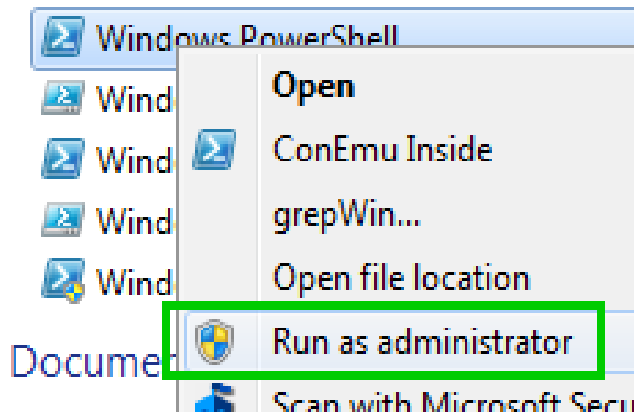
Sauvegarder l'exel

Ensuite avant de commencer a créer le script nous devons installer Powershell ISE



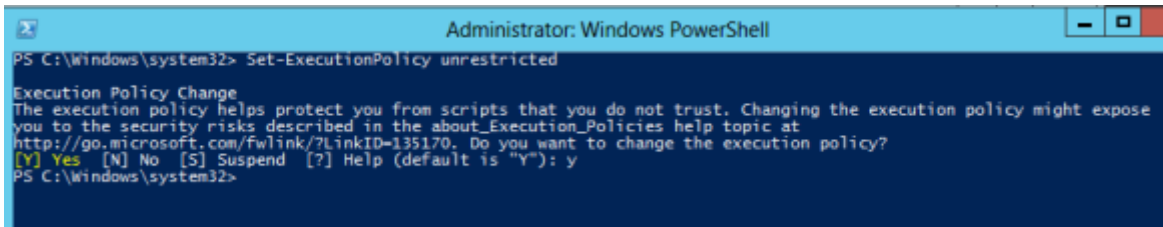
Executer cette commande en lançant windows powershell en mode administrateur

Programs (5)



Changer les droits d'exécution en tapant

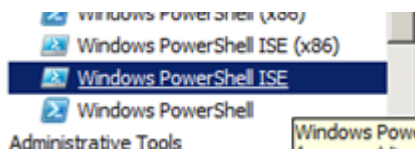
- 1) Get-ExecutionPolicy
- 2) Set-ExecutionPolicy RemoteSigned (au lieu de unrestricted)



Ensuite nous pouvons installer le fameux powershell ISE c'est un éditeur de script powershell

```
Import-Module ServerManager
Add-WindowsFeature PowerShell-ISE
```

Une fois installé exécuter powershell ISE et nous allons commencer a éditer le futur script qui nous permettra d'ajouter les Users sur la base de donnée de l'Active directory allons y :



```

Import-Module ActiveDirectory

$users = Import-Csv -Delimiter "/" -Path
"C:\testpowershell\Fichier_Excel_TP.csv"

foreach ($User in $Users)
{
    New-ADUser -SamAccountName ($User.Nom) -Name ($User.Login) -GivenName
($User.Prenom) -Surname ($User.Nom) -UserPrincipalName ($User.Login) -path
($User.Chemin) -accountpassword (ConvertTo-SecureString $user.Motdepasse -
AsPlainText -Force) -Enabled $true
}

```

MACRO EXEL

```

' Déclaration des variables obligatoire '
Option Explicit

' -----
' ----- '

' Fonction personnalisée permettant de générer des mots de passe contenant
des lettres et, en option, chiffres et symboles. '

' Tous les arguments sont facultatifs. Par défaut, la longueur du mot de
passe est de 8 caractères. '

' =MOTDEPASSE(10;2;1) permet de créer un mot de passe contenant 10
caractères, dont 2 chiffres et 1 symbole. '

' '

' Auteur : Philippe Enderlin <http://www.enderlinphilippe.fr> '
' -----
' ----- '

Function MOTDEPASSE(Optional Longueur As Long = 8, Optional Nbre_chiffres
As Integer = 0, Optional Nbre_symboles As Integer = 0) As String

' Déclaration des constantes '

' On limite à 10 le nombre maximal de chiffres et de symboles '

Const maxChiffres = 10

```

```

Const maxSymboles = 10

' Déclaration des variables '
Dim i, j, n, debut, fin As Long
Dim strLettres, strChiffres, strSymboles, strChaine, strMot As String
Dim varTab, varTemp As Variant

' Chaînes de caractères contenant lettres, chiffres et symboles '
strLettres = "abcdefghijklmnopqrstuvwxyzaBcDEFGHIJKLMNOPQRSTUVWXYZ"
strChiffres = "0123456789"
strSymboles = "@#&$%$£€(){}[]\`~_<>=+-*/!?!?;.:\"
strChaine = strLettres & strChiffres & strSymboles

' On limite le nombre total de chiffres '
If Nbre_chiffres > maxChiffres Then
    Nbre_chiffres = maxChiffres
' On limite le nombre total de chiffres à la longueur du mot de passe '
ElseIf Nbre_chiffres > Longueur Then
    Nbre_chiffres = Longueur
End If

If (Nbre_symboles + Nbre_chiffres) > Longueur Then
    Nbre_symboles = Longueur - Nbre_chiffres
' On limite le nombre total de symboles '
ElseIf Nbre_symboles > maxSymboles Then
    Nbre_symboles = maxSymboles
' On limite le nombre total de symboles à la longueur du mot de passe '
ElseIf Nbre_symboles > Longueur Then
    Nbre_symboles = Longueur
End If

' On redéfinit la taille du tableau en fonction de la longueur du mot de
passe '
ReDim varTab(Longueur)

```

```

' Si l'argument Nbre_chiffres est renseigné '
If Nbre_chiffres > 0 Then
    ' On détermine les positions de début et de fin de la chaîne '
    debut = CLng(Len(strLettres) + 1)
    fin = CLng(debut + Len(strChiffres) - 1)
    ' Boucle permettant de stocker les chiffres dans un tableau '
    For i = 1 To Nbre_chiffres
        varTab(i) = Mid(strChaine, (Rnd() * (fin - debut) + debut), 1)
    Next i
End If

' Si l'argument Nbre_symboles est renseigné '
If Nbre_symboles > 0 Then
    ' On détermine les positions de début et de fin de chaîne '
    debut = CLng(Len(strLettres + strChiffres) + 1)
    fin = CLng(debut + Len(strSymboles) - 1)
    ' Boucle permettant de stocker les symboles dans un tableau '
    For i = 1 To Nbre_symboles
        varTab(CLng(i + Nbre_chiffres)) = Mid(strChaine, (Rnd() * (fin -
debut) + debut), 1)
    Next i
End If

' Si la longueur du mot de passe est supérieure au nombre de chiffres et
symboles '
If (Longueur > (Nbre_symboles + Nbre_chiffres)) Then
    ' Boucle permettant de stocker les lettres dans un tableau '
    For i = 1 To (Longueur - (Nbre_symboles + Nbre_chiffres))
        varTab(CLng(i + Nbre_chiffres + Nbre_symboles)) = Mid(strChaine,
(Rnd() * (Len(strLettres) - 1) + 1), 1)
    Next i
End If

' Tri aléatoire du tableau contenant lettres, chiffres et symboles '
Randomize
For n = LBound(varTab) To UBound(varTab)

```

```
j = CLng(((UBound(varTab) - n) * Rnd) + n)
If n <> j Then
    varTemp = varTab(n)
    varTab(n) = varTab(j)
    varTab(j) = varTemp
End If
Next n

' Reconstitution du mot de passe après tri aléatoire '
For i = LBound(varTab) To UBound(varTab)
    strMot = strMot & varTab(i)
Next i

' On renvoie le mot de passe '
MOTDEPASSE = strMot

End Function

Sub MDPA()

End Sub
```