SCRIPT POWERSHELL

I. Présentation

PowerShell ISE est solution d'automatisation des tâches. Composée d'un interpréteur de commandes, d'un langage de script. Elle est souvent utilisée pour automatiser la gestion des systèmes.
ISE signifie integrated scripting environment c'est-à-dire que celui-ci présente un éditeur de script à l'ouverture de la console. C'est-à-dire dès que la fenêtre s'affiche.

Le langage utilisé pour écrire les scripts, à savoir l'ensembles des commandes pour automatiser une tâche, est le Shell.
Il s'agit d'un langage permettant de formuler des algorithmes et produire des programmes informatiques qui appliquent ces algorithmes. Dans l'objectif de sauvegarder des données, surveiller la charge des équipements.

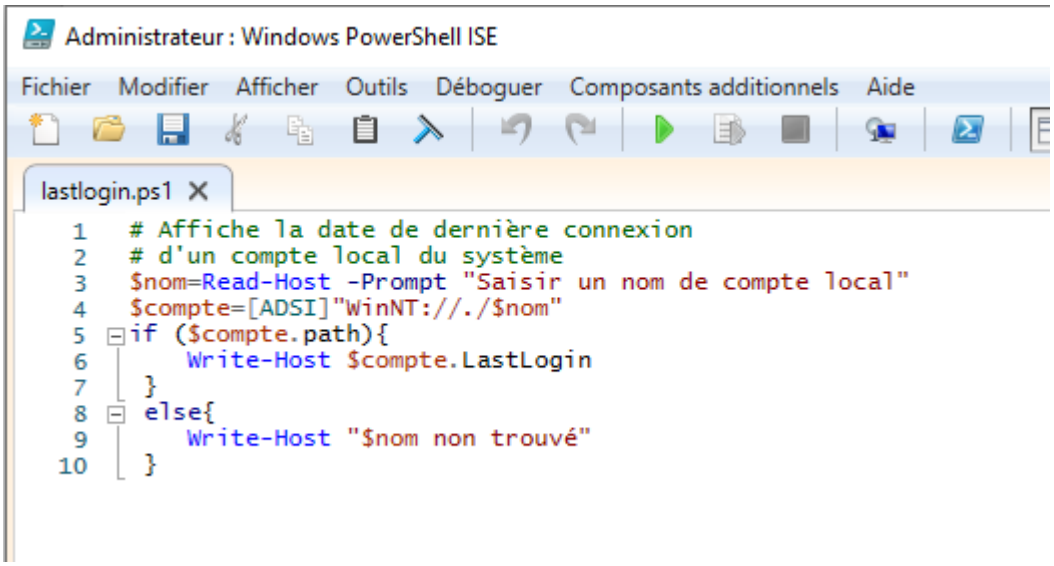
Pour pouvoir exécuter les scripts, il faut mettre en place une stratégie d'exécution des scripts.
La stratégie d'exécution des scripts correspond à autoriser ou pas l'exécution de scripts de confiance se trouvant soit sur l'ordinateur en lui-même, c'est-à-dire en local soit des scripts que l'on a trouvé sur internet, ou bien d'exécuter tous les scripts sans distinction.
Dans PowerShell il existe quatre paramètres de stratégie d'exécution des scripts qui sont résumés dans ce tableau.

Commandes	Stratégies
Restricted	Paramètre par défaut, n'autorise pas l'exécution des scripts
AllSigned	N'exécute que les scripts de confiance, donc signés,
RemoteSigned	Exécute les scripts locaux sans obligation de confiance et les scripts de confiance issus d'Internet
Unrestricted	Autorise l'exécution de tous les scripts.

Le TD que nous allons réaliser va nous montrer comment automatiser la création de comptes utilisateurs. Celui-ci a pour objectif de comprendre comment PowerShell permet d'une part d'administrer un système informatique. Et d'autre part de le contrôler et améliorer les performances du système, au travers de la sécurisation du mécanisme et l'automatisation de l'administration des comptes utilisateurs.

II. Accès aux comptes locaux du système

L'accès à la base locale de comptes utilisateurs d'un système Windows est réalisé avec l'instruction : **[ADSI]"WinNT://.**
Pour filtrer les éléments de la base de comptes, il est possible de spécifier le nom de l'élément recherché.



Ce script permet d'afficher la dernière connexion d'un utilisateur.

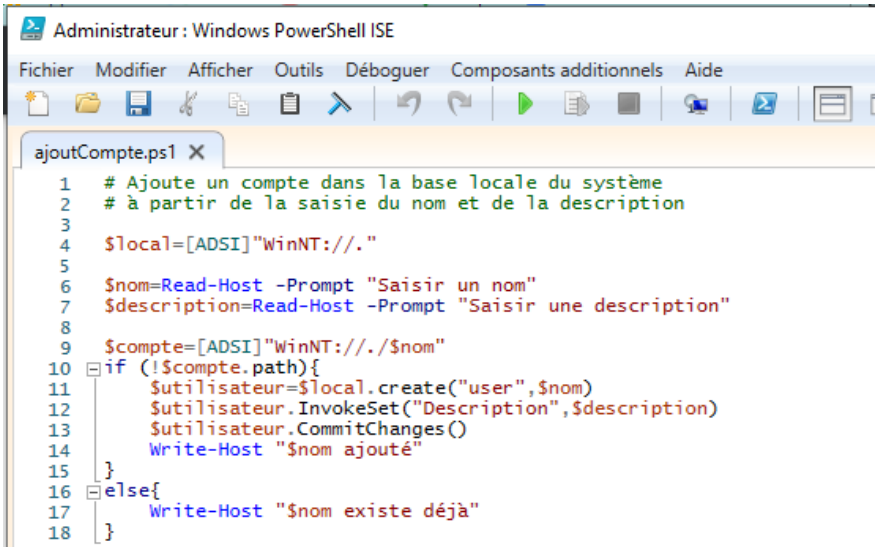
Relt

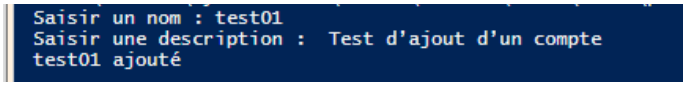
En saisissant **$compte | Get-Member,** dans la console de saisie de commande on obtient les propriétés de la variable $compte. On pourra donc visualiser certaines caractéristiques du compte, en intégrant cette propriété dans le script à la suite de la variable. Comme ci-dessous.

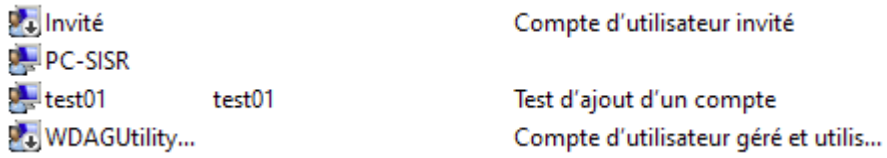
```
$nom=Read-Host -Prompt "Saisir un nom de compte local"
$compte=[ADSI]"WinNT://./$nom"
if ($compte.path){
    Write-Host $compte.LastLogin
    Write-Host $compte.FullName
    Write-Host $compte.Description
 }
 else{
    Write-Host "$nom non trouvé"
 }
```

result

III. Ajout d'un compte local







Modification du script :
Nous avons modifié le script ajoutCompte.ps1 pour que le nom complet soit également saisi et renseigné lors de l'ajout du compte.

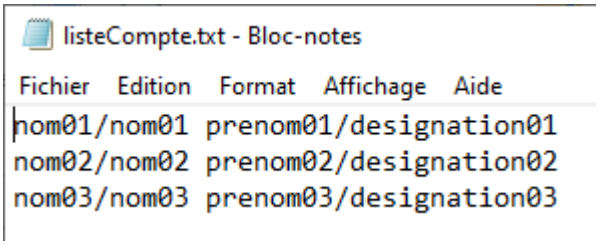
```
4  $local=[ADSI]"WinNT://."
5
6  $nom=Read-Host -Prompt "Saisir un nom"
7  $fullname=Read-Host -Prompt "nom complet"
8  $description=Read-Host -Prompt "Saisir une description"
9
10
11 $compte=[ADSI]"WinNT://./$nom"
12 if (!$compte.path){
13     $utilisateur=$local.create("user",$nom)
14     $utilisateur.InvokeSet("Nom complet",$fullname)
15     $utilisateur.InvokeSet("Description",$description)
16     $utilisateur.CommitChanges()
17     Write-Host "$nom ajouté"
18 }
19 else{
20     Write-Host "$nom existe déjà"
21 }
```

En rajoutant ces lignes nous pouvons renseigner le nom complet du compte. Plus précisément la méthode **InvokeSet** permet de spécifier certaines informations de la variable auquel il est concaténé.

IV. Parcours d'un fichier texte contenant les informations des comptes utilisateurs

Nous avons assigné le chemin de fichier à une variable, pour utiliser son contenu.

```
$fichier="C:\btssio\BTSSIO2\Bloc2\SISR4\poweshell2\TP2\listeCompte.txt"
```



Contenu du fichier listeCompte.txt

Ce script affiche le contenu du fichier listeCompte.txt, que nous avons rentré dans le tableau $colLIgnes. En mettant d'abord le nom puis le nom complet et la description du compte.

```
if (Test-Path $fichier){
    $colLIgnes=Get-Content $fichier

    foreach($ligne in $colLignes){
        $tabCompte=$ligne.Split("/")
        Write-Host $tabCompte[0]
        Write-Host $tabCompte[1]
        Write-Host $tabCompte[2]
    }
}
else{
    Write-Host "$fichier pas trouvé"
}
```

```
nom01
nom01 prenom01
designation01
nom02
nom02 prenom02
designation02
nom03
nom03 prenom03
designation03
```

```
$local=[ADSI]"WinNT://."

$fichier="C:\btssio\BTSSIO2\Bloc2\SISR4\poweshell2\TP2\listeCompte.txt"

if (Test-Path $fichier){
    $colLIgnes=Get-Content $fichier

    foreach($ligne in $colLignes){
        $tabCompte=$ligne.Split("/")
        $nom=$tabCompte[0]
        $nomComplet=$tabCompte[1]
        $descrption=$tabCompte[2]

        $compte=[ADSI]"WinNT://./$nom"
        if (!$compte.path){
            $utilisateur=$local.create("user",$nom)
            $utilisateur.InvokeSet("FullName",$nomcomplet)
            $utilisateur.InvokeSet("Description",$description)
            $utilisateur.CommitChanges()
            Write-Host "$nom ajouté"
        }
        else{
            Write-Host "$nom existe déjà"
        }
```

Ce script va nous permettre de créer trois nouveaux comptes utilisateurs, à partir des noms renseignés dans le fichier listeCompte.txt

V. Conclusion

Nous voyons que grâce aux fonctionnalités proposées par PowerShell ISE à savoir de l'invite de commande et de l'éditeur nous pouvons afficher l'activité d'un compte. En tant qu'administrateur, il est possible d'afficher les détails de l'activité d'un compte en saisissant la variable et la propriété lié à ce que l'on veut voir.
Aussi PowerShell ISE permet de créer des comptes. Il permet donc la gestion de ces comptes. En effet les scripts automatisent la création et la modification de comptes locaux, on peut aussi les supprimer.

VI. Résumé

Par rapport aux objectifs j'ai pu apprendre des nouvelles notions, et différencier les termes commandes et propriété.
Pour ma part, j'ai vu comment administrer des comptes locaux, mais administrer des comptes ne se limite pas à leur création et suppression. Il faut aussi leur accorder des droits. Donc, je teste actuellement des scripts afin d'accorder certains aux utilisateurs.