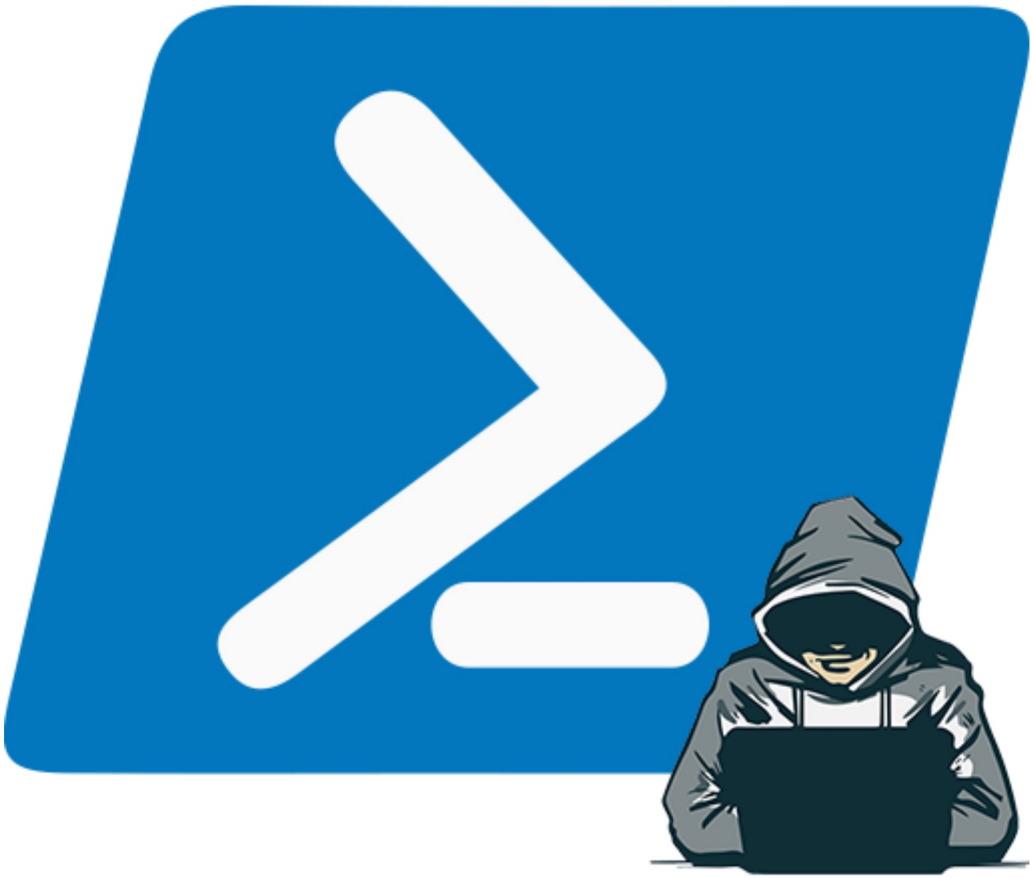
infocysec.com

# PowerShell para Pentesters

*InsiderHack*

6-7 minutos

---



PowerShell es un marco de administración, configuración y automatización de tareas que consta de una shell de línea de comando y lenguaje scripting, este se basa en .NET Common Language Runtime (CLR) que acepta y devuelve objetos, esto le da mucho poder a todos los que nos dedicamos al Pentesting. Los comando de PowerShell se conocen como **cmdlets** y se los

puede utilizar por separado, pero su eficacia radica en la unión o combinación de estos cmdlets para realizar tareas más complejas.

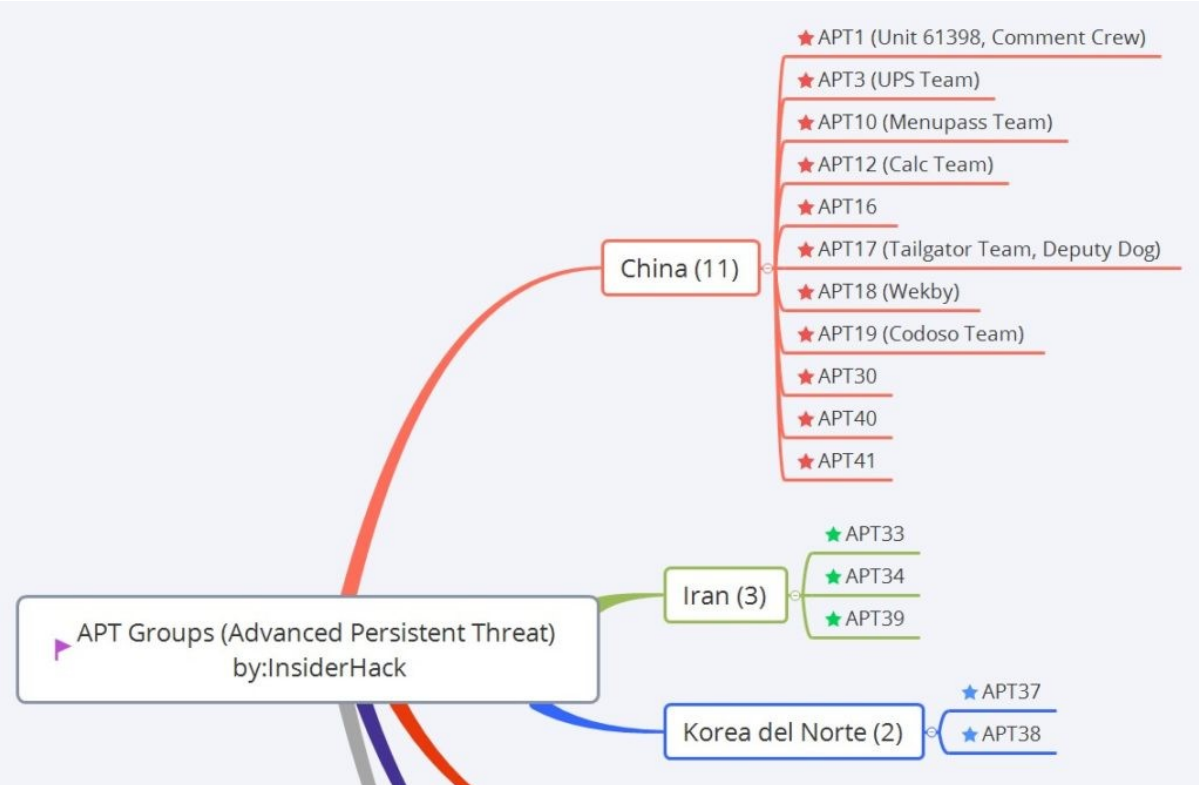
## ¿Porque PowerShell?

En la actualidad PowerShell es utilizado por:

1. Administradores de sistemas y programadores.
2. Expertos en seguridad (RedTeam, BlueTeam, Pentesters, etc)
3. Crimen organizado cibernético.
4. Hacktivistas.

La gran mayoría de los ataques informáticos a través de malware ejecutan en su payload PowerShell, ofuscando su código en Base64 para no ser detectados junto con otras técnicas de evasión, tan popular se ha vuelto que en Metasploit se puede crear payload en este marco.

Muestro en la gráfica algunos grupos APT separados por país que realizan varios tipos de ataques utilizando para la persistencia código en PowerShell como el caso de APT39.




Por este motivo muchas empresas no solo instalan antivirus tradicionales en sus endpoint si no también utilizan soluciones conocidas como EDR (Endpoint Detection and Response) para mejorar las respuestas de un inminente ataque y es aquí donde entra el equipo de Respuesta a Incidentes sea interno o externo.

Como vemos el aprender PowerShell se convierte en algo necesario en el dia a dia, no solo para Pentester si no también para todos los profesionales que trabajan defendiendo sus sistemas, con Powershell podemos correr, descargar, ejecutar código en memoria y esto lo hace muy útil.

Entre las ventajas que tenemos:

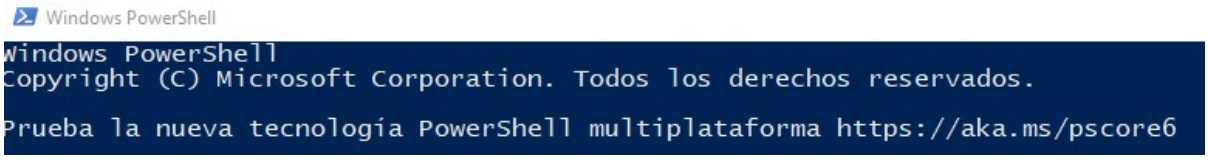
1. Llamar a funciones DLL de Windows.
2. No es necesario instalar ningún otro aplicativo, para usarlo viene por defecto en Windows 10, Windows 7 hasta Windows 2008 R2, las últimas versiones de PowerShell Core 6 (Open Source) en adelante introdujo soporte para macOS y Linux, puedes descargar el Core 6 Open Source en https://github.com/powershell/powershell.
3. Ya existen varias herramientas codificadas en PowerShell que podemos utilizar para realizar Pentesting.

   Por ejemplo la herramienta PowerSploit de HarmJ0y (https://github.com/PowerShellMafia/PowerSploit) tan conocida es la herramienta que también se la nombra en Mitre ATT&CK https://attack.mitre.org/software/S0194/.

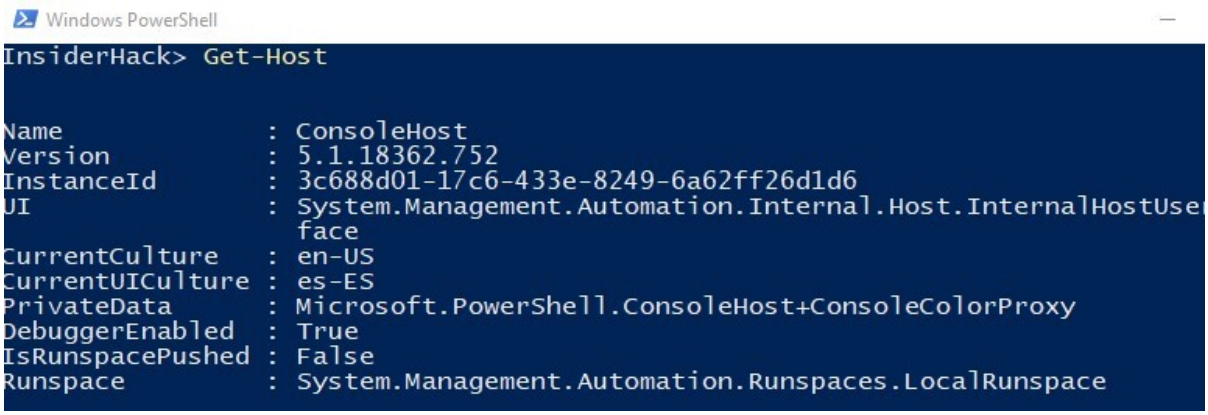
Esta herramienta es una colección de módulos de PowerShell que son usados por los Penetration Tester y bueno también por los piratas informáticos.

Para saber la versión de PowerShell que contamos en nuestro Windows presionamos la tecla **Windows + R** y escribimos **powershell** y se nos abre la ventana de fondo azul por el encabezado Windows PowerShell donde de bienvenida te muestra en consola que actualices a pscore6 https://aka.ms/pscore6.

También podemos ejecutarlos desde la consola de comandos colocando powershell.exe.



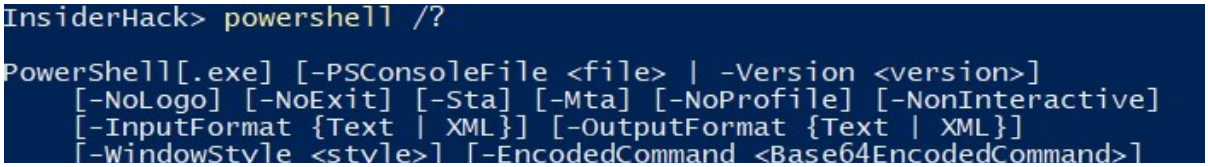

Ingresamos nuestro primer cmdlets para saber qué versión de PowerShell tenemos en nuestros equipos en este caso la versión 5.x

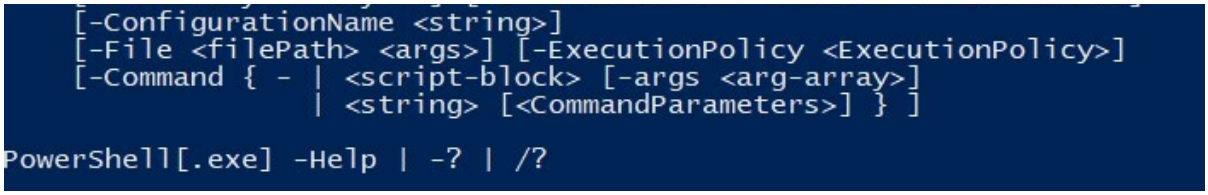



En las últimas versiones se han agregado algunas características de seguridad pero más adelante ya veremos como saltar esto.

Alguien por el año 2001 me dijo que el comando más importantes es el de ayuda así que vamos a ver cómo se utiliza.

**powershell /?**






Los parámetros vistos en la imagen son algunos de los PowerShell command line parameters que utilizaremos. PowerShell por defecto viene configurado para evitar la ejecución de scripts y para hacer un bypass de esto utilizaremos **-ExecutionPolicy.**

```
C:\> powershell.exe -ExecutionPolicy Bypass
.\script.ps1
C:\> powershell.exe -ExecutionPolicy Unrestricted
.\script.ps1
```

Es muy utilizado también **-WindowStyle** ya que con esto evitamos que se abra una ventana al momento de ejecutar cualquier script que realicemos.

```
C:\> powershell.exe -WindowStyle Hidden
.\script.ps1
```

Sin duda la ayuda en línea es algo que utilizarás a diario hasta que poco a poco vayas aprendiendo para qué sirve o cómo se trabaja cada instrucción y lo ejecutamos de la siguiente forma **Get-Help WindowsStyle -Online.**

El parametro -EncodedCommand es ampliamente utilizado para ejecutar Scripts en base64 realizando bypass de algunos antivirus.

```
C:\> powershell.exe -EncodedCommand $Encoded
```

Cuando se habla de perfiles nos referimos generalmente a las preferencias de un usuario y cuando hablamos de perfiles en PowerShell son scripts ejecutados automáticamente al arrancar la consola, estos perfiles pueden intervenir en nuestros propósitos al momento de ejecutar nuestro script y por tal motivo podemos utilizar **-NoProfile**.

```
C:\> powershell.exe -NoProfile .\script.ps1
```

Otro parámetro que será útil es **-Version** ya que si necesitamos que se ejecute nuestro script en versiones anteriores podamos hacerlo, en este caso estamos llamando a la versión 2 de PowerShell siempre y cuando esté instalada o incluso llamar a la versión 1.

```
C:\> powershell.exe -Version 2
```

Entre los parámetros vistos podemos trabajar con abreviaturas las cuales son utilizadas comúnmente en los scripts.

Entre los parámetros vistos podemos trabajar con abreviaturas las cuales son utilizadas comúnmente en los scripts.

| **Parámetro completo** | **Parámetro abreviado** |
|---|---|
| -ExecutionPolicy Bypass | -ep Bypass-ex by |
| -EncodedCommand | -enco-ec |
| -WindowStyle Hidden | -w h-wi hi |

Bueno después de haber visto algunos parámetros que deben de tener nuestros scripts hasta aquí esta primera parte, en la segunda comenzaremos a ver algunos cmdlets para nuestros propósitos.

Aquí dejo un link para que vayas ampliando tus conocimientos sobre PowerShell.

https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7