Microsoft PowerShell pratique

Présentation :

Le but de ce petit recueil est de venir en aide, avec des bouts de scripts, aux administrateurs système pour obtenir telle ou telle information et ensuite arriver à automatiser les tâches plus rapidement et plus facilement. Par conséquent dans ce recueil, nous n'allons pas nous étendre sur l'explication profonde de ces scripts.

Cacher la fenêtre :

Ce script d'une ligne, va rendre la fenêtre du PowerShell invisible mais garde le focus dessus et donc, on peut continuer à entrer des commandes sans avoir la fenêtre à l'écran.

```
$h=(Get-Process -Id
$pid).MainWindowHandle;$ios=[Runtime.InteropServices.HandleRef];$hw=New-
Object $ios (1,$h);$i=New-Object
$ios(2,0);(([reflection.assembly]::LoadWithPartialName("WindowsBase")).GetType(
"MS.Win32.UnsafeNativeMethods"))::SetWindowPos($hw,$i,0,0,100,100,16512)
```

Connaitre la lettre du lecteur nommé « USBKEY »

Ce script d'une ligne, va retourner la lettre du lecteur nommé « USBKEY » pour l'exemple. Si le lecteur n'existe pas il ne retourne rien.

```
[System.IO.DriveInfo]::getdrives() |where-object {$_.VolumeLabel -match
"USBKEY"}|sort {$_.name} |foreach-object {;  echo "$(echo $_.name)";}
```

Lister les programmes qui tournent.

Ce script liste tous les process qui tournent avec le titre de la fenêtre

```
gps | ? {$_.mainwindowhandle -ne 0} | select name, mainwindowtitle
```

Lancer un programme et récupérer son PID.

Exemple de script avec notepad.

```
(Start-Process Notepad -passthru).ID
```

Faire un temps de pause

```
Start-Sleep -Seconds 10
```

Attendre la pression d'une touche

```
$touche = $Host.UI.RawUI.ReadKey('NoEcho,IncludeKeyDown')
```

Dans la variable $touche on pourra récupérer le code de la touche, le caractère, statut NumLock.

Récupérer les mots de passe stockés dans Internet Explorer

```
$ClassHolder = [Windows.Security.Credentials.PasswordVault,Windows.Security.Credentials,ContentType=WindowsRuntime];$VaultObj = new-object Windows.Security.Credentials.PasswordVault;$VaultObj.RetrieveAll() | foreach { $_.RetrievePassword(); $_ }
```

Changer le titre de la fenêtre PowerShell

```
$Host.UI.RawUI.WindowTitle = "Microsoft Windows Update"
```

Ouvrir un popup pour demander des informations (ex. password).

```
$PWD=[System.Reflection.Assembly]::LoadWithPartialName('Microsoft.VisualBasic') | Out-Null

$PWD= [Microsoft.VisualBasic.Interaction]::InputBox("Enter Password ")
```

Encoder un texte en Base64

```
[System.Convert]::ToBase64String( [System.Text.Encoding]::UTF8.GetBytes("SomePassword"))
```

Retourne : U29tZVBhc3N3b3Jk

Décoder un texte en Base64

```
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("U29tZVBhc3N3b3Jk"))
```

Retourne : SomePassword

Informations sur le système avec wmic

```
wmic os get Caption
```

Caption

Microsoft Windows 7 Professionnel

```
wmic os get CSName
```

CSName

MYPERSONALPC

```
wmic os get WindowsDirectory
```

WindowsDirectory

C:\Windows

```
wmic os get MUILanguages
```

MUILanguages

{"fr-FR"}

```
wmic os get OSArchitecture
```

OSArchitecture

64 bits

Informations sur le système avec $env:

Pour avoir la liste de toutes les informations disponibles on peut faire :

```
Get-ChildItem Env:
```

Ensuite on utilise par exemple pour connaitre le nom d'utilisateur

```
$env:username
```

L'emplacement du APPDATA

```
$env:appdata
```

Le dossier home de l'utilisateur

```
$env:homepath
```

Le nom de l'ordinateur (équivalent à wmic os get CSName)

```
$env:computername
```

Lister les privilèges de l'utilisateur

```
whoami /priv
```

Lister les utilisateurs

```
Net users
```

Lister les détails d'un utilisateur (ex. Administrateur)

```
Net user Administrateur
```

Lister les groupes locaux

```
Net localgroup
```

Lister les détails d'un groupe

```
Net localgroup Administrateurs
```

Lister les partages réseaux

```
Net share
```

Chercher des fichiers

```
findstr /si password *.xml *.ini *.txt *.config
```

Chercher dans la base des registres

Dans la base local machine

```
REG QUERY HKLM /F "password" /t REG_SZ /S /K
```

Dans la base Current user

```
REG QUERY HKCU /F "password" /t REG_SZ /S /K
```

Chercher une clé particulière

```
REG QUERY "HKLM\Software\Microsoft\FTH" /V RuleList
```

Lister les SSID des Wifi enregistrés

```
netsh wlan show profile
```

Afficher la clé en clair d'un SSID stocké (ex. Livebox456)

```
netsh wlan show profile Livebox456 key=clear
```

Lister les services

```
Get-Service
```

Autre méthode

```
Get-WmiObject -Query "Select * from Win32_Process" | where {$_.Name -notlike "svchost*"} | Select Name, Handle, @{Label="Owner";Expression={$_.GetOwner().User}} | ft -AutoSize
```

Lister les services lances par le système

```
tasklist /v /fi "username eq système"
```

Savoir la version de PowerShell présente 2.0, 3.0 …

```
REG QUERY "HKLM\SOFTWARE\Microsoft\PowerShell\1\PowerShellEngine" /v PowerShellVersion
```

Lister les programmes installés

```
Get-ChildItem 'C:\Program Files', 'C:\Program Files (x86)' | ft
Parent,Name,LastWriteTime
```

Juste les noms

```
Get-ChildItem -path Registry::HKEY_LOCAL_MACHINE\SOFTWARE | ft Name
```

Lister les programmes au startup

```
wmic startup list full
```

Savoir la règle sur l'exécution des fichier PS1

```
Get-ExecutionPolicy
```

Débloquer l'exécution des .ps1

```
set-ExecutionPolicy unrestricted
```

Tester si connecté à internet

```
Test-Connection -computer "google.com" -count 1 -quiet
```

Retourne True si connecté

Récupérer l'IP du Gateway

```
Get-WmiObject -Class Win32_IP4RouteTable |  where { $_.destination -eq '0.0.0.0'
-and $_.mask -eq '0.0.0.0'} |  Sort-Object metric1 | select nexthop,
metric1,interfaceindex
```

Autre méthode

```
$computer = $env:COMPUTERNAME;Get-WmiObject
win32_networkAdapterConfiguration -ComputerName $computer | Select
index,description,defaultipgateway |Format-Table -AutoSize
```

Savoir si on est Administrateur

```
$currentPrincipal = New-Object
Security.Principal.WindowsPrincipal([Security.Principal.WindowsIdentity]::GetCurre
nt());$currentPrincipal.IsInRole([Security.Principal.WindowsBuiltInRole]::Administra
tor)
```

Retourne True si oui