

6. Combinando comandos

Wednesday, September 9, 2020 8:43 PM

El manejo de cualquier shell para introducir comandos individuales, no representa un reto muy complicado, sin embargo, cuando requerimos ejecutar múltiples comandos o, inclusive, replicarlo en múltiples equipos, estas tareas deben hacerse de forma más eficiente.

Las 2 formas de combinar comandos en PowerShell son:

- Usando el pipeline de PowerShell
- Guardo el código en scripts externos

Iniciando servicios en Windows

Analicemos el siguiente código:

```
PS> $serviceName = 'wuauserv'
PS> Get-Service -Name $serviceName
PS> Start-Service -Name $serviceName
```

Hacer el proceso anterior para iniciar un servicio, es fácil, pero imaginen que vamos a iniciar cientos de servicios en un equipo y luego replicar esos en varias computadoras de nuestra infraestructura.

Utilizando el pipeline

☆ pipeline de PowerShell

Herramienta que permite enviar la salida de un comando directamente como la entrada para otro comando.

Hace uso del *operador pipe* |

Sintaxis:

```
PS> command1 | command2
```

El pipeline es también usado en CMD y BASH sin embargo, en PowerShell la salida resultado de un comando es pasada a otro como un objeto y no como texto.

Ahora rescribiremos los comandos de la sección anterior usando el pipeline.

```
PS> Get-Service -Name 'wuauserv' | Start-Service
```

O, inclusive, podemos escribir el comando anterior de la siguiente manera:

```
PS> 'wuauserv' | Get-Service | Start-Service
```

¿Cómo podrías explicar la lógica detrás de estos comandos en el pipeline?

Ejecutando el contenido de arreglos con el pipeline

Usaremos el siguiente archivo de texto:



servicios

Te recomiendo guardarla directamente en C:

Ahora ejecuta el siguiente comando:

```
PS> Get-Content -Path C:\servicios.txt
```

Resultado obtenido:

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Get-Content -Path C:\servicios.txt
wuauserv
W32Time
```

¿Cómo podríamos obtener el status de todos los servicios del archivo?

¿Cómo podríamos iniciarlos todos?

```
PS> Get-Content -Path C:\servicios.txt | Get-Service
```

```
PS> Get-Content -Path C:\servicios.txt | Get-Service | Start-Service
```

NOTA: No hay límite en la cantidad de comandos que puedo agrupar en el pipeline, pero se recomienda que si se van a agrupar 5 o más se replantee como se resolverá tu problema.

NOTA2: No todos los comandos se pueden colocar en pipeline, ya que hay algunos requieren ciertos tipos de entradas en específico y otros que no aceptan entradas en general.

☆ parameter binding

Es el proceso que se realiza cuando pasas parámetros a un comando.

En este proceso se trata de emparejar a cada objeto que se pasa al comando a los parámetros especificados por el creador de comando.

Intenta ejecutar el siguiente comando, ¿qué ocurre?

```
PS> 'string' | Get-Process
```

Para evitar lo anterior, podemos optar por lo siguiente:

```
PS> Get-Help -Name Get-Service -Full
```

Ejecutando scripts

En PowerShell puedes ejecutar scripts externos utilizando una sola línea.

NOTA: Considera que no hay nada que tu puedas hacer en un script que no que se pueda hacer en la consola, pero es más simple ejecutar un script que ya contenga todo.

Fijar la política de ejecución

Por default, PS no te deja ejecutar ningún script, ya que posee una política de ejecución que tiene 4 configuraciones principales:

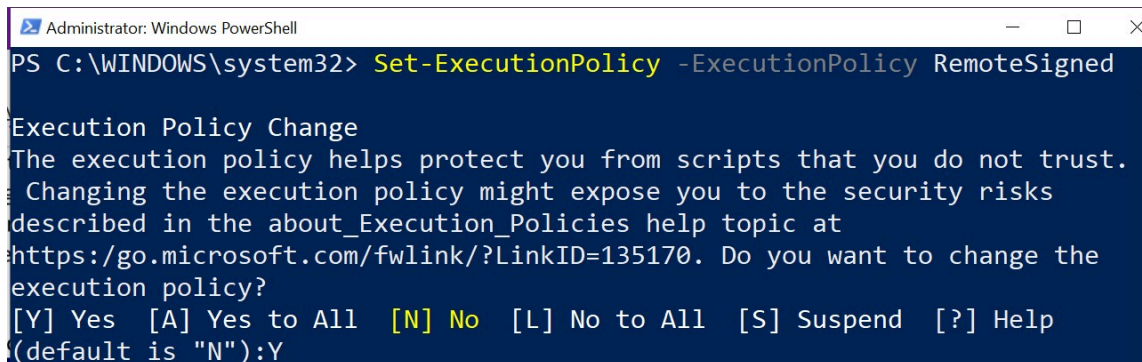
- **Restricted.** No permite ejecutar scripts.
- **AllSigned.** Solo permite ejecutar scripts que han sido firmados criptográficamente por una entidad de confianza.
- **RemoteSigned.** Puedes ejecutar cualquier script local o cualquier script que hayas descargado, siempre que haya sido criptográficamente firmado por una entidad de confianza.
- **Unrestricted.** Puedes ejecutar cualquier script.

Como podemos ver la política de ejecución?

```
PS> Get-ExecutionPolicy
```

Para cambiar la política de ejecución:

```
PS> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust.
Changing the execution policy might expose you to the security risks
described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the
execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "N"):Y
```

☆ Scripts firmados criptográficamente

Es una cadena de texto que se agrega al final del script como un comentario. Estas firmas criptográficas son generadas por un certificado instalado en tu computadora.

Una firma criptográfica permite que PowerShell sepa que la fuente del script y su autor son quienes dicen ser.

Escribiendo scripts en PowerShell - Hola mundo!

Para escribir un script puedes usar cualquier editor de texto, pero se recomienda usar Poser Shell Integrated Scripting Enviroment (PowerShell ISE). Para ejecutarlo desde PowerShell:

```
PS> powershell_ise.exe
```

Realiza lo siguiente:

1. File -> New
2. File -> Save
3. Guárdalo como Example1.ps1 en C:\
4. En la ventana de edición escribe el siguiente comando:
Write-Host "Hello, I am in a script!"
5. En la ventana inferior ejecuta lo siguiente:
PS> C:\Example.ps1

NOTA: Los archivos ps1 no pueden ser insertados en OneNote.

Referencias:

- Bertram, Adam (2020) . *PowerShell for SysAdmins*. "no starch press" <https://nostarch.com/powershellsysadmins>