

SISTEMES INFORMÀTICS

1r DAW - C.F.G.S.

UD 6. Administració de Windows 10

Exercicis 1

Alumne:

Grup:

Estos exercicis estan dissenyats per a fer tasques comunes d'administració de sistemes de fitxers, usuaris i grups locals mitjançant PowerShell. Assegura't d'executar estos comandos amb els privilegis adequats per a realitzar estes operacions.

Cada apartat consta de 10 exercicis. Els 5 primers són de nivell bàsic, mentre que la resta són exercicis avançats que proporcionen desafiaments addicionals i aborden tasques més complexes.

Per a la realització d'estos exercicis has de consultar el fitxer sobre Powershell que hi ha en Aules, així com ampliar coneixements en Internet.

Recorda que pots usar la teua màquina virtual per fer les proves que necessites.

Exercici sobre el sistema de fitxers

1) Situat en el teu directori personal, utilitza PowerShell per llistar tots els arxius del directori C:\Users.

Solució:

Get-ChildItem -Path C:\Users

2) Crea un nou directori anomenat "MisArchivos" en l'escriptori.

Solució

New-Item -ItemType Directory -Path "\$env:USERPROFILE\Desktop\MisArchivos"

3) Còpia tots els arxius amb extensió .txt de C:\Origen a D:\Destino.

Solució

Copy-Item -Path C:\Origen.txt -Destination D:\Destino*

4) Elimina tots els arxius en C:\Backup que tinguin més d'una setmana d'antiguitat.

Nota: estudia el funcionament del cmdlet Where-Object.

Solució

Get-ChildItem -Path C:\Backup | Where-Object { \$_.LastWriteTime -lt (Get-Date).AddDays(-7) } | Remove-Item

5) Canvia el nom de tots els arxius en el directori D:\Fotos, afegint el prefixe

"Vacaciones_".

Nota: estudia el funcionament del cmdlet ForEach-Object

Solució:

Get-ChildItem -Path D:\Fotos | ForEach-Object { Rename-Item \$_.FullName -NewName ("Vacaciones_" + \$_.Name) }

6) Cerca i mostra detalls de tots els arxius en C:\Archivos amb una grandària superior a 100 MB.

Nota: estudia el funcionament del cmdlet Where-Object

Solució

Get-ChildItem -Path C:\Archivos -File | Where-Object { \$_.Length -gt 100MB } | Select-Object Name, Length, LastWriteTime

7) Llig un arxiu CSV amb la estructura de directoris y crea la mateixa estructura en

D:\Datos.

Solució

\$csvData = Import-Csv -Path C:\EstructuraDirectorios.csv

foreach (\$entry in \$csvData) {

\$fullPath = Join-Path -Path "D:\Datos" -ChildPath \$entry.Path

New-Item -ItemType Directory -Path \$fullPath -Force

}

8) Crea un script anomenat `comprimir.ps1`. L'script ha de comprimir tots els arxius en `C:\Documentos` en un arxiu ZIP anomenat "`Documentos.zip`" i encriptat amb contrasenya. Nota: La contrasenya pot ser la que vosaltres considereu.

Solució

\$sourcePath = "C:\Documentos"

\$zipFilePath = "C:\Backup\Documentos.zip"

\$zipPassword = ConvertTo-SecureString -String "MiContraseñaSegura" -AsPlainText -Force

*Compress-Archive -Path \$sourcePath -DestinationPath \$zipFilePath -CompressionLevel
Optimal -Password \$zipPassword*

9) Crea un script que realitzi una còpia de seguretat del arxiu en `C:\Datos` a `D:\Backup` i registre les accions en un arxiu de registre anomenat "`BackupLog.txt`". Nota: El fitxer de log ha de tindre una línia amb el nom de cada fitxer copiat.

Solució

\$sourcePath = "C:\Datos"

\$destinationPath = "D:\Backup"

\$logFilePath = "D:\Backup\BackupLog.txt"

*Copy-Item -Path \$sourcePath -Destination \$destinationPath -Recurse -Force | Out-File -
Append -FilePath \$logFilePath*

10) Crea un script, anomenat "`cercarisubstituir.ps1`" que cerque i substitueixi totes les instàncies de la cadena "`AntiguoTexto`" per "`NuevoTexto`" en archivos `.txt` en `C:\Textos`.

Solució

```
Get-ChildItem -Path C:\Textos -Filter .txt -File | ForEach-Object {  
  
    (Get-Content $_.FullName) -replace 'AntiguoTexto', 'NuevoTexto' | Set-Content  
    $_.FullName  
  
}
```

Exercici sobre usuaris i grups locals

1) Llista tots els usuaris locals del sistema.

Solució

```
Get-WmiObject Win32_UserAccount | Where-Object { $_.LocalAccount -eq $true }
```

2) Crea un nou usuari anomenat "NuevoUsuario" amb contrasenya "P@ssw0rd".

Solució

```
New-LocalUser -Name "NuevoUsuario" -Password (ConvertTo-SecureString "P@ssw0rd" -  
AsPlainText -Force) -Enabled $true
```

3) Canvia la contrasenya de l'usuari creat en l'exercici anterior a "NuevaContraseña123".

Solució

```
Set-LocalUser -Name "NuevoUsuario" -Password (ConvertTo-SecureString  
"NuevaContraseña123" -AsPlainText -Force)
```

4) Crea un nou grup anomenat "NuevosUsuarios" en el sistema.

Solució

```
New-LocalGroup -Name "NuevosUsuarios"
```

5) Afig l'usuari creat en l'exercici 2 al grup "NuevosUsuarios".

Solució

Add-LocalGroupMember -Group "NuevosUsuarios" -Member "NuevoUsuario"

6) Crea un script, anomenat “usuarisigrups.ps1” que lliste tots els usuaris locals i els grups als que pertanyen.

Solució

```
Get-WmiObject Win32_UserAccount | ForEach-Object {  
  
    $user = $_  
  
    $groups = Get-WmiObject Win32_GroupUser | Where-Object { $_.PartComponent -match  
$user.LocalPath }  
  
    [PSCustomObject]@{  
  
        UserName = $user.Name  
  
        Groups = $groups | ForEach-Object { $_.GroupComponent -replace  
'Domain="|",Name=".' }  
  
    }  
  
}
```

7) Crea un script, anomenat “usuaris_inactius.ps1”, que deshabilite tots els usuaris locals que no han iniciat sessió en els últims 30 dies.

Solució

```
$inactiveUsers = Get-WmiObject Win32_UserAccount | Where-Object { $_.LastLogon -lt  
(Get-Date).AddDays(-30) }  
  
$inactiveUsers | ForEach-Object { Disable-LocalUser -Name $_.Name }
```

8) Crea un script per a llegir un arxiu CSV amb detalls d’usuaris i crea-los en el sistema.

Solució

```
$usersData = Import-Csv -Path C:\NuevosUsuarios.csv  
  
foreach ($userData in $usersData) {
```

```
New-LocalUser -Name $userData.UserName -Password (ConvertTo-SecureString  
$userData.Password -AsPlainText -Force) -Enabled $true  
  
}
```

9) Crea un grup anomenat "DepartamentoFinanzas" i otorga permisos de lectura y escritura en D:\Finanzas.

Solució

```
New-LocalGroup -Name "DepartamentoFinanzas"  
  
$folderPath = "D:\Finanzas"  
  
$group = "DepartamentoFinanzas"  
  
icacls $folderPath /grant "$($group):(OI)(CI)RXW"
```

10) Crea un script, anomenat "usuarios_grup_inactius.ps1" que elimine tots els usuaris que no han iniciat sessió en els últims 60 dies del grup "UsuariosActivos".

Solució

```
$activeUsersGroup = "UsuariosActivos"  
  
$inactiveUsers = Get-WmiObject Win32_UserAccount | Where-Object { $_.LastLogon -lt  
(Get-Date).AddDays(-60) }  
  
$inactiveUsers | ForEach-Object { Remove-LocalGroupMember -Group $activeUsersGroup -  
Member $_.Name }
```