Ejercicios de seguridad: analizar información sobre procesos de forma detallada en Windows con PowerShell

¿Cómo saber si un proceso falla en Windows con PowerShell? Tener en cuenta

- Un proceso que no se comporta como siempre es raro.
- Un proceso que abre muchos hilos es raro.
- Analizar los hilos de un proceso en concreto
- Un proceso que abre muchas comunicaciones es raro.
- Un proceso que consume fuera de normal es raro.
- Un proceso que se activa a una hora extraña es raro.
- Un proceso cuya firma ha cambiado es extraño.
- Mostrar los procesos que utiliza un controlador

Scripts que resuelven cada uno de los apartados anteriores

Un proceso que no se comporta como siempre es raro

```
$proceso1=(gps).name
start notepad
Start-Sleep -Seconds 3
$proceso2=(gps).name
Compare-Object -ReferenceObject $proceso1 -DifferenceObject $proceso2
```

Un proceso que abre muchos hilos es raro

```
Get-Process | %{
  $_.name, $_.Threads.Count
}
```

Analizar los hilos de un proceso en concreto

```
# Realizar una función que muestre información sobre los hilos de un proceso.
function hilos($name){
    $processHandle = (Get-Process -Name $name).id
    $Threads = Get-WmiObject -Class Win32_Thread |
    Where-Object { $_.ProcessHandle -eq $processHandle }
    "The $name process has $($threads.count) threads"
     $threads | Format-Table -Property priority, thread*, User*Time, kernel*Time
}

hilos notepad
```

Un proceso que abre muchas comunicaciones es raro (solución compleja)

```
function monitorizar
{
  param()
  begin
  {
  }
  process
  {
    foreach($conex in @(netstat -ano))
    {
        foreach($proceso in ps | where id -EQ ((Get-Process -name chrome | sort cpu -Descending | select -First 2)[0].id))
        {
            if($conex -match $proceso.Id -and $proceso.Id -ne 4 -and $proceso.Id -ne 0)
```

```
            {
                 write-host $conex "PROCESO: " $proceso.Name
$proceso.Id
            }
        }
    }
  }
  end
  {
  }
}

monitorizar
```

Un proceso que consume fuera de lo normal es raro

```
# Realizar una función de muestre el consumo de procesador analizando dos valores (consumo -cpu 20 -id 6000).
# Sirve para ir de uno en uno
(Get-Process | select $proceso).cpu -gt $valor

# Varios valores
Get-Process | Where-Object cpu -gt 10 | Where-Object id -gt 6000
```

Un proceso que se activa a una hora extraña es raro

```
Get-EventLog -LogName Security -InstanceId 4688
Get-EventLog Application | Where-Object Message -Match "explore"
```

Un proceso cuya firma ha cambiado es extraño

```
# Realizar una función que permita detectar que hay una firma que ha cambiado en un programa.
# Función simple para almacenar un hash de un programa
function hashear($ruta){
```

```
    Get-FileHash $ruta
}

(hashear 'C:\Windows\System32\notepad.exe' | select hash).hash
| Out-File hash.txt

(gc hash.txt)

Get-Process | select Path | %{
    hashear $_.Path
}

Get-Process | select Path | %{
    (hashear $_.Path).hash
}

Get-Process | select Path | %{
    if ((hashear $_.Path).hash -match (gc hash.txt))
    {
        "Igual",$_.path
    }
}
```

Mostrar los procesos que utiliza un controlador

```
# Realizar una función que muestre los procesos que utilizan
un controlador.
Get-Process -Module | select-string "OLEAUT32.dll"

Get-Process -Module | %{
    $_ | select-string "OLEAUT32.dll"
}

Get-Process | select name,Modules  | %{
    $_.name
    $_.Modules.ModuleName | select-string "OLEAUT32.dll"
}

Get-Process | select name,Modules | %{
    if($_.Modules.ModuleName -eq "OLEAUT32.dll")
```

```
    {
        $_.Name,$_.Modules.ModuleName
    }
}
```