Simulación de persistencia maliciosa en Windows usando PowerShell
Luis Garcia - Ciberseguridad — Proyecto educativo 2025

🎓 Presentación personal

Mi nombre es **Luis Garcia**, soy estudiante de Ciberseguridad con orientación en análisis ofensivo y simulaciones de ataque ético. Este proyecto fue desarrollado como parte de mi formación práctica, buscando comprender técnicas reales utilizadas por atacantes y cómo defendernos de ellas desde una perspectiva educativa y legal.

Este informe refleja no solo una demostración técnica, sino también mi compromiso con la concientización y divulgación de buenas prácticas en seguridad informática.

Introducción:

En este proyecto se simuló una técnica de persistencia muy utilizada por ciberatacantes: la creación de una tarea programada en Windows que ejecuta un script malicioso al iniciar el sistema operativo.

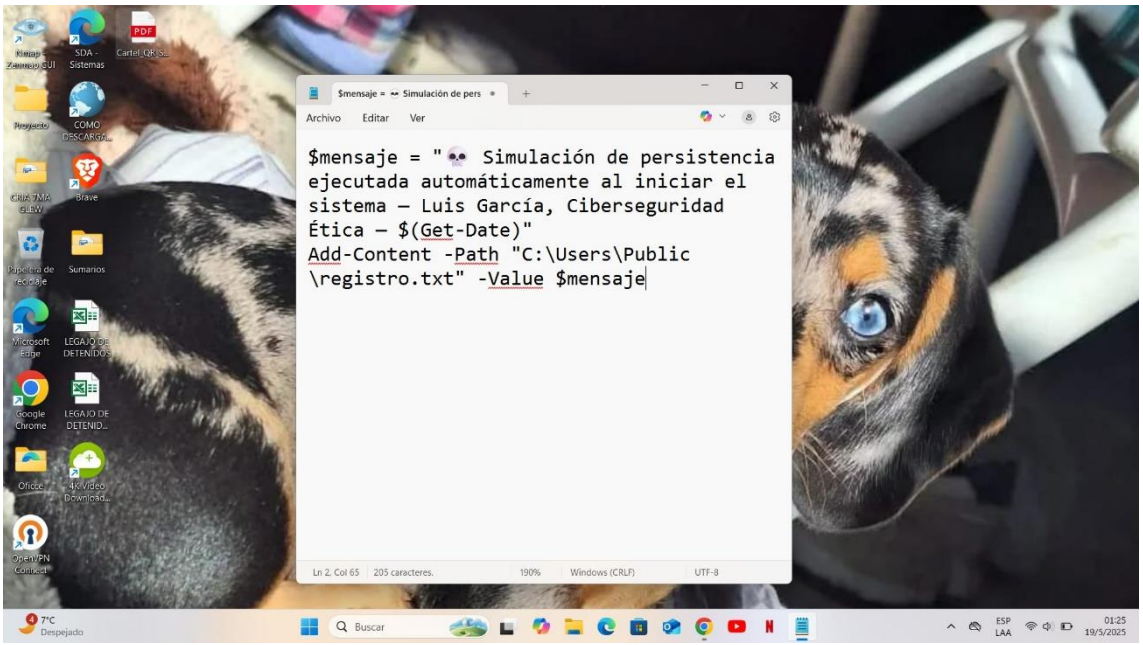
El objetivo es mostrar, paso a paso, lo que haría un atacante real y cómo funciona esta técnica para mantenerse dentro de una máquina comprometida.

❌ ATENCIÓN: Este proyecto se realizó con fines educativos y en entorno controlado. No se ejecutó ningún código dañino. El objetivo es concientizar sobre ciberseguridad.

Captura 1 - Creación del script malicioso

El script fue redactado en **Bloc de notas**, pero escrito en lenguaje **PowerShell**. Simula el comportamiento de un atacante que deja un "registro invisible" en un archivo de texto cada vez que se inicia el sistema. Esta técnica imita a malwares reales como **keyloggers** o **backdoors**.

Este tipo de acciones se pueden usar como parte de un acceso persistente tras una intrusión. Lo interesante es que puede pasar desapercibido para el usuario promedio.



Código usado:

```
$mensaje = "Simulacion de persistencia ejecutada al iniciar el sistema - Luis Garcia - $(Get-Date)"

Add-Content -Path "C:\Users\Public\registro.txt" -Value $mensaje
```

✅ En un entorno real, un atacante usaría este tipo de scripts para automatizar tareas ocultas, como iniciar conexiones a servidores maliciosos o registrar las pulsaciones del teclado.

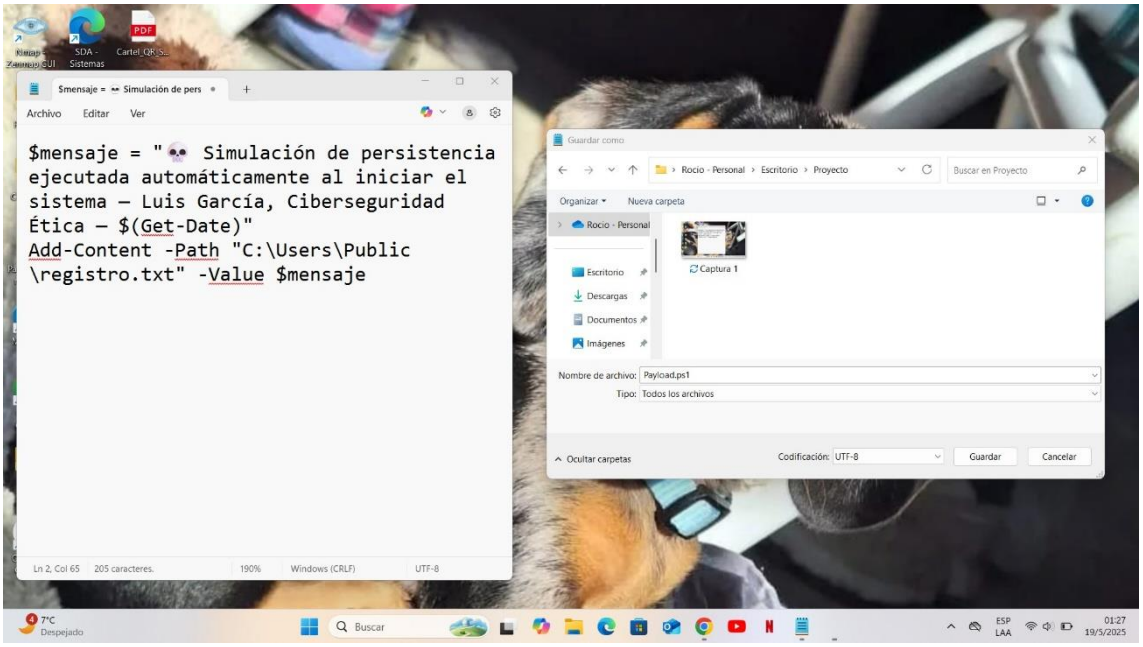
Captura 2 - Guardado del archivo `.ps1`

El script se guarda como Payload.ps1 en el disco C:\. Los atacantes suelen elegir ubicaciones comunes o de confianza para evitar levantar sospechas.

Guardar el archivo con extensión .ps1 permite que PowerShell lo interprete como código ejecutable y pueda ser lanzado automáticamente por tareas programadas.

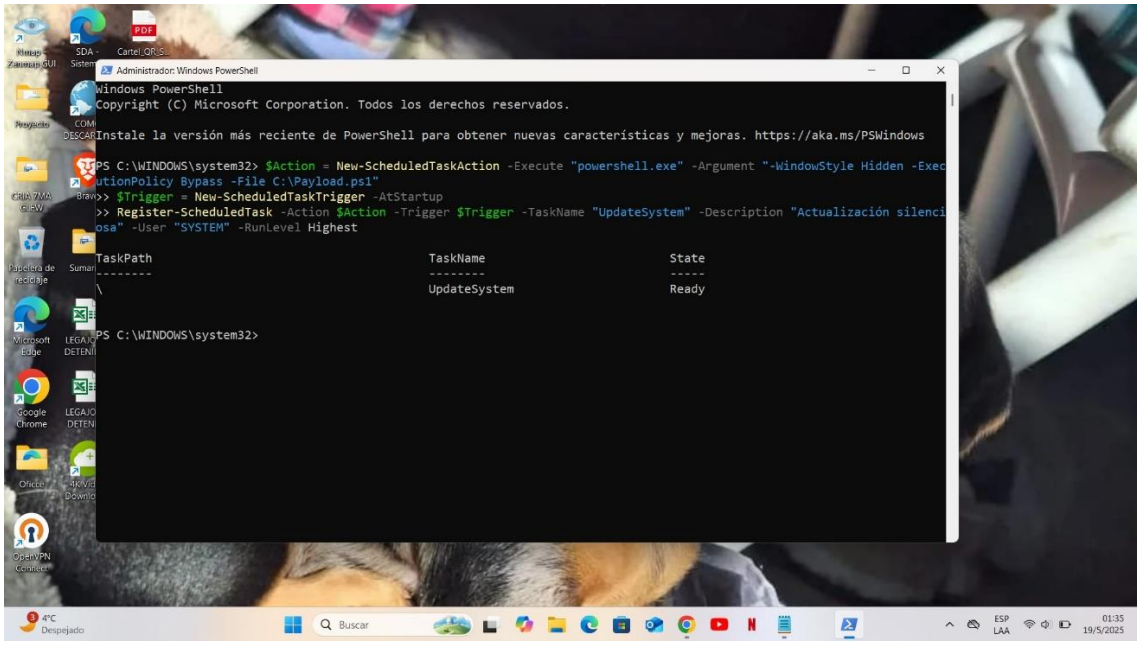
🧠 Esta elección de nombre y ruta también puede estar pensada para parecer parte del sistema o de una actualización legítima, dificultando su detección por parte de usuarios o incluso algunos antivirus.

Además, el atacante puede ocultar este archivo cambiando sus atributos para que sea invisible en el Explorador de archivos, o incluso integrarlo en scripts más grandes, como parte de una cadena de infección más compleja.



Captura 3 - Creación de la tarea oculta

Con PowerShell, se crea una tarea programada llamada UpdateSystem que se ejecuta al iniciar Windows. Esta tarea corre con privilegios SYSTEM y llama al script sin mostrar ventanas al usuario.

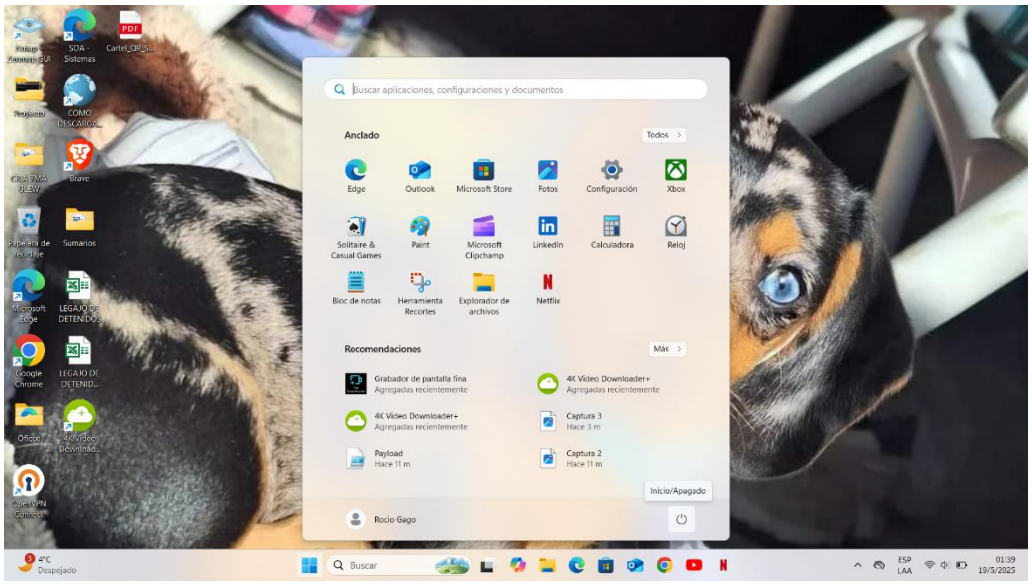


Comando clave:

Register-ScheduledTask -Action $Action -Trigger $Trigger -TaskName "UpdateSystem" -User "SYSTEM" -RunLevel Highest

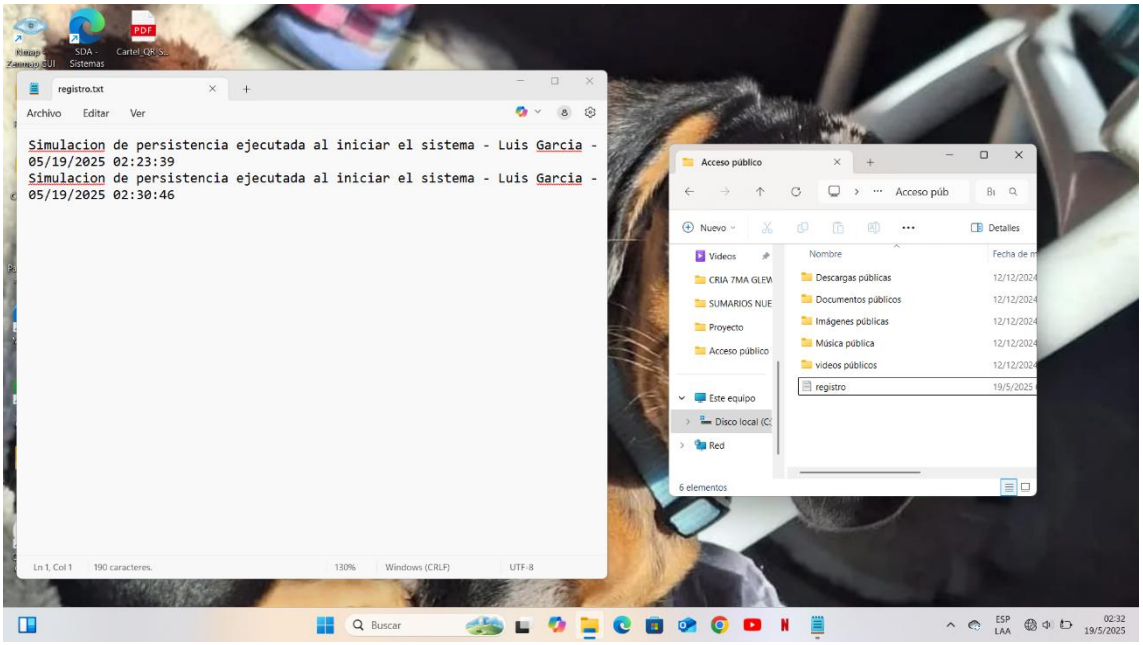
Captura 4 - Simulación del reinicio del sistema

Después de configurar todo, solo queda reiniciar o cerrar sesión. Si la tarea se ejecuta correctamente, significa que el código malicioso quedó instalado de forma persistente.



Captura 5 - Evidencia del registro creado

Al reiniciar la computadora, el script se ejecuta automáticamente y deja una nueva línea en registro.txt, con la fecha y hora exactas de inicio.



Este comportamiento puede parecer inofensivo, pero en un ataque real, esa línea podría ser un volcado de contraseñas o pulsaciones de teclas (keylogger).

Como detalle adicional, la hora del registro generado (02:30:46) coincide exactamente con la hora que marcaba el sistema al momento del reinicio. Esto sirve como evidencia

concreta de que la simulación de persistencia fue ejecutada automáticamente y sin intervención del usuario.

🚫 Lo que haría un atacante real

Un cibercriminal, en vez de dejar una simple línea de texto, podría usar este mismo mecanismo para:

- Ejecutar un **keylogger** y capturar todo lo que escribís
- Levantar una **shell remota** para controlar tu PC a distancia
- Descargar y ejecutar **malware adicional** desde servidores externos

Y todo esto sin que el usuario vea una sola ventana.

🔒 Conclusión

Este experimento muestra lo fácil que es automatizar la ejecución de un script en Windows usando herramientas nativas como PowerShell y el Programador de Tareas.

Como profesionales y aprendices de la ciberseguridad, tenemos que conocer estas técnicas para poder detectarlas, auditarlas y defendernos de ellas.

Luis Garcia
Estudiante y practicante de Ciberseguridad ética
