PowerShell: Monter un lecteur réseaux

Dans cette exemple, on va mettre en place un fichier à plat pour voir l'ensemble des actions pour monter son lecteur réseau. Cette méthode n'est pas sécure car une personne qui a l'habitude va vite voir que vous utiliser la conversion basique. Cependant, vous pouvez utiliser cette méthode avec une conversion via PS2EXE, c'est déjà plus délicat de récupérer les identifiants.

```
[String]$NetworkLetter = 'O'
[String]$NetworkPath = '\\localhost\OSD$'
# Credential zone
[String]$User = 'Share'
# Do not add this on your final code this is here to example
[String]$Pass = '7.fhH98R+a-Ca324'
[String]$Password =
[Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes($Pass))
#Result => Ny5maEg5OFIrYS1DYTMyNA==
[String]$Decoding =
[Text.Encoding]::Utf8.GetString([Convert]::FromBase64String($Password))
#Result => 7.fhH98R+a-Ca324
[SecureString]$SecurePassword = ConvertTo-SecureString $Password -
AsPlainText -Force
[PSCredential]$Creds = New-Object System.Management.Automation.PSCredential
$User, $SecurePassword
New-PSDrive -Name $NetworkLetter -PSProvider FileSystem -Root $NetworkPath -
Credential $Creds -Scope global -Persist
```

Pour sécuriser cela, nous allons utiliser une clef AES pour que le ConvertTo-SecureString soit plus personnel. Attention cette clef AES ne doit pas être perdu.

```
# Prompt you to enter the username and password
$credObject = Get-Credential
# The credObject now holds the password in a 'securestring' format
$passwordSecureString = $credObject.password
# Create C:\temp if the folder isn't exit
if ((test-path C:\temp) -eq $false){New-item -Path C:\Temp\ -Type Directory
| Out-Null}
# Define a location to store the AESKey
$AESKeyFilePath = "C:\temp\aeskey.txt"
# Define a location to store the file that hosts the encrypted password
$credentialFilePath = "C:\temp\credpassword.txt"
# Generate a random AES Encryption Key.
$AESKey = New-Object Byte[] 32
[Security.Cryptography.RNGCryptoServiceProvider]::Create().GetBytes($AESKey)
# Store the AESKey into a file. This file should be protected! (e.g. ACL on
the file to allow only select people to read)
Set-Content $AESKeyFilePath $AESKey # Any existing AES Key file will be
overwritten
$password = $passwordSecureString | ConvertFrom-SecureString -Key $AESKey
```

```
Add-Content $credentialFilePath $password
```

Il suffira donc juste de réutiliser le fichier credpassword.txt dans vos futurs scripts

```
[String]$NetworkLetter = 'O'
[String]$NetworkPath = '\\localhost\OSD$'
$AESKeyFilePath = "C:\temp\aeskey.txt"
$SecurePwdFilePath = "C:\temp\credpassword.txt"
$username = 'Share'
$AESKey = Get-Content $AESKeyFilePath
$pwdTxt = Get-Content $SecurePwdFilePath
$securePwd = $pwdTxt | ConvertTo-SecureString -Key $AESKey
[PSCredential]$creds = New-Object System.Management.Automation.PSCredential
-ArgumentList $username, $securePwd
New-PSDrive -Name $NetworkLetter -PSProvider FileSystem -Root $NetworkPath -
Credential $Creds -Scope global -Persist
```