



CFGs ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

IMPLANTACIÓN DE SISTEMAS OPERATIVOS



Unidades de red y las ACL

Índice

- 1.- Unidades de red
- 2.- Las ACL's.



1.- Unidades de red.

Comandos para crear unidades para acceder a un recurso compartido

```
PS C:\WINDOWS\system32> get-command *psdrive*
```

| CommandType | Name | Version |
|-------------|----------------|---------|
| ----- | ---- | ----- |
| Cmdlet | Get-PSDrive | 3.1.0.0 |
| Cmdlet | New-PSDrive | 3.1.0.0 |
| Cmdlet | Remove-PSDrive | 3.1.0.0 |



1.- Unidades de red.

Get-PSDrive → para ver una lista de las unidades.

```
PS C:\WINDOWS\system32> get-psdrive
```

| Name | Used (GB) | Free (GB) | Provider | Root |
|----------|-----------|-----------|-------------|--------------------|
| ---- | ----- | ----- | ----- | ---- |
| Alias | | | Alias | |
| C | 88,09 | 57,00 | FileSystem | C:\ |
| Cert | | | Certificate | \ |
| D | 326,99 | 138,77 | FileSystem | D:\ |
| E | | | FileSystem | E:\ |
| Env | | | Environment | |
| F | | | FileSystem | F:\ |
| Function | | | Function | |
| HKCU | | | Registry | HKEY_CURRENT_USER |
| HKLM | | | Registry | HKEY_LOCAL_MACHINE |
| Variable | | | Variable | |
| WSMan | | | WSMan | |

```
PS C:\WINDOWS\system32> get-psdrive -PSProvider FileSystem
```

| Name | Used (GB) | Free (GB) | Provider | Root |
|------|-----------|-----------|------------|------|
| ---- | ----- | ----- | ----- | ---- |
| C | 88,09 | 57,00 | FileSystem | C:\ |
| D | 326,99 | 138,77 | FileSystem | D:\ |
| E | | | FileSystem | E:\ |
| F | | | FileSystem | F:\ |



1.- Unidades de red.

Get-PSDrive → para ver una lista de las unidades.

El parámetro **PSProvider** permite mostrar únicamente las unidades de Windows PowerShell compatibles con un proveedor determinado.

Filesystem → muestra las unidades compatibles con el sistemas de archivos de Windows.

Registry → muestra las unidades compatibles con el registro.

```
PS C:\WINDOWS\system32> get-psdrive -PSProvider Registry
```

| Name | Used (GB) | Free (GB) | Provider | Root |
|------|-----------|-----------|----------|--------------------|
| ---- | ----- | ----- | ----- | ---- |
| HKCU | | | Registry | HKEY_CURRENT_USER |
| HKLM | | | Registry | HKEY_LOCAL_MACHINE |



1.- Unidades de red.

New-PSDrive → para crear una unidad de red.

SINTAXIS

```
New-PSDrive -Name "Z" -PSProvider FileSystem -Root \\192.168.0.158\pruebas
```

-Root → indica la ruta de acceso a la raíz de la nueva unidad.

Esta unidad de red es temporal. Si queremos que nos dure, deberemos utilizar la opción **persist**.

```
New-PSDrive -Name "Z" -PSProvider FileSystem -Root \\192.168.0.158\pruebas -Persist
```



1.- Unidades de red.

Para acceder a una unidad de red con credenciales

Si para acceder al recurso se necesita usuario y contraseña (credenciales), tenemos que actuar de la siguiente forma.

1.- SERVIDOR → Compartimos la carpeta d:\pruebas

```
New-SmbShare -Name pruebas -Path d:\pruebas -FullAccess alumno
```

2.- CLIENTE → Aseguramos la contraseña del usuario: nos pedirá la contraseña.

```
$alumno = Get-Credential
```

3.- CLIENTE → Creamos una unidad de red para acceder. → Get-Psdrive

```
New-PSDrive -Name "t" -PSProvider FileSystem -Root \\192.168.0.158\pruebas -Credential $alumno
```



1.- Unidades de red.

4.- CLIENTE → Accedemos a la carpeta compartida y creamos un archivo.

```
PS C:\WINDOWS\system32> z:  
PS z:\> ls  
PS z:\> ni gatitos
```




1.- Unidades de red.

SERVIDOR → Obtener información sobre sesiones establecidas.

```
PS D:\> get-command *smbsession*
```

| CommandType | Name |
|-------------|------------------|
| Function | Close-SmbSession |
| Function | Get-SmbSession |

SERVIDOR → Cerrar sesiones establecidas

```
PS C:\WINDOWS\system32> Get-SmbSession
```

| SessionId | ClientComputerName | ClientUserName | NumOpens |
|--------------|----------------------------|----------------|----------|
| 257698037841 | 192.168.0.23 | TAURO\alumno | 1 |
| 257698037845 | [fe80::3df7:39c5:d9d5:eec] | TAURO\alumno | 1 |

```
PS C:\WINDOWS\system32> Get-SmbSession | Close-SmbSession
```



1.- Unidades de red.

SERVIDOR → Obtener información sobre archivos abiertos.

```
PS D:\> get-help Get-SmbOpenFile
```

NOMBRE

Get-SmbOpenFile

SINOPSIS

Retrieves basic information about the files that are open on behalf of the clients of the SMB server.



2.- Las ACL.

Las ACL

La administración de permisos en Windows PowerShell se deriva de dos cmdlets:
Get-ACL y **Set-ACL**.

Como cabría esperar, **Get-ACL** recupera la ACL a partir de un recurso.

A continuación, puede modificar la ACL para que se adapte a sus necesidades y usar **Set-ACL** para volver a escribirla en el recurso.

Pasos:

- 1.- Obtener la ACL del recurso.
- 2.- Definir una nueva regla.
- 3.- Modificar la ACL
- 4.- Escribir la ACL



2.- Las ACL.

Las ACL

Figure 1 Aplicar un permiso para una entidad de seguridad a los archivos y carpetas de un directorio

```
#ChangeACL.ps1
$Right="FullControl"

#The possible values for Rights are
# ListDirectory, ReadData, WriteData
# CreateFiles, CreateDirectories, AppendData
# ReadExtendedAttributes, WriteExtendedAttributes, Traverse
# ExecuteFile, DeleteSubdirectoriesAndFiles, ReadAttributes
# WriteAttributes, Write, Delete
# ReadPermissions, Read, ReadAndExecute
# Modify, ChangePermissions, TakeOwnership
# Synchronize, FullControl

$StartingDir=Read-Host "What directory do you want to start at?"
$Principal=Read-Host "What security principal do you want to grant" `
"$Right to? `n Use format domain\username or domain\group"

#define a new access rule.
#note that the $rule line has been artificially broken for print purposes.
#it needs to be one line. the online version of the script is properly
#formatted.
$rule=new-object System.Security.AccessControl.FileSystemAccessRule
($Principal,$Right,"Allow")

foreach ($file in $(Get-ChildItem $StartingDir -recurse)) {
    $acl=get-acl $file.FullName

    #Add this access rule to the ACL
    $acl.SetAccessRule($rule)

    #Write the changes to the object
    set-acl $File.Fullname $acl
}
```

Pasos:

- 1.- Obtener la ACL
- 2.- Definir una nueva regla.
- 3.- Modificar la ACL
- 4.- Escribir la ACL



2.- Las ACL.

SERVIDOR → Pasos:

```
New-SmbShare -Name pruebas -Path d:\pruebas -FullAccess alumno
```

1.- Obtener la ACL

```
$acl = get-acl D:\pruebas
```

```
$acl | fl
```

2.- Definir una nueva regla.

```
PS D:\> $ar = New-Object System.Security.AccessControl.FileSystemAccessRule ("alumno", "Fullcontrol", "Allow")
```

3.- Modificar la ACL

```
PS D:\> $acl.SetAccessRule($ar)
```

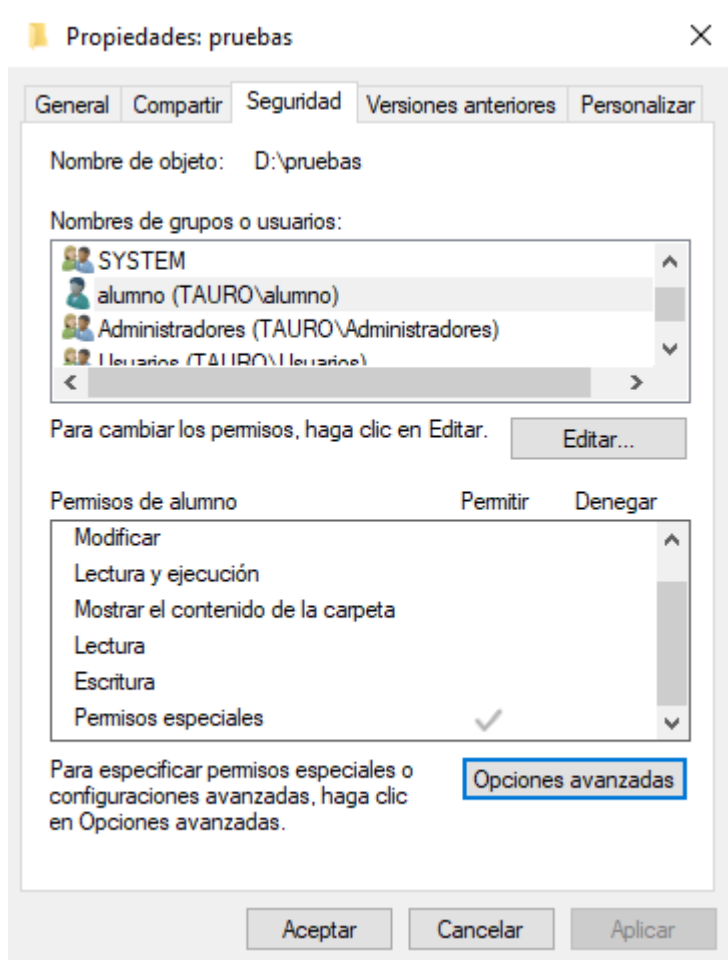
```
$ar
```

4.- Escribir la ACL

```
PS D:\> Set-Acl "d:\pruebas" $acl
```



2.- Las ACL.



```
get-acl d:\pruebas | fl _
```

Entidad de seguridad: alumno (TAURO\alumno) [Seleccionar una entidad de seguridad](#)

Tipo: Permitir

Se aplica a: Solo esta carpeta

Permisos básicos:

- ☒ Control total
- ☒ Modificar
- ☒ Lectura y ejecución
- ☒ Mostrar el contenido de la carpeta
- ☒ Lectura
- ☒ Escritura
- ☐ Permisos especiales

☐ Aplicar estos permisos solo a objetos y/o contenedores dentro de este contenedor



3.- Conclusiones.



Don Jones

Don Jones has been in the IT industry since the mid 1990s, and has been a recipient of Microsoft's "Most Valuable Professional" Award since 2003. He's a co-founder of PowerShell.org and The DevOps Collective, and a Curriculum Director for online video training company Pluralsight. Don authored some of the first books ever published for Windows PowerShell, and is the co-author of *Learn Windows PowerShell in a Month of Lunches*, the bestselling entry-level book for PowerShell. Don's a sought-after speaker at technology conferences like Microsoft Ignite, TechMentor, and many more. You can find him on Twitter @concentratedDon, or on his website, DonJones.com.

www.ScriptingTraining

Opina lo siguiente con respecto a combinar comandos
CMD y cmdlet en Windows PowerShell.



3.- Conclusiones.

Windows PowerShell ofrece **un medio simplificado para trabajar con permisos**.

Puede continuar usando herramientas de línea de comandos **como icacls**, que se diseñaron específicamente para simplificar el trabajo con varios tipos de permisos.

Aunque su sintaxis es completamente diferente a la sintaxis estándar usada por los cmdlets de Windows PowerShell. **Puede usar Windows PowerShell para automatizar estas utilidades.**

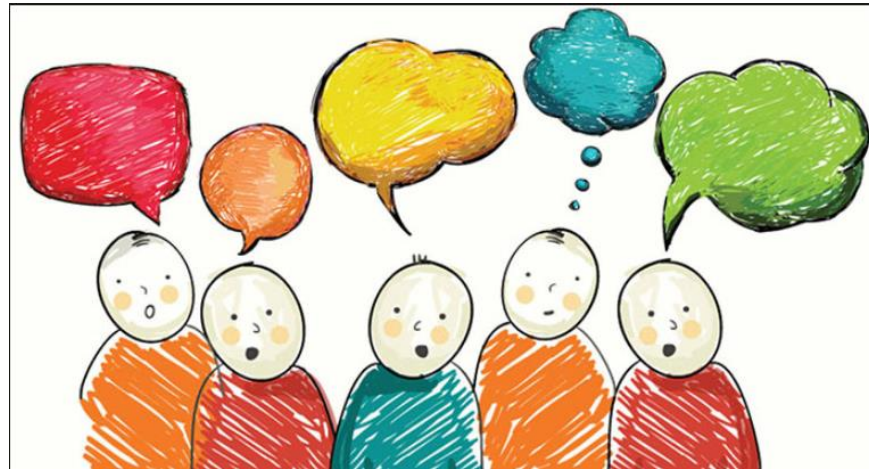
Me he encontrado con muchas personas, los llamaré "**puristas**", que insisten en que no es correcto usar una línea de comandos antigua con Windows PowerShell.

No lo entiendo. No voy a dedicar horas extra a intentar hacer algo de una forma concreta sólo porque parece "más puro". Y el equipo de Microsoft Windows PowerShell tiene una actitud similar. Si una herramienta existente hace el trabajo, entonces Windows PowerShell le permite usar dicha herramienta en el shell. **¿Por qué volver a inventar la rueda?**



3.- Conclusiones.

Tú, ¿qué opinas?





Sugerencias/mejoras del tema



Sugerencias /mejoras del tema