

Recuperación PowerShell→Jesús Padilla Crespo

1.hacer un script que pida un nombre de servicio(get-service nos permite mostrar todos).se debe comprobar que ese servicio existe y en caso contrario debe volver a pedirlo. Si el proceso está parado debe preguntarse si se quiere iniciar y si está iniciado debe preguntarse si se quiere parar.

```
do
{
    $nombre = Read-Host "Dime el nombre de un servicio"
    $compservicio = (Get-Service -name $nombre -ErrorAction
silentlyContinue).count

    if ($compservicio -ne 0)
    {
        $compstatus = (Get-Service -Name $nombre).Status

        if ($compstatus -eq "Stopped")
        {
            $parado = Read-Host "Este proceso se encuentra parado. ¿Desea
iniciarlo? Pulsa Y para iniciarlo y N para no iniciarlo"
            switch ($parado)
            {
                "Y" { Write-Host "Iniciando programa..."
                    Start-Service -Name $nombre}
                "N" {Write-Host "No se han realizado modificaciones"}
            }
        }
        else
        {
            $iniciado = Read-Host "Este proceso se encuentra iniciado. ¿Desea
pararlo? Pulsa Y para pararlo y N para no pararlo"
            switch ($iniciado)
            {
                "Y" {Write-Host "Cerrando programa..."
                    Stop-Service -Name $nombre}
                "N" {Write-Host "No se han realizado modificaciones"}
            }
        }
    }
    else
    {
        Write-Host "No existe ningún proceso con este nombre. Vuelva a
intentarlo"
    }
} while ($compservicio -eq 0)
```

Haré la comprobación con "ALG"

```
PS C:\Windows\system32> get-service

Status      Name                DisplayName
-----
Stopped     AarSvc_10d5dc       Agent Activation Runtime_10d5dc
Stopped     AJRouter            Servicio de enrutador de AllJoyn
Stopped     ALG                 Servicio de puerta de enlace de niv...
Stopped     AppIDSvc            Identidad de aplicación
Running     AppInfo             Información de la aplicación
```

```
PS C:\Windows\system32> C:\Users\asir27\Desktop\script1.ps1
Dime el nombre de un servicio: ALG
Este proceso se encuentra parado. ¿Desea iniciarlo? Pulsa Y para iniciarlo y N para no iniciarlo: Y
Iniciando programa...
```

```
PS C:\Windows\system32> get-service
```

Status	Name	DisplayName
Stopped	AarSvc_10d5dc	Agent Activation Runtime_10d5dc
Stopped	AJRouter	Servicio de enrutador de AllJoyn
Running	ALG	Servicio de puerta de enlace de niv...

```
PS C:\Windows\system32> C:\Users\asir27\Desktop\script1.ps1
```

```
Dime el nombre de un servicio: ALG
```

```
Este proceso se encuentra iniciado. ¿Desea pararlo? Pulsa Y para pararlo y N para no pararlo: Y
```

```
Cerrando programa...
```

2. Hacer un script que nos pida el nombre de un fichero y nos muestre el contenido de u fichero en caso de existir o bien nos diga que no existe. Antes de acabar debe preguntarnos si deseamos volver a ejecutar el script y hacerlo en caso afirmativo.

```
do{
$fighero= Read-Host "Indícame el nombre del fichero"
$comrpobación = Test-Path $fighero
if ($comrpobación -eq "True")
{
write-host "El fichero existe"
type $fighero
}
else{write-Host "El archivo no existe"}
$repetir= Read-Host "Pulsa Y para repetir el script. Pulsa N para salir."
}
while($repetir -eq "Y")
```

```
PS C:\WINDOWS\System32> cd C:\Users\lasir-27\Desktop
PS C:\Users\lasir-27\Desktop> do{
$fighero= Read-Host "Indícame el nombre del fichero"
$comrpobación = Test-Path $fighero
if ($comrpobación -eq "True")
{
write-host "El fichero existe"
type $fighero
}
else{write-Host "El archivo no existe"}
$repetir= Read-Host "Pulsa Y para repetir el script. Pulsa N para salir."
}
while($repetir -eq "Y")
Indícame el nombre del fichero: examen.txt
El fichero existe
comprobaciÃ³n de que el script funciona.
Pulsa Y para repetir el script. Pulsa N para salir.: Y
Indícame el nombre del fichero: blablabla
El archivo no existe
Pulsa Y para repetir el script. Pulsa N para salir.: N
```

3. Obtener el nombre, tamaño y fecha (en ese orden) de todos los ficheros del directorio “C:\Instaladores” y sus subdirectorios que tengan más de 2GB de ordenados por tamaño.

```
Get-ChildItem -File -Recurse C:\Instaladores | Select Name, Length, LastWriteTime | where Length -gt 2000000000 | Sort -Property Length
```

Haremos la comprobación de que coincide el tamaño. Usaré “Kali-Linux.ova”

The screenshot shows a Windows Explorer window with the 'Instaladores' directory selected. The file list shows various ISO and OVA files. The 'Kali Linux.ova' file is highlighted, and its properties are shown in the right-hand pane. The properties pane shows the file size as 3,78 GB (4,063,621,632 bytes) and the location as C:\Instaladores. A yellow arrow points from the file size in the properties pane to the file size in the command prompt output below.

Command Prompt Output:

```
PS C:\Users\lasiir-27\Desktop> Get-ChildItem -File -Recurse C:\Instaladores | Select Name, Length, LastWriteTime | where Length -gt 2000000000 | Sort -Property Length
```

Name	Length	LastWriteTime
bee-box v1.6.ova	2051556864	28/10/2019 13:25:51
bee-box v1.6.ova	2051556864	18/02/2019 18:01:03
Win2012-12ee65d2.vmem	2147483648	26/03/2019 18:10:01
564d9438-4cda-7b55-2f06-4d084c735fd1.vmem	2147483648	16/01/2020 18:20:05
basic_pentesting_1.ova	2740361728	28/10/2019 13:19:44
basic_pentesting_1.ova	2740361728	09/01/2019 12:58:32
Win10-s005.vmdk	2790326272	17/12/2019 19:05:54
ubuntu-21.04-desktop-amd64.iso	2818738176	11/05/2021 19:19:51
Win10-s003.vmdk	2967797760	17/12/2019 19:18:08
Win10-c11-s003.vmdk	3043622912	16/01/2020 19:24:04
Win10-s003.vmdk	3077636096	31/10/2019 12:18:24
Win10-s003.vmdk	3106078720	26/03/2019 19:52:16
Win10-c11-s003.vmdk	3107520512	26/03/2019 20:45:07
Win10-c11-s005.vmdk	3393716224	17/12/2019 19:24:21
Win10-c11-s003.vmdk	3412197376	25/02/2019 21:11:50
Win10-s004.vmdk	3443392512	31/10/2019 12:18:24
Win10-s004.vmdk	3443392512	26/03/2019 19:52:16
kali-linux-2019.3a-vbox-amd64.ova	3570315776	28/10/2019 19:58:38
Win10-c11-s004.vmdk	3661496320	16/01/2020 19:23:55
Win10-c11-s004.vmdk	3661758464	26/03/2019 20:45:07
Win10-s003.vmdk	3749183488	17/12/2019 19:05:54
kali-linux-2018.4-vbox-amd64.ova	3763979776	13/12/2018 11:10:15
kali-linux-2019.2-vbox-amd64.ova	4008224256	28/10/2019 13:23:50
Kali Linux.ova	4063621632	13/03/2021 17:31:39
es_windows_8.1_enterprise_with_update_x64_dvd_6050578.iso	4174084096	30/11/2015 12:19:39
Win10-c11-s003.vmdk	4177133568	17/12/2019 19:25:29
Win10-c11-s001.vmdk	4245749760	16/01/2020 19:24:00
Win10-c11-s001.vmdk	4245749760	26/03/2019 20:45:07

Coincide.

4. Obtén los usuarios del sistema que hayan iniciado sesión alguna vez.

Primero, he realizado un `Get-LocalUser | Format-List` para ver la propiedad "LastLogon".

Todos aquellos que tengan una fecha en ese apartado serán los que necesito sacar en el listado. En este caso solo sería:

```
PS C:\Users\lasir-27\Desktop> Get-LocalUser | Format-List

AccountExpires      :
Description         : Cuenta integrada para la administración del equipo o dominio
Enabled             : True
FullName            :
PasswordChangeableDate : 09/05/2018 12:26:09
PasswordExpires     :
UserMayChangePassword : True
PasswordRequired    : True
PasswordLastSet     : 08/05/2018 12:26:09
LastLogon           : 03/09/2020 10:17:17
Name                : Administrador
SID                 : S-1-5-21-1125474266-2925866343-2985468450-500
PrincipalSource     : Local
ObjectClass         : Usuario

AccountExpires      :
Description         : Cuenta de usuario administrada por el sistema.
Enabled             : False
FullName            :
PasswordChangeableDate :
PasswordExpires     :
UserMayChangePassword : True
PasswordRequired    : False
PasswordLastSet     :
LastLogon           :
Name                : DefaultAccount
SID                 : S-1-5-21-1125474266-2925866343-2985468450-503
PrincipalSource     : Local
ObjectClass         : Usuario

AccountExpires      :
Description         : Cuenta integrada para el acceso como invitado al equipo o dominio
Enabled             : False
FullName            :
PasswordChangeableDate :
PasswordExpires     :
UserMayChangePassword : False
PasswordRequired    : False
PasswordLastSet     :
LastLogon           :
Name                : Invitado
SID                 : S-1-5-21-1125474266-2925866343-2985468450-501
PrincipalSource     : Local
ObjectClass         : Usuario

AccountExpires      :
Description         : Una cuenta de usuario que el sistema administra y usa para escenarios de Protección de aplicaciones de Windows Defender.
Enabled             : False
FullName            :
PasswordChangeableDate : 05/04/2018 16:15:52
PasswordExpires     : 01/10/2018 16:15:52
UserMayChangePassword : True
PasswordRequired    : True
PasswordLastSet     : 04/04/2018 16:15:52
LastLogon           :
Name                : WDAGUtilityAccount
SID                 : S-1-5-21-1125474266-2925866343-2985468450-504
PrincipalSource     : Local
ObjectClass         : Usuario
```

`Get-LocalUser | where LastLogon -gt 01/01/1900`

```
PS C:\Users\lasir-27\Desktop> Get-LocalUser | where lastlogon -gt 01/01/1900

Name      Enabled Description
----
Administrador True    Cuenta integrada para la administración del equipo o dominio
```