	HTB - Challenge: Alien Cradle	
	Dificultad:	**Very Easy**
	Release:	**01/03/2023**
	Skills Learned	
	• Familiarity with PowerShell deobfuscation	

En el presente análisis técnico abordo la resolución del reto **Alien Cradle** de Hack The Box, un ejercicio orientado a evaluar la capacidad del analista para identificar, desofuscar y comprender artefactos ejecutables en entornos Windows. El desafío se articula en torno a un script de PowerShell deliberadamente manipulado para ocultar su lógica interna y dificultar la trazabilidad de su flujo de ejecución, reproduciendo con notable fidelidad las técnicas de evasión y persistencia empleadas en escenarios reales de intrusión. A lo largo del write-up desarrollo un proceso metodológico de inspección, desofuscación y reconstrucción del comportamiento del script, poniendo de relieve tanto la infraestructura remota empleada por el atacante como los mecanismos de validación, carga en memoria y ejecución encubierta del payload. El objetivo es evidenciar un enfoque analítico riguroso, fundamentado en buenas prácticas de laboratorio seguro y en una comprensión profunda de los vectores de abuso más frecuentes en entornos corporativos.

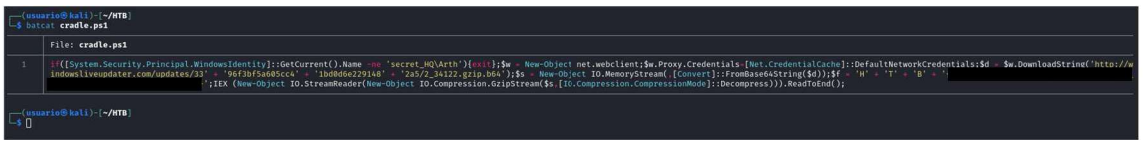
Enumeración

El archivo facilitado corresponde a un script de PowerShell que, a primera vista, implementa una técnica habitual en escenarios de ejecución remota de código: el uso de un comando legítimo para descargar y ejecutar un payload malicioso desde un servidor controlado por el atacante. No obstante, en este caso concreto, el adversario ha recurrido a un nivel significativo de ofuscación con el propósito de dificultar tanto su detección como su análisis estático.

Durante la inspección preliminar del script se observa que la lógica de ejecución incorpora varios mecanismos de control y preparación del entorno. En primer lugar, el código verifica que el usuario en sesión coincida con *secret_HQ\Arth*; en caso contrario, aborta inmediatamente la ejecución, lo que sugiere una medida de restricción operativa destinada a evitar la activación del payload fuera del entorno previsto por el atacante. A continuación, el script instancia un objeto WebClient y configura las credenciales del proxy utilizando las credenciales de red predeterminadas del sistema, una técnica común para camuflar la comunicación saliente dentro del tráfico legítimo del host comprometido.

Posteriormente, el script procede a recuperar desde un recurso remoto una cadena codificada en Base64, la cual es decodificada y almacenada en un flujo de memoria para su posterior tratamiento. De forma paralela, se asigna a una variable interna —presumiblemente $f— un valor igualmente ofuscado, cuyo propósito no es evidente en esta fase inicial del análisis. Finalmente, el contenido decodificado es descomprimido y ejecutado en memoria, completando así la cadena de infección sin dejar artefactos evidentes en disco.

Resulta evidente que los elementos críticos del flujo —concretamente la obtención del recurso remoto y la asignación del valor almacenado en $f— han sido deliberadamente ofuscados. El siguiente paso consiste, por tanto, en desentrañar estos fragmentos para identificar la infraestructura del atacante y determinar la naturaleza del contenido encapsulado en la variable $f.



Solución

Para proceder a la desofuscación del script, la vía más directa y controlada consiste en emplear el propio intérprete de PowerShell. Aunque el contenido analizado no presenta comportamiento malicioso y el endpoint desde el que se servía originalmente se encuentra inactivo, resulta imprescindible mantener una disciplina operativa estricta; por ello, todas las pruebas se realizan dentro de una máquina virtual Windows aislada y dedicada exclusivamente a tareas de análisis.

La estrategia de desofuscación se basa en reconstruir las cadenas originales a partir de los fragmentos codificados presentes en el script. Para ello, basta con asignar la cadena ofuscada a una variable y solicitar su representación en claro mediante el intérprete. Este procedimiento permite recuperar, sin alterar el flujo lógico del script, la URL remota empleada por el atacante. Aplicando la misma técnica sobre el contenido almacenado en la variable $f, es posible extraer el valor íntegro que esta encapsula, revelando finalmente la flag asociada al reto.