

Ejercicios scripts en PowerShell sobre redes

Estos ejercicios están diseñados para ayudarte a mejorar tus habilidades en PowerShell aplicadas a la administración de redes.

Ejercicio 1

Descripción: Escribe un script de PowerShell que obtenga la configuración de la dirección IP de todas las interfaces de red del sistema.

Solución:

```
# Obtener la configuración de la dirección IP de todas las interfaces de red
Get-NetIPAddress
```

Ejercicio 2

Descripción: Crea un script de PowerShell que realice un ping a una lista de direcciones IP o nombres de host y guarde los resultados en un archivo de texto.

Solución:

```
param (
    [string[]]$targets,
    [string]$outputFile = "PingResults.txt"
)

# Verificar cada destino y guardar el resultado en un archivo de texto
foreach ($target in $targets) {
```

```
$result = Test-Connection -ComputerName $target -Count 1 -
ErrorAction SilentlyContinue
if ($result) {
    "$target está en línea." | Out-File -FilePath
$outputFile -Append
} else {
    "$target no responde." | Out-File -FilePath
$outputFile -Append
}
}
```

Write-Output "Resultados guardados en \$outputFile"

Ejercicio 3

Descripción: Escribe un script de PowerShell que obtenga la tabla de enrutamiento del sistema y la guarde en un archivo CSV.

Solución:

```
# Obtener la tabla de enrutamiento y guardarla en un archivo
CSV
Get-NetRoute | Export-Csv -Path "RoutingTable.csv" -
NoTypeInfoInformation

Write-Output "Tabla de enrutamiento guardada en
RoutingTable.csv"
```

Ejercicio 4

Descripción: Crea un script de PowerShell que muestre todas las conexiones TCP activas en el sistema, incluyendo las direcciones IP locales y remotas, y los números de puerto.

Solución:

```
# Obtener todas las conexiones TCP activas
```

```
Get-NetTCPConnection | Select-Object -Property LocalAddress,
LocalPort, RemoteAddress, RemotePort, State
```

Ejercicio 5

Descripción: Escribe un script de PowerShell que configure una dirección IP estática en una interfaz de red específica. El script debe aceptar la dirección IP, la máscara de subred y la puerta de enlace como parámetros.

Solución:

```
param (
    [string]$interfaceAlias,
    [string]$ipAddress,
    [string]$subnetMask,
    [string]$gateway
)

# Configurar la dirección IP estática
New-NetIPAddress -InterfaceAlias $interfaceAlias -IPAddress
$ipAddress -PrefixLength $subnetMask -DefaultGateway $gateway

Write-Output "Dirección IP $ipAddress configurada en la
interfaz $interfaceAlias"
```

Ejercicio 6

Descripción: Escribe un script de PowerShell que obtenga la lista de todos los adaptadores de red y sus estados actuales (activado o desactivado).

Solución:

```
# Obtener la lista de todos los adaptadores de red y sus
estados actuales
Get-NetAdapter | Select-Object -Property Name, Status
```

Ejercicio 7

Descripción: Crea un script de PowerShell que cambie las direcciones de servidor DNS en una interfaz de red específica. El script debe aceptar las nuevas direcciones de servidor DNS como parámetros.

Solución:

```
param (
    [string]$interfaceAlias,
    [string[]]$dnsServers
)

# Configurar las nuevas direcciones de servidor DNS
Set-DnsClientServerAddress -InterfaceAlias $interfaceAlias -
ServerAddresses $dnsServers

Write-Output "Servidores DNS configurados en la interfaz
$interfaceAlias: $($dnsServers -join ', ')"
```

Ejercicio 8

Descripción: Escribe un script de PowerShell que monitoree el uso de ancho de banda de una interfaz de red específica y muestre el tráfico de entrada y salida en tiempo real.

Solución:

```
param (
    [string]$interfaceAlias,
    [int]$interval = 5
)

while ($true) {
    $stats = Get-NetAdapterStatistics -Name $interfaceAlias
    $received = [math]::round($stats.ReceivedBytes / 1MB, 2)
    $sent = [math]::round($stats.SentBytes / 1MB, 2)
```

```
    Write-Host "Tráfico de entrada: $received MB, Tráfico de
salida: $sent MB"
    Start-Sleep -Seconds $interval
}
```